FACTA UNIVERSITATIS
Series: **Electronics and Energetics** Vol. 29, No 2, June 2016, pp. 269 - 283
DOI: 10.2298/FUEE1602269M

DESIGN OF IIR DIGITAL FILTERS WITH CRITICAL MONOTONIC PASSBAND AMPLITUDE CHARACTERISTIC - A CASE STUDY

Dejan Mirković, Miona Andrejević Stošović, Predrag Petković, Vančo Litovski

University of Niš, Faculty of Electronic Engineering, Serbia

Abstract. *A case study is reported related to the design of IIR digital filters exhibiting critical monotonic amplitude characteristic (CMAC) in the pass band. This kind of amplitude characteristic offers several advantages as compared to its non-monotonic counterparts, although it has not been studied thoroughly so far, if at all. After giving a short overview of the way of CMACs generation, arguments will be listed in favor of the IIR version of the digital filter function realization. Next, the IIR implementation of the digital filters will be considered in short. The main part of the paper will be devoted to the design sequence of this kind of filters which will be illustrated on the example of a band-pass filter obtained by a set of transformations from an all-pole low-pass analogue prototype. This will be the first time a CMAC band-pass IIR digital filter is reported.*

Key words: *Digital filters, IIR, monotone amplitude characteristic, all-pole filters*

1. INTRODUCTION

The critical monotonic amplitude characteristic (CMAC) filters represent an extension of the broad family of filtering functions having all transmission zeroes at infinity [1]. They exhibit distinctive properties such as monotonic amplitude response in the pass band, reduced group delay distortions, higher symmetry of the pulse response, improved mapping of tolerances, improved sensitivity, and high selectivity.

The interest for a digital realization of this kind of filtering functions comes from several reasons. First of all, only one sub-class of these functions has already been published in its digital form, the Butterworth filters [2]. As shown in [1] and elsewhere, however, practically all sub-classes of CMAC functions outperform the Butterworth solution in almost every aspect of implementation with the exception of function's simplicity. This study is a part of our effort to make CMACs more popular and to help bridging the gap between designers and CMAC which has deepened during time [3]. Second, due to their monotonic behavior, their sensitivity in the passband is reduced and accordingly, they

Received April 24, 2015; received in revised form August 3, 2015
Corresponding author: Dejan Mirković
Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia
(e-mail: dejan.mirkovic@elfak.ni.ac.rs)

offer a good alternative to their non-monotonic counterparts (e.g. Chebyshev and Least *p*-th [4]). At the same time, this means an improvement in the mapping of the tolerances of the circuit parameters into the tolerances of the attenuation characteristic [5]. Finally, they exhibit smaller distortions of the passband group delay which reduces the complexity of the potential phase-corrector to be used to flatten the group delay characteristic [6]. This also means that CMAC have smaller asymmetry of the response to a Dirac pulse in the time domain which may be of crucial importance for some applications in telecommunication and signal processing.

It is our opinion that the advantages of CMAC filtering functions have not been completely understood in the research and design community. That especially stands for the IIR implementation where no instances of implementation of CMAC may be found. The reason for that, in our opinion, is inertia and the need of some additional (mathematical) knowledge for generation of the CMAC transfer functions as compared with the Chebyshev and Butterworth filters. Here we try to reopen the subject of CMAC design by reporting the results related to the design of band-pass digital IIR filter which is the first implementation of band-pass CMAC of all.

Being a designer, one is first to decide either to go for FIR filters and start the synthesis of transfer functions for each type of CMAC from scratch, or to go for IIR filters and transform the existing analog data into the digital domain. In the text below, a short paragraph is devoted to help the decision. As a conclusion, the designer will be advised to go for an IIR filter with parallel implementation as the most economical solution in almost every respect.

Next, one is to create the CMAC transfer function and to choose among sub-classes. Again, a short paragraph will be devoted to this issue. Four main sub-classes of CMAC will be described from the implementation point of view. Corresponding transfer function generation will be discussed shortly.

Based on these, a design sequence will be advised for finding the coefficients of the transfer function of IIR filters in the z-domain. Note that parallel implementation will be recommended and all the calculations will be performed under that presumption. The transformed function will be studied from both stability and accuracy point of view.

The procedure will be exemplified on the case of a band-pass IIR filter. To get it, a low-pass to band-pass transformation was performed in the analog domain. In that way the analog prototype so obtained was to be transformed into the z-domain by bilinear transformation. The implementation obtained in this way was evaluated by simulation of a filter excited by a complex signal in the time domain. Various possible computing technologies were taken into account by changing the number of significant figures for the computations in order to establish the most economical implementation satisfying the design requirements.

The paper is organized as follows. In the second paragraph arguments will be given for adopting IIR digital filters. In the third paragraph the CMAC function will be introduced. Then, in the fourth paragraph, the bilinear transform implementation to a parallelized analog transfer function will be given. The case study describing the design (and its verification) of a band-pass CMAC filter will be given in the fifth paragraph.

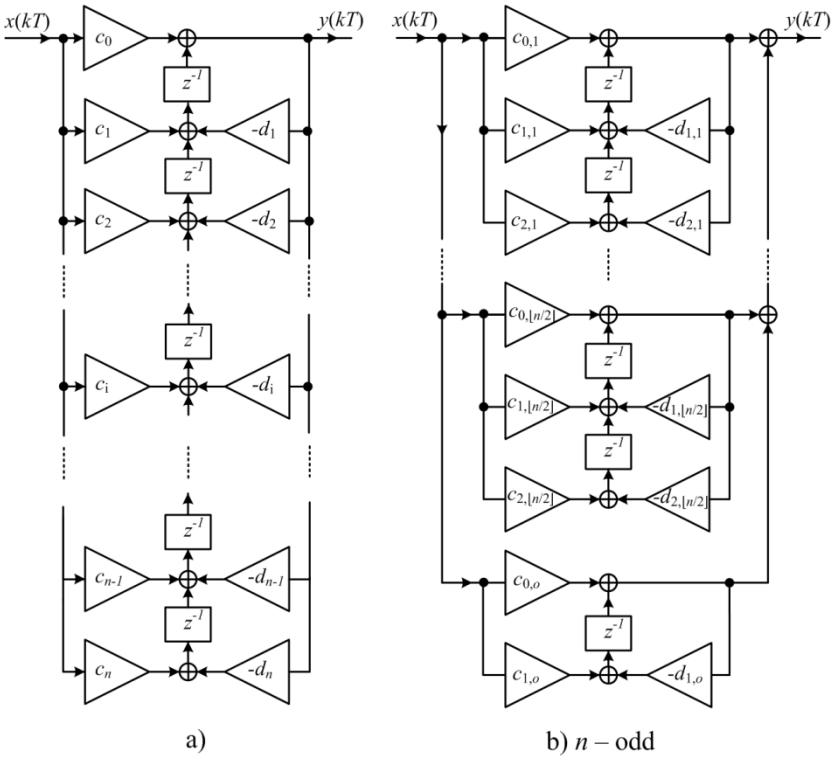
2. PROPERTIES OF THE IIR DIGITAL FILTERS

In digital filter design, one is to decide first on the choice between FIR and IIR filter functions and then to proceed to the approximation problem. Then, one is to choose among

different structures exhibiting the same transfer function. In the case of digital filters, the choice is to be done between the canonical (or state variable) and the parallel form. These two are illustrated in Fig. 1 for an IIR digital filter. It should be noted that if the order of the filter, *n*, is even, first order cell at the bottom of Fig. 1b is omitted, leaving only second order cells in filter realization.

When taking the decision between FIR and IIR filters one has to have in mind several criteria such as complexity of the solution, stability of the system, and processing time.

The first criterion may be fragmented into several having the same origin. Namely, the complexity of the solution will influence the power consumption, the silicon area and the design effort especially when special techniques are to be implemented for reduction of the power consumption [7].



a) b) *n* – odd

Fig. 1 Realization of an *n*th order IIR filter, a) canonical b) parallel (for *n* odd)

Note that not all of the criteria are of equal weight in design. For some applications the latency, i.e. the computational time may be of prime importance since it allows for speed. In others, reduction of heating or silicon area may prevail as a main criterion. Putting all together, the choice is to be made by taking into account several, if not all criteria. In our detailed study [3] we came to the following.

The use of IIR filter has the following advantages: 1. Lower complexity (in some cases, e.g. [8], incomparably lower); 2. Lower dissipation; 3. Lower silicon area; 4. Available analog prototypes to transform.

The use of FIR filters has the following advantages: 1. Lower latency; 2. Easier synthesis of linear phase filters; 3. Better stability.

The use of parallel architecture for the IIR filters as shown in Fig. 1b however, mitigates all disadvantages (stability, latency) of the IIR filters, while there are no methods to do the same for the FIR counterparts. It is to note here that getting a linear phase by FIR filters doubles the complexity of the solution while using a phase corrector for the IIR solution contributes marginally to its complexity [8].

That was the reason why we adopted the parallel architecture and the IIR filter structure for the implementation in the CMAC design.

3. CMAC FILTERS IN THE *s*-DOMAIN

Polynomial (or all-pole) filters with critical monotonic amplitude characteristics (CMAC) in the passband have been available for several decades now [1]. The main property of CMAC is related to the critical monotonicity of the amplitude response in the passband which will be first described here in short.

The squared amplitude characteristic may be expressed as

$$|H(\mathrm{j}\omega)|^2 = 1/\{1+K(\omega^2)\} \tag{1}$$

where $K(\omega^2)$ is the characteristic function. In the simplest form (as proposed in [9]), for n even, one has:

$$K(\omega^2) = \varepsilon^2 L_n(\omega^2) = \varepsilon^2 \frac{\prod_{i=1}^{\lfloor n/2 \rfloor}(\omega^2-\omega_i^2)^2}{\prod_{i=1}^{\lfloor n/2 \rfloor}(1-\omega_i^2)^2}, \tag{2}$$

where ω is the normalized angular frequency, n is the order of the filter, ε defines the insertion loss at the passband edge, i.e. $\varepsilon^2 = 10^{a_{\max}/10} - 1$, $0 < \omega_i < 1$, i=1,2, ..., $\lfloor n/2 \rfloor$ are the abscissa of the inflection points, $a_{\max}$ is the maximum allowable attenuation (in dB) in the passband, and $\lfloor\ \rfloor$ denotes the floor function. $L_n(\omega^2)$ is a polynomial with n second order real zeroes located in the interval (−1,1).

Since the characteristic function has a maximum number of inflection points in the passband, so do the amplitude characteristic and the attenuation, the last one being defined as

$$a(\omega^2) = 10 \cdot \log(1/|H(\mathrm{j}\omega)|^2) \ \ [\mathrm{dB}] \tag{3}$$

The main property of CMAC leads to a good mapping of the element tolerances into the tolerances of the attenuation. Namely, as shown in [4], the tolerance of the attenuation may be expressed in the following form:

$$\Delta a = \frac{\Delta x_i}{x_i} \omega \frac{\partial a}{\partial \omega}, \tag{4}$$

where x_i is the ith parameter of the analog circuit. Having a maximal number of inflexion points (where both the first and the second derivatives are equal to zero) of the amplitude

characteristic in the passband, the CMAC forces the left-hand side of (4) to go through zero a maximal number of times. Note, the derivative of *a* in (4) does not change its sign if CMAC is used since it is monotonic which is different to the non-monotonic functions, e.g. C and LS.

Filters exhibiting CMAC characteristic are also known to have lower group delay distortions in the passband than their C and LS counterparts [10].

Altogether, the existence of CMAC gives to the filter designer an additional freedom in the choice of the best solution for a filter design problem.

There are four main classes of CMAC as discussed in [1] and [10]. They originate from the design criteria implemented for synthesis of the transfer function. These criteria are:

1. Maximally flat in the origin. The class of filters thus obtained is called Butterworth's after the author [11]. These will be here referred to as B-filters.
2. Maximum slope of the characteristic function at the edge of the passband [12] [13] [14]. The name L-filters comes from the fact that for the original derivation Legendre polynomials were used.
3. Maximum asymptotic attenuation. [15]. Here, these will be referred to as H-filters.
4. Least-squares-monotonic. In this case, the reflected power in the pass-band was minimized under the critical monotonicity criterion [16] and named LSM filters.

A catalog of the coefficients of the transfer functions of all four classes of CMAC for *n* up to 10, obtained by these criteria, was published in [1] where a comparative study was also given. To illustrate this here, Fig. 2 depicts the passband attenuation characteristics of the above four classes for n=7.

In the next section, before proceeding to CMAC digital filter design, the arguments for using IIR filters will be discussed in short as based on the comparison of the properties of FIR and IIR filters.

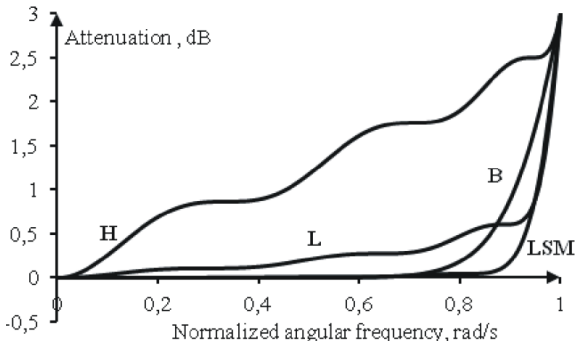


Fig. 2 The four main CMAC approximants for *n*=7

4. THE BILINEAR TRANSFORM AND CMAC IN THE *z*-DOMAIN

There are several transformations claiming to preserve some of the original properties of the analog filter function when producing a digital domain counterpart. As listed in [17], these methods may be categorized in two groups. In the first group are put the ones which implement a specific criterion of approximation such as: the Impulse Response

Invariant method; the Modified Impulse Response Invariant method; the Step Response Invariant method (or Zero Order Hold); the Magnitude-invariance method; and the Phase-invariance method.

There are, however, transformations based on substitution of the complex frequency in the *s*-domain by an expression being a function of *z*. In that way, one has the Matched-*z* Transform method, and three methods obtained by approximation of the analog integrator by a digital one. These are known as the Backward Euler (backward difference); the Trapezoidal method or the Bilinear Transform method, discussed in [18], and the second order formula introduced in [17].

The most popular among all of these is the bilinear transform. Its main properties are simplicity of implementation and good preservation of the properties of the amplitude characteristic of the analog filter. It preserves the stability of the analog prototype. It introduces distortions (reduced by increasing the sampling rate) into the phase (group delay) characteristic which, however, has no importance in many applications. It is implemented via the following transformation into the analog transfer function:

$$s \Rightarrow \frac{2}{T} \cdot \frac{z-1}{z+1}. \tag{5}$$

where z is the complex digital angular frequency, and T is the sampling rate $T=1/f_S$, f_S being the sampling frequency. In that way

$$H_a(s) \Rightarrow H_a\left(\frac{2}{T} \cdot \frac{z-1}{z+1}\right) = H_d(z) \tag{6}$$

is obtained, where H_a stands for the transfer function of the analog filter, while H_d stands for that of the digital filter.

The procedure of implementation of the bilinear transform to a parallelized analog transfer function together with the stability analysis and numerical considerations were discussed in [3] and we will not repeat them here. Instead, in the sequel, we will go for the design of a band-pass filter obtained by low-pass-to-band-pass transformation in the *s*-domain and then transposed into the *z*-domain. It is our goal with that design to study all steps that remain to be performed in order to get an implementable design and to analyze the implementation problems related to the limited number of binary digits that arise in real life situations.

5. DESIGN, VHDL MODELING, AND SIMULATION OF CMAC IIR FILTERS

The following steps are to be performed in order to get an implementable design of the filter: creation of the band-pass filter in the *s*-domain; performing the *s*-to-*z* transform; conversion the decimal coefficient values into binary; scaling the coefficients to become implementable in fixed point arithmetic; and verification of the design by simulation of the filter hardware. Concurrently, based on transfer function evaluation performed after taking into account the finite number of digits used for the representation of the coefficients (after quantization), a final decision will be enabled about the acceptability of the given approximation, i.e. selected number of binary digits. That and scaling are steps of crucial importance for defining the quality of the final solution.

The example filter will be created based on the following requirements: a) Band-Pass (BP); b) Central frequency: f_0 = 3 kHz; c) Bandwidth: f_{BW} = 900 Hz; d) Sampling frequency: f_S = 50 kHz e) Order of the prototype low-pass filter n=7; f) Pass-band amplitude approximation LSM; g) *s*-to-*z* transform used: bilinear; h) Architecture: parallel combination of Transpose Direct Form II (TDF II) filter sections.

The well known [19] low-pass to band-pass transform was used:

$$\omega \leftarrow \frac{1}{BW_r} \cdot \frac{\Omega_0^2 - \Omega^2}{\Omega \cdot \Omega_0}, \tag{7}$$

where ω is the angular frequency of the prototype filter, while Ω is the angular frequency of the band-pass filter. Ω_0 is the central angular frequency, while BW_r=BW/ Ω_0. BW is the bandwidth of the filter expressed as angular frequency. After the substitution of s_{LP}=jω and s_{BP}=jΩ, (7) becomes a second order algebraic equation with complex coefficients which is usually solved by the Geffe algorithm [20]. The new function has fourteen poles obtained by solving (7) as depicted in Table 1 (together with the poles of the prototype LP LSM filter), and seven zeroes in the origin. In this case BW_r=f_{BW} / f_0 =0.3 and Ω_0=1 rad/s was used.

Table 1 Pole locations of the BP and LP LSM filter in the *s*-domain

	Band-pass		Low-pass	
No.	Real part	Imaginary part	Real part	Imaginary part
1/2	-0.08266346190	±0.99657751935	-0.1179475625	±0.9751626241
3/4	-0.02025317565	±1.15676424786	-0.3342221750	±0.7735798237
5/6	-0.01513109310	±0.86421546064	-0.4935853895	±0.4252967357
7/8	-0.05591895577	±1.12151432405	-0.5510897460	0.0
9/10	-0.04434769671	±0.88944037693		
11/12	-0.07876429908	±1.06309950994		
13/14	-0.06931131778	±0.93551048922		

Next, the transfer function of the band-pass filter was expressed as *a sum of partial fractions* to enable parallel realization and, before the bilinear s-to-z transformation was implemented; the poles of the band-pass filter were to be denormalized: every pole coordinate was multiplied by $2\pi \cdot f_0$. Based on this the coefficients of the biquads were calculated and *s*-to-*z*-domain mapping enabled. The resulting coefficients of the biquads in the z domain are given in the first row (entitled *Full precision*) of Table 2. This concludes the synthesis procedure.

We proceed now with the realization. As the first step, we encounter the necessity to express the coefficient values with a finite number of digits as physical implementation is expected. This process is usually referred to as quantization.

Only fixed point, two's complement, biquad's coefficients representation is considered. Fig. 3 shows the transfer function's pole locations in the z-plane for various binary word lengths used to represent the coefficient values. Note that for **all** cases the poles are confined within the unit circle which confirms our claim that parallel realization will mitigate stability problems in IIR realizations.

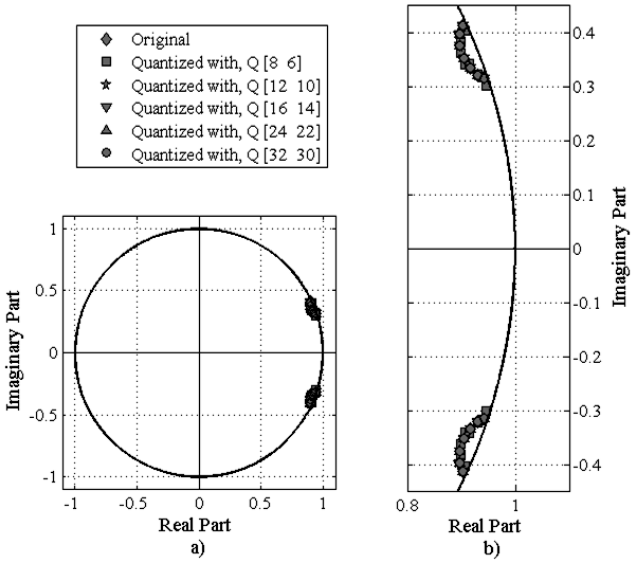


Fig. 3 *z* – plane pole location of the BP LSM filter, a) unit circle, b) zoomed poles location.

The following notation was used for the quantized version of the filter coefficients, Q[*N F*]. *N* stands for the number of bits of the whole digital word and *F* for the number of bits allocated for the digits after the decimal point. Accordingly, Q[*N F*] will populate the *range* (*RNG*) in increments (*INC*) as follows:

$$M = \lceil \log_2 (\lceil c_{\max} \rceil) \rceil + 1; \quad F = N - M \,, \tag{8a}$$

$$RNG = \left[-\frac{2^{N-1}}{2^F}, \frac{2^{N-1}-1}{2^F} \right]; \quad INC = \frac{1}{2^F} \,, \tag{8b}$$

where $c_{\max}$ is the coefficient with maximal absolute value, *M* is the number of bits allocated for the integer part plus the sign, and *F* is the number of bits allocated for the fractional part. The symbol ⌈ ⌉ denotes the ceiling function. Two operations are performed over coefficients: first, scaling is done with the help of the results of (8a) and appropriate number of bits is determined for integer and fractional part; second, coefficients are quantized, i.e. mapped to appropriate values in range given with (8b) using round to nearest method.

Decimal and hexadecimal representation of coefficients quantized with Q[16 14] are given in second and third row of Table 2, respectively. Observing Table 2, one can see that the coefficient with maximal absolute value is the d_1 coefficient of the first section, therefore $M = 2$, $F = 14$ are required for 16 bit representation. For these parameters range $RNG = [-2, 1.99993896484375]$ is covered in $INC = 0.00006103515625$ increments.

Assuming absence of any other source of computational error or noise we calculated the attenuation characteristic of the filter for different quantization formats of the coefficients as discussed above. The results for the example BP LSM filter are depicted in Fig. 4.

Observing Fig. 4, one can conclude that variants with 16-bit word length and higher, produce amplitude characteristics that start to agree with the one obtained with full precision. Therefore, 16-bits representation can be used if attenuation larger than 50 dBs is not required (observing the lower stop-band in Fig. 5). Of course, one can use Q[24 22] or Q[32 30] if more accurate design is required.

Table 2 Original and quantized filter coefficients

		Numerator		
	cell	c_0	c_1	c_2
Full precision	I	+0.0033086494281706663	-0.0018617614854014842	-0.0051704109135721505
	II	+0.0067062285319410561	+0.0085906290469084413	+0.0018844005149673854
	III	-0.044039453119340377	-0.0120564809558326	+0.031982972163507775
	IV	+0.06277875243767074	0	-0.062778752453249653
	V	-0.029334088252293972	+0.015384970101239978	+0.044719058353533951
	VI	-0.006864819539353288	-0.013389525398100878	-0.00652470585874759
	VII	+0.0074447307119548771	+0.0032630379244648162	-0.0041816927874900617
Q[16 14] decimal	I	+0.00329589843750	-0.00189208984375	-0.00518798828125
	II	+0.00671386718750	+0.00860595703125	+0.00189208984375
	III	-0.04406738281250	-0.01208496093750	+0.03198242187500
	IV	+0.06280517578125	+0.00000000000000	-0.06280517578125
	V	-0.02935791015625	+0.01538085937500	+0.04473876953125
	VI	-0.00683593750000	-0.01336669921875	-0.00653076171875
	VII	+0.00744628906250	+0.00323486328125	-0.00421142578125
Q[16 14] hexadecimal	I	0036	FFE1	FFAB
	II	006E	008D	001F
	III	FD2E	FF3A	020C
	IV	0405	0000	FBFB
	V	FE1F	00FC	02DD
	VI	FF90	FF25	FF95
	VII	007A	0035	FFBB

		Denominator	
	cell	d_1	d_2
Full precision	I	-1.886085860514888	+0.98894784642499967
	II	-1.8601293281281488	+0.96799935587139696
	III	-1.8323004517946606	+0.95057717438073264
	IV	-1.8083338880121447	+0.94157013740084117
	V	-1.7935719318070145	+0.94450186279953363
	VI	-1.7923157567661698	+0.96044412883386077
	VII	-1.8052459566148433	+0.98552821289667847
Q[16 14] decimal	I	-1.88610839843750	+0.98895263671875
	II	-1.86010742187500	+0.96801757812500
	III	-1.83227539062500	+0.95056152343750
	IV	-1.80834960937500	+0.94158935546875
	V	-1.79357910156250	+0.94451904296875
	VI	-1.79229736328125	+0.96044921875000
	VII	-1.80523681640625	+0.98553466796875
Q[16 14] hexadecimal	I	874A	3F4B
	II	88F4	3DF4
	III	8ABC	3CD6
	IV	8C44	3C43
	V	8D36	3C73
	VI	8D4B	3D78
	VII	8C77	3F13

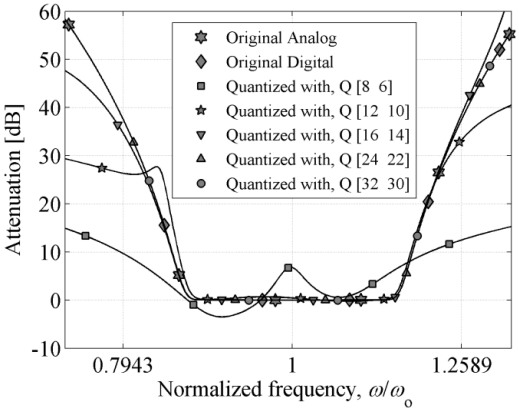
After 16-bits representation is adopted, we perform an additional verification, but now in the time domain. Fig. 5 depicts the results of time domain simulation using coefficients

quantized with Q[16 14]. Both signals and appropriate spectra are presented. The spectra shown in the Fig. 5b and 5d are obtained with N_{FFT} = 65536 point FFT.

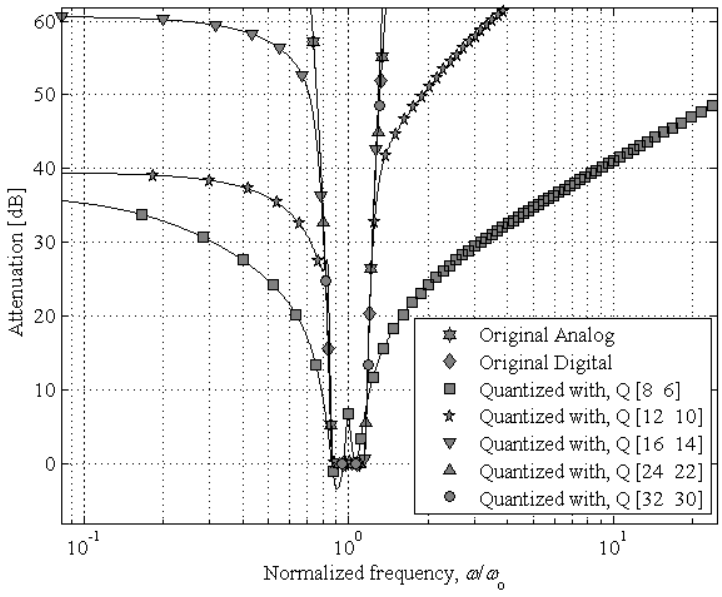
The input test signal is

$$s_{in} = \sin(2\pi f_0 t) + \sin(2\pi f_1 t) + \sin(2\pi f_2 t) + \sin(2\pi f_3 t)\,, \tag{9}$$

with: f_0=3 kHz, f_1=374.60 Hz, f_2=749.97 Hz, and f_3=5999.76 Hz. The bandwidth is limited by $[f_L, f_U]$ = [2583.56, 3483.56] Hz. The values of the test frequencies are picked to match integer multiples of FFT resolution bin (f_s/N_{FFT}) in order to minimize spectral leakage in the resulting FFT image of the spectrum.



a)



b)

Fig. 4 Attenuation of the 14th order LSM Band-Pass Filter: a) Pass-Band, b) Stop-Band

Observing the spectra in Fig. 5b and 5d one can see that after filtering there is only one dominant bin at frequency $f_{0,}$ while the others are filtered out.

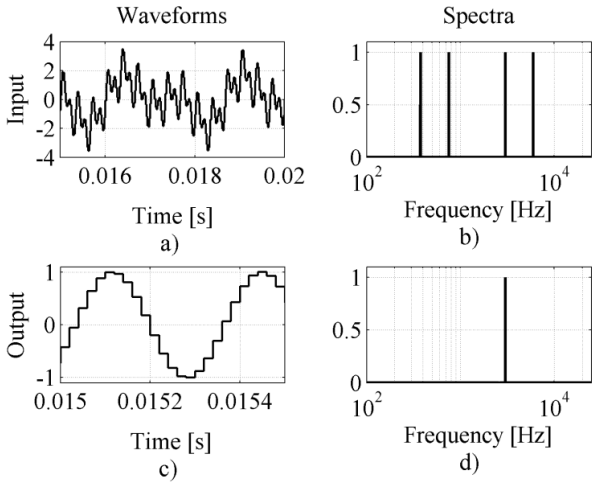


Fig. 5 Time domain simulation of the BP filter mathematical model:
a) input waveform, b) input spectrum, c) output waveform and d) output spectrum

For hardware implementation a versatile VHDL code was written. It combines second and/or first order cells presented in Fig. 1b. Illustrative schematics of the described second order TDFII and top-level filter cells are shown in Fig. 6a and 6b, respectively. Appropriate number/position of bits at each signal path is labeled as well.

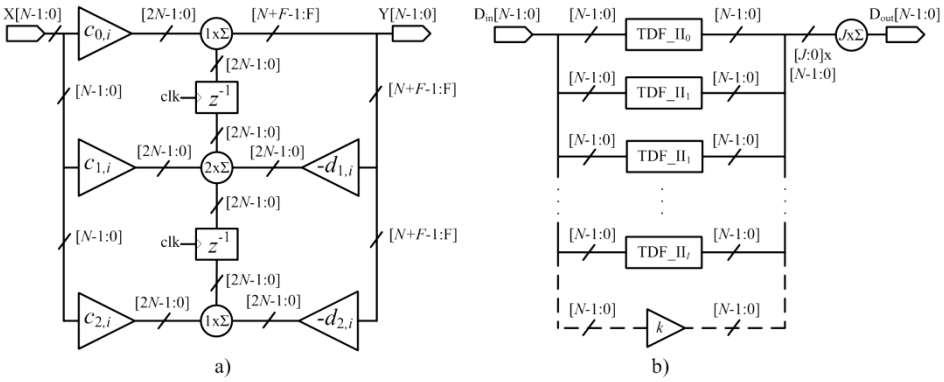


Fig. 6 Schematic representation of a) second order TDF II and b) top level filter cell

Each delay block (z^{-1}) is realized as a register. Parallel multipliers and ripple carry adders (with add and subtract functions) are designed for multiplication and summing operations, respectively. According to Fig. 6a it can be concluded that second order cell requires five multipliers and four adders. On the other hand, assuming zero values for $c_{2,i}$

and $d_{2,i}$ coefficients first order cell stems out from second order one. Therefore, first order cell will require three multipliers and only two adders.

To ensure successful synthesis whole filter is described structurally. Each individual block, starting from basic ones, i.e. multipliers, adders and registers up to top level entity is described. Therefore, no predesigned structures are assumed making the code as portable as possible.

TDF_II$_i$ represents first or second order TDF II cell. D_{in}, and D_{out} are input and output digital word. Index bounds and constants in Fig. 6b are defined as follows,

$$I = 0,\ldots,\mathrm{mod}(r,2) + \lfloor r/2 \rfloor + p - 1\ , \tag{10a}$$

$$J = \begin{cases} \max(I), \text{for } k = 0 \\ \max(I) + 1, \text{for } k \neq 0 \end{cases}, \tag{10b}$$

where I is the index of the filter's section and J is the number of adders used to sum outputs of the sections. The order of the resulting transfer function is marked with r. It should be emphasized that the order of the resulting Band-Pass/Stop transfer function is doubled compared to Low-Pass prototype function. Symbols $\lfloor\ \rfloor$, max(x) and mod(x,y) denotes floor function, maximal value, and modulo operator (reminder after x by y division), respectively. Parameter p is the flag that detects existence (1-exist, 0-do not exist) of two real poles/residues in resulting Band-Pass/Stop transfer function. If this is the case, two first order sections are generated. Finally, k represents direct term of partial fraction expansion of the resulting transfer function. This term is always zero, if the order of the denominator polynomial, m, is less than the order of the numerator polynomial, n. Otherwise, this term is of the order $m - n$. In filter's transfer functions it can only happen to be $m = n$ which gives k as a simple constant factor. Therefore, a branch with the k factor is nothing but a simple buffer stage. Possible values for parameters k and p are given in Table 3.

Table 3 Possible values for p and k parameters

Filter type	even order		odd order	
	p	k	p	k
High-Pass	0	1	0	1
Band-Pass	0	0	1	0
Band-Stop	0	1	1	1

Accordingly, VHDL entity accepts generics and has interface ports shown in Table 4. VHDL code sample is given below.

Next, VHDL description was verified by logic simulation with the excitation described in the previous section. It is important to mention that when dealing with hardware implementation two important effects must be examined, namely: saturation and round-off noise. Saturation is intensely dependent of the input signal waveform and filter's architecture and coefficient values. Even more, internal states usually saturate with different speeds making the tracking of the saturation process a non trivial task.

Table 4 Generics and ports of VHDL entity

	Symbol	Description
generics	*N*	word length
	F	number of bits after decimal point
	r	order of resulting transfer function
	p	flag for detecting two real poles/residues
	k	direct term of partial fraction expansion
	cfs	coefficients of the filter
ports	*clk*	clock signal at sample rate frequency
	rst	reset signal active at negative level
	x	input N bit signal
	y	output N bit signal

Round-off noise is the direct consequence of a fixed-point representation. Simply, product of two *N*-bit fixed-point numbers is a 2*N* bit number. This product must eventually be quantized to *N*-bits by rounding or truncation, which results with the round-off noise. A number of techniques can be used to mitigate this problem [21], [22].

The most commonly used technique to prevent saturation and round-off noise is the dynamic range scaling (or simply scaling) of the input signal prior to filtering action. Namely, each input signal value should be scaled down into a specific range which ideally, ensures no saturation in any of the internal and external nodes of the filter. Luckily, since two's complement representation is exploited, saturation of the internal nodes is to be allowed since it will be interpreted as an overflow (wrap-around) effect. E.g. an overflow occurs when the sum of two positive numbers yields a negative result and vice versa, otherwise the result is correct. Similar occurrence happens when the internal state values reach boundaries of the dynamic range, i.e. first larger/smaller value, then the maximal/minimal is interpreted as the minimal/maximal value of the range. This can be tolerated as long as the final values of the output signal are valid, i.e. wrap-around does not coincide with the moment of the output signal acquisition. This is where parallel realization, adopted in this work, again outperforms the cascade one. Saturation conditions are drastically relaxed when using parallel realization, especially as the order of filter increases. Occurrence of wrap-around is reduced as well. This is simply because no matter how large the filter order is, all second and/or first order cells process the input signal independently of each other. Therefore, scaling of the input signal applies to all cells at once. The bottleneck is, of course, the output summing node. Nevertheless, net sensitivity to saturation is reduced when constrains regarding saturation are relaxed at each individual cell. One may also choose to omit scaling and relay completely on two's complement representation, but this technique requires sound knowledge about the input signal waveform and algorithm for tracking and handling wrap-around effect. Also, all possible cases have to be predicted, therefore extensive simulations are required. This is usually too expensive in the real-world applications, therefore some form of scaling is always applied. Moreover, scaling technique is quite easy to implement in the digital domain, knowing that each scaling down/up by two is nothing but the one simple shift right/left operation. Accordingly, scaling operation can be implemented as tunable (programmable). Even non-linear scaling can be implemented if high accuracy is required. Unfortunately, there is no scaling technique which provides closed-form, general solution and it all depends on concrete application at the end of the day. This implies that some exploration of the time domain simulation results is inevitable in the design process to

determine the appropriate scaling factor. Finally, combining several techniques to cope saturation and round-off noise may result with more efficient solution, but scaling with fixed coefficient, because of its simplicity, is still considered suitable for verity of applications and therefore utilized in this design, as well.

Since our test signal is known in advance, fixed scaling coefficient is to be determined. Finding maximum and minimum of the function given in (3), one can determinate the range of the input signal, i.e. $RNG_{\text{in}} = [-3.73, 3.73]$. Using time domain simulation scaling factor of four turned out to be suitable. This can be also intuitively concluded when looking at the range of filter's coefficients. Namely, dividing RNG_{in} with four gives $RNG_{\text{new}} = [-0.93, 0.93]$, which is smaller than half of the filter's coefficient range $RNG = [-2, 1.99993896484375]$, leaving enough headroom for values of internal states to spread without reaching saturation. After filtering, output is scaled up and the obtained results are presented in Fig. 7.

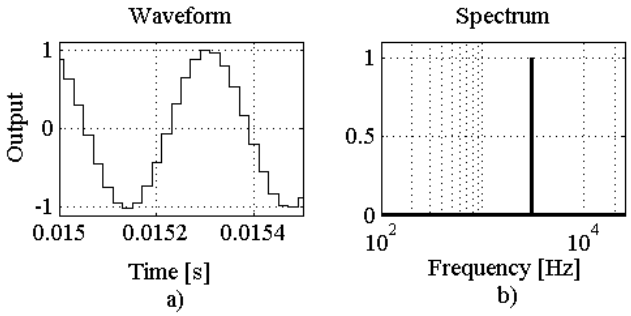


Fig. 7 Time domain simulation of the BP filter VHDL model:
a) output waveform, b) output spectrum

Sound representation of signal's spectrum using FFT usually requires a large number of samples. This inevitably leads to longer time domain simulation. To minimize duration of the time domain logic simulation, a smaller number of FFT points, compared with a case with purely mathematical model which simulates faster (Fig. 5b, 5d), is desired. Therefore, $N_{\text{FFT}} = 16384$ is chosen for representing the output signal spectrum in Fig 7b. It turns out that this number gives a satisfactory compromise between simulation time and FFT accuracy. Finally, comparing Fig. 7a with Fig. 5c and Fig. 7b with Fig. 5d, one can see that the time and frequency domains of the output signal obtained by simulating mathematical and hardware models of the filter match. This proves that hardware representation successfully implements the desired behavior of the designed filter.

6. CONCLUSION

With this case study we intended to fulfill two main goals. First, we wanted to raise the awareness of the salient advantages of the CMAC filtering functions as compared with their non-monotonic counterparts. To achieve this, we gave a short overview of the properties of CMAC amplitude characteristics. The second goal was to give, for the first time, design results characterizing the amplitude characteristic of band-pass IIR digital filters. Accordingly, we went through several steps. First, we gave arguments on the choice of IIR filters. Then, we gave arguments for the parallel implementation of digital

filters that was used throughout the design process. Next, we described and exemplified the complete design procedure including the verification steps needed to support the design decisions taken on the way. All that was performed on the example of a band-pass CMAC IIR digital filter, a solution that was here reported for the very first time.

Acknowledgement: *This research was funded by The Ministry of Education, Science and Technological Development of Republic of Serbia under contract No. TR32004.*

REFERENCES

[1] D. Topisirović, V. Litovski, and M. Andrejević Stošović, "Unified theory and state-variable implementation of critical-monotonic all-pole filters," *International Journal of Circuit Theory and Applications*, vol. 43, no. 4, pp. 502–515, 2015.
[2] JF. Kaiser, *Digital filters*, in: FF. Kuo, JF. Kaiser (Eds.) *System Analysis by Digital Computer*. Wiley: New York, 1996, Chapter 7, p. 245.
[3] D. Mirković, M. Andrejević Stošović, P. Petković and V. Litovski, "IIR digital filters with critical monotonic pass-band amplitude characteristic," *AEU - International Journal of Electronics and Communications*, vol. 69, no. 10, pp. 1495-1505, Oct. 2015, ISSN 1434-8411.
[4] DS. Humpherys, The Analysis, Design, and Synthesis of Electrical Filters, Prentice-Hall, 1970
[5] K. Geher, *Theory of Network Tolerances*, Akademiai Kiadó: Budapest, Hungary, 1971.
[6] V. Litovski, "Synthesis of monotonic passband sharp cutoff filters with constant group delay response," *Circuits and Systems, IEEE Transactions on*, vol. 26, no. 8, pp. 597–602, Aug 1979.
[7] Rabey J. *Low power design essentials*. Springer Science + Business Media, LLC: New York, 2009.
[8] MF. Quélhas, A. Petraglia, MR. Petraglia, "Efficient group delay equalization of discrete-time IIR filters," In Proceedings of the XII European Signal Processing Conference, EUSIPCO-2004, vol. 1, Vienna, Austria, 2004, pp. 125-128.
[9] Rabrenović D, Jovanović V. *Low-pass filters with critical monotonic magnitude*. Publications of Faculty of Electrical Engineering, ETA series: Belgrade, 1973, pp. 59–68.
[10] B. D. Rakovich, "Designing monotonic low-pass filters – comparison of some methods and criteria," *International Journal of Circuit Theory and Applications*, vol. 2, no. 3, pp. 215–221, 1974.
[11] S. Butterworth, "On the Theory of Filter Amplifiers," Experimental Wireless and the Wireless Engineer, vol. 7, 1930, pp. 536-541.
[12] A. Papoulis, "Optimum filters with monotonic response," In Proceedings of the IRE, vol. 46, no. 3, pp. 606–609, 1958.
[13] A. Papoulis, "On monotonic response filters," *Proceedings of the IRE*, vol. 47, pp. 332–333, 1959.
[14] M. Fukada, "Optimum filters of even orders with monotonic response," *Circuit Theory, IRE Transactions on*, vol. 6, no. 3, pp. 277–281, 1959.
[15] P. Halpern, "Optimum monotonic low-pass filters," *Circuit Theory, IEEE Transactions on*, vol. 16, no. 2, pp. 240–242, May 1969.
[16] B. Rakovich and V. Litovski, "Least-squares monotonic lowpass filters with sharp cutoff," *Electronics Letters*, vol. 9, no. 4, pp. 75–76, February 1973.
[17] D. Mirković, P. Petković, and V. Litovski, "A second order s-to-z transform and its implementation to IIR filter design," *COMPEL - The international journal for computation and mathematics in electrical and electronic engineering*, vol. 33, no. 5, pp. 1831–1843, 2014.
[18] W. Park, K.-S. Park, and H.-M. Koh, "Active control of large structures using a bilinear pole-shifting transform with H_∞ control method," *Engineering Structures*, vol. 30, no. 11, pp. 3336–3344, 2008.
[19] H. Orchard and G. C. Temes, "Filter design using transformed variables," *Circuit Theory, IEEE Transactions on*, vol. 15, no. 4, pp. 385–408, 1968.
[20] P. Geffe, "Designers guide to active bandpass filters," *Part III', EDN*, vol. 19, no. 7, 1974.
[21] K. K. Parhi, Scaling and Round-off Noise in: *VLSI digital signal processing systems: design and implementation*. John Wiley & Sons, 2007, Chapter 11.
[22] K. Prasad and P. Sathyanarayana, "Signal scaling in cascade digital filters," *Circuits, Systems and Signal Processing*, vol. 8, no. 4, pp. 421–426, 1989.