On Numerical Robustness of Bi-quad Structures using Fixed-Point Approximate Multiplication

Koch, Peter; Østergaard, Jan; Andersen, Ove Kjeld

Published in:
Proc. 2022 25th International Symposium on Wireless Personal Multimedia Communications (WPMC)

DOI (link to publication from Publisher):
10.1109/WPMC55625.2022.10014781

Publication date:
2022

Document Version
Accepted author manuscript, peer reviewed version

On Numerical Robustness of Bi-quad Structures using Fixed-Point Approximate Multiplication

Peter Koch, Jan Østergaard, and Ove Andersen
Department of Electronic Systems
Aalborg University
Aalborg, Denmark
(pk, jo, oa)@es.aau.dk

***Abstract*—Digital filters are key components in many applications related to wireless personal communication and multimedia devices, some even portable and battery powered. The ability to design and implement cost efficiently such filters is therefore of significant importance. Recursive filters are known to have low computational complexity (number of multiplication and addition) but at the same time they are numerically sensitive due to their feedback loop. Therefore, using arithmetic functional units with reduced accuracy might introduce some challenges, despite their proficiency in physical size, execution time, and power consumption. We are therefore interested in investigating to what extent approximate multiplication can be used in different types of bi-quad sections, which is the fundamental building block in higher order IIR filter systems. We found that it is possible to operate three selected types of such implementation structures in the presence of additive noise from multipliers with different degree of approximation, and we show that there are significant performance differences of the structures, both in the time- and in the frequency domain.**

***Index Terms*—Approximate multiplication, Finite word length, Recursive 2^{nd} order filters, Additive noise, Simulation.**

I. Introduction

For several decades, digital signal processing has been an enabling technology used intensively in many applications related to wireless multimedia communication. For instance, the deployment of real-time signal processing hardware and software has paved the way for Software Defined Radio architectures to be an integrated part of mobile communication, [1]. In particular, digital filtering, sample rate conversion, and signal analysis are examples of important functions which are used in the front-end as well as in the base-band sections of wireless communication devices, [2] [3]. Since many such devices are often defined to comply with non-trivial design constraints on area (A), execution time (T), and power consumption (PW), researchers have developed many different design methodologies to meet given specifications while still maintaining the requirements on the overall system performance. The performance depends directly on the accuracy of the arithmetic computations conducted on the target architecture, i.e., numerical operations executed on a DSP, an MCU, an FPGA or a mix hereof. Since the design metrics are normally mutually dependent, an improvement in one metric will most likely lead to a degradation in one or more of the others, [4].

However, in some specific application domains, primarily multimedia and other types of systems with human interaction based on e.g., sound/hearing and image/vision, it is possible though to relax somewhat the computational accuracy with only a negligible impact on the overall perceived performance. One possible way to do this is to employ arithmetic circuits which intentionally perform the computations with reduced accuracy, thus minimizing the overall circuit complexity [5]. Numerous such circuits have been suggested, mainly for addition and multiplication, but also circuits for division have been reported, [6]. Arithmetic circuits with this behaviour belong to the category of Approximate Computing (AC).

AC circuits have been applied to signal processing functions such as Finite Impulse Response (FIR) filters, Fast Fourier Transform (FFT), and Discrete Cosine Transform (DCT), e.g., [7], [8], [9], demonstrating the ability to obtain A-, T-, and/or PW-reduction at the expense of a decreased accuracy. These functions are all characterized by a feed-forward data flow. Since AC arithmetic performs inexact operations, an error is therefore introduced at their particular locations in the algorithm/architecture, and thus these noise sources impact directly the Signal to Noise Ratio (SNR) at the output.

For recursive algorithms, e.g., frequency selective Infinite Impulse Response (IIR) filters, the application of AC arithmetic is challenging. First and foremost, filters with signal feedback are known to be more sensitive to any noise induced [10], thus potentially leading to an unacceptable reduction in the output SNR. Secondly, IIR filters eventually can become unstable due to unintended misalignment of the pole locations. On the other hand, an IIR filter normally has a significantly lower filter order, i.e., a lower computational complexity, as compared to an FIR filter with a similar specification. This makes IIR filters a strong candidate in many applications with tight $A-$, $T-$ and/or $PW-$budgets.

In order to further improve the overall cost function for digital filters implemented in dedicated hardware, an idea therefore is to introduce AC into IIR filter structures. However, only very few works have previously been reported on AC (or Approximate Processing) as applied to IIR filters. In [11], the authors discuss an approximate filtering approach where the filter order is dynamically adjusted in order to enable time-varying stop band attenuation in proportion to the time-varying input SNR, while maintaining a fixed output SNR. The

978-1-6654-7318-7/22/$31.00

purpose being to disable one or more of the 2^{nd} order sections of the overall filter when not needed. Another and much more relevant work, [12], presents the design of an A-weighting filter implemented as a 6^{th} order IIR filter organized into a cascade of 1^{st} order sections. In this configuration, a subset of the multiplications is conducted using approximate multipliers. The authors show that approximate multiplication impacts the ordering of the cascaded filter sections, and similarly they show that the overall amplitude response is affected by the amount of approximation used in the multipliers. Although being a valuable contribution, this design exercise solely focus on 1^{st} order Direct Form I structures which represent only one among several potential filter structures.

Various studies have shown that different IIR filter structures behave differently in a finite word length context, but to our knowledge no investigations have been published so far illustrating how recursive filter structures generally perform when operated in an AC environment. In our work we therefore conduct a series of simulation experiments where AC-based multiplication is applied to different types of bi-quad structures. For varying i) filter topology and ii) pole location, we investigate how the degree of multiplier inaccuracy in AC-based multiplications impact the overall numerical performance of such filters.

The paper is organized with an introduction to approximate multiplication in Sec. II, a description of the bi-quad structures in Sec. III, a presentation and evaluation of the experimental results in Sec. IV, and finally the conclusion in Sec. V.

II. Approximate Radix-4 Multiplication

For signal processing applications it is almost always necessary to conduct arithmetic operations using signed numbers, and therefore the 2's complement number representation is often the preferred choice. Circuits for signed approximate multiplication are described and evaluated in [6], several of which are based on the Radix-4 Booth algorithm, [13].

Given two d-bit numbers X and Y being the multiplicand and the multiplier, respectively, the product is $P = X \cdot Y$. In order to express Y as a 2's complement number, we use a notation where the MSB is indexed 0. This is opposite to most literature, where the MSB is indexed $d-1$, but in a digital filter context where we scale the input signal (X), and the coefficients (Y) to the dynamic range $[-1; 1[$, this is a convenient notation since Y can then be written as

$$Y = -y_0 + \sum_{j=1}^{d-1} y_j \cdot 2^{-j} \tag{1}$$

where the fixed point is located just after the sign bit y_0, and where the product can then be expressed as

$$P = -y_0 \cdot X + \sum_{j=1}^{d-1} (y_j \cdot X) \cdot 2^{-j} \tag{2}$$

It can be shown that the multiplier Y can be rewritten as

$$Y = \sum_{j=1,odd}^{d-1} (y_j + y_{j+1} - 2 \cdot y_{j-1}) \cdot 2^{-j} \tag{3}$$

which enables the product to be calculated alternatively as

$$P = \sum_{j=1,odd}^{d-1} (z_j \cdot X) \cdot 2^{-j} \tag{4}$$

where

$$z_j = y_j + y_{j+1} - 2 \cdot y_{j-1}; z_j \in \{0, \pm 1, \pm 2\} \tag{5}$$

From Equ. (4) and (5) it is concluded that P is the sum of $d/2$ left-shifted and sign-extended partial products (PP) which can take on the values $\{0, \pm X, \pm 2X\}$ depending on the pattern of three consecutive bits of the multiplier Y, starting with y_0 at the MSB end.

In the general case where d can be even or odd, Equ. (3) can now conveniently be expressed as

$$Y = \sum_{j=0}^{\lceil \frac{d}{2} \rceil - 1} (y_{2j+1} + y_{2j+2} - 2 \cdot y_{2j}) \cdot 2^{-(2j+1)} \tag{6}$$

from which we finally derive the expression for the product

$$P = \frac{1}{2} \cdot \sum_{j=0}^{\lceil \frac{d}{2} \rceil - 1} X \cdot z_j \cdot 4^{-j} \tag{7}$$

Equ. (7) shows that the PPs are individually shifted two bit positions against each other (i.e., Radix 4), and that P is obtained after a 1-bit right shift of the sum of the $\lceil \frac{d}{2} \rceil$ PPs.

In our experiments we have opted for an AC multiplier denoted "Broken Booth Multiplier" (BBM) which is based on Equ. (7). The argument for this choice is that the BBM represents a sound compromise between execution time and power consumption, against the metric *Mean Relative Error Distance* which is often used for evaluation of AC circuits, [14].

The BBM can be implemented using two different modes, denoted as Type_0 and Type_1, respectively. We use Type_0 where all the PPs are completely calculated prior to sign-extension and addition using a 2's complement number representation, no matter the actual sign of the individual PPs. In the Type_1 scenario, some of the negative PPs are represented as 1's complement numbers, thus potentially reducing the need for one or more LSB additions. This saving is possible since $X_{2's\,comp} = X_{1's\,comp} + LSB = \overline{X} + LSB$ which indicates that the Type_1 scheme may introduce a larger error due to the omitted LSB addition. Fig. 1 shows the two BBM types.

Note from Fig. 1 that i) the PPs, according to equation 4, are consecutively left shifted two bit positions due to their individual numerical weighting, and ii) the PP word length equals $d+1$ bit which stems from the potential multiplication with ± 2. The figure also illustrates a dotted vertical line known as the *Vertical Breaking Level* (VBL), which represents the fundamental concept in this approximate multiplier. The idea is that all bits to the right of the VBL are nullified (grey dots on Fig. 1), thus eliminating the need for additions in these bit positions, and thereby reducing the circuit complexity at the expense of inexact products.

In [14], the VBL is defined in the interval $[0; d-1]$, 0 representing the LSB position of the least significant (i.e., the

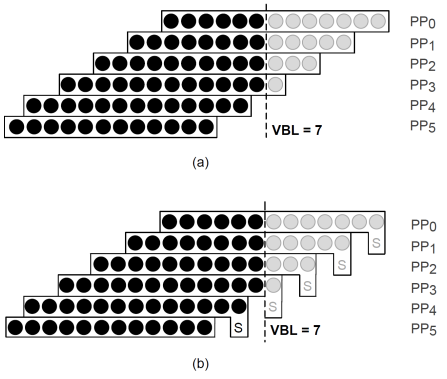

Fig. 1. The BBM shown for a 12x12 bit multiplication, Type_0 (a) and Type_1 (b). In Type_0, the addition of 1 LSB is included into the negative valued PPs, whereas in Type_1 this is true only for the negative valued PPs which doesn't have their LSB nullified, i.e., the LSBs to the left of the VBL, [14].

topmost) PP. In our implementation of the Type_0 BBM, we extend VBL to be defined in the interval $[0; 2d-1]$, thus enabling all bits in the product to be nullified.

We have built a simulation model of the Type_0 BBM which is a parameterised $d \times d$ bit multiplier, where the accuracy can be adjusted, i.e., d and VBL are presented as input parameters along with the multiplier and the multiplicand. For $\mathrm{VBL} = 0$, an exact product is calculated, given the word length d. The two input operands are fed into the multiplier as floating point numbers, both numerically less than 1, and are next converted into d-bit 2's complement numbers, d restricted to be even.

Using Equ. (7), a total of $\frac{d}{2}$ 2's complement PPs are next derived. Starting with the least significant PP, zeros are then inserted from LSB towards MSB according to the given VBL value, and the PPs are next converted back to floating point number representation, scaled due to their individual weight factor, and added. Finally, the sum is down-scaled with a factor of 2. The final product P is therefore a floating point number, numerically less than 1, and with an accuracy equivalent to a $2d$-bit 2's complement representation.

In order to get an insight into the numerical behaviour of the BBM, we generated 10^4 products based on pseudo-random input operands drawn from a uniform discrete distribution, all in the interval $[-1; 1[$. We measure the *Error Distance* (ED) defined as the difference between the approximate and the exact products. Overall, we found that the Type_0 BBM produces errors which are always negative and which tend to be normal distributed with mean and standard deviation being dependent on d and VBL. In order to consider how we should model this error, we investigated the correlation ρ between the exact product P and the ED;

$$\rho = E[P \cdot ED] \simeq \frac{\sum_{j=1}^{N} P_j \cdot ED_j}{\sqrt{\sum_{j=1}^{N}(P_j)^2 \cdot \sum_{j=1}^{N}(ED_j)^2}} \qquad (8)$$

Fig. 2 shows an example of the ED distribution for the situation $d = 16$, $\mathrm{VBL} = 14$, and $N = 10^4$. For these values we typically found $|\rho| < 0.01$ indicating a limited correlation between the product and the error. Thus, we safely conclude that the error can be considered as an additive noise source.

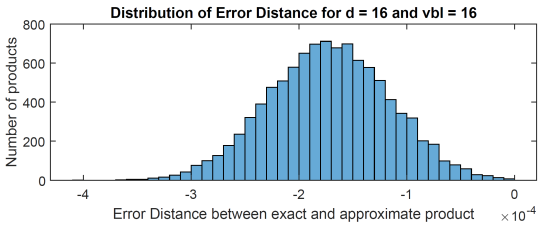

Fig. 2. The ED distribution for the Type_0 BBM excited by 10^4 random input operand pairs all numerically less than 1. The partial products which are impacted by the VBL-operation are all decreased numerically due to the nature of 2's complement, and therefore the error is always negative.

III. Fixed-point Bi-quad Filter Sections

The general transfer function for the bi-quad sections investigated in this work is given as

$$H(z) = \frac{\sum_{i=0}^{2} b_i z^{-i}}{1 - \sum_{j=1}^{2} a_j z^{-j}} \qquad (9)$$

which we express in the time-domain by the constant coefficient difference equation

$$y[n] = \sum_{j=1}^{2} a_j \cdot y[n-j] + \sum_{i=0}^{2} b_i \cdot x[n-i] \qquad (10)$$

We implement Equ. (10) using three different bi-quad structures; the Direct Form I (DF-I), the Direct Form II (DF-II), and the Direct Canonical Form (DCF) as shown in Fig. 3. Using floating point arithmetic, these structures are characterized by numerically equivalent I/O-relations. In a d-bit fixed-point environment however, this behaviour is it not guaranteed due to the introduction of quantization errors which originate from either i) the individual products or ii) the accumulated products being quantized from $2d$ to d bits, [15].

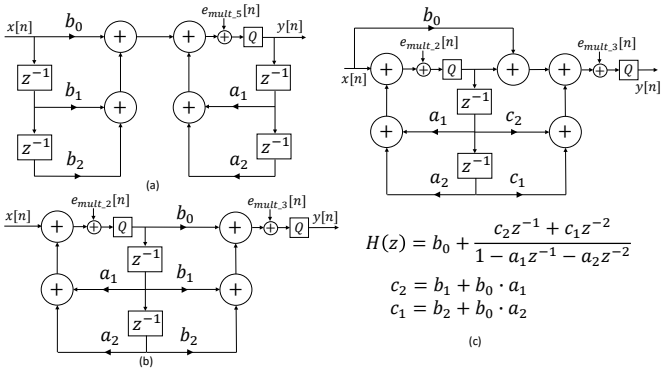

Fig. 3. Three different types of bi-quad sections; (a) Direct Form I, (b) Direct From II, and (c) Direct Canonic Form. The Direct Form I and II are implementations of Equ. (9) whereas the Direct Canonic Form is based on a rewritten version of the transfer function, here shown under the structure.

In our simulation model we perform all product accumulations in double precision, which means that quantization

is introduced only at the locations indicated with a Q in Fig. 3. The quantization Q is implemented using rounding. The $2d$-bit additions are performed by adder modules, which i) accept floating point operands on the input, ii) convert these operands into 2's complement representations, iii) perform bit-parallel addition using a Ripple Carry Adder (RCA) including overflow detection, and finally iv) convert the resulting sum back to floating point representation. This strategy implements an efficient interface between the multipliers and the adders, essentially enabling the multiplications and the additions to be conducted in any word length, respectively.

We use the BBM for all multiplications in our simulations. Due to its approximate behaviour, the generated products are negatively biased which we model as an additive normal distributed noise sequence $e_{mult}[n]$ injected after each multiplication, see Fig. 4. The additive nature of the multiplier error combined with the linearity of Equ. (10) enables us to reduce the multiplier noise sources into one single source denoted $e_{mult\_k}[n]$ in Fig. 3, where k indicates the number of multiplications which add up to the resulting noise source.

Thus, in our recursive filter structures, there are two different noise sources in the feedback path, Q and $e_{mult\_k}[n]$, which are functions of d, as well as d, VBL, and k, respectively. Normally, Q can also be modelled as an additive sequence which, under certain conditions, is assumed being a white signal equally distributed in $[-\Delta/2; \Delta/2[$ (for rounding) and with variance $\sigma_q^2 = \frac{\Delta^2}{12}$, where $\Delta = 1/2^{(d-1)}$ represents 1 LSB. Despite this assumption, we do not conclude that the two noise sources are mutually independent, and we do also not investigate further this question, thus keeping them separated. We next address how the filter coefficients potentially influence $e_{mult\_k}[n]$, i.e., the overall numerical robustness of the filter structures.

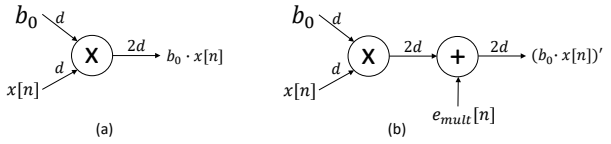

Fig. 4. An exact $d \times d$ bit multiplier (a) generates a $2d$-bit product. Similarly does the BBM (b), but the product has superimposed noise for VBL $\neq$ 0. Referring to Fig. 2, for $d = 16$ and VBL $= 16$, we found $e_{mult}[n]$ to be normal distributed with mean $\mu \approx -1.7e^{-4}$ and variance $\sigma^2 \approx 3.1e^{-9}$ for uniform distributed inputs. For VBL $= 8$, $\mu \approx -3.2e^{-7}$ and $\sigma^2 \approx 2.3e^{-14}$. In comparison we note that for $d = 16$, $\sigma_q^2 = 7.8e^{-11}$, and mean $\mu_q = 0$.

IV. Experimental Results and Discussion

In order to evaluate the numerical properties of the different bi-quad sections, we conduct a series of experiments where we investigate the time domain as well as the frequency domain behaviour. We excite the filters with the impulse signal $s \cdot \delta[n]$, where $n \in [0; N-1]$, and s is a scalar used to adjust the signal level in order to avoid internal overflow in the filters. Since we are not aiming for the design of filters with the highest possible output SNR, but rather want to derive relative performance indicators, we do not perform scaling in a strict mathematical sense. Alternatively, we empirically select $s = 0.5$ which leads to a safe compromise between utilization of the 2's complement dynamic range $[-1; 1[$ and elimination of overflow in the variables where overflow is not allowed.

We measure the impulse responses from two versions of each structure, one exact response $h_{ex}[n]$ from a floating-point implementation, and another approximate response $h_{ap}[n]$ obtained using a d-bit BBM and a $2d$-bit RCA. From these responses we calculate the residual energy defined as

$$\sigma_h^2 = \sum_{n=0}^{N-1} (h_{ex}[n] - h_{ap}[n])^2 \tag{11}$$

where N is chosen sufficiently large in order for the impulse response to reach its steady state. We found that $N = 1000$ is a viable value for our experiments. This metric gives an insight into how much an inexact filter deviates from the exact one in the time domain. Excitation signals other than $\delta[n]$ could have been chosen, but since we are also interested in the frequency domain behavior, we opted for this input.

According to the Parseval relation, a residual energy, identical to σ_h^2, will emerge if the two associated spectra, $|H_{ex}(e^{j\omega})|$ and $|H_{ap}(e^{j\omega})|$, are compared directly. Since our aim is to evaluate how the different filter structures perform against each other in the frequency domain, we alternatively apply the RMS Logarithmic Spectral Distance, [16], defined as

$$\hat{S} = \sqrt{\frac{2}{N} \sum_{n=0}^{\frac{N}{2}-1} [\ln(\frac{|H_{ex}(e^{j\frac{2\pi}{N}n})|}{|H_{ap}(e^{j\frac{2\pi}{N}n})|})]^2} \tag{12}$$

where we evaluate the frequency range from DC to $f_{sample}/2$. No practical filter is exactly band limited and thus $|H| > 0$ holds for both amplitude responses involved.

Since in all three bi-quad structures the BBM noise is introduced into the critical feedback path, we limit our experiments to focus on varying the pole locations, i.e., for all experiments we fix the b_i coefficients and alter the a_j coefficients only, Equ. (9). We opt for a double zero in $z = -1$, i.e., $b_0 = b_2 = 0.5$ and $b_1 = 1$, ($1 - \Delta$ in 2's complement). All experiments are therefore conducted on filters with a low-pass characteristic. The a_j coefficients are derived from pole locations specified using polar coordinates, (r, Θ). We choose only pole locations for which $r < 1$. In many cases however, it turns out that the coefficient $1 \leq |a_1| < 2$, which cannot be represented by the Q1.d-1 format that we use for the coefficients. We therefore split the coefficient into two coefficients each equal to $a_1/2$ and then perform two multiplications and one more addition. This eventually increases the number of BBM noise sources by 1 (not shown in Fig. 3).

Our simulations are all conducted with word length $d = 16$, and with VBL $= \{0, 8, 16\}$. The results in terms of σ_h^2 and $\hat{S}$ for each of the three bi-quad structures are shown for selected r and Θ in Table I, II, and III. Note that Table I represents an exact 16-bit reference.

Initially, we note that in almost all cases, σ_h^2 and $\hat{S}$ deteriorate when $r \to 1$ no matter the VBL value. This general trend

TABLE I
RESIDUAL ENERGY AND SPECTRAL DISTANCE FOR DIRECT FORM I, DIRECT FORM II AND DIRECT CANONICAL FORM. $d = 16$, VBL $= 0$.

r, Θ	$\pi/6$	$\pi/3$	$\pi/2$	$2\pi/3$	$5\pi/6$
0.8	9.384e-7	5.995e-7	8.951e-7	6.026e-7	9.198e-7
	0.2085	0.1670	0.1216	0.0968	0.2985
0.95	1.327e-5	1.284e-5	2.004e-5	1.314e-5	1.884e-6
	0.3160	0.2463	0.3427	0.2526	0.1466
0.99	1.381e-4	1.012e-4	3.063e-4	1.423e-4	1.204e-4
	0.4346	0.4610	0.5234	0.5270	0.2214

r, Θ	$\pi/6$	$\pi/3$	$\pi/2$	$2\pi/3$	$5\pi/6$
0.8	3.679e-6	6.035e-7	4.520e-7	7.193e-9	4.617e-9
	0.2646	0.2410	0.2242	0.1632	0.0345
0.95	8.531e-5	2.530e-5	9.697e-6	2.070e-6	4.019e-7
	0.4277	0.3623	0.2554	0.2464	0.0554
0.99	2.251e-4	2.533e-4	8.727e-5	2.645e-5	1.150e-6
	0.4788	0.5301	0.4311	0.4560	0.1669

r, Θ	$\pi/6$	$\pi/3$	$\pi/2$	$2\pi/3$	$5\pi/6$
0.8	8.288e-6	2.677e-6	8.951e-7	3.028e-7	9.169e-7
	0.0018	0.0008	0.0012	0.0018	0.0003
0.95	1.926e-4	6.025e-5	1.939e-5	8.567e-6	1.889e-5
	0.0042	0.0022	0.0015	0.0021	0.0011
0.99	12.00e0	5.504e-4	1.747e-4	8.542e-5	5.086e-5
	2.7245	0.0083	0.0029	0.0026	0.0019

TABLE II
RESIDUAL ENERGY AND SPECTRAL DISTANCE FOR DIRECT FORM I, DIRECT FORM II AND DIRECT CANONICAL FORM. $d = 16$, VBL $= 8$.

r, Θ	$\pi/6$	$\pi/3$	$\pi/2$	$2\pi/3$	$5\pi/6$
0.8	9.354e-7	5.995e-7	8.951e-7	6.019e-7	9.198e-7
	0.2068	0.1670	0.1216	0.1316	0.2985
0.95	3.260e-6	1.283e-5	2.004e-5	1.314e-5	1.795e-6
	0.1909	0.2666	0.3447	0.2647	0.1919
0.99	1.329e-5	9.202e-5	2.802e-4	1.241e-4	6.331e-5
	0.1502	0.4537	0.5267	0.5195	0.2006

r, Θ	$\pi/6$	$\pi/3$	$\pi/2$	$2\pi/3$	$5\pi/6$
0.8	3.710e-6	1.490e-6	4.520e-7	3.044e-7	6.557e-9
	0.2263	0.3039	0.2242	0.1910	0.1037
0.95	1.297e-5	2.923e-5	9.692e-6	3.375e-6	1.800e-7
	0.3346	0.2776	0.2556	0.1882	0.1054
0.99	2.900e-5	2.385e-4	7.934e-5	2.419e-5	4.110e-7
	0.3186	0.4846	0.4382	0.4797	0.2438

r, Θ	$\pi/6$	$\pi/3$	$\pi/2$	$2\pi/3$	$5\pi/6$
0.8	8.299e-6	2.677e-6	8.963e-7	3.026e-7	9.088e-7
	0.0018	0.0008	0.0012	0.0018	0.0003
0.95	2.908e-5	6.028e-5	1.937e-5	8.557e-6	1.712e-6
	0.0034	0.0020	0.0015	0.0021	0.0008
0.99	12.00e0	5.042e-4	1.588e-4	7.278e-5	1.222e-5
	2.7247	0.0080	0.0027	0.0023	0.0014

is expected due to a decreased SNR when the pole locations approach the unit circle [10], in particular for the DCF with pole locations at low frequencies.

Next, we make the following general observations. Compared against the reference, i.e., VBL $= 0$, for VBL $= 8$ and $\Theta = \pi/6$ none of the structures, for any pole radius, show impaired performance in neither time nor frequency. Rather we see improvements in many cases. This is interesting since a degradation would normally be expected when noise is introduced. One reason for this result may relate to the fact that (for all structures) poles located at a low frequency lead to a slowly varying impulse response which enters into steady-state after a certain number of samples, depending on the value of r. In steady-state, the impulse response will be characterized by zero-input limit cycle oscillations for VBL $= 0$ which however, do not occur to the same extent for VBL $= 8$. Due to the BBM-induced noise, a signal is circulating in the recursive part of the structures thus preventing the output signal from entering the limit cycle dead band, and therefore there are no or reduced output signal oscillations. In such a case, we see a better agreement between the exact and the approximate responses, and thus a smaller residual energy.

TABLE III
RESIDUAL ENERGY AND SPECTRAL DISTANCE FOR DIRECT FORM I, DIRECT FORM II AND DIRECT CANONICAL FORM. $d = 16$, VBL $= 16$.

r, Θ	$\pi/6$	$\pi/3$	$\pi/2$	$2\pi/3$	$5\pi/6$
0.8	2.841e-3	4.943e-4	6.137e-7	6.055e-5	7.711e-5
	0.6300	0.3775	0.3862	0.3829	0.3785
0.95	5.078e-3	6.763e-4	2.619e-4	2.559e-4	1.900e-3
	0.6988	0.6051	0.4544	0.2189	0.5068
0.99	1.585e-2	3.240e-3	2.434e-3	2.641e-3	4.181e-3
	0.5689	0.7540	0.6815	0.6503	0.4760

r, Θ	$\pi/6$	$\pi/3$	$\pi/2$	$2\pi/3$	$5\pi/6$
0.8	1.379e-2	2.229e-3	7.417e-7	3.037e-4	3.438e-4
	0.7896	0.6009	0.3899	0.2918	0.1073
0.95	1.824e-2	2.580e-3	1.468e-4	2.700e-4	1.383e-4
	0.8222	0.5752	0.3856	0.3092	0.3146
0.99	5.982e-2	8.666e-3	8.833e-4	8.653e-4	2.237e-4
	0.8726	0.6615	0.5812	0.5684	0.2889

r, Θ	$\pi/6$	$\pi/3$	$\pi/2$	$2\pi/3$	$5\pi/6$
0.8	3.481e-2	5.229e-3	2.212e-6	1.082e-3	6.469e-4
	0.0323	0.0365	0.0032	0.0524	0.0495
0.95	4.295e-2	6.408e-3	2.471e-4	7.519e-4	9.872e-4
	0.0411	0.0441	0.0501	0.0426	0.0554
0.99	12.14e0	1.769e-2	1.623e-3	2.387e-3	3.563e-3
	2.7233	0.0450	0.0458	0.0250	0.0442

Comparing the structures for VBL $= 8$ against the reference, we observe that the DF-II has the highest instances of performance degradation, in particular for $r = 0.8$ and $r = 0.95$. On the other hand, in most cases the DF-I shows comparable performance, with several exceptions in the frequency domain for $r = 0.95$. Most remarkable though, is the DCF which in almost all cases shows comparable (or better) performance. In addition to the previously mentioned argument concerning reduced limit cycles activity, we explain this with a reference to Fig. 3 which shows that the DCF has only two multiplier-related noise sources in its feedback loop, whereas the DF-I has 5. The DF-II also has only two noise sources in its recursion, but due to the direct coupling from input to output (via the coefficient b_0) in the DFC, this structure benefits from a non-filtered input signal added directly at the output. We assume this feature overrides the noise induced in the feedback loop. Note that the structures are comparable in the sense that both has 3 noise sources acting directly at the output.

Performing a time-domain comparison between the three structures for VBL $= 8$, we find a consistent indication that DF-I is superior for $\Theta = \pi/6$ and $\Theta = \pi/3$, whereas DF-II

is the better for other Θ-values, no matter the r-value. One reason for this may relate to the number of oscillations in the variables which increases for larger Θ. With fewer BBM noise sources in its recursion, the DF-II structure may be better suited to cancel the negative biased BBM noise for a fast fluctuating signal due to more positive and negative samples per time unit for larger Θ. Also, it is worth pointing out that for all structures, we see very little deviation in the performance as compared to the reference for $\Theta = \pi/2$. We explain this behaviour by the fact that for this Θ-value, the coefficient $a_1 = 0$ which therefore eliminates one of the multiplications, and therefore one noise source in the feedback loop.

Now, turning into our second experiment where $\mathrm{VBL} = 16$, we first observe an almost consistent performance degradation for all structures and pole locations (the exception being for DF-I, (r, Θ) equal to $(0.8, \pi/2)$ and $(0.95, 2\pi/3)$). First we observe, despite deviations in σ_h^2 up to a factor of 10^4, and a factor of 10 in $\hat{S}$ (as compared to the reference), that we never experienced unstable behaviour in any combination of implementation structure and pole location. This clearly indicates the ability of all three structures to operate with an approximate multiplier which generates products having the least significant half of the word nullified. This is a valuable result as it indicates (at least for the BBM) a potential significant saving in A-, T-, and PW.

Next, our simulation results provide a sound basis for a direct comparative study of the structures under the severe $\mathrm{VBL} = 16$ condition. As compared to $\mathrm{VBL} = 8$, for all structures we note a significantly smaller variation in term of both σ_h^2 and $\hat{S}$ when r increases from 0.8 towards 0.99, for all Θ-values. This indicates that all three structures become more affected when the BBM noise is increased, even for filters with a low qualify factor (i.e., for $r = 0.8$). Comparing the structures in the time domain, we see that for $r = 0.8$ and $r = 0.95$, the DF-I outperforms the DF-II and the DCF, in that order, for Θ equal to $\pi/6$, $\pi/3$, and $2\pi/3$. Again, $\Theta = \pi/2$ is a special case with $a_1 = 0$ which eliminates one multiplication in the recursion, thus making the performance of the structures almost identical. For $r = 0.99$, the DF-II has the best time domain performance for $\Theta \geq \pi/2$. Finally, a comparison in the frequency domain shows an almost opposite situation with the DCF being superior to both DF-I and DF-II (the exception being for $r = 0.99$ and $\Theta = \pi/6$). More in-depth analysis and comparisons of the time-domain and frequency-domain responses are needed before we draw any definitely conclusion on this important observation.

V. Conclusion

We have addressed the very challenging problem of using approximate multiplication in recursive 2^{nd} order filters. We have opted for the Radix-4 Broken Booth Multiplier and shown that it generates a negative biased error which can be modelled as a normal distributed additive noise source with a variance dependent on the word length d and the VBL-value. In addition to the traditional quantization noise Q, the BBM noise impacts the feedback loop and thus the output of three selected bi-quad structures, DF-I, DF-II, and DCF. For $d = 16$ we have experimented with VBL up to 16 bit, and we found that it is possible to safely operate the filters under this condition, where the lower half of the products is nullified. The structures show very distinct numerical performance towards the VBL-value as well as to the pole locations, both in the time- and frequency domain. Our results clearly indicates that for a high VBL-value and for pole locations close to the unit circle, which represents the most critical design situation, the DF-II has the better time-domain performance, whereas the DCF provides the best frequency-domain performance, the exception being for low frequency pole locations where the DF-I is superior. We have derived numerous important results which, as expected, have also led to many new research questions. Thus, our work ahead will address theoretical research which should clarify i) determination of the output SNR as a function of d and VBL, and ii) the relation between e_{mult} and Q. Additionally, we will extend our experiments beyond simulation studies and evaluate how AC bi-quads perform in real wireless and multimedia communication applications.

References

[1] P. Koch and R. Prasad, "The universal handset," *IEEE Spectrum*, vol. 46, no. 4, pp. 36–41, 2009.

[2] P. B. Kenington, *RF and Baseband Techniques for Software Defined Radio*. Artech House, 2005.

[3] D. McCarthy, "Modern receiver architectures : Considerations for spectrum monitoring applications," *IEEE Int. Symposium on Electromagnetic Compatibility, Signal and Power Integrity*, pp. 18–21, 2019.

[4] Y. Sun, G. Wang, B. Yin, J. R. Cavallaro, and T. Ly, *High-level Design Tools for Complex DSP Applications, Chapter 8 in "DSP for Embedded and Real-Time Systems"*. Elsevier Inc., 2012.

[5] S. Mittal, "A survey of techniques for approximate computing," *ACM Computing Surveys*, vol. 48, no. 4, pp. 62:1–62:23, 2016.

[6] H. Jiang, F. J. H. Santiago, H. Mo, L. Liu, and J. Han, "Approximate arithmetic circuits: A survey, characterization, and recent applications," *Proceedings of the IEEE*, vol. 108, no. 12, pp. 2108–2135, 2020.

[7] M. Pashaeifar and M. Kamal, "A theorerical framework for quality estimation and optimization of dsp applications using low-power approximate adders," *IEEE Trans. on Circuits and Systems-I: Regular Papers*, vol. 66, no. 1, 2019.

[8] ——, "Approximate adder sysnthesis for area- and energy-efficient fir filters in cmos vlsi," *IEEE 13th Int. New Circuits and Systems Conference*, 2015.

[9] W. Hui, G. Chang, V. Gormathi, R. Valarmathi, V. S. Balaji, and V. Elamara, "Revisiting fpga implementation of digital filters and exploring approximate computing on biomedical signals," *Jour. of Medical Imaging and Health Informatics*, vol. 10, no. 9, pp. 2020–2004, 2020.

[10] H. J. Butterweck, J. H. F. Ritzerfeld, and M. J. Werter, *Finite Word Length Effects in Digital Filter: A Review*. Eindhoven Univ. of Tech., Research Report, ISBN 90-6144-205-2, 1988.

[11] J. Ludwig, "Low power digital filtering using adaptive approximate processing; iir filter structures," *Int. Jour. of Electronics Communication and Computer Engineering*, vol. 12, no. 4, 2021.

[12] R. Pilipovic, V. Risojevic, and P. Bulic, "On the design of an energy efficient digital iir a-weighting using approximate multiplication," *Sensors*, vol. 21, no. 732, 2021.

[13] B. Parhami, *Computer Arithmetic, Algorithms and Hardware Designs*. Oxford University Press, 2000.

[14] F. Farshchi, M. S. Abrishami, and S. M. Fakhrarie, "New approximate multiplier for low power digital signal processing," *Proc. 17th Int. Symp. on Computer Architecture and Digital Systems*, pp. 25–30, 2013.

[15] D. Schlichthärle, *Digital Filters, Basics and Design, 2nd Ed.* Springer, ISBN 978-3-642-14324-3, 2011.

[16] A. H. Gray and J. D. Markel, "Distance measures for speech processing," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. ASSP-24, no. 5, pp. 380–391, 1976.