Accurate Rotations Based on Coefficient Scaling

Mario Garrido Gálvez, Oscar Gustafsson and Jesus Grajal

Linköping University Post Print

N.B.: When citing this work, cite the original article.

Accurate Rotations Based on Coefficient Scaling

Mario Garrido, *Member, IEEE*, Oscar Gustafsson, *Senior Member, IEEE*, and Jesús Grajal

Abstract—This brief presents a novel approach for improving the accuracy of rotations implemented by complex multipliers, based on scaling the complex coefficients that define these rotations. A method for obtaining the optimum coefficients that lead to the lowest error is proposed. This approach can be used to get more accurate rotations without increasing the coefficient wordlength as well as to reduce the wordlengh without increasing the rotation error.

The paper analyzes two different situations where the optimization method can be applied: rotations that can be optimized independently and sets of rotations that require the same scaling. These cases appear in important signal processing algorithms such as the discrete cosine transform (DCT) and the fast Fourier transform (FFT). Experimental results show that the use of scaling for the coefficients clearly improves the accuracy of the algorithms. For instance, improvements of about 8 dB in the Frobenius norm of the FFT are achieved with respect to using non-scaled coefficients.

Index Terms—Rotation, complex multiplier, coefficient scaling, error minimization.

I. INTRODUCTION

A rotation is a multiplication by a complex number whose magnitude is equal to one. This operation must be carried out in many signal processing algorithms, including transforms such as the fast Fourier transform (FFT) [1] and the fast discrete cosine transform (DCT) [2], and digital filters such as filter banks, lattice IIR filters [3] and FIR filters [4].

In digital systems, rotations can be implemented using the CORDIC algorithm [5] or complex multipliers. On one hand, the CORDIC algorithm breaks the rotation angle down into a series of micro-rotations that are carried out by shifts and additions. On the other hand, complex multipliers usually consist of four real multipliers and two adders. When complex multipliers are used, the sine and cosine components of the rotation angles are usually stored in a memory and quantized to a finite number of bits. This quantization leads to an error in the rotation. Thus, there is a trade-off between accuracy and hardware resources based on the coefficient wordlength: large coefficients lead to a lower error but increase the memory size, whereas short coefficients decrease both memory size and accuracy.

This work proposes a method for reducing the rotation error when using complex multipliers. The method is based on scaling the coefficients stored in the memory and allows

M. Garrido and O. Gustafsson are members of the Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden, e-mails: mariog@isy.liu.se, oscarg@isy.liu.se

J. Grajal is member of the Department of Signal, Systems and Radiocommunications, Universidad Politécnica de Madrid, 28040 Madrid, Spain, e-mail: jesus@gmr.ssr.upm.es

for finding the coefficients with the lowest error for a given wordlength. This not only improves the accuracy, but also has the advantage that can be done without increasing the memory size. Alternatively, the method can be used to reduce the coefficient wordlength without increasing the rotation error. This not only reduces the memory size, but also enables the simplification of the multipliers that carry out the rotation, as input coefficients have a shorter wordlength.

This method can be applied to signal processing algorithms that require the computation of rotations, such as the aforementioned ones. However, each particular case may have different optimization constraints. Sometimes rotations are independent, as occurs in lattice FIR filters [4] or in some specific DCT designs where rotations are moved to the last stage [6]. In this context, the optimum coefficients can be obtained independently for each rotation. Conversely, in common DCT and FFT hardware architectures [7]–[12] all rotations at the same stage of the architecture must be uniformly-scaled, which assures correct output results. In this case, the optimum coefficients are obtained by considering a set of angles with a common scaling factor. Both of these cases are discussed in the paper.

Moreover, many approaches in the literature consider constant multiplications and focus on reducing the number of adders required by the multiplier [6], [13], [14]. In this context, the proposed method could be combined with these approaches in order to simplify the multipliers required by the rotator.

The paper is organized as follows: the following section studies the rotation error resulting from coefficient quantization in fixed-point arithmetic. In Section III, the coefficient optimization for the case of a single angle is developed and an example of application is provided. In Section IV the optimization method is extended to a set of angles and it is applied to the rotations of the FFT. In Section V the impact that scaling has on data wordlength is discussed. Finally, some conclusions are drawn in Section VI.

II. ROTATION ERROR IN FIXED-POINT ARITHMETIC

A rotation of a complex number $x + jy$ by an angle α is calculated according to the following equations:

$$
\begin{aligned}
X &= x \cdot \cos\alpha - y \cdot \sin\alpha \\
Y &= x \cdot \sin\alpha + y \cdot \cos\alpha,
\end{aligned}
\tag{1}
$$

where $X + jY$ is the result of the rotation.

If C and S are the representations of $\cos\alpha$ and $\sin\alpha$ with a finite number of bits, a rotation using a complex multiplier can be described as

$$
\begin{aligned}
X_D &= x \cdot C - y \cdot S \\
Y_D &= x \cdot S + y \cdot C,
\end{aligned}
\tag{2}
$$

where $X_D + jY_D$ is the result of the rotation, which includes the error resulting from quantization of the coefficients.

For any angle, α, it is always true that: $\cos \alpha \in [-1, 1]$ and $\sin \alpha \in [-1, 1]$. Thus, it is generally assumed that C and S are also numbers in the range $[-1, 1]$, with a certain number of fractional bits, b. According to this typical interpretation of the coefficients, the magnitude of the rotation is always one, independent of the number of bits, b, as happens in the definition of a complex rotation in (1).

Conversely, as C and S have b bits, they can also be considered integers in 2's complement in the range $[-2^{b-1}, 2^{b-1})$, i.e., $C, S \in \mathbb{Z}_b$, where

$$\mathbb{Z}_b = \left\{ z \in \mathbb{Z} : -2^{b-1} \leq z < 2^{b-1} \right\}. \qquad (3)$$

Throughout this paper we will be following this approach, according to which, the values of the cosine and sine components of the angle, α, will be

$$\begin{aligned} C &= R \cdot (\cos \alpha + \epsilon_c) \\ S &= R \cdot (\sin \alpha + \epsilon_s). \end{aligned} \qquad (4)$$

where ϵ_c and ϵ_s are the relative quantization errors of the cosine and sine components, respectively, and R is the scaling factor of the coefficients.

It is important to note that the typical interpretation of C and S in the range $[-1, 1]$ does not take into account any scaling of the coefficients. According to the proposed approach, the typical interpretation is limited to the cases for which R is a power of two. Throughout the paper the typical interpretation will be referred to as *Non-scaled*.

Taking the aforementioned into account, the rotation error resulting from quantization of the coefficients is defined as the distance between the rotated value and the ideal one, both referring to a circumference of radius one. This requires to scale $X_D + jY_D$ by R. Therefore, in accordance with the euclidean norm,

$$\|E\| = d((X_D + jY_D)/R, X + jY) = \sqrt{\epsilon_c^2 + \epsilon_s^2} \cdot \sqrt{x^2 + y^2}. \qquad (5)$$

The total error depends on the magnitude of the input data and on the relative quantization errors. However, as the magnitude is determined by the input data, we should focus on the quantization errors of the coefficients. According to this, for a given number of bits, b, the scaling factor that leads to the lowest rotation error can be obtained by minimizing

$$\epsilon = \sqrt{\epsilon_c^2 + \epsilon_s^2}. \qquad (6)$$

III. INDEPENDENT ROTATIONS

A scaling factor R can be applied to the real and imaginary coefficients of a complex rotation in order to reduce the rotation error. Given the angle, α, and the wordlength of the coefficients, b, the optimization process consists of finding the complex number $C + jS$ that minimizes equation (6), where $C, S \in \mathbb{Z}_b$. By substituting equation (4) in (6), the optimum values are obtained by minimizing

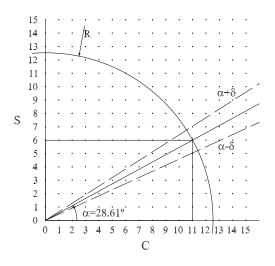$$\epsilon = \frac{\sqrt{(C - R \cdot \cos \alpha)^2 + (S - R \cdot \sin \alpha)^2}}{R}. \qquad (7)$$

Fig. 1.   Illustration of the method for obtaining the optimum coefficient that approximates the angle $\alpha = 28.61°$ using 5 bits.

As a single angle is considered, the scaling applied to the coefficients will be $R = \sqrt{C^2 + S^2}$.

Taking into account the fact that C and S can only take a finite number of values, the optimum coefficient can be obtained by calculating the error of every complex number of the form $P = C + jS$. The computations, however, can be reduced by first selecting the complex numbers with angles close to α and calculating the error only for them. This is shown in Fig. 1, where only the numbers between the angles $\alpha - \delta$ and $\alpha + \delta$ are selected, i.e., for any value of C only those values of S that meet

$$C \cdot \tan(\alpha - \delta) \leq S \leq C \cdot \tan(\alpha + \delta) \qquad (8)$$

must be considered. After this selection, the error of each candidate is calculated and finally the optimum complex coefficient, which has the lowest error, is chosen.

Figure 1 shows the case of the angle $\alpha = 28.61°$. It can be approximated by the 5-bit coefficients $11 + j6$, because $\alpha \approx \tan^{-1}(6/11)$, leading to a scaling of $R = 12.53$ and an error of $\epsilon = 8.02 \cdot 10^{-6}$. Conversely, in order to obtain a lower error, the non-scaled case requires a rotation by $14383 + j7845$. These coefficients need 15 bits, as they are in the range $[-2^{15-1}, 2^{15-1} - 1)$. In Fig. 1 it can also be visually seen that $11 + j6$ is the closest point to the line defined by the angle.

A. Example of Application: DCT with Independent Rotations

If the rotations of a DCT are moved to the last stage [6], scaling can be applied to each rotation independently. If there is a quantizer after the DCT, the scaling factors can also be incorporated into the constants in the quantizer [6]. In this context, the proposed method allows either the rotation error or the wordlength of the coefficients to be reduced at no cost.

Figure 2 shows the rotation error as a function of the number of bits for the angles used in the 8-point DCT, i.e., $\pi/16$, $\pi/8$ and $3\pi/16$. The figure includes the minimum error (Proposed), achieved by the procedure described in the previous section, as well as the error of the typical coefficients (Non-scaled).

Fig. 2 shows that the optimization method can reduce the rotation error up to several orders of magnitude. Besides, this
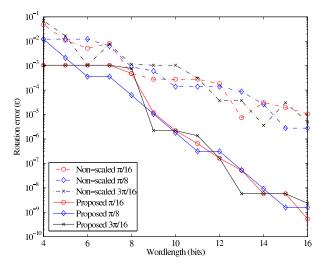
Fig. 2. Typical and optimum rotation errors as a function of the coefficient wordlength for the angles $\pi/16$, $\pi/8$ and $3\pi/16$. Each angle is optimized independently.

Fig. 3. Illustration of the method for obtaining the optimum coefficients for a set of two angles, α_1 and α_2.

improvement increases with the number of bits. Alternatively, the coefficient wordlength can be reduced significantly without increasing the error. For instance, typical 16-bit coefficients allow for obtaining errors in the range of $2 \cdot 10^{-6}$ to $2 \cdot 10^{-5}$. This error can be improved with the proposed coefficients by using only 10 bits, leading to a reduction of 6 bits.

IV. UNIFORMLY-SCALED ROTATIONS

When a set of angles α_i, $i = 1, \ldots, M$ must be scaled by the same scaling factor, R, the maximum error of the whole set should be minimized. Therefore, the function to be minimized is

$$\epsilon = \max_i(\epsilon_i) = \max_i \left(\sqrt{\epsilon_c^2(i) + \epsilon_s^2(i)} \right), \quad i = 1, \ldots, M. \quad (9)$$

As in the case of a single angle, when several angles with the same scaling are considered, a finite number of cases must be analyzed. Therefore, the minimization problem can be solved by calculating the error for all these cases. However, the number of combinations to be evaluated is $(2^b \cdot 2^b)^M$, which makes the computation of all possibilities inaccessible in many cases. Accordingly, a method for obtaining the coefficients leading to the lowest error is outlined in the next section.

A. Minimizing the Error

Let us consider the set of angles α_i, $i = 1, \ldots, M$, and the available wordlength for the coefficients, b. In this context a number $P_i = C_i + jS_i$ where $C, S \in \mathbb{Z}_b$ must be obtained for each angle α_i. The set of numbers P_i, $i = 1, \ldots, M$ defines a kernel or set of rotations. Besides, for a uniform scaling, all the points P_i must be close to the same circumference of radius R. Under these circumstances the optimization consists of finding the value of R that leads to the lowest ϵ according to equation (9).

The determination of the kernels starts by reducing the number of sets of points that have to be studied. This reduction is carried out in the first two steps of the method. Then, the selected kernels are analyzed in order to find the optimum. These steps are explained as:
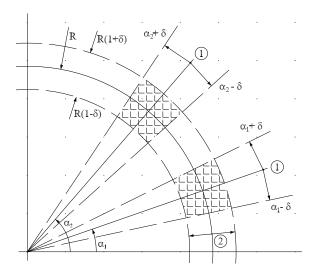
Step 1. Each angle is approximated as was done in Section III for the case of a single angle. Thus, for each angle, α_i, only those points laying between $\alpha_i - \delta$ and $\alpha_i + \delta$ are selected. This step is illustrated in Fig. 3 by ①.

Step 2. Kernels of M points are formed by picking a point P_i for each of the angles. As low-error kernels demand points with a similar radius for all the angles, each kernel will only be considered for the next step if the magnitude of all the selected points is similar. Otherwise, the kernel is discarded. This step is illustrated in Fig. 3 by ②, according to which, the kernel is selected only if any pair of points of the kernel, P_i and P_k, with magnitudes $|P_i| < |P_k|$ fulfills

$$\frac{|P_k| - |P_i|}{|P_k| + |P_i|} < \delta. \quad (10)$$

Step 3. The radius and the error of each kernel are calculated. The radius is obtained assuming that the error is the same for those two points of the kernels with the greatest error. Notice that otherwise the radius could be increased or decreased in order to reduce the greatest of these errors. The error of these two points, P_i and P_k, is equalized if

$$d\left(P_i, R(\cos\alpha_i + j \cdot \sin\alpha_i)\right) = d\left(P_k, R(\cos\alpha_k + j \cdot \sin\alpha_k)\right), \quad (11)$$

which leads to the radius being computed as

$$R = \frac{C_i^2 + S_i^2 - C_k^2 - S_k^2}{2\left(C_i \cos\alpha_i + S_i \sin\alpha_i - C_k \cos\alpha_k - S_k \sin\alpha_k\right)}. \quad (12)$$

The distances in equation (11) provide the error of the kernel, which can also be calculated by equation (9), taking into account that

$$\begin{aligned} \epsilon_c(i) &= C_i/R - \cos\alpha_i \\ \epsilon_s(i) &= S_i/R - \sin\alpha_i. \end{aligned} \quad (13)$$

Step 4. Finally, the optimum kernel, which has the lowest error, is chosen.

This method reduces the computational complexity by a factor $4M/(\delta^{M-1} \cdot \tan^M(\delta))$. For instance, if $M = 2$ and $\delta = 1°$ the computational complexity is divided by $1.5 \cdot 10^6$.

TABLE I
OPTIMUM SETS OF ROTATIONS FOR THE FFT.

Parameters		Angles					Optimum values	
Points	Bits	0	$\pi/16$	$\pi/8$	$3\pi/16$	$\pi/4$	R	ϵ
8	4	7	-	-	-	$5+j5$	7.04	5.05×10^{-3}
	5	14	-	-	-	$10+j10$	14.08	5.05×10^{-3}
	6	17	-	-	-	$12+j12$	16.99	8.67×10^{-4}
	7	41	-	-	-	$29+j29$	41.01	1.49×10^{-4}
	8	99	-	-	-	$70+j70$	98.99	2.55×10^{-5}
16	4	7	-	$7+j3$	-	$5+j5$	7.31	4.30×10^{-2}
	5	13	-	$12+j5$	-	$9+j9$	12.87	1.07×10^{-2}
	6	31	-	$29+j12$	-	$22+j22$	31.19	6.17×10^{-3}
	7	55	-	$51+j21$	-	$39+j39$	55.12	2.18×10^{-3}
	8	120	-	$111+j46$	-	$85+j85$	120.10	8.67×10^{-4}
32	4	7	$7+j1$	$7+j3$	$6+j4$	$5+j5$	7.21	5.74×10^{-2}
	5	11	$11+j2$	$10+j4$	$9+j6$	$8+j8$	11.03	2.61×10^{-2}
	6	31	$31+j6$	$29+j12$	$26+j17$	$22+j22$	31.25	1.16×10^{-2}
	7	47	$46+j9$	$43+j18$	$39+j26$	$33+j33$	46.77	4.96×10^{-3}
	8	117	$115+j23$	$108+j45$	$97+j65$	$83+j83$	117.05	2.79×10^{-3}

B. Example of Application: FFT

The procedure explained in the previous section can be used for the FFT algorithm. Because of the symmetries of the angles of the FFT in the complex plane, for an L-point kernel only the $M = L/8 + 1$ angles in the range $[0, \pi/4]$ are necessary for the optimization process. The rest of rotations of the kernel can be calculated from these angles by interchanging the real and imaginary components and/or their signs.

Table I shows the optimum kernels for 8, 16 and 32 points, which correspond to the twiddle factors W_8^i, W_{16}^i, W_{32}^i of the FFT, respectively, where $W_L^i = e^{-j \cdot 2\pi i / L}$. The errors indicated in the table, ϵ, are the minimum errors for the given number of points and bits, and each radius, R, is the optimum one, leading to the minimum error. Notice that the scaling is the same for all angles of the kernel, including $0°$. Analogously, the CORDIC algorithm also scales rotations by $0°$, which enables uniform scaling.

In Table I it can be observed that the error decreases with the number of bits, as could be expected. Likewise, for the same number of bits, the larger the number of points, the greater the error, due to the fact that the optimum conditions for a larger number of angles have to be found.

Figure 4 compares the optimum obtained kernel for 32 points and 5 bits (Proposed) with two other approaches. The first one is the typical one, for which the coefficients are not scaled (Non-scaled). The second one (Truncated one [15]) is an approximation of Non-scaled that substitutes the value 2^{b-1} by $2^{b-1} - 1$ in the coefficients, which makes it possible to represent all the coefficients using a 2's complement representation with b bits. Note that b bits in 2's complement define the range $[-2^{b-1}, 2^{b-1} - 1]$, but 2^{b-1} is out of this range.

For the comparison, the kernels in Fig. 4 are normalized. It can be noted that the proposed optimum kernel has the lowest error, as the points are closer to the intersections between the circumference and the lines defined by the angles.

The presented kernels have been applied to the FFT computation in order to reduce the output error resulting from coefficient quantization. For this purpose, a 256-point DIF radix-2 FFT [10] has been considered. For each of the cases (Proposed, Non-scaled, and Truncated one), the matrices that define the transfer function of the FFT have been obtained and
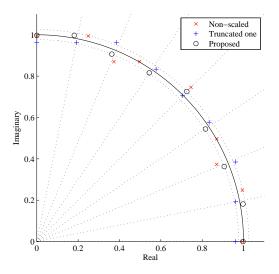

Fig. 4. Comparison of 32-point 5-bit kernels according to different approaches: Proposed, Non-scaled and Truncated one.

they have been compared to the ideal transfer function of the DFT, according to the Frobenius norm as in [13]:

$$\text{FN} = 10 \cdot \log_{10} \left(\frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |Y_{i,j} - \hat{Y}_{i,j}|^2 \right) \quad \text{(dB)}, \quad (14)$$

where Y is the ideal matrix, $\hat{Y}$ is the approximated one using quantized coefficient and N is the FFT size.

The experimental results are depicted in Fig. 5 as a function of the number of bits of the coefficients. It can be seen that the optimized kernels that have been proposed clearly improve the accuracy of the FFT as a result of the reduction in the quantization error of the coefficients. The average improvement of the proposed approach is 8 dB with respect to Non-scaled and 12 dB with respect to Truncated one.

The improvements of the proposed kernels have also been studied as a function of the FFT size, N. The results for 16-bit coefficients are depicted in Figure 6. It can be observed that for a greater number of points, the proposed approach improves 8 dB with respect to Non-scaled coefficients and 12 dB with respect to Truncated one. Note that for a small number
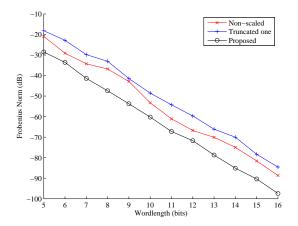
Fig. 5. Frobenius norm of a 256-point DIF FFT as a function of the number of bits for different approaches: Proposed, Non-scaled, and Truncated one.
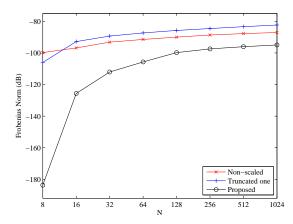

Fig. 6. Frobenius norm of a DIF FFT for 16-bit coefficients and FFT sizes from 8 to 1024 points. Different approaches are compared: Proposed, Non-scaled and Truncated one.

of angles, fewer constraints are present in the optimization, and, hence, better accuracy can be obtained. As a result, the improvements of the Frobenius norm of the proposed approach with respect to the Non-scaled case are: 84 dB for an 8-point FFT, 29 dB for 16 points, 19 dB for 32 points, and 14 dB for 64 points.

V. IMPACT ON DATA WORDLENGTH

When data truncation effects are considered, a change in the magnitude of the coefficients can affect the signal-to-noise ratio (SNR). Assuming a constant data wordlength, the scaling of the coefficients will reduce the signal level between 0 and 6 dB, while the round-off noise level is kept constant. However, when several cascaded stages are considered, data can be scaled with a factor of two everytime the signal level drops below the most significant bit. Based on this, an average reduction of 3 dB in SNR can be expected, which corresponds to half a data bit.

When dynamic data scaling is used, such as [7], [16], the scaling is determined by the actual signal levels. While the SNR may change for a specific data sequence, on average it

will not change compared to other approaches, since it is just as likely that the SNR will increase and decrease.

VI. CONCLUSIONS

This paper has presented a new approach for improving the accuracy of rotations implemented by complex multipliers. This approach is based on scaling the complex coefficients that define these rotations and a method for obtaining the optimum coefficients that lead to the lowest error has been proposed.

This optimization method can be used for improving the accuracy of rotations without increasing the memory size, by only changing the values of the coefficients stored in the memory. Alternatively, the method can be used to reduce the coefficient wordlength, while maintaining the accuracy.

The proposed approach allows the rotation error of independent rotations to be reduced up to several orders of magnitude, as shown for typical angles of the DCT. Likewise, this approach can be applied to sets of rotations that require uniform scaling, as happens in the FFT, where the Frobenius norm can be reduced about 8 dB with respect to non-scaled approaches.

REFERENCES

[1] A. Oppenheim and R. Schafer, *Discrete-Time Signal Processing*. Prentice Hall, 1989.

[2] C. Loeffler, A. Ligtenberg, and G. Moschytz, "Practical fast 1-D DCT algorithms with 11 multiplications," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, vol. 2, May 1989, pp. 988–991.

[3] J. Gray, A. and J. Markel, "Digital lattice and ladder filter synthesis," *IEEE Trans. Audio Electroacoust.*, vol. 21, no. 6, pp. 491–500, 1973.

[4] P. P. Vaidyanathan, "Passive cascaded-lattice structures for low-sensitivity FIR filter design, with applications to filter banks," *IEEE Trans. Circuits Syst.*, vol. 33, no. 11, pp. 1045–1064, Nov. 1986.

[5] M. Garrido and J. Grajal, "Efficient memoryless CORDIC for FFT computation," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, vol. 2, Apr. 2007, pp. 113–116.

[6] Z. Wu, J. Sha, Z. Wang, L. Li, and M. Gao, "An improved scaled DCT architecture," *IEEE Trans. Consum. Electron.*, vol. 55, no. 2, pp. 685–689, May 2009.

[7] Y. Chen, Y.-C. Tsao, Y.-W. Lin, C.-H. Lin, and C.-Y. Lee, "An indexed-scaling pipelined FFT processor for OFDM-based WPAN applications," *IEEE Trans. Circuits Syst. II*, vol. 55, no. 2, pp. 146–150, Feb. 2008.

[8] M. Sánchez, M. Garrido, M. López, and J. Grajal, "Implementing FFT-based digital channelized receivers on FPGA platforms," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 44, no. 4, pp. 1567–1585, Oct. 2008.

[9] M. Garrido, K. K. Parhi, and J. Grajal, "A pipelined FFT architecture for real-valued signals," *IEEE Trans. Circuits Syst. I*, vol. 56, no. 12, pp. 2634–2643, Dec. 2009.

[10] M. Garrido, "Efficient hardware architectures for the computation of the FFT and other related signal processing algorithms in real time," Ph.D. dissertation, Universidad Politécnica de Madrid, 2009.

[11] C. Cheng and K. K. Parhi, "High-throughput VLSI architecture for FFT computation," *IEEE Trans. Circuits Syst. II*, vol. 54, no. 10, pp. 863–867, Oct. 2007.

[12] S.-N. Tang, J.-W. Tsai, and T.-Y. Chang, "A 2.4-Gs/s FFT processor for OFDM-based WPAN applications," *IEEE Trans. Circuits Syst. II*, vol. 57, no. 6, pp. 451–455, Jun. 2010.

[13] S. Chan and P. Yiu, "An efficient multiplierless approximation of the fast Fourier transform using sum-of-powers-of-two (SOPOT) coefficients," *IEEE Signal Process. Lett.*, vol. 9, no. 10, pp. 322–325, Oct. 2002.

[14] O. Gustafsson and F. Qureshi, "Addition aware quantization for low complexity and high precision constant multiplication," *IEEE Signal Process. Lett.*, vol. 17, no. 2, pp. 173–176, Feb. 2010.

[15] D. James, "Quantization errors in the fast Fourier transform," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 23, no. 3, pp. 277–283, Jun. 1975.

[16] T. Lenart and V. Öwall, "Architectures for dynamic data scaling in 2/4/8k pipeline FFT cores," *IEEE Trans. VLSI Syst.*, vol. 14, no. 11, pp. 1286–1290, Nov. 2006.