Power Spectral Density of the Quantization Noise in Block-Floating-Point FFT

Gil Naveh
Toga Research Center
Huawei Technologies Co. Ltd
Tel-Aviv, Israel
Email: gil.naveh@huawei.com

***Abstract*— The Power Spectral Density (PSD) of quantization noise in fixed-point Fast Fourier Transform (FFT) implementations is traditionally assumed to be white. However, this assumption often fails in practical hardware, particularly in CPUs and DSPs employing Block-Floating-Point (BFP) arithmetic. Due to hardware efficiency constraints, these processors frequently utilize Rounding-Half-Up (RHU), a native operation that introduces an inherent statistical bias. Throughout the BFP-FFT stages, these cumulative biases propagate to the output, coloring the quantization noise and creating frequency-dependent power variations. This correlation can severely degrade performance in sensitive applications such as OFDM modems and high-fidelity audio codecs. This paper analyzes the impact of rounding bias on the noise spectrum and provides a comparative study of Decimation-In-Time (DIT) vs. Decimation-In-Frequency (DIF) topologies. We demonstrate that the DIF architecture is significantly more sensitive to this bias. Finally, we present three mitigation schemes, showing that convergent rounding, when supported by the hardware, effectively restores the output noise to a near-white spectral state.**

Keywords - BFP; Fixed Point; DIT; DIF; PSD; SQNR;

I. INTRODUCTON

The Fast Fourier Transform (FFT) serves as an important tool in many signal processing applications, such as spectral analysis, filtering, audio coding, Digital-Subscriber-Line (DSL) modems [1] wireless Orthogonal-Frequency-Division-Modulation (OFDM) modems [2] and advanced fiber optic modems [3].

Finite-word-length effects (denoted hereafter also as quantization noise) have substantial effect on the accuracy performance of FFTs. This is a result of the native characteristic of the FFT in which quantization noise that is added at the output of each stage of the FFT is accumulated toward the FFT output. Since the maximal value at each stage's output grows as we proceed with the stages [4], in many hardware implementations, the performance degradation due to the quantization noise is mitigated by adapting the register size at each stage to accommodate the signal growth [5][6]. On the other hand, in software implementations (as in CPUs and Digital Signal Processors - DSPs), increasing the bit-width is not possible. For those cases, a dynamic-scaling BFP based schemes are commonly used.

The averaged effects of the finite-word-length in FFT processing has been deeply investigated for various use cases and applications [7]-[9]. The average SQNR is an informative metric when the output quantization noise spreads evenly over all the FFT's output nodes. In practice, however, this is not necessarily the case. The quantization noise that is being added at each stage is a result of rounding the result of some arithmetic operation (multiplication, or addition followed by right shift). Usually, when the rounding is non-biased, indeed the output quantization noise spreads almost evenly over the FFT output nodes. Non-biased rounding also results in lower average noise power at the FFT output. Since non-biased rounding has higher hardware cost, common processors use a low-cost, hardware-friendly rounding, which is a biased rounding. The inherent bias in hardware-friendly rounding comes into effect in rounding half-way numbers. Although seemingly minor, this bias significantly alters the spectral shape of the quantization noise at the BFP-FFT output. Specifically, it causes the noise to become colored, meaning the noise power varies across frequency. Such non-white noise can have detrimental effects on system performance; for instance, standard OFDM channel estimation algorithms typically operate on the assumption that the additive noise is white and dominated by receiver thermal noise. When this assumption is violated by colored noise, the accuracy of the channel estimation degrades, potentially leading to increased bit-error rates in high-order modulation schemes.

The fact that biased rounding degrades the quality of FFT processors is not new. In many dedicated hardware FFTs, a convergent rounding, [10], is deployed as a non-biased rounding as in e.g., [9].

In this paper, we analyze the effects of the biased rounding used in common processors on the quantization noise at the FFT output. We focus on the PSD of the output quantization noise and compare the DIT and DIF in that regard. We show that DIT topology leads to better quantization performance over common fixed-point processors. It is also shown that DIT remains favorable even when support for non-biased, convergent rounding, is added to those processors. We use radix-2 Cooley-Tuckey FFT to demonstrate those effects, but the analysis can be extended for any other FFT radix and topology.

The paper is organized as follows: Section II introduces the models used throughout the paper covering the DIT and DIF BFP-FFT models, the underline processor model, and the quantization noise models. Section III discusses the effects of the biased quantization in the DIT and DIF BFP-

FFT. The implications on the output PSD are provided in Section IV. Algorithms for reducing the effects of the biased rounding and for whitening the noise PSD are discussed in Section V, and conclusions are given in Section VI.

II. FFT, PROCESSOR AND QUANTIZATION NOISE MODELS

We relate to fixed-point representation of fractional datatypes. We assume a processor having registers of b bits (including sign) and accumulators of at least $B = 2b + 2$ bits. The numbers represented by the registers are in 2's complement representation and in the range $-1 \le x \le 1 - 2^{-(b-1)}$. The numbers represented by the accumulators are in the rage $-2^2 \le x < 2^2$. The width of the data stored to memory is always of b bits.

Our focus is of radix-2, BFP, Cooley-Tukey, DIT and DIF FFTs. The model of a finite-word-length radix-2 butterfly of the DIT-FFT is given in Fig.1 and of DIF-FFT in Fig. 2. In the DIT topology, the inputs loaded from the memory are first multiplied by the Twiddle Factors (TFs), w_N^{kn}, then added and subtracted by the radix-2 butterfly operation before being stored back to the memory. The ADD/SUB operation is modeled as the sum of the inputs multiplied by the butterfly internal coefficients (which are $\{1, -1\}$). The processing model that we will deal here with, is a model that is common to most DSPs and dedicated FFT processors. In this model the inputs x_n and the TFs, w_N^{kn}, are represented by b bits per component (b bits for the real component and b bits for the imaginary component) and are within the range of $[-1, 1 - 2^{-(b-1)}]$. When multiplied, the multiplication is spanned over $2b + 1$ bits (recalling that the TF multiplication is a complex multiplication). The bit-width of the butterfly's output can grow to span over up to B bits and then potentially scaled down by a factor of α, where we restrict α to be a power of 2. The scaled down butterfly output is quantized to b bits per component, via rounding, before being stored to memory. In the DIF topology the inputs loaded from memory are first added and subtracted by the butterfly operation and then fed (on one of the butterfly branches) into a TF multiplier. Since the bit-width may grow by one bit after the ADD/SUB operation, a down scaling followed by quantization may take place before the multiplication. Finally, after the TF multiplication, another quantization step is required before storing the results to the memory. The quantization operation is modeled as an additive noise v and u in the diagrams. The quantization model that is used by most common processors is the RHU [10], which is also known as hardware-friendly-

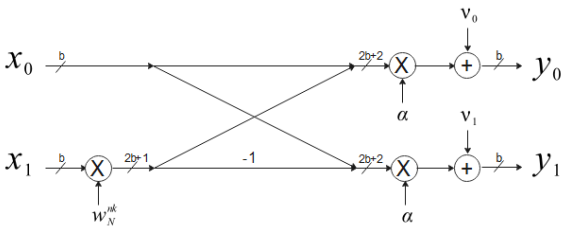

Figure 1:DIT butterfly model

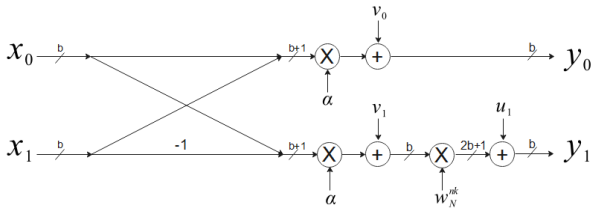

Figure 2: DIF butterfly model

rounding. The mathematical function of RHU rounding to b bits is

$$y = Q[s] \triangleq 2^{-b} \cdot \lfloor s \cdot 2^b + 0.5 \rfloor, \quad (1)$$

where $\lfloor a \rfloor$ is maximal integer lower than a and $s \in [-1{,}1 - 2^{-(b-1)}]$. The quantization error is $v = s - y$ and in the general case, when the bit-width of s is much larger than b, v is modeled as a uniformly distributed additive noise [11]

$$v \sim U[-2^{-b}, 2^{-b}), \quad (2)$$

and is statistically independent of s. As we deal here with finite-word-length, in fact v has a discrete distribution. However, for large enough b it is common to treat it as a zero mean continuous uniform distribution. As such its variance is

$$\sigma_v^2 = \frac{2^{-2(b-1)}}{12}. \quad (3)$$

In the cases that the bit-width of s is not much larger than b, the quantization noise does not behave as zero mean uniform Random Variable (RV) anymore [11]. This is also the case when a b bits number is scaled down by few bits before being quantized back to b bits. For example, relate to a b bits number, s, that is being scaled down by q bits. Since our reference numbering scheme is of fractions, the bit-width of the scaled down s is $b + q$ bits. Quantizing it back to b bits results in

$$y = Q[s \cdot 2^{-q}] = 2^{-b} \cdot \lfloor s \cdot 2^{-q} \cdot 2^b + 0.5 \rfloor, \quad (4)$$

which reflects the fact that the q least significant bits of s have been rounded out. In those cases, the resultant quantization noise is a RV having a non-zero mean, discrete distribution and its Probability-Mass-Function (PMF) depends on the number of right shifts took place. For example, in the case that such a number is shifted one bit to the right, the quantization noise ε_1 is distributed as

$$\varepsilon_1 = \begin{cases} 0 & w.p.\ 0.5 \\ -\frac{1}{2}2^{-(b-1)} & w.p.\ 0.5\,, \end{cases} \quad (5)$$

where the subscript 1 in ε_1 refers to the case of quantization noise generated by right shift of the b-bits number by one bit. The expected value of this noise equals $-2^{-(b-1)}/4$ and since it is not zero, when dealing with Signal-to-Quantization-Noise-Ratios of those RVs we will relate to the noise power rather than to its variance. We treat such noise sources herein as biased noise sources. To distinguish the power from the variance we use the symbol ρ^2 for power. The expected value of the power of ε_1 RV then is

$$\rho_1^2 = \frac{1}{2}\cdot 0 + \frac{1}{2}\cdot\left(\frac{1}{2}2^{-(b-1)}\right)^2 = \frac{2^{-2(b-1)}}{8}. \quad (6)$$

Due to the embedded bias, as expected, this is larger than the variance of the zero mean uniformly distributed quantization noise of (3). In a similar way, we can calculate the noise power of quantization noises that are generated due to the rounding after right shift of a b-bits number by q bits. In most BFP-FFT topologies and radices up to Radix-5, the right shifts are in the range of 0 to 3. Moreover, for right shifts of 4 and above the quantization noise power is very close to the variance of the zero mean uniform quantization noise of (3). Therefore, for analytical derivations we use

$$\rho_q^2 = \begin{cases} 0 & ;\ q = 0 \\ \frac{1}{8}2^{-2(b-1)} & ;\ q = 1 \\ \frac{3}{32}2^{-2(b-1)} & ;\ q = 2 \\ \frac{11}{128}2^{-2(b-1)} & ;\ q = 3 \\ \frac{1}{12}2^{-2(b-1)} & ;\ q \geq 4\,. \end{cases} \quad (7)$$

Throughout the paper, we treat the zero-mean uniformly distributed quantization noise of (3) as type-1 quantization noise, and the discrete, non-zero-mean, quantization noise with noise power of (7), as type-2 quantization noise.

III. Quantization Noise of BFP-FFT and Output SQNR

The noise at the output of a given butterfly is composed of two components: the noise that is generated by that particular butterfly, which we call butterfly self-noise, and the noise that is propagated through the butterfly (noise that was generated at earlier stages), which we call propagated-noise [12]. The propagated-noise variance, assuming type-1 noise sources, is multiplied by a factor of $2\alpha^2$ as each butterfly output is composed of the sum of two i.i.d. noise values and is multiplied by a scaling factor α. The self-noise, v, in DIT (v or u in DIF), is the noise generated by the quantization operations within the butterfly, and present at the butterfly output after being multiplied by α (refer to Figures 1 and 2). When the quantization model is of RHU, some of the noise sources are non-biased (type-1) and some are biased (type-2). Below we analyze the effects of the bias on the BFP-FFT output SQNR.

A. Biased Noise Sources in DIT BFP-FFT

In DIT butterflies, type-2 noise sources arise when all the TFs preceding a given butterfly are among the set $\mathcal{T}_1 \triangleq \{1, -1, j, -j\}\ ;\ j = \sqrt{-1}$. In the sequel, we designate the set of butterflies that all their inputs were multiplied by TFs belonging to $\mathcal{T}_1$ set, as the $\mathcal{B}_1$ set, or $\mathcal{B}_1$ butterflies. The multiplication of a b-bits value $x \in \left[-1\,, 1 - 2^{-(b-1)}\right]$ by the TF $w \in \mathcal{T}_1$ would result in a $2b$-bits number, $t = w \cdot x$, that it's lower b bits, before down scaling, are equal to zero. When such a number is scaled down by very few bits, we get the type-2 quantization noise described above. The power of those noise sources is larger than that of the type-1 noise sources, and hence they have negative effect on the power of the quantization noise at the FFT output. The distribution of the $\mathcal{B}_1$ butterflies among the FFT stages and among the butterflies within each stage is not uniform. This implies that the quantization noise power is not distributed evenly over the outputs of a given stage. By the nature of the FFT scheme, and the fact that the type-1 quantization noises are mutually independent [13], the variance of the quantization noise at each of the FFT output nodes, for FFT bearing only type-1 noise sources, is the sum of the variance of the self-noise sources of all the butterflies that this output node is connected to through the FFT flow graph, attenuated properly by the scaling factors along the flow. When incorporating the implications of the type-2 noise sources, the noise power at the output nodes grows. Moreover, due to the distribution of the $\mathcal{B}_1$ butterflies among the FFT, the noise power is not distributed uniformly at the FFT output nodes. To illustrate this, let us relate to a 16-point radix-2 FFT shown in Fig. 3. The set of TFs involved in this FFT are w_{16}^0 to w_{16}^7. Among those w_{16}^0 and w_{16}^4 belong to the $\mathcal{T}_1$ set. We also recall that one of the butterfly's inputs is always multiplied by 1, denote it as the first input, and the other, the second input, is multiplied by w_N^{kn}. As a result, the butterflies that their second input is multiplied by $w_{16}^1, w_{16}^2, w_{16}^3, w_{16}^5, w_{16}^6$ or w_8^7 result in type-1 noise source, while the butterflies that their second input is multiplied by w_{16}^0 or w_{16}^4, belong to the $\mathcal{B}_1$ set and results in type-2 self-noise source. The butterflies belonging to the $\mathcal{B}_1$ set are red-colored in Fig. 3. Since we deal here with radix-2, all the butterflies in the first two stages are among the $\mathcal{B}_1$set [12]. From the flow graph of the 16-point DIT FFT we observe that the number of $\mathcal{B}_1$ butterflies per stage is larger at the first stages and is decreased by a factor of 2 from stage to stage down to two butterflies at the last stage. This is the case for any size for radix-2 DIT FFT. From the figure it is also clear that some output nodes of the FFT are connected to more $\mathcal{B}_1$ preceding butterflies along the flow graph, while other nodes are connected to less $\mathcal{B}_1$ preceding butterflies. For example, all the butterflies preceding the output node 0 ($k = 0$) are among the $\mathcal{B}_1$ set, while for output node 1, the $\mathcal{B}_1$ butterflies preceding it are only at the first two stages. The butterflies preceding it from the last two stages are non-$\mathcal{B}_1$ butterflies. As a result, the power of the quantization noise at output node 0 is larger or equal to that at the output node 1, i.e., $\rho_0^2 \geq \rho_1^2$where equality is obtained if-and-only-if both q_3 and q_4 are equal to zero with probability 1.

Next, we wish to analyze the effects of the bias on the FFT output noise power. The propagated quantization noise at the output of a butterfly is of the form

$$\varepsilon_n^{p,(m)} = \varepsilon_n^{(m-1)} \pm w_N^{lk}\varepsilon_l^{(m-1)}, \quad (8)$$

where $\varepsilon_n^{p,(m)}$ is the propagated quantization noise at the output of node n at stage m, and $\varepsilon_n^{(m-1)}$ is the total quantization noise at the output of node n at stage $m-1$. In an FFT that incorporates type-2 noise sources, both $\varepsilon_l^{(m-1)}$, and $\varepsilon_n^{(m-1)}$ are composed of zero-mean random RV, ξ, plus some bias

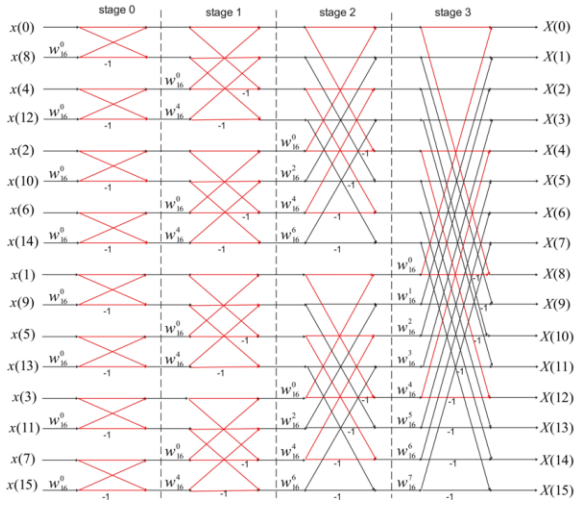

Figure 3: 16-point DIT FFT

term, μ,

$$\varepsilon_l^{(m-1)} = \xi + \mu. \tag{9}$$

In the Colley-Tuckey factorization, the two inputs to any butterfly are passing through the same set of TFs and butterflies' internal coefficients, prior to that specific butterfly. This can be observed from Fig. 3. As a result, the bias at the two inputs of any particular butterfly is the same. Denote the bias at the input of a given butterfly as μ^{in}. The propagated output bias at the two butterfly's outputs reads,

$$\begin{aligned} \mu_{i_0}^p &= \alpha\mu^{in}(1 + w_N^{nk}) \\ \mu_{i_1}^p &= \alpha\mu^{in}(1 - w_N^{nk}), \end{aligned} \tag{10}$$

where μ_i^p is the propagates bias at the butterfly's output, and α is the down scaling factor at that particular stage. From (10) it is clear that the bias at the two outputs is not identical. This is also the reason that the bias at different FFT output nodes is not the same. Let us now examine the bias at output nodes due to bias that has been generated at the first stage. The quantization noise sources at the first stage of the 16 point-FFT of Fig. 3 are all type-2 noise source. As such, they do bear an inherent bias. The actual bias value depends on the scaling factor of the first stage. Let us denote this bias as μ_1. This bias is propagated through the FFT flow graph toward the FFT output nodes. The multiplication factor it experiences on the path from the output of the first stage to any output node k of the FFT is of the form

$$\mu_k^{p_1,(3)} = \mu_1 \prod_{i=1}^{3} \alpha_i \left(1 \pm w_{16}^{n_i k}\right), \tag{11}$$

where $\mu_k^{p_1,(3)}$ is the bias that has been propagated from stage 1 toward the k^{th} output of stage 3. In the 16-point FFT there are four stages so the output of stage 3 is the output node of the FFT. From (11) it is clear that the values of the TFs, $w_{16}^{n_i k}$, along the path dictates the value of the bias at each output node. The values $w_{16}^{n_i k}$ can be constructive or destructive along the path, where the two extremes arises where $w_{16}^{n_i k}$ equals 1, leading to $\left(1 \pm w_{16}^{n_i k}\right)$ equals 0 or 2. Hence, the maximal value of the output bias due to propagation from stage 1 is $\mu_1 2^3 \prod_{i=1}^{3} \alpha_i$, while the minimum is zero. Note that if only one of the products $\left(1 \pm w_{16}^{n_i k}\right)$ along the path is zero, the propagated bias from earlier stages toward the output node will be zero.

In theory, it is possible to calculate the bias at each and every output node as a function of the scaling factors α_i. This calculation is out of the scope of this paper, but to get a glimpse of the effects of the bias on the output noise power, we will examine the noise power at the two extreme output nodes. In Fig. 4, the tracking of the paths from the output of the first stage toward output nodes $k = 0$ and $k = 8$ are highlighted. On the path toward output node $k = 0$ all the TFs are $w_{16}^0 = 1$ and all the butterflies' internal coefficients are also equal to 1. Therefore, the bias at that node gets the maximum value of

$$E_0^{(3)} = \mu_1 2^3 \prod_{i=1}^{3} \alpha_i. \tag{12}$$

On the path toward output node $k = 8$, all the TFs are also $w_{16}^0 = 1$, as well as the butterflies' internal coefficients of stages 1 and 2. But the internal coefficient at stage 3 is -1. The output bias therefore reads

$$E_8^{(3)} = \mu_1(1 + 1)(1 + 1)(1 - 1) = 0. \tag{13}$$

In order to assess the effect of the difference between the bias at output nodes $k = 0$ and $k = 8$, we have to take into account the noise power of the random component of the quantization noise, i.e., the output noise variance contributed by ξ (refer to (9)) that was generated at the outputs of stage 1. As ξ is an RV, the accumulation along the paths from the outputs of stage 1 toward the output nodes is a non-coherent accumulation. The noise variance that it contributes at output node 0 is

$$\sigma_0^2 = \sigma_\xi^2 2^3 \prod_{i=1}^{3} \alpha_i^2, \tag{14}$$

and it is identical at all the output nodes. The noise power at a given output node is the variance of the noise contributed by ξ, plus the power of the bias that was contributed by the bias μ. The ratio between the noise power at output node 0 and output node 8 is

$$\begin{aligned} \frac{\rho_0^2}{\rho_8^2} &= \frac{\left(\sigma_o^2 + \left|E_0^{(3)}\right|^2\right)}{\sigma_o^2} \\ &= \frac{\sigma_\xi^2 2^3 \prod_{i=1}^{3} \alpha_i^2 + 2^6 |\mu_1|^2 \prod_{i=1}^{3} \alpha_i^2}{\sigma_\xi^2 2^3 \prod_{i=1}^{3} \alpha_i^2} \\ &= \frac{\sigma_\xi^2 + 2^3 |\mu_1|^2}{\sigma_\xi^2} = 1 + 2^3 \frac{|\mu_1|^2}{\sigma_\xi^2}. \end{aligned} \tag{15}$$

The actual values of μ_1 and σ_ξ^2 at the output of the first stage, depend on the number of right-shifts took place at that stage. To get some sense of the ratio of (15), let us assume that there

was a single right shift at the output of the first stage. In that case, the RV $\varepsilon_1 = \xi + \mu_1$ is distributed according to (6). Its mean is $\mu_1 = -2^{-(b-1)}/4$ and its variance is $\sigma_\xi^2 = \frac{1}{16}2^{-2(b-1)}$. Plugging this into (15) we get

$$\frac{\rho_0^2}{\rho_8^2} = 1 + 2^3 \frac{\left|-2^{-(b-1)}/4\right|^2}{\frac{1}{16}2^{-2(b-1)}} = 9. \qquad (16)$$

Therefore, in the case that there was a single right-shift at the output of the first stage, the noise power at the output node 0 resulting from the bias of the first stage, is about 9.5 dB higher than the noise power at output node 8 for a 16-points FFT. As the size of the FFT grows, this ratio also grows.

B. *Biased Noise Sources in DIF BFP-FFT*

In DIF BFP-FFT butterfly, there are two quantization noise sources. The first, v, is the quantization after the down scaling that follows the ADD/SUB operation of the butterfly, and the second, u, is on the output branch that is multiplied by the TF (refer to Fig. 2). In DIF topology, the u source is of type-1, independent of the value of the TF. This is since the down scaling that follows the ADD/SUB operation of the butterfly, is determined such as to guarantee that the result of the TF multiplication never overflows. As such, no down scaling is done after the TF multiplication and the quantization noise obeys the zero-mean uniform distribution of (2). The u source, on the other hand, is a type-2 noise source, since it is a result of quantizing a $b+1$ bits number that is down scaled by very few bits to the right. The implications of the above is two folded: (a) since there are three quantization noise sources in a butterfly of a DIF BFP-FFT, as compared to only two in DIT BFP-FFT, we expect approximately $10\log_{10}1.5 = 1.76\ dB$ lower SQNR in DIF BFP-FFT implementations, and (b) All the output nodes are connected to the same number of butterflies containing type-2 noise sources. Despite the fact that all the output nodes are connected to the same number of butterflies containing type-2 noise sources, the noise power at the output nodes is not evenly distributed. This is a consequence of the multiplication factors that each noise source passes through, toward each output node, as explained for the case of DIT BFP-FFT, and is reflected in the flow graphs in Fig. 4 and

IV. Output Noise PSD

Based on the analysis of Section 0FIII, it is clear that the quantization noise at the FFT output of a DIT BFP-FFT is not white. Its PSD is not flat and the two phenomena that lead to larger noise power in some of the output nodes, are the fact that more type-2 noise sources are accumulated into some of the output nodes, and among those, at some of the nodes, the biases are accumulated "more coherently" than in other nodes. The outcome of those phenomena is reflected in Fig. 5 for 4096 points FFT (the FFT size used in the fifth generation cellular standard 3GPP 5G-NR [14]). The figure presents the SQNR per output node, which is an important metric in most systems using FFT. The figure reveals that output nodes at the DC vicinity are most affected and suffers more than 25 dB extra noise power as compared to most of the other FFT output

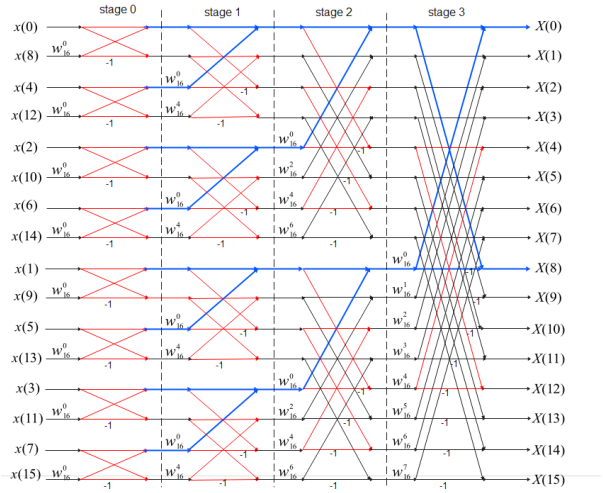

Figure 4: Path tracking toward output nodes 0 and 8

nodes. Yet this significant degradation is concentrated at small set of output nodes at the DIT case. It shall be emphasized that the effect, and the ratio of, type-2 noise sources in DIT BFP-FFT is a function of the scaling pattern, $\boldsymbol{q} = [q_1, q_2, \ldots, q_M]$, which by itself is a function of the realization of the input sequence, x_n, and can vary from one realization to another. If for example, the variance of the input sequence is very low, no down scaling would be done at the first several stages. This will lead to noise power at the output FFT nodes that is flatter than the one shown in Fig. 5. Nevertheless, for most practical use-cases, the scaling patterns lead to similar output noise PSD and SQNR as reflected in Fig. 5.

The SQNR of a DIF BFP FFT is also shown in Fig. 5. The differences to that of the DIT are clearly seen. The averaged noise power per output node of the DIF is, as expected, higher (lower SQNR), and many more output nodes in the DC vicinity suffer SQNR degradation. In the figure, periodicity patterns are observed. This stems from the fact that there are several sets of output nodes, each set suffers the same amount of noise power, but due to the nature of the output ordering (known as bit-reversal ordering), the pattern is periodic. In the DIF, the periodicity is not only due to the bias that accumulates differently between the various sets of output nodes. The imbalance of the two output-branches of the DIF butterfly with relate to the quantization noise (the upper one noise source while the lower branch suffers two), results in

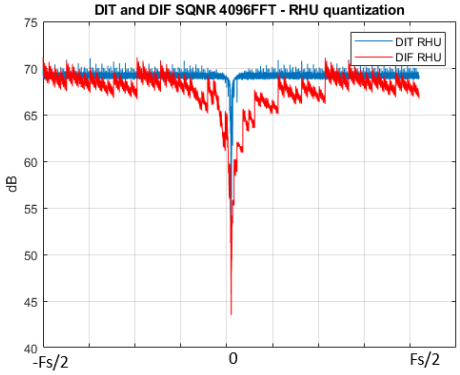

Figure 5: DIT and DIF SQNR - RHU quantization

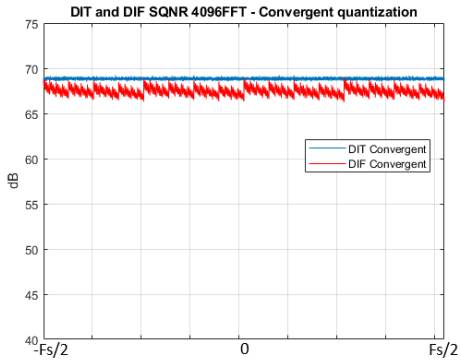

Figure 6: DIT and DIF SQNR - Convergent quantization

periodicity as well. Therefore, in DIF, even if the bias of the noise sources will be removed, a periodicity pattern will still be present.

V. WHITENING THE OUTPUT QUANTIZATION NOISE

In most of the applications using the FFT, colored quantization noise is undesirable. In many applications, some minimum SQNR for all the output nodes is required. This is the case, for example, in OFDM modems, such as 5G-NR [14]. A simple solution that sometimes is adopted in practical systems is to process the FFT at higher precision, e.g., using 32bit instead of 16bit multipliers. This is a straight forward scheme that increases significantly the output SQNR, but its drawback, of course, is the large increase in complexity. Another alternative, is to analytically calculate the bias term in each and every output node, and for each and every FFT size, and subtract it from the relevant output node. This bias term depends on the scaling pattern $\boldsymbol{q}$, and hence has to be calculated per realization of input sequence x_n . This alternative is an analytically accurate bias removal algorithm, but its drawback is that it significantly increases the complexity and the latency of the BFP-FFT. The third alternative is to use convergent rounding, [10], instead of the RHU, such as done in some hardware FFTs. Typically, the cost of adding convergent rounding to the processor's hardware is low in extra logic area, but may have slight effect on the processor's maximal frequency (and this is the reason that it is usually avoided). The effect of this alternative is presented in Fig. 6. It is clearly seen that the high noise power in the DC vicinity has been eliminated completely in both BFP-FFT topologies. The residual periodicity in the DIF BFP-FFT is the result of the imbalance of the two output branches of the butterfly as explained in Section IV.

VI. CONCLUSIONS

The effects of the classical RHU based quantization on the PSD of the quantization noise at the output of BFP-FFT is analyzed. A comparison between DIT and DIF BFP-FFT that are implementable on most of the CPUs and DSPs is provided. It is shown that for common CPU and DSP, the DIF BFP-FFT is much more sensitive to the finite-word-length effects of the processor, and results in worse SQNR and larger number of output nodes that suffer sever SQNR degradation. Three schemes to overcome the bias effects have been presented. Amending the processors with convergent rounding resolves the problem without increasing the complexity and the associated processing cycles count.

We used radix-2 FFT to convey the ideas and to present the sources of the matter. The same ideas can be extended to any other type, and topology, of BFP-FFT.

While this study focuses on uncorrelated input sequences, single-tone or other types of correlated inputs may be of interest for specific applications and are left for future work.

REFERENCES

[1] J. M. Cioffi et al. "Very-high-speed digital subscriber lines," *IEEE Communications Magazine*, vol. 37, no. 4, pp. 72-79, 1999.

[2] B. F. Frederiksen and R. Prasad, "An overview of OFDM and Related Techniques Towards Development of Future Wireless Multimedia Communications," in *IEEE Proc. Radio and Wireless Conference*, Boston, 2002, pp. 19-22.

[3] N. Cvijetic, "OFDM for Next-Generation Optical Access Networks," *IEEE Journal of Lightwave Technology*, vol. 30, no. 4, pp. 384-398, 2012.

[4] A. V. Oppenheim and C. J. Weinstein, "Effects of Finite Register Length in Digital Filtering and the Fast Fourier Transform," *Proceedings of the IEEE*, vol. 60, no. 8, pp. 957-976, 1972.

[5] W.-H. Chang and N. Q. Truong, "On the Fixed-Point Accuracy Analysis of FFT Algorithms," *IEEE Transactions on Signal Processing*, vol. 56, no. 10, pp. 4973-4682, 2008.

[6] P. Gupta, "Accurate Performance Analysis of a Fixed Point FFT," in *Twenty Second National Conference on Communication (NCC)*, Guwahati, 2016, pp. 1-6.

[7] P. D. Welch, "A Fixed-Point Fast Fourier Transform Error Analysis," *IEEE Transactions on audio and Electroacoustics*, vol. 17, no. 2, pp. 151-157, 1969.

[8] H. G. Kim, K. T. Yoon, J. S. Youn, and J. R. Choi, "8K-point Pipelined FFT/IFFT with Compact Memory for DVB-T using Block Floating-point Scaling Technique," in *International Symposium on Wireless Pervasive Computing (ISWPC)*, Melbourde, 2009, pp. 1-5.

[9] J.-R. Choi, S.-B. Park, D.-S. Han, and S.-H. Park, "A 2048 Complex Point FFT Architecture for Digital Audio Broadcasting System," in *IEEE International Symposium on Circuits and Systems*, Geneva, 2000, pp. 696-696.

[10] L. Xia, M. Athonissen, M. Hochstenbach, and B. Koren, "Improved Stochastic Rounding," *arXiv , 2020, Available: https://arxiv.org/abs/2006.00489.*

[11] B. Widrow, I. Kollar, and M.-C. Liu, "Statistical theory of Quantization," *IEEE Transactions on Instrumentation and Measurement*, vol. 45, no. 2, pp. 353-361, 1996.

[12] G. Naveh, "Finite-Word-Length-Effects in Practical Block-Floating-Point FFT," in *SIGNAL 2025*, Lisbon, 2025, pp.33-39.

[13] C. J. Weinstein, "Quantization Effects in Digital Filters," M.I.T. Lincoln Lab. Tech. Rep. 468, ASTIA doc. DDC AD-706862, 1969.

[14] *NR; Physical Channels and Modulation*, 3GPP TS 38.211, 2025.