# Finite-Word-Length-Effects in Mixed-Radix, Non-Power-of-2, Practical BFP-FFT

Gil Naveh
Toga Research Center
Huawei Technologies Co., Ltd
Hod-Hasharon, Israel
gil.naveh@huawei.com

*Abstract*— **Fixed-Point implementation of FFT is very sensitive to finite-word-length-effects due to the large quantization noise that is being accumulated throughout the FFT stages. In FFT implementations on fixed register size processors like CPUs and DSPs, Block-Floating-Point is a well-known scheme for controlling the tradeoffs between the fixed-point register size and the resultant accuracy. The performance of the radix-2 ideal Block-Floating-Point FFT, in terms of the output SQNR, has been investigated in depth. The ideal BFP-FFT suffers from implementation complexity, and especially non-deterministic latency. This results from the inherent mechanism which requires recalculating an entire FFT stage if one of that stage's outputs overflows. Because of this, most of the implementations are of a more practical variant of the BFP-FFT that does guarantee fixed latency. This, however, comes on the expense of reduced accuracy (degraded SQNR). In this paper we derive the SQNR formulas for the practical BFP-FFT for radix-2 and radix-4 Cooley-Tukey Decimation-In-Time FFTs, as well as for mixed-radix and non-power-of-2 FFTs. The derived model is compared to computer simulations and found highly accurate (less than 0.2 dB difference for the fixed radix, and less than 0.5 dB for the non-power-of-2 mixed radix). We use the derived model to compare the SQNR performance of the practical algorithm to the ideal one and show a 6-14 dB penalty for guaranteeing fixed latency implementation. For the mixed radix, the model enables to determine the optimal order of radices in terms of maximal SQNR at the FFT output.**

***Keywords- Block-Floating-Point; Fixed-Point; DIT; SQNR; Mixed-Radix; Non-power-of-2.***

## I. INTRODUCTION

The Fast Fourier Transform (FFT) serves as an important tool in many signal processing applications. It has been successfully used in many applications such as radar, spectral analysis, filtering, voice processing and modems. Some of those are heavily relying on fixed-point processing. Since the FFT algorithm is known to be highly sensitive to finite-word-length effects (which are manifested as quantization noise), many attempts to derive an analytical model of the quantization noise have been conducted throughout the years. A rigorous such analysis for Decimation-In-Time (DIT) Block-Floating-Point FFT (BFP-FFT) for radix-2 and radix-4 is given in [1].

The vast majority of the applications and use-cases where FFT is being used require a power-of-2 FFT (FFT who's size, $N$, is a power of 2, e.g., 256, 512, 1024, etc.). This is the case in filtering via the convolution theorem, in DSL multitone modems [2], in wireless modems for multimedia [3], in fiber optics modems [4] and more. In the last few decades, a demand for mixed-radix, non-power-of-2 FFT has been showing up. Examples for this demand are cellular OFDM based modems like LTE, [5], and 5G-NR, [6].

In the cases that non-power-of-2 FFTs are required, when possible, one will extend the size to the next power-of-2 size and implement a power-of-2 FFT. However, there are cases where such extension is not possible, just like in LTE [5] or 5G-NR [6]. In those OFDM modems, there exist an uplink channel which relies on modulation scheme known as single-carrier OFDM. This modulation scheme is composed of two different FFT sections. In the first section the antenna data passes IFFT of sizes that are the product of $2^{m_1}3^{m_2}$ where $m_1 \geq 7$ and $m_2$ belong to the set $\{0,1\}$. In the second section the sequence of QAM modulated symbols is FFT transformed by a non-power-of-2 FFTs of sizes that are of the product $2^{m_1}3^{m_2}5^{m_3}$ where $m_1$, $m_2$ and $m_3$ are integers complying to $m_1 \geq 2, m_2 \geq 1, m_3 \geq 0$.

Finite-word-length effects have substantial implications on the accuracy performance of FFT. 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 [7], 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 [8], [9], [10]. On the other hand, in Software implementations (as in CPUs and Digital Signal Processors - DSPs), or hardware implementations where intermediate values are forced to be written to memory, gradually increasing the bit-width of the stored values is not possible. For those cases, BFP based schemes are commonly used.

Among the BFP schemes, the dynamic-scale BFP lead to the highest accuracy for a given register size.

The straight-forward dynamic-scale BFP is such that throughout the calculation of each FFT stage, the butterflies' outputs are tested for an overflow. If an overflow is detected, the entire stage is recalculated and scaled down to prevent the overflow before being stored to memory. The advantage of this BFP scheme is that the scale down is done only on a concrete need, which leads to the best accuracy performance among other BFP-FFT algorithms. For that reason, we relate to the straight-forward dynamic-scale BFP-FFT as "ideal BFP-FFT" herein. The drawbacks of this algorithm are its complexity and the fact that that it results in non-deterministic latency. Deterministic latency may have high importance when the FFT is used within a synchronized pipelined system, such as a modulator or demodulator in OFDM modems [5].

An alternative BFP algorithm is such that there is a pre-defined down-scale factor at every stage [11]. This alternative has lower complexity and deterministic latency, but it's accuracy performance in terms of Signal-to-Quantization-Noise-Ratio (SQNR) is degraded as compared to the dynamic-scale BFP-FFT algorithms [12]. Another dynamic scale BFP scheme has been proposed by Shively [13]. In this scheme the decision of whether to scale down a certain FFT stage is determined as a function of the values of the outputs of the previous stage. That is, the decision whether to scale down a certain FFT stage is taken before the calculation of that stage is started, leading to a deterministic latency. This, on the other hand, comes on the expense of some loss in the FFT accuracy. Nevertheless, thanks to the deterministic latency of this scheme, it turns to be among the most commonly used schemes in practical implementations, e.g., [14], [15]. We refer to the Shively's scheme herein as "practical BFP-FFT". The original Shively's scheme aimed at Cooley-Tuckey, radix-2 FFT, and we extend it here to any Cooley-Tuckey, mixed-radix, power-of-2 and non-power-of-2 FFT.

The accuracy of non-BFP fixed-point FFT has been intensively analyzed as well as that of the pre-defined down-scale at every stage, e.g., [16]. The ideal BFP-FFT was originally analyzed in [17], which provided a lower and upper bound for the output quantization noise variance. In [7] and [12] a more accurate statistical model was used to project the expected value of the ideal BFP-FFT output noise power for an uncorrelated input sequence. A rigorous accuracy analysis of the practical BFP-FFT for power-of-2, fixed-radix DIT FFT is found in [1]. In the current paper we extend this analysis to mixed-radix and non-power-of-2 FFTs, where for power-of-2 we restrict ourselves to radix-2 and radix-4 (denoted as $\mathcal{R}2$ and $\mathcal{R}4$ hereafter) only. We derive an analytical model for the signal and noise power at the FFT output for any mixed-radix FFT by which the resultant SQNR can be predicted. Using the derived model one can also estimate the performance loss paid for using practical BFP as compared to an ideal BFP-FFT. For mixed-radix FFT, we show how the optimal order of radices of the given FFT size can be determined.

The problem of Twiddle Factors (TFs) quantization is not treated in this paper since the quantization effects of those are considerably lower than the computation roundoff errors [12].

The structure of the paper is as follows: in Section II the models of the underline FFT, the quantization, and the noise that are being used throughout the paper are defined. In Section III the SQNR formulas for a generic BFP-FFT scheme are derived. Section IV presents the scaling policies, and in Section V the SQNR formula for each of the scaling policies is provided. Section VI discusses the radices allocation throughout the FFT stages and the relations to the output SQNR for mixed-radix FFT. Results are presented in Section VII and the conclusions are given in Section VIII.

## 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 $B = 2b + \lceil \log_2 R \rceil + 1$ bits, where $R$ is the FFT radix and $\lceil a \rceil$ is the smallest integer that is larger than $a$. 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 range $-2^{\lceil \log_2 R \rceil + 1} \le x < 2^{\lceil \log_2 R \rceil + 1}$. The width of the data stored to memory is always of $b$ bits. We assume complex multipliers that multiply complex multiplicands of $b$ bits per component (b bits for the real component and b bits for the imaginary component). The output of the multiplier is of $2b + 1$ bits (as being a complex multiplication) that can be either scaled down and rounded to $b$ bits, or added to an accumulator.

A generic scheme of a radix-R butterfly of DIT-FFT is given in Fig. 1. The inputs, $x_n$, are loaded from the memory and first multiplied by the Twiddle Factors (TFs), $w_N^{kn}$. After multiplication by the TFs, they are multiplied by the butterfly's coefficients $\gamma_{r,t}\,; r, t \in \{0, 1, \dots, R-1\}$, and then summed up within the butterfly to get the butterfly outputs, $y_n$, before being stored back to the memory. The processing model that we will deal here with, is a model that is most common to 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 $\left[-1\,, 1 - 2^{-(b-1)}\right]$. When multiplied, the multiplication is spanned over $2b + 1$ bits. In $\mathcal{R}2$ and $\mathcal{R}4$ FFTs the butterfly's

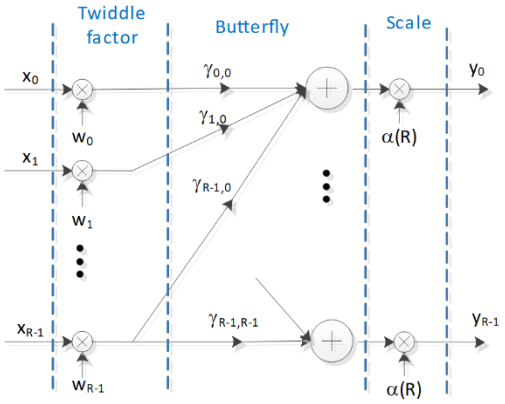


*Fig. 1: Generic model of DIT FFT butterfly*

internal coefficients, $\gamma_{r,t}$, belong to the sets $\{1,-1\}$ and $\{1,-1,j,-j\}$ ; $j=\sqrt{-1}$ respectively, and thus there are no actual multiplications within the butterfly. In those radices, the butterfly operation is, in fact, an addition or subtraction of the complex numbers or of their real-imaginary exchanged versions. This implies that the $2b+1$ bit-wide results of the multiplication by the TFs are not quantized before being summed up toward the butterfly output. The bit-width of the butterfly's output can grow and span over up to $B$ bits and then potentially scaled down by a factor of $\alpha$, where we restrict $\alpha$ to be of the form $\alpha=2^{-q}$ and $q$ is a positive integer (the number of right shifts at the butterflies' outputs). The scaled down butterfly output is quantized to $b$ bits per component, via rounding, before being stored to memory.

In radices other than $\mathcal{R}2$ and $\mathcal{R}4$ (i.e., non-power-of-2 radices), the butterflies' internal coefficients, $\gamma_{r,t}$, belong to the set $\left\{e^{-j\frac{2\pi rt}{R}}\right\}_{r,t=0}^{R-1}$, which implies that true complex multiplication takes place. Since the multiplier's multiplicands must be of $b$ bits, the results of the TF multiplication are quantized to $b$ bits before being multiplied by the butterfly internal coefficients.

The quantization model that we use here is the so-called Rounding-Half-Up (RHU) [18], which is also known as hardware-friendly-rounding and is being used in most digital signal processors and hardware implementations of digital signal processing functions. The mathematical function of RHU in rounding the value of $s$ to $b$ bits is

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

where $\lfloor a\rfloor$ is maximal integer lower than $a$ and $s\in\left[-1,1-2^{-(b-1)}\right]$. The quantization error is $v=s-y$ and in the general case is modeled as an additive noise having uniform distribution [19]

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

and is 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 continuous uniform distribution. We note also that by the definition of the RHU, $v$ has an implicit bias since half way values of $s\cdot 2^{b}$ are always rounded up. Nevertheless, in most cases that $s$ is of $2b$ bits, and $b$ is large enough, this bias is negligible and hence the variance of the quantization noise is well approximated by the uniform RV variance

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

The model representing quantized values in $\mathcal{R}2$ and $\mathcal{R}4$ butterfly is given in Fig. 2. In this model there is a single quantization operation taking place at the butterfly output before being stored to memory. It is modeled as an additive noise source $v$, and we treat $v$ as per (2). The model representing quantized values of butterflies with non-power-of-2 radices is given in Fig. 3. Here there is a noise source, $v$, modeling the quantization at the butterfly output, and a second noise source after the multiplication by the TF, $u$, that models the quantization noise caused by quantizing the result of the input multiplied by the TF.

In addition, throughout the FFT there are plenty of cases at which the butterfly's output value, before being scaled down and quantized, is a result of the summation of $b$-bits numbers multiplied by TF coefficients from the set

$$\mathcal{T}_1\triangleq\{1,-1,j,-j\}\ ;\ j=\sqrt{-1}\,, \tag{4}$$

i.e., all the coefficients toward a given butterfly output are among the set $\mathcal{T}_1$. We define those outputs as the set $\mathcal{O}$. In $\mathcal{R}2$ and $\mathcal{R}4$ butterflies, the outputs belonging to the set $\mathcal{O}$ are the outputs of butterflies that all the TFs preceding the butterfly belong the set $\mathcal{T}_1$. For the non-power-of-2 radices, the first output of any butterfly belongs to the set $\mathcal{O}$ (since $\gamma_{r,0}=e^{-j\frac{2\pi r\cdot 0}{R}}=1$ belong to the $\mathcal{T}_1$ set). In those cases, where all the coefficients toward a given butterfly output are among the set $\mathcal{T}_1$, 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 $a=w\cdot x$ that it's lower $b$ bits are equal to zero. When such

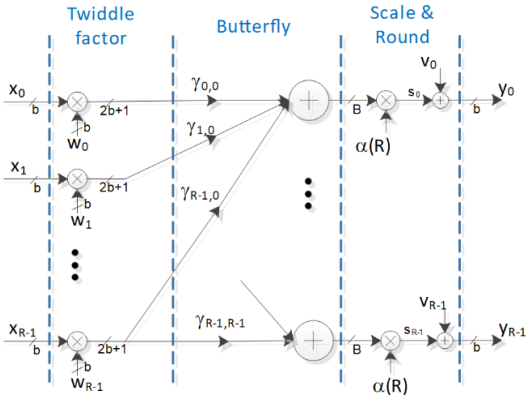


*Fig. 2: $\mathcal{R}2$ and $\mathcal{R}4$ quantization noise butterfly model*

a number is scaled down by very few bits, the quantization noise does not obey to the uniform distribution anymore [19]. In this case we get an RV having a discrete distribution and non-zero mean. For example, in the case that such a number is shifted one bit to the right, the quantization noise $\varepsilon_1$ is distributed as

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

where the subscript 1 in $\varepsilon_1$ 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 hence, when dealing with SQNRs of those RVs, we will relate to the noise power rather than to its variance. To distinguish the power from the variance we use the symbol $\rho^2$ for power. The expected value of the power of $\varepsilon_1$ then is

$$\rho_{\varepsilon_1}^2 = \frac{1}{2}\cdot 0 + \frac{1}{2}\cdot\left(\frac{1}{2}2^{-(b-1)}\right)^2 = \frac{2^{-2(b-1)}}{8}. \tag{6}$$

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 FFT topologies and radices up to $\mathcal{R}5$, 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 our analytical derivations we use

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

## III. SQNR OF A GENERIC BFP-FFT

By "generic BFP-FFT" we refer to a BFP-FFT that incorporates down-scaling by right shifts at the outputs of the FFT stages using an arbitrary scaling policy, where a scaling policy refers to the decision at which stages to scale down, and by what factor. For now, at which stages to scale down and by what factor will be parameters in the derivation. In the following paragraphs we will relate to specific BFP scaling policies and will analyze their SQNR performance. We assume zero mean i.i.d. input sequence, $x(n)$, and that the

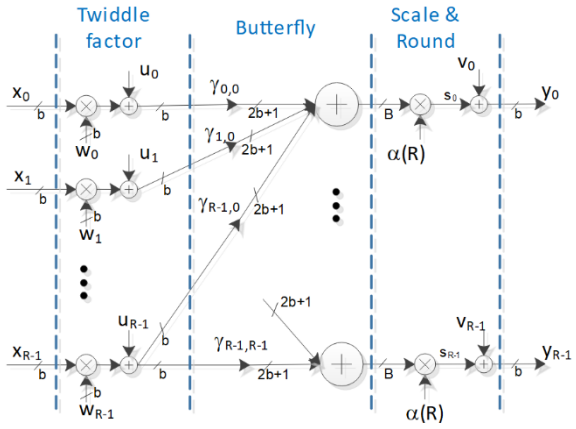


*Fig. 3: Quantization noise model for Non-power-of-2 butterfly*

quantization is regarded as an i.i.d. noise source. Moreover, multiple quantization noises at the input to a given butterfly that have been generated at earlier stages are mutually uncorrelated [12]. In order to derive the analytical expression of the SQNR, we will adopt the analysis strategy of Weinstein [12]. Let us relate to an input sequence of length $N$, $x(n)$, and a mixed-radix FFT with $M$ stages. Denote the radix of the $m^{th}$ stage as $R_m$ such that $\prod_{m=1}^{M} R_m = N$. The scale value at the $m^{th}$ stage is $\alpha_m$, $m \in \{1, 2, \dots, M\}$, where we restrict $\alpha_m$ to be of the form $\alpha_m = 2^{-q_m}$ and $q_m$ is the number of right shifts at the butterflies' outputs of the $m^{th}$ stage. We denote $x_m(n)$ as the array values at the output of the $m^{th}$ stage, where $x_M(k) \triangleq X(k)$ is the FFT output, and $x_0(n) \triangleq x(n)$ is the FFT input. For a zero mean, i.i.d. sequence $x(n)$, the variance of the signal at the FFT output is given by

$$\sigma_{x_M}^2 = N\sigma_{x_0}^2 \prod_{m=1}^{M} \alpha_m^2 = N\sigma_{x_0}^2 2^{-2\sum_{m=1}^{M} q_m}. \tag{8}$$

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. At butterflies of $\mathcal{R}2$ and $\mathcal{R}4$, the self-noise is composed of a single noise source, $v$, at the butterfly output (refer to Fig. 2), while at the other, non-power-of-2, radices it is composed of the sum of $R_m$ noise sources $u_n$, $n = 0 \cdots R_m - 1$, scaled down by $\alpha_m$, plus a single $v$ noise source at the butterfly output (refer to Fig. 3). Defining a uniform RV $\xi$ distributed as $\xi \sim U[-2^{-b}, 2^{-b})$, and denoting the variance of the self-noise at each of the stage outputs as $\sigma_B^2$, we have

$$\sigma_B^2(m) = C_m \cdot \sigma_\xi^2, \tag{9}$$

where

$$C_m = \begin{cases} 1 & ; \quad R_m \in \{2,4\} \\ (R_m \alpha_m^2 + 1) & ; \quad R_m \notin \{2,4\}. \end{cases} \tag{10}$$

To simplify the description in the sequel, we define the set of radices $\mathcal{R}2$ and $\mathcal{R}4$ as the set $\mathcal{S}$.

The propagated-noise power passing through a butterfly is multiplied by a factor of $R_m \alpha_m^2$ as each butterfly output is composed of the sum of $R_m$ i.i.d. input noise values and is multiplied by a scaling factor $\alpha_m$. Looking at the propagated-noise at the output of an M stages FFT, it is observed that the self-noise from the first stage propagates through the following M-1 stages, which results in accumulation of $\prod_{m=2}^{M} R_m$ such i.i.d. noise sources, each attenuated by a factor of $\prod_{m=2}^{M} \alpha_m^2$. The propagation of the self-noise from the second stage results in accumulation $\prod_{m=3}^{M} R_m$ such i.i.d. noise sources, each attenuated by a factor of $\prod_{m=3}^{M} \alpha_m^2$, and so on. The total output noise variance, $\sigma_E^2$, for an M stages FFT, assuming all the quantization operations are modeled as uniform RVs, $U[-2^{-b}, 2^{-b})$, is therefore given by the following expression

$$\sigma_E^2 = \sigma_\xi^2 \left( C_M + \sum_{m=1}^{M-1} C_m \prod_{i=m+1}^{M} R_i \alpha_i^2 \right). \tag{11}$$

For the sake of simplicity of the formulation, we define a virtual $(M+1)^{th}$ stage at which $\alpha_{M+1} = \frac{1}{\sqrt{R_{M+1}}}$, and re-write (11) as

$$\sigma_E^2 = \sigma_\xi^2 \left( \sum_{m=1}^{M} C_m \prod_{i=m+1}^{M+1} R_i \alpha_i^2 \right). \tag{12}$$

In (11) and (12) it was assumed that the self-noise is a continuous RV and have the same PDF at all the outputs of all the butterflies. For $b$ sufficiently large (e.g., $b = 16$) this assumption is commonly accepted. However, as explained in paragraph II, This is not the case for butterfly outputs of the set $\mathcal{O}$. The noise at the outputs of those butterflies is a discrete RVs and its Probability-Mass-Function (PMF) depends on the number of right shifts took place at the butterfly output. The power of those noise sources is larger than that of the zero-mean uniform RV, and hence they also have negative effect on the quantization noise power at the FFT output. In order to be able to evaluate the effect of those noise sources, we want to incorporate their statistical model in the derivation of $\sigma_E^2$ (or $\rho_E^2$). Before doing so, it worth mentioning two important notes. The first is that the distribution of the TFs of the set $\mathcal{T}_1$ among the FFT stages and among the butterflies within each stage is not uniform. Therefore, in each stage of radix $R \in \mathcal{S}$ there are some outputs that their self-noise has a non-uniform, non-zero-mean discrete PMF, and other outputs that their self-noise behaves as a continuous, zero-mean uniform RV. Similarly, in stages of radix $R \notin \mathcal{S}$, the first output of each butterfly is of the set $\mathcal{O}$, which has a non-uniform, non-zero-mean discrete PMF. The self-noise at the other outputs of those radices behaves as a continuous, zero-mean uniform RV. In addition, since each FFT output is connected (through the FFT flow graph) to a subset of the butterflies in each stage (except the first stage), the SQNR at the FFT output will not be identical at all output points. We will not relate to those effects here and will calculate the average SQNR at the FFT output sequence (average over all the output points). In fact, the noise power at the output of every stage of the FFT is not distributed evenly. But since we are interested in the average SQNR at the FFT output, we will also relate to the average noise power at the output of each stage of the FFT. The second note is the fact that the power of the sum of two non-zero-mean RVs does not equal to the sum of the powers like in two independent, zero-mean RVs as assumed in (12). However, since different noise sources are passing through different set of coefficients toward the same FFT output node, they can be assumed random and independent, justifying the use of the model of (12). There are very few FFT output nodes near the DC vicinity (near $k = 0$), that the set of coefficients along the path is correlated and the above assumption does not hold. Nevertheless, since the assumption does not hold only for a very small number of FFT output nodes, the effect on the overall averaged SQNR is negligible and the model of (12) can be used.

We denote by $\rho_{q_m}^2$ the noise power of a butterfly output noise source (noise source $v$) that belong to the set $\mathcal{O}$. The output noise power at those outputs is

$$\rho_O^2(m) = \begin{cases} \rho_{q_m}^2 & ; \quad R_m \in \mathcal{S} \\ (\sigma_\xi^2 R_m \alpha_m^2 + \rho_{q_m}^2) & ; \quad R_m \notin \mathcal{S}. \end{cases} \tag{13}$$

Denoting also by $\beta_m$ the fraction of the outputs belonging to the set $\mathcal{O}$ at the $m^{th}$ stage, we incorporate the effects of those outputs into the expression of the total output noise variance/power getting

$$\rho_E^2 = \sum_{m=1}^{M} [(1-\beta_m)\sigma_B^2(m) + \beta_m \rho_O^2(m)] \prod_{i=m+1}^{M+1} R_i \alpha_i^2. \tag{14}$$

Rearranging (14) and using (9), (10) and (13) we get

$$\rho_E^2 = \sum_{m=1}^{M} \left[ C_m \sigma_\xi^2 + \beta_m \left( \rho_{q_m}^2 - \sigma_\xi^2 \right) \right] \prod_{i=m+1}^{M+1} R_i \alpha_i^2 . \tag{15}$$

The second term in (15), $\beta_m \left( \rho_{q_m}^2 - \sigma_\xi^2 \right)$, is a positive quantity that represents the increased output noise power caused by outputs of the set $\mathcal{O}$. The precise expression of $\beta_m$

as a function of the radix R can be extracted from the flow graphs of the FFTs. As stated before, for $\mathcal{R}2$ and $\mathcal{R}4$, outputs of the set $\mathcal{O}$ are caused by butterflies that all their preceding TFs are among the set $\mathcal{T}_1$, and for the non-power-of-2 radices, the first output of each butterfly belong to the set $\mathcal{O}$. The general rule is that at stages of non-power-of-2 radix, the fraction of the outputs of the set $\mathcal{O}$ is the reciprocal of the radix itself, i.e., $R_m^{-1}$, while for stages of radices $\mathcal{R}2$ or $\mathcal{R}4$, the fraction of outputs of the set $\mathcal{O}$ is one at the first stage ($m = 1$), and the product of the reciprocal of all preceding radices, $\prod_{i=1}^{m-1} R_i^{-1}$ for $m > 1$. Alternatively, this can be written as $R_m \prod_{i=1}^{m} R_i^{-1}$ for any $m$. An exception is the case that $R_m = 2$ and the radices of all preceding stages are among $\mathcal{S}$. In such a case the fraction of outputs of the set $\mathcal{O}$ is $2\prod_{i=1}^{m-1} R_i^{-1} = 4\prod_{i=1}^{m} R_i^{-1}$, $m > 1$. This is given by

$$\beta_m = \begin{cases} R_m^{-1} & ;\ R_m \notin \mathcal{S} \\ 4\prod_{i=1}^{m} R_i^{-1} & ;\ R_m = 2, \{R_i, i < m\} \in \mathcal{S} \\ R_m \prod_{i=1}^{m} R_i^{-1} & ;\ Otherwise. \end{cases} \tag{16}$$

Using (16) in (15), we can calculate the quantization noise at the FFT output, $\rho_E^2$. The output SQNR for a given scale pattern, $\boldsymbol{q} = [q_1, q_2, \dots, q_M]$, can be calculated, the by $\sigma_{x_M}^2/\rho_E^2$ from (8) and (15) respectively where assigning $\alpha_i = 2^{-q_i}$.

For a mixed-radix FFT, the output noise power of (15) is a function of the radices' distribution among the FFT stages. A precise expression for the output noise is a bit cumbersome. For fixed-radix FFTs, we can get a closed form for the output noise by introducing the expression of $\beta_m$ into (15). For $\mathcal{R}2$ this results in

$$\begin{aligned} \rho_E^2 = \sigma_v^2 \sum_{m=1}^{M} R^{M-m+1} \prod_{i=m+1}^{M+1} \alpha_i^2 \\ +\left(\rho_{q_1}^2 - \sigma_\xi^2\right) R^M \prod_{i=2}^{M+1} \alpha_i^2 \\ + \sum_{m=2}^{M} \left(\rho_{q_m}^2 - \sigma_\xi^2\right) R^{M-2m+3} \prod_{i=m+1}^{M+1} \alpha_i^2 , \end{aligned} \tag{17}$$

for $\mathcal{R}4$ it results in

$$\begin{aligned} \rho_E^2 = \sigma_\xi^2 \sum_{m=1}^{M} R^{M-m+1} \prod_{i=m+1}^{M+1} \alpha_i^2 \\ + \sum_{m=1}^{M} \left(\rho_{q_m}^2 - \sigma_\xi^2\right) R^{M-2m+2} \prod_{i=m+1}^{M+1} \alpha_i^2, \end{aligned} \tag{18}$$

and for non-power-of-2, fixed-radix, in

$$\begin{aligned} \rho_E^2 = \sigma_\xi^2 \sum_{m=1}^{M} (R\alpha_m^2 + 1) \prod_{i=m+1}^{M+1} R\alpha_i^2 \\ + \sum_{m=1}^{M} \left(\rho_{q_m}^2 - \sigma_\xi^2\right) R^{-1} \prod_{i=m+1}^{M+1} R\alpha_i^2 . \end{aligned} \tag{19}$$

## IV. Scaling Policies

Theoretically, one would like to pick a scaling policy that maximizes the Signal-to-Computation-Noise-Ratio of the finite-word-length FFT algorithm. Such maximization requires the allowance of overflows, which generates overload noise, and the optimization would be over the quantization plus overload noise. However, in most practical systems, such overflows are not allowed. As a result, the scaling policy is selected to maximize the SQNR under the constraint of zero-overflows. At the ideal BFP-FFT, the scaling policy is such that throughout the butterflies' computation, every butterfly's output is tested for an overflow before it is quantized down to $b$ bits. If the real or the imaginary components of the butterfly output are smaller than $-1.0$ or larger than $1 - 2^{-(b-1)}$, the entire stage is re-calculated and the butterflies' outputs are scaled down by $q$ bits before being rounded to $b$ bits and stored to memory. The value $q$ is selected to guarantee that the scaled result does not overflow anymore. For example, if one of the absolute values of the real or imaginary butterfly's outputs is within the range $\left[1\,, 2 - 2^{-(b-1)}\right]$, the entire stage will be re-calculated while the butterflies' outputs will be shifted by one bit to the right ($q = 1$). If one of the of the absolute values of the real or imaginary butterfly's outputs is within the range $\left[2\,, 4 - 2^{-(b-1)}\right]$, the entire stage will be re-calculated while the butterflies' outputs will be shifted by two bits to the right, and so on. As was mentioned in the introduction, this scheme suffers from non-deterministic latency and therefore is less favorable in practical implementations. The second, more common, policy is the one proposed by Shively [13], which guarantees deterministic latency and lower complexity at the expense of decreased SQNR. In this policy, the decision whether to down-scale the outputs of stage $m$ and by what factor is taken based on the values of the outputs of stage $m - 1$, which are guaranteed to fit in the range $\left[-1\,, 1 - 2^{-(b-1)}\right]$. While writing the outputs of stage $m - 1$ to the memory, the processor finds the maximal absolute value among the real and imaginary components of the whole stage, and the down-scaling decision for the next stage is made according to this value. The down-scaling criterion is similar to the criterion being used by the scaling policy of the ideal BFP-FFT, i.e., to guarantee that no overflow will occur at the output of the next stage. Here, there is a need to consider the fact that the maximal absolute value at the butterflies' output of the $m^{th}$

stage would grow by a factor that is between 1 and $\sqrt{2}R_m$ relative to the outputs of stage $m-1$. In order to formalize this, let us define $x_m^c(n)$ for $n \in \{0, 1, \dots, N-1\}$ as

$$\begin{aligned} x_m^c(2n) &= real\big(x_m(n)\big) \\ x_m^c(2n+1) &= imag\big(x_m(n)\big), \end{aligned} \tag{20}$$

and

$$\tilde{x}_m = \max_n\{|x_m^c(n)|\}\,. \tag{21}$$

The scaling policy of the practical BFP-FFT can now be written as

$$q_m = \begin{cases} 0 & ;\ \tilde{x}_{m-1} < \dfrac{1}{\sqrt{2}R} \\ 1 & ;\ \dfrac{1}{\sqrt{2}R} \le \tilde{x}_{m-1} < \dfrac{2}{\sqrt{2}R} \\ 2 & ;\ \dfrac{2}{\sqrt{2}R} \le \tilde{x}_{m-1} < \dfrac{4}{\sqrt{2}R} \\ \vdots & \\ \lceil log_2(R) \rceil + 1 & ;\ \dfrac{1}{\sqrt{2}} \le \tilde{x}_{m-1} \end{cases}. \tag{22}$$

We denote the scaling policy of the ideal BFP-FFT as $\vartheta_i$ and of the practical BFP-FFT as $\vartheta_p$.

## V. SQNR CALCULATION

From the previous paragraph it is clear that the SQNR at the FFT output of a particular realization of the FFT depends on the scale pattern that has been used throughout this realization. Each scale pattern $\boldsymbol{q} = [q_1\,, q_2\,, \dots, q_M]$ is associated with a resultant SQNR. We adopt Weinstein's definition for "theoretical" SQNR as the weighted sum of the SQNR per scale pattern, i.e., the SQNR per scale pattern weighted by the probability of the particular scale pattern to occur [12]. The probability of a scale pattern depends on the radices allocation among the stages and the PDF of the input sequence. Of course, the radices allocation among the stages is a design parameter, therefore, for a given radices allocation, the probability of a scale pattern is solely dependent on the PDF of the input sequence and the scaling policy. In the sequel we will derive the scale patterns probabilities as well as the SQNR for the practical BFP-FFT algorithm and for the ideal BFP-FFT algorithm, for Gaussian input sequences. The Gaussian assumption simplifies the description, yet, the derivation can be adapted to any input sequence distribution.

### A. SQNR of practical BFP-FFT

We start with the derivation of the probabilities of scale patterns. Given the practical BFP-FFT's scaling policy, the probability that there will be exactly $q > 0$ right shifts at stage $m$ is equal to

$$\begin{aligned} &Pr\big(q_m = q; \vartheta_p\big) \\ &\quad = Pr\big(2^{q-1} \le \sqrt{2}R_m\tilde{x}_{m-1} \le 2^q\big) \\ &\quad = Pr\left(\frac{2^{q-1}}{\sqrt{2}R_m} \le \tilde{x}_{m-1} \le \frac{2^q}{\sqrt{2}R_m}\right) \\ &= Pr\left(-\frac{2^q}{\sqrt{2}R_m} \le \underset{n}{all}\{x_{m-1}^c(n)\} \le \frac{2^q}{\sqrt{2}R_m}\right) \\ &\quad -Pr\left(-\frac{2^{q-1}}{\sqrt{2}R_m} \le \underset{n}{all}\{x_{m-1}^c(n)\} \le \frac{2^{q-1}}{\sqrt{2}R_m}\right), \end{aligned} \tag{23}$$

whereas for $q = 0$

$$\begin{aligned} Pr\big(q_m = 0; \vartheta_p\big) &= Pr\big(\sqrt{2}R_m\tilde{x}_{m-1} \le 1\big) \\ &= Pr\left(\tilde{x}_{m-1} \le \frac{1}{\sqrt{2}R_m}\right). \end{aligned} \tag{24}$$

By the assumption that the input sequence, $x_{m-1}^c(n); n \in \{0, 1, \dots, 2N-1\}$ is an i.i.d. sequence, (23) and (24), can be written as

$$\begin{aligned} &Pr\big(q_m = q; \vartheta_p\big) \\ &\quad = \left[Pr\left(-\frac{2^q}{\sqrt{2}R_m} \le x_{m-1}^c(n) \le \frac{2^q}{\sqrt{2}R_m}\right)\right]^{2N} \\ &\quad -\left[Pr\left(-\frac{2^{q-1}}{\sqrt{2}R_m} \le x_{m-1}^c(n) \le \frac{2^{q-1}}{\sqrt{2}R_m}\right)\right]^{2N}, \end{aligned} \tag{25}$$

and

$$\begin{aligned} &Pr\big(q_m = 0; \vartheta_p\big) \\ &\quad = \left[Pr\left(-\frac{1}{\sqrt{2}R_m} \le x_{m-1}^c(n) \le \frac{1}{\sqrt{2}R_m}\right)\right]^{2N}. \end{aligned} \tag{26}$$

We now define the following auxiliary variables

$$T_m = 2^{-2Q_m}\,, \tag{27}$$

where

$$\begin{aligned} Q_m &= \sum_{i=1}^{m} q_i\ ;\ m \in \{1, 2, \dots, M\} \\ Q_0 &= 1\,, \end{aligned} \tag{28}$$

and

$$P_m = \prod_{i=1}^{m} R^i. \tag{29}$$

Using those, the variance of the sequence at the output of the $m^{th}$ stage is

$$\sigma_{x_m}^2 = \sigma_{x_0}^2 P_m T_m, \tag{30}$$

and the variance of the real and imaginary individual components at the output of the $m^{th}$ stage is $\sigma_{x_m}^2/2 = \sigma_{x_0}^2 P_m T_m/2$.

For an i.i.d. complex Gaussian input sequence, $x_0^c(n) \sim N\left(0, \sigma_{x_0}^2/2\right)$; $n \in \{0, 1, \dots, 2N-1\}$, it can be shown that all the intermediate sequences $x_m^c(n)$, $m \in \{1, 2, \dots, M\}$ are also Gaussian i.i.d. [12]. Therefore, the probability that the outputs of the $m^{th}$ stage would be shifted by exactly $q > 0$ right shifts, given that there were accumulated $Q_{m-1}$ right shifts at the stages preceding stage $m$ is

$$\begin{aligned} &Pr\left(q_m = q \mid Q_{m-1}; \sigma_{x_0}^2, \vartheta_p\right) \\ &= \left[erf\left(\frac{\frac{2^q}{\sqrt{2}R_m}}{\sqrt{2}\frac{\sigma_{x_{m-1}}}{\sqrt{2}}}\right)\right]^{2N} - \left[erf\left(\frac{\frac{2^{q-1}}{\sqrt{2}R_m}}{\sqrt{2}\frac{\sigma_{x_{m-1}}}{\sqrt{2}}}\right)\right]^{2N} \\ &= \left[erf\left(\frac{2^q}{\sigma_{x_0}\sqrt{2P_m R_m T_{m-1}}}\right)\right]^{2N} \\ &\quad - \left[erf\left(\frac{2^{q-1}}{\sigma_{x_0}\sqrt{2P_m R_m T_{m-1}}}\right)\right]^{2N}, \end{aligned} \tag{31}$$

and the probability that there would be no right shifts ($q_m = 0$) is given by

$$\begin{aligned} &Pr\left(q_m = 0 \mid Q_{m-1}; \sigma_{x_0}^2, \vartheta_p\right) \\ &= \left[erf\left(\frac{\frac{1}{\sqrt{2}R_m}}{\sqrt{2}\frac{\sigma_{x_{m-1}}}{\sqrt{2}}}\right)\right]^{2N} \\ &= \left[erf\left(\frac{1}{\sigma_{x_0}\sqrt{2P_m R_m T_{m-1}}}\right)\right]^{2N}, \end{aligned} \tag{32}$$

where $erf(x)$ is defined by

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt\,. \tag{33}$$

We use those per-stage probabilities to calculate the probability of a specific scale pattern, $\boldsymbol{q} = [q_1, q_2, \dots, q_M]$,

$$\begin{aligned} &Pr\left(\boldsymbol{q}; \sigma_{x_0}^2, \vartheta_p\right) \\ &= Pr\left(q_1; \sigma_{x_0}^2, \vartheta_p\right) \prod_{m=2}^{M} Pr\left(q_m | Q_{m-1}; \sigma_{x_0}^2, \vartheta_p\right), \end{aligned} \tag{34}$$

and the output SQNR is calculated by the weighted sum of the SQNRs per scale pattern as

$$\begin{aligned} SQNR_{\vartheta_p} &= \sum_{\boldsymbol{q}} Pr\left(\boldsymbol{q}; \sigma_{x_0}^2, \vartheta_p\right) \cdot SQNR\left(\boldsymbol{q}, \sigma_{x_0}^2\right) \\ &= \sum_{\boldsymbol{q}} Pr\left(\boldsymbol{q}; \sigma_{x_0}^2, \vartheta_p\right) \cdot \frac{\sigma_{x_M}^2\left(\sigma_{x_0}^2\right)}{\rho_E^2\left(\boldsymbol{q}, \sigma_{x_0}^2\right)}. \end{aligned} \tag{35}$$

In (35) the expression $Pr\left(\boldsymbol{q}; \sigma_{x_0}^2, \vartheta_p\right)$ is calculated by (34), $\sigma_{x_M}^2\left(\sigma_{x_0}^2\right)$ is calculated by (8) and $\rho_E^2\left(\boldsymbol{q}, \sigma_{x_0}^2\right)$, with $\alpha_i = 2^{-q_i}$, is calculated by (17), (18) or (19) for $\mathcal{R}2$, $\mathcal{R}4$ and non-power-of-2 radices respectively. The number of different $\boldsymbol{q}$ patterns is quite large (e.g., for $\mathcal{R}2$, since $q_m$ can take one of three options $\{0,1,2\}$ there are $3^{\log_2 N}$ optional different patterns). Nevertheless, the summation over all the $\boldsymbol{q}$ patterns in (35) can be calculated in reasonable time via a computer program.

Since we focus the analysis here on Gaussian inputs which are un-bounded in their values on one hand, while the FFT under analysis requires inputs in the range $\left[-1, 1 - 2^{-(b-1)}\right]$ on the other hand, we select the variance of the input signal such that the probability for values outside the allowed range at the input is sufficiently low. For the sake of the current analysis, we used $\sigma_{x_0} = 0.15$ which leads to a very low probability of having a sample outsize the allowed range. For example, for 4096 points FFT, the probability of having a vector of size 4096 with a sample outside the range $[-1,1]$ is approximately $10^{-7}$ (once per ten million FFT realizations, in average, there will be an input sample that has to be saturated to $\left[-1, 1 - 2^{-(b-1)}\right]$).

### *B. SQNR of the ideal BFP-FFT*

At the scaling policy of the ideal BFP-FFT, $\vartheta_i$, there are no pre-decisions for per-stage scaling. An FFT stage is calculated without scaling and throughout the calculations, if any of the stage's outputs overflows the allowed range, the whole stage is re-calculated while the outputs are down-scaled before being written to memory. Note that in the ideal policy there may be multiple re-calculation of the same stage if the strategy is to initiate the re-calculation upon the first overflowed value (strategy (a)). Different strategies that will eliminate the multi re-calculations of the same stage are: (b) upon the detection of the first overflow - set the scale value to the maximal scale value, and (c) always calculate the stage to its end and if overflows have been detected throughout the calculation, set the scale value according the largest magnitude among the detected overflowed values. Note that strategy (b) suffers degradations in the SQNR performance due to potential mismatch between the scale value and the actual maximal overflow value. Nevertheless, here, for the sake of SQNR comparison, we assume strategy (a) or (c), meaning that the scale is according to the largest magnitude output sample and no performance loss is involved. As opposed to the practical case where the scale decision for

stage $m$ depends on $x_{m-1}(n)$, which are the outputs of stage $m-1$ after being scaled down, the scale decision of the ideal BFP-FFT depends on the output of stage $m$ before being scaled down. Let us denote the output of stage $m$ before being scaled down as $s_m(n)$, such that the scaled down values are

$$x_m(n) = \alpha_m s_m(n), \tag{36}$$

and define $s_m^c(n)$ and $\tilde{s}_m$ in analogous to (20) and (21) as

$$\begin{aligned} s_m^c(2n) &= real\big(s_m(n)\big) \\ s_m^c(2n+1) &= imag\big(s_m(n)\big), \end{aligned} \tag{37}$$

and

$$\tilde{s}_m = \max_n\{|s_m^c(n)|\}. \tag{38}$$

Now the SQNR analysis using the ideal BFP-FFT policy follows the steps of the analysis of the practical BFP-FFT scheme. The output signal variance and the output noise power follow (8) and (15), respectively. The probability that there will be exactly $q > 0$ right shifts at stage $m$ is equal to

$$\begin{aligned} Pr(q_m = q;\vartheta_i) &= Pr(2^{q-1} \le \tilde{s}_m \le 2^q) \\ &= Pr\left(-2^q \le \underset{n}{all}\{s_m^c(n)\} \le 2^q\right) \\ &-Pr\left(-2^{q-1} \le \underset{n}{all}\{s_m^c(n)\} \le 2^{q-1}\right), \end{aligned} \tag{39}$$

and the probability that there will be no right shifts at stage $m$, i.e., $q = 0$, is

$$\begin{aligned} Pr(q_m = 0;\vartheta_i) &= Pr(\tilde{s}_m \le 1) \\ &= Pr\left(-1 \le \underset{n}{all}\{s_m^c(n)\} \le 1\right). \end{aligned} \tag{40}$$

Under the i.i.d. Gaussian input assumption, we get for $q > 0$

$$\begin{aligned} Pr\big(q_m = q \mid Q_{m-1};\sigma_{x_0}^2,\vartheta_i\big) &= \left[erf\left(\frac{2^q}{\sigma_{x_m}}\right)\right]^{2N} \\ &- \left[erf\left(\frac{2^{q-1}}{\sigma_{x_m}}\right)\right]^{2N} \\ &= \left[erf\left(\frac{2^q}{\sigma_{x_0}\sqrt{P_m T_{m-1}}}\right)\right]^{2N} \\ &- \left[erf\left(\frac{2^{q-1}}{\sigma_{x_0}\sqrt{P_m T_{m-1}}}\right)\right]^{2N}, \end{aligned} \tag{41}$$

and for $q_m = 0$

$$\begin{aligned} Pr\big(q_m = 0 \mid Q_{m-1};\sigma_{x_0}^2,\vartheta_i\big) &= \left[erf\left(\frac{1}{\sigma_{x_m}}\right)\right]^{2N} \\ &= \left[erf\left(\frac{1}{\sigma_{x_0}\sqrt{P_m T_{m-1}}}\right)\right]^{2N}. \end{aligned} \tag{42}$$

## VI. RADICES ALLOCATION

For a mixed-radix FFT, the order of the radices (the allocation of radices to the various stages which forms a radices pattern) is a design parameter. Different orders will result in different scale pattern distributions and as a result - different output SQNR. In fact, the total amount of scaling (right shifts) of the ideal BFP-FFT for a given input realization depends solely on the values of the instantaneous input realization, and is independent of the order of radices. The number of right shifts in this case can be shown to be

$$Q_M = \left\lceil \log_2 \max_k(|Real\{X(k)\}|, |Imag\{X(k)\}|) \right\rceil, \tag{43}$$

where $X(k)$ is the FFT output for the specific input realization, assuming no scaling take place throughout the FFT. At the practical BFP-FFT the total number of down scaling is not completely independent on the order of the radices. It depends on the radix allocated to the last stage, stage $M$, and is in the range $\{Q_M, Q_M+1, \cdots, Q_M + \lceil \log_2(\sqrt{2}R_M)\rceil\}$.

The output noise, on the other hand, does depend on the scaling patterns, while those depend on the order of the radices. The variance of the resultant SQNR between various radices-patterns is not large and is shown to be in the range of 0.2 $dB$ to 2.25 $dB$ for the LTE DFT sizes. An easy way to determine the best order of radices is to calculated the SQNR (according to (35)) for all the radices permutations and pick the one with the highest SQNR. In Fig. 4 the best and worst SQNR among all the radices permutations for each of the LTE DFT sizes is shown. An interesting observation from Fig. 4 is that for the non-power-of-2, mixed-radix FFT of the LTE sizes, the SQNR is not necessarily a monotonic function of the FFT size. As can be seen there is an average monotonicity, but not local monotonicity. The reason is the fact that in close sizes, despite the fact that the size is close, the set of radices involved is different. Since the quantization noise generated by a butterfly of non-power-of-2 radix is larger than that of a butterfly of power-of-2 radix (refer to (9) and (10)), an FFT that involves more non-power-of-2 radices, is likely to result in larger output quantization noise. For example, the sizes of 324, 360 and 384 are three consecutive sizes in Fig. 4, and show monotonic increasing SQNR. When examining the radices involved, we find that size 324 include four stages of non-power-of-2 radices (since $324 = 4 \cdot 3^4$), size 360 include three stages of non-power-of-2 radices ($360 = 4 \cdot 2 \cdot 5 \cdot 3^2$), and the size 384 include only one radix which is non-power-of-2 ($384 = 2 \cdot 4^3 \cdot 3$).

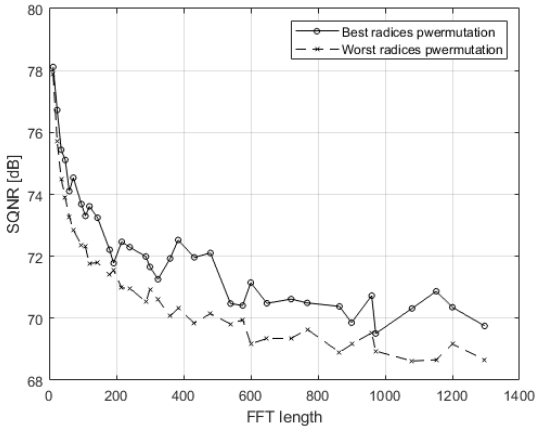


*Fig. 4: SQNR of best and worst radices permutations for non-power-of-2 FFTs of LTE sizes*

## VII. RESULTS

The derived models of the SQNR of the practical and the ideal BFP-FFT have been validated against simulation. The model and the simulation results for 16-bit datatype ($b = 16$) and Gaussian i.i.d. input with standard deviation of $\sigma_{x_0} = 0.15$ are shown in Fig. 5 and Fig. 6 for radix-2 and radix-4 respectively. The simulation result vs. the BFP model for non-power-of-2, mixed radix, practical BFP-FFT of the LTE sizes is shown in Fig. 7. For the simulation results we have averaged the SQNR of 1000 FFT runs per FFT length. As can be seen, there is a very good match between the simulation results and the derived model in all cases. The gap between the refined statistical model (that incorporate the refinement for butterfly outputs of the set $\mathcal{O}$) and the simulation result for the practical BFP-FFT is in the order of 0.2 dB for the fixed-radix, power-of-2 FFTs and in the order of 0.5 dB for the mixed-radix, non-power-of-2 FFTs. The simulation results for the ideal BFP-FFT are not shown in the figures since the model has almost perfect match to the simulation result with gaps that are in the order of 0.05 dB.

In Fig. 5 and Fig. 6 we can also see the effect of the refined statistical model for the butterfly outputs of the set $\mathcal{O}$. In Fig. 5 is it seen that the model neglecting the effects of the butterfly outputs of the set $\mathcal{O}$, for radix-2 BFP-FFT, is optimistic by about 0.5 dB for the practical BFP-FFT and in Fig. 6 it is optimistic by about 1 dB for radix-4.

One of the main goals of the paper is to provide an analytical tool that enables the prediction of the SQNR penalty one needs to pay for getting fixed latency BFP-FFT. This penalty is clearly seen for radix-2 and radix-4 in Fig. 5 and Fig. 6 respectively. We see that such a penalty is in the order of 6 dB when the number of stages is above five, and grows up to 13.5 dB for lower number of stages as seen at the case of 64 points radix-4 FFT. The reason that for low number of stages the degradation of the practical BFP-FFT is larger, is the fact that the difference between the number of truly required down-scales (used by the ideal BFP-FFT) and the number of down-scales used by the practical BFP-FFT (Shively's scheme) reduces as the number of stages grows and that in the practical BFP-FFT the scaling take place at earlier stages.

Another interesting observation that the model reveals relates to the comparison of the SQNR between radix-2 and radix-4 BFP-FFT implementations for a power-of-2 fixed-radix FFT. It is well known that from complexity perspective, the radix-4 has advantages over radix-2 (at least in the number of multiplications). From the results in Fig. 5 and Fig. 6, we can also see that radix-4 have better SQNR in the ideal BFP-FFT implementation. We get 4 dB advantage for 64-points FFT down to about 2 dB advantage for 4096-points FFT. However, for the practical BFP-FFT we see an opposite behavior. The radix-2 practical BFP-FFT results in 2.8 dB better SQNR for 64-point FFT, down to 1.2 dB better SQNR for 4096-points FFT. The reason for this phenomenon is that the number of the quantization noise sources depends on the number of stages, such that in the radix-4 FFT there are half the number of noise sources as compared to radix-2, while the total down-scaling depends on the type of the BFT-FFT. For ideal BFP-FFT the total down scaling of radix-2 and radix4 is the same (as given in (43)). Hence, since radix-2 has more quantization sources, it also has lower SQNR performance as compared to radix-4. For the practical BFP-FFT, number of down-scaling of the radix-2 and radix-4 FFTs may not be the same. Since the maximal absolute value is a monotonic, non-decreasing, function of the stage index (it always non-decreasing between consecutive stages) [7], the number of down-scales of the practical BFP-FFT would be greater or equal to that of the radix-2. As a result, the signal power at the output of the radix-4 practical BFP-FFT is lower or equal that that of the radix-2 and hence, despite the fact that there are more noise sources in radix-2 the total SQNR is better

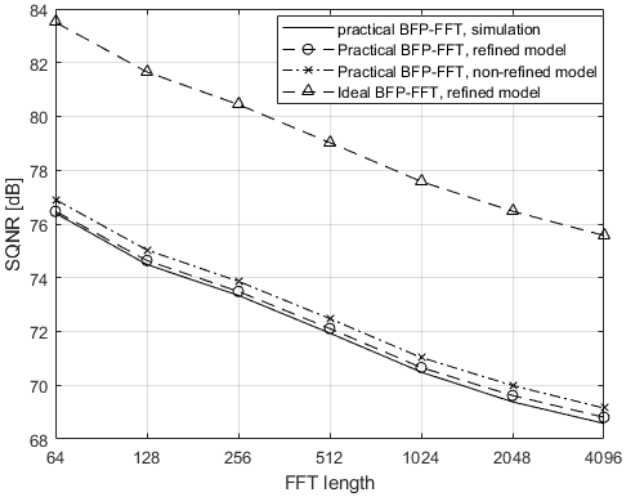


*Fig. 5: Radix-2 Practical BFP-FFT*

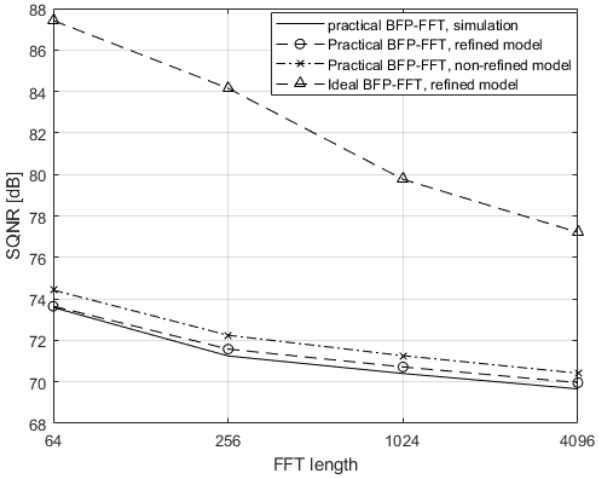


*Fig. 6: Radix-4 Practical BFP-FFT*

## VIII. CONCLUSIONS

In this paper we extended the analytical model of the finite-word-length-effects of Cooley Tukey DIT BFP-FFT of [1] to cover fixed-radix, as well as mixed-radix, non-power-of-2 FFTs. We incorporate butterfly outputs belonging to the $\mathcal{O}$ set as a refined model, and derived the analytical expressions for the ideal and practical BFP-FFTs. The models have been validated against simulation and found highly accurate for both, the ideal and the practical BFP-FFTs. The model enables to accurately predict the SQNR for the practical BFP-FFT and the performance degradation compared to the ideal BFP-FFT scheme. The model also can be used to determine the best radix order of mixed-radix FFTs as described in paragraph VI.

The derivation covers DIT-FFT and refer to a straight-forward implementation model of non-power-of-2 butterfly. The framework used can be easily adapted to other topologies and other implementation models of the non-power-of-2 butterflies.

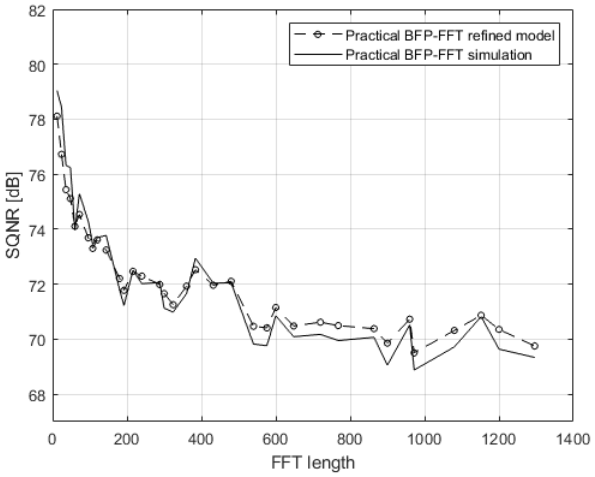


*Fig. 7: Mixed-Radix LTE sizes Practical BFP-FFT*

## REFERENCES

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

[2] J. M. Cioffi, V. Oksman, J. J. Werner, T. Pollet, P. Spruyt, J. S. Chow and K. S. Jacobsen, "Very-high-speed digital subscriber lines," *IEEE Communications Magazine*, vol. 37, no. 4, pp. 72-79, 1999.

[3] 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.

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

[5] *LTE-A; Evolved Universal Terrestrial Radio Access (E-UTRA), Physical Channels and Modulation*, 3GPP TS 36.211, 2011.

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

[7] 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.

[8] 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.

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

[10] A. Monther and K. Zsolk, "Analysis of Quantization Noise in FFT Algorithms for Real-Valued Input Signals," in *International Conference on Radioelektronika*, Kosice, 2022.

[11] S. Qadeer, M. Z. Ali Khan and S. A. Sattar, "On Fixed Point error analysis of FFT algorithm," *ACEEE Int. Journal on Information Technology*, vol. 01, no. 03, 2011.

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

[13] R. R. Shively, "A Digital Processor to Generate Spectra in Real Time," *IEEE Transactions on Computers*, Vols. C-17, no. 5, pp. 485-491, 1968.

[14] 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.

[15] S. J. Huang and S. G. Chen, "A High-Parallelism Memory-Based FFT Processor with high SQNR and novel addressing scheme," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, Montreal, 2016.

[16] Tran-Thong and B. Liu, "Fixed-Point Fast Fourier Transform Error Analysis," *IEEE Transactions on Acoustic, Specch, and Signal Processing*, vol. 24, no. 6, pp. 563-573, 1976.

[17] 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.

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

[19] 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.