FINITE WORD LENGTH ANALYSIS OF THE RADIX-2^2 FFT

Hernán G. Rey and Cecilia Galarza

Facultad de Ingeniería, Universidad de Buenos Aires
Paseo Colón 850 (1063), Buenos Aires, Argentina
{hrey@fi.uba.ar} {cgalar@fi.uba.ar}

ABSTRACT

In this paper, we analyze the quantization error effects of the radix-2^2 FFT algorithm. We propose *per tone* models for the error power. This is a different approach from the common choice of a maximum or mean value over the spectrum. In particular, we treat three different errors: 1) due to input quantization, 2) due to coefficient quantization and 3) due to quantization after a multiplication. This analysis is applied to a DMT scheme. Simulation results agree with the theoretical predictions.

1. INTRODUCTION

The *Discrete Fourier Transform* (DFT) is probably the most important tool in discrete time signal processing. Since the appearance of Cooley-Tuckey algorithm, the search of efficient algorithms for computing the DFT has been covered extensively in the literature [3].

When these algorithms, usually grouped into the *Fast Fourier Transform* (FFT) family, are implemented with finite precision arithmetics, several error sources appear. This subject has been thoroughly studied since the middle of the sixties [4]. In this work, we focus on three particular errors:

1. due to input quantization.
2. due to coefficient quantization.
3. due to quantization after a multiplication.

Per tone error models are provided for the radix-2^2 FFT algorithm, a novel fast computation scheme introduced by He and Torkelson [2]. It has the benefits of having the computational performance of the radix-4 FFT, but with the hardware requirements and ease of implementation of the radix-2 FFT.

The idea is to provide a model that predicts the noise power due to quantization errors *at each output tone*. Usually, the maximum or mean value over all the spectrum is used to represent the error power. But we could be particularly interested in a *per tone* error power. This would be the case when each tone represents an independent channel, and th FFT is considered as a mapping (modulation) tool. An example of this situation appears in the *Discrete MultiTone* (DMT) modulation technique [5]. The consequence of the FFT quantization errors is the degradation of the *Signal to Noise Ratio* (SNR).

We show that the radix-2^2 FFT is a suitable algorithm for DMT schemes, and in particular to an *Asymmetric Digital Subscriber Line* (ADSL) modem. Trough a comparison with the widely used radix-2 FFT algorithm, we found that the improvement is not only referred to its computational efficiency and low hardware requirements, but also to its better performance to finite word length effects.

2. THE RADIX-2^2 FFT

The FFT is a family of efficient algorithms for computing the DFT of a discrete time signal. This transform is defined as:

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk} \qquad k = 0, 1, \ldots, N-1,$$

where the DFT coefficients are powers of $W_N = e^{-j2\pi/N}$, also known as *twiddle factors*, and represent the N-th roots of unity. The basic tool of FFT algorithms is *divide and conquer*: an N-point FFT is divided in a set of smaller ones.

If $N = 2^{\nu_2}$, with $\nu_2 \in \mathbb{Z}$, the radix-2 decimation in frequency FFT could be used. This approach divides by half the FFT lengths at each stage. By performing two consecutive decompositions, it results:

$$X(4k_3 + 2k_2 + k_1) = \sum_{n_3=0}^{N/4-1} \sum_{n_2=0}^{1} \sum_{n_1=0}^{1} x(n_1 N/2 + \\ + n_2 N/4 + n_3) W_N^{(n_1 N/2 + n_2 N/4 + n_3)(4k_3 + 2k_2 + k_1)}, \quad (1)$$

where $n = n_1 N/2 + n_2 N/4 + n_3$. The procedure is followed until the ν_2 stages are completed. The total number of multiplications is $\mathcal{O}(N \log_2 N)$, i.e., proportional to $N \log_2 N$. The savings in the number of computations come from applying the *complex conjugate symmetry* and the *periodicity* properties satisfied by the FFT coefficients [3].

If $N = 4^{\nu_4}$, with $\nu_4 \in \mathbb{Z}$, the sequence could be divided in four at each stage. This procedure is performed by the radix-4 decimation in frequency FFT, which has a complexity of $\mathcal{O}(N \log_4 N)$. However, the basic structures used in the computation, also known as *butterflies*, are more complicated than the *butterflies* used in the radix-2 FFT.

He and Torkelson [2] proposed the radix-2^2 FFT, an algorithm with the computational performance of the radix-4, but with the hardware requirements and ease of implementation provided by the radix-2. If we sum over n_1 in Eq. (1), we find:

$$X(4k_3 + 2k_2 + k_1) = \sum_{n_3=0}^{N/4-1} \sum_{n_2=0}^{1} \Big[B_{N/2}^{k_1}(n_2 N/4 + n_3) \cdot \\ \cdot W_N^{(n_2 N/4 + n_3)k_1} \Big] W_N^{(n_2 N/4 + n_3)(4k_3 + 2k_2)}, \quad (2)$$

where $B_{N/2}^{k_1}(u) = x(u) + (-1)^{k_1} x(u + N/2)$.

The main idea of the new algorithm is to join the twiddle factor $W_N^{(n_2 N/4 + n_3)k_1}$ with the other exponentials *before* forming the next butterfly. This combination improves the

This work has been supported in part by the Universidad de Buenos Aires, in the context of the project UBASoft.

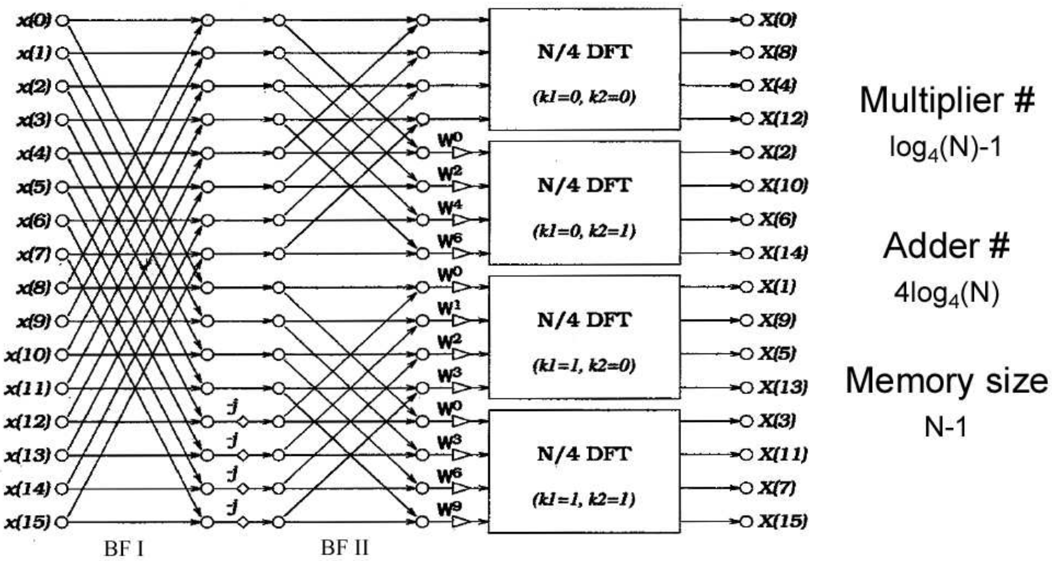

Figure 1: Radix-2^2 FFT scheme with $N = 16$ [2].

number of multiplications as:

$$W_N^{(n_2N/4+n_3)(4k_3+2k_2+k_1)} = W_N^{Nn_2k_3} W_N^{n_2N/4(2k_2+k_1)} W_N^{4n_3k_3} \cdot$$
$$\cdot W_N^{n_3(2k_2+k_1)} = (-j)^{n_2(2k_2+k_1)} W_N^{n_3(2k_2+k_1)} W_N^{4n_3k_3}.$$

Using this fact and summing over n_2 in (2), i.e., doing the second stage of the DFT decomposition, the FFT takes the form of:

$$X(4k_3+2k_2+k_1) = \sum_{n_3=0}^{N/4-1} \left[H(k_1,k_2,n_3) W_N^{n_3(2k_2+k_1)} \right] W_{N/4}^{n_3k_3}, \tag{3}$$

where

$$H(k_1,k_2,n_3) = \overbrace{B_{N/2}^{k_1}(n_3)}^{\text{BFI}} + (-j)^{(2k_2+k_1)} \underbrace{B_{N/2}^{k_1}(n_3+N/4)}_{\text{BFI}} \tag{4}$$

The first stage DFT decomposition is carried out by BF I, while the second one is done by BF II (see Fig. 1). The last step of the algorithm performs the multiplications by $W_N^{n_3(2k_2+k_1)}$. Finally, Eq. (3) shows how to regroup the results to form 4 DFTs of length $N/4$. The described procedure could be applied again to these DFTs until the desired output is obtained. The output comes in bit reversed order, as shown in Fig. 1.

3. QUANTIZATION ERRORS ANALYSIS

From this point on, we consider that the real and imaginary parts of the input signal are represented in fixed point two's complement system and have magnitude less than one.

When an FFT is computed using finite word length, three different errors appear, which are caused by: 1) input quantization. 2) coefficient quantization. 3) quantization after a multiplication. We treat them separately, meaning that when one of them is considered, the others are assumed to be negligible.

3.1 Input Quantization Error

An A/D (*Analog to Digital*) converter is usually the first stage of a digital system. The real an imaginary parts of the input (which are assumed to be independent) are quantized using $B_i + 1$ bits, introducing a complex error with variance $\sigma^2(\varepsilon_i)$.

When two points of the input are added (or subtracted), the resulting error has twice the original power. As the roots of unity have magnitude 1 and assuming that the errors are decorrelated, each output tone will have an error power due to input quantization equal to the sum of all the error powers of the input components. Thus, at the $\overline{k}$-th output node[1], the mean squared error is:

$$E[|e_i(\overline{k})|^2] = N\sigma^2(\varepsilon_i). \tag{5}$$

This result is true for the radix-2 and radix-2^2 FFTs, because they only differ in the way they distribute the multiplications along the signal flow.

3.2 Coefficient Quantization Error

When an FFT is computed, the roots of unity are required. They could be previously computed and stored or computed on-line. In both cases, an error source appears by using a quantized coefficient W_N^q (represented with $B_c + 1$ bits) instead of the original W_N.

We want to model the error due to coefficient quantization at the $\overline{k}$-th output node, i.e. $e_c(\overline{k})$. To this end, we start by looking at Fig. 1. From (4), the BF II gives the result:

$$x_{BFII}(\overline{k}) = x(\overline{k}) + (-1)^{k_1} x(\overline{k}+N/2) + (-j)^{(2k_2+k_1)} \cdot$$
$$\cdot \left(x(N/4+\overline{k}) + (-1)^{k_1} x(3N/4+\overline{k}) \right),$$

where the index are computed modulus N. After that, the first coefficient multiplication takes place. During next stage,

[1] The output node index $\overline{k}$ corresponds to the *k-th* tone after a bit reversing operation.

four quantities of the form $x_{BFII}(i)W_i^q$ (W_i^q is the quantized coefficient that is used depending on the position in the signal flow) are summed up. Then, the result is multiplied by another coefficient W_j^q. Thus, the resulting error has the form of:

$$e_c(\overline{k}) = \sum_{i=1}^{N/4} x_{BFII}(i) \left(\prod_{j=1}^{v_4} W_{i,j}^q - \prod_{j=1}^{v_4} W_{i,j} \right).$$

If we assume that the input sequence is i.i.d. (independent and identically distributed), the mean squared error becomes:

$$E[|e_c(\overline{k})|^2] = 4\sigma_x^2 \sum_{i=1}^{N/4} \left| \prod_{j=1}^{v_4} W_{i,j}^q - \prod_{j=1}^{v_4} W_{i,j} \right|^2, \qquad (6)$$

where σ_x^2 is the input signal variance.

3.3 Quantization After a Multiplication Error

Each time we perform a nontrivial multiplication, i.e., the ones that are not by ±1 or $\pm j$, an error occurs due to the quantization of the result to $B_r + 1$ bits. We model this error as an additive term ε_r at the output of the multiplier with power $\sigma^2(\varepsilon_r)$. We make the following assumptions [3]:

1. The error takes values on the interval $[-\Delta/2; \Delta/2]$, where $\Delta = 2^{-B_r}$. The first and second order moments of the error are the same as if it was uniformly distributed in this interval.
2. The errors are decorrelated among them and with respect to the input sequence.

These assumptions have been widely discussed in the bibliography [1][6]. Its validity depends on certain conditions on the characteristic function of the number to be quantized . Although these conditions are not satisfied by many distributions, they could be *approximately satisfied.*

It is important to notice that each complex coefficient has magnitude one, because we are not considering the coefficient quantization error. Then each error source makes the same average power contribution to the output. Therefore, the total average error power at one output node, e.g. $E[|e_r(\overline{k})|^2]$, will be $\sigma^2(\varepsilon_r)$ multiplied by the total number of multiplications that affect the considered node.

Due to the strategy used by the radix-2^2 FFT, the first stage performs a decomposition in four blocks. The first one, ($k_1 = 0$ and $k_2 = 0$) has only trivial multiplications by one. The other blocks, has one multiplier equal to one (W_N^0) and in particular, the block associated with $k_1 = 0$ and $k_2 = 1$ presents one multiplication by $-j$. Through this decomposition procedure, the total number of nontrivial multiplications needed to calculate an FFT of length $N > 4$, has the form of:

$$\sum_{i=1}^{v_4 - 1} \frac{3}{4}N - 4^i = N\left(\frac{3}{4}(v_4 - 1) - \frac{1}{3}\right) + \frac{4}{3}.$$

On the other hand, the radix-2 FFT makes $N((v_2 - 1)/2 - 1) + 2$ nontrivial multiplications. Given that $v_4 = v_2/2$, when N is large, it could be seen that the radix-2^2 FFT gives a 25% save in the total number of nontrivial multiplications over the radix-2 FFT.

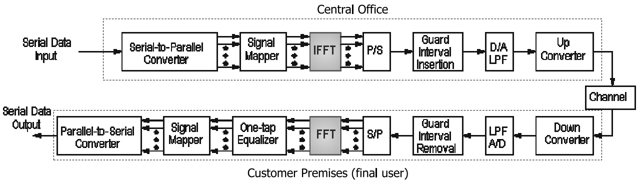

Figure 2: DMT transmission block diagram. FFT blocks are the shaded blocks.

However, for the *per tone* model we are interested in the total number of multiplications that propagate to a certain output node. So at each stage, there is one butterfly that should be considered, and only if it has a nontrivial multiplication associated. As a consequence of the exposed analysis, we formulate the following theorem:

Theorem 1 *Let $0 \leq \overline{k} \leq N-1$ be the index of a node in the output array. This value could be represented in a quaternary system using v_4 bits. If the least significant bit (b_0) is dropped and $b_{v_4 - 1}$ is considered as the most significant bit, the total number of nontrivial multiplications associated with the tone at position $\overline{k}$ is:*

$$\sum_{i=1}^{v_4 - 1} m_i,$$

where

$$m_i = \begin{cases} 0 & \text{if} b_i = 0, \\ 4^i - 2 & \text{if} b_i = 1, \\ 4^i - 1 & \text{if} b_i = 2 \, \text{or} 3, \end{cases}$$

■

This result is obtained by careful analysis of the signal flow (see Fig. 1).

In this case, the maximum number of nontrivial multiplications for a single tone takes place on each output node where its quaternary index is represented using only the digits 2 or 3. Then,

$$\sum_{i=1}^{v_4 - 1} 4^i - 1 = \frac{4^{v_4}}{3}(1 - 4^{-v_4}) - v_4 = \frac{N - 3v_4 - 1}{3}. \qquad (7)$$

In the radix-2 FFT, the maximum number of multiplications for a single tone is $N - 2v_2$. So, when $N = 256$ this is equal to 240. If we do the same computation for a radix-2^2 FFT using (7), the maximum number of nontrivial multiplications is 81. Thus, there is a saving of three times in the maximum error variance per tone by using the radix-2^2 FFT instead of the radix-2 FFT.

4. AN APPLICATION: THE ADSL TRANSMISSION

4.1 The ADSL modem

A brief scheme of a DMT transmission system could be seen in Fig. 2. The data sent by the modem at the central office is modulated as a DMT symbol. This symbol is formed by 256 complex values for the particular case of an ADSL transmission. Each complex number belongs to a QAM constellation, whose size was previously determined by a loading algorithm. After that, the set of 256 numbers is complex conjugate duplicated, resulting in a 512-point sequence which is viewed as the frequency characteristic of the symbol. Now, an IFFT (*Inverse Fast Fourier Transform*) takes place, in order to obtain the 512-point real valued sequence that is transmitted. When this data is received at the final user's modem, a 512 real FFT is computed, in order to recover the original data.

4.2 Experimental setup

Now, we focus on how the results of the previous section could be applied to a DMT communication system, and particularly, to an ADSL receiver.

By exploiting the fact that the input sequence is real, we use a 256-point complex radix-2^2 FFT. The error is generated by subtracting a double precision FFT to the quantized FFT. The sequence length was set to $N = 256$ points and real arithmetic was used. The real and imaginary parts of the input sequence were independent and uniformly distributed. The simulation results shown are obtained by ensemble averaging over 2000 independent trials of the experiment.

4.3 Analysis of results

We corroborate the results predicted by (5), (6) and Theorem 1 for the different error sources. Due to space limitations, we don't show all the simulation results; although there is a good match with the theoretical ones.

In the coefficient quantization case, the variance of the error is not constant for all tones and the shape of the picture depends on the choice of B_c, as predicted by (6). Although the input variance does not change, the deterministic part associated with the values of W^q causes the mismatch between plots with different B_c.

An important aspect to notice is based on how the error power is distributed along the output array (and as a consequence, along the whole bandwidth) in the *per tone* model due to quantization after a multiplication. As we said before, at each stage of the radix-2^2 FFT there is a division in four blocks. The first quarter of the output array has less noise power than the other quarters. This pattern is repeated inside each block. This is in contrast with the radix-2 FFT, where a monotonically increasing (in groups of 4 consecutive points) behavior is presented.

Finally, the combination of the three errors was simulated for a 64-QAM, and the results are shown in Fig. 3. For the theoretical prediction, we assume that the three error sources are independent. Thus the total error variance would be:

$$E[|e_{tot}(\overline{k})|^2] = E[|e_i(\overline{k})|^2] + E[|e_c(\overline{k})|^2] + E[|e_r(\overline{k})|^2]. \quad (8)$$

With the simulated parameters $B_i = 7$, $B_c = 15$ and $B_r = 7$, the hypothesis made in the formulation of our models are not far from being satisfied. This allows a good agreement between theoretical and simulated values. In particular, it can be seen from (8) that the error due to input quantization is the dominant one. This fact is usually encountered in practice because the use of more accurate A/D converters increases the cost of the modem.

5. CONCLUSIONS

In this work we proposed *per tone* quantization error models for the radix-2^2 FFT. This is important when each tone represents an independent channel and the FFT is considered as a mapping tool. An example of this situation is DMT modulation. Simulation results are in agree with the theoretical ones.

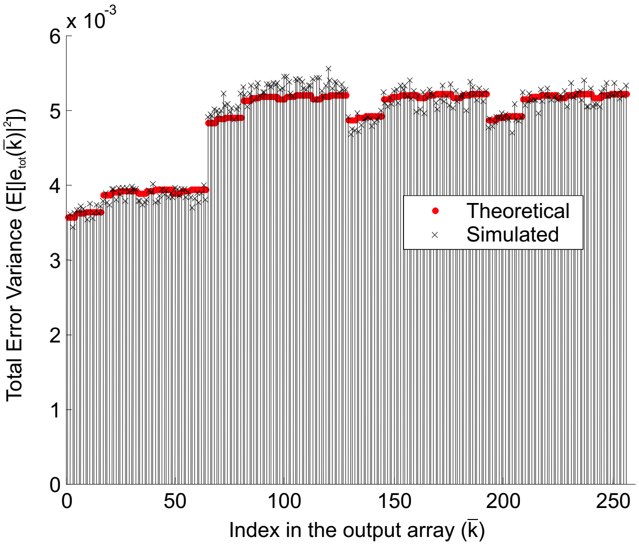

Figure 3: Error variance considering the three errors. Input: 64-QAM. $B_i = 7$, $B_c = 15$, $B_r = 7$.

Particularly, the radix-2^2 FFT appears as a suitable choice for an ADSL receiver. The first reason for this is that it gives an easy implementation with few hardware requirements and better computational performance than the radix-2. Second, it has an error power reduction with respect to the radix-2 FFT. It has not only a lower maximum error power value, but also a different shape of the noise power characteristic.

REFERENCES

[1] C.W. Barnes, B.N. Tran and S.H. Leung, "On the Statistics of Fixed-Point Roundoff Error", *IEEE Transactions on Signal Processing*, Vol. 33, No. 3, pp. 595–606, Jun. 1985.

[2] S. He and M. Torkelson, "A New Approach to Pipeline FFT Processor", *Proc. IPPS-96*, Honolulu, Hawaii, Apr. 1996, pp. 766–770.

[3] A.V. Oppenheim and R.W. Schafer, *Discrete-Time Signal Processing*, International Edition, Prentice Hall, New Jersey, 1989.

[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, Aug. 1972.

[5] T. Starr , J.M. Cioffi and P. Silverman, *Understanding Digital Subscriber Line Technology*, Prentice-Hall, New Jersey, 1998.

[6] P.W. Wong, "Quantization Noise, Fixed-Point Multiplicative Roundoff Noise, and Dithering", *IEEE Transactions on Signal Processing*, Vol. 38, No. 2, pp. 286–300, Feb. 1990.