Article

Selecting FFT Word Length for an OFDM Receiver That Supports Undersampling

Nikos Petrellis

Department of Electrical and Computer Engineering, University of Peloponnese, 26334 Patra, Greece; npetrellis@uop.gr; Tel.: +30-2610-369236

Received: 28 February 2020; Accepted: 19 March 2020; Published: 3 April 2020

Abstract: In this paper, we focus on Orthogonal Frequency Division Multiplexing (OFDM) transceivers where undersampling is employed by the receiver Analog/Digital Converter (ADC) when sparse information is exchanged. Several Fast Fourier Transform (FFT) symmetry properties are exploited to allow the substitution of specific input values by others that have already been sampled by the ADC. Several architectures have been proposed in the literature for efficient FFT implementations in terms of power, speed and hardware resources. The FFT input/output values, twiddle factors, etc., are complex numbers with their real and imaginary parts being represented using fixed point format. A tradeoff has to be made between rounding error and complexity. The optimal minimum FFT word length is investigated by combining the undersampling and the rounding error. A configurable new FFT architecture has been developed in hardware description language to test the error model with various FFT sizes, word lengths and Quadrature Amplitude Modulations (QAM). A system designer can take into account the sparseness of the input data and define the desired rounding and undersampling error relation. The developed error model would then predict the required word length and ADC resolution with average Root Mean Square Error (RMSE) less than 1.

Keywords: FFT; sparse; OFDM; word length; rounding error

1. Introduction

Recovering information from fewer samples is possible if data are sparse or compressible. In this case, an ADC can operate in a sampling rate closer to the actual information rate rather than the Nyquist one [1]. The ADCs in this case are often called Analog to Information Converters (AIC) [2]. Compressive Sampling or Sensing (CS) methods employ iterative optimization techniques like Regressive Analysis or Orthogonal Matching Pursuit [3] to recover information from fewer measurements. The hardware implementation of a CS algorithm requires a large number of resources due to its increased complexity. CS techniques are applied in image processing applications such as radar, medical imaging (such as Magnetic Resonance Imaging (MRI), ultrasounds, X-ray imaging), surveillance systems, etc. For example, in [4], the acquisition time needed for MRI scans is significantly reduced. In [5], radar data are compressed and decompressed using CS techniques with Normalized Mean Square Error (NMSE) ranging between 1.5 and 2.5 under certain measurement conditions.

In OFDM environments channel estimation is achieved using CS techniques since it can be assumed that the OFDM channel is sparse. In this way, the number of pilots can be substantially reduced as described in [6], where 511 subcarriers and 20 pilots are used, and only a few of the 40 channels taps are assumed non-zero. The Bit Error Rate (BER) is approximately 0.003 if Signal/Noise Ratio (SNR) is 30dB and 5 of the 40 channel taps are non-zero. The efficient implementation of sparse FFT and Inverse FFT (IFFT) in OFDM environments is an important target since these modules are computationally intensive and power greedy. In [7] sub-sampling of the input signal is performed with

$O(k \cdot \log k)$ complexity. For example, using a 280×280 pixel image in [7] with the number of non-zero pixels being 3509, the NMSE is 0.0085 when 35308 samples are used.

In the work presented in [8,9], the sparse IFFT input samples of an OFDM transmitter are ordered appropriately to allow the information recovery on the receiver side from fewer FFT input samples. In this way, the power consumption of the receiver ADC is reduced down to the half while the memory requirements, power dissipation and the speed of the IFFT and the FFT modules can also be reduced. More specifically, a wired OFDM architecture is studied in [8], where 1024-point FFT/IFFT is employed. The simulation results show that full information recovery can be achieved if 50% of the time the ADC operates in 7/8 of its normal rate and the sparseness s in input data is less than 2% i.e., less than 2% of the input bits are non-trivial. If a lower ADC sampling rate is employed, an error floor appears that may be acceptable in some applications. Full information recovery can also be achieved if lower ADC sampling rate is applied when sparseness level is higher. The only noise source taken into consideration in [8] is Additive White Gaussian Noise (AWGN). The effect of the ADC Quantization Errors (QE) and the Round-off Errors (RE) caused by the representation of the operands with limited number of bits is not studied in [8]. The proposed undersampling technique is extended in [9], to cover wireless OFDM transceivers with Space-Time Block Code (STBC) encoding. A general Signal to Interference, Distortion Noise Ratio (SIDNR) expression is derived taking into consideration the ADC quantization noise as modeled in [10] as well as wireless channel features like Inter-Symbol-Interference (ISI).

In the present work, we focus on the most critical part of the OFDM transceiver system that is the FFT on the receiver side. A precise error model has been developed by describing how the employed undersampling technique can affect the overall error in the received data. This Undersampling Error (UE) is combined with the RE/QE errors. The target is to optimally select the appropriate word length for the representation of the FFT input values and coefficients in order to avoid excessive additional error overhead beyond the error caused by the UE. For example, the end user may ask for an RE error that will not exceed 10% of the UE. Selecting the desired modulation scheme and FFT size the developed error model will estimate the minimum word length to achieve the specified relation between RE and UE for a particular sparseness level of the input data. The real numbers (or the real/imaginary parts of a complex number) can be represented in IEEE-754 standard 32-bit floating-point format [11]. In this format, a real number is described by one bit for the sign, a number of bits for the significand and a signed exponent. Although the IEEE-754 floating point format can cover a wide range of real numbers with a very high precision, the number of resources that it needs is excessively high. For this reason, fixed point format is employed in this work. In fixed point format (b, f) , b bits are used for the description of a number and f of these b bits are used for the fraction: $B_{b-1} \dots B_{f+1} B_f B_{f-1} \dots B_1 B_0 = B_{b-1} 2^{b-f-1} + \dots + B_0 2^{-f}$. Several operations such as multiplications and divisions by 2 can be implemented with simpler circuits in fixed point format but the results of these operations have to be monitored for scaling and overflow errors.

A memory-based pipeline configurable FFT has been developed in synthesizable Very high speed IC Hardware Description Language (VHDL) to accurately assess the proposed error model. Two FFT sizes have been tested (256 and 1024 points) with different number of word lengths (ranging from 5 to 10 bits). The alternative modulations tested were 16-QAM and 4-QAM with various levels of data sparseness ranging between 0.5% and 10%. The OFDM configuration in terms of FFT size, modulation and word length cannot change dynamically in real time. They are statically defined in order to measure the error for various sparseness levels. The average RMSE error between the predicted word length and the experimental results is less than 1, for the various QE error targets that have been set. The predicted word length can also be used as the ADC resolution.

This paper is organized as follows: In Section 2, the undersampling method presented in [8] and [9] is briefly described as well as some indicative QE and RE error estimation methods presented in the literature. The combined UE, RE error model proposed in this paper is presented in Section 3. The synthesizable memory-based pipeline FFT architecture used to assess the error model is presented

in Section 4. Finally, in Section 5 the simulation results are presented along with a discussion on how the error modeling followed can be useful for other non-OFDM telecommunication systems.

2. Background

2.1. Proposed Undersampling Method on OFDM Receiver Side

In an OFDM transmitter, the binary input data stream is encoded for the generation of a parity bit stream used for error correction on the receiver side. The data and parity bit streams are usually interleaved in order to avoid burst errors. Then, groups of $\log_2 q$ bits are mapped to q -QAM constellation symbols X_k ($0 \leq k < N$) that form the parallel input to an N -point IFFT. The output of this IFFT are the symbols x_n ($0 \leq n < N$) that are serially transmitted over the channel. Pilot symbols with known value are placed on reserved subcarriers for channel estimation and equalization. A cyclic prefix is also appended to avoid Inter-Symbol Interference (ISI). Digital/Analog Conversion (DAC) is required for the transmission of the resulting symbols using an appropriate pulse shaping method [12,13]. In wired OFDM transceivers the channel noise is assumed to be Additive White Gaussian Noise (AWGN) and the y_n symbols received are $y_n = x_n + z_n$, (z_n is the noise with variance σ_n^2). A different model is used for wireless channels that takes into consideration the reflections, interference, Rayleigh fading, etc. In optical communications the fiber channels are affected by several sources of distortion including Kerr non-linearity, chromatic dispersion, optical filtering, double Rayleigh scattering, shot and thermal noise and especially Amplified Spontaneous Emission (ASE) [12]. The y_n symbols at the output of the receiver ADC, form the input of an FFT. The FFT output symbols Y_k ($0 \leq k < N$) are mapped to the closest QAM symbol and then QAM demodulation is performed (e.g., using hard or preferably soft decision demodulators). Forward Error Correction (FEC) decoding (Viterbi, Turbo codes, etc) exploits the available parity bits in order to correct as many errors as possible on the receiver side.

A Recursive Systematic Convolutional (RSC) FEC encoder can be used such as the one with feed forward and feedback polynomials: $1 + D + D^2 + D^3$ and $1 + D + D^2$ respectively, where the D^p denotes a delay of p clock periods. In order to apply the proposed undersampling method, the Interleaver should generate q -QAM symbols derived from parity or data bits only. Thus, a pair of small buffers at the FEC encoder output can store temporarily $\log_2(q)$ bits from the systematic and the parity output of the RSC encoder before they are mapped to the corresponding q -QAM symbol. Most of the q -QAM symbols derived from sparse data bits will have a common value X_c . However, several parity q -QAM symbols derived from parity bits are likely to have identical values because the parity output of the employed RSC encoder described above remains '0' until a first data '1' appears. Then, the 7-bit pattern "0111010" is repeated as long as the data input remains '0'. A different 7-bit pattern is generated when another '1' appears at the input and this is also repeated until a third '1' appears. The distance between each '1' at the FEC encoder input is expected to be high due to the sparseness in the input. Consequently, several identical consecutive 7-bit patterns are expected to appear at the parity output of the FEC encoder. These identical parity patterns can be treated as if they were X_c symbols as will be explained below.

The proposed undersampling method is taking advantage of the data sparseness in time domain in conjunction with some well-known properties of the Discrete Fourier Transform (DFT). Proper symbol arrangement is employed at the IFFT input, allowing the substitution of several symbols at the receiver FFT input, by others without significant loss of information. In this way, the receiver ADC can periodically relax (operate in lower rate) without sampling some y_n symbols since they can be substituted by others, that have already been received. The adopted symbol arrangement at the IFFT input, makes trivial several IFFT/FFT operations that can be omitted in order to achieve lower power and faster operation. This is similar to output pruning described in [14].

If $w_N^r = e^{i2\pi r/N}$ are the twiddle factors then DFT is defined as:

$$Y_k = \sum_{n=0}^{N-1} y_n w_N^{-kn}, \quad (1)$$

and the Inverse DFT (IDFT):

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k w_N^{kn} \quad (2)$$

One of the IDFT symmetry properties used by the proposed undersampling method concerns the relationship between x_n and $x_{n+N/2}$ when n is odd:

$$x_n = \frac{1}{N} \left(\sum_{k=0,2,\dots}^{N-2} X_k w_N^{kn} + \sum_{k=1,3,\dots}^{\frac{N}{2}-1} (X_k - X_{k+\frac{N}{2}}) w_N^{kn} \right), \quad (3)$$

$$x_{n+N/2} = \frac{1}{N} \left(\sum_{k=0,2,\dots}^{N-2} X_k w_N^{kn} - \sum_{k=1,3,\dots}^{\frac{N}{2}-1} (X_k - X_{k+\frac{N}{2}}) w_N^{kn} \right), \quad (4)$$

According to Equations (3) and (4), $x_n = x_{n+N/2}$ if $X_k = X_{k+N/2}$. As already mentioned, X_k and $X_{k+N/2}$ with k being odd, are likely to have equal value if they are both sparse data q-QAM symbols or if they are parity q-QAM symbols generated in a relatively close distance. Consequently, if for example data q-QAM symbols are placed in the odd positions of the IFFT input, then up to half ($N/4$) of the odd y_n symbols at the FFT input can be replaced by their counterparts at distance $N/2$: $y_{n+N/2}$. It is obvious that no error would occur only if all the data q-QAM symbols placed the odd positions were equal to X_c , e.g., if the data input is constantly zero. An error occurs if some of the data q-QAM symbols at the odd positions are not trivial (not equal to X_c). To reduce the probability of errors, the number of samples R that can be substituted at the input of the receiver FFT can be lower than $N/4$, e.g., $N/8$ or $N/16$ [8].

Another DFT symmetry property had also been employed in [9] to extend the maximum number of samples that can be replaced from $N/4$ to $3N/8$. It is based on the following properties: $w_N^{n(\frac{N}{2}-k)} = -w_N^{-nk}$, $w_N^{n(\frac{N}{2}+k)} = -w_N^{nk}$ and $w_N^{n(N-k)} = w_N^{-nk}$. The IFFT output x_n can be expressed as:

$$x_n = \left(\sum_{k=1,3,\dots,\frac{N}{4}} (X_k w_N^{kn} - X_{\frac{N}{2}-k} w_N^{-kn}) + \sum_{k=1,3,\dots,\frac{N}{4}} (-X_{k+\frac{N}{2}} w_N^{kn} + X_{N-k} w_N^{-kn}) \right. \\ \left. + \sum_{k=2,\dots,\frac{N}{2}-2} (X_k w_N^{kn} + X_{N-k} w_N^{-kn}) + X_0 - X_{\frac{N}{2}} \right) \frac{1}{N} = x_{\frac{N}{2}-n} \quad (5)$$

From Equation (5) it can be deduced that $x_n = x_{N/2-n}$ ($n < N/4$) if the following three conditions hold: a) $X_k = X_{\frac{N}{2}-k}$ (with odd k and $k \leq N/4$), b) $X_{N-k} = X_{\frac{N}{2}+k}$ (with odd k and $k \leq N/4$) and c) $X_k = X_{N-k}$ with even k and $0 \leq k < N/2$. In a similar way, it can be shown that $x_{N/2+n} = x_{N-n}$ ($n < N/4$). Consequently, the y_n samples with odd $n \leq N/4$ can be substituted by the samples $y_{\frac{N}{2}-n}$ and the $y_{\frac{N}{2}+n}$ samples can be substituted by y_{N-n} .

If 16-QAM modulation is employed, the IFFT input packet structure shown in Figure 1 can be used in order to apply the proposed undersampling technique. The data QAM symbols have been placed at the even positions and most of them have identical value (X_c). If the RSC FEC encoder described earlier is employed then, the repeated 7-bit parity pattern can be padded with one more bit. The Most Significant 4-bits (Parity MSB) and the Least Significant 4-bits (Parity LSB) of the padded 8-bit parity patterns are placed in Figure 1 in appropriate positions in order to apply the proposed undersampling scheme with the lower possible error. In [9], the proposed undersampling method is extended to

is between $-\Delta/2$ and $+\Delta/2$. The error probability is assumed to be uniform ($1/\Delta$) in these limits. The variance of the error is then estimated as: $\sigma_{QE}^2 = \Delta^2/12$. The RE error caused by the use of finite word-length in an N-point FFT is also viewed as QE in [10]. The quantization noise power P_{QE} in all real and imaginary parts of the DFT outputs as defined in Equation (1) is estimated in [10] as:

$$P_{QE} = N \frac{\Delta^2}{12} \quad (7)$$

Figure 2. Undersampling error propagation in a 16-point DIT FFT.

In FFT implementations, this P_{QE} noise is reduced as indicated by the relation (8) below [10]. The upper limit of the relation (8) corresponds to the classic FFT implementation by Cooley and Turkey [20].

$$P_{QE} \leq \frac{\Delta^2}{6} (\log_2 N - 2). \quad (8)$$

Swartzlander and Saleh [19], describe an FFT Implementation with fused floating-point operations and explain the worst-case error caused in Radix-2 and Radix-4 FFT implementations. Errors of $\pm 1/2$ Least Significant Bit (LSB) are caused by rounding and normalization at the output of the adder or the multiplier. Thus, one of the Radix-2 butterfly outputs may have a $1/2$ LSB error, while the other may have a 2 LSB error. For the fused implementation the second error is reduced to 1 LSB. In Radix-4, all of the butterfly outputs may have $2 1/2$ LSB errors. In the fused implementation presented in [21], rounding and normazation error is reduced to $1 1/2$ LSB. Although Radix-2 butterfly error is smaller, the error in an FFT is expected to be smaller for a Radix-4 implementation due to fewer stages and this is confirmed in [21] by simulating 64K point FFT.

The effect of fixed-point format with limited precision for different FFT algorithms is studied in [22]. The error of a single quantization operation is modelled as above and then, the error of a complex multiplication is estimated as $4\sigma_{QE}^2$. A matrix representation of error propagation model is proposed to analyze the rounding effect in DIT and Decimation in Frequency (DIF) FFTs. A similar propagation model will be examined in the next subsection for the FFT of our case that supports undersampling [8,9]. The radix-2 DIT FFT algorithm has better accuracy in term of Signal-to-Quantization-Noise Ratio

(SQNR) [22]. For this reason, we focus on the error modelling of DIT FFT in this paper. The overall output error ΔY of a DIT FFT is estimated in [22] as:

$$\Delta Y =_{d,d} \sum_{i=0}^{d-1} \left(\prod_{j=i+1}^{d-1} w_{T_j} B_{d,j} \right) e_i \quad (9)$$

Where w_{T_j} is the equivalent twiddle factor matrix at the i -th stage of DIT FFT algorithm. $B_{d,d-i}$ is the equivalent butterfly matrix at the i -th stage of 2^d -point FFT and e_i ($0 \leq i \leq d$) is the corresponding additive $N \times 1$ additive noise vector of w_{F_i} (the equivalent twiddle factor matrix at the i -th stage of DIF FFT: $w_{F_i} = w_{T_{d-i-1}}$) with variance σ_c^2 . The total quantization noise power P_{nt} of the DIT FFT algorithm is:

$$P_{nt} = \sum_{i=0}^{d-1} 2^{d-i} n_{T_i} \sigma_c^2 \quad (10)$$

The parameter n_{T_i} is the number of nontrivial twiddle factors at the i -th stage. In [23], the case where the twiddle factor word length is different from the register word length is studied. First, the statistical noise model for the prediction of the RE after a multiplication of two quantized signals u and a , of different precision, is presented: $\hat{v} = \hat{u} * \hat{a} = (u + \varepsilon_1)(a + \varepsilon_2) + \varepsilon_3 = au + noise$. The parameters ε_1 , ε_2 , ε_3 are the errors caused by the rounding of u , a , and v , respectively. The total noise is $u\varepsilon_2 + a\varepsilon_1 + \varepsilon_1\varepsilon_2 + \varepsilon_3$. The variance of noise σ_n^2 given u , a , is:

$$\sigma_n^2 = \frac{2^{-2b_3}}{12} + u^2 \frac{2^{-2b_2}}{12} + a^2 \frac{2^{-2b_1}}{12} + \frac{2^{-2(b_1+b_2)}}{12} = \frac{2^{-2b}}{12} (1 + u^2 + a^2 + 2^{-2b}) \quad (11)$$

The parameters b_1 , b_2 , b_3 are the word lengths of u , a and v , respectively. The last part of the Equation holds if all the word lengths were equal ($b_1 = b_2 = b_3 = b$). The total variance of the RE error in the Radix-2 DIT FFT presented in [23] is simplified to $P_{RE} = 8\sigma_n^2$ for large N values. This Radix-2 variance is compared with Radix-4 and Radix-8 variance (approximately $16\sigma_n^2$ and $24\sigma_n^2$, respectively). The approach presented in [23] will serve as the base for the development of our error model as will be described in the following section.

In [24] fast fixed-point algorithms are used to estimate the RE in the DFT. The RE variance depends on frequency index. Modelling RE as a random variable with uniform distribution holds only for the first FFT stage. RE probability density function is not uniform in the following stages. The discretization error variance at each stage p is estimated recursively as:

$$\sigma_p^2(k) = \frac{1}{4}\sigma_{p-1}^2(k) + \frac{1}{4}\sigma_{p-1}^2(2^{p-1} + k) + \delta_{k,p}, \text{ with } \delta_{k,p} = \begin{cases} 4\sigma_h^2, & \text{if } k = 0 \text{ or } 2^{p-2} \\ 2\sigma_h^2 + 4\sigma_{cs}^2, & \text{otherwise} \end{cases} \quad (12)$$

where $k=0, \dots, 2^{p-1}-1$ and $\sigma_h^2 = 2^{-2b-4}$ if the operations are performed with $b+1$ precision. The parameter σ_{cs}^2 is the variance of the discretization error caused by the multiplication with the cos/sin coefficients. These theoretical models are statistically checked in [24] using DIT FFT with two rounding methods.

In [25], the authors optimize the FFT word length in a memory based architecture attempting to avoid a very pessimistic estimation. A butterfly operation is expressed as $Z = \frac{1}{2}w_N^{kn}(A + B)$ where A and B are the complex inputs. The Z norms are related to the norm of A and B by $\|Z\|^2 \leq (\|A\|^2 + \|B\|^2)/2$. The covariance of the FFT error at stage p of an N -point FFT (with $p = \log_2 N$ stages) is estimated as:

$$\sigma_p^2 = \sum_{k=0}^p \left(\frac{1}{2} \right)^k q(o-k), \text{ with } q(o) = \frac{2^{-2b(p)} + 2^{-2a(p)-2}}{3} > 0 \quad (13)$$

where $a(p)$ and $b(p)$ are the number of bits used to represent the butterfly inputs and sine/cosine coefficients at stage p , respectively. The upper bound of Signal to Noise and Quantization error Ratio

(SNQR) is less than $-10 \log_{10} \sigma_p^2$. The target of the optimization problem is to maximize the vector of q elements subject to $\sigma_p^2 \leq 10^{-SNQR/10}$.

In our approach, the optimization target is to find the minimum number of bits b that can be used as the word length subject to $QE \leq UE \times p_f$ where p_f is a fraction or a multiple of UE depending on the application specifications.

3. Proposed UE, RE Error Model

We focus on a 16-point FFT on the OFDM receiver side that is used as a case study and appears in Figure 2. It is attempted to estimate first, how the error, caused by the undersampling process described in Section 2.1, propagates and calculate the expected error on the FFT outputs. The FFT output error affects the Symbol Error Rate (SER) and the Bit Error Rate (BER) of the OFDM system. However, in this work the FFT output error caused by the undersampling process (P_{UE}) is compared to the QE error (P_{QE}) in order to find an acceptable word length for the FFT.

In the four propagation cases examined, a single error is caused by a QAM symbol $\hat{y}_i$ that does not have the expected trivial value y_i (they differ by ε) in order to generate FFT outputs equal to zero (that can be pruned) or the expected trivial intermediate butterfly outputs. Instead, let us assume that: $\hat{y}_i = y_i + \varepsilon$. The propagation of the error in the first two levels of butterflies (a pair of 2-point and one 4-point FFT) is denoted with the dashed lines in Figure 2. In the first case, the error ε appears at the even input i of the 2-point FFT at the top. The error at the outputs of the 4-point FFT is $e_{\log N-1}^{(i)} = [\varepsilon \ \varepsilon \ \varepsilon \ \varepsilon]$. In the second case that appears from top to the bottom of Figure 2, the non-trivial y_i symbol appears at the even input $i+2$ of the four-point FFT. The error in this case is $e_{\log N-1}^{(i+2)} = [\varepsilon w^0 \ \varepsilon w^{2^{\log N-2}} \ \varepsilon w^0 \ \varepsilon w^{2^{\log N-2}}]$. In the third case the error appears at the odd input $i+1$: $e_{\log N-1}^{(i+1)} = [\varepsilon w^0 - \varepsilon w^0 \ \varepsilon w^0 - \varepsilon w^0]$. In the last case at the bottom of Figure 2, the error at the odd input $i+3$ is $e_{\log N-1}^{(i+3)} = [\varepsilon w^0 w^0 - \varepsilon w^0 w^{2^{\log N-3}} - \varepsilon w^0 w^0 \ \varepsilon w^0 w^{2^{\log N-3}}]$. In the previous expressions of $e_{\log N-1}$ the twiddle factors w_N^j are used without the index N for simplicity. The differences in the four $e_{\log N-1}$ expressions are owed to the different twiddle factors that multiply the error ε as it propagates through different paths. Figure 3 shows how the $e_{\log N-1}$ error of stage $\log N-p$ propagates to the next butterfly stage $\log N-(p-1)$. The arrows in Figure 3 are buses and Figure 3a shows the propagation of the error from the top butterfly input, while Figure 3b shows the propagation of the error from the bottom input. In the first case $e_{\log N-(p-1)} = [e_{\log N-p} \ e_{\log N-p}]$ while in the latter case $e_{\log N-(p-1)} = [w \cdot * e_{\log N-p} \ -w \cdot * e_{\log N-p}]$. The symbol “ $\cdot *$ ” implies multiplication of the corresponding elements of the vectors and w is the vector of all the twiddles of a specific butterfly stage. If multiple errors exist in the FFT inputs, their effect is added at the FFT output. For example, if the errors ε_3 and ε_7 occur in the bit reversed 16-point DIT

FFT inputs y_3 and y_7 (corresponding to Y_{12} and Y_{14} outputs), the individual output errors ($e_0^{y_3}$ and $e_0^{y_7}$, respectively) would be:

$$e_0^{y_3} \begin{bmatrix} w^0 w^0 \\ w^2 w^1 \\ w^4 w^2 \\ w^6 w^3 \\ -w^0 w^4 \\ -w^2 w^5 \\ -w^4 w^6 \\ -w^6 w^7 \\ -w^0 w^0 \\ -w^2 w^1 \\ -w^4 w^2 \\ -w^6 w^3 \\ w^0 w^4 \\ w^2 w^5 \\ w^4 w^6 \\ w^6 w^7 \end{bmatrix} \varepsilon_3, e_0^{y_7} = \begin{bmatrix} w^0 w^0 w^0 \\ w^4 w^2 w^1 \\ -w^0 w^4 w^2 \\ -w^4 w^6 w^3 \\ -w^0 w^0 w^4 \\ -w^4 w^2 w^5 \\ w^0 w^4 w^6 \\ w^4 w^6 w^7 \\ -w^0 w^0 w^0 \\ -w^4 w^2 w^1 \\ w^0 w^4 w^2 \\ w^4 w^6 w^3 \\ w^0 w^0 w^4 \\ w^4 w^2 w^5 \\ -w^0 w^4 w^6 \\ -w^4 w^6 w^7 \end{bmatrix} \varepsilon_7 \quad (14)$$

Figure 3. Error propagation from top (a) or bottom (b) of the butterfly to the next stage of a DIT FFT.

The combined output error if both ε_3 and ε_7 occur is $e_0^{y_3, y_7} = e_0^{y_3} + e_0^{y_7}$. Of course, the same error propagation model holds for the IFFT on the transmitter side. Moreover, the estimation of the error ε is easier on the transmitter side since, the IFFT inputs are QAM symbols with integer values. The 16-QAM constellation shown in Figure 4 is used as a case study, and we assume that $X_c = "1111"$, i.e., the trivial input is '1'. We can see that there are: a) $N_{(2)} = 4$ neighboring QAM symbols ($X_{(2)}$) that differ by 2 in the real or imaginary direction from X_c (the continuous arrows), b) $N_{(2,2)} = 4$ symbols ($X_{(2,2)}$) that differ by 2 in each direction from X_c (dashed arrows with big dashes), c) $N_{(2,4)} = 4$ symbols ($X_{(2,4)}$) that differ by 2 in one direction and by 4 in the other from X_c (dotted arrows), d) $N_{(4)} = 2$ symbols ($X_{(4)}$) that differ by 4 in either the imaginary or real direction from X_c (dashed arrows with small dashes), and e) a single ($N_{(4,4)} = 1$) symbol ($X_{(4,4)}$) that differs from X_c by 4 in each direction. $X_{(2)}$ symbols correspond to 4 input bits with one '0' while $X_{(2,2)}$ and $X_{(4)}$ symbols are derived from 4 input bits with 2 zeros. Finally, the $X_{(2,4)}$ symbols correspond to 4 input bits with 3 zeros and the QAM symbol (-3,-3) is derived by "0000". The corresponding probability of each symbol is $p_c, p_{(2)}, p_{(2,2)}, p_{(4)}, p_{(2,4)}, p_{(4,4)}$. The order of these probabilities is $p_c > p_{(2)} > p_{(2,2)} = p_{(4)} > p_{(2,4)} > p_{(4,4)}$ due to data sparseness.

Figure 4. Distance of 16QAM constellations from X_c .

Since $X_c = (1,1)$, the minimum error in 16-QAM modulation is $\varepsilon_{min} = \sqrt{(1-1)^2 + (1-(-1))^2} = 2 = \varepsilon_{(2)}$. The rest of the errors are: $\varepsilon_{(2,2)} = \sqrt{2*2^2} = 2\sqrt{2}$, $\varepsilon_{(4)} = \sqrt{4^2} = 4$, $\varepsilon_{(2,4)} = \sqrt{2^2 + 4^2} = 2\sqrt{5}$, $\varepsilon_{(4,4)} = \sqrt{2*4^2} = 4\sqrt{2}$. The expected value of ε will be:

$$E[\varepsilon] = \sum_k N_{(k)} p_{(k)} \varepsilon_k \quad (15)$$

For 16-QAM, $E[\varepsilon]=3.37$ if only the symbols that are different than X_c are taken into consideration and they have equal probability. In general, $E[\varepsilon]$ depends on the sparseness level $s < 1$ of the input and the QAM modulation. The sparseness level s means that a fraction s of the input data bits is non-trivial. If all the non-trivial symbols have equal probability to appear, then $p_{(k)} = s/(N-1)$. In this case, Equation (15) can be rewritten as:

$$E[\varepsilon] = \frac{s}{N-1} \sum_k N_k E[\varepsilon_k] = \frac{sE[\varepsilon_k]}{N-1} \sum_k N_k = sE[\varepsilon_k] \quad (16)$$

The effect of the employed modulation scheme to the BER/SER of a telecommunications' system is explained in detail in Appendix C of [12]. The estimation of $E[\varepsilon]$ for different modulation schemes can be assisted by the analysis performed in [12]. There are several alternative options to place the constellation symbols that correspond to a specific number of bits. The lower error is achieved when adjacent constellation symbols differ only in one digit (Gray mapping). In one-dimensional or ring constellations, the Gray mapping can be easily found while square QAM constellations can be Gray encoded hierarchically by examining smaller blocks. There are, however, constellations where there is no perfect Gray mapping. Each constellation X is surrounded by a decision region with minimum distance called Voronoi. When a received symbol Y resides within the Voronoi region of X it is decoded as X on the receiver. The higher the minimum distance of a modulation scheme, the lower the BER/SER that can be achieved. For example, if hard decoding is employed, i.e., if a received QAM symbol is first demodulated to its corresponding bits and then these bits are corrected by the supported FEC method, SER can be expressed as [12]:

$$SER = 1 - \sum_X P(X) \int_{Y \in R_X} p(Y|X) dY \quad (17)$$

where Y is the received symbol, X is the constellations of a specific modulations scheme, $p(Y|X)$ is the probability to get Y at the receiver given that the symbol transmitted is X and R_x is the decision region (Voronoi) of X . In the 16QAM modulation examined earlier, if one of the constellation bits is inverted, the effect on the BER of this error is 1/4 of the effect on the SER. It can be stated that SER represents the worst case effect of the error to the OFDM system. This is also confirmed by various simulations performed in the Appendix C of [12] where several modulations schemes are tested.

In order to estimate the variance of the UE in each one of the FFT outputs we have to define all the possible paths that the error can follow from the FFT input. The paths can be determined in a systematic way starting from a specific output. Let us assume that when a butterfly crossing is reached following the upper branch is denoted by '0' while following the lower branch is denoted by '1'. All the paths can be determined in this way by the combination of $\log_2 N$ bits. For example, in Figure 2, the dashed line that reaches output Y_{12} shows a potential path that the error has followed from input y_6 . The error propagation path in this case can be denoted by "0110" and the initial error ε can be multiplied in each branch by 1, a twiddle factor w or $-w$. Table 1, lists for example, all the potential errors that can occur at each output of an 8-point FFT as well as the expected output error values (in all cases they are 0 except from Y_0 due to orthogonality) and its complex variance. Since the complex variance is the sum of the variances of the real and imaginary parts and the expected values of the error at each output is 0 (except Y_0), the complex variance is actually the sum of the squares of the sine and cosine of the same number (the power of the corresponding twiddle factor) which results in 1. Thus, the complex variance is equal to $1 \times \varepsilon^2$ in all cases but Y_0 . This fact holds for any N -point FFT. If R is the number of samples substituted by the undersampling procedure (e.g., $R=1/16$ means that $N/16$ of the FFT inputs have been substituted by others), the total power of the UE error (P_{UE}) can be estimated as a function of $N, R, s, E[\varepsilon]$:

$$P_{UE} = \sum_{k=0}^{N-1} \sigma_{US}^2(k) = f(N, R, s, E[\varepsilon]) \quad (18)$$

Table 1. Expected UE error in each output of an 8-point DIT FFT.

Path:	000 ($\times \varepsilon$)	001 ($\times \varepsilon$)	010 ($\times \varepsilon$)	011 ($\times \varepsilon$)	100 ($\times \varepsilon$)	101 ($\times \varepsilon$)	110 ($\times \varepsilon$)	111 ($\times \varepsilon$)	Expected Error ($\times E[\varepsilon]$), Variance $\sigma_{US}^2(k) (\times E[\varepsilon]^2)$
Y_0	1	w_8^0	w_8^0	$w_8^0 w_8^0$	w_8^0	$w_8^0 w_8^0$	$w_8^0 w_8^0$	$w_8^0 w_8^0 w_8^0$	1, 0
Y_1	1	w_8^1	w_8^2	$w_8^2 w_8^3$	$-w_8^0$	$-w_8^0 w_8^1$	$-w_8^0 w_8^2$	$-w_8^0 w_8^2 w_8^3$	0, 1
Y_2	1	w_8^2	$-w_8^0$	$-w_8^0 w_8^2$	w_8^0	$w_8^0 w_8^2$	$-w_8^0 w_8^0$	$-w_8^0 w_8^0 w_8^2$	0, 1
Y_3	1	w_8^3	$-w_8^2$	$-w_8^2 w_8^3$	$-w_8^0$	$-w_8^0 w_8^3$	$w_8^0 w_8^2$	$w_8^0 w_8^2 w_8^3$	0, 1
Y_4	1	$-w_8^0$	w_8^0	$-w_8^0 w_8^0$	w_8^0	$-w_8^0 w_8^0$	$w_8^0 w_8^0$	$-w_8^0 w_8^0 w_8^0$	0, 1
Y_5	1	$-w_8^1$	w_8^2	$-w_8^2 w_8^1$	$-w_8^0$	$w_8^0 w_8^1$	$-w_8^0 w_8^2$	$w_8^0 w_8^2 w_8^1$	0, 1
Y_6	1	$-w_8^2$	$-w_8^0$	$w_8^0 w_8^2$	w_8^0	$-w_8^0 w_8^2$	$-w_8^0 w_8^0$	$w_8^0 w_8^0 w_8^2$	0, 1
Y_7	1	$-w_8^3$	$-w_8^2$	$w_8^2 w_8^3$	$-w_8^0$	$w_8^0 w_8^3$	$w_8^0 w_8^2$	$-w_8^0 w_8^2 w_8^3$	0, 1

The selection of an appropriate FFT word length and ADC resolution should target to the restriction of QE and RE errors within a fraction or a multiple p_f (e.g., $P_{QE}, P_{RE} \leq 10\%$ of the P_{UE}) of the UE error estimated in the way described above. More specifically, using $P_{RE} = 8\sigma_n^2$ from Equation (11) and defining $c = 1 + u^2 + a^2$, P_{RE} can be expressed as:

$$P_{RE} = \frac{8}{12} 2^{-2b} (c + 2^{-2b}) \quad (19)$$

If $g = 2^{-2b+1}$, Equation (20) can be expressed as a 2nd degree equation as follows:

$$P_{RE} = \frac{8}{3} \left(c + \frac{g}{2} \right) \Rightarrow g^2 + 2cg - 6P_{RE} = 0 \quad (20)$$

The determinant Det in Equation (20) above is $Det = 4c^2 + 24P_{RE} = 4c^2(1 + 6P_{RE})$. Solving (21) and keeping only the positive square root of the determinant (the negative square root is not applicable) we get:

$$g = \frac{-2c + \sqrt{Det}}{2} = \frac{-2c + 2c \sqrt{1 + \frac{6P_{RE}}{4c^2}}}{2} = c \left(-1 + 1 + \frac{1}{2} \cdot \frac{6P_{RE}}{4c^2} - \frac{1}{8} \cdot \frac{36P_{RE}^2}{16c^4} + \dots \right) \approx \frac{3P_{RE}}{4c} \quad (21)$$

The first two terms of the Taylor series approximation of the square root were preserved in Equation (20). Since the variables α and u in the definition of c change in the various operations of the FFT, it is attempted to estimate only c and not α and u separately. Replacing g in Equation (20), the required word length b can be estimated as follows:

$$2^{-2b+1} = \frac{3P_{RE}}{4c} \Rightarrow 2^b = \sqrt{\frac{8c}{3P_{RE}}} \Rightarrow b = \frac{1}{2} \log_2(8c) - \frac{1}{2} \log_2(3P_{RE}) \quad (22)$$

If $c_1 = \frac{1}{2} \log_2(8c)$, and Equation (18) is written in a general form in order to adjust the weight of the parameters s, N, p_f, R , the word length b can be expressed as:

$$b = c_1 - \frac{1}{2} \log_2 \left(3s^{c_2} p_f^{c_3} R^{c_4} N^{c_5} E[\varepsilon] \right) \quad (23)$$

In order to select appropriate values for the unknown c_i values, the Octave fsolve function for non-linear equations is used with a small number of instances of Equation (23) i.e., with a small number of b, p_f, s, R, N , combinations. The experimental results show that Equation (23) can be used then to accurately estimate the required word length b for other p_f values given a specific OFDM configuration (s, R, N, ε). The physical meaning of the estimated c_i parameters will be explained in Section 5.

4. The Employed FFT Architecture

A DFT requires $O(N^2)$ operations that are reduced to $O(N \cdot \log N)$ if the original FFT architecture is employed [18]. The number of points used by the FFT can be expressed as a product of numbers that are powers of 2. Thus, a 1024-point FFT can be implemented by 10 Radix-2 stages, or 5 Radix-4 stages. If the number of points of the FFT is not a power of 2, then Radix-3 or Radix-5 butterflies can also be employed. For example, a 100-point FFT can be implemented with one stage of Radix-4 and two stages of Radix-5 butterflies [26]. The round-off errors depend on the architecture of the FFT (serial/parallel, Decimation in Time or Frequency, etc.) and the number of stages. An FFT can be implemented either in software if slower operation is acceptable or in hardware for faster response. Modern telecommunication systems require high speed hardware FFTs. Hardware FFTs can either consist of a large number of hardware resources working in parallel or reusable components for more compact, low power implementations with a slightly higher latency overhead.

In this paper, a robust memory-based pipeline FFT has been developed to test the effect of round-off errors in conjunction with the undersampling scheme described in Section 2. It consists of $\log_2 N$ stages (one of them appears in Figure 5). The inputs of stage l are stored in the double buffer l (its size is $2 \times N \times b$ bits). The word length of a butterfly output can be larger by one bit compared to its inputs for optimal resource utilization. However, we use a constant size of b -bits for the inputs/outputs and twiddles of all stages in order to get similar results with the case where a single reusable pipeline stage was used iteratively. One buffer l of the pair is used to store the real and the other for the imaginary part of the FFT inputs/outputs. Buffer l is accessed for write through the buses $w1(l)$ and

$w2(l)$, and for read through the buses $r1(l)$ and $r2(l)$. Each one of these buses consists of a $\log_2 N$ bits, address bus ($ra(l)$ or $wa(l)$) and a pair of b -bits data buses ($Re\{rd(l)\}$ and $Im\{rd(l)\}$, or $Re\{rd(l)\}$ and $Im\{rd(l)\}$). Each data bus carries real numbers in fixed point format with a size of b bits. The inputs of each Radix-2 butterfly are the $rd1(l)$ and $rd2(l)$ while its outputs are $wd1(l)$ and $wd2(l)$. The real and imaginary parts of the twiddle factors w are retrieved from the twiddle Read Only Memory (ROM). The size of the twiddle ROM of stage l is $2 \times N/2^{l+1}$.

Figure 5. A memory-based pipeline DIT FFT stage.

The operations performed at a Butterfly block are:

$$Re\{O_0\} = Re\{I_0\} + Re\{I_1\} \cdot Re\{tw\} - Im\{I_1\} \cdot Im\{tw\} \quad (24)$$

$$Im\{O_0\} = Im\{I_0\} + Re\{I_1\} \cdot Im\{tw\} + Im\{I_1\} \cdot Re\{tw\} \quad (25)$$

$$Re\{O_1\} = Re\{I_0\} - Re\{I_1\} \cdot Re\{tw\} + Im\{I_1\} \cdot Im\{tw\} \quad (26)$$

$$Im\{O_1\} = Im\{I_0\} - Re\{I_1\} \cdot Im\{tw\} - Im\{I_1\} \cdot Re\{tw\} \quad (27)$$

The address buses $ra(l)$ and $wa(l)$ are driven by the Address Generator module that is based on an up counter with $\log_2(N)-1$ resolution. In each stage l the pair of addresses used for the retrieval of the butterfly inputs/outputs ($Addr0$ for I_0 and O_0 , $Addr1$ for I_1 , O_1) and $AddrT$ for the corresponding twiddle factor are the following:

$$AddrT = Cnt \% \frac{N}{2^{l+1}} \quad (28)$$

$$Addr0 = AddrT + \left(\left\lfloor Cnt / \left(\frac{N}{2^{l+1}} \right) \right\rfloor \% 2^l \right) \frac{N}{2^l} \quad (29)$$

$$Addr1 = Addr0 + \frac{N}{2^{l+1}} \quad (30)$$

In Equations (28)–(30), $\%$ is the modulo operator, $\left\lfloor Cnt / \left(\frac{N}{2^{l+1}} \right) \right\rfloor$ is the floor function, and Cnt is the current value of the counter in the Address Generator. The stages of the developed FFT operate in a ping-pong manner. For example, in the first $N/2$ clock cycles the FFT inputs are loaded on the input Buffer at stage $l = \log N - 1$. In the next $N/2$ cycles, the butterfly of stage l is driving its outputs to Buffer $l-1$. Then in the next $N/2$ cycles, the stage $l-1$ is reading inputs from Buffer $l-1$ and driving the outputs to the Buffer $l-2$. At the same time the input Buffer l can be loaded with the next set of FFT inputs. The FFT latency is $\frac{N}{2} \log N$ cycles and the throughput is an FFT output completed every N cycles.

The FFT architecture described in this section can be used to evaluate the complexity of the system in relation with the word length. Focus is given on the main FFT blocks: adders/subtractors and multipliers in the butterflies, counter in the address generator, input/output buffers and twiddle factor ROM. The silicon area or gate count of a ripple carry adder/subtractor is proportional to the word length of the operands. However, if the word length of an b -bit adder with carry look-ahead

is increased by 1 ($b+1$) the required gate count will be increased by more than $1/b$ since the carry look-ahead logic needed to generate the additional carry is more complicated than the logic needed to generate the carry of the least significant bits. The gate count required by a multiplier depends on its architecture. For example, an b -bit Scaling Accumulator Multiplier (SAM) consists of b AND gates, a b -bit adder, a b -bit shift register and the b -bit output register. Thus, the SAM multiplier gate count is proportional to the word length b . The same holds for Serial by Parallel Booth Multipliers i.e., the gate count is proportional to the word length. In these kind of multiplier architectures one of the operands has to be inserted serially bit by bit. Ripple Carry Multipliers (RCM) require all the operand bits in parallel but the area needed is proportional to the square of the word length. Other multiplier architectures like row adder tree and carry-save multipliers require approximately the same gate count as RCM but can achieve faster operation. The storage area needed by the twiddle ROMs and the input/output buffers is proportional to the word length but the area needed by the corresponding address decoders is proportional to the logarithm of the word length. In general, it can be stated that the complexity of the FFT/IFFT modules in an OFDM transceiver is approximately proportional to the employed word length.

The proposed FFT architecture has been described in synthesizable VHDL and has been tested in Modelsim. The description of this module in VHDL is sufficient for the assessment of the effect of the finite word length in the overall error in the OFDM receiver. Implementation on a Field Programmable Gate Array (FPGA) would also be useful to estimate the speed and the power consumption of this module and this will be part of our future work.

5. Simulation Results and Discussion

In this section, the word length estimation based on Equation (23) is evaluated. The combinations examined are the following: 16QAM or Quadrature Phase Shift Keying (QPSK) modulation, $N=256$ or 1024 , $R=1/4$ or $1/16$, $s=0.5\%$, 1% , 2% or 10% . When 16QAM is used $E[\varepsilon]=3.37$ as estimated in Section 3. In a similar way $E[\varepsilon]=2.276$ is estimated for QPSK. The number of bits estimated as the required word length b were between 5 and 10. The total number of configurations simulated are 116. A number of these configurations (four sets including between 5 and 16 non-linear equations) have been used in Octave in order to solve the non-linear Equation (23) for the unknown values of the c_i parameters. Then, the rest of the $L=116$ configurations were tested and the RMSE between the real value of b and the estimated b_{est} for a specific p_f value is extracted as shown in Equation (31). In this way, the minimum number of non-linear equations that have to be solved in order to determine the c_i parameters precisely is found.

$$RMSE = \sqrt{\frac{1}{L} \sum_{j=1}^L (b_{est} - b)^2} \quad (31)$$

The sets of non-linear equations described in Tables 2–5 have been used to estimate the values of the c_i parameters. The average RMSE achieved in the word length estimation of all the 116 configurations is also listed in the 1st row of these tables along with the estimated c_i values for each case. As can be seen from Table 3, determining the c_i values from 12 instances of Equation (23) leads to the lowest RMSE (0.736 for 16QAM and 1.09 for QPSK modulation). A relatively low RMSE is also estimated if 16 equations are used as shown in Table 2. The c_i parameters estimated in Tables 2 and 3 are rounded to $c_1 = -5$, $c_2 = -2$, $c_3 = 2$, $c_4 = 1$, $c_5 = 2$ in order to explain the physical meaning of these values and how they lead to an accurate word length estimation.

Table 2. Using a set of 16 non-linear Equations (23) to determine the c_i parameters.

RMSE: 0.816 (16QAM), 1.132 (QPSK), $c_1 = -5.6315$, $c_2 = -2.0292$, $c_3 = 2.0669$, $c_4 = 1.1994$, $c_5 = -2.3163$						
Equation	Word Length (b)	$p_f = P_{QE}/P_{UE}$	QAM Modulation	FFT Size N	Number of Substituted Samples R	Sparseness s
1	5	0.0046	16QAM	1024	1/4	0.5%
2	8	0.000344	16QAM	1024	1/4	0.5%
3	5	0.046	16QAM	256	1/4	10%
4	8	0.002624	16QAM	256	1/4	10%
5	5	0.12	16QAM	1024	1/16	10%
6	10	0.00773	16QAM	1024	1/16	10%
7	5	0.002625	16QAM	256	1/16	0.5%
8	8	0.00026	16QAM	256	1/16	0.5%
9	5	0.00377	QPSK	1024	1/4	0.5%
10	10	0.001	QPSK	1024	1/4	0.5%
11	5	0.0245	QPSK	256	1/4	10%
12	10	0.000519	QPSK	256	1/4	10%
13	5	0.15	QPSK	1024	1/16	10%
14	10	0.036	QPSK	1024	1/16	10%
15	5	0.0016	QPSK	256	1/16	0.5%
16	10	0.000273	QPSK	256	1/16	0.5%

Table 3. Using a set of 12 non-linear equations (23) to determine the c_i parameters.

RMSE: 0.736 (16QAM), 1.09 (QPSK), $c_1 = -4.3811$, $c_2 = -1.8297$, $c_3 = 1.9560$, $c_4 = 1.2176$, $c_5 = -2.0358$						
Equation	Word Length (b)	$p_f = P_{QE}/P_{UE}$	QAM Modulation	FFT Size N	Number of Substituted Samples R	Sparseness s
1	8	0.000344	16QAM	1024	1/4	0.5%
2	5	0.046	16QAM	256	1/4	10%
3	8	0.002624	16QAM	256	1/4	10%
4	5	0.12	16QAM	1024	1/16	10%
5	5	0.002625	16QAM	256	1/16	0.5%
6	8	0.00026	16QAM	256	1/16	0.5%
7	5	0.00377	QPSK	1024	1/4	0.5%
8	10	0.001	QPSK	1024	1/4	0.5%
9	10	0.000519	QPSK	256	1/4	10%
10	5	0.15	QPSK	1024	1/16	10%
11	10	0.036	QPSK	1024	1/16	10%
12	10	0.000273	QPSK	256	1/16	0.5%

Table 4. Using a set of 8 non-linear equations (23) to determine the c_i parameters.

RMSE: 1.118 (16QAM), 1.62 (QPSK), $c_1 = 4.00823$, $c_2 = -1.29494$, $c_3 = 1.55213$, $c_4 = 1.04125$, $c_5 = -0.16769$						
Equation	Word Length (b)	$p_f = P_{QE}/P_{UE}$	QAM Modulation	FFT Size N	Number of Substituted Samples R	Sparseness s
1	8	0.000344	16QAM	1024	1/4	0.5%
2	5	0.046	16QAM	256	1/4	10%
3	5	0.12	16QAM	1024	1/16	10%
4	8	0.00026	16QAM	256	1/16	0.5%
5	5	0.00377	QPSK	1024	1/4	0.5%
6	10	0.000519	QPSK	256	1/4	10%
7	5	0.15	QPSK	1024	1/16	10%
8	10	0.000273	QPSK	256	1/16	0.5%

Table 5. Using a set of 5 non-linear Equation (23) to determine the c_i parameters.

RMSE: 3.694 (16QAM), 2.622 (QPSK), $c_1 = -33.2488$, $c_2 = -3.6799$, $c_3 = 4.8570$, $c_4 = 4.9535$, $c_5 = -5.9386$						
Equation	Word Length (b)	$p_f = P_{QE}/P_{UE}$	QAM Modulation	FFT Size N	Number of Substituted Samples R	Sparseness s
1	8	0.002624	16QAM	256	1/4	10%
2	5	0.002625	16QAM	256	1/16	0.5%
3	5	0.00377	QPSK	1024	1/4	0.5%
4	5	0.15	QPSK	1024	1/16	10%
5	10	0.036	QPSK	1024	1/16	10%

Using the specific c_i values, Equation (23) can be written as:

$$b = c_1 - \frac{1}{2} \log_2 \left(3s^{-2} p_f^2 R^1 N^{-2} E[\varepsilon] \right) = \frac{1}{2} \log_2(8c) - \frac{1}{2} \log_2(3RE[\varepsilon]) - \frac{1}{2} \log_2(s^{-2} p_f^2) + \log_2 N \quad (32)$$

From $c_1 = \frac{1}{2} \log_2(8c) = \frac{1}{2} \log_2(8(1 + u^2 + a^2)) = -5$ we get that $1 + u^2 + a^2 = 0.000122$ or $u^2 + a^2 = -0.99988$ which is impossible unless u and a are assumed complex numbers. When the set of equations listed in Table 4 is used $c_1 = +4$ and $u^2 + a^2 = 31$ which is more consistent with the model presented in [22] and Equation (11). However, the initial definition of c_1 will be ignored in an attempt to define the overall error model that matches the experimental results more accurately. In this perspective, the rest of the terms in the right side of Equation (23) are interpreted as follows: $\frac{1}{2} \log_2 3RE[\varepsilon]$ takes its minimum value (-0.185) for the experiments conducted in this paper when $R=1/16$ and $E[\varepsilon]=2.276$ with QPSK modulation and its maximum value (0.2) with $R=1/4$ and $E[\varepsilon]=3.37$ when 16QAM modulation is employed. The term $s^{-2} p_f^2$ is close to a constant since p_f is proportional to the sparse level s : if only non-sparse FFT inputs were present, there would be errors in all the FFT outputs and $p_{ns} = P_{QE}/P_{UE}$. The higher value measured for p_{ns} is 0.3. If the input is sparse, the ratio p_f of the quantization error to the undersampling error is proportional to the sparseness level s : $p_f \propto p_{ns}s$. When the input is too sparse, the UE and QE errors are both low. When the input is less sparse (s value is higher), UE raises but the raise of QE is even higher. This is owed to the fact that although UE gets worse, there may be still FFT outputs unaffected by the undersampling process if some samples are replaced by the others with identical value. However, if s is higher, more operations with numbers that are not zero will be performed and the QE will increase respectively since all the results of these non-trivial operations will have QE error. In this sense, s^2 counterbalances p_f^2 and the

maximum value for the 3rd term of Equation (23) will be $\frac{1}{2} \log_2 s^{-2} p^2 = \frac{1}{2} \log_2 p_{ns} = \frac{1}{2} \log_2 0.3 = -0.87$. If p_{ns} is lower, a higher positive offset in Equation (23) occurs.

The last term of Equation (23) can have two values in the results presented in this paper: either $\log_2 1024 = 10$ or $\log_2 256 = 8$. This is the larger positive offset that counterbalances the negative value of the constant value c_1 . The specific c_i parameters have been approximated for these two FFT sizes. Should different FFT sizes be covered, the set of nonlinear equations that have to be used for the approximation of c_i parameters must also include configurations with these FFT sizes. If we try to use the approximated c_i parameters of Table 3 for the case of a 64-point FFT size, $c_1 + \log_2 N$ would be 0. The term $-\frac{1}{2} \log_2 (3 \cdot R \cdot E[\varepsilon])$ results in small signed offsets between -0.185 and 0.2 as explained above and thus, the word length would be actually determined by the factor $-\frac{1}{2} \log_2 s^{-2} p_f^2 = -\frac{1}{2} \log_2 p_{ns}$. In order to get a realistic estimation of at least 5 bits as a word length, p_{ns} should be 10^{-3} , or, in other words, UE error should be 1000 times larger than QE error. Such a relation between UE and QE errors is not always guaranteed.

The estimated and expected word lengths for all the 16QAM and QPSK OFDM configurations tested when the c_i parameters listed in Table 3 are used, are compared in Figures 6 and 7, respectively. In these figures, the required minimum ADC resolution is also included. This ADC resolution b_{ADC} has been estimated in Equation (33) that has been derived from Equation (8), the definition $\Delta = V_{ref}/2^{b_{ADC}}$ and the specification that P_{QE} should be equal to P_{RE} . V_{ref} was selected equal to 1V but approximately the same results would have been achieved if a different voltage reference had been selected, such as 3V. The ADC resolution b_{ADC} should match the FFT word length b thus, b_{ADC} should be selected equal to b since $b_{ADC} < b$ in all cases as shown in Figures 6 and 7.

$$b_{ADC} = \frac{1}{2} \log_2 \frac{\log_2 N - 2}{6P_{QE}} \quad (33)$$

Figure 6. Estimated, expected word length and minimum ADC resolution for all the 16QAM OFDM configurations tested.

Figure 7. Estimated, expected word length and minimum ADC resolution for all the QPSK OFDM configurations tested.

The procedure followed in this paper to define a model that optimizes the word length subject to error restrictions and a predetermined relation between undersampling error and quantization/rounding error can be followed in other non-OFDM telecommunication systems. For example, in optical networks, a model can be created that selects the appropriate modulation (number of bits/symbol: R_s) in order to achieve a desired capacity, given a specific power budget P_o . Based on the analysis presented in [12], the capacity C_o can be expressed as:

$$C_o = R_s \log_2 \left(1 + \frac{P_o}{N_o R_s} \right) \quad (34)$$

The noise term $N_o R_s$ can be expressed as a weighted combination of the various noise sources in an optical channel [12]: the beat noise $\sigma_{beat}^2 = 4S_b^2 N_o P_{LO} B_e$, the shot noise $\sigma_{shot}^2 = 2eS_b P_{LO} B_e$ and the thermal/electronic noise σ_{elec}^2 . Concerning the constants (we assume that their values are known) used in these noise variance expressions, S_b is the photodetector responsivity, N_o is the noise spectral density, P_{LO} the optical power, B_e the power equivalent bandwidth of the entire receiver and e the elementary charge. The most important optical noise is Amplified Spontaneous Emission (ASE) which actually describes the attenuation of the optical signal by a factor $a_{ASE} = 0.2 \text{ dB/Km}$. The distortion posed by the required N_A repeater/amplifiers placed at distance L_a can be described by the noise spectral density $N_{ASE}^{EDRA} = N_A (e^{a_{ASE} L_a} - 1) h \nu_s n_{sp}$ or $N_{ASE}^{IDRA} = a_{ASE} L_a h \nu_s K_T$ if Erbium-doped fiber amplifiers (EDFAs) or Ideal Distributed Raman Amplification (IDRA) is used, respectively. The parameters used in these noise spectral densities are also assumed to have known values: h is the Plank constant, ν_s is the optical frequency, K_T is the photon occupancy factor and $n_{sp} < 1$ the spontaneous emission factor. The model can be trained by a number of non-linear equations that combine the channel error sources with various capacity and power requirements for specific predefined modulations schemes. The target of this training would be to estimate the weights of the channel error sources. After updating the error model with these weights, it can be used to select an appropriate modulation scheme for different capacity and power specifications or channel conditions.

6. Conclusions

An error model has been developed for an OFDM transceiver architecture that supports undersampling when sparse information is exchanged. The error model combines undersampling and

round-off noise and determines the word length that should be employed by the FFT/IFFT modules. The desired round-off error can be defined as a fraction or a multiple of the undersampling error and an appropriate word length is estimated to achieve this goal. A new FFT pipeline module has been developed in synthesizable VHDL in order to evaluate the correctness of the predicted word length. Simulation for all the combinations of QPSK or 16QAM modulation, FFT sizes of $N=256$ or $N=1024$ points, sparseness levels 0.5%, 1%, 2% and 10%, and FFT input replacement of either $N/4$ or $N/16$ samples have been tested. The simulation results show that the appropriate word length can be determined with RMSE less than 1 and the ADC resolution should match the estimated word length.

Future work will focus on extending the developed error model for pipeline FFTs with different word length in each stage. A different range of FFT sizes and QAM modulations will also be tested. Finally, the FFT/IFFT as well as other OFDM modules will be implemented in real hardware (FPGA) to measure their power consumption and speed.

7. Patents

The undersampling method described in Section 2 is protected by the patents 1008130 and 1008564, Greek Patent Office, published November 2014 and September 2015 respectively.

Author Contributions: Conceptualization, N.P.; methodology, N.P.; Software, N.P.; validation, N.P.; writing—original draft preparation, N.P.; writing—review and editing, N.P.; investigation, N.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Candès, E.J.; Wakin, M.B. An Introduction to Compressive Sampling. *IEEE Signal. Process. Mag.* **2008**, *25*, 25–30. [CrossRef]
2. Kirolos, S.; Ragheb, T.; Laska, J.; Duarte, M.F.; Massoud, Y.; Baraniuk, R.G. Practical issues in implementing analog-to-information converters. In Proceedings of the 2006 6th International Workshop on System on Chip for Real Time Applications, Cairo, Egypt, 27–29 December 2006; pp. 141–146. Available online: http://www.ece.rice.edu/~duarte/images/A21_IWSOC.pdf (accessed on 22 March 2020).
3. Donoho, D.L. Compressed Sensing. *IEEE Trans. Inf. Theory* **2006**, *52*, 1289–1306. [CrossRef]
4. Lust, M.; Donoho, D.; Santos, J.; Pauly, J. Compressed Sensing MRI. *IEEE Signal. Process. Mag.* **2008**, *25*, 72–82. [CrossRef]
5. Xu, L.; Liang, Q. Compressive Sensing in Radar Sensor Networks Using Pulse Compression Waveforms. In Proceedings of the 2012 IEEE International Conference on Communications (ICC), Ottawa, ON, Canada, 10–15 June 2012. [CrossRef]
6. Cheng, P.; Gui, L.; Tao, M.; Guo, Y.; Huang, X.; Rui, Y. Sparse Channel Estimation for OFDM Transmission over Two-Way Relay Networks. In Proceedings of the 2012 IEEE International Conference on Communications (ICC), Ottawa, ON, Canada, 10–15 June 2012. [CrossRef]
7. Pawar, S.; Ramchandran, K. Computing a k -sparse n -length Discrete Fourier Transform using at most $4k$ samples and $O(k \log k)$ complexity. In Proceedings of the 2013 IEEE International Symposium on Information Theory, Istanbul, Turkey, 7–12 July 2013. [CrossRef]
8. Petrellis, N. Under-sampling in OFDM Telecommunication Systems. *Mdpi Appl. Sci.* **2014**, *4*, 79–98. [CrossRef]
9. Petrellis, N. Optimal Reconstruction of Sub-sampled Time-Domain Sparse Signals in Wired/Wireless OFDM Transceivers. *Eurasip J. Wirel. Commun. Netw.* **2016**, *122*. [CrossRef]
10. Widrow, B.; Kollár, I. *Quantization Noise: Roundoff Error in Digital Computation, Signal. Processing, Control., and Communications*; Cambridge University Press: Cambridge, UK, 2008.
11. *IEEE Standard for Floating Point Arithmetic*; ANSI/IEEE Standard 754/2008; IEEE: New York, NY, USA, 2008.
12. Essiambre, R.J.; Kramer, G.; Winzer, P.; Foschini, G.; Goebel, B. Capacity limits of optical fiber networks. *IEEE Journal of Lightwave Technology* **2010**, *28*, 662–701. [CrossRef]

13. Semrau, T.; Xu, T.; Shevchenko, N.; Paskov, M.; Alvarado, A.; Killey, R.; Bayvel, P. Achievable information rates estimates in optically amplified transmission systems using nonlinearity compensation and probabilistic shaping. *Osa Opt. Lett.* **2017**, *42*, 121–124. [[CrossRef](#)] [[PubMed](#)]
14. Ayhan, T.; Dehaene, W.; Verhelst, M. A 128:2048/1536 point FFT hardware implementation with output pruning. In Proceedings of the IEEE 22nd European Signal Processing Conference (EUSIPCO), Lisbon, Portugal, 1–5 September 2014.
15. Welch, P.D. A fixed-point fast Fourier transform error analysis. *IEEE Trans. Audio Electroacoust.* **1969**, *17*, 151–157. [[CrossRef](#)]
16. Pálfi, V.; Kollár, I. Roundoff Errors in Fixed-Point FFT. In Proceedings of the IEEE International Symposium on Intelligent Signal Processing, Budapest, Hungary, 26–28 August 2009.
17. Horvath, G. Coefficient Quantization Error in FFT-based Spectrum Analysis. In Proceedings of the IFAC 10th Triennial World Congress, Munich, Germany, 27–31 July 1987.
18. Oppenheim, A.V.; Weinstein, C.J. Effects of finite register length in digital filtering and the fast Fourier Transform. *Proc. IEEE* **1972**, *60*, 957–976. [[CrossRef](#)]
19. Knight, W.R.; Kaiser, R. A simple fixed point error bound for the fast Fourier transform. *IEEE Trans. Acoust. Speech Sign. Process.* **1979**, *27*, 615–620. [[CrossRef](#)]
20. Cooley, J.W.; Tukey, J.W. An Algorithm for the Machine Calculation of Complex Fourier Series. *Math. Comput.* **1965**, *19*, 297–301. [[CrossRef](#)]
21. Swartzlander, E., Jr.; Saleh, H.H.M. FFT Implementation with Fused Floating-Point Operations. *IEEE Trans. Comput.* **2012**, *61*, 284–288. [[CrossRef](#)]
22. Chang, W.H.; Nguyen, T.Q. On the Fixed-Point Accuracy Analysis of FFT Algorithms. *IEEE Trans. Signal. Process.* **2008**, *56*, 4673–4682. [[CrossRef](#)]
23. Qadeer, S.; Ali Khan, Z.M.Z.; Abdul Sattar, S. On Fixed Point error analysis of FFT algorithm. In Proceedings of the Colloquiums on Computer Electronics Electrical Mechanical and Civil, Kerala, India, 20–21 September 2011.
24. Dakovic, M.; Stankovic, L.; Lutovac, B.; Stankovic, I. On the Fixed-point Rounding in the DFT. In Proceedings of the IEEE EUROCON 2017, Ohrid, Northern Macedonia, 6–8 July 2017.
25. Wei, C.-J.; Liu, S.-M.; Chen, S.-J.; Hu, Y.-H. Optimal fixed-point fast fourier transform. In Proceedings of the IEEE Workshop on Signal Processing Systems (SiPS), Taipei, Taiwan, 16–18 October 2013; pp. 377–382.
26. Löfgren, J.; Nilsson, P. On hardware implementation of radix 3 and radix 5 FFT kernels for LTE systems. In Proceedings of the IEEE NORCHIP Conference, Lund, Sweden, 14–15 November 2011.

© 2020 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).