

FFT-Based Simultaneous Calculations of Very Long Signal Multi-Resolution Spectra for Ultra-Wideband Digital Radio Frequency Receiver and Other Digital Sensor Applications

Chen Wu *  and Michael Low

Defence Research and Development Canada—Ottawa Research Centre, Ottawa, ON K1A 0Z4, Canada; michael.low@forces.gc.ca

* Correspondence: chen.wu@forces.gc.ca

Abstract: The discrete Fourier transform (DFT) is the most commonly used signal processing method in modern digital sensor design for signal study and analysis. It is often implemented in hardware, such as a field programmable gate array (FPGA), using the fast Fourier transform (FFT) algorithm. The frequency resolution (i.e., frequency bin size) is determined by the number of time samples used in the DFT, when the digital sensor's bandwidth is fixed. One can vary the sensitivity of a radio frequency receiver by changing the number of time samples used in the DFT. As the number of samples increases, the frequency bin width decreases, and the digital receiver sensitivity increases. In some applications, it is useful to compute an ensemble of FFT lengths; e.g., 2^{P-j} for $j = 0, 1, 2, \dots, J$, where j is defined as the spectrum level with frequency resolution $2^j \cdot \Delta f$. Here Δf is the frequency resolution at $j = 0$. However, calculating all of these spectra one by one using the conventional FFT method would be prohibitively time-consuming, even on a modern FPGA. This is especially true for large values of P ; e.g., $P \geq 20$. The goal of this communication is to introduce a new method that can produce multi-resolution spectrum lines corresponding to sample lengths 2^{P-j} for all $J + 1$ levels, concurrently, while one long 2^P -length FFT is being calculated. That is, the lower resolution spectra are generated naturally as by-products during the computation of the 2^P -length FFT, so there is no need to perform additional calculations in order to obtain them.

Keywords: digital receiver; discrete Fourier transform; fast Fourier transform; multi-resolution spectrum; blocking fast Fourier transform; signal processing; field programmable gate array



Citation: Wu, C.; Low, M. FFT-Based Simultaneous Calculations of Very Long Signal Multi-Resolution Spectra for Ultra-Wideband Digital Radio Frequency Receiver and Other Digital Sensor Applications. *Sensors* **2024**, *24*, 1207. <https://doi.org/10.3390/s24041207>

Academic Editor: Nicola Lusardi

Received: 8 January 2024

Revised: 4 February 2024

Accepted: 9 February 2024

Published: 13 February 2024



Copyright: © 2024 by the authors. 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 (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Multi-resolution frequency spectrum analysis is widely used in many areas to process signals from different sources, such as signals from mechanical structures, acoustic sensors, human body signal sensors, and digital RF/microwave receivers. A few examples are given in [1–4]. Multi-resolution analysis (MRA) is a powerful tool used in image and signal processing. In wavelet analysis [5], a mother wavelet function [6] is scaled in frequency and shifted in time to analyze images or signals at different resolutions [6,7]. Recently, Ref. [8] gives a comprehensive review of the developments, challenges, and opportunities of wavelet analysis, and also covers the evolution of wavelet neural networks. In parallel with the developments in wavelet-based MRA, Fourier transform (FT)-based approaches are also being developed. This framework includes the fractional FT [9–11], and the relatively recently developed quadratic-phase FT [12,13], which offers a unified approach to processing transient and non-transient signals and is very useful for analyzing signals with time-varying properties. Although many FT-based MRA methods have been developed, in modern digital receiver development the conventional FT is still widely used. In general, for a fixed digitizer sampling rate, calculating DFTs having different numbers of samples results in spectra having different frequency resolutions. In the FT-based MRA, the short-time Fourier transform (STFT) is often used to calculate signal time-frequency

distributions. There are a number STFT-based methods, such as the multi-resolution STFT [14], adaptive STFT [15,16], and STFT frequency-domain (STFT-FD) [17] methods. The idea of the multi-resolution STFT is based on using different window sizes to achieve different frequency resolutions. It uses bandpass filters to divide the signal into frequency bands. Therefore, multi-resolution STFT relies on the use of filters to obtain the multi-resolution spectra. The adaptive STFT adjusts the window size for each time instant based on the local signal characteristics. The STFT-FD computes a transform using the basic idea of the STFT; however, it uses a fixed window size in the frequency domain. This approach is similar to that of the multi-resolution STFT and wavelet methods, but with a different methodology. The key of the STFT-FD is to define the window size as the number of cycles of the frequency component being considered rather than defining it in time. The STFT-FD needs neither filtering nor the evaluation of local signal characteristics as in the multi-resolution STFT and the adaptive STFT, respectively.

It is well known that any STFT-based time-to-frequency transform has a limitation. This is the inherent trade-off between time and frequency resolution, i.e., the short-time window has good time resolution, but poor frequency resolution, and vice versa. In [18], the piecewise-constant window blocking fast Fourier transform (PCW-BFFT) method was introduced for ultra-wideband digital receivers in order to achieve a fine frequency resolution and a high receiver sensitivity. By applying a number of window functions with different widths in the time domain, the PCW-BFFT method can simultaneously produce multi-time-resolution spectrograms with a very fine frequency resolution for a very long-time sequence (e.g., $N = 2^{20}$); however, this method cannot produce multi-frequency-resolution spectra concurrently.

The DFT-based time-to-frequency transform is the most widely used signal processing method in ultra-wideband digital sensors and is commonly implemented using the fast Fourier transform (FFT) on field programmable gate arrays (FPGAs) in real-time applications. Ref. [19] presents several methods using the FFT to detect and analyze intercepted signals with multi-resolution frequency spectral lines (MR-FSL). However, up to now, to obtain the MR-FSL of a signal, multiple FFT calculations with different signal lengths (e.g., $N = 2^{P-j}$ where $j = 0, 1, 2, \dots, J$) must be performed one by one. The disadvantages of the traditional FFT-based methods are: (1) for large signal length (e.g., $P = 20$ and $j = 0, 1, 2, 3$), four different long-length FFTs have to be calculated; (2) one has to wait for a large number of time samples to be generated by the analog-to-digital converter (ADC) in the receiver before the FFTs can be performed; (3) many FPGA resources are needed to store those time samples before the FFTs can be performed; and (4) more importantly, the computational effort spent on the calculation of the longest FFT cannot be reused in the calculations of any of the shorter length FFTs.

To mitigate the problem of requiring a large amount of hardware resources to compute very long length FFTs and effectively calculate the frequency spectrum of long-sequence signals in an embedded device in real time, the blocking FFT (BFFT) method was first introduced in [20]. It also demonstrated the implementation of the method on Xilinx's ZYNQ-7000 device. In order to perform an N -point frequency bin calculation, the BFFT conceptually divides N time samples into K consecutive timeslot blocks consisting of M samples each. As each block of $M (= N/K)$ samples is generated by the ADC, the BFFT method performs K independent M -point FFTs, where suitable phase rotation factors are applied to the M input samples of each of the K FFTs. Each of these FFTs can be computed in parallel (e.g., on an FPGA or GPU). The resulting $K \times M = N$ frequency data are accumulated with the results from previous timeslots to form an N -bin spectrum line. Each spectrum line can be viewed as a group of K equal-length frequency blocks where the k^{th} frequency block has frequency bins f_{kq} , where $k = 0, 1, 2, \dots, K-1$, and $q = k, M+k, \dots, (K-1)M+k$. The final N -bin spectrum line is calculated as the accumulation of the frequency results from processing all K timeslots. The merit of the BFFT method is that instead of having to first store a large number of N time samples before being able to compute a single, long N -point FFT, which may be a challenge even

on today's high-end FPGAs, the BFFT performs K many M -point FFTs in parallel, as soon as M samples are available from the current timeslot. This tremendously reduces the hardware resource requirements, produces useful intermediate results with low latency, and also enables the development of other useful methods for ultra-wideband digital receiver designs, such as the multi-time-resolution piecewise-constant window blocking Fourier transform (PCW-BFFT) method [18], the accumulatively increasing receiver sensitivity (AIRS) method [21], and the multi-frequency-resolution spectrum line calculation method introduced in this paper.

Taking advantage of the aforementioned BFFT process, this paper for the first time introduces an FFT-based MR-FSL calculation method, called the multi-frequency-resolution BFFT (MFR-BFFT). It can calculate the spectrum line having $N_0 = 2^P$ frequency bins, while simultaneously producing all the spectrum lines having $N_j = 2^{P-j}$ frequency bins, where $j = 0, 1, 2, \dots, J$, without needing to perform extra FFT calculations on the M samples from the current timeslot. That is, the computational efforts to obtain the 2^P -bin spectrum line are reused to obtain the spectrum lines that have $2^{P-1}, 2^{P-2}, 2^{P-3}, \dots, 2^{P-J}$ frequency bins at other frequency resolution levels. With a fixed ADC sampling rate (i.e., $f_s = \frac{1}{t_s}$ where t_s is the time interval between samples), the MFR-BFFT uses 2^{P-j} time samples (where $j = 0, 1, 2, \dots, J$) to compute spectrum lines up to $J = \log_2(K_0)$ levels with 2^j different spectrum lines at Level- j . Here, K_0 is the number of timeslots in Level-0. Since the ADC rate is fixed, the frequency resolutions at different spectrum levels are $2^j \cdot \Delta f$, where $\Delta f = f_s/2^P$ is the Level-0 frequency resolution. The signal duration used to get different level spectrum lines is $N_j \times t_s$. More details will be presented in the next section.

Compared with other aforementioned existing MRA methods, although the MFR-BFFT method does not have the flexibility to obtain different levels of resolutions as wavelet-based MRA methods do, it can tremendously reduce the computational effort on hardware to produce spectra with a fixed set of frequency resolutions. Compared with other FT-based MRA methods, the MFR-BFFT has the advantage of being able to process very long-length signal samples, and to produce a set of different frequency resolution spectra simultaneously. Note that the very long-length signal samples do not necessarily mean a long signal in the time domain. It can be a long signal sampled at a low ADC sampling rate, or a very short signal sampled at high sampling rate. For example, if an ADC samples at 2.5 GSample/s, within about 0.42 ms, it can produce 2^{20} time samples.

The rest of the paper is organized as follows: The derivations and discussions of the MFR-BFFT methods are presented in the next section. Section 3 shows the MFR-BFFT results using data obtained from an ultra-wideband digital radio receiver and compares the results with those obtained from the “fft” function of Mathworks MATLAB (R2023b) to prove the correctness of the MFR-BFFT implementation in MATLAB. The conclusion can be found in the last section. The hardware implementation including an expansion of the comparative analysis between MFR-BFFT and conventional FFT, such as computational complexity, power consumption, and execution time, etc., will be reported in another full paper.

2. FFT-Based Multi-Frequency-Resolution Spectrum Line Calculation

Based on the discrete Fourier transform (DFT), the frequency spectrum line at Level-0 can be expressed as:

$$X_{01}(v_0) = \sum_{n=0}^{N_0-1} x(n) W_{N_0}^{nv_0} \quad (1)$$

where $v_0 = 0, 1, 2, \dots, N_0 - 1$ are the frequency bin centers on the Level-0 spectrum line, $W_{N_0}^{nv_0} = e^{-\frac{2\sqrt{-1}\pi n v_0}{N_0}}$, and x is a complex-valued signal. X_{01} means the first spectrum line at Level-0, although at this level there is only one spectrum line. Using the BFFT, N_0 time

samples are divided into K_0 equal timeslots and each timeslot has M samples; therefore, (1) can be written as:

$$X_{01}(v_0) = \sum_{k=0}^{K_0-1} \left\{ \sum_{n=kM}^{(k+1)M-1} x(n) W_{N_0}^{nv_0} \right\} \quad (2)$$

By substituting $u = n - kM$, then for the k^{th} timeslot of M samples, the portion inside the curly braces can be rewritten as:

$$\sum_{n=kM}^{(k+1)M-1} x(n) W_{N_0}^{nv_0} = \sum_{u=0}^{M-1} x(u + kM) W_{N_0}^{(u+kM)v_0} \quad (3)$$

We now define:

$$Y_{01_k}(v_0) = \sum_{u=0}^{M-1} x(u + kM) W_{N_0}^{(u+kM)v_0} \quad (4)$$

which is the frequency information obtained from the k^{th} timeslot of M samples for the Level-0 spectrum line. Using (4), (2) can be rewritten as:

$$X_{01}(v_0) = \sum_{k=0}^{K_0-1} Y_{01_k}(v_0) \quad (5)$$

In order for the M samples in the k^{th} timeslot to contribute to the overall N_0 bins, the BFFT also divides N_0 frequency bins into K_0 equal-length frequency blocks, where each frequency block also has M frequency bins. By substituting $v_0 = m_0 + rK_0$, where $m_0 = 0, 1, 2, \dots, K_0 - 1$ to define the total K_0 frequency blocks at Level-0, and $r = 0, 1, 2, \dots, M - 1$ to specify the M frequency bins in each frequency block, then (4) becomes:

$$Y_{01_k}(m_0 + rK_0) = \sum_{u=0}^{M-1} x(u + kM) W_{N_0}^{m_0(u+kM)} W_M^{ru} \quad (6)$$

where $MK_0 = N_0$ and $W_{N_0}^{kMK_0r} = 1$ are used. Hence, using the BFFT to calculate (1), we have:

$$X_{01}(m_0 + rK_0) = \sum_{k=0}^{K_0-1} \left\{ \sum_{u=0}^{M-1} x(u + kM) W_{N_0}^{m_0(u+kM)} W_M^{ru} \right\} \quad (7)$$

It can be seen that to contribute to all the frequency bins in X_{01} using M samples in the k^{th} timeslot, the samples need to be multiplied by phase rotation factors $W_{N_0}^{m_0(u+kM)}$, which are determined by the frequency block (m_0) and time index ($n = u + kM$) in the overall time sequence, before performing the M -point FFT (the operations specified in curly braces) indicated in (7). Since $m_0 = 0, 1, 2, \dots, K_0 - 1$, a total of K_0 many M -point FFTs are computed at Level-0.

Now, let us consider the Level-1 spectrum line calculations with the same time sequence $x(n)$ where ($n = 0, 1, 2, \dots, N_0 - 1$). Using the DFT, the spectrum lines can be computed as:

$$X_{1i}(v_1) = \sum_{n=(i-1)N_1}^{iN_1-1} x(n) W_{N_1}^{nv_1} \quad (8)$$

where $i = 1, 2$ indicates that there are the two spectrum lines in Level-1. The first is calculated from samples 0 to $N_1 - 1$, and the second is calculated from samples N_1 to $N_0 - 1$. Here $N_1 = N_0/2$ and $W_{N_1}^{nv_1} = e^{-\frac{2\sqrt{-1}\pi nv_1}{N_1}}$. Note that (1) these two spectrum lines have the same frequency bins $v_1 = 0, 1, 2, \dots, N_1 - 1$; (2) they only use half of the available samples in $x(n)$ compared with what was used in Level-0; and (3) the frequency bin size at Level-1 is double that of Level-0, since the ADC sampling rate is fixed.

Again, if we use the BFFT to calculate the Level-1 spectrum lines with the same number of samples (M) in a timeslot that is used at Level-0, we have:

$$X_{1i}(v_1) = \sum_{k=(i-1)K_1}^{iK_1-1} \left\{ \sum_{n=kM}^{(k+1)M-1} x(n) W_{N_1}^{nv_1} \right\} \quad (9)$$

where the first and second $K_1 (= K_0/2)$ groups of timeslots are used to calculate $X_{11}(v_1)$ and $X_{12}(v_1)$, respectively. By substituting $u = n - kM$, then for the k^{th} timeslot's M samples, we have:

$$\sum_{n=kM}^{(k+1)M-1} x(n) W_{N_1}^{nv_1} = \sum_{u=0}^{M-1} x(u + kM) W_{N_1}^{(u+kM)v_1} \quad (10)$$

In order for the M samples in k^{th} timeslot to contribute to the overall N_1 bins, the BFFT also divides the N_1 frequency bins into K_1 blocks, each of which contains M frequency bins. By substituting $v_1 = m_1 + rK_1$, where $m_1 = 0, 1, 2, \dots, K_1 - 1$ to define K_1 frequency blocks in each spectrum line, and $r = 0, 1, 2, \dots, M - 1$ (i.e., there are M frequency bins in each frequency block), then from (10) we have:

$$Y_{1k}(m_1 + rK_1) = \sum_{u=0}^{M-1} x(u + kM) W_{N_1}^{m_1(u+kM)} W_M^{ru} \quad (11)$$

where $MK_1 = N_1$ and $W_{N_1}^{kMK_1r} = 1$ are used. Then, using the BFFT, the spectrum lines at Level-1 can be expressed as:

$$X_{1i}(m_1 + rK_1) = \sum_{k=(i-1)K_1}^{iK_1-1} \left\{ \sum_{u=0}^{M-1} x(u + kM) W_{N_1}^{m_1(u+kM)} W_M^{ru} \right\} \quad (12)$$

In general, using the BFFT, the spectrum lines at Level- j can be expressed as:

$$X_{ji}(m_j + rK_j) = \sum_{k=(i-1)K_j}^{iK_j-1} \left\{ \sum_{u=0}^{M-1} x(u + kM) W_{N_j}^{m_j(u+kM)} W_M^{ru} \right\} \quad (13)$$

where $j = 0, 1, 2, \dots, \log_2(K_0)$ defines the spectrum line levels; $i = 1, 2, 3, \dots, 2^j$ denotes the spectrum lines in Level- j ; and the i^{th} spectrum line is obtained using timeslots from $(i-1)K_j$ to $iK_j - 1$, where K_j specifies the number of timeslots in Level- j .

From the above discussions we can observe that:

1. with fixed N_0 samples used in Level-0, the maximum number of levels is limited by the number of samples (M) in a timeslot to $J = \log_2(K_0) = \log_2(N_0/M)$;
2. although the M samples of a given timeslot can be used in all levels, the differences are:
 - the number of M -point FFTs is determined by $K_j (= K_0/2^j)$, and
 - the number of frequency bins is determined by $N_j (= N_0/2^j)$;
3. more importantly, at different levels, the same M samples in the k^{th} timeslot are multiplied with phase rotation factors $W_{N_j}^{m_j(u+kM)}$ in (13), which can be rewritten as $W_{N_0}^{2^j \cdot m_j(u+kM)}$, given that $N_j = N_0/2^j$.

Therefore, (13) can be expressed as:

$$X_{ji}(m_j + rK_j) = \sum_{k=(i-1)K_j}^{iK_j-1} \left\{ \sum_{u=0}^{M-1} x(u + kM) W_{N_0}^{2^j \cdot m_j(u+kM)} W_M^{ru} \right\} \quad (14)$$

Comparing the portion inside the curly braces in (14) with the portion inside the curly braces in (7), we find that after we calculate the DFT (the portion inside the curly braces of (7)) using M samples in a timeslot, the DFT results can not only be used for frequency bins at Level-0, but some of them can also be used for other-level spectrum lines, as long as:

1. these other-level spectrum lines also use M samples from the current timeslot for the calculation; and
2. the frequency block number m_j satisfies the relation given in (15), which means these other-level spectrum lines not only share the timeslot, but also use the same phase rotations factors.

$$2^j \cdot m_j = m_0 \quad (15)$$

where again $m_0 = 0, 1, 2, \dots, K_0 - 1$, and $m_j = 0, 1, 2, \dots, K_j - 1$, where $j = 0, 1, 2, \dots, \log_2(K_0)$ and $K_j = K_0/2^j$.

The above discussion infers that the other-level spectrum lines can be obtained at the same time that the Level-0 spectrum line is being calculated using the BFFT. Since the ADC sampling rate is fixed, and the number of time samples is reduced by half at each progressively higher spectrum level, while the frequency resolution is also reduced by half. The number of spectrum lines is doubled when going from Level- j to Level- $(j + 1)$ if all the time samples used in Level-0 are used in the computation of the spectrum lines at other levels.

To make the above discussion clear, let us see an example. Here, we use $N_0 = 32$ time samples with $M = 4$ samples in each timeslot. Therefore, there are $K_0 = 8$ timeslots, and there are 4 levels of spectrum lines. Table 1 shows the timeslots and time samples that are shared between spectrum lines (X_{ji}) at different levels. For example, 4 samples in $TS = 0$ are used in the calculations of X_{01} , X_{11} , X_{21} , and X_{31} , and 4 samples in $TS = 4$ are used in the calculations of X_{01} , X_{12} , X_{23} , and X_{35} .

Table 1. Timeslots (TSs) and time samples used in different levels spectrum line (SL) calculations.

TS	0				1				2				3				4				5				6				7				
SL	Time Samples $x(n)$																																
X_{01}	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
X_{11}	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
X_{12}																																	
X_{21}	0	1	2	3	4	5	6	7																									
X_{22}									8	9	10	11	12	13	14	15																	
X_{23}																	16	17	18	19	20	21	22	23									
X_{24}																									24	25	26	27	28	29	30	31	
X_{31}	0	1	2	3																													
X_{32}					4	5	6	7																									
X_{33}									8	9	10	11																					
X_{34}													12	13	14	15																	
X_{35}																	16	17	18	19													
X_{36}																					20	21	22	23									
X_{37}																									24	25	26	27					
X_{38}																													28	29	30	31	

There are 8, 4, 2, and 1 frequency blocks at Level-0, -1, -2, and -3, respectively. Table 2 shows that the Level-0 M -point FFT results can also be used to generate other-level spectrum lines, as long as the samples in a timeslot

1. are shared between these spectrum lines (see Table 1), and
2. are multiplied by the same phase rotation factors that meet the criterion described in (15).

Table 2. The M -point FFT results can be reused at different levels $(K_0, K_1, K_2, K_3) = (8, 4, 2, 1)$.

X_{0i}	X_{1i}	X_{2i}	X_{3i}
$\sum_{k=0}^{K_0-1} \left\{ \sum_{u=0}^{M-1} x(u+kM) W_{N_0}^{0(u+kM)} W_M^{ru} \right\}$	$\sum_{k=(i-1)K_1}^{iK_1-1} \left\{ \sum_{u=0}^{M-1} x(u+kM) W_{N_0}^{0(u+kM)} W_M^{ru} \right\}$	$\sum_{k=(i-1)K_2}^{iK_2-1} \left\{ \sum_{u=0}^{M-1} x(u+kM) W_{N_0}^{0(u+kM)} W_M^{ru} \right\}$	$\sum_{k=(i-1)K_3}^{iK_3-1} \left\{ \sum_{u=0}^{M-1} x(u+kM) W_{N_0}^{0(u+kM)} W_M^{ru} \right\}$
$\sum_{k=0}^{K_0-1} \left\{ \sum_{u=0}^{M-1} x(u+kM) W_{N_0}^{1(u+kM)} W_M^{ru} \right\}$			
$\sum_{k=0}^{K_0-1} \left\{ \sum_{u=0}^{M-1} x(u+kM) W_{N_0}^{2(u+kM)} W_M^{ru} \right\}$	$\sum_{k=(i-1)K_1}^{iK_1-1} \left\{ \sum_{u=0}^{M-1} x(u+kM) W_{N_0}^{2(u+kM)} W_M^{ru} \right\}$		
$\sum_{k=0}^{K_0-1} \left\{ \sum_{u=0}^{M-1} x(u+kM) W_{N_0}^{3(u+kM)} W_M^{ru} \right\}$			
$\sum_{k=0}^{K_0-1} \left\{ \sum_{u=0}^{M-1} x(u+kM) W_{N_0}^{4(u+kM)} W_M^{ru} \right\}$	$\sum_{k=(i-1)K_1}^{iK_1-1} \left\{ \sum_{u=0}^{M-1} x(u+kM) W_{N_0}^{4(u+kM)} W_M^{ru} \right\}$	$\sum_{k=(i-1)K_2}^{iK_2-1} \left\{ \sum_{u=0}^{M-1} x(u+kM) W_{N_0}^{4(u+kM)} W_M^{ru} \right\}$	
$\sum_{k=0}^{K_0-1} \left\{ \sum_{u=0}^{M-1} x(u+kM) W_{N_0}^{5(u+kM)} W_M^{ru} \right\}$			
$\sum_{k=0}^{K_0-1} \left\{ \sum_{u=0}^{M-1} x(u+kM) W_{N_0}^{6(u+kM)} W_M^{ru} \right\}$	$\sum_{k=(i-1)K_1}^{iK_1-1} \left\{ \sum_{u=0}^{M-1} x(u+kM) W_{N_0}^{6(u+kM)} W_M^{ru} \right\}$		
$\sum_{k=0}^{K_0-1} \left\{ \sum_{u=0}^{M-1} x(u+kM) W_{N_0}^{7(u+kM)} W_M^{ru} \right\}$			

3. Comparisons of MFR-BFFT Results with FFT Results

Figure 1 shows a segment of I/Q data recorded by an ultra-wideband digital receiver. There is a total of 2^{20} samples at Level-0, and $M = 256$ samples in each timeslot. Four spectrum levels are calculated with 2^{20} , 2^{19} , 2^{18} , and 2^{17} frequency bins at each of the different levels. The ADC sampling rate is 533.333 MHz, so that the frequency resolutions are 508.6, 1017.3, 2034.5, and 4069.0 Hz for Level-0 through Level-3, respectively.

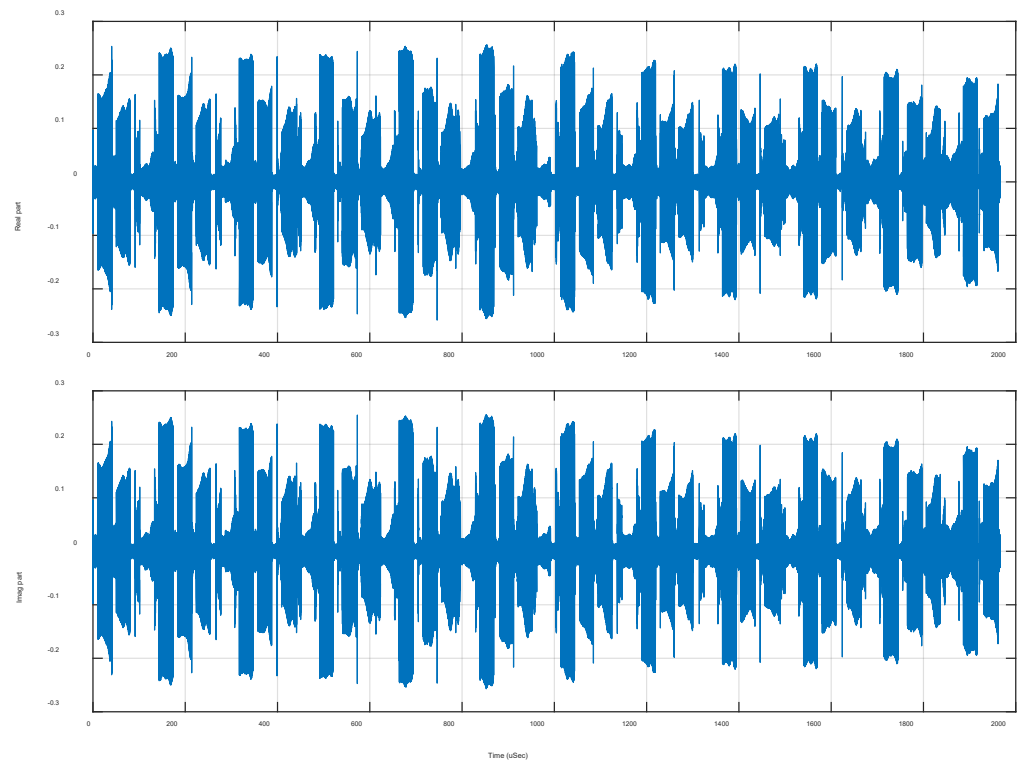
**Figure 1.** A segment of a digitized I/Q signal with a total of 2^{20} samples.

Figure 2 shows the MFR-BFFT results at different spectrum levels. To verify the MFR-BFFT method, the results obtained using the MATLAB “fft” function are overlaid in the figure (with an arbitrary offset in order to improve visibility). The error distributions

between the results in Figure 2 are shown in Figure 3 using histogram plots. The very small errors, which are due to differences in the order in which numerical rounding has occurred, show the correctness of our MFR-BFFT method.

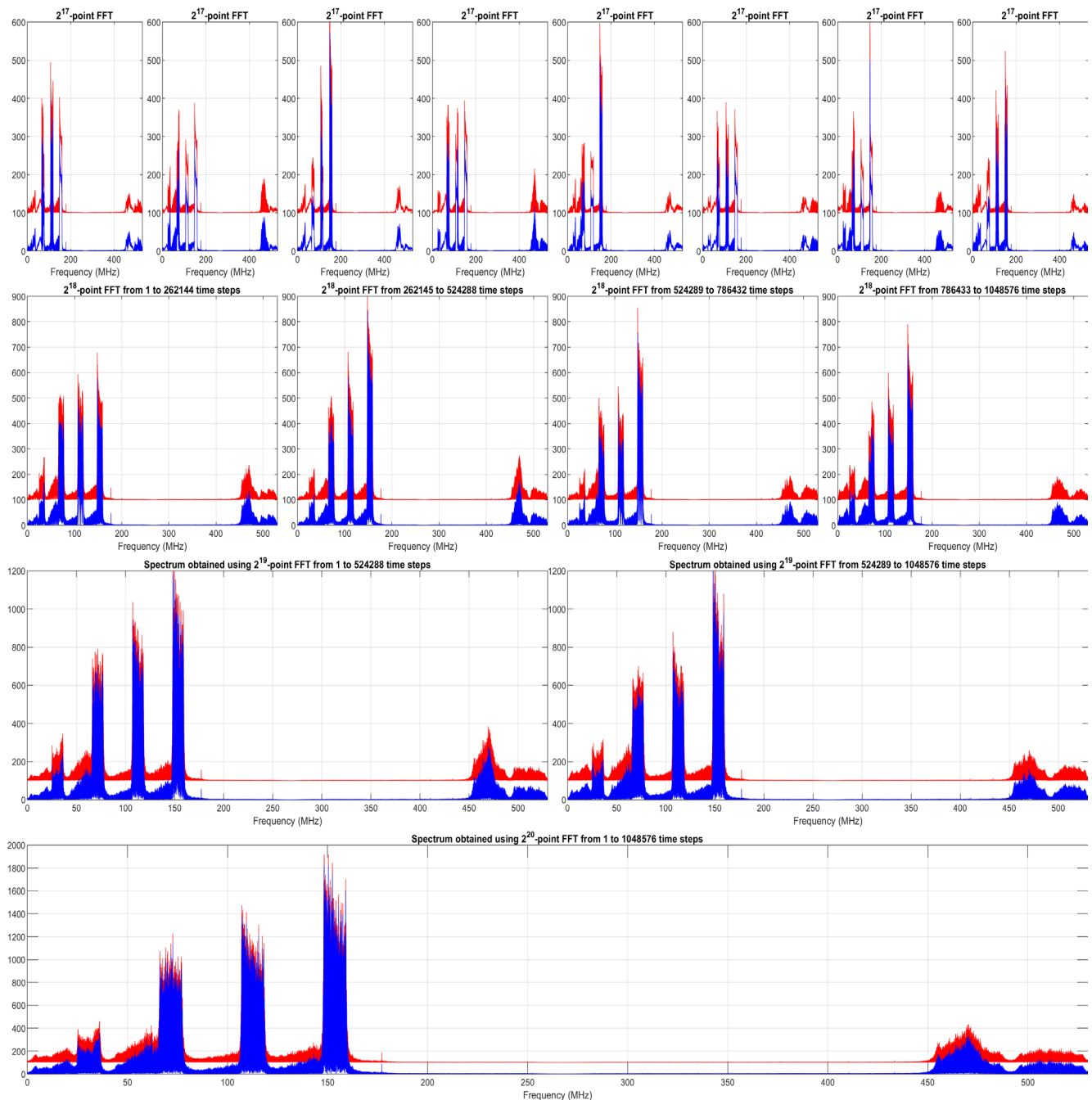


Figure 2. Spectrum lines calculated using different numbers of samples using the MFR-BFFT (blue) and the MATLAB “fft” function (red). Since results are very close, an arbitrary offset of 100 was added to the magnitude of the MATLAB results in order to improve visibility.

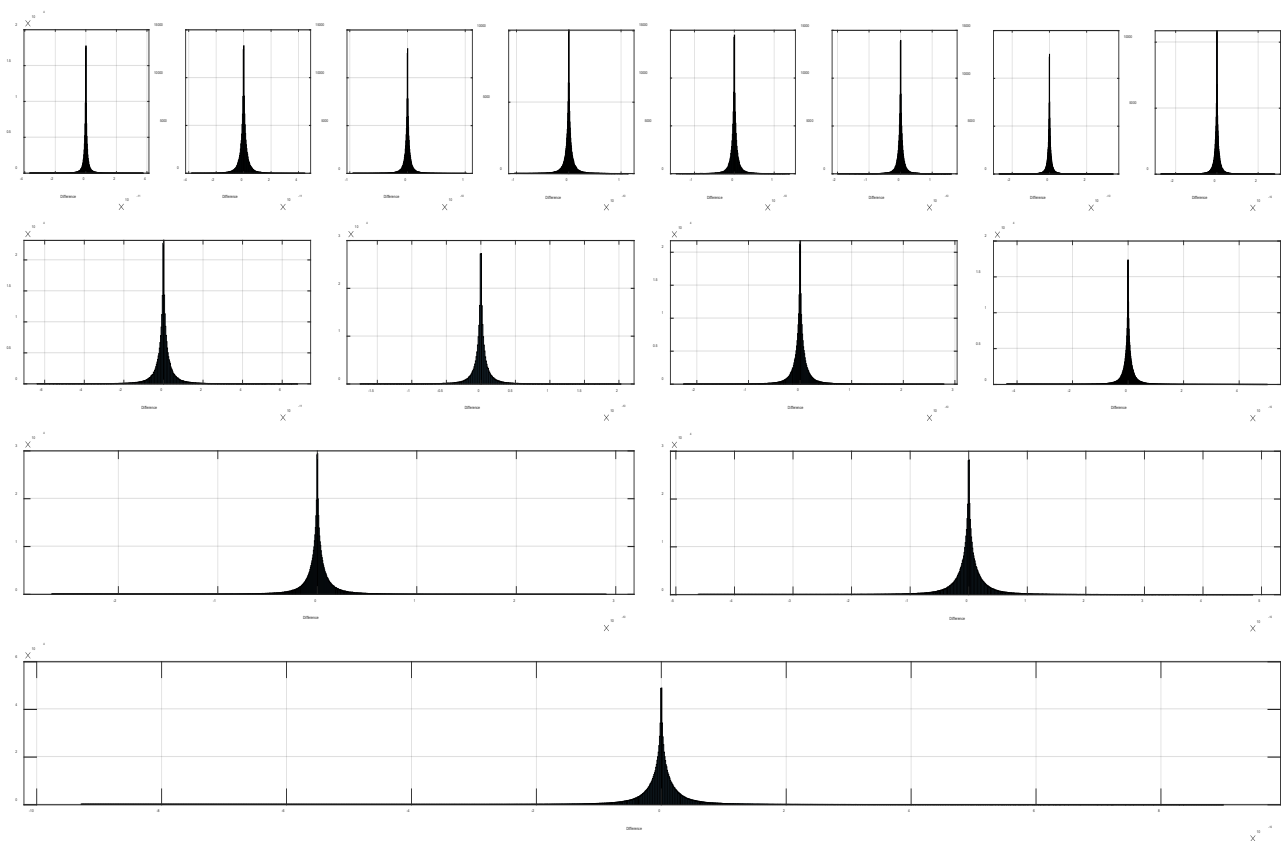


Figure 3. Histograms of the differences between the results in Figure 2 obtained using the MFR-BFFT method and the MATLAB “fft” function at different frequency resolution levels.

4. Conclusions

This communication introduces a new multi-resolution spectrum line calculation method, called the multi-frequency-resolution blocking fast Fourier transform (MFR-BFFT). It can simultaneously calculate a set of very long multi-resolution signal spectra by reusing the BFFT results on all samples used to calculate the finest frequency resolution spectrum. The paper gives the detailed derivation of the MFR-BFFT method and uses actual recorded data to demonstrate the use of the method. The MFR-BFFT results are also compared with conventional FFT results. They are identical for all practical purposes. The success of the MFR-BFFT in simultaneously calculating multi-frequency-resolution spectrum lines using the same set of samples leads to a new and efficient approach for analyzing long-length digitized samples in real-time systems, such as FPGA-based hardware. Although this paper discusses the MFR-BFFT method for applications with long-length signals, this method can be considered for any applications needing multi-resolution spectra and using FPGAs for near-real-time calculations. In follow-on work, the MFR-BFFT method will be combined with the PCW-BFFT method to develop a new DFT-based time-frequency distribution calculation method that can concurrently produce multi-frequency-resolution and multi-time-resolution spectrograms, while consuming digitized data in K smaller blocks of M samples produced by high-speed ADCs.

Author Contributions: The conceptualization, algorithm, funding, and original draft preparation tasks were carried out by C.W. The concept discussions, reviewing, and editing were carried out by M.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Defence Research and Development Canada—Ottawa Research Centre, Department of National Defence, Government of Canada.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All simulated data are available to the readers.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zhuge, J. Multi-Resolution Spectrum Analysis, Application Note 057, Crystal Instruments Corporation, March 2019, Multi-Resolution+Spectrum+Analysis+2023.pdf. Available online: <https://www.squarespace.com> (accessed on 15 September 2023).
2. Calway, A.D.; Knutsson, H.; Wilson, R. Multiresolution Estimation of 2-D Disparity Using a Frequency-Domain Approach. University of Warwick. Department of Computer Science. (Department of Computer Science Research Report). (Unpublished) CS-RR-196. 1991. Available online: https://wrap.warwick.ac.uk/60885/14/WRAP_cs-rr-196.pdf (accessed on 17 September 2023).
3. Shaker, M. EEG Waves Classifier using Wavelet Transform and Fourier Transform, World Academy of Science, Engineering and Technology, Open Science Index 3. *Int. J. Bioeng. Life Sci.* **2007**, *1*, 169–174.
4. Borowska, M.; Białobłocka, N. Multiresolution Analysis of EEG Signals. *Stud. Log. Gramm. Rhetor.* **2017**, *47*, 21–31. [CrossRef]
5. Mallat, S.G. Multiresolution approximations and wavelet orthonormal bases of $L^2(\mathbb{R})$. *Trans. Am. Math. Soc.* **1989**, *315*, 69–87.
6. Mallat, S.G. *A Wavelet Tour of Signal Processing: The Sparse Way*, 3rd ed.; Academic Press: Cambridge, MA, USA, 2008; ISBN 9780123743701/9780080922027.
7. Daubechies, I. The wavelet transform time-frequency localization and signal analysis. *IEEE Trans Inf. Theory* **1990**, *36*, 961–1005. [CrossRef]
8. Guo, T.; Zhang, T.; Lim, E.; López-Benítez, M.; Ma, F.; Yu, L. A Review of Wavelet Analysis and Its Applications: Challenges and Opportunities. *IEEE Access* **2022**, *10*, 58869–58903. [CrossRef]
9. Namias, V. The fractional order Fourier transform and its application to quantum mechanics. *IMA J. Appl. Math.* **1980**, *25*, 241–265. [CrossRef]
10. Almeida, L.B. The fractional Fourier transform and time-frequency representations. *IEEE Trans. Signal Process.* **1994**, *42*, 3084–3091. [CrossRef]
11. Candan, C.; Kutay, M.A.; Ozaktas, H.M. The discrete fractional Fourier transform. *IEEE Trans. Signal Process.* **2000**, *48*, 1329–1337. [CrossRef]
12. Castro, L.P.; Haque, M.R.; Murshed, M.M.; Saitoh, S.; Tuan, N.M. Quadratic Fourier transforms. *Ann. Funct. Anal.* **2014**, *5*, 10–23. [CrossRef]
13. Gupta, B.; Kaur, N.; Verma, A.K.; Agarwal, R.P. Quadratic Phase Multiresolution Analysis and the Construction of Orthonormal Wavelets in $L^2(\mathbb{R})$. *Axioms* **2023**, *12*, 927. [CrossRef]
14. Pihlajamäki, T. *Multi-Resolution Short-Time Fourier Transform Implementation of Directional Audio Coding*; Helsinki University of Technology: Espoo, Finland, 2009.
15. Gnann, V.; Becker, J. Signal reconstruction from multiresolution STFT magnitudes with mutual initialization. In Proceedings of the 45th International Conference: Applications of Time-Frequency Processing in Audio, Helsinki, Finland, 1–4 March 2012; pp. 2–3.
16. Pei, S.C.; Huang, S.G. STFT with adaptive window width based on the chirp rate. *IEEE Trans Signal Process* **2012**, *60*, 4065–4080. [CrossRef]
17. Mateo, C.; Talavera, J.A. Short-time Fourier transform with the window size fixed in the frequency-domain (STFT-FD): Implementation. *Digit. Signal Process.* **2018**, *77*, 13–21. [CrossRef]
18. Wu, C.; Elangage, J. Very Long-Length FFT Using Multi-Resolution Piecewise-Constant Windows for Hardware-Accelerated Time-Frequency Distribution Calculations in an Ultra-Wideband Digital Receiver. *Sensors* **2022**, *22*, 9192. [CrossRef] [PubMed]
19. Tsui, J.B.Y. *Special Design Topics in Digital Wideband Receivers*; Artech House: Norwood, MA, USA, 2010.
20. Xu, P.; Xu, F. A real-time spectral analysis method and its FPGA implementation for long-sequence signals. *Meas. Sci. Technol.* **2019**, *31*, 035006. [CrossRef]
21. Wu, C.; Tang, T.; Elangage, J.; Krishnasamy, D. Accumulatively Increasing Sensitivity of Ultrawide Instantaneous Bandwidth Digital Receiver with Fine Time and Frequency-resolution for Weak Signal Detection. *Electronics* **2022**, *11*, 1018. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.