electronics

Article

Accumulatively Increasing Sensitivity of Ultrawide Instantaneous Bandwidth Digital Receiver with Fine Time and Frequency Resolution for Weak Signal Detection

Chen Wu *, Taiwen Tang, Janaka Elangage and Denesh Krishnasamy

Defence Research and Development Canada, Ottawa Research Centre, Ottawa, ON K1A 0Z4, Canada; taiwen.tang@forces.gc.ca (T.T.); janaka.elangage@forces.gc.ca (J.E.); denesh.krishnasamy@forces.gc.ca (D.K.)
* Correspondence: chen.wu@forces.gc.ca

Abstract: It is always an interesting research topic for digital receiver (DRX) designers to develop a DRX with (1) ultrawide instantaneous bandwidth (IBW), (2) high sensitivity, (3) fine time-of-arrival-measurement resolution (TMR), and (4) fine frequency-measurement resolution (FMR) for weak signal detection. This is because designers always want their receivers to have the widest possible IBW to detect far away and/or weak signals. As the analog-to-digital converter (ADC) rate increasing, the modern DRX IBW increases continuously. To improve the signal detection based on blocking FFT (BFFT) method, this paper introduces the new concept of accumulatively increasing receiver sensitivity (AIRS) for DRX design. In AIRS, a very large number of frequency-bins can be used for a given IBW in the time-to-frequency transform (TTFT), and the DRX sensitivity is cumulatively increased, when more samples are available from high-speed ADC. Unlike traditional FFT-based TTFT, the AIRS can have both fine TMR and fine FMR simultaneously. It also inherits all the merits of the BFFT, which can be implemented in an embedded system. This study shows that AIRS-based DRX is more efficient than normal FFT-based DRX in terms of using time-domain samples. For example, with a probability of false alarm rate of 10^{-7}, for $N = 2^{20}$ frequency-bins with TMR = 50 nSec, FMR = 2.4414 KHz, IBW > 1 GHz and ADC rate at 2.56 GHz, AIRS-based DRX detects narrow-band signals at about −42 dB of input signal-to-noise ratio (Input-SNR), and just uses a little less than $N/2$ real-samples. However, FFT-based DRX have to use all N samples. Simulation results also show that AIRS-based DRX can detect frequency-modulated continuous wave signals with ±0.1, ±1, ±10 and ±100 MHz bandwidths at about −39.4, −35.1, −30.2, and −25.5 dB of Input-SNR using about 264.6 K, 104.7 K, 40.2 K and 18.3 K real-samples, respectively, in 2^{20} frequency-bins for TTFT.

Keywords: weak signal detection; blocking fast fourier transform; fast fourier transform; time-to-frequency transform; ultra-wideband digital receiver; receiver sensitivity; instantaneous bandwidth; accumulatively increasing receiver sensitivity

check for updates

Citation: 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. https://doi.org/10.3390/electronics11071018

Academic Editor: Abdeldjalil Ouahabi

Received: 14 February 2022
Accepted: 22 March 2022
Published: 24 March 2022

1. Introduction

Ultra-wideband digital receiver (DRX) can be considered as a unique type of RF/microwave/millimeter-wave-receiving system, because sometimes not only it does not have any prior knowledge of the input signals, but also often the intercepted signals that are trying to avoid being detected with low-probability intercept (LPI) features [1]. Because it works in the unique electromagnetic signal environment, it requires (1) as wide an instantaneous bandwidth (IBW) as possible to capture signals in a wide frequency spectrum simultaneously, (2) high receiving sensitivity in order to detect signals as far or as early as possible, (3) fine time-measurement resolution (TMR) to determine the intercepted signal time-of-arrival, and (4) fine frequency-measurement resolution (FMR) for better de-interleaving signals and other signal processing that needs frequency information. Broadening a DRX IBW and simultaneously increasing its sensitivity are interesting and challenging research topics for ultra-wideband DRX designers.

In recent years, although many time-to-frequency transform (TTFT) methods have been developed [2], especially the Compressive Sensing based methods [3], which provide an alternative approach to Shannon vision to reduce the number of samples and the sampling rate, for signal reconstruction with noise reduction [4] and signal detection [5], the fast Fourier transform (FFT) is still the most commonly used method after high-speed analog-to-digital convertor (ADC), as it can produce the narrowest measurement band in a conventional DRX design. The latter normally determines the equivalent noise bandwidth B_e [6] and sets the FMR, if there is no further signal processing for frequency measurement in the DRX after FFT. Since the ADC sampling speed determines the Nyquist bandwidth $\left(B_{Nyquist}\right)$ of a DRX, which sets the upper-limit of the DRX IBW, in FFT-based DRX using more samples (N) in TTFT results in narrower B_e, since B_e is proportional to $B_{Nyquist}/N$, and B_e is one of the main factors that determines the DRX noise floor for weak-signal detection. The details of the relation between B_e and a receiver sensitivity can be found in [7–11]. Refs. [7,8] discuss the general concept of receiver tangential sensitivity and how to measure it. Refs. [9–11] present the sensitivity of DRX from an electronic warfare application perspective.

As the sampling frequency increases, the current ADC can easily give the DRX more than 1 GHz IBW with more than 10 bits samples. For example, Xilinx Zynq UltraScale+ RFSoc [12] has 8 14-bit 5 GSamples/s ADCs. Annino [13] predicted that the DRX IBW will be increased to 4 GHz in the near future. It is well known that increasing the samples in the FFT can improve the receiver sensitivity and FMR. Ref. [14] introduced an FFT-based method to measure the intercepted signal pulse descriptor words with multiple FFT frame sizes, which can achieve good TMR and FMR. However, to the authors' best knowledge, most current multi-bit DRX designs, for example in [14], use a few thousand samples in FFT. This is mainly because the hardware, such as field-programmable gate array (FPGA), does not have enough resources to process very large amount of ADC samples for close-to-real-time applications. Furthermore, long-length FFT has poor TMR.

In parallel to the ADC development, very large or long-length FFT on FPGA have been studied for non-real-time applications. Kanders and Mellqvist [15] introduced one-million-point (2^{20}) FFT that was implemented on a single FPGA with a throughput of 233 MSamples/Sec and 95.6 dB signal to quantization noise ratio. FPGA-based 4-channel with about 2^{17}-point FFT was introduced in [16] for space-based synthetic aperture radar application. For biology, astronomy, and medical imaging applications, an FPGA-based high-throughput 2^{20}-point sparse Fourier transform was presented in [17], which can be applied on frequency-sparse data to generate the latest 500 frequency locations and values every 1.16 millisecond. Its FPGA implantation can process streamed input at 0.86 GSamples/Sec. There are many research and development studies on this topic; here, we give just a few examples.

Recently, Xu et al. [18] introduced the blocking FFT (BFFT) method to improve real-time performance to calculate the frequency spectrum of very long-sequence signals in embedded spectrum measurement devices. It also demonstrated the implementation of the method on Xilinx's Zynq-7000 device. The idea of the BFFT is that the total N-point sequential samples are divided into K consecutive time-slots for the calculation of N frequency-bins. Using the $M(=N/K)$ samples in current time-slot that are only available from the ADC, the BFFT does K independent M-point FFTs, after a phase rotation factor is applied onto each of M time-domain samples. Note that these K M-point FFTs can be parallel processed on FPGA to speed up calculations, as M is K times smaller than N. After calculations in the current time-slot, the $K \times M$ frequency components are added to N frequency-bins. The final N-bin spectrum is obtained by superimposing the results from all frequency results from those K time-slots. The merit of the BFFT method is that instead of waiting for a large amount (N) of samples available by storing them in a large onboard memory, and then performing a large N-point FFT, BFFT performs K M-point FFTs simultaneously on a FPGA as soon as M samples are available in the current time-slot. This method tremendously reduces the resource requirement on hardware.

The main contributions of this paper and some comparisons of them with the current technology published in the literature are given below:

- First, we demonstrate how to increase DRX sensitivity and obtain fine FMR, e.g., a few kHz, using a very large number of samples in an ultrawide IBW, e.g., 1 million real samples in more than 1 GHz IBW. The current DRX only uses a few thousands of samples, which results in much coarser FMR in 1 GHz IBW.
- Then for weak-radar-signal detection, we introduce the new concept of accumulatively increasing receiver sensitivity (AIRS). In the AIRS, depending on the intercepted signal bandwidth, DRX does **not** need the full N (e.g., 1 million) real samples to be processed by the BFFT in order to detect a weak signal in N frequency-bins. The signal detection can be progressive, or the DRX sensitivity can be cumulatively increased, while the new time-slot data is becoming available from the ADC and being processed by the BFFT-based TTFT. As soon as the signal amplitudes/powers in certain frequency-bins are higher than a certain threshold, the signal detections can be asserted [19,20]. Since large frequency-bins are used in the given IBW and time-slot-based signal detection is used, the AIRS-based DRX can achieve both fine TMR and FMR. Although the multiframe FFT-based DRX in [14] has good TMR, in order to achieve the FMR and sensitivity as our AIR-based DRX, the DRX in the reference still have to wait until all the samples for the longest frame size (e.g., $N = 2^{20}$) are available. This results in the inability of the multiframe FFT-based DRX to be processed on FPGA.
- The novel concept of time-slot-based thresholds of a given probability of false alarm rate (P_{fas}) for weak signal detection is introduced for the first time in this paper.
- Since the time-slot-based thresholds are used, using a very large frequency-bin size in a given IBW, an AIRS-based DRX just uses less than $N/2$ real-samples for very weak signal detection, where as an FFT-based DRX needs the full N samples to detect the same signal.
- Based on the AIRS, we demonstrate how to design a DRX that has wide IBW with super-high sensitivity, simultaneously with fine TMR and fine FMR.

Note that, in this paper, the input signal-to-noise ratio (Input-SNR) and the output signal-to-noise ratio (Output-SNR) are defined as the SNRs before and after AIRS processing or FFT processing.

This paper is organized as follows. In the next section, the signal detection with very long-length FFT is discussed. The results will be used to compare with AIRS results in the later sections. Section 3 has a brief discussion of the BFFT method, which helps us to present the concept of the AIRS. More details of the BFFT algorithm with FPGA hardware implementation can be found in [18]. Using two examples, Section 4 presents the concept of the AIRS. The thresholds for 90% of probability of detection (POD) with P_{fas} of 10^{-7} for time-slot-based thresholds used in AIRS are also presented in Section 4. Using the thresholds developed in the Section 4, Section 5 demonstrates the continuous wave (CW) signal detection using AIRS. Using different frequency modulation bandwidths (FMBW) of linear frequency-modulated continuous wave (FMCW) signals, Section 6 discusses wideband signal detection using AIRS. Note that, these FMCW signals include both positive and negative chirp slopes. The last section has the conclusion. Appendix A shows the examples that compare the results obtained from MATLAB® FFT function and our BFFT code. Excellent agreements of the results support the correct functionality of the BFFT method used in this study. Appendix B gives the acronym list.

2. Signal Detection from FFT-Based DRX with Different Number of Real Samples

This is a well-known problem and has been discussed in detail in [9]. The purpose of this section is to use the approach in the reference to find the POD with given P_{fas} for very long-length FFT-based signal detections, as the data suitable for a direct comparison with our AIRS results is not available in the open literature.

The parameters used in the study were (1) ADC sampling frequency was at 2.56 GHz, (2) since the real samples were used in the FFT, the single-tone CW signal frequency was randomly picked within 100 and 1180 MHz in each calculation, hence the DRX IBW was about 1080 MHz, and (3) the data lengths used in FFT were from 2^5 to 2^{20}. The signal detection in this study was defined as not only the peak signal in a frequency-bin being higher than the threshold, but also the frequency-bin location should be within ±3 bins of the input signal frequency. Hence, the frequency check was also considered during detection, in this study.

Table 1 compares the thresholds calculated in this study with the results given in [9] from 2^5 to 2^{10} lengths of FFT. The comparison confirms our calculations were correct. The details of the threshold calculation can be found in the Chapter 6 of [9], which will also be briefly discussed in Section 5, when we develop the time-slot-based thresholds for AIRS. Figure 1 shows the calculated thresholds for P_{fas} of 10^{-7} versus the length of FFT from 2^5 to 2^{20}. Using the parameters given above, Figure 2 shows the POD with P_{fas} of 10^{-7}, when the Input-SNR before FFT is against different lengths of FFT. Table 2 summarizes the required minimum Input-SNR for different lengths of FFT in order to have 90% POD with P_{fas} of 10^{-7}. It shows that the FFT lengths of half-a-million and 1 million with −38.9 and −41.9 dB Input-SNR, respectively, can have 90% POD the intercepted signal with 10^{-7} P_{fas}.

Table 1. The comparison between the calculated thresholds for 90% POD with P_{fas} of 10^{-7} in this study and the results given in [9].

FFT Length	32	64	**128**	**256**	512	1024
Threshold	22.67	32.16	45.52	64.21	90.90	128.42
Threshold from [9]	22.92	32.07	45.26	64.24	90.66	128.33

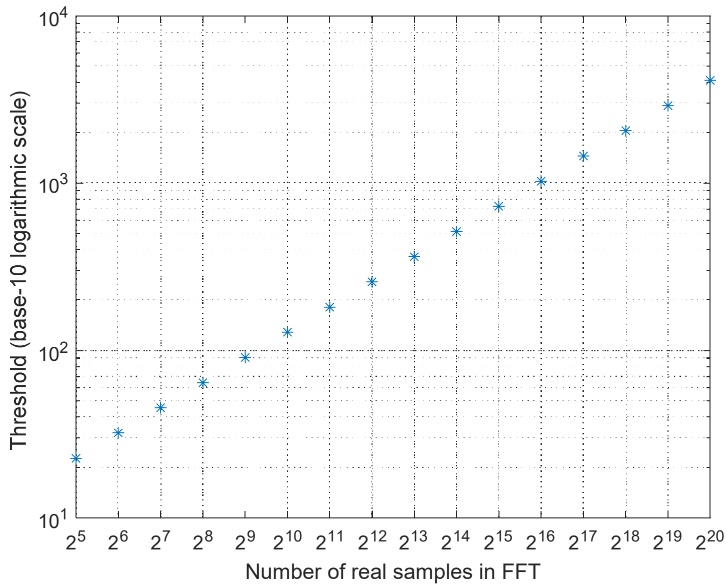

Figure 1. Thresholds for 90% POD with P_{fas} of 10^{-7} versus lengths of FFT.

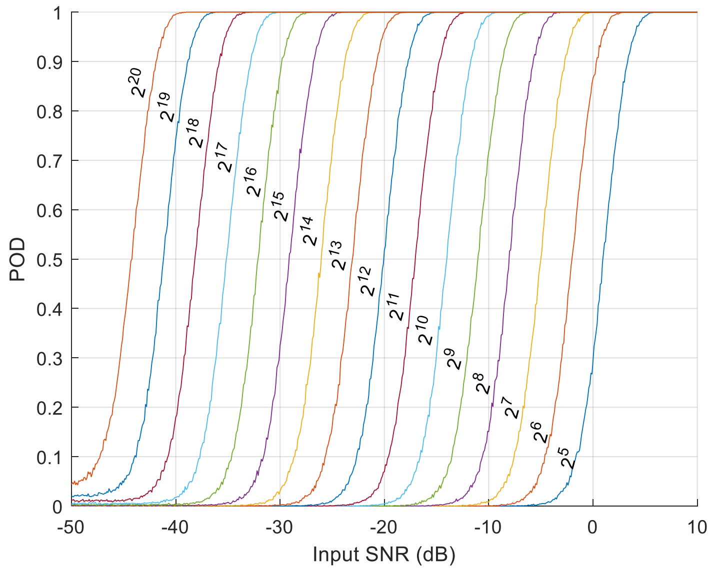

Figure 2. The POD with P_{fas} of 10^{-7} versus Input-SNR before FFT.

Table 2. Required minimum Input-SNR and minimum Output-SNR of 90% POD with P_{fas} of 10^{-7} for different lengths of FFT.

FFT Length	32	64	128	256	512	1024	2^{16}	2^{17}	2^{18}	2^{19}	2^{20}
Min. Input-SNR (dB)	3.3	0.4	−2.7	−5.7	−8.7	−11.7	−29.8	−32.8	−35.9	−38.9	−41.9
Min. Output-SNR (dB)	15.3	15.6	15.4	15.4	15.4	15.4	15.4	15.4	15.3	15.3	15.3

Considering the FFT processing gain, the minimum FFT Output-SNR is also shown in Table 2. The minimum Output-SNR for any length of FFT is about 15.4 dB, when the Input-SNR is at the levels given in second row of the table. As described in [9], the FFT-processing gain is defined as $10\log_{10}(\text{Bandwidth reduction through FFT})$. For example, since the sampling rate is 2.56 GHz the $B_{Nyuist} = 1280$ MHz before FFT, and the frequency-bin size after the 1024-point FFT is 2.5 MHz, the bandwidth reduction is 512, so the processing gain is 27.1 dB. The processing gain also can be calculated using $10\log_{10}(N/2)$, which we will use to calculate AIRS processing gain in the discussions of Sections 5 and 6.

Although FFT with long-length input data can detect weak signals, the traditional FFT-based signal detection has following inherent problems:

- It cannot achieve fine TMR and fine FMR, simultaneously.
- To detect a weak signal, long-length FFT has to be used. This delays detection time, since all the samples have to be collected before starting the processing.
- The collected samples require large memory and resources in the embedded system.

To overcome these problems and take the full advantages of both short-FFT (with fine TMR) and long-length FFT (with fine FMR), in this paper, for the first time we introduce the AIRS method, which is based on the BFFT method.

3. Briefing on the BFFT Method

In order to assist us presenting the AIRS method, in this section we briefly summarize the BFFT method. More details can be found in [18]. The discrete Fourier transform (DFT) is defined as:

$$X(v) = \sum_{n=0}^{N} x(n) W_N^{nv} \tag{1}$$

where N is total number of time-domain samples, $v = 0, 2, \ldots\ N-1$ and $W_N^{nv} = e^{-\frac{j2\pi nv}{N}}$.

In BFFT, the N samples are divided into K equal time-slots and each time-slot has M time samples generated by an ADC, and then (1) can be written as:

$$X(v) = \sum_{n=0}^{M-1} x(n) W_N^{nv} + \cdots + \sum_{n=kM}^{(k+1)M-1} x(n) W_N^{nv} + \ldots + \sum_{n=(K-1)M}^{N-1} x(n) W_N^{nv} \tag{2}$$

Let $u = n - kM$ and considering the k^{th} time-slot samples:

$$\sum_{n=kM}^{(k+1)M-1} x(n) W_N^{nv} = \sum_{u=0}^{M-1} x(u + kM) W_N^{(u+kM)v} \tag{3}$$

BFFT defines:

$$Y_k(v) = \sum_{u=0}^{M-1} x(u + kM) W_N^{(u+kM)v} \tag{4}$$

Then (2) can be written as:

$$X(v) = \sum_{k=0}^{K-1} Y_k(v) \tag{5}$$

Note that, $Y_k(v)$ has the same form as in (1). However, since there are only M time samples in (4), the FFT can only produce M frequency-bins. In order to let the M samples in k^{th} time-slot contribute to overall N-point spectrum, BFFT also divides N frequency-bins into K equal length bin-blocks, and each frequency block also has M frequency-bins.

Let $v = Kr + m$, where $m = 0,\ 1,\ 2,\ \ldots,\ K-1$ and $r = 0,\ 1,\ 2,\ \ldots, M-1$, then (4) can be written as:

$$Y_k(Kr + m) = \sum_{u=0}^{M-1} x(u + kM)\ W_N^{(u+kM)(Kr+m)} = \sum_{u=0}^{M-1} x(u + kM) W_N^{m(u+kM)} W_M^{ru} \tag{6}$$

In (6), $W_N^{kMKr} = 1$ and $KM = N$ are used.
We define:

$$y(u + kM, m) = x(u + kM)\ W_N^{m(u+kM)} \tag{7}$$

In (7), $x(u + kM)$ denotes the k^{th} time-slot samples, which are in between kM and $(k+1)M - 1$ time steps, and u is the time-index in the k^{th} time-slot ($u = 1,\ 2,\ \ldots M-1$). (7) tells that, as soon as these M time-domain samples are available, each needs to be multiplied by a phase rotation factor ($W_N^{m(u+kM)}$) before being used in spectrum calculations by FFT. A phase rotation factor is determined by the local frequency location (m) in a frequency block and the time-step $n = (u + kM)$ in overall sampling time series. Table 3 shows the relation between time-index u in each time-slot and the time-index n in overall ADC sampled data.

Table 3. The relation between the time-index u in each time-slot and the overall sampled data.

Time-Slot	0	1	$\cdots$	k	$\cdots$	$K-1$
u				$0,\ 1,\ 2,\ 3,\ \ldots\ldots\ M-1$		
$n\ (= u + kM)$	$0, 1, \ldots\ M-1$	$M,\ M+1,\ \ldots\ 2M$		$kM,\ kM+1,\ \ldots\ (k+1)M$		$(K-1)M,\ (K-1)M+1, \ldots N$

Then (7) can be expressed as:

$$y(n, m) = x(n)\ W_N^{nm} \tag{8}$$

From (6), we have following modified M-point DFT, which can be calculated by FFT.

$$Y_k(Kr + m) = \sum_{u=0}^{M-1} y(n, m) W_M^{ru} \tag{9}$$

Since $m = 0, 1, 2, \ldots, K-1$, (9) can be calculated by K independent M-point FFTs in a FPGA. This is the reason why the method is called BFFT.

The DFT in (1) with consideration of implementing using the BFFT, the $(Kr+m)^{th}$ frequency component can be expressed as:

$$X(Kr+m) = \sum_{k=0}^{K-1} Y_k(Kr+m) = \sum_{k=0}^{K-1} \sum_{u=0}^{M-1} y(n,m) W_M^{ru} \tag{10}$$

with $m = 0, 1, 2, \ldots, K-1$, $r = 0, 1, 2, \ldots, M-1$ and $N = KM$.

Tables 4–7 illustrate how to apply the BFFT for the DFT with very long sampled data.

Table 4. $K \times M$ frequency-domain results obtained from M samples in the 1st time-slot ($k = 0$), and n in the table is given in the 2nd column of Table 3.

M	$Y_0(Kr+m)$	After K-independent M-point FFTs, the first set of $K \times M$ frequency-domain data Y_0 that will be added to N frequency-bins.
0	$\sum_{u=0}^{M-1} y(n,0)W_M^{ru}$	$Y_0(0)$, $Y_0(K)$, $Y_0(2K)$, ..., $Y_0((M-1)K)$
1	$\sum_{u=0}^{M-1} y(n,1)W_M^{ru}$	$Y_0(1)$, $Y_0(K+1)$, $Y_0(2K+1)$, ..., $Y_0((M-1)K+1)$
2	$\sum_{u=0}^{M-1} y(n,2)W_M^{ru}$	$Y_0(2)$, $Y_0(K+2)$, $Y_0(2K+2)$, ..., $Y_0((M-1)K+2)$
...	...	...
$K-1$	$\sum_{u=0}^{M-1} y(n,K-1)W_M^{ru}$	$Y_0(K-1)$, $Y_0(2K-1)$, $Y_0(3K+1)$, ..., $Y_0(N-1)$

Table 5. $K \times M$ frequency-domain results obtained from M samples in the 2nd time-slot ($k = 1$), and n in the table is given in the 3rd column of Table 3.

M	$Y_1(Kr+m)$	The 2nd set of frequency $-$ domain data Y_1 that will be added to N frequency-bins.
0	$\sum_{u=0}^{M-1} y(n,0)W_M^{ru}$	$Y_1(0)$, $Y_1(K)$, $Y_1(2K)$, ..., $Y_1((M-1)K)$
1	$\sum_{u=0}^{M-1} y(n,1)W_M^{ru}$	$Y_1(1)$, $Y_1(K+1)$, $Y_1(2K+1)$, ..., $Y_1((M-1)K+1)$
2	$\sum_{u=0}^{M-1} y(n,2)W_M^{ru}$	$Y_1(2)$, $Y_1(K+2)$, $Y_1(2K+2)$, ..., $Y_1((M-1)K+2)$
...	...	...
$K-1$	$\sum_{u=0}^{M-1} y(n,K-1)W_M^{ru}$	$Y_1(K-1)$, $Y_1(2K-1)$, $Y_1(3K+1)$, ..., $Y_1(N-1)$

Table 6. $K \times M$ frequency-domain results obtained from the k^{th} time-slot M samples, and n in the table is given in the $(k+1)^{th}$ column of Table 3.

M	$Y_k(Kr+m)$	The k^{th} set of frequency-domain data Y_k that will be added to N frequency-bins.
0	$\sum_{u=0}^{M-1} y(n,0)W_M^{ru}$	$Y_k(0)$, $Y_k(K)$, $Y_k(2K)$, ..., $Y_k((M-1)K)$
1	$\sum_{u=0}^{M-1} y(n,1)W_M^{ru}$	$Y_k(1)$, $Y_k(K+1)$, $Y_k(2K+1)$, ..., $Y_k((M-1)K+1)$
2	$\sum_{u=0}^{M-1} y(n,2)W_M^{ru}$	$Y_k(2)$, $Y_k(K+2)$, $Y_k(2K+2)$, ..., $Y_k((M-1)K+2)$
...	...	...
$K-1$	$\sum_{u=0}^{M-1} y(n,K-1)W_M^{ru}$	$Y_k(K-1)$, $Y_k(2K-1)$, $Y_k(3K+1)$, ..., $Y_k(N-1)$

Table 7. $K \times M$ frequency-domain results obtained from the k^{th} time-slot samples $(k = K-1)$ and n in the last column of Table 3.

m	$Y_{K-1}(Kr+m)$	The last set of frequency-domain data Y_{K-1} that will be added to N frequency-bins.
0	$\sum_{u=0}^{M-1} y(n,0)W_M^{ru}$	$Y_{K-1}(0),\ Y_{K-1}(K),\quad Y_{K-1}(2K),\ \ldots\ldots,\ Y_{K-1}((M-1)K)$
1	$\sum_{u=0}^{M-1} y(n,1)W_M^{ru}$	$Y_{K-1}(1),\ Y_{K-1}(K+1),\ Y_{K-1}(2K+1),\ldots,Y_{K-1}((M-1)K+1)$
2	$\sum_{u=0}^{M-1} y(n,2)W_M^{ru}$	$Y_{K-1}(2),Y_{K-1}(K+2),\ Y_{K-1}(2K+2),\ldots,\ Y_{K-1}((M-1)K+2)$
...	...	...
$K-1$	$\sum_{u=0}^{M-1} y(n,K-1)W_M^{ru}$	$Y_{K-1}(K-1),\ Y_{K-1}(2K-1),\ Y_{K-1}(3K+1),\ldots,Y_{K-1}(N-1)$

From the above discussions, we can find that:

- In each table, BFFT does K independent M-point FFTs on the currently available M samples from the ADC in the current time-slot.
- Those K FFTs make the current M samples contribute to the spectrum calculation in N frequency-bins, i.e., when current K FFTs are completed, based on (10), the $K \times M$ frequency results will be added to the corresponding frequency-bins from 0 to $N-1$.
- A complete BFFT needs the K^2 number of M-point FFTs to finish full N-point frequency spectrum calculation, which equals one N-point FFT. However, from signal detection perspective in a DRX, as soon as a frequency-bin power exceeds the threshold, a signal detection can be asserted.

In summary, the advantages of the use of the BFFT in DRX application are not only producing very-large frequency-bins within a given IBW using much shorter length FFT calculations, but also providing the AIRS capability for DRX, which will be discussed and demonstrated in following sections.

4. Accumulatively Increasing Receiver Sensitivity (AIRS) for Ultra-Wideband DRX

4.1. Examples to Demostrate the Comcept of AIRS

As discussed in the last section, the BFFT method not only can partition very large number of samples into time-slots with much smaller number of samples for spectrum calculations, but more importantly, from signal-detection perspective, it also can give DRX the ability of using the currently available M real-samples to contribute $N/2$ frequency-bin spectrum calculations. Figure 3 shows the calculated spectrum using the first 24 time-slots for a CW detection, when the Input-SNR is at -20 dB. The results in the figure show that, at the beginning, the peak gives wrong estimated frequency since noise power dominates the power in the bin. As more time-slot data are being used, the peak location is getting closer to the signal frequency, and the peak amplitude increases in the bin. In reality, when the peak is bigger than the time-slot-based threshold corresponding to the number of time-slot data used, the signal detection can be declared, and the time of detection can be recorded using the time of that time-slot. Therefore, the TMR of an AIRS-based DRX is equal to the length of the time-slot, even though a large number of frequency-bins are considered. The FMR is determined by the bin size. Figure 4 shows similar results as in Figure 3 for an FMCW signal with 50 MHz FMBW.

Figure 3. Peak amplitude and estimated frequency at the peak location using real-samples from (**a**) the first time-slots, (**b**) the first 2 time-slots, and (**c**–**x**) are the first 3 to 24 time-slots of the CW signal at 305.01 MHz, where $N = 2^{20}$, $M = 256$, Input-SNR = -20 dB.

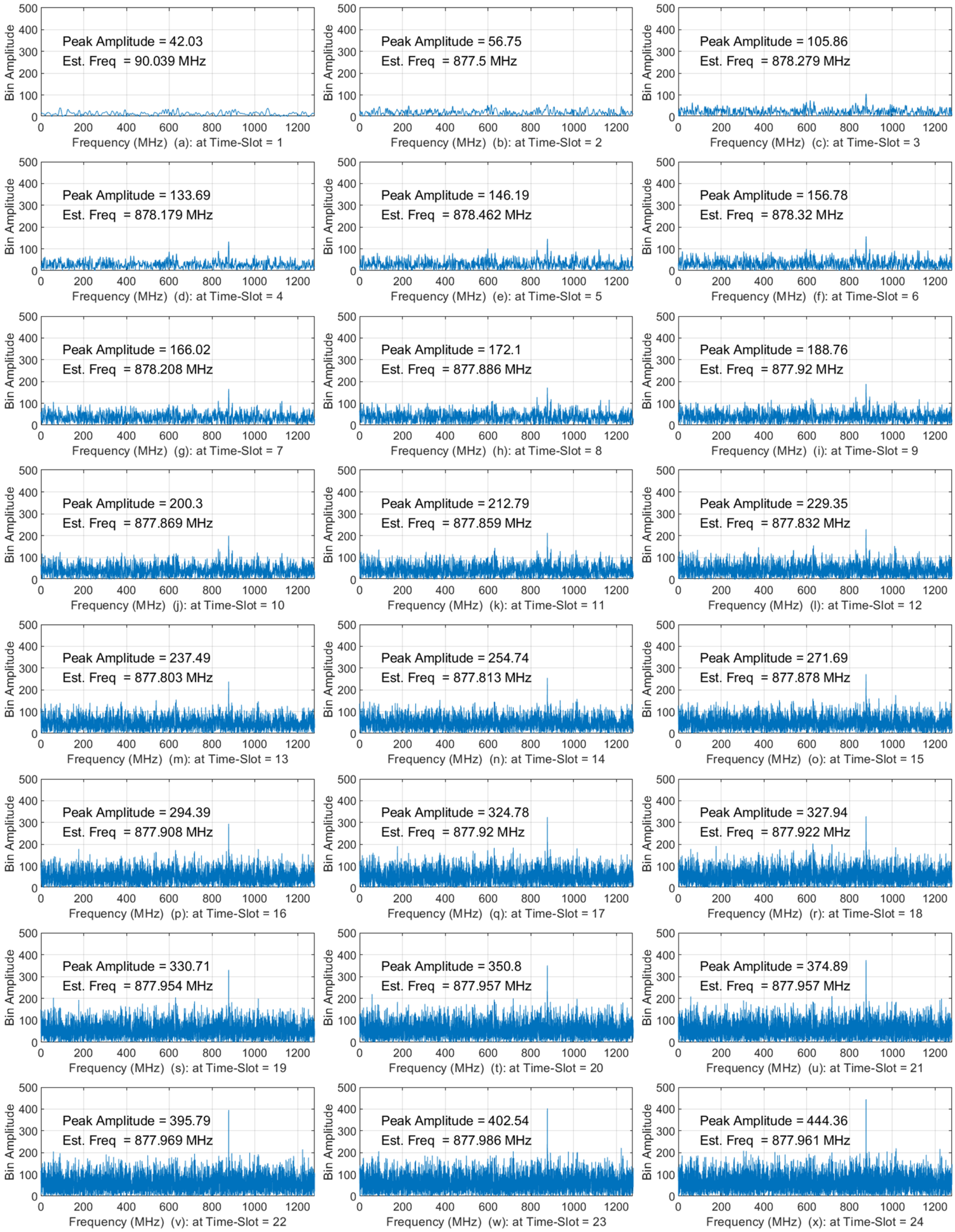

Figure 4. Peak amplitude and estimated frequency at the peak location using real samples from (**a**) the first time-slots, (**b**) the first 2 time-slots, and (**c**–**x**) the first 3 to 24 time-slots of the FMCW signal starting at 855.01 MHz with 50 MHz FMBW, where $N = 2^{20}$, $M = 256$, Input-SNR = -20 dB. Since the signal has a wide frequency band, the frequency portion in the N samples in this example was from 877.918 to 888.174 MHz, and the frequency components of the FMCW in the first 24 time-slots of N samples more focused on the beginning part of the spectrum.

4.2. Thresholds of 90% POD with P_{fas} of 10^{-7} in AIRS Using Different Number of Time-Slot Data

Using the same threshold calculation method applied in Section 2, in this section, we study the time-slot-based thresholds, when different number of time-slot data are used in AIRS as deliberated in Figures 3 and 4.

First, we assumed that the ADC samples had no signal and its output noise samples had Gaussian distribution. In order to obtain the frequency-domain noise (amplitude) distribution, we collected amplitude data in N frequency-bins after each time-slot noise samples being used in AIRS. We used $N = 2^{20}$ and $M = 128$ as an example. After the calculations in Table 4 using the 1st time-slot noise data ($k = 0$), there were total of $10 \times N/2$ complex noise data in the frequency-domain, where 10 came from 10 independent simulations. The amplitude data in the first half of N frequency-bins were useful since inputs were real noise samples. The histogram plot of frequency-domain noise amplitude data is shown in the first subplot in Figure 5. It can be approximated by the Rayleigh distribution. Then, the 2nd time-slot noise samples ($k = 1$) were used in the calculations as described in Table 5 to obtain the 2nd set of $10 \times N/2$ complex noise data. The histogram plot of the 2nd subplot of Figure 5 has the noise amplitude contribution using both the 1st and 2nd time-slot noise samples. The distribution also can be approximated by the Rayleigh distribution. This process was carried out until using the K^{th} time-slot noise samples ($k = K - 1$) according to the calculations given in Table 7.

The first 10 subplots in Figure 5 show frequency-domain noise distributions, after each addition of time-slot noise data processed by the AIRS. Figure 5 also shows frequency-domain noise amplitude distributions, when the subsequent time-slot noise data are used. Results display that when more time-slot noise data are used in the calculations, the distributions can be clearly represented by the Rayleigh distribution. Further simulations show that the Rayleigh distribution can also be found in other N and M combinations that can be used in AIRS. From a statistical perspective, it has been shown [21] that the squared coefficients of the sliding-window DFT with white noise input signals is asymptotically distributed as Chi-square [22] with two degrees of freedom. Thus, the distribution of the amplitude of these noise coefficients is Rayleigh distribution.

Once we confirm that the time-slot-based frequency-domain noise amplitude distribution is Rayleigh distribution in AIRS, the equations from (6.2) to (6.4) of [9] can be used to calculate the thresholds from the 1^{st} to K^{th} time-slots for any given N and M combinations. These equations are given in (11) to (14). The Rayleigh distribution is defined as:

$$p(r) = \frac{r}{\sigma^2} e^{\frac{-r^2}{2\sigma^2}} \tag{11}$$

where σ^2 is a constant and can be calculated by either the mean (μ) or the standard deviation (s) of the measured noise distribution. They were obtained from the above simulations.

$$\sigma = \frac{\mu}{\sqrt{\pi/2}} \tag{12}$$

$$\sigma = \sqrt{\frac{s}{4 - \pi}} \tag{13}$$

The σ values obtained from (12) and (13) were very close. In this study the average of the two values was used. Once the noise distribution is approximated by the Rayleigh, the threshold can be set a function of P_{fas}:

$$thr = \sqrt{-2\sigma^2 \ln\left(P_{fas}\right)} \tag{14}$$

The time-slot-based thresholds are shown in Figure 6 for

- The number of frequency-bins equals to 2^{17}, 2^{18}, 2^{19} and 2^{20};

- In each case, the number of real noise samples (M) in a time-slot equals 128, 256 and 512, respectively.

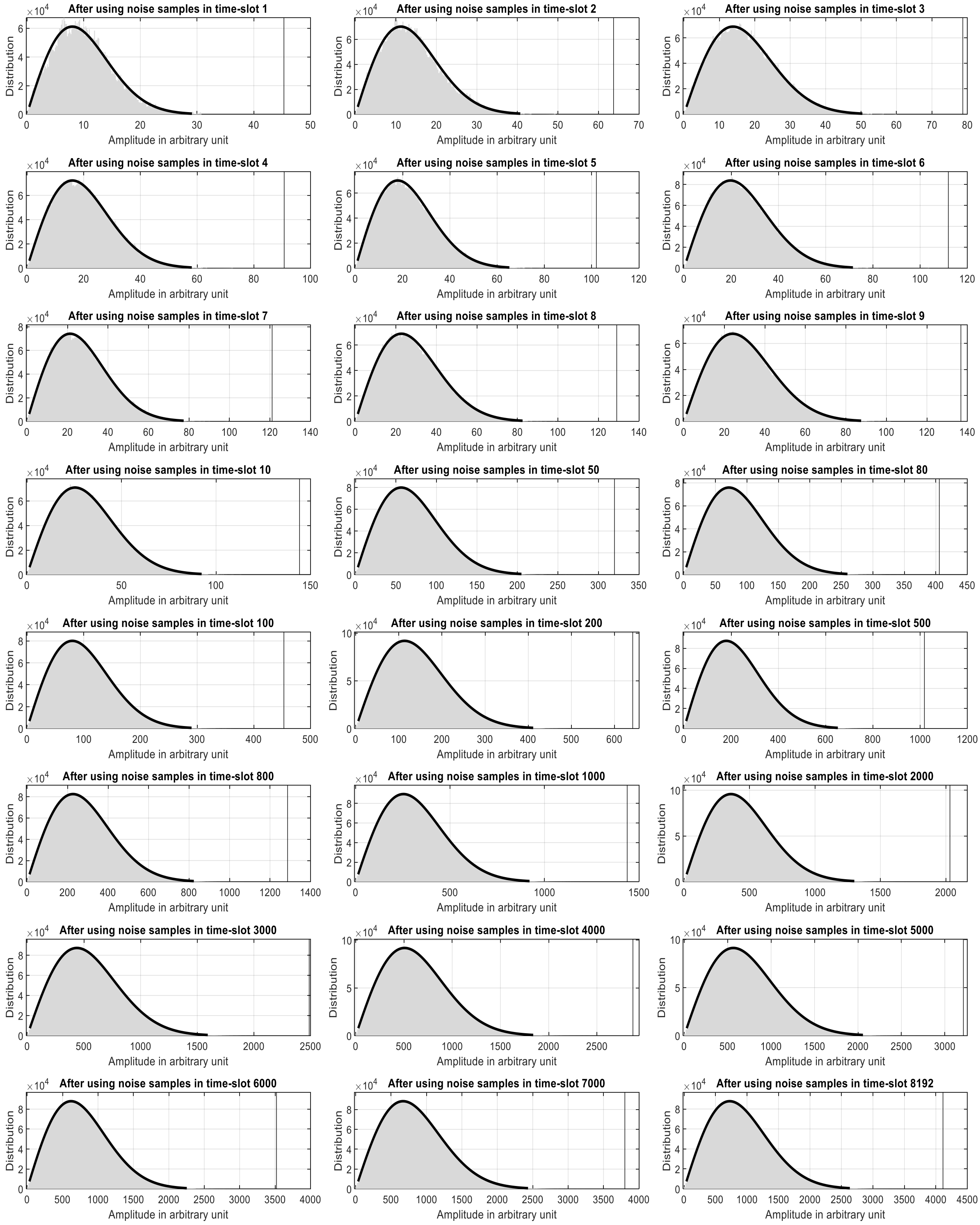

Figure 5. Examples of the frequency-domain noise distribution (histogram plot in gray color) when noise amplitude data from different time-slots are used, where $N = 2^{20}$ and $M = 128$. The solid curve line and vertical line in each subplot are the Rayleigh distribution and the threshold for P_{fas} of 10^{-7}. Those time-slot-based thresholds will be used in detection after corresponding time-slot samples being processed in AIRS.

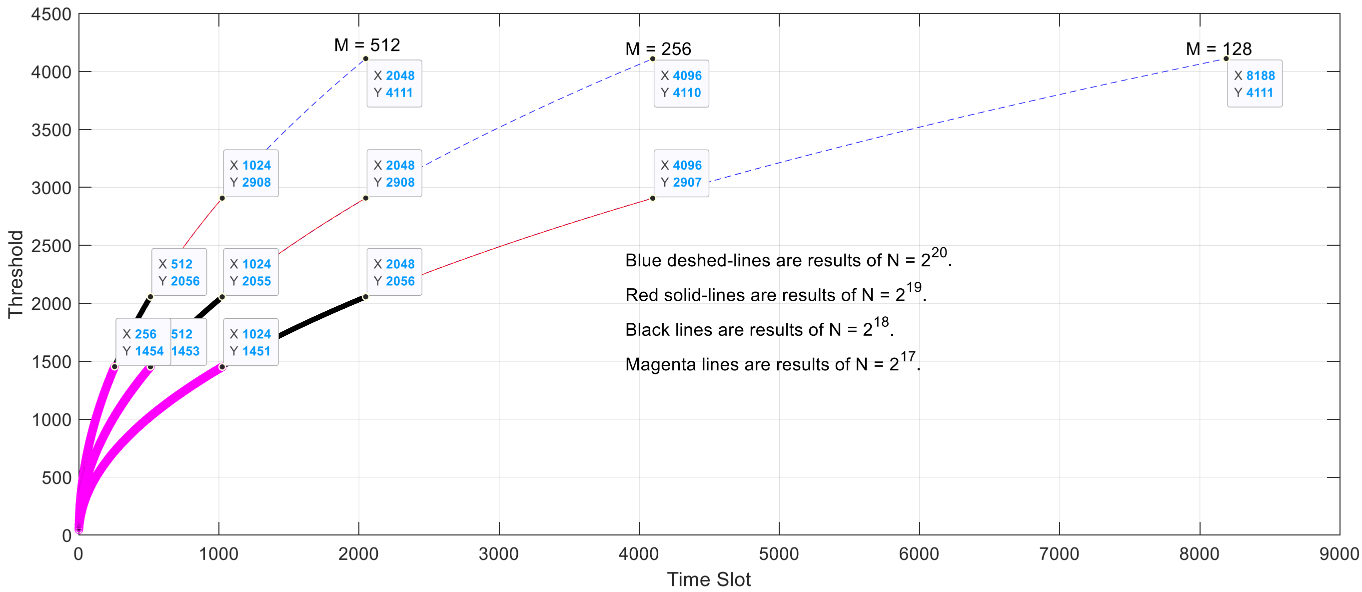

Figure 6. The time-slot-based thresholds with P_{fas} of 10^{-7} versus time-slots. Note that all curves start from the first time-slot and for a given M different color curves are superposed on top of each other in early time-slots.

The results in Figure 6 show that:

1. For a predefined number of frequency-bins in TTFT, to obtain the same threshold level, the total number of noise samples used in the calculations should be the same, even though the samples of time-slots can be different. For example, when the number of frequency-bins is 2^{20}, in order to reach the same threshold 4111, 512×2048 number of samples are needed irrespective of how the partition of the real noise samples is done in a time-slot. This result can help us to determine how to partition the samples in time-slots based on
   a. The required TMR, and/or
   b. The available hardware resources in the embedded system.
2. The threshold versus time-slot is independent to the predefined number of bins in TTFT, and only depends on how to partition samples in a time-slot. This is because when calculating the threshold in a given time-slot, the M Gaussian-distributed noise samples make the contribution to all N bins after being multiplied by the phase rotation factors and KM-point FFTs. Although for different N the phase rotation factors are different, these factors do not introduce any new noise. So, when M is fixed, for the same time-slot the threshold should be the same regardless of how many predefined frequency-bins in the TTFT. This result allows us to keep just one set of threshold tables in AIRS for a given M, when the method is implemented in an embedded system. For example, we just use the dashed-line results in Figure 6 in following simulations.

In AIRS, after using the real samples in the current time-slot to finish KM-point FFTs and adding the results to the N frequency-bins, bin-values in $[0\ N/2]$ are compared with the time-slot-based threshold of the corresponding time-slot given in Figure 6. Those bins with their values bigger than the threshold are reported. In reality, a peak-bin tracking program is needed to track detected signals for further processing.

5. The POD with P_{fas} of 10^{-7} versus Input-SNR in AIRS for CW Signal Detection

Using the thresholds developed in the last section, this section discusses POD of AIRS with different N and M combinations using the same method used in normal FFT-based signal detection. As discussed in Section 2, the single tone CW signal frequency was randomly picked in between 110 to 1180 MHz.

For $N = 2^{17}$, Figures 7 and 8 show the POD with samples from different number of time-slots and the required Input- and Output-SNR for 90% POD, respectively, with $M = 128$, 256 and 512 cases in AIRS. From these two figures, we find that:

1. As more time-slot samples are used, the required Input-SNR is reduced for a given POD. However, the rate of reduction decreases, as shown in Figure 7. Curves get more crowded as Input-SNR decreases.
2. When samples in all the time-slots are used, the left-end curves of the three subplots in Figure 7 are all the same as the 2^{17}-POD curve shown in Figure 2, and the required Input-SNR for 90% POD is about −32.8 dB as given in the Table 2. This is the expected result, since using all the time-slot data in AIRS is the same as using N-point FFT.
3. As long as the same amount of samples are used in AIRS, it produces the same POD curve. For example, the red curve of four time-slots with $M = 128$ in Figure 7a is the same as the red curve of two time-slots with $M = 256$ in (b) and also is the same as the red curve of the first time slot with $M = 512$ in (c). The required Input-SNRs for 90% POD of these three curves are listed in the figure, which are −10.42, −10.42 and −10.52 dB, respectively.
4. The required minimum Output-SNRs are about the same, as long as the same number of samples is used in the AIRS, regardless of how the partition of the samples in a time-slot is done. For example, as shown in the bottom subplot of Figure 8, to detect signal at −31.5 dB Input-SNR with 90% POD, AIRS needs minimum of −14 dB Output-SNR, which needs samples from 560, 280 and 140 time-slots when $M = 128$, 256 and 512, respectively. Here the Output-SNR is defined as:

$$\text{Output-SNR} = \text{Input-SNR} + 10 \times \log_{10}(kM/2) \tag{15}$$

   where k is the index of a time-slot, and $10 \times \log_{10}(kM/2)$ can be viewed as AIRS-processing gain, which is equivalent to the FFT progressing gain discussed in Section 2.
5. Figure 8 also shows that for AIRS, the minimum required Output-SNR is about 15.4 dB for 90% POD with P_{fas} of 10^{-7} when all $N = 2^{17}$ samples are used regardless of how the samples are partitioned in a time-slot. This is similar to the results given in Table 2. However, the AIRS can be more flexible to detect a signal using much less time-samples and maintain good TMR and FMR. Table 8 shows the TMR and FMR with different N and M combinations used in this study.
6. All the above observations can also be found in
   a. Figures 9 and 10 for $N = 2^{18}$ case;
   b. Figures 11 and 12 for $N = 2^{19}$ case, and
   c. Figures 13 and 14 for $N = 2^{20}$ case.

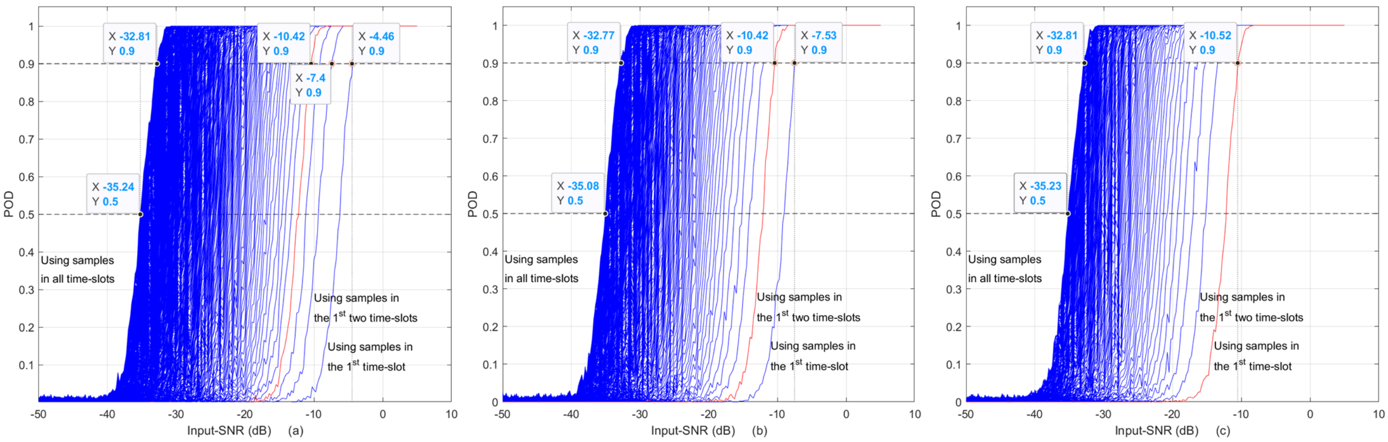

Figure 7. $N = 2^{17}$, the POD (with P_{fas} of 10^{-7}) vs. Input-SNR, in each subplot when only the 1st time-slot samples (**right**) and all time-slots sample (**left**) are used in AIRS, (**a**) $M = 128$, (**b**) $M = 256$ and (**c**) $M = 512$.

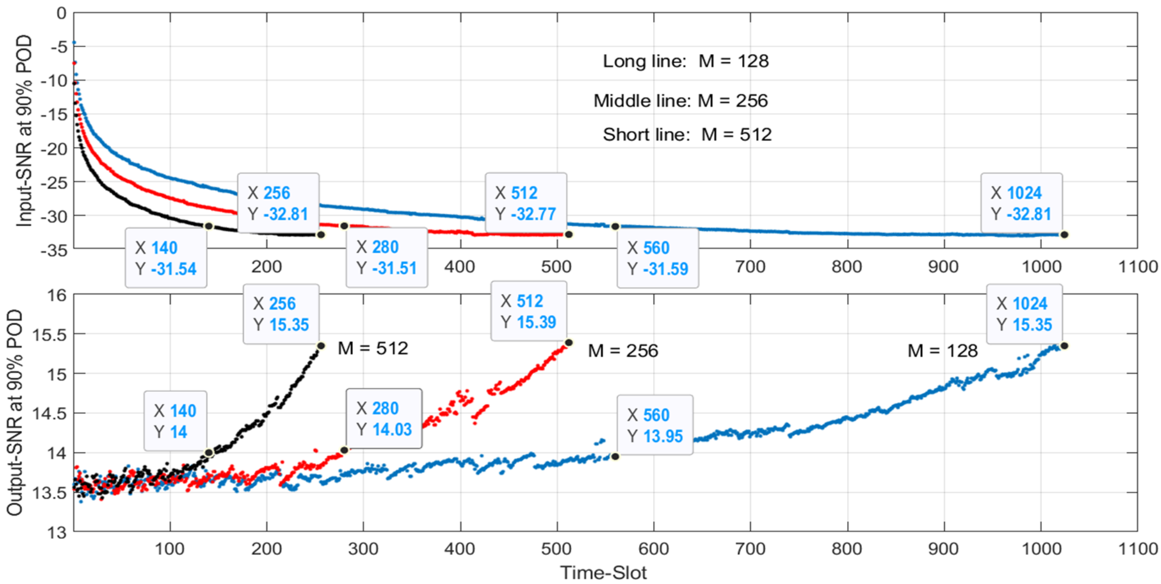

Figure 8. $N = 2^{17}$, Input-SNR and Output-SNR at 90% POD with P_{fas} of 10^{-7}, when different time-slot samples are used in AIRS, blue line: $M = 128$, red line: $M = 256$ and black line: $M = 512$.

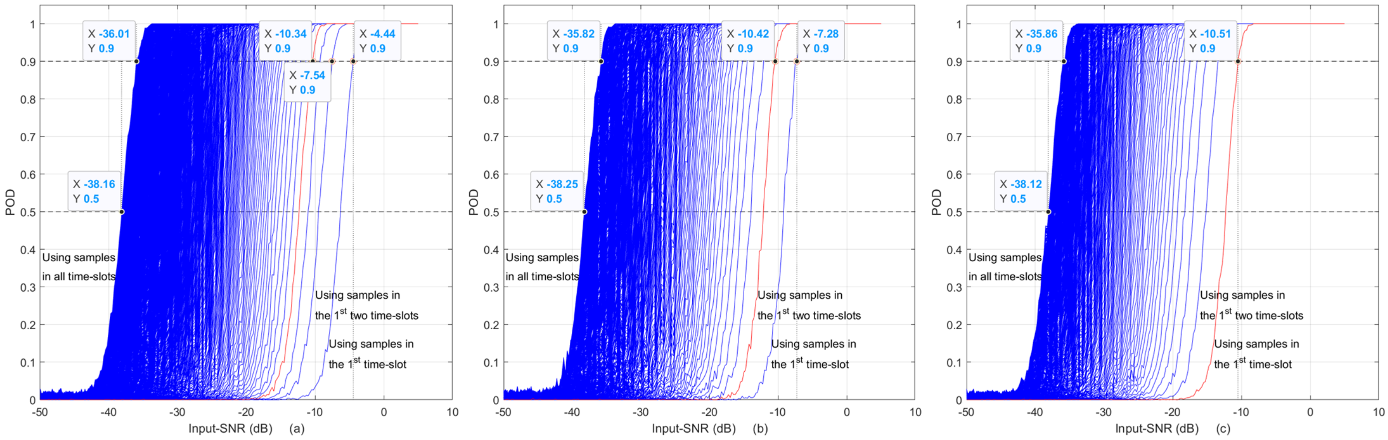

Figure 9. $N = 2^{18}$, the POD (with P_{fas} of 10^{-7}) vs. Input-SNR, in each subplot when only the 1st time-slot samples (**right**) and all time-slots sample (**left**) are used in AIRS, (**a**) $M = 128$, (**b**) $M = 256$ and (**c**) $M = 512$.

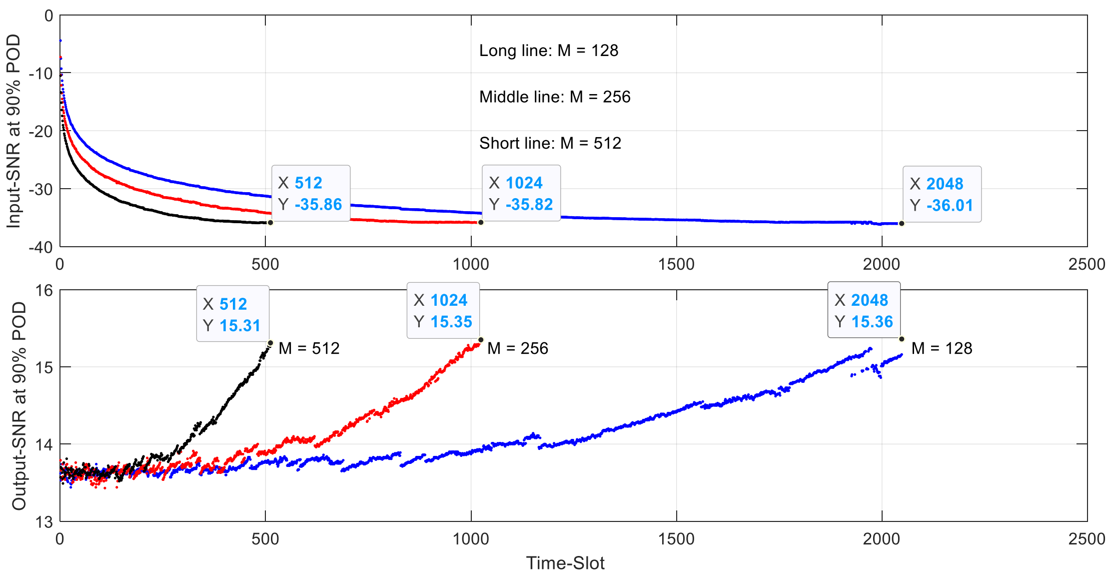

Figure 10. $N = 2^{18}$, Input-SNR and Output-SNR at 90% POD with P_{fas} of 10^{-7}, when different time-slot samples are used in AIRS, blue line: $M = 128$, red line: $M = 256$ and black line: $M = 512$.

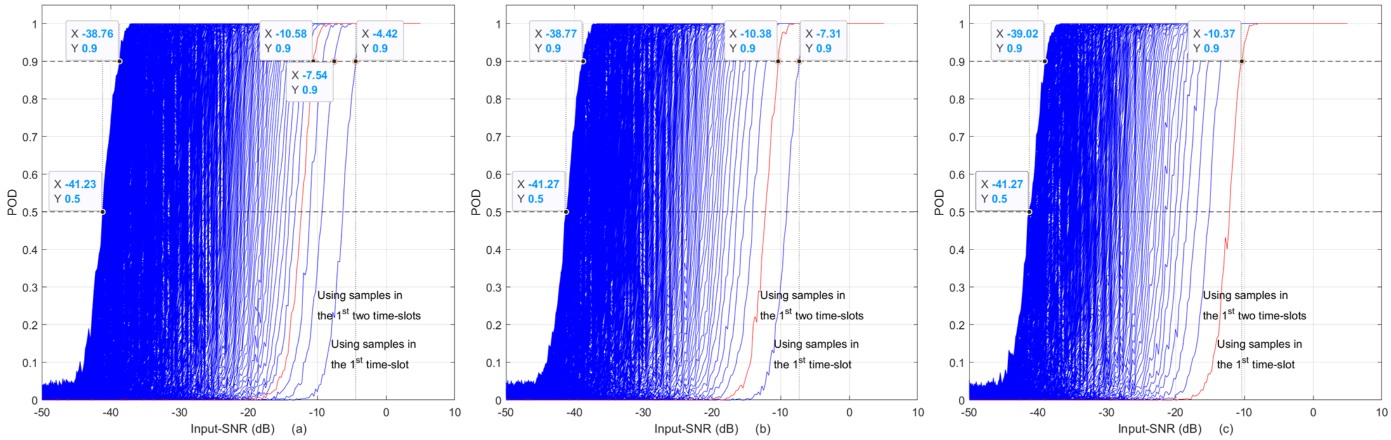

Figure 11. $N = 2^{19}$, the POD (with P_{fas} of 10^{-7}) vs. Input-SNR, in each subplot when only the 1st time-slot samples (**right**) and all time-slots sample (**left**) are used in AIRS, (**a**) $M = 128$, (**b**) $M = 256$ and (**c**) $M = 512$.

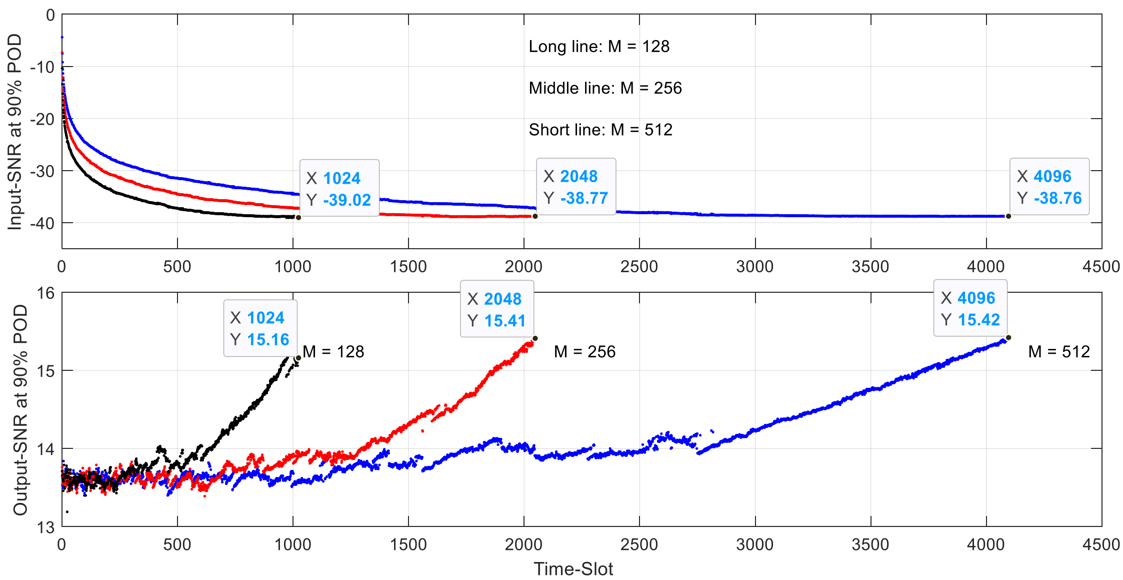

Figure 12. $N = 2^{19}$, Input-SNR and Output-SNR at 90% POD with P_{fas} of 10^{-7}, when different time-slot samples are used in AIRS, blue line: $M = 128$, red line: $M = 256$ and black line: $M = 512$.

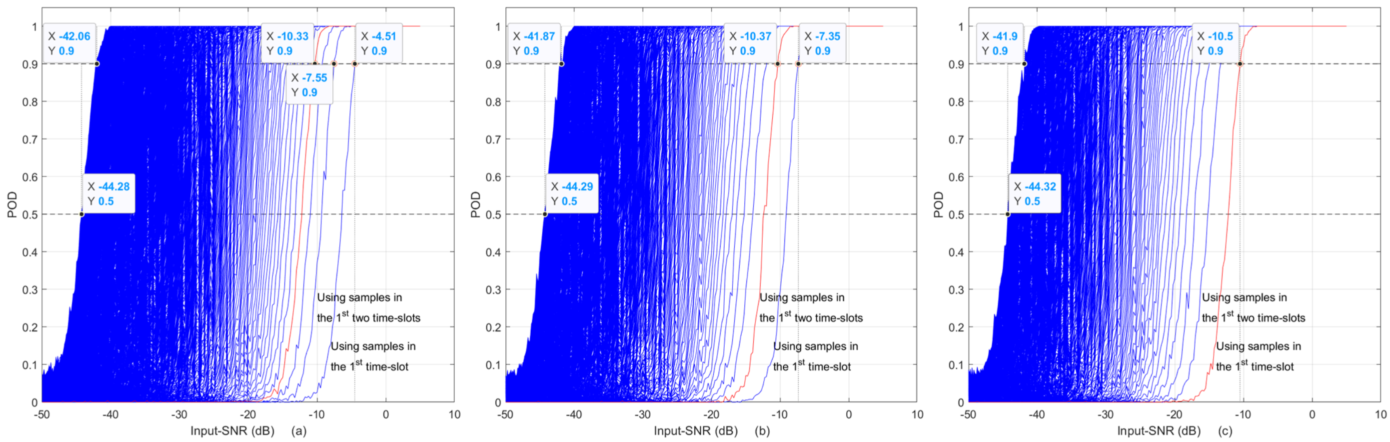

Figure 13. $N = 2^{20}$, the POD (with P_{fas} of 10^{-7}) vs. Input-SNR, in each subplot when only the 1st time-slot samples (**right**) and all time-slots sample (**left**) are used in AIRS, (**a**) $M = 128$, (**b**) $M = 256$ and (**c**) $M = 512$.

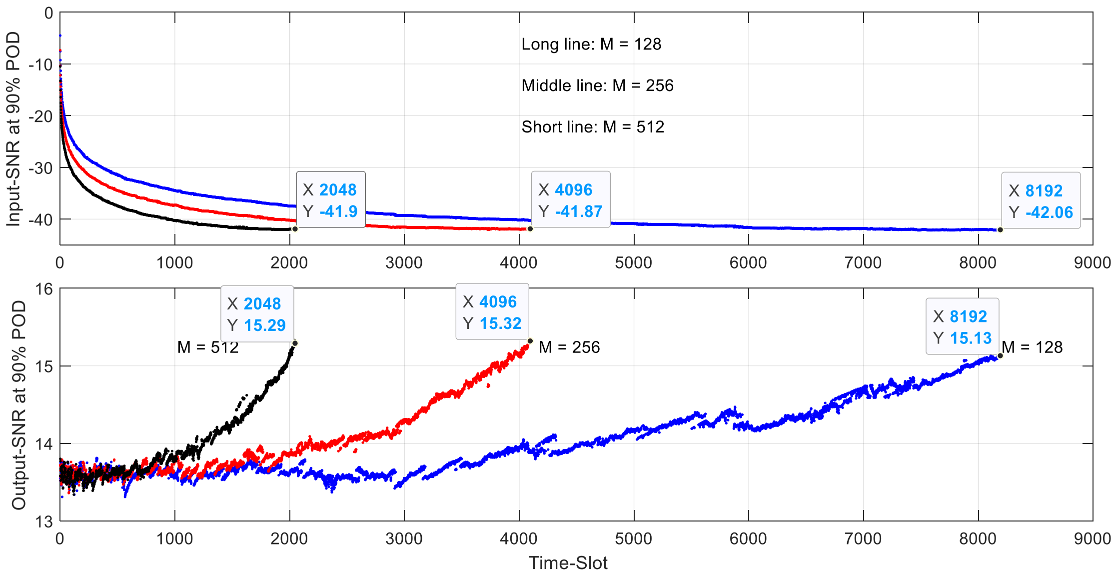

Figure 14. $N = 2^{20}$, Input-SNR and Output-SNR at 90% POD with P_{fas} of 10^{-7}, when different time-slot samples are used in AIRS, blue line: $M = 128$, red line: $M = 256$ and black line: $M = 512$.

Table 8. TMR and FMR for different N and M combinations used in this study with ADC sampling frequency F_s = 2.56 GHz.

	M		
N	128	256	512
2^{17}	50 (nSec) 19,531 (Hz)	100 (nSec) 19,531 (Hz)	200 (nSec) 19,531 (Hz)
2^{18}	50 (nSec) 9765.6 (Hz)	100 (nSec) 9765.6 (Hz)	200 (nSec) 9765.6 (Hz)
2^{19}	50 (nSec) 4882.8 (Hz)	100 (nSec) 4882.8 (Hz)	200 (nSec) 4882.8 (Hz)
2^{20}	50 (nSec) 2441.4 (Hz)	100 (nSec) 2441.4 (Hz)	200 (nSec) 2441.4 (Hz)

In addition, the more samples are used, the lower Input-SNR is required.

6. Narrow and Wideband Signal Detections Using AIRS

In the last section, we discussed the relation between Input-SNR vs. time-slot in AIRS in order to detect a CW signal with 90% POD and P_{fas} of 10^{-7}. In this section, we study the relation between Input-SNR and time-slot that AIRS has the first detection for narrow- and wide-band signals.

6.1. Input-SNR against Time-Slot That Has the First Detection

Using the thresholds given in Section 4, this section shows the weak FMCW signal detections in AIRS. The signals used in this section and their corresponding color codes used in the subsequent figures are given in Table 9. The carriers of these FMCW were randomly picked and ensured that all the frequency components were in between 100 to 1180 MHz.

Table 9. FMBW of FMCW signals, chirp duration or pulse width (PW) is 2 (mSec) and amplitude is 1, F_s is the ADC sampling frequency.

FMBW (Hz)	1	100	1000	F_s/N	10 K	100 K	1 M	10 M	100 M
Colors and line styles in Figures 15 and 16	Black solid	Black solid	Black solid	Black solid	Black solid	Red dashed	Blue dashed	Magenta dashed	Green dashed

Figure 15 shows the Input-SNR versus the averaged time-slot number that has the first detection of the signals given in Table 9, when $N = 2^{20}$ and $P_{fas} = 10^{-7}$. The time-slot numbers in the plots are the average numbers from 100 independent simulations, which means that for a given FMBW 100 FMCW signals are randomly generated in between 100 to 1180 MHz with different Input-SNR values from −47 to −24 dB at 0.1 dB step. From the figure, we have following observations:

1. For a given signal in Table 9, an Input-SNR vs. the time-slot having the first detection curve can be divided into two sections by the value, called Input-SNR$_{\text{peak}}$, on the curve, which is indicated by a solid or dashed vertical line. (More on Input-SNR$_{\text{peak}}$ will be discussed later.)
    - Before the Input-SNRpeak, the required time-slot numbers are increased as the Input-SNR decreases. In this range, the signal-power accumulation is faster than that of the noise. However, we can see that more time-slots are needed to detect lower Input-SNR signals, as the curves are in a concave shape.
    - After the Input-SNRpeak, the required time-slot numbers are quickly reduced. The reason for this is faster accumulation of noise than the signal, and the detections happen only occasionally.
    - When the Input-SNR is less than about −45 dB, there is no detection.
2. For the first 5 narrow band signals in Table 9, shown in black-lines in the figure,
    a. AIRS can detect them at about −42 dB of Input-SNR level. This is the same results given in Table 2 that the minimum required Input-SNR for $N = 2^{20}$ is −41.9 dB for 90% POD with P_{fas} of 10^{-7}.
    b. The required time-slot numbers at Input-SNR$_{\text{peak}}$ are less than $K/2$ for a given M. This shows that the AIRS can start detecting narrow band signal using less than half of the real-samples that N-point FFT needs.
    c. It is obvious that AIRS is much more efficient than N-point FFT to detect signals when Input-SNR is higher than Input-SNR$_{\text{peak}}$. AIRS requires much less real-samples compared to the total bin size and it gives very good TMR (see Table 8) that the traditional FFT with large-bin numbers cannot achieve.
3. For last 4 signals in Table 9, when the FMBW increases the AIRS is less capable of detecting them compared to detecting the narrow band signals. The reason for this will be discussed later. However, AIRS still can quickly detect them at lower than −25 dB Input-SNR with samples from a very small number of time-slots.
4. Comparing the three subplots in Figure 15, the required time-domain samples are about the same at a given Input-SNR. It does not depend on the method of partitioning the samples in a time-slot. This is an anticipated result as the threshold in each time-slot depends only on the total noise samples used in the calculation.

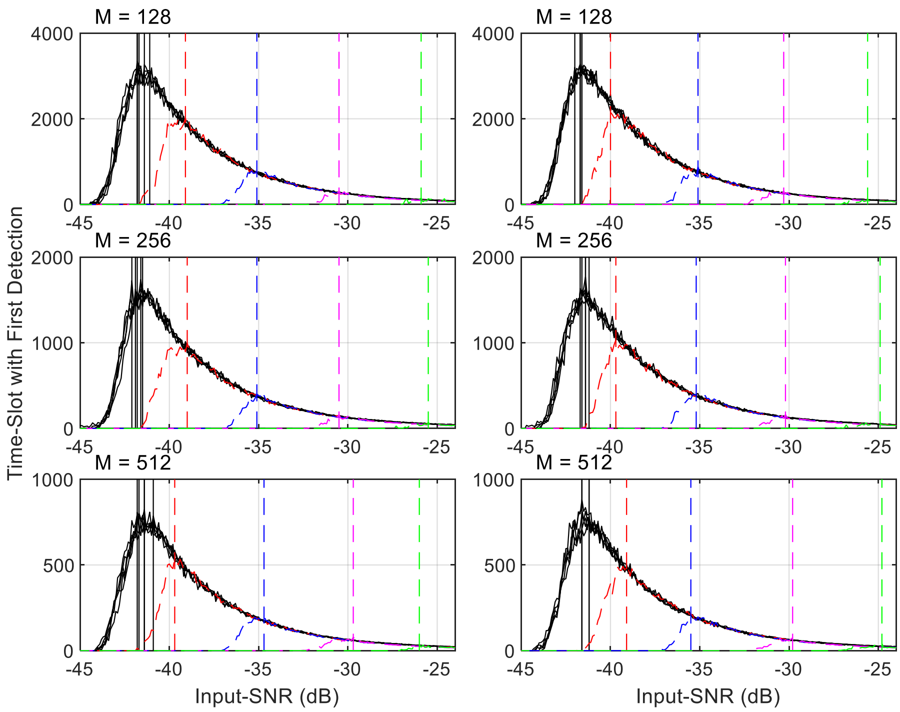

Figure 15. $N = 2^{20}$, the time-slot that had the first detection versus Input-SNR for different FMCW signals (averaged from 100 simulations); left- and right-columns are the results of the positive and negative chirp slopes with the FMBW given in Table 9.

Figures 16–18 show the same results when $N = 2^{19}$, 2^{18} and 2^{17}, respectively. In addition to the same observations from Figure 15, the Input-SNR$_{peak}$ locations move to higher SNR levels as N is reduced for corresponding signals, i.e., each time N is reduced by half, Input-SNR$_{peak}$ moves up by about 3 dB for narrow band signals.

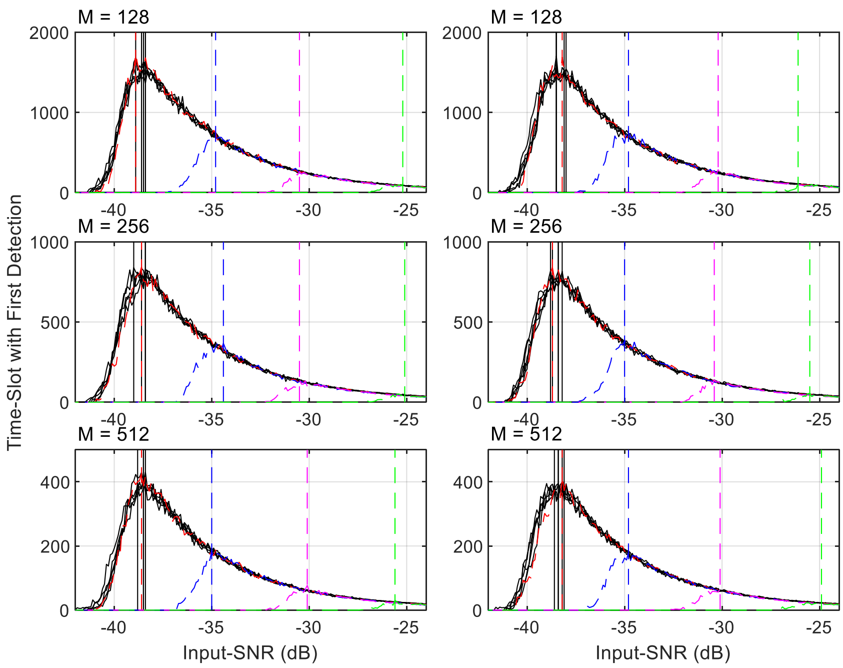

Figure 16. $N = 2^{19}$, the time-slot that had the first detection versus Input-SNR for different FMCW signals (averaged from 100 simulations); left- and right-columns are the results of the positive and negative chirp slopes with the FMBW given in Table 9.

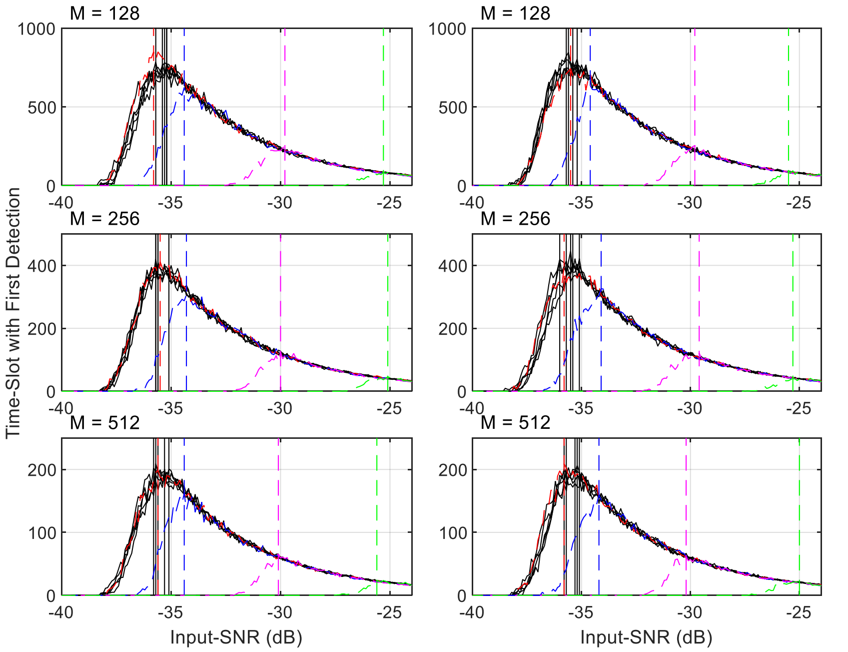

Figure 17. $N = 2^{18}$, the time-slot that had the first detection versus Input-SNR for different FMCW signals (averaged from 100 simulations); left- and right-columns are the results of the positive and negative chirp slopes with the FMBW given in Table 9.

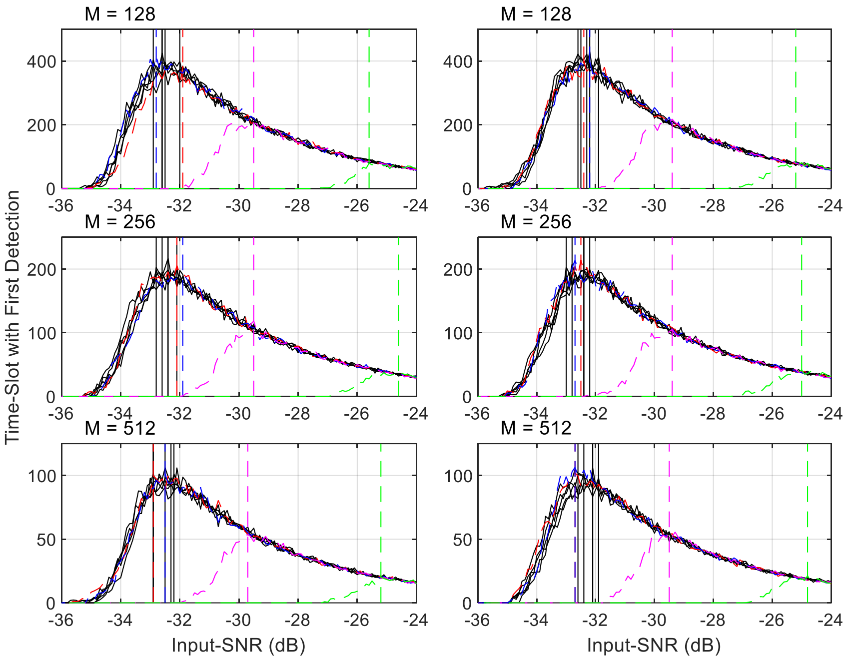

Figure 18. $N = 2^{17}$, the time-slot that had the first detection versus Input-SNR for different FMCW signals (averaged from 100 simulations); left- and right-columns are the results of the positive and negative chirp slopes with the FMBW given in Table 9.

6.2. Explanation of Wideband Signal Detection in AIRS

Using the information in Table 10, let us discuss why AIRS has more difficulty in detecting wideband signals than detecting narrow band signals. The data in the table shows that a narrow-band signal samples in all the time-slots can make contribution to one frequency-bin in AIRS processing, while the data from only about one time-slot can contribute to the same frequency-bin, when FMBW is 100 MHz. This means that, for the positive chirp slope signals having 100 MHz FMBW, if the current time-slot data contain the frequency components that can contribute to a frequency-bin, then the next time-slot samples contribute to the next frequency-bin after AIRS processing.

Table 10. Frequency contents of different linear FMCW signals, when $N = 2^{20}$, $PW = 2$ (mSec), $M = 128$, $K = 8192$, $F_s = 2.56$ GHz, the frequency-bin width = F_s/N = 2441.4 Hz, (BW: Bandwidth).

Linear FMCW FMBW (Hz)	1	100	1000	F_s/N	10 K	100 K	1 M	10 M	100 M
Frequency BW/sample (Hz)	1.95×10^{-7}	1.95×10^{-5}	1.95×10^{-4}	4.77×10^{-4}	1.95×10^{-3}	1.95×10^{-2}	0.195	1.95	19.5
Samples having frequency BW equals to bin width	$1.25 \times 10^{+10}$ $\gg N$	$1.25 \times 10^{+8}$ $\gg N$	$1.25 \times 10^{+7}$ $\gg N$	$5.12 \times 10^{+2}$ $>N$	$1.25 \times 10^{+6}$ $\approx N$	$1.25 \times 10^{+5}$ $<N$	$1.25 \times 10^{+3}$ $\ll N$	1250 $\ll N$	125 $\ll N$
Convert 3rd row data into time-slots	97,656,250 $\gg K$	976,563 $\gg K$	97,656 $\gg K$	40,000 $>K$	9766 $\approx K$	977 $<K$	98 $\ll K$	9.77 $\ll K$	0.98 $\ll K$

6.3. The Input-SNR$_{peak}$ and Output-SNR$_{peak}$ for Narrow-Band Signals

In this section, we summarize the Input-SNR$_{peak}$ for those narrow band signals in each subplot from Figures 15–18. The top plot of Figure 19 shows the Input-SNR$_{peak}$ of the first 4 narrow-band signals in Table 9, when the different N and M combinations were used. The bottom plot of the figure shows the Output-SNR$_{peak}$ related to Input-SNR$_{peak}$ calculated using (15). Again, we can find that:

- When N is doubled the Input-SNR$_{peak}$ reduces 3 dB.
- The Output-SNR$_{peak}$ values are in between 10.75 to 12.3 dB, and the averaged Output-SNR$_{peak}$ is about 11.64 dB.

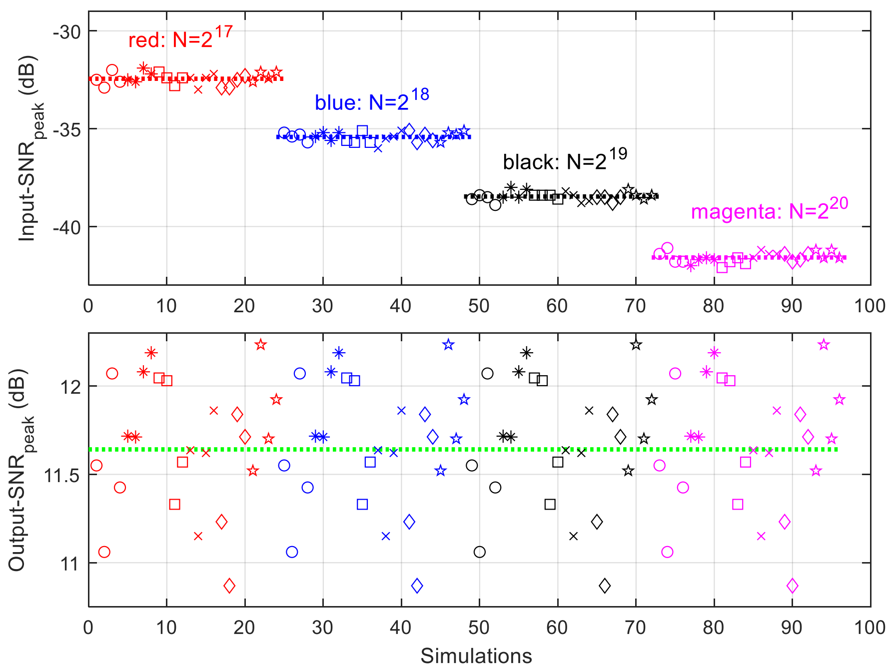

Figure 19. Input-SNR$_{peak}$ and Output-SNR$_{peak}$ of different N and M combinations for the first 4 narrow band signals in Table 9. FMCW with positive chip slopes are shown in circle for $M = 128$, square for $M = 256$ and diamond for $M = 512$. FMCW with negative chip slopes are shown in asterisk for $M = 128$, cross for $M = 256$ and pentagram for $M = 512$.

Note that:

1. The Output-SNR$_{peak}$ discussed here is related to the time-slot that has the first detection of narrow-band signals with Input-SNR$_{peak}$, when P_{fas} of 10^{-7} is considered. The

Output-SNR discussed in Section 5 was to consider 90% POD for CW signal. However, what is important to note is that at the Input-SNR$_{\text{peak}}$ levels given in Table 11, which are about the same levels of Input-SNR in Table 2, AIRS only needs less than $N/2$ real-samples to detect these signals, while FFT needs all the N real-samples. The reason is that AIRS uses the time-slot-based thresholds, which only invokes the noise up to that specific time-slot.

2. We also want to emphasize that the advantage of the AIRS processing is that it can detect signals using the currently available samples for the N frequency-bins without waiting for the availability of all the N samples. This offers a feasible way of hardware implementation when a large number of samples are available from a high-speed ADC.

Table 11. The average values of Input-SNR$_{\text{peak}}$ and Output-SNR$_{\text{peak}}$ for different N.

N	2^{17}	2^{18}	2^{19}	2^{20}
Averaged Input-SNR$_{\text{peak}}$ (dB)	−32.44	−35.42	−38.67	−41.58
Averaged Output-SNR$_{\text{peak}}$ (dB)		11.64		

Figure 20 summarizes the Input-SNR$_{\text{peak}}$ and Output-SNR$_{\text{peak}}$ of different N and M combinations for wideband signals in Table 9.

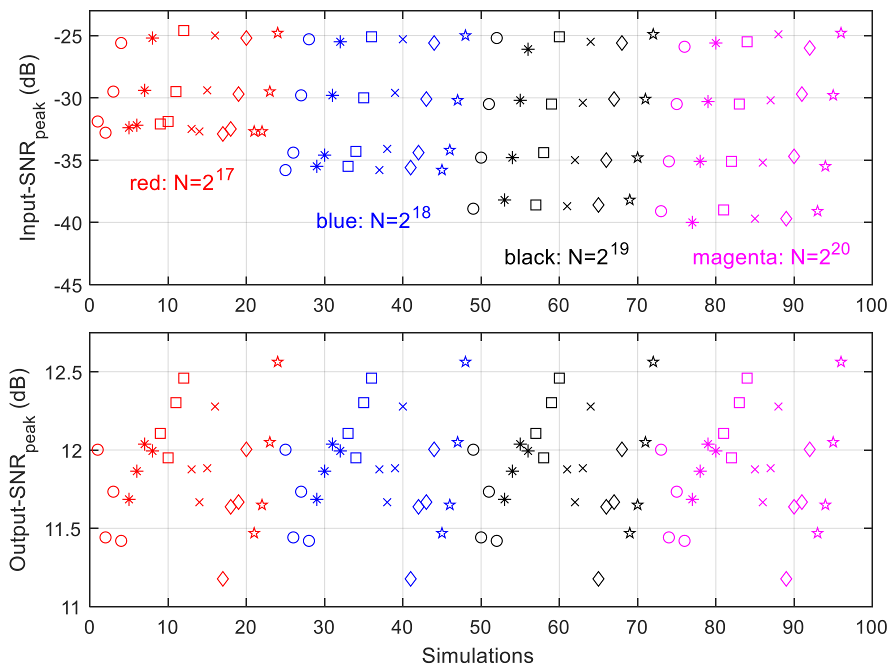

Figure 20. Input-SNR$_{\text{peak}}$ and Output-SNR$_{\text{peak}}$ of different N and M combinations for the last 4 wideband signals in Table 9. FMCW with positive chip slopes are shown in circle for $M = 128$, square for $M = 256$ and diamond for $M = 512$. FMCW with negative chip slopes are shown in asterisk for $M = 128$, cross for $M = 256$ and pentagram for $M = 512$.

7. Conclusions

Based on the BFFT method, this paper introduces the AIRS concept of weak signal detections for an ultra-wideband DRX application for the first time. Unlike the conventional FFT-based wideband DRX design, AIRS-based DRX design can use a very large number of frequency-bins in a given IBW, achieve both fine TMR and FMR, and detect low Input-SNR signals. Using time-slot-based thresholds with P_{fas} of 10^{-7}, we demonstrate that the AIRS requires much fewer time samples compared to the number of frequency-bins, in order to detect narrow-band signals. In the worst case, when SNR is at Input-SNR$_{\text{peak}}$ level, AIRS still only needs the number of samples less than half of the number of frequency-bins. For wide-band signals, simulation results show that AIRS can detect them, when the Input-SNR is even lower than −25 dB with a much lower number of real samples than the number of frequency-bins. This is because a very fine equivalent noise bandwidth can be achieved in AIRS-based DRX.

Author Contributions: Conceptualization, algorithm software, funding and original draft preparation were done by C.W.; concept discussions and reviewing and editing were done by T.T. and J.E.; real-time implementation discussions were done by D.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the research funding of Defence Research and Development Canada-Ottawa Research Centre, Department of National Defence, Canada.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

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

Acknowledgments: Authors would like to thank Pascal Truchon and Jamie Simms for preparing three computer servers with advanced GPUs, which helped us to finish massive simulations and generate the data used in this paper.

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

Appendix A. The Comparison between Results Obtained by the MATLAB FFT Function and the BFFT Code

This appendix shows the comparison between results calculated by our BFFT code and MATLAB FFT function to verify the BFFT code used for this study. The parameters used in the example are shown in Table A1. The comparisons of the frequency spectrum are shown in Figures A1 and A2, and show that these two methods give the same results.

Table A1. The parameters used in the example to compare results.

Figure	Number of Samples	Signal	Duration (mSec)	Chirp Start (MHz)	Carrier End (MHz)	ADC F_s (GHz)	Input-SNR (dB)	BFFT	
								K	M
A1	2^{20}	Chirp	0.4096	350.1	550.1	2.56	100	4096	256
A2	2^{19}	Chirp	0.2048	650.1	860.1	2.56	0	4096	128

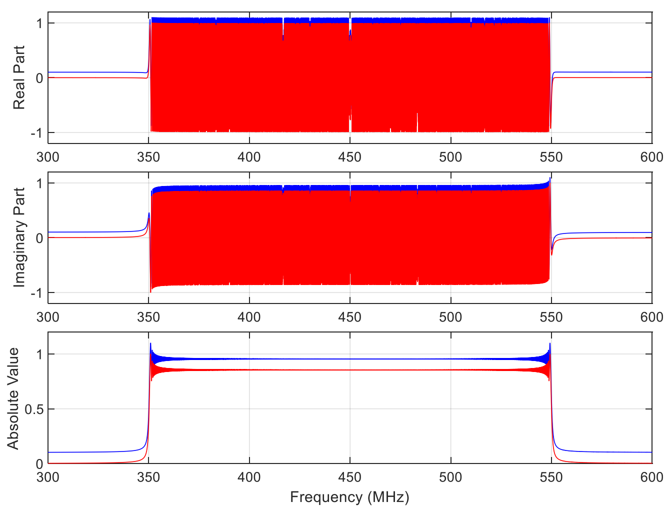

Figure A1. Normalized frequency spectrum obtained from BFFT (red) and MATLAB FFT function (blue). To clearly show the comparison, the blue results are moved up 0.1. The maximum difference between the absolute values of two results is 1.2×10^{-12}.

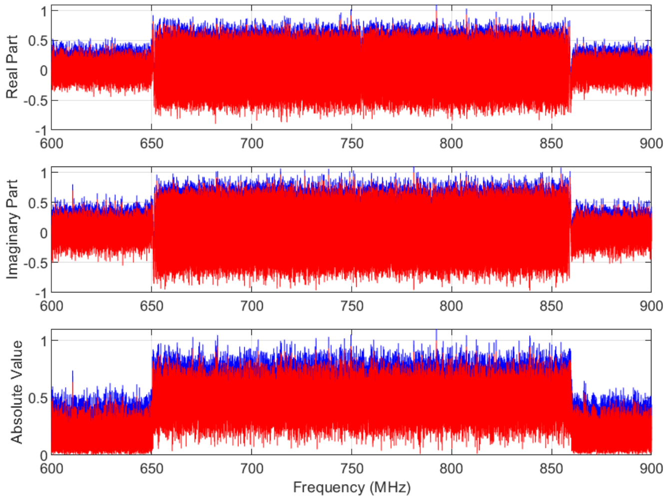

Figure A2. Normalized frequency spectrum obtained from BFFT (red) and MATLAB FFT function (blue). To clearly show the comparison, the blue results are moved up 0.1. The maximum difference between the absolute values of two results is 8.0×10^{-13}.

Appendix B. Acronym List

ADC	Analog-to-Digital Converter
AIRS	Accumulatively Increasing Receiver Sensitivity
BFFT	Blocking FFT
BW	Bandwidth
CW	Continuous Wave
DFT	Discrete Fourier Transform
DRX	Digital Receiver
FFT	Fast Fourier Transform
FMBW	Frequency Modulation Bandwidths
FMCW	Frequency Modulated Continuous Wave
FPGA	Field Programmable Gate A19rray
FMR	Frequency Measurement Resolution
IBW	Instantaneous Bandwidth
Input-SNR	Input Signal-to-Noise Ratio
LIP	Low Probability Intercept
Output-SNR	Output Signal-to-Noise Ratio
POD	Probability of Detection
PW	Pulse Width
TMR	Time-of-arrival Measurement Resolution
TTFT	Time-to-Frequency Transform

References

1. Pace, P.E. *Detecting and Classifying Low Probability of Intercept Radar*, 2nd ed.; Artech House: Norwood, MA, USA, 2009.
2. Boashash, B. *Time-Frequency Signal Analysis and Processing*, 2nd ed.; Academic Press: Oxford, UK, 2016.
3. Haneche, H. Compressed Sensing in Mobile Systems, Networking and Internet Architecture [cs.NI]. Ph.D. Thesis, University of Science and Technology Houari Boumediene, Bab Ezzouar, Algeria, 2020.
4. Haneche, H.; Boudraa, B.; Ouahabi, A. A new way to enhance speech signal based on compressed sensing. *Measurement* **2020**, *151*, 107117. [CrossRef]
5. Wu, C.; Rajan, S. Fast Fourier sampling for ultra-wide band digital receiver applications. In *Proceedings Volume 8753, Wireless Sensing, Localization, and Processing VIII*; International Society for Optics and Photonics: Bellingham, WA, USA, 2013; Volume 8753. [CrossRef]
6. Layne, D. Receiver Sensitivity and Equivalent Noise Bandwidth, High Frequency Electronics, J2014. Available online: https://www.highfrequencyelectronics.com/index.php?option=com_content&view=article&id=553:receiver-sensitivity-and-equivalent-noise-bandwidth&catid=94&Itemid=189 (accessed on 23 October 2021).
7. Green, H.E. A theoretical examination of tangential signal to noise ratio. *IEEE Trans. Microw. Theory Tech.* **1991**, *39*, 566–567. [CrossRef]

8. Packard, H. The Criterion for the Tangential Sensitivity Measurement, Application Note 956-1. 1990. Available online: http://www.hp.woodshot.com/hprfhelp/4_downld/lit/diodelit/an956-1.pdf (accessed on 25 September 2021).
9. Tsui, J.B.Y. *Special Design Topics in Digital Wideband Receivers*; Artech House: Norwood, MA, USA, 2010.
10. Neri, F. *Introduction to Electronic Defence Systems*, 3rd ed.; Artech House: Norwood, MA, USA, 2018.
11. Hedge, J.; Tsui, J.B.Y.; Sharpin, D. Sensitivity of digital electronic warfare receivers. *IEEE Conf. Aerosp. Electron.* **1990**, *3*, 981–984. [CrossRef]
12. Xilinx, Zynq UltraScale+ RFSoc, ZCU208 Evaluation Board User Guide. Available online: https://www.xilinx.com/content/dam/xilinx/support/documentation/boards_and_kits/zcu208/ug1410-zcu208-eval-bd.pdf (accessed on 6 September 2021).
13. Annino, B. SFDR Considerations in Multi-Octave Wideband Digital Receivers, Analog Dialogue, Volume 55, No. 1, January 2021. Available online: https://www.analog.com/media/en/analog-dialogue/volume-55/number-1/sfdr-considerations-in-multi-octave-wideband-digital-receivers.pdf (accessed on 6 September 2021).
14. Cheng, C.; Lin, D.M.; Liou, L.L.; Tsui, J.B. Electronic Warfare Receiver with Multiple FFT Frame Sizes. *IEEE Trans. Aerosp. Electron. Syst.* **2012**, *48*, 3318–3330. [CrossRef]
15. Kanders, H.; Mellqvist, T. One Million-Point FFT. Master's Thesis, Electrical Engineering Department of Electrical Engineering, Linköping University, Linköping, Sweden, 2018.
16. Li, Y.; Chen, H.; Xie, Y. An FPGA-based four-channel 128k-point FFT processor suitable for space-borne SAR. *Electronics* **2021**, *10*, 816. [CrossRef]
17. Agarwal, A.; Hassanieh, H.; Abari, O.; Hamed, E.; Katabi, D.; Arvind. High-throughput implementation of a million-point sparse Fourier Transform. In Proceedings of the 2014 24th International Conference on Field Programmable Logic and Applications (FPL), Munich, Germany, 2–4 September 2014; pp. 1–6. [CrossRef]
18. Xu, P.K.; Xu, F.Y. A real-time spectral analysis method and its FPGA implementation for long-sequence signals. *Meas. Sci. Technol.* **2020**, *31*, 035006. [CrossRef]
19. Naval Air Warfare Center Weapon Division. *Electronic Warfare and Radar Systems Engineering Handbook, NAWCWD TP 8347*, 4th ed.; Naval Air Warfare Center Weapon Division: Point Mugu, CA, USA, 2013.
20. Wu, C.; Krishnasamy, D.; Elangage, J. Hardware Implementation and RF High-Fidelity Modeling and Simulation of Compressive Sensing Based 2D Angle-of-Arrival Measurement System for 2–18 GHz Radar Electronic Support Measures. *Sensors* **2021**, *21*, 6823. [CrossRef] [PubMed]
21. Richardson, L. The Sliding Window Discrete Fourier Transform. Ph.D. Thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 2019. [CrossRef]
22. Viswanathan, M. Chi Square Distribution–Demystified, GaussianWaves: Signal Processing for Communication Systems, 11 September 2022. Available online: https://www.gaussianwaves.com/2012/09/chi-squared-distribution (accessed on 16 January 2022).