Received: 20 July 2023 | Revised: 31 August 2023 | Accepted: 3 September 2023

DOI: 10.1049/rsn2.12471

IET Radar, Sonar & Navigation

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH

Implementation of a coherent real-time noise radar system

Martin Ankel[1,2] | Mats Tholén[3,4] | Tomas Bryllert[1,2] | Lars M. H. Ulander[5] | Per Delsing[1]

[1]Department of Microtechnology and Nanoscience, Chalmers University of Technology, Göteborg, Sweden

[2]New Concepts and System Studies, Surveillance, Saab, Göteborg, Sweden

[3]Nanostructure Physics, KTH Royal Institute of Technology, Stockholm, Sweden

[4]Intermodulation Products AB, Segersta, Sweden

[5]Department of Space, Earth and Environment, Chalmers University of Technology, Göteborg, Sweden

Correspondence

Martin Ankel.
Email: ankel@chalmers.se

Funding information

Knut and Alice Wallenberg, Grant/Award Number: 2017.0449

Abstract

The utilisation of continuous random waveforms for radar, that is, noise radar, has been extensively studied as a candidate for low probability of intercept operation. However, compared with the more traditional pulse-Doppler radar, noise radar systems are significantly more complicated to implement, which is likely why few systems exist. If noise radar systems are to see the light of day, system design, implementation, limitations etc., must be investigated. Therefore, the authors examine and detail the implementation of a real-time noise radar system on a field programmable gate array. The system is capable of operating with 100% duty cycle, 200 MHz bandwidth, and 268 ms integration time while processing a range of about 8.5 km. Additionally, the system can perform real-time moving target compensation to reduce cell migration. System performance is primarily limited by the memory bandwidth of the off-chip dynamic random access memory.

KEYWORDS

correlation methods, CW radar, digital signal processing chips, field programmable gate arrays, LPI radar, pseudonoise codes, radar signal processing

1 | INTRODUCTION

Using noise-like waveforms for radar detection was proposed in the later part of the 1950s [1] to mitigate range and Doppler ambiguities. Nowadays, the ongoing noise radar research is motivated mainly by the expectation that random high bandwidth waveforms will provide low probability of intercept/identification (LPI/LPID) properties [2–4]. Whether this is the case or not can be debated [5–7]. Regardless, noise radars have several other benefits that make them worthwhile investigating [8–12], such as low probability of exploitation (for example, the radar mode is challenging to deduce), low mutual interference, robustness against intelligent jamming, favourable ambiguity function etc.

Despite all these advantages, no operational systems are in use (to the authors' knowledge). The primary reasons for this are the masking effect [13–15] (see Section 2.1) and the relatively complicated design and implementation associated with noise radar systems. Noise radar performs detection by cross-correlating the received signal with a delayed copy of the transmitted signal. Early proof-of-concept systems [16, 17] performed analogue correlation processing, which requires one microwave delay line for every range resolution cell of interest. Achieving a good range coverage and range resolution, thus, constitutes a highly complex system.

The solution to this is digital correlation processing [18, 19]. In the digital domain, each range resolution cell of interest can be processed in parallel. However, digital implementation

requires significant computational resources and high-speed data converters to accommodate high bandwidths. It is only recently digital systems demonstrating different aspects of real-time noise radar operation have been presented [20–25].

A notable and recent advancement in the context of digital electronics is the Zynq UltraScale+ RFSoC, a radio-frequency system on a chip, integrating radio frequency (RF) data converters, many cores of central-processing units (CPU) and a large field programmable gate array (FPGA). The RFSoC platform on-chip integration of high-speed data converters enables direct digital synthesis (DDS) of RF waveforms. Multiple RF data converters also make the platform suitable for multiple-input and multiple-output (MIMO) noise radar, potentially enhancing LPI properties [26].

In this paper, we report on the construction and implementation of a monostatic L-band (1.3 GHz) noise radar system based on a digital microwave platform called the Vivace [27], in which the main component is the first generation Xilinx RFSoC hardware [28]. The system can perform continuous range and Doppler processing with 100% duty cycle while operating with a bandwidth of 200 MHz and processing a range of 8.5 km. When operating with 200 MHz bandwidth, the integration time is limited to about 268 ms by the off-chip dynamic random access memory (DRAM), yielding a time-bandwidth (TB) product of roughly 77 dB. At such high TB products, range migration becomes significant. Therefore, a real-time moving target compensation algorithm has been implemented [29–31].

A video showing the system in action while performing real-time detection of a small UAV—with an estimated radar cross section (RCS) of 0.01 m^2—is provided in the supplementary information S1. Because we operate a monostatic noise radar, the systems' detection range for the UAV is restricted to around 100 m by the masking effect. For targets with greater RCS, the detection range will increase, but for most targets, it will still be limited to hundreds of metres. Nevertheless, the demonstration shows that digital electronics have advanced to the point that digital high-bandwidth real-time noise radar processing is no longer in the future. Pairing the developed system with a bistatic transmitter [32] and performing spatial filtering [33] can extend the detection range to several kilometres - this will be implemented and investigated in the future.

The choice in operating frequency is purely due to practical reasons such as available equipment, direct sample synthesising, transmission permit etc., and the frequency chosen is not necessarily optimal for noise radar operation. Other operating frequencies could be considered in the future by including analog mixers in the system design.

The structure of the paper is as follows. Section 2 covers the theory relevant to the real-time system and comments on aspects relating to hardware implementation. Section 3 provides a detailed description of the complete and qualified system. The measurement results are presented in Section 4 and an overall discussion is given in Section 5. Lastly, Section 6 provides the conclusions.

2 | THEORY

This section presents the theory of the different processing steps implemented in the real-time processor, including range processing, velocity processing—which uses a batched implementation—and a hardware-efficient moving target compensation algorithm [29–31].

2.1 | Range calculation—Fast time processing

Assume that the noise radar continuously transmits band-limited pseudo-random noise, exhibiting both amplitude and phase fluctuations. Let $x(t)$ represent the baseband modulation of the continuous reference signal and $x_n = x(n/f_s)$ the corresponding discrete samples, where f_s is the baseband sampling rate. Detection is performed by cross-correlating the received signal y_n with the conjugated reference signal x_n,

$$R[k] = \sum_n y_n x^*_{n-k}. \tag{1}$$

A correlation detector contains the function $|R[k]|^2$, exhibiting peaks at indices corresponding to strong correlations between the signal and the reference, indicating a reflection corresponding to that delay.

Unfortunately, the cross-correlation of the noise waveform also produces a noise floor, which we will refer to as the correlation noise floor (CNF), also known as the masking effect [13–15]. The CNF, relative to the strongest scatterer, closely follows the TB product of the waveform. Therefore, operating with a high TB product is favourable to reduce the effect of strong echoes masking weaker ones.

The cross-correlation calculation in Equation (1) is effectively implemented in the frequency domain [34], utilising the Fast Fourier Transform (FFT). Let the vectors $\vec{x} = (x_0, x_1, \ldots, x_{N-1})^\top$ and $\vec{y} = (y_0, y_1, \ldots, y_{N-1})^\top$ collect a sequence of N samples, where N is the length of the coherent processing interval (CPI). Then the cross-correlation is calculated as

$$\vec{R} = \mathrm{IFFT}[\mathrm{FFT}(\vec{y}) \odot \mathrm{FFT}(\vec{x}^*)], \tag{2}$$

where $\odot$ refers to the Hadamard product, this implementation calculates the circular cross-correlation. Often the linear cross-correlation is preferred; in that case, the vectors $\vec{x}$ and $\vec{y}$ have to be zero-padded. Zero-padding doubles the data rate, leading to an increase in, for example, digital signal processor (DSP) and block random access memory (BRAM) resources.

Implementing Equation (2) in hardware is relatively simple. Most hardware description languages (HDL) have predefined FFT routines available. For example, Vivado design suite has the *Xilinx® LogiCORE™ intellectual property (IP) FFT* [35], which efficiently can perform an FFT of maximum $2^{16} = 65{,}536$ samples. Now, with a baseband sampling rate of 250 MS/s, an FFT of 2^{16} samples would only correspond to an integration time

of roughly 260 μs. Increasing the integration time requires either incoherent integration or batched processing. Batched processing [30, 36] is the preferred alternative, and its implementation will be detailed in the following subsection.

2.2 | Velocity calculation using batched processing—Slow time processing

In offline processing, the velocity is often determined by calculating the cross-ambiguity function (CAF) [9, 12], as

$$R[k,v] = \sum_n y_n x^*_{n-k} \cdot e^{2\pi i(2vf_c/c)\cdot\frac{n}{f_s}}, \tag{3}$$

where v is the velocity, c is the speed of light, and f_c is the carrier frequency. The received signal is cross-correlated with a Doppler-shifted reference signal for all velocities of interest. Calculating the CAF in real-time limits the coherent integration time to 260 μs and would require more computational resources than is available in a single state-of-the-art FPGA. Instead, the velocity is calculated using a batched implementation, illustrated in Figure 1.

The idea behind batched processing is to calculate the range-Doppler map similarly to the pulse-Doppler radar. The two signals $\vec{y}$ and $\vec{x}$ consisting of N samples are segmented into P batches, each containing $M = N/P$ samples. The vectors corresponding to batch p is denoted as $\vec{y}_p$ and $\vec{x}_p$. Fast time processing, range processing, is now performed by cross-correlating $\vec{y}_p$ with $\vec{x}_p$, using Equation (2). Slow time processing, Doppler processing, is performed by calculating the inter-batch phase variation, realised by taking the FFT over all batches for every range resolution cell. In the FPGA, slow time processing requires intermediate data storage in, for example, DRAM.

Batched processing results in an effective pulse repetition interval (PRI), PRI $= M/f_s$, resulting in Doppler ambiguities.

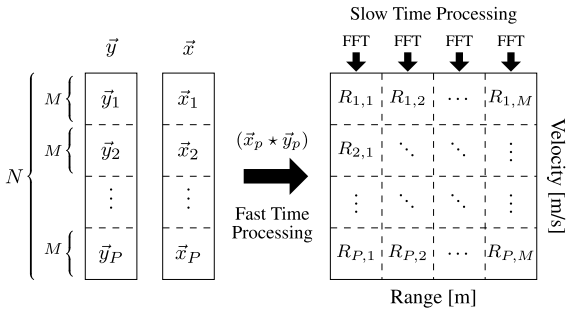

FIGURE 1 Illustration of batched operation, inspired by pulse-Doppler radar. The received signal $\vec{y}$ and the reference signal $\vec{x}$ are split into P batches, where each batch contains M samples. Each respective batch pair is then cross-correlated using Equation (2), that is, $\vec{R}_p = \left(\vec{x}_p \star \vec{y}_p\right) = \mathrm{IFFT}\left[\mathrm{FFT}\left(\vec{y}_p\right) \odot \mathrm{FFT}\left(\vec{x}^*_p\right)\right]$, this gives the fast time information. The Doppler information is retrieved by performing the Fast Fourier Transform (FFT) over each range cell, calculating the inter-batch phase variation. This illustration calculates the circular correlation. The linear correlation is calculated by zero-padding $\vec{x}_p$ and $\vec{y}_p$.

However, a noise radar can operate with arbitrary batch lengths, allowing for parallel processing with different PRIs to resolve the Doppler ambiguities.

2.3 | Moving target compensation

High TB products lead to significant cell migration of moving targets. As an example, a system operating with a bandwidth of 200 MHz, a baseband sampling rate of 250 MS/s and a CPI of 268 ms will suffer a signal-to-interference-plus-noise-ratio (SINR) loss of about 12 dB for a target moving with a velocity of 10 m/s [30].

The target's movement can be accounted for in the correlation processing by resampling the reference waveform, a method referred to as stretch processing [37]. However, resampling is unnecessary if the distance covered between two batches is negligible with respect to the range resolution. It is enough to perform time translation between batches, that is, in slow time [29–31].

The authors have previously investigated how to compensate for target movement in slow time. See ref. [31] for a detailed analysis and experimental results. The idea is to perform time translation between batches, utilising the Fourier transform property that $\mathcal{F}[f(n-a)] = \mathcal{F}[f(n)]e^{-\frac{2\pi i}{M}qa}$. Between batches, the reference is shifted with the factor $a = 2v_rM/c$ to compensate for target motion, as

$$\mathbf{X}'_{p,q} = \mathbf{X}_{p,q} e^{-2\pi i q\cdot p\frac{2v_r}{c}}, \tag{4}$$

where $\mathbf{X}_{p,q}$ is the Fourier transform of $\vec{x}_p$ and v_r is the reference velocity. Considering the FFT of $\vec{x}_p$ is already calculated in the batched processing, this algorithm requires a minimum of additional computations. Only two additional operations are required: calculating one phase factor for each sample and one element-wise multiplication. In FPGA terms, this would consist of a cosine block, a sine block, product blocks, counters and some logic gates. The algorithm is thus easy to implement, resource efficient and practical, considering data does not have to be buffered. Therefore, it is suitable for real-time implementation.

In a real scenario, the target velocity is unknown, and for good use of moving target compensation, parallel or sequential processing of multiple v_r hypotheses is necessary. Luckily, there are ways to factorise the computations. See, for example, reference [38].

3 | REAL-TIME Noise RADAR SYSTEM

In this section, everything relating to the hardware implementation of the real-time system is detailed, including the FPGA platform, FPGA model and synthesises, FPGA resource utilisation, analogue hardware etc.

3.1 | Vivace—Microwave synthesis and analysis platform

Vivace [27] is a microwave synthesis and analysis platform for frequencies up to 4 GHz. The platform comes with infrastructure for system initialisation and data transfer, and for this application, only a measurement core is added.

Vivace is based on the first generation Xilinx RFSoC hardware [28], intended for software designed 5G radio. Therefore, it comes with 8 digital-to-analogue converters (DACs) running at up to 6.4 GS/s, 8 analogue-to-digital converters (ADCs) running at up to 4 GS/s, low phase-noise clocking infrastructure and significant FPGA, CPU and memory resources. These features make the platform well-suited for implementing a noise radar demonstrator. The FPGA synthesises and analyses the radar signal in real-time, and the high rate of the DACs and ADCs enables DDS at the carrier frequency of 1.3 GHz.

The analogue bandwidth of the inputs and outputs of the RFSoC and Vivace is high compared to the sampling rate. With proper reconstruction and anti-aliasing filters, signals in the second Nyquist zone can be directly synthesised and measured, enabling carrier frequencies up to 4 GHz.

3.2 | Measurement core—Simulink noise radar model

The noise radar processor, the FPGA measurement core, was developed in the model-based design tool Simulink [39]. Simulink allows for fast and efficient FPGA implementation at the cost of reduced functionality compared to other hardware description language (HDL) tools, such as Vivado.

A simplified block diagram of the noise radar model is shown in Figure 2. The different functions of the model can roughly be divided into 1) Signal Generation, Transmission and Reception, 2) Fast Time Processing, 3) DRAM Operation, and 4) Slow Time Processing and Data Readout. The implementation of each function is detailed below. The model was synthesised to a very high-speed integrated circuit hardware description language IP core using Mathworks HDL coder [40], and the synthesised IP core operates with a clock frequency of 250 MHz.

Most of the model parameters are configurable but not run-time configurable. The only run-time configurable parameters are the threshold and the velocity, which are changeable at any time. Whereas the batch length and the number of pulses, for example, can only be changed between CPIs and logic to handle this is then required. Changing the batch length and the number of pulses requires rebooting the system, which only takes a few seconds. How choice in parameters affects the radar's mode of operation is illustrated in Table 1.

3.2.1 | Signal generation, transmission and reception

Two linear-feedback shift registers (LFSRs) [41, 42], one for real and one for imaginary numbers, generate 16-bit packed samples to create a complex baseband signal. A configurable finite impulse response (FIR) filter sets the preferred signal bandwidth. The system can operate with any bandwidth below 250 MHz by changing the filter coefficients. Currently, the

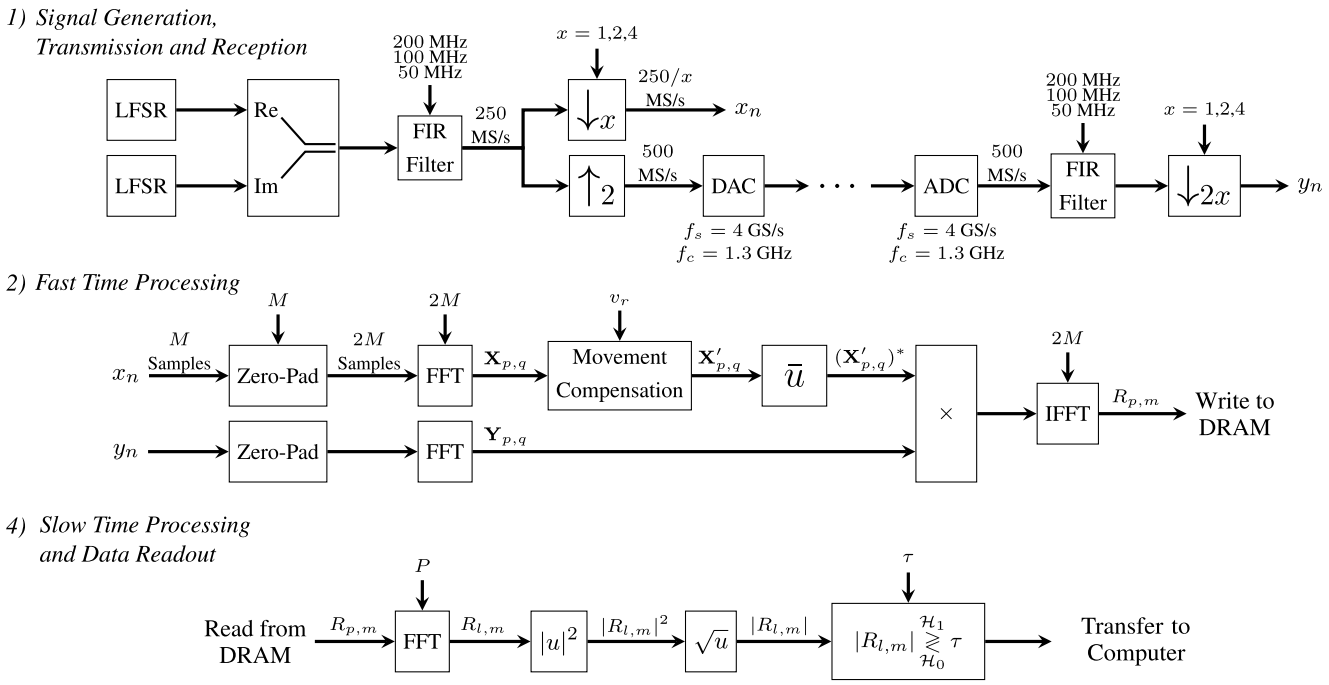

FIGURE 2 Block diagram of the field programmable gate array based real-time processor. Boxes with arrows entering at the top indicate that the parameter is configurable. Each processing step is explained in detail in Section 3.2.

TABLE 1 Different configurations of the parameters M, P and D.

Configuration	M	P	D	B [MHz]	R_{res} [m]	R_{pro} [km]	PRI [μs]	V_{una} [m/s]	T_{int} [ms]	V_{res} [m/s]	TB [dB]
A	16,384	4096	1	200	0.75	8.5	65.5	880	268.4	0.43	77.3
B	16,384	4096	2	100	1.5	17	131	440	536.9	0.21	77.3
C	16,384	4096	4	50	3	34	262	220	1073.7	0.11	77.3
D	16,384	2048	1	200	0.75	8.5	65.5	880	134.2	0.86	74.3
E	16,384	2048	2	100	1.5	17	131	440	268.4	0.43	74.3
F	16,384	2048	4	50	3	34	262	220	536.9	0.21	74.3
G	8192	4096	1	200	0.75	8.5	32.8	1760	134.2	0.86	74.3
H	8192	4096	2	100	1.5	17	65.5	880	268.4	0.43	74.3
I	8192	4096	4	50	3	34	131	440	536.9	0.21	74.3
J	8192	2048	1	200	0.75	8.5	32.8	1760	57.1	1.72	71.3
K	8192	2048	2	100	1.5	17	65.5	880	134.2	0.86	71.3
L	8192	2048	4	50	3	34	131	440	268.4	0.43	71.3

Note: The resulting bandwidth (B), range resolution (R_{res}), range processed (R_{pro}), pulse repetition interval (PRI), unambiguous velocity (V_{una}), integration time (T_{int}), velocity resolution (V_{res}) and time-bandwidth product (TB). The value for R_{cov} is valid under the assumption that the radar always operates with 100 % duty cycle. If the duty cycle is lowered, R_{cov} can be increased, see Section 3.2.3) for more information.

system uses three different sets of coefficients, resulting in bandwidths of 200 MHz, 100 MHz and 50 MHz. The frequency response for the different sets of filter coefficients is shown in Figure 3.

Depending on the bandwidth chosen, the filtered signal is downsampled with either a factor 1, 2 or 4, resulting in sampling rates of 250 MS/s, 125 MS/s or 62.5 MS/s respectively. It is the downsampled signal that makes up the reference signal x_n.

The minimum allowed sampling rate of the signals connected to the DAC and ADC is 500 MS/s. Therefore, the filtered signal is upsampled by a factor of 2. It is then upsampled and upconverted by the DAC to a sample rate of 4 GS/s and a carrier frequency of 1.3 GHz. Similar to the DAC, the ADC samples the received signal directly at the carrier frequency, downconverts to complex baseband and downsamples to a sample rate of 500 MS/s. The signal is then FIR filtered and further downsampled to the correct sampling rate, this constitutes the received signal y_n.

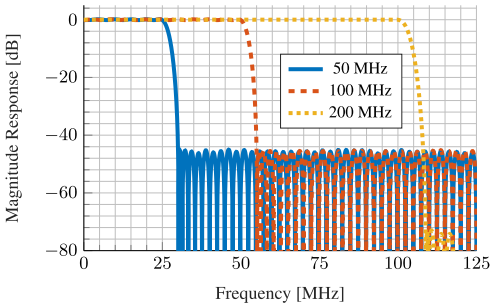

FIGURE 3 Frequency response for the three different sets of finite impulse response filter coefficients. Since the input is a complex baseband pseudorandom noise signal, the resulting output is psuedorandom noise with a bandwidth of either 50 MHz, 100 MHz or 200 MHz.

3.2.2 | Fast Time Processing

To calculate the linear cross-correlation, x_n and y_n are zero-padded with a factor of 2, doubling the sampling rate, before the FFT of the signal and reference is calculated. Both the zero-padding and the FFT can be configured to process batches of either $M = 16{,}384$ or $M = 8192$. However, due to limitations in Simulink, two FFTs of different lengths have to run in parallel for the processor to have a configurable batch length. This results in unnecessary computations.

After the reference has been Fourier transformed, the moving target compensation algorithm, detailed in Section 2.3, is applied. The product of the conjugated moving target compensated reference $\left(\mathbf{X}'_{p,q}\right)^*$ and the Fourier transformed signal $\mathbf{Y}_{p,q}$ is then calculated. Lastly, the product is inverse Fourier transformed, and the result is saved to the off-chip DRAM, concluding the fast time processing.

Preferably the FFT length should be run-time configurable, which it is for the *Xilinx® LogiCORE™ IP FFT*. Compared to a fixed-length FFT, *Xilinx's* run-time configurable FFT only requires slightly more resources and, if implemented, would significantly improve the current system. It would not only reduce the overall resource usage, but it would also increase the number of available batch lengths.

3.2.3 | DRAM Operation

For this implementation, the main limitation for achieving high TB real-time processing is the read-and-write data transfer rate

of the DRAM. The bit-depth of the samples saved to DRAM is 32-bits for each I and Q sample—see Section 3.4 for a short discussion on the required bit depth. After zero-padding, the maximum sampling rate is 500 MS/s, resulting in a data transfer rate of 4 GB/s. Since the DRAM both have to write and read, the effective data rate is 8 GB/s.

The Vivace FPGA logic has access to 4 GB DDR4-2666 memory, with a sequential throughput of 21 GB/s. However, in this application, data can be viewed as a matrix written as rows but read as columns. From the DRAM's point of view, this sequence can be considered as random access, leading to a performance of about 700 MB/s or 3% of the sequential rates—more than an order of magnitude less than required.

A DRAM matrix transpose is implemented to increase the data rate. First, data is written to the DRAM in bursts of 128 samples. Data is then read from the DRAM as 128 × 128 sample sub-matrices, transposed using true random access memory (BRAM inside the FPGA) and written back to the DRAM [43]. Lastly, the transposed data is read from DRAM and sent to the Doppler processing subsystem. Using this method, all read and write operations are performed as bursts of 128 samples that are contiguous in the DRAM, increasing the effective throughput to about 3.5 GB/s—still not high enough.

Two solutions have been implemented to resolve the problem of insufficient DRAM data rate. One is implementing write-first priority and adding a guard period between CPIs, allowing the DRAM to catch up before processing the next CPI. Hence, the system does not continuously process data when operating at 200 MHz. The system still transmits continuously, but the signal processing operates at a duty cycle of around 43% to stay within the maximum effective throughput of 3.5 GB/s.

Another way to reduce the effective throughput is to discard samples. If, for example, only a fraction of the range interval processed is of interest, samples corresponding to other ranges can be neglected and do not have to be saved to the DRAM. Many samples are used to achieve a high TB product, but some distances are excluded before the Doppler processing, this is not optimal. Typically, the entire processed range interval will be of interest. However, the option of discarding samples still exists. Roughly 57% of the range samples must be discarded to stay within the maximum throughput of the DRAM, resulting in a processed range of 8.5 km when operating with 200 MHz bandwidth.

3.2.4 | Slow Time Processing and Data Readout

The velocity is calculated by taking the FFT for each range cell over all batches. This procedure is also configurable, and the number of batches that can be integrated is $p = 2048$ or $p = 4096$. Similar to the fast time processing, two FFTs are running in parallel.

In order to transfer the resulting data to the computer and for the computer to visualise the results in real-time, the data rate has to be drastically lowered before readout. Therefore, a variant of run-length encoding (RLE) is implemented. The RLE implementation determines whether the squared value of a resolution cell, $|R_{l,m}|^2$, is above or below a specified threshold τ. Values above the threshold will be read out, whereas values below the threshold will be discarded, and instead, a counter is raised by one. The RLE operation can be summarised as

$$|R_{l,m}|^2 \gtrless_{\mathcal{H}_0}^{\mathcal{H}_1} \tau, \quad \begin{cases} \mathcal{H}_1: & \text{Readout} \\ \mathcal{H}_0: & \text{Raise Counter} \end{cases} \tag{5}$$

The counter keeps track of the number of consecutive resolution cells that fall below the threshold. Hence, a single sample can represent multiple resolution cells that have fallen below the threshold. The most significant bit indicates whether the sample represents a detection or the number of zeros.

Since we assume that most resolution cells will fall below the threshold, the RLE operation will drastically reduce the data rate. The threshold can either be adaptively configured or user specified. If adaptively configured, the FPGA will continuously change the threshold to keep the data rate within the user-specified value. The maximum allowed data rate when the computer collects, unpacks and visualises the data is about 25 MB/s.

3.3 | Matlab user interface and data visualisation

The noise radar system is initialised and controlled by a Matlab user interface. Data processed by the FPGA is transferred directly to a computer via Ethernet. The computer collects the data, unpacks it (decodes the RLE sequence) and plots a range-Doppler map. These operations are performed in Matlab and could be better optimised. With the current computer, data rates of roughly 25 MB/s can be handled continuously, limited by the unpacking operation. The threshold and velocity are run-time configured using a graphical user interface.

3.4 | Bit depth requirements

The bit depth at each processing step is essential to the implementation. It can be done in two ways: the bit depth is designed concerning signal-to-noise ratio (SNR) requirements or concerning the TB product. Considering the CNF will always be roughly the TB product below the strongest scatterer, for noise waveforms, a bit depth much larger than the TB product is unnecessary. For a TB product of 77.3 dB, 13 bit will be sufficient. Of course, some guard bits are eligible.

We have chosen to design based on SNR requirements because that makes it more versatile if, for example, other types of waveforms are of interest. But this results in a much higher bit depth and, thus, higher computational resources and data rates. Especially as high TB product and continuous wave operation are two factors that drastically increase the required

bit depth. Also, waveforms with high and random amplitude modulations demand extra guard bits, as the response of the FIR filters is unknown beforehand.

The ADC has a dynamic range of 12 bits, always assumed to be maximised. For a thorough discussion about the effect the ADC bit depth has on the receiver's performance, see [36]. After filtering and downsampling from 4 GHz to 62.5 MHz, the required bit depth is 15 bits. Performing the matched filter for $M = 2^{14} = 16{,}384$ increases the bits needed by 7. The Doppler FFT with $p = 2^{12} = 4096$ further increases the required bits by 6, meaning the minimum required bit depth to avoid SNR losses is 28. With 4 guard bits, the final bit depth is 32 bits - roughly double the number of bits compared to only designing for noise waveforms.

The primary resources of the FFT routines are DSPs. One DSP can perform the multiplication of two 27-bit samples. For 28-bit samples, two DSPs are required to perform multiplication. Hence, 27-bits is a crucial number, and the FPGA functions should preferably operate with 27-bits or less. Currently, both the IFFT and Doppler FFT use more than 27-bits.

3.5 | FPGA resource utilisation

The percentage of available FPGA resources used is shown in Figure 4. There are several ways to optimise the model, lowering the resources used. Most have already been discussed. The main limitation is, however, the memory bandwidth between the off-chip DRAM and the FPGA. Increasing the memory bandwidth is necessary to achieve higher throughputs. Increased read and write burst sizes from 128 to, for example, 256 will likely only yield a modest improvement. Better off-chip memories are required. Graphics processing units (GPUs) have reported theoretical memory bandwidths ranging up to 2 TB/s [44], making GPU an exciting alternative, especially as GPUs also have high processing capabilities. However, high bandwidth memories intended for FPGA platforms are also being produced [45]. The effective memory bandwidth of such memories when performing radar-specific operations, such as range-Doppler processing, remains to be seen. A high-speed radar processing design will likely consist of a mix of GPUs, CPUs, and FPGAs - see references [46, 47] for more details.

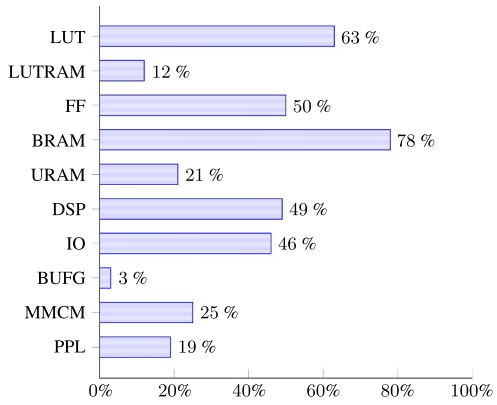

FIGURE 4 Field programmable gate array resources utilised by the current implementation. These resources include look-up table (LUT), LUT random access memory (LUTRAM), flip flops (FF) block random access memory (BRAM), ultra random access memory (URAM), digital signal processor (DSP), inputs and outputs (IO), global clock buffer (BUFG), mixed-mode clock manager (MMCM) and phase-locked loop (PLL).

3.6 | Analogue hardware

Because we operate a monostatic continuous wave system, we have significant short-range ground clutter returns and direct signal leakage, which restricts the relation between receiver gain and transmitter output power.

The ADC has a maximum input power of 6 dB and an input noise power density of −147 dBm/Hz. Preferably the receiver gain should amplify the thermal noise (−174 dBm/Hz) to the extent that the ADC's noise contribution becomes negligible. The constructed receiver has a total gain of 31.5 dB, a noise figure (NF) of 2 and 2 dB losses. The ADC noise will, therefore, contribute 1.3 dB to the NF, giving a total NF of 3.3 dB. The quantisation noise of the ADC is negligible.

The total signal return depends on the clutter environment, choice of antennas, positioning and tilting of the antennas etc. Typically, we measure signal returns around −60 dB of the transmitted power, and then to not saturate the receiver, the maximum allowed output power is about 2.5 W. However, the CNF will limit the detection range, and the output power can be lowered several orders of magnitude without affecting the detection performance. In the performed measurements, the transmitted signal was band-pass filtered and amplified to a power of about 10 mW. The radar parameters are presented in Table 2, and based on these, an SNR of around 15 dB is expected at a distance of about 550 m when integrating for 268 ms. Shorter detection ranges would suggest that the system is clutter limited, not noise limited.

The radar system and the measurement setup can be seen in Figure 5. It consists of the analogue receiver, a laptop, the Vivace, a DJI Matrice 600 UAV, a car battery to power the electronics and two 2 × 2 element antennas - with an estimated directivity gain of 10.8 dB - one for transmission and one for

TABLE 2 Radar parameters.

Transmitted power	P_T	10 mW
Antenna gain	G	10.8 dB
Wavelength	λ	0.23 m
Integration time	τ_I	57.1–1073.7 ms
Target RCS	σ	0.01 m^2
Reference temperature	T_0	290 K
Noise Figure	F	3.3 dB
Compound loss	L	6 dB

reception. The analogue receiver is prepared for 8-channel reception to utilise all of Vivace's available ADCs in the future.

The system is monostatic in the sense that the separation between the two antennas is small and, therefore, the system operates with monostatic characteristics, including that the receiver and the transmitter are both disciplined by the same reference oscillator, leading to excellent synchronisation in time and frequency. This type of configuration can sometimes be referred to as quasi-monostatic [34]. In bistatic systems, the transmitter and the receiver are separated by a considerable distance, resulting in the radar operating with more complicated characteristics and synchronisation between the receiver and transmitter is difficult to achieve as the two are disciplined by different reference oscillators.

4 | MEASUREMENTS—UAV DETECTION

Measurements on a UAV were performed to verify and demonstrate the performance of the constructed real-time system. A video showing the system performing real-time detection of the UAV is provided in the supplementary information S1. Only configuration A (see Table 1) is shown for this demonstration. The plot update rate was set to 2 Hz, and the threshold was set relatively high to have few false alarms.

The UAV flies past the radar with a constant speed of 10 m/s. A signal loss of roughly 12 dB is expected without any moving target compensation—see Section 2.3. Hence, an SINR increase of 12 dB is expected when the moving target compensation is applied. Figure 6 shows that when the moving target compensation is applied, the SINR is increased significantly, and the target signal is less widened in range and Doppler. On average, the increase in SINR is about 7 dB. Why the improvement is not better is uncertain, and it is not easy to analyse without access to the raw data. The effect of applying moving target compensation is also shown in the supplementary video S1, where it is seen that moving target compensation results in an extended detection range.

The CNF makes it complicated to conduct experiments. It requires the UAV to fly very close to the radar, which in itself is difficult, but also results in short measurement times and gives problems with aspect angles—aspect angles are the reason why the UAV appears to be initially accelerating in the video. The complications resulted in the UAV batteries running low before a suitable antenna placement, fly path etc., had been determined, which led to few measurements, only one radar configuration, and the UAV, unfortunately, flying too short.

FIGURE 5 The measurement setup includes the Vivace, the analogue receiver, transmitter and receiver antennas, a laptop, the target UAV, and a car battery to power the electronics. The analogue transmitter consisted of a band-pass filter and a low-noise amplifier. The analogue receiver is prepared for 8-channel reception to utilise all of Vivace's available ADCs in the future.

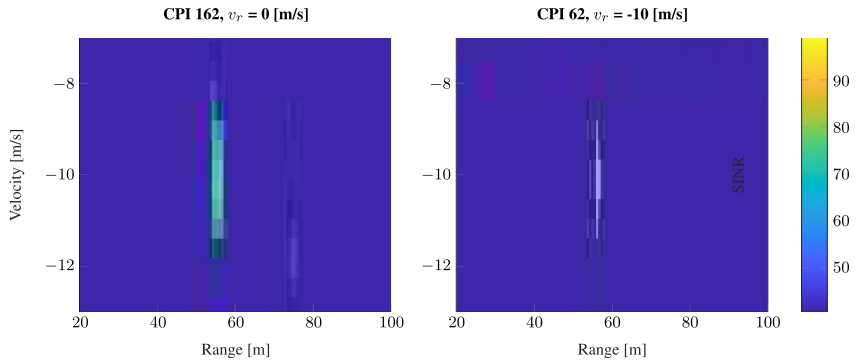

FIGURE 6 Range-Doppler snapshots of the real-time measurement results. (**Left**) No moving target compensation is applied, that is, $v_r = 0$ m/s. (**Right**) Moving target compensation applied for $v_r = -10$ m/s. Applying moving target compensation results in a signal increase of 7 dB, and less widening of the target in range and Doppler, compared to no movement compensation.

The maximum detection range was likely slightly longer than 100 m. However, the UAV decelerates too early, see the video, this was initially not known because the CNF limited the detection range to approximately 60 m—before a more optimal antenna placement was found. For experiments to be conducted more effectively, the system must be improved to be less limited by the CNF—see Section 6.

5 | DISCUSSION

The results demonstrate that the constructed system is capable of real-time detection and that the moving target compensation improves the SINR. However, the system has a short detection range due to the CNF, limiting the usefulness of the system. To realise a proper system, the CNF must be reduced. There are different ways to achieve this, where two common approaches are waveform design [11, 12, 48–51]—the waveform is altered to achieve a lower CNF—and clutter suppression [13, 52–57], where the idea is to estimate the reflection coefficient of strong scatters and then subtract their contribution to the received signal. However, more than these methods are required for many applications. Not only are the suggested methods insufficient, but many of the proposed algorithms are also computationally expensive and likely unsuitable for real-time implementation.

The miniaturisation of electronic components has progressed to the point that the components are approaching physical limitations, making further miniaturisation impossible [58]. Thus, unless completely new devices are invented, Moore's law will soon no longer hold, and the prospect of noise radar implementation can not necessarily rely on the expectation that better hardware will solve the problems. It is essential to start considering how noise radar systems should be implemented, where an important consideration regarding implementation is cost. Considerable computational resources will increase the cost. Monostatic noise radar systems require complicated antenna architecture to cancel the direct signal leakage, further increasing the cost while still leaving the system vulnerable to interference from close-range ground clutter. Additionally, it is highly unlikely that noise radar systems will replace the pulse-Doppler radar, at least in the foreseeable future. Digital radar systems practically have software-defined signal processing and can, therefore, process arbitrary waveforms. Ideally, such systems should be able to operate both pulsed and continuously, but the analogue electronics and physical system architecture are often widely different for the two. Constructing a radar for both pulsed and continuous operation would likely surge the costs, and having two completely separate systems is not a particularly attractive solution.

It is the authors' opinion that monostatic noise radar systems suffer extensively from the CNF and very expensive implementations, preventing the implementation of such systems. However, these problems can likely be overcome by considering a bistatic system. We primarily envision noise radar operation as a compliment to the traditional pulse-Doppler radar in the form of a bistatic noise radar mode of operation. Since digital radar systems can be considered software-defined, digital pulse-Doppler systems should have no problem operating as receivers in conjunction with a bistatic noise transmitter. The analogue electronics and system architecture of pulsed receivers and continuous receivers are basically the same. Therefore, the bistatic implementation does not require the pulse-Doppler architecture to change in any significant way, nor are any abilities sacrificed. The bistatic transmitter is an add-on feature that can likely be constructed at a low cost.

Other advantages of bistatic operations are (I) The direct signal and the clutter interference are reduced thanks to the naturally high isolation achieved by separating the receiver and transmitter antennas. This results in a lower CNF and decreased dynamic range requirements, without increasing the computational burden, (II) Spatial filtering can be applied to mitigate interference from direct signal and strong clutter scatters, (III) Low-cost bistatic transmitters can be expandable, allowing for the receiver to stay silent. The receiver is, thus, protected and many resources can be invested to make it highly capable, (IV) Additionally, if the transmitters are low cost, many can be constructed and operated simultaneously, which might be necessary in order to achieve LPI/LPID, (V) Potential for dual use-case, a bistatic noise transmitter can simultaneously operate both as a barrage jammer and as a radar illuminator, (VI) Clutter filtering and waveform design can still be applied if necessary.

In order to facilitate bistatic or multistatic operation, the reference has to be generated deterministically, for example, using LFSRs as in the system presented. Electronic support measures (ESM) systems tasked with intercepting a noise radar will employ the correlation receiver, where two (or more) receiver channels are cross-correlated [59]. The correlation receiver will perform incoherent integration if the radar signal is below the ESM receiver's thermal noise. But if the LFSRs sequence is known or guessed, the ESM receiver can integrate coherently. Meaning pseudorandom noise has a theoretical disadvantage compared to actual thermal noise.

The maximum length sequence of a l-stage LFSR has a repetition interval of $2^l - 1$ samples. Suitable choices of l make the repetition interval irrelevant, but an ESM receiver only requires $2l - 1$ samples to break the sequence due to the linear properties of the generator [41]. However, a radar signal can not be compared with a telecommunications signal, where all the bits are error corrected. Aside from the actual signal, the radar signal will contain a fair amount of noise, interference, non-linearities, and other disturbances and breaking the sequence will likely be difficult. Additionally, several pseudo-random sequences can be combined non-linearly, and the sequences used can be frequently altered. Furthermore, angle-dependent waveforms can be utilised, that is, MIMO, which makes it harder for multiple distributed ESM system to coordinate their efforts. Moreover, having multiple transmitters further significantly complicates the situation for the ESM systems.

Thanks to Gallium nitride semiconductor technology advancements, it is possible to fabricate small, high-powered

amplifiers. For example, Qorvo's QPD1006 [60] can operate continuously with an output power in the hundreds of Watts. Only a few of these chips are required to achieve adequate output power, and several transmitters can, thus, be built small and at a low cost. The antenna gain would inevitably be low, and the task of providing the required angular resolution will fall on the receiver. The low gain will also result in wide illumination, resulting in low spectral power density.

Digital receivers can perform parallel beamforming and processing. Hence, the reduction in SNR due to low transmitter gain can be compensated for by simultaneously processing the entire illuminated volume and extending the integration time. To what extent transmitter antenna gain can be replaced with longer integration times will depend on the coherence time of the target, something that requires much investigation. Additionally, moving target compensation algorithms might have to be implemented, depending on the bandwidth.

Noise radar operation is, in many regards, limited by the available processing power. A system capable of bistatic operation, spatial filtering, parallel beamforming, moving target compensation, operating with high bandwidth, clutter filtering, waveform design, and MIMO all at once would be very capable. However, implementing a system performing all the mentioned tasks is, likely, not feasible currently, and priorities between different processing steps are necessary. The authors believe that the best way to protect the radar system is to operate low-cost and expandable transmitters, therefore, we advocate for prioritising parallel beamforming and moving target compensation. The radar should still be able to perform all the above-mentioned tasks, considering a digital radar is essentially one massive toolbox, applying the right tools for the task.

6 | CONCLUSIONS AND FUTURE IMPROVEMENTS

A complete real-time noise radar system has been constructed and demonstrated. The systems processing throughput is limited by the off-chip memory bandwidth. Other types of off-chip memory or platforms must be considered to achieve significant improvements. The CNF limits the systems detection range to about 100 m. Future work will aim to reduce the CNF by constructing a real-time bistatic system, thereby increasing the detection range significantly.

To construct a real-time bistatic system, the receiver will be expanded to use Vivace's 8 available ADCs to allow for digital beamforming. Considering the system has a fractional bandwidth of about 19% when operating at the highest possible bandwidth of 250 MHz, true time delay might have to be considered. The improved receiver will then operate together with a bistatic noise transmitter, similar to the one described in reference [32] but with increased output power. If successful, the new bistatic system will be able to perform real-time detection of a small UAV at a distance of several kilometres. Thus, genuinely demonstrating a capable and proper noise radar system.

AUTHOR CONTRIBUTIONS

Martin Par Ankel: Conceptualisation; data curation; formal analysis; investigation; methodology; project administration; software; supervision; validation; visualisation; writing—original draft; writing—review and editing. **Mats Tholén**: Data curation; formal analysis; investigation; methodology; software; validation; visualisation; writing—original draft; writing—review and editing. **Tomas Bryllert**: Data curation; investigation; project administration; supervision; validation; writing—original draft; writing—review and editing. **Lars Ulander**: Project administration; supervision; writing—original draft; writing—review and editing. **Per Delsing**: Conceptualisation; funding acquisition; project administration; resources; supervision; writing—original draft; writing—review and editing.

ACKNOWLEDGEMENTS

The authors acknowledge support from the Knut and Alice Wallenberg (KAW) Foundation through the Wallenberg Centre for Quantum Technology. (WACQT). The authors also thank Robert Jonsson and Stefan Eilertsen for rewarding discussions and insightful comments. Funding Information. Knut and Alice Wallenberg Foundation, Grant/Award Number: KAW 2017.0449.

CONFLICT OF INTEREST STATEMENT

We have no conflicts of interest to disclose.

DATA AVAILABILITY STATEMENT

Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

ORCID

Martin Ankel https://orcid.org/0000-0003-2259-4500

REFERENCES

1. Horton, B.M.: Noise-modulated distance measuring systems. Proc. IRE 47(5), 821–828 (1959). https://doi.org/10.1109/jrproc.1959.287275
2. Pace, P.E.: Detecting and Classifying Low Probability of Intercept Radar, 2nd ed. Artech house (2009)
3. Thayaparan, T., Daković, M., Stanković, L.: Mutual interference and low probability of interception capabilities of noise radar. IET Radar, Sonar Navig. 2(4), 294–305 (2008). https://doi.org/10.1049/iet-rsn:20070146
4. Galati, G., et al.: Counter-interception and counter-exploitation features of noise radar technology. Rem. Sens. 13(22), (2021). https://doi.org/10.3390/rs13224509
5. Galati, G., Pavan, G.: Measuring the anti-intercept features of noise radar waveforms: the way ahead. In: 2022 IEEE 9th International Workshop on Metrology for AeroSpace (MetroAeroSpace), pp. 174–178. Pisa (2022)
6. Galati, G., Pavan, G., Wasserzier, C.: Interception of continuous-emission noise radars transmitting different waveform configurations. In: 2022 23rd International Radar Symposium (IRS), pp. 153–158. Gdansk (2022)
7. Heino, M., et al.: Theoretical and experimental analysis of the supposed stealthiness of noise radar. In: 2023 IEEE Radar Conference (RadarConf23), pp. 1–6. San Antonio (2023)

8. Dawood, M., Narayanan, R.M.: Generalised wideband ambiguity function of a coherent ultrawideband random noise radar. IEE Proc. - Radar, Sonar Navig. 150(5), 379–386 (2003). https://doi.org/10.1049/ip-rsn:20030702
9. Axelsson, S.R.J.: Noise radar using random phase and frequency modulation. IEEE Trans. Geosci. Rem. Sens. 42(11), 2370–2384 (2004). https://doi.org/10.1109/tgrs.2004.834589
10. Savci, K., et al.: Noise radar—overview and recent developments. IEEE Aero. Electron. Syst. Mag. 35(9), 8–20 (2020). https://doi.org/10.1109/maes.2020.2990591
11. De.Palo, F., et al.: Introduction to noise radar and its waveforms. Sensors 20(18), (2020). https://doi.org/10.3390/s20185187
12. Galati, G., Pavan, G., Wasserzier, C.: Signal design and processing for noise radar. EURASIP J. Appl. Signal Process. 2022(52), (2022). https://doi.org/10.1186/s13634-022-00884-1
13. Kulpa, K.S., Czekała, Z.: Masking effect and its removal in PCL radar. IEE Proc. - Radar, Sonar Navig. 3, 174–178 (2005). https://doi.org/10.1049/ip-rsn:20045026
14. Axelsson, S.R.J.: Analysis of random step frequency radar and comparison with experiments. IEEE Trans. Geosci. Rem. Sens. 45(4), 890–904 (2007). https://doi.org/10.1109/tgrs.2006.888865
15. Stec, B., Susek, W.: Theory and measurement of signal-to-noise ratio in continuous-wave noise radar. Sensors 18(5), (2018). https://doi.org/10.3390/s18051445
16. Craig, S.E., Fishbein, W., Rittenbach, O.E.: Continuous-wave radar with high range resolution and unambiguous velocity determination. IRE Transactions on Military Electronics(2), 153–161 (1962). **MIL-6**. https://doi.org/10.1109/iret-mil.1962.5008422
17. Chadwick, R.B.: Measurement of distributed targets with the random signal radar. IEEE Trans. Aero. Electron. Syst.(6), 743–750 (1972). **AES-8**. https://doi.org/10.1109/taes.1972.309604
18. Forrest, J.R., Price, D.J.: Digital correlation for noise radar systems. Electron. Lett. 14(18), 581–582 (1978). https://doi.org/10.1049/el:19780390
19. Wasserzier, C., Galati, G.: On the efficient computation of range and Doppler data in noise radar. International Journal of Microwave and Wireless Technologies 11(7), 584–592 (2019). https://doi.org/10.1017/s175907871800171x
20. Narayanan, R.M.: Through-wall radar imaging using uwb noise waveforms. J. Franklin Inst. 345(6), 659–678 (2008). https://doi.org/10.1016/j.jfranklin.2008.03.004
21. Chen, P.H., et al.: A portable real-time digital noise radar system for through-the-wall imaging. IEEE Trans. Geosci. Rem. Sens. 50(10), 4123–4134 (2012). https://doi.org/10.1109/tgrs.2012.2188411
22. Savci, K., Erdogan, A.Y., Gulum, T.O.: Software defined l-band noise radar demonstrator. In: 2016 17th International Radar Symposium (IRS), pp. 1–4. Poland (2016)
23. Lukin, K.A., et al.: Noise radar design based on fpga technology: onboard digital waveform generation and real-time correlation processing. In: 2017 18th International Radar Symposium (IRS), pp. 1–7. Prague (2017)
24. Barreto, A., et al.: FPGA design and implementation of a real-time FM/PM pseudo random waveform generation for noise radars. In: 2020 IEEE Radar Conference (RadarConf20), pp. 1–6. Florence (2020)
25. Kim, E., et al.: A wideband noise radar system using a phased array with true time delay. Rem. Sens. 14(18), (2022). https://doi.org/10.3390/rs14184489
26. Li, J., Stoica, P.: MIMO Radar Signal Processing. John Wiley and Sons (2008)
27. Intermodulation Products AB: Intermodulation Products – Vivace. Intermodulation Products AB, Segersta. Accessed 17 May 2023
28. Xilinx, Inc.: ZCU111 Evaluation Board User Guide (UG1271) v1.4. Xilinx, Inc, San Jose. Accessed 17 May 2023
29. Shan, T., et al.: Efficient architecture and hardware implementation of coherent integration processor for digital video broadcast-based passive bistatic radar. IET Radar, Sonar Navig. 10(1), 97–106 (2016). https://doi.org/10.1049/iet-rsn.2015.0006
30. Bok, D., O'Hagan, D., Knott, P.: Effects of movement for high time-bandwidths in batched pulse compression range-Doppler radar. Sensors 21(7), (2021). https://doi.org/10.3390/s21072492
31. Ankel, M., et al.: Experimental evaluation of moving target compensation in high time-bandwidth noise radar. In: 2023 20th European Radar Conference (EuRAD). IEEE, Berlin (2023). accepted for publication
32. Ankel, M., et al.: Bistatic noise radar: demonstration of correlation noise suppression. IET Radar, Sonar Navig. 17(3), 351–361 (2023). https://doi.org/10.1049/rsn2.12345
33. Li, J., Stoica, P.: Robust Adaptive Beamforming. John Wiley and Sons (2005)
34. Skolnik, M.I.: Radar Handbook, 3rd ed. McGraw-Hill Education (2008)
35. Xilinx, Inc.: Fast Fourier Transform v9.1 LogiCORE IP Product Guide (PG109). Xilinx, Inc, San Jose. Accessed 17 May 2023
36. Axelsson, S.R.J.: Noise radar for range/Doppler processing and digital beamforming using low-bit ADC. IEEE Trans. Geosci. Rem. Sens. 41(12), 2703–2720 (2003). https://doi.org/10.1109/tgrs.2003.816665
37. Kulpa, K.S., Misiurewicz, J.: Stretch processing for long integration time passive covert radar. In: 2006 CIE International Conference on Radar, pp. 1–4. Shanghai (2006)
38. Ulander, L.M.H., Hellsten, H., Stenström, G.: Synthetic-aperture radar processing using fast factorized back-projection. IEEE Trans. Aero. Electron. Syst. 39(3), 760–776 (2003). https://doi.org/10.1109/taes.2003.1238734
39. The MathWorks Inc.: Simulink® User's Guide (R2022b). The MathWorks Inc, Natick. Accessed 17 May 2023
40. The MathWorks Inc.: HDL CODERTM User's Guide (R2022b). The MathWorks Inc., Natick. Accessed 17 May 2023
41. Proakis, J.G., Salehi, M.: Digital Communications, 5th ed. McGraw-hill, New York (2007)
42. The MathWorks Inc: PN Sequence Generator. The MathWorks Inc, Natick. Accessed 14 August 2023
43. Frigo, M., et al.: Cache-oblivious algorithms. In: 40th Annual Symposium on Foundations of Computer Science (Cat. No. 99CB37039), pp. 285–297. New York (1999)
44. NVIDIA Corporation: NVIDIA A100 TENSOR CORE GPU. NVIDIA Corporatio, Santa Clara. Accessed 25 June 2023
45. Wang, Z., et al.: Shuhai: benchmarking high bandwidth memory on fpgas. In: 2020 IEEE 28th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM), pp. 111–119. Fayetteville (2020)
46. Rupniewski, M., et al.: A real-time embedded heterogeneous gpu/fpga parallel system for radar signal processing. In: 2016 Intl IEEE Conferences on Ubiquitous Intelligence and Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld), pp. 1189–1197. Toulouse (2016)
47. Li, W., et al.: Design of high-speed software defined radar with gpu accelerator. IET Radar, Sonar Navig. 16(7), 1083–1094 (2022). https://doi.org/10.1049/rsn2.12244
48. Kulpa, J.S., Maślikowski, L.u., Malanowski, M.: Filter-based design of noise radar waveform with reduced sidelobes. IEEE Trans. Aero. Electron. Syst. 53(2), 816–825 (2017). https://doi.org/10.1109/taes.2017.2665145
49. Savci, K., et al.: Trials of a noise-modulated radar demonstrator – first results in a marine environment. In: 20th International Radar Symposium (IRS 2019), pp. 1–9. Ulm, Germany (2019)
50. Savci, K., Galati, G., Pavan, G.: Low-PAPR waveforms with shaped spectrum for enhanced low probability of intercept noise radars. Rem. Sens. 13(12), (2021). https://doi.org/10.3390/rs13122372
51. Galati, G., et al.: Noise radar technology: waveforms design and field trials. Sensors 21(9), (2021). https://doi.org/10.3390/s21093216
52. Kulpa, K.: The CLEAN type algorithms for radar signal processing. In: 2008 Microwaves, Radar and Remote Sensing Symposium, pp. 152–157. Kiev (2008)
53. Colone, F., et al.: A multistage processing algorithm for disturbance removal and target detection in passive bistatic radar. IEEE Trans. Aero.

Electron. Syst. 45(2), 698–722 (2009). https://doi.org/10.1109/taes.2009.5089551
54. Malanowski, M., Kulpa, K.S.: Detection of moving targets with continuous-wave noise radar: theory and measurements. IEEE Trans. Geosci. Rem. Sens. 50(9), 3502–3509 (2012). https://doi.org/10.1109/tgrs.2011.2181521
55. Wasserzier, C., et al.: Doppler-spread clutter suppression in single-channel noise radar. In: 2019 International Radar Conference (RADAR), pp. 1–4. Toulon (2019)
56. PaoloBlasone, G., et al.: Passive radar DPCA schemes with adaptive channel calibration. IEEE Trans. Aero. Electron. Syst. 56(5), 4014–4034 (2020). https://doi.org/10.1109/taes.2020.2987478
57. Jonsson, R., et al.: Experimental analysis of a clutter suppression algorithm for high time-bandwidth noise radar. In: 2023 IEEE International Radar Conference (RADAR). IEEE, Sydney (2023). accepted for publication
58. Theis, T.N., Wong, H.S.P.: The end of Moore's law: a new beginning for information technology. Comput. Sci. Eng. 19(2), 41–50 (2017). https://doi.org/10.1109/mcse.2017.29
59. O'Donoughue, N.: Emitter Detection and Geolocation for Electronic Warfare. Artech House (2019)
60. Qorvo Us, Inc.: QPD1006 Data Sheet Rev. C. Qorvo Us, Inc., Greensboro. Accessed 20 June 2023

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

How to cite this article: Ankel, M., et al.: Implementation of a coherent real-time noise radar system. IET Radar Sonar Navig. 18(7), 1002–1013 (2024). https://doi.org/10.1049/rsn2.12471