# An evaluation research of constant false alarm rate algorithm for the application of underwater vehicles management system based on USBL sonar principles in shallow water

Doan Trung Thanh*, Tran Quang Giang, Vu Hai Lang, Bach Nhat Hoang

Institute of Electronics, Academy of Military Science and Technology, Cau Giay, Hanoi, Vietnam.
*Corresponding author: doantrungthanhvdt@gmail.com

*Received 08 Nov. 2023; Revised 25 Dec. 2023; Accepted 28 Dec. 2023; Published 25 Feb. 2024.*
**DOI:** https://doi.org/10.54939/1859-1043.j.mst.93.2024.3-11

## ABSTRACT

*The paper presents an overview of underwater traffic management systems utilizing sonar principles. Based on the evaluation of the current situation both domestically and internationally, the research team has proposed the development of a underwater signal processing algorithm based on an adaptive threshold in the Ultra-short Baseline System (USBL). The theoretical results were implemented on FPGA hardware, not only enhancing signal detection capabilities but also meeting real-time signal processing requirements when deployed in complex underwater conditions in various shallow water regions of Vietnam. The research's future direction indicates substantial potential in the practical application of hydroacoustic signal processing algorithms when executing hardware implementations.*

**Keywords:** Sonar; USBL; CFAR; Adaptive algorithm.

## 1. THE CURRENT STATE OF ADAPTIVE ALGORITHM RESEARCH

Research, development, and optimization of underwater vehicle management systems are urgently required in our country due to the unique geographical factors of our extensive coastline. This need has particularly garnered increased attention recently as maritime security, defence, and economic developments become increasingly complex. In practice, underwater vehicle management systems operating on the USBL principle are commonly utilized due to their effectiveness, flexibility, and performance in shallow water regions [1]. In USBL-based management systems, solutions for reducing noise interference typically focus on improving the signal quality during the transmission and reception processes. Many theoretical studies concentrate on various modulation schemes, such as frequency modulation, frequency-coded modulation, or the combination of different modulation rules in a pulse sequence to enhance target detection and positioning [2]. However, noise filtering techniques based on signal structure are unable to eliminate the impact of multipath reflections in many cases. Setting the decision threshold too low can result in false alarms, while setting it too high can lead to signal omissions, resulting in inaccurate outcomes in signal detection problems [3].

To address the limitations of fixed decision thresholds, Constant False Alarm Rate (CFAR) algorithms, commonly used in radar-related applications for target detection, have been applied. However, research on CFAR within USBL systems still has various constraints [4-8].

***Table 1.*** *Summary of the curent state of research.*

| Solutions | Advantages | Limitations |
|---|---|---|
| Frequency Hopped Signals [4] (2001) – USA | Addressed bandwidth limitations of the acoustic transmission channel | Utilized simulated signal data as input |
| Fuzzy frequency Hopped Signals [5] (2014) - China | Proposed a processing solution for both coherent and non-coherent transmission channels | - Complex algorithm, challenging to implement on compact devices |

| | | |
|---|---|---|
| CA-CFAR for Frequency Hopped Signals [6] (2015) - China | Improved accuracy and enhanced underwater communication efficiency | - Hardware not yet fully realized on mobile devices<br>- High computational |
| Adaptive CFAR [7, 8] (2021, 2023) - Iran | - High adaptability; Allows the selection of various CFAR types | Hardware implementation effectiveness not yet disclosed |

Research into adaptive algorithms began in 2001 when the researchers used a combined CFAR approach for frequency hopped signals. However, the algorithm remained theoretical, with no subsequent research on hardware implementation [4]. Published research involving the practical use of a fuzzy logic-based probability distribution model and CFAR for frequency hopped signals emerged in 2014. It revealed the algorithm's complexity and suitability only for large-scale hardware in LBL systems [5]. To address this limitation, scientists continued to enhance the adaptive CFAR channel model for frequency hopped signals in 2015 [6]. This solution reduced algorithm complexity but lacked reported hardware implementations on compact mobile devices. More recently, in 2021 [7] and 2023 [8], the researchers introduced a solution employing adaptive CFAR algorithms for processing underwater acoustic signals in shallow water regions. This solution allows for the selection of various CFAR algorithm types and has improved signal detection accuracy, enhancing the quality of underwater vehicle management systems.

In Vietnam, the research group at the Institute of Electronics – Academy Military of Science and Technology have published paper on building a USBL underwater vehicle management system using linear frequency signals [9]. The research group have also proposed an optimized algorithm that allows for the selection of baseline sizes for USBL systems using Costas-coded frequency hopping signals [10]. Based on the global research landscape discussed earlier and their existing results, this paper presents the research outcomes, algorithm implementation, and the use of adaptive CFAR algorithms for Costas frequency hopped signals in a USBL system. To achieve this, section 2 describes the theoretical foundation and the system's approach. Section 3 constructs the CFAR algorithm flowchart, outlines the algorithm's execution steps and implementation, compares its performance on hardware and software, presents verification results with real-world data collected under the same conditions in Lan Ha Bay, Vietnam, and evaluates and compares it against the results without using the CFAR algorithm. Section 4 outlines the direction for further research and provides a general conclusion.

## 2. THEORETICAL LITERATURE AND PROPOSED SOLUTION FOR CFAR

### 2.1. CFAR algorithm

The algorithms belonging to the CFAR family focus on estimating the decision threshold value based on the noise power from signal samples collected over a time interval equal to the processing window's size of the detector. Algorithm execution steps need to be performed on each data cell within the data vector. Typically, the CFAR algorithm processing block is placed at the output of the matched filter. The general principle of CFAR algorithms is summarized in figure 2. Currently, two main CFAR algorithm types used in hydroacoustic systems for estimating noise power are the Cell Averaging (CA) method and the Order Statistic (OS) approach [7, 8]. In this setup, the processing window includes: one Cell Under Test (CUT) in the center, a group of Guard Cells, and a group of Reference Cells, evenly divided on both sides of the CUT. During the computation in one loop, the value Z is estimated through statistical operations on the group of reference cells. The scale factor $\alpha$ is chosen according to the estimation method and the desired false alarm probability value that the system aims for. The value "$\alpha.Z$" is referred to as the decision threshold, and its magnitude is compared with the magnitude of the central CUT cell to make the final decision [3, 4]. Two hypotheses are set to make a decision about the presence of the signals; $H_0$: Decision "y" has no useful signal and $H_1$: Decision "y" contains a useful signal "d" and noise "g"

$$H_0: y = g$$

$$H_1: y = d + g$$

In this case, the decision criterion is expressed by formula (1):

$$e(y) = \begin{cases} H_0, CUT < \alpha Z \\ H_1, CUT \geq \alpha Z \end{cases} \tag{1}$$

## 2.2. CA-CFAR algorithm

The Cell Averaging CFAR (CA-CFAR) algorithm uses the operation to compute the average value from the reference samples through formula (2) [8] within the statistical estimation block:

$$Z = \frac{1}{2n}\sum_{i=1}^{2n} x_i \tag{2}$$

In practical systems, the false alarm probability value (PFA) and the reference window size 2n are usually predetermined and considered fixed. Thus, the scale factor αCA is a constant, determined by formula (3):

$$\alpha_{CA} = 2n\left(P_{FA}^{-1/2n} - 1\right) \tag{3}$$

## 2.3. OS-CFAR algorithm

The idea behind the Order Statistics CFAR (OS-CFAR) algorithm is to estimate the noise power based on the “k-th” ordered reference cell value after sorting in ascending order. Therefore, if another signal appears within the reference window, its value will not affect the peak detection within the CUT cell.

To determine the decision threshold, first, an appropriate “k” value needs to be found. According to the research [2], the value “k” to be 3/4 of the reference window size is considered a standard choice. The scale factor αOS is calculated using formula 4 [2].

$$P_{FA} = k\binom{2n}{k}\frac{(k-1)!(\alpha_{OS} + 2n - k)!}{(\alpha_{OS} + 2n)!} \tag{4}$$

## 2.4. Dataset

The hardware module using the CFAR algorithm is designed for the purpose of processing data at the output of the matched filter in the receive channel of the USBL system [12, 13], which has been tested in Lan Ha Bay, Vietnam. Some technical parameters of the USBL management system are summarized in table 2:

***Table 2.*** *Some Parameters of the USBL System.*

| *Parameter* | *Value* |
|---|---|
| Frequency | 28 – 36 KHz |
| Signal Modulation | Costas hopping |
| Pulse width | 8 ms |
| Sampling Frequency | 200 KHz |
| Speed of Sound in Water | 1500 |

## 2.5. Flowchart of proposed solution for CFAR algorithm

The flowcharts of the CA-CFAR and OS-CFAR algorithms are depicted in figure 1. “S" denotes the input data set, "In_Buf" signifies the data buffer with a size of L samples, "CUT_Buf" represents the buffer responsible for retaining samples at the CUT position, "TCA_Buf" functions as a buffer for storing threshold values computed based on the CA-CFAR algorithm, and "TOS_Buf" serves as a buffer for storing threshold values calculated using the OS-CFAR algorithm. The primary distinction between these algorithms lies in the method used for estimating noise power. In the CA-CFAR algorithm, the noise power is determined by averaging over the entire reference window. In contrast, the OS-CFAR algorithm calculates the noise power by selecting the power at the "k-th" position in the reference window after sorting the data within the reference window in ascending order.

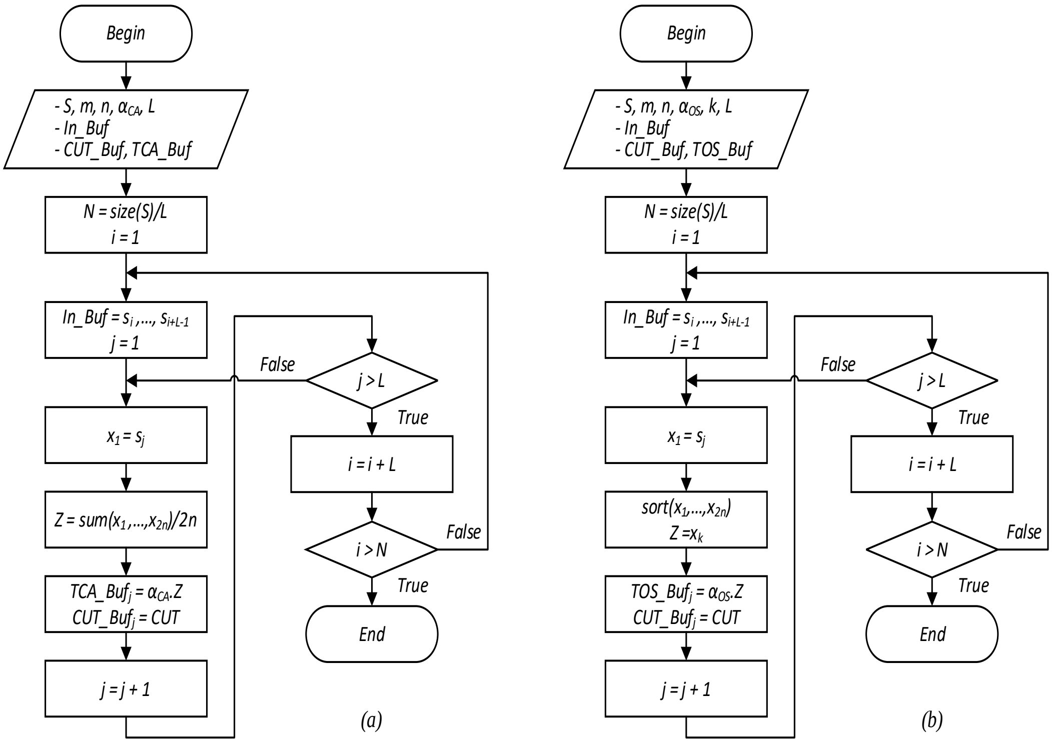



***Figure 1.*** *Flowchart of CFAR algorithm.*
*(a) CA-CFAR algorithm; (b) OS-CFAR algorithm.*

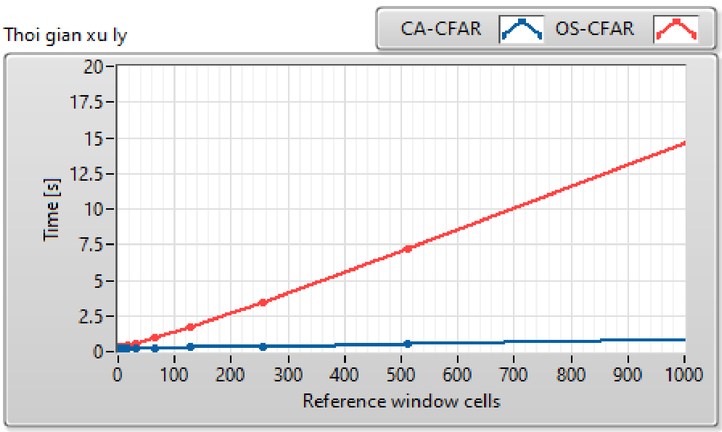



***Figure 2.*** *Dependence of calculation time on reference window size.*

Due to the additional computational overhead introduced by the sorting process, the OS-CFAR detection scheme exhibits a higher processing delay compared to the CA-CFAR method.

When adhering to the criterion of maximizing the reference window size within real-time processing constraints, the CA-CFAR detector permits a larger reference window. Figure 2 illustrates the results of simulation evaluations of the processing time for both algorithms, utilizing the same input data set with a size of 137216 samples, on a Dell Precision T3600 computer system equipped with an Intel Xeon CPU E5-2670 operating at 2.60 GHz and 16.0 GB of RAM. The simulation results reveal the processing time of the OS-CFAR and CA-CFAR at various reference window sizes. With a sampling rate of 200 KHz, which corresponds to a data sample processing time of 5 µs, the CFAR detectors must ensure that the cumulative processing delay remains under 5 µs per processed sample to achieve real-time processing.

### 2.6. Calculation of Parameters for the CFAR Detector

Two important parameters in the design of the CFAR detector are the choice of the reference window size "2n" and the guard cell size "2m". The final result in the CFAR detector is the calculation of the decision threshold at time "i" in the "i-th" moment of the input data sequence, meaning the CFAR detector operates in real-time. With the characteristics of the Costas dataset in section 2.4, the paper constructs a CFAR detector with the following proposed parameters:

***Table 3.*** *Proposed Parameters of the CFAR Detector.*

| ***Parameter*** | ***Value*** |
|---|---|
| $P_{FA}$ | ~1.3x$10^{-6}$ |
| Guard Cells *2m* | 30 |
| Reference Cells *2n* | 512 |
| $\alpha_{CA}$ | 14 |
| $\alpha_{OS}$ | 10 |
| *k* | 384 |

The threshold value depends on the following parameters:

1. Number of reference cells (Reference window size) "2n": In every CFAR algorithm, the size of the reference window "2n" reflects the variation of noise power characteristics over the computation time. The larger "2n" becomes, the more characteristic and stable the parameter Z is. However, increasing the value of "2n" also leads to an increase in system processing delay. Therefore, the requirement when choosing the reference window size is to select the largest size within the real-time processing capability.

2. Number of guard cells "2m": When data passes through the CFAR detector, a guard interval "2m" is good if it ensures that the entire portion of data is considered useful signals and should be protected by the number of guard cells.

## 3. IMPLEMENTING THE CFAR ALGORITHM IN FPGA IN HARDWARE

### 3.1. Excution model

The CA-CFAR test model comprises two functional blocks: a display interface built using LabVIEW software and a CA-CFAR module constructed using the VHDL hardware programming language, compiled and synthesized using Xilinx's ISE Design Suite 14.7 software. These two functional blocks are interconnected through the Ethernet RJ45 communication standard. The display interface block is responsible for reading data files and transmitting them to the CA-CFAR module. The CFAR module then processes the data and sends the results back to the display interface. The processing results include detection thresholds calculated using the CA-CFAR algorithm and the Cell Under Test (CUT) values, which are displayed for comparison. The paper utilizes an FPGA board equipped with the Xilinx Spartan-6 series XC6SLX9 chip. The Spartan-6 FPGA chip represents an optimal choice for striking a balance between cost, space, performance, and affordability, making it suitable for both medium-sized and small-scale applications.

### 3.2. Modul CA-CFAR algorithm

The CA-CFAR detection system is structured with three sub-modules: the Input Buffer, the algorithm module, and the Output Buffer.

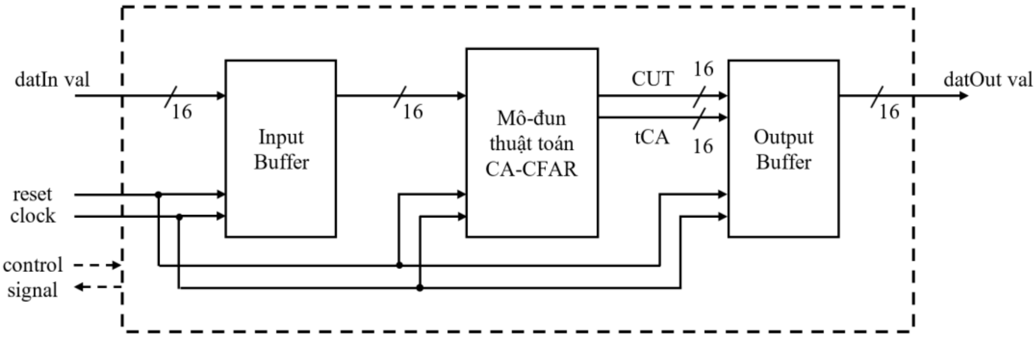



***Figure 3.*** *CA-CFAR module block diagram.*

The Input Buffer and Output Buffer are designed to facilitate communication between the module and the processing system. The Input Buffer is utilized to store received signal samples before forwarding them to the CA-CFAR algorithm block. Similarly, the Output Buffer stores all computed results and transfers the display data to the computer.

In practical signal processing tasks, to apply time-frequency transformations, the buffer sizes are often designed in powers of 2. Furthermore, when performing data computations and processing on an FPGA, using values in the form of powers of 2 significantly enhances processing speed. In such cases, multiplication, division, and averaging operations can be efficiently carried out using shift operations.

With a clock frequency of $f_{CLK}$ = 120 MHz applied to the CA-CFAR algorithm module, assuming that the input buffer and output buffer have a size of L = 512 samples, the module will meet real-time processing requirements if the total processing time for the CA-CFAR samples ($T_{CA\text{-}CFAR}$) is less than the time required to write data into the $T_W$. buffer. The value of $T_W$ is calculated as follows:

$$T_W = L \times T_S = L \times \frac{1}{f_S} = 512 \times \frac{1}{200000} = 2.56\ [ms]$$

Given the chosen parameters for the CFAR module ($m$ = 15, $n$ = 256 and $\alpha_{CA}$ = 16), the size of the processing window of the CA-CFAR module is:

$$L_{CA-CFAR} = 2m + 2n + 1 = 30 + 512 + 1 = 543\ [sample]$$

Therefore, the total processing time ($T_{CA\text{-}CFAR}$) can be calculated as:

$$T_{CA-CFAR} = L \times L_{CA-CFAR} \times \frac{1}{f_{CLK}} = 512 \times 543 \times \frac{1}{120 \times 10^6} \approx 2.4\ [ms]$$

Thus, $T_{CA\text{-}CFAR}$ is less than $T_W$, satisfying the real-time processing requirement. The algorithm module is at the core of the design, consisting of the CFAR window block and computational logic block. The CFAR window block's task is to receive each data cell shifted into the input buffer. Meanwhile, the computational logic block calculates the average over 2n reference samples within the CFAR window to determine the threshold value. The detection threshold, along with the corresponding CUT value, is then sent to the output buffer, concluding one iteration. This process is repeated for each sample in the input buffer.

The input buffer, output buffer, and CFAR window block are constructed based on Xilinx-supported Block RAM (RAMB8BWERS). Table 4 provides a summary of the hardware synthesis results for the CA-CFAR module on the Spartan-6 Series FPGA board XC6SL9 with the specified computational design parameters.

***Table 4.*** *Hardware Synthesis Results for the CA-CFAR Module on Spartan-6.*

| *Resources* | *Total* | *Usage* |
|---|---|---|
| Slice Registers | 11.440 | 210 |
| Slice LUTs | 5720 | 277 |
| RAMB8BWERs | 64 | 6 |

### 3.3. Detection capabilities of CA-CFAR and OS-CFAR

The execution results of the algorithm demonstrate that when signals appear within the CFAR window (either one or multiple signals simultaneously), the OS-CFAR algorithm exhibits better responsiveness compared to CA-CFAR. As illustrated in figure 4, at the surveillance position, the decision threshold value deviates by approximately 1.17 dB. This brings about a benefit in terms of maximum detection capability for sonar systems.

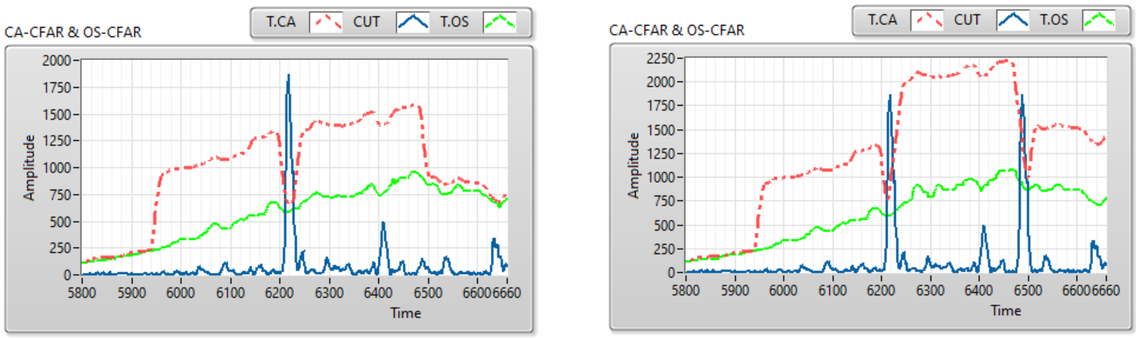
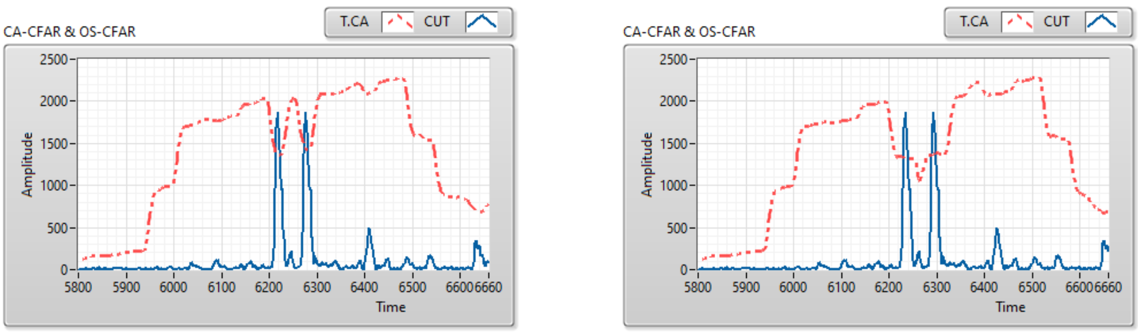



***Figure 4.*** *Detection threshold of two CFAR algorithms in case there exists one signal and two signals in the CFAR window.*

The reason for the consistently higher decision threshold of the CA-CFAR detector over OS-CFAR arises from the noise power estimation method. For the CA-CFAR algorithm, noise power is estimated from the average of all reference cells. Consequently, all noise and signal values within the reference window contribute to the estimation process. This results in an estimation bias. In the case of the OS-CFAR algorithm, only a specific value at the k-th position in the reference window participates in the noise power estimation process. Thus, the influence of other signals on the signal in the test cell is almost eliminated.

### 3.4. Computational costs of CA-CFAR and OS-CFAR




*(a) 2m=30* *(b) 2m=64*

***Figure 5.*** *Detection threshold of the CA-CFAR algorithm.*

For a known signal structure, when the total number of guard cells can cover all the signal samples within a pulse, it provides the best noise power estimation results. If the total number of guard cells is less than the number of samples in a signal pulse, the estimated noise power will be influenced by some of the signal samples within the signal pulse. As a result, the probability of missing useful signals increases because the detection threshold value is elevated.

Conversely, if the total number of guard cells significantly exceeds the number of samples in a signal pulse, not only does the estimated noise power not adequately capture the characteristics of the nearby noise when a signal is present, but it can also lead to missing useful signals when multiple signals simultaneously appear within the CFAR window.

Table 5 illustrates the relationship between the CA-CFAR threshold values and the size of the guard cell.

***Table 5.*** *CA-CFAR Threshold Values Based on Guard Cell Size.*

| ***CA-CFAR threshold*** | ***Guard cells 2m*** |
|---|---|
| 1203 | 2 |
| 1109 | 4 |
| 942 | 8 |
| 735 | 16 |
| 662 | 32 |
| 628 | 64 |

Figure 5(a) demonstrates that when the total number of guard cells is approximately equal to the signal pulse width (2m = 30), the signal detection output distinguishes two useful signals coexisting within the CFAR window, as the amplitude of the signal is 4 times the threshold's reversal. However, when 2m = 64, the amplitude change only reverses twice, as shown in figure 5(b). Therefore, an additional algorithm is required to differentiate between the two targets.

## 4. CONCLUSIONS

The article originates from theoretical research, progressing towards the simulation, analysis, and implementation of the CA-CFAR algorithm on FPGA hardware. Based on the characteristics of known signals and the parameters of the Spartan-6 board, the algorithm's parameters are carefully selected to ensure that the hardware module can meet real-time processing requirements. The results demonstrate that the CA-CFAR algorithm module meets the criteria for processing latency. Therefore, it can be seamlessly integrated behind the combined filtering block in the transmit-receive chain of active sonar systems to enhance the detection capability of underwater acoustic signals. The results obtained using this statistical algorithm can be seamlessly integrated into sonar depth measurement systems employing a bidirectional transducer feedback loop with an omnidirectional USBL hydrophone. However, alongside these accomplishments, the article acknowledges certain limitations arising from its focus on typical algorithms. Hence, to enhance the practical applicability, diversification in the application of various CFAR algorithms is imperative. These research directions are reserved for future endeavors.

## REFERENCES


[1]. Arkhipov, Mikhail. *“Utilizing Johnson solids for designing multielement USBL systems.”* IEEE Journal of Oceanic Engineering 41.4: 783-793, (2016).

[2]. Hague, David A.; Buck, John R. *“The generalized sinusoidal frequency-modulated waveform for active sonar”*. IEEE Journal of Oceanic Engineering, 42.1: 109-123, (2016).

[3]. Diamant, Roee; lampe, Lutz. *“Low probability of detection for underwater acoustic communication: A review”*. IEEE Access, 6: 19099-19112, (2018). DOI: 10.1109/ACCESS.2018.2818110.

[4]. Preisig, James C.; johnson, Mark P. *“Signal detection for communications in the underwater acoustic environment”*. IEEE journal of oceanic engineering, 26.4: 572-585, (2001). DOI: 10.1109/48.972096.

[5]. Xu, Yanwei, et al. *"Fuzzy statistical normalization CFAR detector for non-Rayleigh data"*. IEEE transactions on aerospace and electronic systems, 51.1: 383-396, (2015). DOI: 10.1109/TAES.2014.130683
[6]. Liao, Ho-En, et al. *"A computation efficiency and-cfar for fmcw radar receiver"*. In: 2013 Ninth International Conference on Intelligent Information Hiding and Multimedia Signal Processing. IEEE. p. 108-112, (2013). DOI: 10.1109/IIH-MSP.2013.36.
[7]. Pakdel azar, Omid, et al. *"Enhanced target detection using a new combined sonar waveform design"*. Telecommunication Systems, 77: 317-334, (2021). DOI : https://doi.org/10.1007/s11235-021-00761-6
[8]. Pakdel azar, Omid, et al. *"Enhanced target detection using a new cognitive sonar waveform design in shallow water"*. Applied Acoustics, 205: 109270, (2023). DOI : https://doi.org/10.1016/j.apacoust.2023.109270
[9]. T.Q. Giang., et al, *"Improving location precision for USBL system using four hydrophone array"*, Journal of Military Science and Technology Research, Special Issue of the Institute of Electronics, pp. 133-138, (2020).
[10]. Hoang B.N., et al, *"Optimizing baseline in USBL using Costas hopping to increase navigation precision in shallow water"*, In 16th International Conference on Ubiquitous Information Management & Communication, IEEE, pp.1-6, (2022).

## TÓM TẮT

### Nghiên cứu đánh giá thuật toán ổn định xác suất báo động lầm ứng dụng trong hệ thống quản lý phương tiện giao thông dưới nước theo nguyên lý sonar trong vùng biển nước nông

*Bài báo nghiên cứu tổng quan các hệ thống quản lý phương tiện giao thông dưới nước theo nguyên lý sonar. Dựa trên đánh giá hiện trạng trong và ngoài nước, nhóm tác giả đã đề xuất xây dựng thuật toán xử lý tín hiệu thủy âm dựa trên ngưỡng thích nghi trên hệ thống đường cơ sở cực ngắn (Ultra-short baseline-USBL). Các kết quả lý thuyết được cài đặt thực tế trên phần cứng FPGA không những cải thiện được khả năng phát hiện tín hiệu, mà còn đáp ứng được yêu cầu xử lý tín hiệu thời gian thực khi triển khai thực nghiệm với điều kiện thủy văn phức tạp tại một số vùng biển nước nông của Việt Nam. Hướng phát triển của nhóm tác giả đã cho thấy tiềm năng lớn khi ứng dụng các thuật toán xử lý tín hiệu thủy âm khi thực thi phần cứng thực tế.*

**Từ khoá:** Sonar; USBL; CFAR; Xử lý tín hiệu thích nghi.