SIGNAL 2026

The Eleventh International Conference on Advances in Signal, Image and Video Processing

ISBN: 978-1-68558-360-6

March 8th –12th, 2026

Valencia, Spain

SIGNAL 2026 Editors

Pavel Loskov, ZJU-UIUC, China

SIGNAL 2026

Foreword

The Eleventh International Conference on Advances in Signal, Image and Video Processing (SIGNAL 2026), held between March 8 - 12, 2026, continued the inaugural event considering the challenges mentioned above. Having these motivations in mind, the goal of this conference was to bring together researchers and industry and form a forum for fruitful discussions, networking, and ideas.

Signal, video and image processing constitutes the basis of communications systems. With the proliferation of portable/implantable devices, embedded signal processing became widely used, despite that most of the common users are not aware of this issue. New signal, image and video processing algorithms and methods, in the context of a growing-wide range of domains (communications, medicine, finance, education, etc.) have been proposed, developed and deployed. Moreover, since the implementation platforms experience an exponential growth in terms of their performance, many signal processing techniques are reconsidered and adapted in the framework of new applications. Having these motivations in mind, the goal of this conference was to bring together researchers and industry and form a forum for fruitful discussions, networking, and ideas.

We take here the opportunity to warmly thank all the members of the SIGNAL 2026 Technical Program Committee, as well as the numerous reviewers. The creation of such a high quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and efforts to contribute to SIGNAL 2026. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

Also, this event could not have been a reality without the support of many individuals, organizations, and sponsors. We are grateful to the members of the SIGNAL 2026 organizing committee for their help in handling the logistics and for their work to make this professional meeting a success.

We hope that SIGNAL 2026 was a successful international forum for the exchange of ideas and results between academia and industry and for the promotion of progress in the field of signal processing.

We are convinced that the participants found the event useful and communications very open. We also hope that Valencia provided a pleasant environment during the conference and everyone saved some time for exploring this beautiful city

SIGNAL 2026 Chairs:

SIGNAL 2026 Steering Committee
Constantin Paleologu, Polytechnic University of Bucharest, Romania
Wilfried Uhring, Université de Strasbourg, France
Jérôme Gilles, San Diego State University, USA
Sergey Y. Yurish, Excelera, S. L. | IFSA, Spain
Pavel Loskot, ZJU-UIUC Institute, China

SIGNAL 2026 Publicity Chairs
Francisco Javier Díaz Blasco, Universitat Politècnica de València, Spain
Ali Ahmad, Universitat Politècnica de València, Spain
Sandra Viciano Tudela, Universitat Politecnica de Valencia, Spain

José Miguel Jiménez, Universitat Politecnica de Valencia, Spain

SIGNAL 2026

Committee

SIGNAL 2026 Steering Committee
Constantin Paleologu, Polytechnic University of Bucharest, Romania
Wilfried Uhring, Université de Strasbourg, France
Jérôme Gilles, San Diego State University, USA
Sergey Y. Yurish, Excelera, S. L. | IFSA, Spain
Pavel Loskot, ZJU-UIUC Institute, China

SIGNAL 2026 Publicity Chairs
Francisco Javier Díaz Blasco, Universitat Politècnica de València, Spain
Ali Ahmad, Universitat Politècnica de València, Spain
Sandra Viciano Tudela, Universitat Politecnica de Valencia, Spain
José Miguel Jiménez, Universitat Politecnica de Valencia, Spain

SIGNAL 2026 Technical Program Committee

Waleed H. Abdulla, The University of Auckland, New Zealand
Ahmed Al Hilli, Technical College of Najaf | Al-furat Al-Awsat Technical University, Iraq
Kiril Alexiev, Institute for Information and Communication Technologies -Bulgarian Academy of Sciences, Bulgaria
Djamila Aouada, SnT | University of Luxembourg, Luxembourg
Nadia Baaziz, Université du Québec en Outaouais, Canada
Vesh Raj Sharma Banjade, Intel Coporation, USA
Joan Bas, CTTC, Spain
Wassim Ben Chikha, Tunisia Polytechnic School, Tunisia
Amel Benazza-Benyahia, SUP'COM | COSIM lab. | University of Carthage, Tunisia
Anirban Bhowmick, Vellore Institute of Technology | Bhopal University, India
Larbi Boubchir, LIASD - University of Paris 8, France
Moez Bouchouicha, LIS - Laboratoire d'Informatique et Systèmes | Toulon University, France
Salah Bourennane, Ecole Centrale de Marseille, France
Geraldo Braz, Federal University of Maranhão, Brazil
Valeria Bruschi, Università Politecnica delle Marche, Ancona, Italy
Paula María Castro Castro, University of A Coruña, Spain
M. Girish Chandra, TCS Research & Innovation, India
Aizaz Chaudhry, Carleton University, Canada
Zhuyun Chen, South China University of Technology,Guangzhou, China
Qiang Cheng, University of Kentucky, USA
Ying-Ren Chien, National Taipei University of Technology, Taiwan
Doru Florin Chiper, Technical University Gheorghe Asachi of Iasi, Romania
Sergio Cruces, Universidad de Sevilla, Spain
João Dallyson Sousa de Almeida, Federal University of Maranhão, São Luís, Brazil
Natasja M. S. de Groot, Erasmus Medical Center | Technical University Delft, Netherlands
Laura-Maria Dogariu, National University of Science and Technology POLITEHNICA Bucharest, Romania

Copyright Information

For your reference, this is the text governing the copyright release for material published by IARIA.

The copyright release is a transfer of publication rights, which allows IARIA and its partners to drive the dissemination of the published material. This allows IARIA to give articles increased visibility via distribution, inclusion in libraries, and arrangements for submission to indexes.

I, the undersigned, declare that the article is original, and that I represent the authors of this article in the copyright release matters. If this work has been done as work-for-hire, I have obtained all necessary clearances to execute a copyright release. I hereby irrevocably transfer exclusive copyright for this material to IARIA. I give IARIA permission or reproduce the work in any media format such as, but not limited to, print, digital, or electronic. I give IARIA permission to distribute the materials without restriction to any institutions or individuals. I give IARIA permission to submit the work for inclusion in article repositories as IARIA sees fit.

I, the undersigned, declare that to the best of my knowledge, the article is does not contain libelous or otherwise unlawful contents or invading the right of privacy or infringing on a proprietary right.

Following the copyright release, any circulated version of the article must bear the copyright notice and any header and footer information that IARIA applies to the published article.

IARIA grants royalty-free permission to the authors to disseminate the work, under the above provisions, for any academic, commercial, or industrial use. IARIA grants royalty-free permission to any individuals or institutions to make the article available electronically, online, or in print.

IARIA acknowledges that rights to any algorithm, process, procedure, apparatus, or articles of manufacture remain with the authors and their employers.

I, the undersigned, understand that IARIA will not be liable, in contract, tort (including, without limitation, negligence), pre-contract or other representations (other than fraudulent misrepresentations) or otherwise in connection with the publication of my work.

Exception to the above is made for work-for-hire performed while employed by the government. In that case, copyright to the material remains with the said government. The rightful owners (authors and government entity) grant unlimited and unrestricted permission to IARIA, IARIA's contractors, and IARIA's partners to further distribute the work.

Table of Contents

Genetic Algorithm Optimization of Bi-phase Codes: Surpassing Barker Code Performance in ISAC Applications

Sevda Şahin
S2 System Engineering
Voorschoten, The Netherlands
e-mail: sevda.sahin@s2system.nl

Tolga Girici
Faculty of Science and Technology
Middlesex University, NW4 4BT
London, United Kingdom
e-mail: t.girici@mdx.ac.uk

***Abstract*—This paper formalizes the ambiguity function for bi-phase coded waveforms in Integrated Sensing and Communication (ISAC) systems. Utilizing a generalized formula, this study optimizes the Peak to Side Lobe Ratio (PSLR) through a genetic algorithm, enhancing both radar and communication performances simultaneously. The optimized codes show substantial improvements in detection and transmission capabilities using traditional Barker codes, validating the proposed method's efficacy. These results highlight the potential of bi-phase codes to revolutionize ISAC technologies by effectively balancing dual-function requirements.**

Keywords-Barker code; Integrated Sensing and Communication (ISAC); Symbol Error Rate (SER); waveform design; Genetic Algorithm

I. INTRODUCTION

The ISAC concept has become quite popular in recent years, aiming to address issues such as spectrum congestion resulting from the increasing variety of both military and civilian wireless communication products, and electromagnetic interference problems on platform installations. Various studies in the literature aim to solve all these problems [1]. However, radar and communication systems process information in different ways. While radar systems extract information from observations gathered in noisy environments, communication systems focus on transmitting predefined signals and extracting information from the noisy environment on the receiver's side. ISAC aims to combine these two functions to achieve a balanced performance through trade-offs [1]–[3]. These studies are generally categorized into two groups: sensing using communication signals and communication using sensing signals. In ISAC studies focusing on communication signals, Orthogonal Frequency Division hlMultiplexing (OFDM) signals are frequently used for sensing over communication signals [4], [5]. However, when OFDM signals are used for radar detection, the bandwidth required for the radar's high range resolution is greater than that of a typical communication data link. In this case, high-sampling-rate Analog-to-Digital Converters (ADC) and thus high-cost hardware must be incorporated into the system. Other communication signals, such asOrthogonal Chirp-Division Multiplexing (OCDM) and Orthogonal Time-Frequency Space (OTFS), have also been studied [4], [6]. All of these multi-carrier communication signals must support full-duplex operation when used for target detection in a mono-static scenario. In this case, ensuring complete isolation between the receiver and transmitter while maintaining receiver sensitivity and dynamic range presents a separate design challenge. Within the scope of communication over sensing signals, Linear Frequency Modulated (LFM) radar signals have been recommended in [7], [8] and phase coded waveforms has been recommended in [9] for transmitting communication signals. In order to evaluate the radar performance (e.g., range and Doppler resolution) of the proposed waveforms for ISAC systems, it is useful to mathematically formulate and analyze the ambiguity functions. In [10], the optimized OTFS waveform is proposed as an ISAC waveform, and ambiguity function analyses have been conducted. In [11], Hyperbolic Fractional Fourier Transform (HFrFT) multicarrier signals have been applied to ISAC systems, and their performance has been analyzed using the ambiguity function. In [12], the authors design integrated radar and communication systems using weighted pulse trains with Oppermann sequences, deriving an analytical expression for the ambiguity function and showing that it depends on a single sequence parameter, thus simplifying the design process. In [13], the authors demonstrate that embedding Phase-Shift Keying (PSK) symbols in radar waveforms can benefit both radar and communication in a dual system, showing that PSK modulation reduces ambiguity function sidelobe peaks and increases the number of orthogonal transmit waveforms without raising sidelobe levels. In [14], the authors use a genetic algorithm to Optimize Code-shift Keying (CSK) sequences in a Frequency Hopping (FH) MIMO radar system, analyzing the ambiguity function with and without information symbol embedding. The optimized sequences significantly reduce range sidelobe levels and clutter modulation, addressing challenges from hopping frequency re-use, while also providing good spectrum containment and supporting high communication data rates. Despite not using the ambiguity function, the authors in [15] focus on the simultaneous optimization of autocorrelation and cross-correlation characteristics of Oppermann sequences as a multiobjective problem. They evaluate the performance of various state-of-the-art multiobjective evolutionary meta-heuristic algorithms for designing Oppermann sequences in integrated radar and communication systems.

In this study, we contribute to the literature by mathematically formulating the ambiguity functions for pulsed ISAC-Barker waveforms, as proposed in our previous work [9]. This formulation takes into account the matched filter bank structure at the radar receiver, as detailed in our recent study [16]. The resulting expressions provide a general framework applicable

 ISBN: 978-1-68558-360-6

to biphase-coded sequences.

Additionally, the ambiguity functions for zero-delay and zero-Doppler cuts are plotted and compared both theoretically and through signal simulations. Subsequently, an analysis of the PSLR for the zero delay and zero Doppler cuts of the uncertainty function of the ISAC-Barker waveform was conducted. The target detection performance in radar functionality is directly related to the PSLR. On the other hand, the distance between symbols directly affects the SER performance of the communication function. In this study, the objective function was defined using the peak-to-sidelobe ratio values of the ambiguity function at the zero Doppler and zero delay cross sections, along with the Hamming distances of the codes. Optimal codes were generated for various lengths and symbol counts using a genetic algorithm. Additionally, a global optimization was performed, taking into account not only the zero Doppler and zero delay cross sections but all cross sections of the ambiguity function. The radar and communication performances of the ISAC system were evaluated for both the codes generated as a result of the optimization and for Barker codes.

The remainder of this paper is organized as follows. In Section II, the signal model and ambiguity function formulation for pulsed ISAC waveforms are presented. Section III describes the design of biphase-coded sequences using a genetic algorithm, including the dual-objective optimization framework. Section IV provides numerical results, comparing the performance of the optimized binary sequences with conventional Barker codes in terms of PSLR and SER. Finally, Section V concludes the paper with a summary of the contributions and a discussion of potential future work.

II. Ambiguity Function

The radar ambiguity function represents the output of the matched filter and describes the interference caused by a target's range and/or Doppler shift compared to a reference target with the same RCS (Radar Cross Section). The ambiguity function calculated at $(\tau, f_d) = (0, 0)$ corresponds to the matched filter output that perfectly matches the signal reflected from the target of interest. In other words, the reflections from the nominal target are located at the origin of the ambiguity function. Therefore, the ambiguity function values at non-zero τ and f_d represent reflections related to different ranges and Doppler shifts from those of the nominal target [17]. For a moving target, the output of the matched filter is calculated as given in (1). Here, f_d represents the Doppler frequency.

$$\chi(\tau, f_d) = \int_{-\infty}^{\infty} \tilde{x}(t)\tilde{x}^*(t-\tau)e^{j\pi f_d t}dt \tag{1}$$

The square of the absolute value of the matched filter output yields the ambiguity function given in (2).

$$|\chi(\tau, f_d)|^2 = \left| \int_{-\infty}^{\infty} \tilde{x}(t)\tilde{x}^*(t-\tau)e^{j\pi f_d t}dt \right|^2 \tag{2}$$

The radar ambiguity function is used by radar designers to examine different waveforms. The suitability of various radar waveforms for different radar applications can be analyzed using the ambiguity function. It is also used to determine the range and Doppler resolutions for a specific radar waveform. A three-dimensional plot of the ambiguity function against frequency and time delay is referred to as the radar ambiguity diagram [17].

A. Barker-Coded ISAC Joint Waveform Ambiguity Function

In this work, the term *joint waveform refers to a waveform that simultaneously performs radar sensing and communication functions. The joint waveform is generated according to the method defined in [9]. This allows the radar pulse to carry information for the communication receiver while retaining its radar sensing properties. While the fundamental ambiguity function for pulsed phase-coded waveforms is well-established in classical radar theory [17], the formulation presented here explicitly extends these results to ISAC-Barker joint waveforms with multiple communication symbols and pulse sequences, providing a practical framework for subsequent optimization analyses. The complex envelope of a single phase-coded pulse is expressed as*,

$$x_1(t) = e^{jw_0 t} \sum_{n=1}^{N} P_n(t) e^{j\theta_n}. \tag{3}$$

Where w_0 denotes the carrier angular frequency of the pulse and $P_n(t)$ represents the n-th sub-pulse envelope. When a binary phase code is used, θ_n is either 0 or π. Each sub-pulse is modulated in phase according to the corresponding phase code $D_n = e^{j\theta_n} = \pm 1$ for a bi-phase Barker code. Specifically, the transmitted waveform can be expressed as a sum of phase-coded sub-pulses, where D_n multiplies $P_n(t)$ to impose the desired phase shift. For the joint waveform, D_n changes in accordance with the transmitted communication symbol. Since the Barker code is a bi-phase code, the ambiguity function of a Barker-coded pulse for $0 < \tau < N\tau_0$ is expressed by (4) [17].Where τ_0 is a single sub-pulse with duration.

$$\begin{aligned} \chi_{ISAC-Barker}(\tau, f_d) = & \\ & \chi_0(\tau', f_d) \sum_{n=1}^{M-k} D_n D_{n+k} e^{-j2\pi f_d (n-1)\tau_0} \\ & + \chi_0(\tau_0 - \tau', f_d) \sum_{n=1}^{M-k} D_n D_{n+k+1} e^{-j2\pi f_d n \tau_0} \end{aligned} \tag{4}$$

Here, M is number of communication symbols, τ is given in (5), and $\chi_0(\tau', f_d)$ is given in (6).

$$\tau = k\tau_0 + \tau' \begin{cases} 0 < \tau' < \tau_0 \\ k = 0, 1, 2, ..., M \end{cases} \tag{5}$$

$$\chi_0(\tau', f_d) = \int_0^{\tau_0 - \tau'} e^{-j2\pi f_d t} dt, 0 < \tau' < \tau_0 \tag{6}$$

For a sequence of L pulses,each consisting of N sub-pulses, with Pulse Repetition Interval (PRI) T, where the complex envelope of a single pulse is given by (3), the transmitted

signal consisting of L consecutive pulses is expressed as in (7).

$$\tilde{x}_{LT}(t) = \frac{1}{\sqrt{L}} \sum_{i=0}^{L-1} \tilde{x}_1(t - iT) \tag{7}$$

Where $\tilde{x}_1(t)$ represents the transmitted phase-coded pulse corresponding to the complex envelope $x_1(t)$ in (3).The ambiguity function for a sequence of L pulses is given in (8).

$$|\chi_{LT}(\tau, f_d)| = \frac{1}{L} \sum_{q=-(L-1)}^{L-1} |\chi_1(\tau - qT, f_d)| \times \left| \frac{\sin[\pi f_d (L - |q|)T]}{\sin(\pi f_d T)} \right|, |\tau| \leq LT \tag{8}$$

As proven in [18], the ambiguity function for a sequence of L pulse can be expressed as given in (9) for $|\tau| \leq \tau_0$. This form results from the linearity of the matched filter and the periodic structure of the pulse sequence, where the total ambiguity function is obtained as a weighted sum of the individual pulse ambiguity functions, accounting for Doppler-induced phase shifts.

$$|\chi_{LT}(\tau, f_d)| = |\chi_1(\tau, f_d)| \left| \frac{\sin[\pi f_d LT]}{N \sin(\pi f_d T)} \right| \tag{9}$$

Using (8), the ambiguity function for a sequence of L ISAC-Barker pulses is expressed by (10).

$$|\chi(\tau, f_d)| = \frac{1}{L} \sum_{q=-(L-1)}^{L-1} |\chi_{ISAC-Barker}(\tau - qT, f_d)| \times \left| \frac{\sin[\pi f_d LT]}{\sin(\pi f_d T)} \right|, |\tau| \leq LT \tag{10}$$

Using (9) and (10), the ambiguity function for a sequence of L ISAC-Barker pulses can be calculated for $|\tau| \leq \tau_0$ as shown in (11).

$$|\chi(\tau, f_d)| = |\chi_{ISAC-Barker}(\tau, f_d)| \left| \frac{\sin[\pi f_d LT]}{N \sin(\pi f_d T)} \right|, |\tau| \leq \tau_0 \tag{11}$$

In (12), the zero-Doppler cut for a pulse sequence with the ISAC-Barker waveform for $|\tau| \leq \tau_0$ is provided.

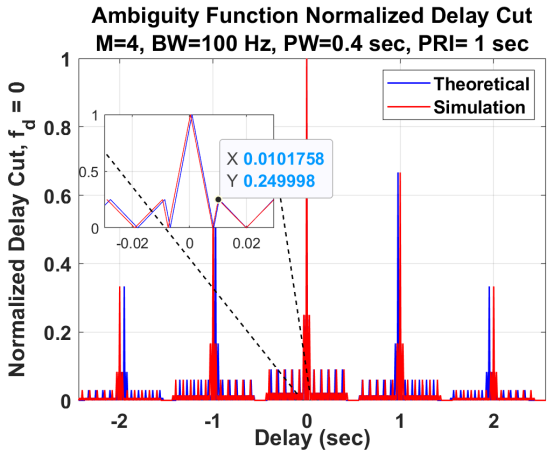
$$\begin{aligned} &|\chi(\tau, 0)| \\ &= \lim_{f_d \to 0} \frac{1}{L} \sum_{q=-(L-1)}^{L-1} |\chi_{ISAC-Barker}(\tau - qT, f_d)| \times \left| \frac{\sin[\pi f_d LT]}{L \sin(\pi f_d T)} \right| \\ &= \frac{1}{L} \sum_{q=-(L-1)}^{L-1} |\chi_{ISAC-Barker}(\tau - qT, 0)| \\ &= \frac{1}{L} \sum_{q=-(L-1)}^{L-1} ((\tau_0 - \tau' + qT) \sum_{n=1}^{N-k} D_n D_{n+k} + (\tau' + qT) \sum_{n=1}^{N-k} D_n D_{n+k+1}) \end{aligned} \tag{12}$$

The locations of the peaks on the delay axis at the zero Doppler cross-section depend on the PRI of the pulse sequence, regardless of the waveform. In this context, the function given by (12), which describes the zero-Doppler cut for a pulse sequence with the ISAC-Barker waveform, exhibits its first peak at $\tau = 0$. Other peaks are observed at $\tau = (n-1)T$ for $n = 1, ..., L$. The widths of all peaks are $2\tau_c$, where $\tau_c = 1/B$ as stated in [17]. The peak values depend only on the number of pulses L in the sequence and are given by $(L - (n-1))/L$. The zero-delay cut for a pulse sequence with the ISAC-Barker waveform is,

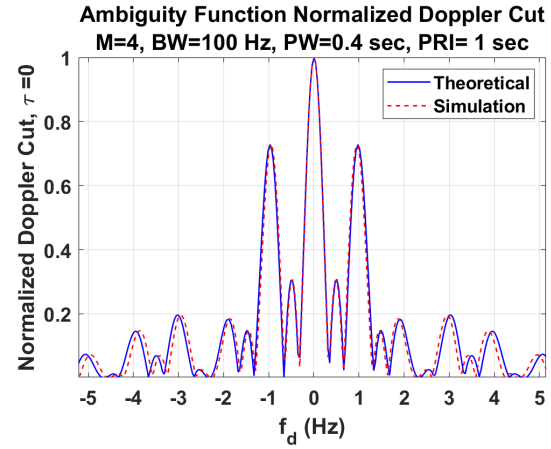
$$\begin{aligned} &\chi_{ISAC-Barker}(0, f_d) \\ &= \left(\chi_0(0, f_d) \sum_{n=1}^{N-k} D_n D_{n+k} e^{-j2\pi f_d (n-1)\tau_0} + \chi_0(\tau_0, f_d) \sum_{n=1}^{N-k} D_n D_{n+k+1} e^{-j2\pi f_d n \tau_0} \right) \\ &\quad \times \left| \frac{\sin[\pi f_d (L - |q|)T]}{\sin(\pi f_d T)} \right|. \end{aligned} \tag{13}$$

The locations of the peak values on the Doppler frequency axis at the zero-delay cross-section depend on the PRI of the pulse sequence, regardless of the waveform, and the peak values are observed at integer multiples of $f_d = 1/T$. The widths of peaks are $2/LT$ [19]. The peak values are given by $\frac{\sin[\pi(k/T)\tau_0]}{\pi(k/T)} \tau_0$, where k is a positive integer.

Figure 1 shows the normalized Doppler cut and the normalized delay cut for ISAC-Barker waveforms with L pulses. As can be seen from the zoomed in section of the graph in Figure 1, the PSLR for the ISAC-Barker waveform is approximately 4 (12 dB). This situation is considered to cause erroneous range measurements for the ISAC-Barker waveform, especially in reflections from nearby targets. At this stage of the study, a genetic algorithm was used to generate an optimal binary phase-coded waveform, in order to reduce the observed PSLR in the ISAC-Barker waveform as proposed in [14].



(a) Delay cut



(b) Doppler cut

Figure 1. Ambiguity function cuts for ISAC-Barker waveform. (a) Delay cut. (b) Doppler cut.

III. Biphase Sequence Optimization by Using Genetic Algorithm

In Section II, the ambiguity function we derived for an ISAC system with pulsed waveforms coded with Barker codes within a pulse is applicable to all biphase phase codes. Motivated by this generalization, we employ a genetic algorithm to design alternative biphase-coded sequences with improved performance. The sequence design problem is discrete, highly non-convex, and involves a large combinatorial search space. Furthermore, the optimization targets multiple conflicting objectives, namely minimizing the PSLR while maintaining reliable communication performance in terms of Hamming distance. Genetic algorithms are well suited for such problems as they efficiently explore large search spaces without requiring gradient information and naturally support multi-objective optimization. In implementing the genetic algorithm, we define two types of objective (fitness) functions. The first function aims to optimize radar performance by maximizing the PSLR at the zero-delay ($PSLR_{Zdel}$), and zero-Doppler ($PSLR_{Zdopp}$) cuts, while also optimizing the Hamming distance, d_{min}, between communication symbols to optimize communication performance (14).

TABLE I. Genetic Algorithm Parameters

Parameter	N44M4	N88M8
Population Size	40	140
Number of Generations	100	100
Crossover Fraction	0.8	0.8
Elite Count	5	5
Population Type	bitstring	bitstring

$$f_{ZeroCut} = \frac{w_1 M}{N} PSLR_{Zdel} - \frac{w_2 M}{N} PSLR_{Zdopp} + \frac{w_3}{N} d_{min} \quad (14)$$

where N is the code length and M is the number of symbols. The second objective function aims to optimize radar performance by maximizing the first peak-to-second peak ratio, PR_{Global}, in all delay and Doppler cuts, while also maximizing the Hamming distance, d_{min}, between communication symbols to optimize communication performance (15).

$$f_{Global} = \frac{w_1 M}{N} PR_{Global} + \frac{w_3}{N} d_{min} \quad (15)$$

Here w_1, w_2 and w_3 are binary coefficients to emphasize different components of the fitness function. In both objective functions, PLSR computation is performed according to the ambiguity function given in (10). A genetic algorithm is used with the parameters given in Table I. In addition, the best fitness value is plotted at each iteration to monitor optimization progress. For k bit/pulse communications, $M = 2^k$ biphase codes are needed. A random candidate solution can be formed by concatenating M random binary codes. The genetic algorithm is initialized by generating a number of these candidate solutions, each of which is of length NM. Then, fitness evaluation, selection, crossover, mutation and replacement procedures are repeated until a converge criterion is met. MATLAB Global Optimization Toolbox is used in implementing the genetic algorithm.

IV. Numerical Analysis

We initially performed our analysis using sequences of length $N = 44$ with $M = 4$ communication symbols (2 bits per pulse repetition interval). The optimization results demonstrated significant improvements in performance compared to the Barker code. Specifically, the optimized biphase sequences achieved better PSLR for zero Doppler cuts while maintaining competitive Hamming distances.

Key observations for $N = 44, M = 4$ include:

- The PSLR for zero Doppler cut was consistently reduced in the optimized codes, as shown in Table II.
- Adjusting the weights (w_1, w_2, and w_3) allowed for trade-offs between PSLR and minimum Hamming distance, leading to sequences tailored for specific design goals.

Encouraged by the promising results for $N = 44, M = 4$, we extended our analyses to longer sequences and more communication symbols. Building on the success of the $N = 44, M = 4$ case, we repeated the analyses with sequences of length $N = 88$ and $M = 8$ communication symbols (i.e. 3 bits/PRI). The objective was to evaluate whether the observed improvements could be replicated with longer sequences and higher symbol counts.

Similar to the $N = 44, M = 4$ case, the optimized sequences for $N = 88, M = 8$ showed enhanced PSLR for zero Doppler cuts while achieving competitive or better minimum Hamming distances. Key findings include:

- The PSLR improvements in the optimized codes were more pronounced for the zero Doppler case, as seen in Table II.
- The flexibility provided by the objective function weights enabled the design of sequences that balance PSLR and Hamming distance effectively.

To demonstrate the difference in radar performance between the Barker and optimized codes through the ambiguity function graph, the analyses provided in Table II for the $N = 44$, $M = 4$ case are illustrated with ambiguity function and contour plots for the Barker and optimized codes in Figure 2 and Figure 3, respectively. As seen in Figure 2, the Barker code exhibits higher side lobes in the zero-Doppler and zero-delay cuts, which may lead to erroneous target detection or range estimation. In contrast, Figure 3 shows that the optimized biphase code suppresses these side lobes, resulting in a cleaner main peak and reduced ambiguity. This clearly demonstrates the improvement in radar resolution and potential communication reliability achieved by the genetic algorithm-optimized codes.

After analyzing the radar performance of these codes, we also evaluated the SER performance of each code for a communication receiver using correlation [9], as illustrated in Figure 4. The SER graphs show how the optimized codes outperform the Barker code in terms of symbol detection accuracy under the same Signal-to-Noise Ratio (SNR) conditions. Since our communication system performs demodulation using correlation, the autocorrelation and cross-correlation characteristics of the waveforms play a significant role in the symbol error rate. The optimized codes have better autocorrelation performance compared to the Barker code because their side lobes are suppressed. As a result, as shown in the SER graphs, the communication performance of the optimized codes is significantly better than that of the Barker code for both $N = 44, M = 4$ and $N = 88, M = 8$. While the analysis emphasizes zero-Doppler and zero-delay cuts, it is important to evaluate the performance under high Doppler shifts, which are relevant in high-mobility scenarios such as automotive or surveillance ISAC applications. Future work should include analyzing the ambiguity function and PSLR for significant Doppler frequencies to assess the robustness of the optimized biphase codes in such environments.

When analyzing the SER performance for the three different optimization objective (Global, Zero Cut with $w_1 = 1, w_2 = 1, w_3 = 1$, and Zero Cut with $w_1 = 1, w_2 = 1, w_3 = 0$), we observe that the results are very similar for both $N = 44, M = 4$ and $N = 88, M = 8$. The SER performance for these optimization objectives is almost identical, with only minor differences at lower SNR levels, as can be seen in the Figure 4. It should be noted that the current SER evaluations assume ideal AWGN channels. The robustness of the proposed codes under practical channel conditions—including multipath fading, frequency selectivity, and hardware impairments—remains an important direction for future investigation.

TABLE II. PSLR AND HAMMING DISTANCE FOR OPTIMIZED AND BARKER CODES. PSLR IS COMPUTED FOR ZERO DOPPLER AND ZERO DELAY CUTS OF THE AMBIGUITY FUNCTION. HAMMING DISTANCE INDICATES THE MINIMUM DISTANCE BETWEEN CODEWORDS.

Code	PSLR Zero Doppler	PSLR Zero Delay	Hamming Distance
Barker44M4	4.0000	4.6031	22
N44M4W111ZC	4.8889	4.6031	22
N44M4W110ZC	5.5000	4.6031	20
N44M4W111G	4.0000	4.6031	25
Barker88M8	4.0000	4.6040	44
N88M8W111ZC	6.2857	4.6040	44
N88M8W110ZC	6.7692	4.6040	37
N88M8W111G	5.5000	4.6040	46

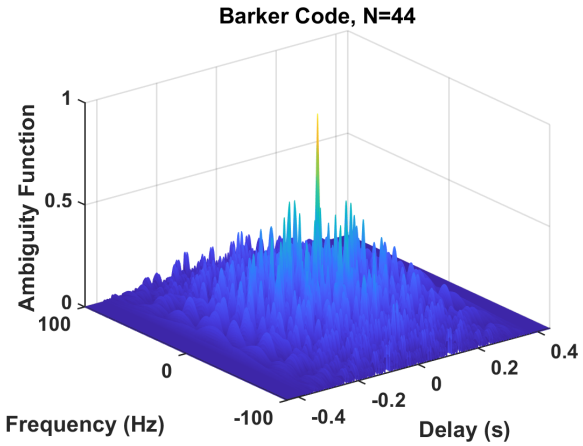
This similarity in SER performance of the optimized sequences can be attributed to the fact that the autocorrelation and cross-correlation characteristics of the optimized sequences are similar. The suppression of side lobes in these sequences leads to similar autocorrelation properties, which, in turn, results in comparable performance when the demodulation is performed using correlation-based methods.

In this study, nested Barker sequences are selected as the primary baseline due to their well-known ambiguity function characteristics, low implementation complexity, and frequent use in pulsed radar waveform designs. While other sequence families such as Zadoff-Chu, Golay, and Gold sequences are typically designed under different signal structure assumptions, such as constant-envelope transmission or complementary coding, which can be used to design an ISAC waveform. A comprehensive performance comparison with these advanced sequence families is considered an important direction for future work.

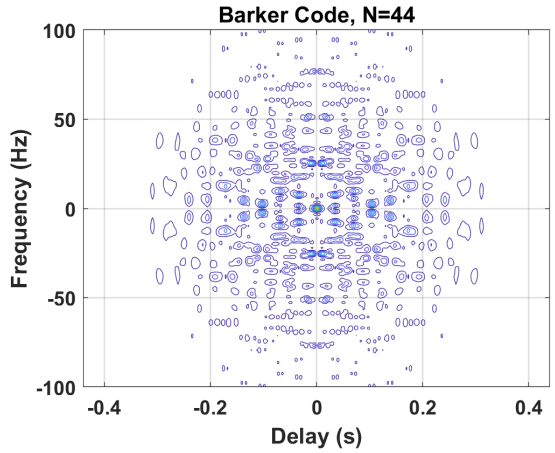
V. CONCLUSION AND FUTURE WORK

In this study, we considered a usage concept where both radar and communication functions can be realized using a joint waveform in ISAC systems. We used pulsed signals with Barker intra-pulse modulation as the ISAC joint waveform. After mathematically expressing the ambiguity function of the ISAC-Barker waveform, we performed ambiguity function analyses using both mathematical expressions and signal simulations. We observed that the PSLR of the $B44$ Barker sequences we proposed for $M = 4$ is somewhat lower compared to the optimized biphase codes under the same Pulse Width (PW), PRI, and bandwidth conditions.

Furthermore, we performed optimizations to improve the communication performance by using biphase codes. Specifi-

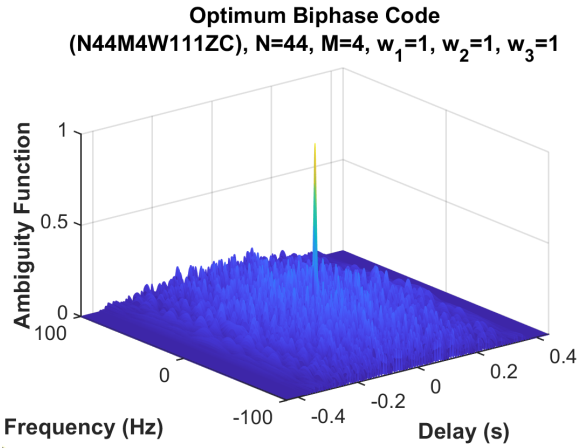


(a) Ambiguity function

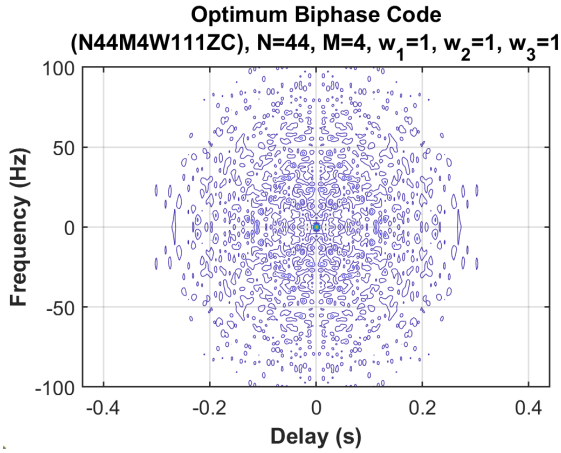


(b) Contour plot for the Barker code (N=44)

Figure 2. (a) Ambiguity function. (b) Contour plot for the Barker code (N=44).



(a) Ambiguity function



(b) Contour plot for the optimum biphase code, $w_1 = 1$, $w_2 = 1$, $w_3 = 1$ (N=44, M=4)

Figure 3. (a) Ambiguity function. (b) Contour plot for the optimum biphase code, $w_1 = 1$, $w_2 = 1$, $w_3 = 1$ (N=44, M=4).

cally, we applied genetic algorithms to optimize the codes for different objective functions: Zero Cut, and Global optimization. We compared the results of these optimized codes to the traditional nested Barker code. The optimization led to significant improvements in the side lobe suppression, which in turn enhanced the radar performance as well as the communication performance.

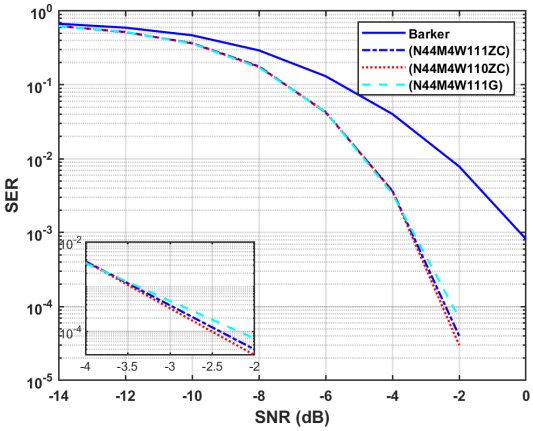
When analyzing the SER for the optimized codes, we observed that their performance was notably better than the Barker code, especially in terms of autocorrelation and cross-correlation characteristics. The optimized codes exhibited better autocorrelation properties, resulting in reduced side lobes and improved communication performance. However, the SER performance of the three optimization schemes (Global, Zero Cut with $w_1 = 1, w_2 = 1, w_3 = 1$, and Zero Cut with $w_1 = 1, w_2 = 1, w_3 = 0$) was found to be very similar for both $N = 44, M = 4$ and $N = 88, M = 8$ codes. Minor differences in performance were observed only at lower SNR levels. The optimized sequences demonstrate improved SER performance compared to the Barker code, due to their enhanced autocorrelation and cross-correlation characteristics.

It is important to emphasize that the communication-over-sensing paradigm considered in this work is not primarily intended for high-throughput data transmission, as targeted in enhanced mobile broadband scenarios. Instead, the main objective is to preserve sensing performance while enabling reliable exchange of sensing-related information, such as target detection results, tracking data, and control or coordination messages among distributed sensing nodes. Within this operational concept, relatively low data rates (e.g., 2–3 bits per Pulse Repetition Interval) are sufficient.

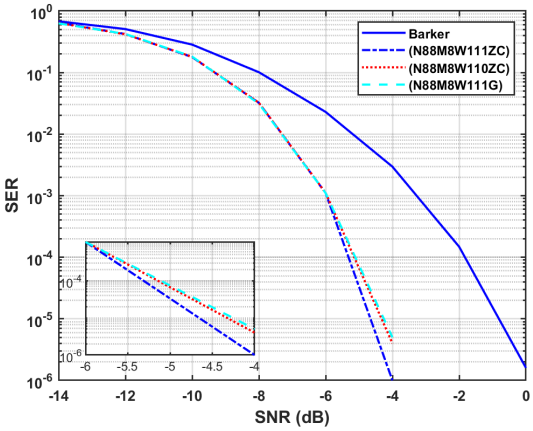
Future research can investigate the extension of the optimization to multi-level phase codes, which may provide further improvements in radar and communication performance.

In conclusion, the optimization of biphase codes led to improved radar and communication performance compared to the nested Barker code. The results demonstrated the effectiveness of the optimization methods in enhancing both the radar and communication capabilities in ISAC systems.

 ISBN: 978-1-68558-360-6



(a) SER $N = 44$, $M = 4$



(b) SER $N = 88$, $M = 8$

Figure 4. SER performance for different code lengths and symbol counts.

REFERENCES

[1] F. Liu, C. Masouros, A. P. Petropulu, H. Griffiths, and L. Hanzo, "Joint radar and communication design: Applications, state-of-the-art, and the road ahead", *IEEE Transactions on Communications*, vol. 68, no. 6, pp. 3834–3862, 2020. DOI: 10.1109/TCOMM.2020.2973976.

[2] Y. Cui, F. Liu, X. Jing, and J. Mu, "Integrating sensing and communications for ubiquitous iot: Applications, trends, and challenges", *IEEE Network*, vol. 35, no. 5, pp. 158–167, 2021. DOI: 10.1109/MNET.010.2100152.

[3] Z. Cui, J. Hu, J. Cheng, and G. Li, "Multi-domain noma for isac: Utilizing the dof in the delay-doppler domain", *IEEE Communications Letters*, vol. 27, no. 2, pp. 726–730, 2023. DOI: 10.1109/LCOMM.2022.3228873.

[4] L. Giroto de Oliveira et al., "Joint radar-communication systems: Modulation schemes and system design", *IEEE Transactions on Microwave Theory and Techniques*, vol. 70, no. 3, pp. 1521–1551, 2022. DOI: 10.1109/TMTT.2021.3126887.

[5] D. Garmatyuk, J. Schuerger, and K. Kauffman, "Multifunctional software-defined radar sensor and data communication system", *IEEE Sensors Journal*, vol. 11, no. 1, pp. 99–106, 2011. DOI: 10.1109/JSEN.2010.2052100.

[6] X. Lv, J. Wang, Z. Jiang, and W. Jiao, "A novel papr reduction method for ocdm-based radar-communication signal", in *2018 IEEE MTT-S International Microwave Workshop Series on 5G Hardware and System Technologies (IMWS-5G)*, 2018, pp. 1–3. DOI: 10.1109/IMWS-5G.2018.8484372.

[7] M. B. Alabd, L. G. de Oliveira, B. Nuss, W. Wiesbeck, and T. Zwick, "Time-frequency shift modulation for chirp sequence based radar communications", in *2020 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*, 2020, pp. 1–4. DOI: 10.1109/ICMIM48759.2020.9298987.

[8] S. Jha, N. M. Balasubramanya, B. K. Chalise, and M. G. Amin, "Performance analysis of lora waveform for joint communications and radar system", *IEEE Transactions on Radar Systems*, vol. 1, pp. 243–248, 2023. DOI: 10.1109/TRS.2023.3294799.

[9] S. Şahin and T. Girici, "Phase coded waveforms for integrated sensing and communication systems", *IET Radar, Sonar and Navigation*, 2024. DOI: 10.1049/rsn2.12661.

[10] M. Weiß, "A novel waveform for a joint radar and communication system", in *2022 23rd International Radar Symposium (IRS)*, 2022, pp. 259–263. DOI: 10.23919/IRS54158.2022.9904976.

[11] M. R. Mousavi and S. Ludwig, "Ambiguity function analysis of hyperbolic fractional fourier transform in joint radar and communication applications", in *2024 International Radar Symposium (IRS)*, 2024, pp. 206–211.

[12] M. Jamil, H.-J. Zepernick, and M. I. Pettersson, "On integrated radar and communication systems using oppermann sequences", in *MILCOM 2008 - 2008 IEEE Military Communications Conference*, 2008, pp. 1–6. DOI: 10.1109/MILCOM.2008.4753277.

[13] I. P. Eedara, A. Hassanien, M. G. Amin, and B. D. Rigling, "Ambiguity function analysis for dual-function radar communications using psk signaling", in *2018 52nd Asilomar Conference on Signals, Systems, and Computers*, 2018, pp. 900–904. DOI: 10.1109/ACSSC.2018.8645328.

[14] I. P. Eedara, M. G. Amin, and A. Hoorfar, "Optimum code design using genetic algorithm in frequency hopping dual function mimo radar communication systems", in *2020 IEEE Radar Conference (RadarConf20)*, 2020, pp. 1–6. DOI: 10.1109/RadarConf2043947.2020.9266623.

[15] M. Jamil, H.-J. Zepernick, and X.-S. Yang, "Sequence optimization for integrated radar and communication systems using meta-heuristic multiobjective methods", in *2017 IEEE Radar Conference (RadarConf)*, 2017, pp. 502–507. DOI: 10.1109/RADAR.2017.7944255.

[16] S. Şahin and T. Girici, "A novel radar receiver for joint radar and communication systems", in *2023 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET)*, 2023, pp. 61–66. DOI: 10.1109/ICRAMET60171.2023.10366608.

[17] B. R. Mahafza, *Radar Signal Analysis and Processing Using Matlab*. Taylor & Francis Group, 2009.

[18] N. Levanon, "The periodic ambiguity function — its validity and value", in *2010 IEEE Radar Conference*, 2010, pp. 204–208. DOI: 10.1109/RADAR.2010.5494626.

[19] A. De Maio and A. Farina, "Waveform diversity: Past, present, and future", *RTO-EN-SET-119*, 2010.

YOLO-Based Deep Learning models for Real-Time Volleyball Action Recognition

Brandon Labio
The Harker School, San Jose, CA, USA
e-mail: brandonlabio@gmail.com

Atul Dubey
AIClub Research Institute, CA, USA
e-mail: atul.dubey@aiclub.world

***Abstract*—Analyzing player performance through highlights and statistics is critical for volleyball coaching, but remains a labor-intensive bottleneck. Current methods are often hindered by impractical deployment, inefficiency, and lack of portability. To overcome these limitations, this study introduces a transportable, deep-learning-based object detection model for real-time identification of volleyball actions. Using a dataset of 1,236 images, models were trained based on both YOLOv8 and YOLO11. The YOLOv8 model yielded superior performance, achieving a mAP50 value of 87.8%, and was optimized at a learning rate of 0.0005 over 80 epochs. This model was subsequently tested using the test dataset and achieved a mAP50 of 79.8%. The system was then deployed on a Raspberry Pi to evaluate real-time feasibility, reaching a latency of 1 FPS. These findings establish a critical baseline for the development of live automated highlight generation systems and are essential for efficient creation of highlights and analysis of performance.**

Keywords-Volleyball action detection; deep learning; YOLO11; YOLOv8

I. INTRODUCTION

In volleyball, understanding players' strengths, weaknesses, and playing styles is essential for coaches to support player development and plan for future matches. Players also rely on performance highlights to showcase their abilities for college recruiting [1]. Both performance analysis and highlight creation require careful review of match recordings. However, manually analyzing full-length games to extract key plays and generate statistics is inefficient and time-consuming.

In addition to requiring substantial effort, manual review is prone to human error, as important actions may be overlooked or misclassified. Therefore, there is a clear need for automated systems capable of generating highlights and match statistics accurately and efficiently. While solutions such as Balltime [2] by Hudl provide action detection and automated statistics using a proprietary computer vision model, they lack the ability to generate full game highlights, limiting their usefulness for comprehensive performance evaluation.

Our unique approach focuses on creating both highlights and statistics in real time, enabling coaches to guide players during matches and also for highlights to be ready immediately after the game for others to watch. Our model detects 6 different volleyball actions: blocking, digging, passing, serving, setting, and blocking. In addition, the device tracks the detection time, which helps generate highlights for specific actions and statistics on the frequency of different actions.

To achieve this, transfer learning was performed on pretrained YOLOv8 and YOLO11 models to identify different actions from a recording of a volleyball game. Their performance was compared with respect to the mAP50 value and latency during inference on a resource-constrained device like Raspberry Pi. The device we built provides immediate access to game insights and supports better decision-making.

The rest of the paper is structured as follows: Section 2 reviews related works in similar domains and contrasts them with our approach. Section 3 describes the tools, techniques, and methods used to conduct this research. Section 4 presents the results, followed by a discussion of findings and potential directions for future work.

II. RELATED WORKS

In the domain of action recognition, different techniques and models have been explored, which have achieved some levels of success for various datasets.

A. Convolutional Neural Network-based approach

In [3], for real-time action detection, CNNs have been employed to decompose complex human activities into three distinct levels, yielding mean average precision scores of 76.6% for frame-based and 83.5% for video-based evaluations on the ICVL (Intelligent Computer Vision and Learning Lab) Video Surveillance Dataset. This approach highlights the value of breaking down complex actions for enhanced representation and detection.

In another study, researchers explored multi-sport action recognition using methods such as 3D CNNs, two-stream approaches, and SlowFast. These techniques were tested against a dataset featuring 3,200 video clips with annotated actions, achieving a frame-based mAP of 27.72% and varying video-based mAPs. Such results emphasize the capability of these models to handle complex, spatio-temporally localized actions [4].

In [5], researchers demonstrated the use of Kinect and CNN classifiers to interpret geometric limb relations. This method achieved an average recognition rate of 93% across several datasets, showcasing the potential of combining skeletal data with deep learning for improved action identification.

For volleyball action recognition, dual-stream CNNs were employed within the DNet model to process both spatial and temporal data. This approach led to classification accuracies of 94.12% on the UCF101 dataset, illustrating how combining different data streams can enhance model performance [6]. Another study focused on applying Gaussian Mixture Models with multi-resolution 3D CNNs to recognize movements from volleyball game videos, improving accuracy by 3.3% while reducing complexity by 33.6% [7].

B. YOLO based approach

Volleyball setting tactic detection has benefited from the use of YOLO-based player detection and multi-object tracking systems. These were applied to RGB images from national team matches to classify set types, resulting in improved accuracies under both clean and noisy conditions [8]. This demonstrates the effectiveness of integrating object detection and tracking with action classification.

C. CNN with vision transformer-based approach

Meanwhile, a study leveraging CNNs and Transformers, such as VideoMAE with ViT-B, focused on detecting human interactions in sports. This method utilized spatio-temporal context reasoning, achieving mAP scores for classification at 10.69% and detection at 4.93%, which highlights the complexity of modeling interactions within dynamic environments [9].

D. Long Short-Term Memory-based approach

Lastly, for sports motion recognition in the MARS dataset, the combination of millimeter-wave radar with Kalman filtering and LSTMs demonstrated robust results. By producing 5D point clouds and reconstructing models to capture temporal features, the approach attained accuracy rates of 95.6% outdoors and 97.9% indoors, underscoring its effectiveness for precise action recognition [10]. Collectively, these studies reflect the diverse approaches and outcomes in the evolving field of sports action recognition.

E. Our approach - YOLO model deployed in an edge device

Building upon the limitations observed in previous works, this project addresses key challenges in efficiency and practical deployment within volleyball action recognition. Previous studies using dual-stream CNNs and GMM-based 3D CNNs [6][7] achieved solid accuracy but relied on computationally heavy models that limited real-time application. Similarly, YOLO-based detection systems [8] improved volleyball setting recognition but did not provide automated data logging or real-time feedback. The firmware used in this project integrates a lightweight YOLOv8-based detection pipeline capable of on-device inference, enabling efficient, real-time processing of volleyball actions. Additionally, in contrast to earlier CNN- and Kinect-based methods [3][4][5], this system enhances data management and evaluation consistency by timestamping detections, saving results automatically to CSV files, and overlaying detections on live video feeds. Furthermore, by exposing both the live feed and detection data over an HTTP connection, the project provides a practical framework that bridges the gap between research-oriented recognition models and real-world volleyball analytics applications.

III. Material and Methods

The following describes the dataset, model architecture, training procedure, and hardware implementation used to develop and evaluate the volleyball action detection system.

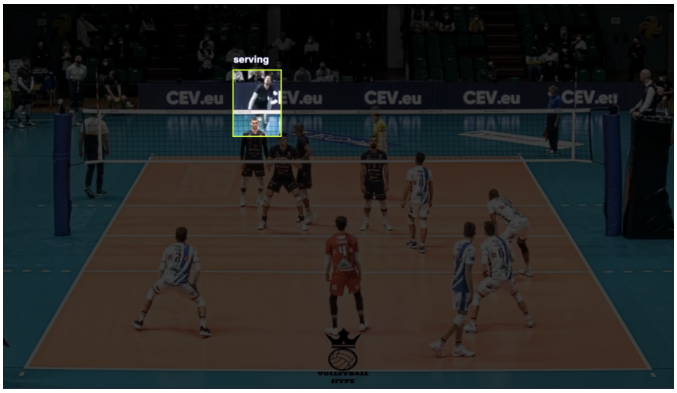


Figure 1. Example of Image From Dataset

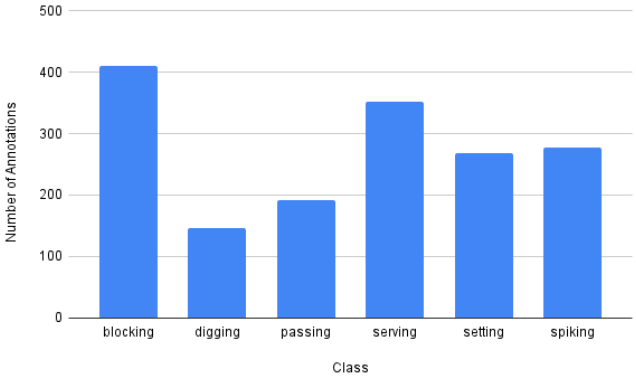


Figure 2. Distribution of annotations across images

A. Dataset

The dataset used for building deep learning models has been taken from an open source repository in Roboflow Universe [11]. It contains 1,236 images and 1646 total annotations from the same match, with there being 6 different types of annotations: blocking, digging, passing, serving, setting, and spiking. An example of an image from the dataset is presented in Figure 1. The overall distribution of annotations across images is presented in Figure 2. There is no need for negative examples since that would primarily be needed for detecting "no action," but actions are present in every frame.

B. Deep Learning models

1) YOLOv8 [12]: YOLOv8 is an efficient and lightweight object detection model, capable of real-time detection. YOLOv8 demonstrates enhanced accuracy over its predecessors by incorporating advancements in feature extraction and model architecture, resulting in better precision and recall. We have chosen to experiment with this model due to its real-time detection capability and ability to deploy on an edge device.

2) YOLO11 [13]: YOLO11 achieves a higher mean Average Precision (mAP) on the COCO dataset while using 22% fewer parameters than YOLOv8, making it computationally efficient without compromising accuracy. We chose this as it is the latest model in the YOLO series and is considered to be an improvement over YOLOv8.

C. Model Training

As we can see in Figure 3, before the start of the experiment, the dataset was split into 3 different parts: train (936), validation (150), and test (150). The respective splits were around 75%, 12.5%, 12.5%, combining the common 80/10/10 and 70/15/15 strategies, designed to balance the need for sufficient training data with the need for reliable, unbiased performance estimation. After the splitting, the dataset was exported from Roboflow in YOLOv8 format. The Ultralytics Python library with Google Colab was used for performing transfer learning on the YOLOv8 and YOLO11 models to customize them for detecting volleyball actions. Experiments were conducted by varying the number of epochs from 50 to 100 and adjusting the learning rates between 0.00001 and 0.01. These hyperparameters were chosen as a starting point. However, during the training, we realized that even with lower learning rates and higher epochs, there would not be much improvement in performance. The best-performing model was subsequently evaluated using the test dataset. The code for the project is available for reference in [14].

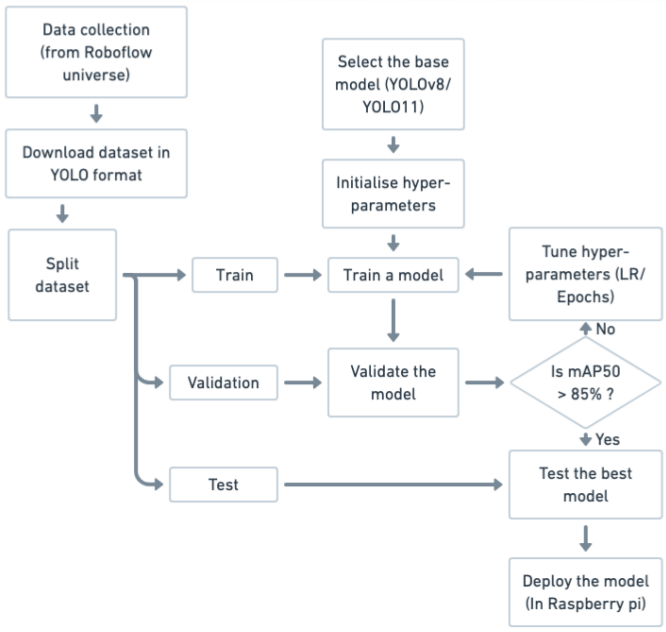


Figure 3. Machine Learning Pipeline

D. Hardware

A Raspberry Pi and a Logitech Brio 101 Full HD 1080p camera were used for testing the best model. The Raspberry Pi is a low-cost single-board computer with enough computational power to run an object detection model on the device, and the Logitech Brio 101 Full HD 1080p camera helped to capture video frames for input. The power supply provided a stable 5V/3A power source to the Raspberry Pi, ensuring reliable operation.

E. Hardware Architecture

The device consists of a Raspberry Pi 5 module, a USB camera, and a power source (Currently a power adapter, but can be changed to a powerbank for a portable setup). The camera connects to one of the USB ports of the Raspberry Pi and is powered through the port itself. Figure 4 shows connections between various components of the device.

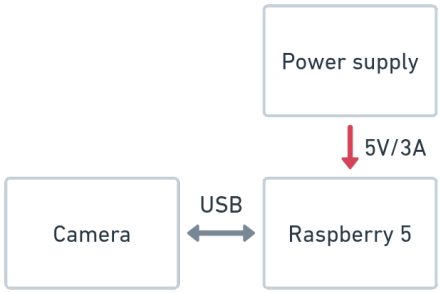


Figure 4. Hardware architecture diagram

F. Firmware

In the firmware of the device, first, the camera is initialized. After that, the YOLO11 model is loaded using the Ultralytics Python library. The camera begins capturing frames, which are subsequently processed by the model for detection. Detected actions are recorded in a CSV file along with a timestamp. A web server is implemented to enable viewing of the live feed and detection results over a local IP address. The web portal, hosted by the Raspberry Pi, displays both the live stream and a table of previous detections. Figure 5 illustrates the logic flow incorporated in the firmware.

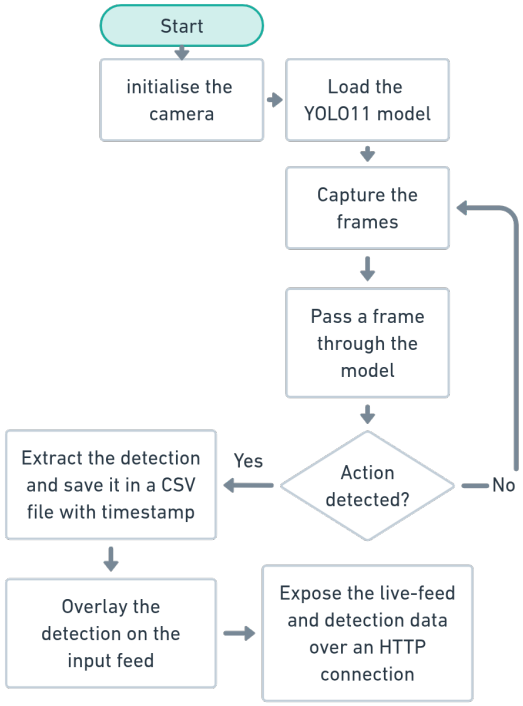


Figure 5. Firmware flowchart

IV. Results

The following presents experimental findings for YOLOv8 and YOLO11 models, highlighting validation performance, convergence behavior, and detection accuracy across multiple evaluation metrics.

A. YOLOv8

A total of 30 experiments were performed by varying epochs between 50 and 100 and learning rates between 0.00001 and 0.01. The best validation mAP50 of 87.8% was achieved at a learning rate of 0.0005 with 80 epochs. Figure 6 represents the variation of mAP50 values on the validation set across different experiments performed.

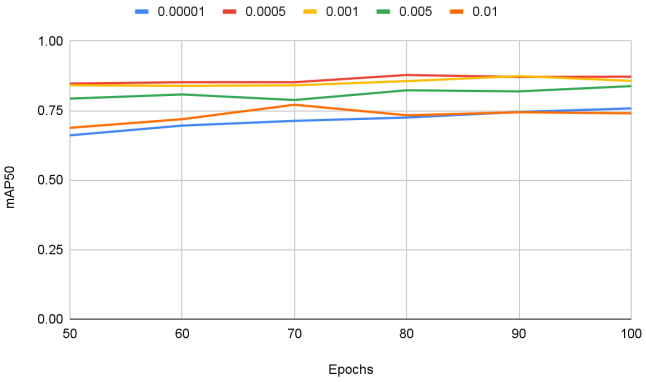


Figure 6. YOLOv8 - mAP50 vs epochs for various learning rates

Figure 7 shows the validation losses and evaluation metrics during training of a YOLOv8 model.

The following metrics were used to assess the performance of the detection model:

- **Box loss:** Box loss measures how accurately a predicted bounding box aligns with the ground-truth object location. It penalizes discrepancies in box coordinates—typically width, height, and center position—and encourages the model to produce tighter, well-localized boxes.
- **classification loss:** Classification loss evaluates how well the model assigns the correct class label to each detected object. It penalizes incorrect or uncertain predictions of object categories, guiding the network to improve its discriminative capability.
- **Distribution Focal Loss:** Distribution Focal Loss (DFL) is a localization loss used in modern object detectors to improve the precision of bounding-box regression. Instead of predicting a single continuous box value, DFL models each box coordinate as a discrete probability distribution over a set of bins. The loss encourages the predicted distribution to place higher probability mass near the true target location, enabling finer-grained and more accurate localization. This formulation reduces quantization error and enhances the model's ability to learn high-resolution bounding-box boundaries, particularly when combined with IoU-based localization losses.
- **mAP50:** mAP50 refers to the mean Average Precision computed at an Intersection over Union (IoU) threshold of 0.50. It measures how well the detector identifies objects and aligns predicted boxes with ground-truth boxes. A prediction is considered correct if IoU > 0.50.
- **mAP50-95:** mAP50–95 is the mean Average Precision computed over ten IoU thresholds, from 0.50 to 0.95 in steps of 0.05. This metric provides a comprehensive evaluation of detection performance by assessing both coarse and highly precise localization. Compared to mAP50, it is significantly more stringent, rewarding models that produce accurate bounding boxes across varying levels of overlap.

The box loss, classification loss, and DFL (Distribution Focal Loss) all decrease sharply in the early epochs and then stabilize, indicating the model is learning to localize and classify objects effectively. The mAP@0.5 metric increases rapidly within the first 20 epochs and then plateaus around 0.85–0.9, showing strong detection performance at a lenient IoU threshold. Similarly, the stricter mAP@0.5–0.95 rises steadily and stabilizes near 0.6, reflecting good overall precision-recall tradeoff across IoU thresholds.

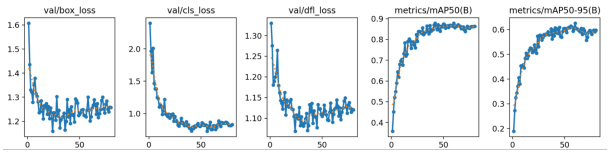


Figure 7. YOLOv8 - (a)variation of box-loss across epochs, (b) variation of class loss across epochs, (c) variation of distributed focal loss across epochs, (d)variation of mAP50 across epochs, and (e) variation of mAP50-95 across epochs

B. YOLO11

A total of 30 experiments were performed by varying epochs between 50-100 and nd learning rates between 0.00001 - and0.01. The best validation mAP50 of 87.1% was achieved at a learning rate of 0.0005 with 60 epochs. Figure 8 represents the variation of mAP50 values on the validation set across different experiments performed.

Figure 9 shows the validation losses and evaluation metrics during training of a YOLO11 model. The model converged

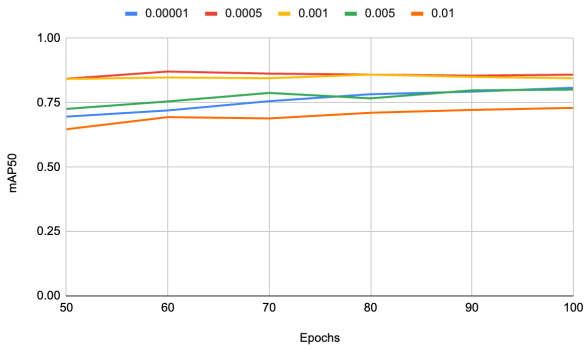


Figure 8. YOLO11 - mAP50 vs epochs for various learning rates

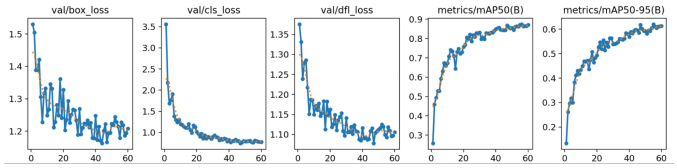


Figure 9. YOLO11 - (a)variation of box-loss across epochs, (b) variation of class loss across epochs, (c) variation of distributed focal loss across epochs, (d)variation of mAP50 across epochs, and (e) variation of mAP50-95 across epochs

well, as shown by a steady decline in validation losses and the corresponding rise in accuracy metrics. Specifically, the box loss, classification loss, and distribution focal loss decreased notably over the epochs, indicating better bounding box alignment and class prediction. The detection accuracy showed significant gains, with mAP50 increasing from 0.25 to approximately 0.88 and the stricter mAP50-95 rising from 0.15 to about 0.62.

C. Summary and Test Results

TABLE I. VALIDATION RESULT SUMMARY

Model	Best mAP50 (%)
YOLOv8	87.8
YOLO11	87.1

Among the models presented in Table I, YOLOv8 demonstrated the highest performance and was subsequently evaluated using the test dataset, achieving a mAP50 of 79.8%. Table II shows the detailed test results for each of the detected classes.

TABLE II. TEST RESULTS FOR THE BEST YOLOV8 MODEL

Class	Box Precision	Box Recall	mAP@50	mAP@50-95
blocking	0.968	0.871	0.925	0.682
digging	0.667	0.462	0.543	0.418
passing	0.773	0.548	0.643	0.490
serving	1.000	0.972	0.986	0.689
setting	0.800	0.800	0.819	0.526
spiking	0.964	0.750	0.870	0.594
Average	**0.862**	**0.734**	**0.798**	**0.567**

Figure 10 shows the Precision–Recall (PR) curve for the YOLOv8 model on the test set.

Precision and recall help reveal different types of detection errors: low precision indicates many false positives, meaning the model is predicting objects that are not present or assigning the wrong class, while low recall indicates many false negatives, where the model fails to detect objects that actually exist. Mispredictions contribute to both errors—counting as a false positive for the incorrect class and a false negative for the true class—thereby reducing both precision and recall.

Figure 10 illustrates how well the model balances precision and recall across different volleyball actions. The results show that the model performs exceptionally well on serving and blocking, with both achieving very high precision and recall. Spiking and setting also perform strongly, though with slightly lower precision at higher recall levels. In contrast, passing and especially digging (0.543) are more challenging, with lower overall precision-recall areas, suggesting room for improvement in detecting these actions.

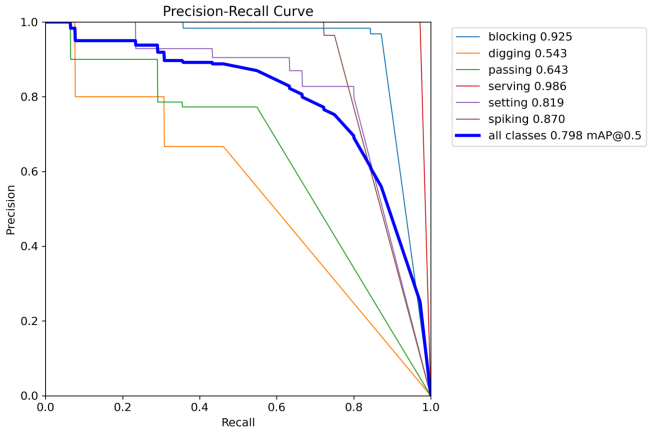


Figure 10. Precision-recall curve for the test results obtained from the best YOLOv8 model

D. Device Test Results

The device with the best model deployed was tested on simulated scenarios as well as on the recorded videos of the game. Currently, we're able to achieve a latency of 1fps with the PyTorch model deployed using the Ultralytics library.

V. DISCUSSION

The comparative analysis of YOLOv8 and YOLO11 performance across varying learning rates reveals nuanced differences in their training efficiency and final performance. In general, a large learning rate causes the model to skip the global minimum, resulting in lower accuracy, and a very low learning rate requires more epochs to reach the global minimum. According to Figures 6 and 7, the YOLOv8 model achieves its peak mAP50 of 0.878 with a learning rate of 0.001 at 80 epochs, and its performance at the same learning rate remains strong across the training duration. In contrast, the YOLO11 model reaches a similar peak mAP50 of 0.871 with a learning rate of 0.0005, but it achieves this peak significantly earlier, at just 60 epochs. This suggests that while both models can reach a comparable level of high performance, YOLO11 converges more rapidly to its optimal solution. Furthermore, the YOLO11 model's performance with a learning rate of 0.0005 shows greater stability from epoch 60 onwards, whereas the YOLOv8's performance at its peak learning rate of 0.001 shows a slight decrease after reaching its highest point at epoch 90.

As evident from Table II, the best YOLOv8 model exhibits a varied performance across different classes. The model excels at detecting 'serving', achieving a perfect box precision of 1.0 and a recall of 0.972, leading to a near-perfect mAP50 of

0.986. It also performs exceptionally well for 'blocking' and 'spiking', with mAP50 scores of 0.925 and 0.87, respectively, indicating a high level of accuracy for these actions. However, the model shows a notable drop in performance for 'digging' and 'passing', with mAP50 values of 0.543 and 0.643. This reduced performance is likely due to lower recall scores (0.462 for digging and 0.548 for passing), suggesting the model struggles to identify all instances of these classes. The overall average mAP50 for the model is 0.798, which, while respectable, is pulled down by the poor performance on these specific classes. These findings highlight the need for further model refinement, perhaps through additional data augmentation or architectural modifications, to improve detection of the less-represented or more challenging classes.

Furthermore, several challenges were encountered during the testing process. Establishing stable connections between the YOLO models, live camera input, and the HTTP server also posed difficulties, as latency and inconsistent frame rates occasionally disrupted real-time detection. Finally, the frame-by-frame processing approach, while efficient for isolated detections, limited temporal context awareness and occasionally led to misclassifications when consecutive actions appeared visually similar. These challenges highlight potential directions for improvement, with the most prominent next step being the incorporation of temporal modeling techniques to improve future iterations.

VI. Conclusion and Future Work

This study confirms that deep learning-based object detection is a highly effective approach for analyzing volleyball actions. Our optimized YOLOv8 model achieved an mAP50 value of 0.798, which is an acceptable result for action detection. While the current deployment on Raspberry Pi delivers accurate recognition, minimizing latency remains a priority. Future iterations will address this by converting the model to more compact formats (quantization) or utilizing external edge-TPU accelerators. Additionally, we aim to enhance the system's analytical depth by incorporating spatio-temporal features via LSTM networks and developing an ensemble architecture. This will enable simultaneous action recognition and specific player identification, paving the way for fully automated and comprehensive match reporting.

Furthermore, the current model is limited to detecting only actions but not identifying the players who did the actions. In the future, we can create an ensemble of two models where we can identify the players as well as the actions done by them. This can significantly improve the quality of reports and highlights generated.

References

[1] B. Schunzel, *Video in the Recruiting Process: Simple and Significant | Junior Volleyball Association — jvavolleyball.org*, https://jvavolleyball.org/video-recruiting-process-simple-significant/, [Accessed 12-10-2025].

[2] *Balltime from Hudl is the leading AI Volleyball Solution • Hudl — hudl.com*, https://www.hudl.com/en_gb/products/balltime, [Accessed 10-11-2025].

[3] C.-B. Jin, S. Li, and H. Kim, "Real-time action detection in video surveillance using sub-action descriptor with multi-cnn," *arXiv preprint arXiv:1710.03383*, 2017, [Accessed 12-10-2025].

[4] Y. Li et al., "Multisports: A multi-person video dataset of spatio-temporally localized sports actions," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, [Accessed 12-10-2025], 2021, pp. 13 536–13 545.

[5] J. Tang, "An action recognition method for volleyball players using deep learning," *Scientific Programming*, vol. 2021, no. 1, p. 3 934 443, 2021, [Accessed 12-10-2025].

[6] B. Li and M. Tian, "Volleyball movement standardization recognition model based on convolutional neural network," *Computational Intelligence and Neuroscience*, vol. 2023, no. 1, p. 6 116 144, 2023, [Accessed 12-10-2025].

[7] H. Wang, X. Jin, T. Zhang, and J. Wang, "Convolutional neural network-based recognition method for volleyball movements," *Heliyon*, vol. 9, no. 8, 2023, [Accessed 12-10-2025].

[8] H. Xia et al., "Advanced volleyball stats for all levels: Automatic setting tactic detection and classification with a single camera," in *2023 IEEE International Conference on Data Mining Workshops (ICDMW)*, [Accessed 12-10-2025], IEEE, 2023, pp. 1407–1416.

[9] T. Wu, R. He, G. Wu, and L. Wang, "Sportshhi: A dataset for human-human interaction detection in sports videos," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, [Accessed 12-10-2025], 2024, pp. 18 537–18 546.

[10] Z. Zhang, Y. Tian, and J. Qi, "Volleyball technical action recognition based on cnn-lstm," *Discover Artificial Intelligence*, vol. 5, no. 1, p. 188, 2025, [Accessed 12-10-2025].

[11] *Volleyball_dataset Model by MIKHAIL KLYUKIN — universe.roboflow.com*, https://universe.roboflow.com/mikhail-klyukin/volleyball_dataset/images/KZi1uOP1ZvEcztaMm7Ed, [Accessed 10-11-2025].

[12] R. Varghese and M. Sambath, "Yolov8: A novel object detection algorithm with enhanced performance and robustness," in *2024 International conference on advances in data engineering and intelligent computing systems (ADICS)*, [Accessed 12-10-2025], IEEE, 2024, pp. 1–6.

[13] R. Khanam and M. Hussain, "Yolov11: An overview of the key architectural enhancements," *arXiv preprint arXiv:2410.17725*, 2024, [Accessed 12-10-2025].

[14] B. Labio, *GitHub - brandonl-byte/VolleyballActionDetection — github.com*, https://github.com/brandonl-byte/VolleyballActionDetection, [Accessed 12-10-2025], 2025.

Power Spectral Density of the Quantization Noise in Block-Floating-Point FFT

Gil Naveh
Toga Research Center
Huawei Technologies Co. Ltd
Tel-Aviv, Israel
Email: gil.naveh@huawei.com

***Abstract*— The Power Spectral Density (PSD) of quantization noise in fixed-point Fast Fourier Transform (FFT) implementations is traditionally assumed to be white. However, this assumption often fails in practical hardware, particularly in CPUs and DSPs employing Block-Floating-Point (BFP) arithmetic. Due to hardware efficiency constraints, these processors frequently utilize Rounding-Half-Up (RHU), a native operation that introduces an inherent statistical bias. Throughout the BFP-FFT stages, these cumulative biases propagate to the output, coloring the quantization noise and creating frequency-dependent power variations. This correlation can severely degrade performance in sensitive applications such as OFDM modems and high-fidelity audio codecs. This paper analyzes the impact of rounding bias on the noise spectrum and provides a comparative study of Decimation-In-Time (DIT) vs. Decimation-In-Frequency (DIF) topologies. We demonstrate that the DIF architecture is significantly more sensitive to this bias. Finally, we present three mitigation schemes, showing that convergent rounding, when supported by the hardware, effectively restores the output noise to a near-white spectral state.**

Keywords - BFP; Fixed Point; DIT; DIF; PSD; SQNR;

I. INTRODUCTON

The Fast Fourier Transform (FFT) serves as an important tool in many signal processing applications, such as spectral analysis, filtering, audio coding, Digital-Subscriber-Line (DSL) modems [1] wireless Orthogonal-Frequency-Division-Modulation (OFDM) modems [2] and advanced fiber optic modems [3].

Finite-word-length effects (denoted hereafter also as quantization noise) have substantial effect on the accuracy performance of FFTs. This is a result of the native characteristic of the FFT in which quantization noise that is added at the output of each stage of the FFT is accumulated toward the FFT output. Since the maximal value at each stage's output grows as we proceed with the stages [4], in many hardware implementations, the performance degradation due to the quantization noise is mitigated by adapting the register size at each stage to accommodate the signal growth [5][6]. On the other hand, in software implementations (as in CPUs and Digital Signal Processors - DSPs), increasing the bit-width is not possible. For those cases, a dynamic-scaling BFP based schemes are commonly used.

The averaged effects of the finite-word-length in FFT processing has been deeply investigated for various use cases and applications [7]-[9]. The average SQNR is an informative metric when the output quantization noise spreads evenly over all the FFT's output nodes. In practice, however, this is not necessarily the case. The quantization noise that is being added at each stage is a result of rounding the result of some arithmetic operation (multiplication, or addition followed by right shift). Usually, when the rounding is non-biased, indeed the output quantization noise spreads almost evenly over the FFT output nodes. Non-biased rounding also results in lower average noise power at the FFT output. Since non-biased rounding has higher hardware cost, common processors use a low-cost, hardware-friendly rounding, which is a biased rounding. The inherent bias in hardware-friendly rounding comes into effect in rounding half-way numbers. Although seemingly minor, this bias significantly alters the spectral shape of the quantization noise at the BFP-FFT output. Specifically, it causes the noise to become colored, meaning the noise power varies across frequency. Such non-white noise can have detrimental effects on system performance; for instance, standard OFDM channel estimation algorithms typically operate on the assumption that the additive noise is white and dominated by receiver thermal noise. When this assumption is violated by colored noise, the accuracy of the channel estimation degrades, potentially leading to increased bit-error rates in high-order modulation schemes.

The fact that biased rounding degrades the quality of FFT processors is not new. In many dedicated hardware FFTs, a convergent rounding, [10], is deployed as a non-biased rounding as in e.g., [9].

In this paper, we analyze the effects of the biased rounding used in common processors on the quantization noise at the FFT output. We focus on the PSD of the output quantization noise and compare the DIT and DIF in that regard. We show that DIT topology leads to better quantization performance over common fixed-point processors. It is also shown that DIT remains favorable even when support for non-biased, convergent rounding, is added to those processors. We use radix-2 Cooley-Tuckey FFT to demonstrate those effects, but the analysis can be extended for any other FFT radix and topology.

The paper is organized as follows: Section II introduces the models used throughout the paper covering the DIT and DIF BFP-FFT models, the underline processor model, and the quantization noise models. Section III discusses the effects of the biased quantization in the DIT and DIF BFP-

FFT. The implications on the output PSD are provided in Section IV. Algorithms for reducing the effects of the biased rounding and for whitening the noise PSD are discussed in Section V, and conclusions are given in Section VI.

II. FFT, PROCESSOR AND QUANTIZATION NOISE MODELS

We relate to fixed-point representation of fractional datatypes. We assume a processor having registers of b bits (including sign) and accumulators of at least $B = 2b + 2$ bits. The numbers represented by the registers are in 2's complement representation and in the range $-1 \le x \le 1 - 2^{-(b-1)}$. The numbers represented by the accumulators are in the rage $-2^2 \le x < 2^2$. The width of the data stored to memory is always of b bits.

Our focus is of radix-2, BFP, Cooley-Tukey, DIT and DIF FFTs. The model of a finite-word-length radix-2 butterfly of the DIT-FFT is given in Fig.1 and of DIF-FFT in Fig. 2. In the DIT topology, the inputs loaded from the memory are first multiplied by the Twiddle Factors (TFs), w_N^{kn}, then added and subtracted by the radix-2 butterfly operation before being stored back to the memory. The ADD/SUB operation is modeled as the sum of the inputs multiplied by the butterfly internal coefficients (which are $\{1, -1\}$). The processing model that we will deal here with, is a model that is common to most DSPs and dedicated FFT processors. In this model the inputs x_n and the TFs, w_N^{kn}, are represented by b bits per component (b bits for the real component and b bits for the imaginary component) and are within the range of $[-1, 1 - 2^{-(b-1)}]$. When multiplied, the multiplication is spanned over $2b + 1$ bits (recalling that the TF multiplication is a complex multiplication). The bit-width of the butterfly's output can grow to span over up to B bits and then potentially scaled down by a factor of α, where we restrict α to be a power of 2. The scaled down butterfly output is quantized to b bits per component, via rounding, before being stored to memory. In the DIF topology the inputs loaded from memory are first added and subtracted by the butterfly operation and then fed (on one of the butterfly branches) into a TF multiplier. Since the bit-width may grow by one bit after the ADD/SUB operation, a down scaling followed by quantization may take place before the multiplication. Finally, after the TF multiplication, another quantization step is required before storing the results to the memory. The quantization operation is modeled as an additive noise v and u in the diagrams. The quantization model that is used by most common processors is the RHU [10], which is also known as hardware-friendly-

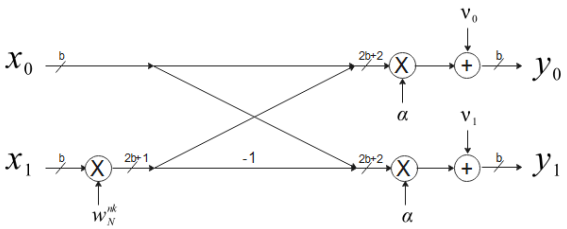


Figure 1:DIT butterfly model

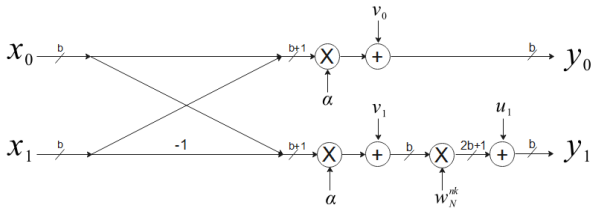


Figure 2: DIF butterfly model

rounding. The mathematical function of RHU rounding to b bits is

$$y = Q[s] \triangleq 2^{-b} \cdot \lfloor s \cdot 2^b + 0.5 \rfloor, \quad (1)$$

where $\lfloor a \rfloor$ is maximal integer lower than a and $s \in [-1{,}1 - 2^{-(b-1)}]$. The quantization error is $v = s - y$ and in the general case, when the bit-width of s is much larger than b, v is modeled as a uniformly distributed additive noise [11]

$$v \sim U[-2^{-b}, 2^{-b}), \quad (2)$$

and is statistically independent of s. As we deal here with finite-word-length, in fact v has a discrete distribution. However, for large enough b it is common to treat it as a zero mean continuous uniform distribution. As such its variance is

$$\sigma_v^2 = \frac{2^{-2(b-1)}}{12}. \quad (3)$$

In the cases that the bit-width of s is not much larger than b, the quantization noise does not behave as zero mean uniform Random Variable (RV) anymore [11]. This is also the case when a b bits number is scaled down by few bits before being quantized back to b bits. For example, relate to a b bits number, s, that is being scaled down by q bits. Since our reference numbering scheme is of fractions, the bit-width of the scaled down s is $b + q$ bits. Quantizing it back to b bits results in

$$y = Q[s \cdot 2^{-q}] = 2^{-b} \cdot \lfloor s \cdot 2^{-q} \cdot 2^b + 0.5 \rfloor, \quad (4)$$

which reflects the fact that the q least significant bits of s have been rounded out. In those cases, the resultant quantization noise is a RV having a non-zero mean, discrete distribution and its Probability-Mass-Function (PMF) depends on the number of right shifts took place. For example, in the case that such a number is shifted one bit to the right, the quantization noise ε_1 is distributed as

$$\varepsilon_1 = \begin{cases} 0 & w.p.\ 0.5 \\ -\frac{1}{2}2^{-(b-1)} & w.p.\ 0.5\,, \end{cases} \quad (5)$$

where the subscript 1 in ε_1 refers to the case of quantization noise generated by right shift of the b-bits number by one bit. The expected value of this noise equals $-2^{-(b-1)}/4$ and since it is not zero, when dealing with Signal-to-Quantization-Noise-Ratios of those RVs we will relate to the noise power rather than to its variance. We treat such noise sources herein as biased noise sources. To distinguish the power from the variance we use the symbol ρ^2 for power. The expected value of the power of ε_1 RV then is

$$\rho_1^2 = \frac{1}{2}\cdot 0 + \frac{1}{2}\cdot\left(\frac{1}{2}2^{-(b-1)}\right)^2 = \frac{2^{-2(b-1)}}{8}. \quad (6)$$

Due to the embedded bias, as expected, this is larger than the variance of the zero mean uniformly distributed quantization noise of (3). In a similar way, we can calculate the noise power of quantization noises that are generated due to the rounding after right shift of a b-bits number by q bits. In most BFP-FFT topologies and radices up to Radix-5, the right shifts are in the range of 0 to 3. Moreover, for right shifts of 4 and above the quantization noise power is very close to the variance of the zero mean uniform quantization noise of (3). Therefore, for analytical derivations we use

$$\rho_q^2 = \begin{cases} 0 & ;\ q = 0 \\ \frac{1}{8}2^{-2(b-1)} & ;\ q = 1 \\ \frac{3}{32}2^{-2(b-1)} & ;\ q = 2 \\ \frac{11}{128}2^{-2(b-1)} & ;\ q = 3 \\ \frac{1}{12}2^{-2(b-1)} & ;\ q \geq 4\,. \end{cases} \quad (7)$$

Throughout the paper, we treat the zero-mean uniformly distributed quantization noise of (3) as type-1 quantization noise, and the discrete, non-zero-mean, quantization noise with noise power of (7), as type-2 quantization noise.

III. Quantization Noise of BFP-FFT and Output SQNR

The noise at the output of a given butterfly is composed of two components: the noise that is generated by that particular butterfly, which we call butterfly self-noise, and the noise that is propagated through the butterfly (noise that was generated at earlier stages), which we call propagated-noise [12]. The propagated-noise variance, assuming type-1 noise sources, is multiplied by a factor of $2\alpha^2$ as each butterfly output is composed of the sum of two i.i.d. noise values and is multiplied by a scaling factor α. The self-noise, v, in DIT (v or u in DIF), is the noise generated by the quantization operations within the butterfly, and present at the butterfly output after being multiplied by α (refer to Figures 1 and 2). When the quantization model is of RHU, some of the noise sources are non-biased (type-1) and some are biased (type-2). Below we analyze the effects of the bias on the BFP-FFT output SQNR.

A. Biased Noise Sources in DIT BFP-FFT

In DIT butterflies, type-2 noise sources arise when all the TFs preceding a given butterfly are among the set $\mathcal{T}_1 \triangleq \{1, -1, j, -j\}\ ;\ j = \sqrt{-1}$. In the sequel, we designate the set of butterflies that all their inputs were multiplied by TFs belonging to $\mathcal{T}_1$ set, as the $\mathcal{B}_1$ set, or $\mathcal{B}_1$ butterflies. The multiplication of a b-bits value $x \in \left[-1\,, 1 - 2^{-(b-1)}\right]$ by the TF $w \in \mathcal{T}_1$ would result in a $2b$-bits number, $t = w \cdot x$, that it's lower b bits, before down scaling, are equal to zero. When such a number is scaled down by very few bits, we get the type-2 quantization noise described above. The power of those noise sources is larger than that of the type-1 noise sources, and hence they have negative effect on the power of the quantization noise at the FFT output. The distribution of the $\mathcal{B}_1$ butterflies among the FFT stages and among the butterflies within each stage is not uniform. This implies that the quantization noise power is not distributed evenly over the outputs of a given stage. By the nature of the FFT scheme, and the fact that the type-1 quantization noises are mutually independent [13], the variance of the quantization noise at each of the FFT output nodes, for FFT bearing only type-1 noise sources, is the sum of the variance of the self-noise sources of all the butterflies that this output node is connected to through the FFT flow graph, attenuated properly by the scaling factors along the flow. When incorporating the implications of the type-2 noise sources, the noise power at the output nodes grows. Moreover, due to the distribution of the $\mathcal{B}_1$ butterflies among the FFT, the noise power is not distributed uniformly at the FFT output nodes. To illustrate this, let us relate to a 16-point radix-2 FFT shown in Fig. 3. The set of TFs involved in this FFT are w_{16}^0 to w_{16}^7. Among those w_{16}^0 and w_{16}^4 belong to the $\mathcal{T}_1$ set. We also recall that one of the butterfly's inputs is always multiplied by 1, denote it as the first input, and the other, the second input, is multiplied by w_N^{kn}. As a result, the butterflies that their second input is multiplied by $w_{16}^1, w_{16}^2, w_{16}^3, w_{16}^5, w_{16}^6$ or w_8^7 result in type-1 noise source, while the butterflies that their second input is multiplied by w_{16}^0 or w_{16}^4, belong to the $\mathcal{B}_1$ set and results in type-2 self-noise source. The butterflies belonging to the $\mathcal{B}_1$ set are red-colored in Fig. 3. Since we deal here with radix-2, all the butterflies in the first two stages are among the $\mathcal{B}_1$ set [12]. From the flow graph of the 16-point DIT FFT we observe that the number of $\mathcal{B}_1$ butterflies per stage is larger at the first stages and is decreased by a factor of 2 from stage to stage down to two butterflies at the last stage. This is the case for any size for radix-2 DIT FFT. From the figure it is also clear that some output nodes of the FFT are connected to more $\mathcal{B}_1$ preceding butterflies along the flow graph, while other nodes are connected to less $\mathcal{B}_1$ preceding butterflies. For example, all the butterflies preceding the output node 0 ($k = 0$) are among the $\mathcal{B}_1$ set, while for output node 1, the $\mathcal{B}_1$ butterflies preceding it are only at the first two stages. The butterflies preceding it from the last two stages are non-$\mathcal{B}_1$ butterflies. As a result, the power of the quantization noise at output node 0 is larger or equal to that at the output node 1, i.e., $\rho_0^2 \geq \rho_1^2$ where equality is obtained if-and-only-if both q_3 and q_4 are equal to zero with probability 1.

Next, we wish to analyze the effects of the bias on the FFT output noise power. The propagated quantization noise at the output of a butterfly is of the form

$$\varepsilon_n^{p,(m)} = \varepsilon_n^{(m-1)} \pm w_N^{lk}\varepsilon_l^{(m-1)}, \quad (8)$$

where $\varepsilon_n^{p,(m)}$ is the propagated quantization noise at the output of node n at stage m, and $\varepsilon_n^{(m-1)}$ is the total quantization noise at the output of node n at stage $m-1$. In an FFT that incorporates type-2 noise sources, both $\varepsilon_l^{(m-1)}$, and $\varepsilon_n^{(m-1)}$ are composed of zero-mean random RV, ξ, plus some bias

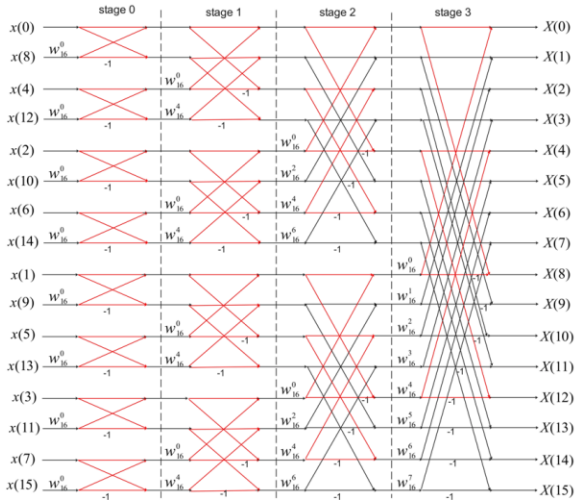


Figure 3: 16-point DIT FFT

term, μ,

$$\varepsilon_l^{(m-1)} = \xi + \mu. \tag{9}$$

In the Colley-Tuckey factorization, the two inputs to any butterfly are passing through the same set of TFs and butterflies' internal coefficients, prior to that specific butterfly. This can be observed from Fig. 3. As a result, the bias at the two inputs of any particular butterfly is the same. Denote the bias at the input of a given butterfly as μ^{in}. The propagated output bias at the two butterfly's outputs reads,

$$\begin{aligned} \mu_{i_0}^p &= \alpha\mu^{in}(1 + w_N^{nk}) \\ \mu_{i_1}^p &= \alpha\mu^{in}(1 - w_N^{nk}), \end{aligned} \tag{10}$$

where μ_i^p is the propagates bias at the butterfly's output, and α is the down scaling factor at that particular stage. From (10) it is clear that the bias at the two outputs is not identical. This is also the reason that the bias at different FFT output nodes is not the same. Let us now examine the bias at output nodes due to bias that has been generated at the first stage. The quantization noise sources at the first stage of the 16 point-FFT of Fig. 3 are all type-2 noise source. As such, they do bear an inherent bias. The actual bias value depends on the scaling factor of the first stage. Let us denote this bias as μ_1. This bias is propagated through the FFT flow graph toward the FFT output nodes. The multiplication factor it experiences on the path from the output of the first stage to any output node k of the FFT is of the form

$$\mu_k^{p_1,(3)} = \mu_1 \prod_{i=1}^{3} \alpha_i \left(1 \pm w_{16}^{n_i k}\right), \tag{11}$$

where $\mu_k^{p_1,(3)}$ is the bias that has been propagated from stage 1 toward the k^{th} output of stage 3. In the 16-point FFT there are four stages so the output of stage 3 is the output node of the FFT. From (11) it is clear that the values of the TFs, $w_{16}^{n_i k}$, along the path dictates the value of the bias at each output node. The values $w_{16}^{n_i k}$ can be constructive or destructive along the path, where the two extremes arises where $w_{16}^{n_i k}$ equals 1, leading to $\left(1 \pm w_{16}^{n_i k}\right)$ equals 0 or 2. Hence, the maximal value of the output bias due to propagation from stage 1 is $\mu_1 2^3 \prod_{i=1}^{3} \alpha_i$, while the minimum is zero. Note that if only one of the products $\left(1 \pm w_{16}^{n_i k}\right)$ along the path is zero, the propagated bias from earlier stages toward the output node will be zero.

In theory, it is possible to calculate the bias at each and every output node as a function of the scaling factors α_i. This calculation is out of the scope of this paper, but to get a glimpse of the effects of the bias on the output noise power, we will examine the noise power at the two extreme output nodes. In Fig. 4, the tracking of the paths from the output of the first stage toward output nodes $k = 0$ and $k = 8$ are highlighted. On the path toward output node $k = 0$ all the TFs are $w_{16}^0 = 1$ and all the butterflies' internal coefficients are also equal to 1. Therefore, the bias at that node gets the maximum value of

$$E_0^{(3)} = \mu_1 2^3 \prod_{i=1}^{3} \alpha_i. \tag{12}$$

On the path toward output node $k = 8$, all the TFs are also $w_{16}^0 = 1$, as well as the butterflies' internal coefficients of stages 1 and 2. But the internal coefficient at stage 3 is -1. The output bias therefore reads

$$E_8^{(3)} = \mu_1(1 + 1)(1 + 1)(1 - 1) = 0. \tag{13}$$

In order to assess the effect of the difference between the bias at output nodes $k = 0$ and $k = 8$, we have to take into account the noise power of the random component of the quantization noise, i.e., the output noise variance contributed by ξ (refer to (9)) that was generated at the outputs of stage 1. As ξ is an RV, the accumulation along the paths from the outputs of stage 1 toward the output nodes is a non-coherent accumulation. The noise variance that it contributes at output node 0 is

$$\sigma_0^2 = \sigma_\xi^2 2^3 \prod_{i=1}^{3} \alpha_i^2, \tag{14}$$

and it is identical at all the output nodes. The noise power at a given output node is the variance of the noise contributed by ξ, plus the power of the bias that was contributed by the bias μ. The ratio between the noise power at output node 0 and output node 8 is

$$\begin{aligned} \frac{\rho_0^2}{\rho_8^2} &= \frac{\left(\sigma_o^2 + \left|E_0^{(3)}\right|^2\right)}{\sigma_o^2} \\ &= \frac{\sigma_\xi^2 2^3 \prod_{i=1}^{3} \alpha_i^2 + 2^6 |\mu_1|^2 \prod_{i=1}^{3} \alpha_i^2}{\sigma_\xi^2 2^3 \prod_{i=1}^{3} \alpha_i^2} \\ &= \frac{\sigma_\xi^2 + 2^3 |\mu_1|^2}{\sigma_\xi^2} = 1 + 2^3 \frac{|\mu_1|^2}{\sigma_\xi^2}. \end{aligned} \tag{15}$$

The actual values of μ_1 and σ_ξ^2 at the output of the first stage, depend on the number of right-shifts took place at that stage. To get some sense of the ratio of (15), let us assume that there

was a single right shift at the output of the first stage. In that case, the RV $\varepsilon_1 = \xi + \mu_1$ is distributed according to (6). Its mean is $\mu_1 = -2^{-(b-1)}/4$ and its variance is $\sigma_\xi^2 = \frac{1}{16}2^{-2(b-1)}$. Plugging this into (15) we get

$$\frac{\rho_0^2}{\rho_8^2} = 1 + 2^3 \frac{\left|-2^{-(b-1)}/4\right|^2}{\frac{1}{16}2^{-2(b-1)}} = 9. \qquad (16)$$

Therefore, in the case that there was a single right-shift at the output of the first stage, the noise power at the output node 0 resulting from the bias of the first stage, is about 9.5 dB higher than the noise power at output node 8 for a 16-points FFT. As the size of the FFT grows, this ratio also grows.

B. Biased Noise Sources in DIF BFP-FFT

In DIF BFP-FFT butterfly, there are two quantization noise sources. The first, v, is the quantization after the down scaling that follows the ADD/SUB operation of the butterfly, and the second, u, is on the output branch that is multiplied by the TF (refer to Fig. 2). In DIF topology, the u source is of type-1, independent of the value of the TF. This is since the down scaling that follows the ADD/SUB operation of the butterfly, is determined such as to guarantee that the result of the TF multiplication never overflows. As such, no down scaling is done after the TF multiplication and the quantization noise obeys the zero-mean uniform distribution of (2). The u source, on the other hand, is a type-2 noise source, since it is a result of quantizing a $b+1$ bits number that is down scaled by very few bits to the right. The implications of the above is two folded: (a) since there are three quantization noise sources in a butterfly of a DIF BFP-FFT, as compared to only two in DIT BFP-FFT, we expect approximately $10\log_{10}1.5 = 1.76\ dB$ lower SQNR in DIF BFP-FFT implementations, and (b) All the output nodes are connected to the same number of butterflies containing type-2 noise sources. Despite the fact that all the output nodes are connected to the same number of butterflies containing type-2 noise sources, the noise power at the output nodes is not evenly distributed. This is a consequence of the multiplication factors that each noise source passes through, toward each output node, as explained for the case of DIT BFP-FFT, and is reflected in the flow graphs in Fig. 4 and

IV. OUTPUT NOISE PSD

Based on the analysis of Section 0FIII, it is clear that the quantization noise at the FFT output of a DIT BFP-FFT is not white. Its PSD is not flat and the two phenomena that lead to larger noise power in some of the output nodes, are the fact that more type-2 noise sources are accumulated into some of the output nodes, and among those, at some of the nodes, the biases are accumulated "more coherently" than in other nodes. The outcome of those phenomena is reflected in Fig. 5 for 4096 points FFT (the FFT size used in the fifth generation cellular standard 3GPP 5G-NR [14]). The figure presents the SQNR per output node, which is an important metric in most systems using FFT. The figure reveals that output nodes at the DC vicinity are most affected and suffers more than 25 dB extra noise power as compared to most of the other FFT output

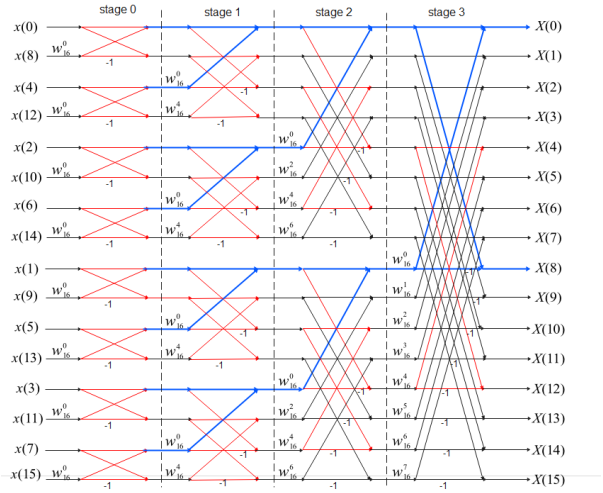


Figure 4: Path tracking toward output nodes 0 and 8

nodes. Yet this significant degradation is concentrated at small set of output nodes at the DIT case. It shall be emphasized that the effect, and the ratio of, type-2 noise sources in DIT BFP-FFT is a function of the scaling pattern, $\boldsymbol{q} = [q_1, q_2, \ldots, q_M]$, which by itself is a function of the realization of the input sequence, x_n, and can vary from one realization to another. If for example, the variance of the input sequence is very low, no down scaling would be done at the first several stages. This will lead to noise power at the output FFT nodes that is flatter than the one shown in Fig. 5. Nevertheless, for most practical use-cases, the scaling patterns lead to similar output noise PSD and SQNR as reflected in Fig. 5.

The SQNR of a DIF BFP FFT is also shown in Fig. 5. The differences to that of the DIT are clearly seen. The averaged noise power per output node of the DIF is, as expected, higher (lower SQNR), and many more output nodes in the DC vicinity suffer SQNR degradation. In the figure, periodicity patterns are observed. This stems from the fact that there are several sets of output nodes, each set suffers the same amount of noise power, but due to the nature of the output ordering (known as bit-reversal ordering), the pattern is periodic. In the DIF, the periodicity is not only due to the bias that accumulates differently between the various sets of output nodes. The imbalance of the two output-branches of the DIF butterfly with relate to the quantization noise (the upper one noise source while the lower branch suffers two), results in

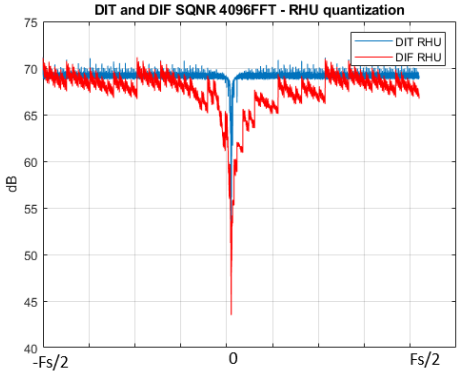


Figure 5: DIT and DIF SQNR - RHU quantization

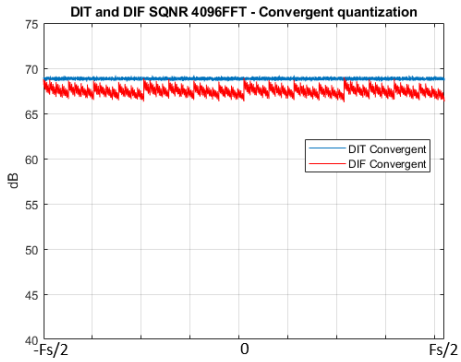


Figure 6: DIT and DIF SQNR - Convergent quantization

periodicity as well. Therefore, in DIF, even if the bias of the noise sources will be removed, a periodicity pattern will still be present.

V. WHITENING THE OUTPUT QUANTIZATION NOISE

In most of the applications using the FFT, colored quantization noise is undesirable. In many applications, some minimum SQNR for all the output nodes is required. This is the case, for example, in OFDM modems, such as 5G-NR [14]. A simple solution that sometimes is adopted in practical systems is to process the FFT at higher precision, e.g., using 32bit instead of 16bit multipliers. This is a straight forward scheme that increases significantly the output SQNR, but its drawback, of course, is the large increase in complexity. Another alternative, is to analytically calculate the bias term in each and every output node, and for each and every FFT size, and subtract it from the relevant output node. This bias term depends on the scaling pattern $\boldsymbol{q}$, and hence has to be calculated per realization of input sequence x_n . This alternative is an analytically accurate bias removal algorithm, but its drawback is that it significantly increases the complexity and the latency of the BFP-FFT. The third alternative is to use convergent rounding, [10], instead of the RHU, such as done in some hardware FFTs. Typically, the cost of adding convergent rounding to the processor's hardware is low in extra logic area, but may have slight effect on the processor's maximal frequency (and this is the reason that it is usually avoided). The effect of this alternative is presented in Fig. 6. It is clearly seen that the high noise power in the DC vicinity has been eliminated completely in both BFP-FFT topologies. The residual periodicity in the DIF BFP-FFT is the result of the imbalance of the two output branches of the butterfly as explained in Section IV.

VI. CONCLUSIONS

The effects of the classical RHU based quantization on the PSD of the quantization noise at the output of BFP-FFT is analyzed. A comparison between DIT and DIF BFP-FFT that are implementable on most of the CPUs and DSPs is provided. It is shown that for common CPU and DSP, the DIF BFP-FFT is much more sensitive to the finite-word-length effects of the processor, and results in worse SQNR and larger number of output nodes that suffer sever SQNR degradation. Three schemes to overcome the bias effects have been presented. Amending the processors with convergent rounding resolves the problem without increasing the complexity and the associated processing cycles count.

We used radix-2 FFT to convey the ideas and to present the sources of the matter. The same ideas can be extended to any other type, and topology, of BFP-FFT.

While this study focuses on uncorrelated input sequences, single-tone or other types of correlated inputs may be of interest for specific applications and are left for future work.

REFERENCES

[1] J. M. Cioffi et al. "Very-high-speed digital subscriber lines," *IEEE Communications Magazine*, vol. 37, no. 4, pp. 72-79, 1999.

[2] B. F. Frederiksen and R. Prasad, "An overview of OFDM and Related Techniques Towards Development of Future Wireless Multimedia Communications," in *IEEE Proc. Radio and Wireless Conference*, Boston, 2002, pp. 19-22.

[3] N. Cvijetic, "OFDM for Next-Generation Optical Access Networks," *IEEE Journal of Lightwave Technology*, vol. 30, no. 4, pp. 384-398, 2012.

[4] A. V. Oppenheim and C. J. Weinstein, "Effects of Finite Register Length in Digital Filtering and the Fast Fourier Transform," *Proceedings of the IEEE*, vol. 60, no. 8, pp. 957-976, 1972.

[5] W.-H. Chang and N. Q. Truong, "On the Fixed-Point Accuracy Analysis of FFT Algorithms," *IEEE Transactions on Signal Processing*, vol. 56, no. 10, pp. 4973-4682, 2008.

[6] P. Gupta, "Accurate Performance Analysis of a Fixed Point FFT," in *Twenty Second National Conference on Communication (NCC)*, Guwahati, 2016, pp. 1-6.

[7] P. D. Welch, "A Fixed-Point Fast Fourier Transform Error Analysis," *IEEE Transactions on audio and Electroacoustics*, vol. 17, no. 2, pp. 151-157, 1969.

[8] H. G. Kim, K. T. Yoon, J. S. Youn, and J. R. Choi, "8K-point Pipelined FFT/IFFT with Compact Memory for DVB-T using Block Floating-point Scaling Technique," in *International Symposium on Wireless Pervasive Computing (ISWPC)*, Melbourde, 2009, pp. 1-5.

[9] J.-R. Choi, S.-B. Park, D.-S. Han, and S.-H. Park, "A 2048 Complex Point FFT Architecture for Digital Audio Broadcasting System," in *IEEE International Symposium on Circuits and Systems*, Geneva, 2000, pp. 696-696.

[10] L. Xia, M. Athonissen, M. Hochstenbach, and B. Koren, "Improved Stochastic Rounding," *arXiv , 2020, Available: https://arxiv.org/abs/2006.00489.*

[11] B. Widrow, I. Kollar, and M.-C. Liu, "Statistical theory of Quantization," *IEEE Transactions on Instrumentation and Measurement*, vol. 45, no. 2, pp. 353-361, 1996.

[12] G. Naveh, "Finite-Word-Length-Effects in Practical Block-Floating-Point FFT," in *SIGNAL 2025*, Lisbon, 2025, pp.33-39.

[13] C. J. Weinstein, "Quantization Effects in Digital Filters," M.I.T. Lincoln Lab. Tech. Rep. 468, ASTIA doc. DDC AD-706862, 1969.

[14] *NR; Physical Channels and Modulation*, 3GPP TS 38.211, 2025.

Beyond Calculus: Modernizing System Modeling and Computational Methods

Pavel Loskot
ZJU-UIUC Institute
Haining, China
e-mail: pavelloskot@intl.zju.edu.cn

***Abstract*—Mathematical modeling has been used for centuries to create understanding of natural phenomena, and recently, to enable constructing man-made systems. Computational models in engineering are mostly based on calculus and linear algebra, and they are now well-established. However, the rapidly rising complexity of modern intelligent and autonomous systems calls for rethinking the present modeling strategies to allow beyond calculus modeling with different levels of abstractions, semantics, and information granularity. This vision paper offers a non-mathematical outline of selected advanced mathematical topics, which can be adopted not only to enhance modeling methods in engineering, but also to modernize engineering curricula by bridging the gap between engineering and mathematics. The key idea is to allow algebraic manipulations of mathematical objects, and assigning numerical values to these objects, so they can be used as natural data structures within mathematical models.**

Keywords-abstract algebra; algebraic geometry; algebraic topology; calculus; curriculum; mathematical object; modeling.

I. Introduction

The interest into studying scientific problems evolved since the age of Enlightenment. The first attempts were focused on creating simple physical models of measurable quantities. This then evolved into investigating the problems of disorganized complexity assuming more complex systems and phenomena, but with well-defined behaviors. The scientific progress following the World War 2 (WW2) changed the focus on problems of organized complexity that embrace all factors influencing whole systems.

The scientific methods were initially mainly experimental. The strong interest into theoretical sciences including mathematical modeling and analysis emerged already during the WW2. The advent of microprocessors leading to affordable personal computers and software tools in the 1990's gave rise to a widespread adoption of computer simulations to design and analyze complex engineering systems. The main advantage of computer simulations is that they are universal, require fewer assumptions, and can be used to evaluate complex mathematical models. The disadvantage is that they can be rather slow when they are used iteratively.

The most recent breakthrough in scientific methods emerged in the mid 2000's when not only the computing capabilities started to exponentially improve, but also the vast volumes of data started to be collected. The new era of big data and large computing models enabled many new applications including knowledge mining, and analyzing, predicting and controlling highly complex systems. This encouraged developments of machine learning models, which may be considered to be calibrated computer simulations having the real-time observations as their inputs. The universality of large-scale (deep) machine learning models made them very popular, and widely used in many applications across different scientific and engineering domains. However, the main drawbacks of large-scale machine learning models are their excessive consumption of resources, slow training, and difficulty, or even impossibility of validating the correctness of their outputs. The deep learning models are inherently probabilistic, and their outputs are only valid with a certain probability within a given context.

The current developments of deep (machine) learning models, and more generally, of intelligent systems are purely experimental. There is a substantial cost incurred for their training and testing, and many trials-and-errors are required for selecting their hyper-parameters, and fine-tuning. Furthermore, these models are neither explainable, nor is their optimality guaranteed. Hence, there is a great need to develop design guidelines for deep learning models, which can serve the same purpose as the known laws for designing electrical circuits, and electromagnetic systems, to name some examples.

The persisting lack of design guidelines may be caused by the excessive and constantly rising complexity of today's mathematical models. These models must be considered across many different scales and hierarchies. In addition to quantitative evaluations, the model semantics can play a crucial role in designing many intelligent and autonomous systems. A viable strategy could be to consider the methods and strategies that were adopted for taming the complexity in software development. At a basic level, these strategies include the model granularization and modularization. More advance strategies can be inspired by object-oriented programming, and include the model encapsulation, inheritance and polymorphism, which allow high-level abstractions and reusability.

The aim of this paper is to explore advanced mathematical structures, which can enrich traditional modeling methods. The traditional methods mostly rely on computations involving basic operations and functions of vectors and matrices [1]. The main advantage of these methods is that they are optimized for software and hardware implementations. The complexity of today's models dictates using more abstract and more flexible modeling strategies that can capture sophisticated relationships and semantic representations of multiple sub-models at different scales and levels of hierarchy.

The rest of this paper is organized as follows. Section II reviews traditional modeling methods that have been used in designing and analyzing engineering systems for many decades. Section III outlines mathematical structures, and related topics, which could be considered for advanced mathematical modeling.

Section IV discusses opportunities and challenges in using using advanced mathematical objects in building models not only in engineering, and mention possible strategies. Finally, Section V concludes the paper.

II. Traditional Methods

Traditional modeling methods in engineering are well-established. They rely on basic mathematical concepts that are normally covered in typical engineering curricula. The corresponding mathematical courses include calculus, linear algebra, and discrete mathematical structures [2]. In later years, the students usually progress to using mathematical models in various engineering applications.

A. Calculus and Differential Equations

Calculus at engineering schools is mainly about continuous functions of real and complex variables. Differential and integral calculus is useful, for example, in computing the areas and volumes of geometric objects, and characterizing scalar and vector fields along or within defined surfaces and lines in two or three dimensions. The solutions of ordinary and partial differential equations of small order can be obtained directly, or by using various transforms. For higher orders, and when computing high-dimensional integrals, numerical methods must be used. These techniques are sufficient for modeling physical and other phenomena that are encountered when designing engineering systems including electromagnetic radiation, electrical circuits, chemical reactions, and others.

B. Linear Algebra

Multiple variables can be arranged in regular shapes as vectors and matrices. These objects exist in vector spaces, which may be endowed with additional properties, such as inner products and other more general metrics. The two-dimensional matrices are important in describing systems of linear equations, as well as (locally) linear transformations. The spectral properties of matrices are related to their eigenvalues and eigenvectors. The matrices can be diagonalized and factored in various ways, which is important in many signal and data processing applications. The standard matrix product is only defined in two dimensions, but it can be extended to higher dimensions using the methods of multi-linear algebra.

C. Discrete Structures

In addition to matrices in different number of dimensions, the basic discrete objects are sets and graphs. These structures are normally encountered in undergraduate courses on data structures and algorithms, and on designing logical circuits. Graphs can represent not only systems of interacting entities, but they also capture the pairwise relationships between such entities. In graph signal processing, the graph edges define linear or non-linear transformations between neighboring variables, which are assigned to graph vertices. Graphs can also represent time-evolution of states in dynamical systems, for example, when assuming recurrences. Social network analysis is mostly concerned with graph invariants, which are scalar metrics representing the graph structure. Graph theory provides many useful properties and methods, for example, how to enumerate and count graphs of certain properties.

D. Signals and Systems

The courses on signals and systems are intended to provide the students with fundamental skills to be able to work with various models in engineering applications. The signals are time-varying variables, and every system must have at least one input or output. The signals can be deterministic or random, and they can be studied not only in time, but also in other domains. The time and frequency domains can be considered to be either continuous, or discrete. The signal properties are usually defined to be scalar invariants. In many applications, it is desirable to derive the properties of output signals given a system and its inputs. For example, the systems can be designed, so that their outputs have desired properties given specific input signals. Alternatively, the task is to define the input signals to obtain the desired outputs given a specific system. The inverse problems infer the latent properties from the input and/or output observations.

The system is said to be observable, if the hidden state or parameter value can be recovered from available observations. The system is said to be controllable, if it can be driven to a desired state from any given state in finite time.

Unlike universal modeling methods that are based on deep learning, the specialized models of signals and systems are numerically much more efficient, interpretable, and provably optimum. There can be a family of models of the same phenomena, which have varying degree of faithfulness. The current trend is to reduce the complexity of universal models by incorporating well-defined specialized models whenever they are available. The model complexity can be also reduced by analytically solving some parts of the model, which leads to efficient semi-numerical modeling strategies.

E. Optimizations

The optimizations are useful in maximizing the performance while minimizing the use of resources. The optimization objective is often constrained by equality and inequality constraints, which restrict the feasible set of possible solutions. There are also problems involving multiple objectives, which can be conflicting, and lead to Pareto trade-offs. Convex optimization problems guarantee the existence of a single global minimum. The problems involving continuous optimization variables can be solved numerically using the methods of gradient descent. Combinatorial optimization problems involve at least one optimization variable that is discrete, The challenge is how to find practical methods for online optimizations, which must update the solution iteratively as the new observations arrive.

In practice, the solutions of primary problems are often obtained as the optimization problems. This is the case, for example, of many parameter estimation methods.

III. Advanced Methods

There are many important results in mathematics, which may likely be useful for designing engineering systems. Even though these results may be considered to be common knowledge in mathematics, they are likely completely unknown in engineering. In this paper, our focus is specifically on exploring advanced mathematical objects, which could be adopted in building enhanced models of engineering systems.

Computational analysis of engineering models generally involves manipulating basic objects, such as numbers and the corresponding arithmetic operations, and functions. The key observation is that, at more abstract level, the binary arithmetic operations can be defined for combining mathematical objects beyond numbers. For example, one can define how to sum and subtract graphs, or multiply and divide high-dimensional tensors. Specifically, a binary operation, $\square$, and a unary operation, $\triangle$, respectively, defined for mathematical objects, O_i, from a certain class of such objects are:

$$O_3 = O_1 \square O_2\,, \quad O_2 = \triangle\, O_1. \tag{1}$$

It should be emphasized that the operations, $\square$, and, $\triangle$, are neither arbitrary nor ad-hoc. These operations can be precisely defined using fundamental results, for example, in Abstract Algebra, and Category Theory. Likewise, it may be possible to introduce basic arithmetic functions, F_i, such as exp, log, sin, and cos to define the maps:

$$O_2 = F_i(O_1). \tag{2}$$

Furthermore, in addition to object transformations (1) and (2), which create new objects from the existing ones, abstract mathematical objects can be assigned numerical values, i.e., numerical objects. For example, the graph vertices and edges can be assigned scalar variables or parameters. These values are then manipulated using standard arithmetic operations and functions. Note that mathematical objects that hold numerical values are usually referred to as data structures in implementing the data processing algorithms. In general, defining signal and data processing over given mathematical structures containing numerical values is largely an open research problem. The challenge is how to account for the structural constraints imposed by the mathematical structures considered. For example, graph signal processing is concerned with signals that are defined over graphs. A transformation, T_O, of numerical values, $\{v_i\}$, which are constrained by mathematical structure, O, can be formally written as,

$$\{v_i'\} = T_O\,[\{v_i\}] \tag{3}$$

where v_i' is the transformed value of the initial value, v_i.

In the sequel, a few specific mathematical objects are outlined, which may be useful in defining the binary and unary transformations, $\square$, and, $\triangle$, and the functions, F_i, of these objects. However, the precise definitions and analyses are beyond the scope of this paper including investigating practical applications. Moreover, although mathematical literature on these topics is plentiful, not all mathematical resources are accessible by researchers having pure engineering backgrounds.

A. Tensors and Manifolds

Tensors have been introduced to describe multi-dimensional linear transformations [3–7]. This turned out to be very useful, for example, in modeling physical phenomena independently from the observer, and a chosen coordinate system (i.e., a frame of reference). It should be noted that multi-dimensional matrices in machine learning models are often called tensors; for example, tensors are often mentioned as low-rank approximations. However, this is a misnomer, since tensors are primarily linear transformations, which must satisfy certain properties, and not simple data structures.

Tensors exist in a vector space, V, which itself is a mathematical object. There is an isomorphism between the vector space and its dual. Linear forms map the vectors to scalars, i.e., $V \mapsto \mathbb{R}$. General tensors are of type, (p,q), where q is the covariant order, and p is the contravariant order. The order of tensor is equal to the number of upper and lower indices required to address its components.

The covariant and contravariant components differ whether they change along or against the change of the basis. They correspond to the orthogonal and parallel projections, respectively. A simple linear transformation, $V \mapsto V$, is a $(1,1)$-tensor. The $(0,1)$ tensor is a simple linear form or covector, whereas $(1,0)$ tensor is a contravariant vector. The multi-linear forms combine multiple vectors, and produce a scalar; the vector dot-product is a special case of a bilinear form. The dot-product (more generally, an inner product) induces a metric structure on the vector space. The tensor product, which is generally non-commutative, is a product of constituent multi-linear forms. The purpose of the tensor product is to linearize multi-linear forms. For example, a bilinear form, $V \times V \mapsto \mathbb{R}$, is a covariant 2-tensor of type $(0,2)$. The matrix determinant is an example of a multi-linear form. The multi-linear forms can be also considered for complete vector spaces.

Manifolds are another important structure that can be assumed for advanced modeling of complicated spaces (informally, shapes). They are parameterized locally as Euclidean sub-spaces of the same dimension. The collection of these overlapping sub-spaces and their pairwise mappings is referred to as an atlas of the underlying space. The smooth manifolds are differentiable, and allows calculus. The special case is tangent manifolds that can be assigned a tangent plane at every point. It is sometimes desirable to follow a continuous curve that lies within the manifold. The manifold hypothesis assumes that manifolds are low-dimensional smooth objects that are embedded in high-dimensional spaces. Thus, manifolds represent the target phenomenon to be modeled, which are observed in high-dimensional noise space.

Computing and machine learning models can benefit from considering the definitions and properties of standard tensors. The signal and data processing can be made more robust by extracting information manifolds from noisy measurements. These methods are mathematically more evolved, and usually not considered in standard textbooks. For example, the manifolds can be considered for analyzing and visualizing deep

learning models with large number of parameters.

B. Set Theory and Logic

Zermelo-Fraenkel (ZF) set theory is a very general formal approach to ordinary mathematics [8–10]. It adopts a small set of axioms, which are then used to derive the true statements about the universe of sets. The Axiom of Choice assumes that the elements in any subset can be distinguished. Such a property cannot be proved, but it is fundamental in the proofs of many theorems. The key idea is that many mathematical objects, such as numbers can be encoded as sets including natural, ordinal and cardinal numbers. Basic arithmetic operations can be defined for all these number sets. It is possible to compare the sizes of even uncountably infinite sets.

Binary relations on sets allow defining their orders. The relations can be transitive, symmetric, and reflexive. The set is said to be well-ordered if the ordering is possible for all pairs of elements, and every subset has the smallest element, otherwise the set may be only partially ordered. Equivalence relations allow partitioning elements into equivalence classes.

Mathematical statements are often formulated as logical statements. Even the notion of mathematical proof can be precisely defined. There is presently no proof that the ZF theory is consistent (i.e., there are no statements that contradict themselves). The logical system is said to be complete, if any true statements that can be described in this system can be proved from axioms. The theorems proved by Gödel in 1930's showed that no mathematical system can be complete and consistent at the same time [11]. Moreover, consistency of a system cannot be proved even from within that system.

Mathematical logic is formal, unambiguous language of statements and their proofs. This language is exact, verifiable and reproducible. Propositional logic formalizes deductions (i.e., drawing specific conclusions from proofs or general knowledge). It is language with syntactical rules how mathematical statements are constructed as strings. There are also associated semantic rules allowing valuations of statements for given truth assignments of propositional variables. The valuations can be satisfiable and logically valid. Logical formulas are logically equivalent to Boolean functions. More importantly, every satisfiable propositional theory is consistent, and every complete propositional theory is satisfiable.

Propositional logic is only concerned with implications, which is rather limiting. First-order logic extends propositional logic with quantifiers, predicates, relations, functions and constants. It is sufficiently powerful, so that many mathematical concepts can be naturally expressed as the first-order formulas. There are again precise syntactic rules how to define statements, and semantic rules how to interpret their validity for given truth assignments. Gödel's theorems are normally considered within first-order logic.

Mathematical logic can greatly expand the capabilities of computational systems. Gödel's completeness and incompleteness theorems are fundamental for understanding the limits of Artificial Intelligence (AI) systems. Moreover, since mathematical statements are much more constrained than statements in natural languages, it facilitates automated reasoning systems and formal provers.

C. Abstract Algebra

Abstract Algebra is a classical field of mathematics, which is normally covered in undergraduate curricula as part of mathematical, but not engineering programs. Some introductory textbooks that are more accessible to beginners are, for example, [12–17]. The main idea is to consider sets that have special properties. The elements of these sets are mere representations of mathematical objects; the actual objects considered are not important. This is a very powerful abstraction, which has connections to many other areas of mathematics including geometry, topology, and number theory.

The most common objects in Abstract Algebra are groups, rings, and fields. The key property is that a small number of unary and binary operations defined over the elements of these sets are closed. It is often useful to consider subsets, so that they retain the key properties of the original set. These subsets are called sub-groups and sub-rings, respectively. The existence of inverse elements allows defining complementary operations, such as subtraction for addition, and division for multiplication. The groups can be cyclic, and generated by a small number of elements. The permutation groups are useful in systematically searching a space of all possibilities. The cosets allow natural partitioning and factoring of groups. Multiple groups can be combined using internal and external direct products. Group homomorphism enables finding representative sub-groups of small order. Rings adds additional properties to groups. Integral domains and their special case, fields, have additional properties over those that are required for rings.

There are many other advanced group topics, which may be useful for defining and working with complex engineering and machine learning models. Examples of these advanced topics include geometric constructions of groups, classes and their equations, group symmetries, groups with additional conditions yielding many specific properties, actions on groups, group compositions, group representations, and other. Abstract Algebra allows exploring interesting analogies between different mathematical objects, such as integers and polynomials. There is an interplay between group theory and linear algebra, whereas number theory can be interpreted geometrically.

Commutative rings have applications in algebraic coding theory and cryptography. Complex shapes of molecules and crystals can be described using the concept of groups. Fields and Galois theory are of interest when designing data processing and machine learning methods involving finite precision numerical values. Such number representations greatly reduce the memory requirements of large models, as well as speed up arithmetic computations while providing exact results.

Newer research problems can consider how to define abstract representations of large complex models, such as those that are used in deep learning applications. If these abstract representations form a group, many properties and results from group theory would immediately follow. This can pave the way for more guided and interpretable design of deep

learning models, and create new connections between statistics, probability, and algebraic methods.

D. Algebraic Geometry and Topology

Topological spaces generalize metric spaces, which generalize vector spaces [18][19]. Topology allows comparing shapes of geometrical objects. For example, the topologist cannot tell the difference between a circle and a square that lie in a two-dimensional plane. More specifically, topology is a collection of certain subsets of a given set. It is useful to consider continuous maps between topological sets. A special, but common case is Euclidean topology.

Homeomorphism is a relaxed notion of equality in topology. It is a continuous bijection, but it may be too restrictive when comparing shapes. The important property of homeomorphic spaces is that they share all topological invariants including connectedness (spaces are a single piece), compactness (spaces are closed and bounded), and Hausdorffness (all points can be separated by spaces). Topological spaces can be compactified by adding extra points. New topological spaces can be created from simpler topological spaces using Cartesian products and connected sums. It is also possible to create new topological spaces using group actions. It allows partitioning the space into pieces that are connected by group elements.

An important concept in topology are identification spaces. They are unified representations of topological shapes, which greatly simplifies reasoning and visualizations. The surfaces are topological equivalents of manifolds defined in Calculus. The identification spaces of complex topological surfaces are collections of polygons. The polygons consist of vertices, edges and faces; they can be viewed as hyper-graphs. In addition, it can be shown that any closed surface is homeomorphic to either a sphere, a torus, or a projective plane.

Homotopy can be used to define equivalences between maps of topological spaces. It is much less restrictive than homeomorphism, since the former only requires equivalence of topological structures. Homotopy classes of loops in a topological space forms a fundamental group of that space. The fundamental group is another topological invariant.

Homology plays a fundamental role in topological algebra. It assigns algebraic objects to topological spaces, so that they remain unchanged under homeomorphic deformations. This can be used to detect holes in different number of dimensions. Homotopy can be used to represent topological spaces as simplical complexes, which can be practically obtained, for example, using triangular meshing. Such a compact representation is suitable for developing algorithms for counting holes within topological spaces; these counts are known as Betti numbers. In practice, it is sufficient to assume abstract simplical complexes that do not depend on particular geometric embeddings. The boundary maps are connected sums of simplices along the boundary. Considered recursively, the underlying linear boundary maps form a chain of simplical complexes of decreasing orders. The boundary kernel and the previous boundary image then define a homology group.

Persistent homology enables algorithmically evaluating topological features in metric spaces, such as point clouds [20]. The idea is to associate points with parameterized simplical complexes, such as Vietoris-Rips complex. When the distance parameter increases, the underlying homology changes; this process is referred to as filtering. The task is to identify, which homological features are most persistent during the filtration. The filtered simplical complexes can be then assumed to be approximations of the underlying topology of data points. Topological Data Analysis (TDA) is very robust in scenarios with missing and noisy data. It can be visualized, for example, as barcodes, and persistence diagrams. However, computing the filtration is numerically very costly.

Geometric Algebra is concerned with k-vectors as a generalization of ordinary 1-vectors that are placed in $(n \geq k)$-dimensional vector spaces [21–24]. The 0-vectors are referred to as scalers, or simply scalars. Every k-vector has a magnitude, direction, and orientation. The k-vectors can be multiplied using an outer (wedge) product, which is anti-commutative. There is also an inner product of k-vectors. These products have simple geometric interpretation; i.e., the wedge product of parallel or linearly dependent vectors is zero, and so is the inner product of orthogonal or linearly independent vectors. Geometric product combines both inner and outer products. Moreover, every k-vector has its inverse, which can be used to divide k-vectors; such operation is not defined for the standard vectors. Multi-vectors are the sums of simple k-vectors with possibly different orders, k.

The power of Geometric Algebra lies in its ability to greatly simplify mathematical models and expressions, especially those that are related to calculus over vector fields. Multi-vectors can be also used to compactly define differentiable manifolds using directional and geometric derivatives.

E. Category Theory

Category Theory generalizes foundational mathematical concepts by formulating a high-order theory, which is governed by the laws of free algebra [25–29]. Categories consist of objects, and morphisms between these objects. Morphisms are maps that are associative, and composable, and as any other maps, they can be surjective, injective, or both. The focus is on high-level properties of mathematical objects, while abstracting away intrinsic details. A graph-like structure of categories defines external associations between objects, which can yield information about axioms, rules of association, and the properties that are transferable between objects, as well as universal to all objects. Categories can be studied as hierarchical graphs of objects and their morphisms, or as algebras of functions, i.e., functions of objects, and composition of functions. The associativity and composability of morphisms implicitly restrict the structure of categories.

Some examples of categories include sets and functions over sets, vector spaces and linear transformations, metric spaces and continuous maps, and groups and homomorphisms. The categories are described assuming generalizations of common objects, such as initial and terminal objects, pushouts and

pullbacks, (co-) products, (co-) limits, and (co-) completeness. For instance, products and pullbacks generalize limits, whereas sums and pushouts generalize co-limits. The morphisms between complete categories are referred to as functors. Like morphisms, functors must preserve composition. For example, there is a functor from category of topology spaces with continuous maps to category of sets of points with functions between these sets. The functors themselves can form a category with morphisms referred to as natural transforms. Further generalizations lead to the notion of fibers, and sheafs. The idea of fibers is to index one category over another category akin to indexing of one set over another set. The sheafs introduce locally defined data that are attached to open sets in a topological space.

The composability of large systems from simple parts using simple rules is one of the demonstrated applications of Category Theory. The objective is to scale-up systems without unnecessarily increasing the system complexity, for example, by avoiding creating unnecessary couplings.

IV. Discussion and Future Outlook

Modern modeling methods must incorporate advanced mathematical objects to be capable of describing processes and phenomena across different abstraction layers and at different scales. Mathematical objects themselves can be subject to algebraic operations and functions in order to form, for example, an algebraic group. Mathematical objects can be assigned numerical values to enable calculus in addition to manipulating their structure. For example, graphs can be assigned numerical values as attributes to all vertices and edges. The numerical values can be assigned assuming certain probability distributions, which would lead to signal and data processing methods over these mathematical structures.

Mathematics has been traditionally much closer to computer science than to engineering. Mathematics likely has many results that are very useful in creating and analyzing advanced models of modern intelligent and autonomous systems, but these results are mostly unknown in engineering. Bridging such a knowledge gap requires a dialogue between engineers and mathematicians. The former can identify unsolved practical problems, while the latter may suggest relevant mathematical concepts and tools that could be considered.

Another highly desirable strategy is to revise the current engineering curricula that can no longer keep up with very fast evolving complex technologies including the emergence of AI powered systems. It is crucial to identify knowledge invariants with a long-term validity. This also concerns choosing the key topics in applied or even pure mathematics that should be added to engineering curricula. However, teaching advanced mathematics to engineering students can be challenging. It may require choosing plenty of motivating examples to gain the intuition, even before attempting to solve any problems.

A very interesting research avenue is re-representing the results from mathematical literature, so they become accessible to technical researchers without training in advanced mathematics [30–37]. Many mathematical concepts can easily become inaccessible even to expert mathematicians, depending how deeply a given topic is explored. In general, mathematical literature appreciates and strives for notational brevity, which is a roadblock for engineers. It would be very desirable to explain common mathematical notations to non-mathematicians including how to read mathematical literature.

However, neither mathematical objects, nor algebraic structures were explicitly defined in this paper. At present, it is unclear how to define the models involving advanced mathematical objects and structures that could be constructed and analyzed with moderate efforts. The justification of these models may require substantial research in order to demonstrate their effectiveness as well as interpretability. It also includes computational feasibility. There is likely a trade-off between model compactness, informativeness, and its utility in the real-world engineering applications.

Furthermore, large language models and other AI tools undoubtedly have enormous impact on teaching and research. However, the current generation of these tools did not reach the required level of factual accuracy. Fortunately, these issues are more of a concern when working on advanced tasks that require more in-depth investigations. For introductory teaching and learning of advanced mathematics (and other subjects), the AI tools can reliably identify and briefly summarize the key concepts and terminologies, and play a role of a powerful interactive tutor akin to how Wikipedia was used until recently. Nevertheless, hand-picked, human chosen teaching and learning materials still appear to be much better suited to the student needs, especially when these materials are prepared by teachers who have many years of teaching and research experience.

V. Conclusions

The main objective of this paper was to outline mathematical structures beyond vectors and matrices as the first-class compositional objects that could be used in defining advanced mathematical models. Basic algebraic operations and functions could be defined for general mathematical structures to construct other such structures. These operations and functions cannot be arbitrary, but they must be rigorously derived, for example, using external associations of categories in category theory. In addition, mathematical structures can be assigned numerical attributes, which would allow numerical calculations over these structures. The important question is whether rather complex mathematical notations, and indeed even mathematical theories themselves could be simplified for the pragmatic use in engineering. Moreover, the strict mathematical rigor could be sacrificed in exchange of allowing novel practical modeling and design methods for complex engineering systems.

It is obvious that there are many open research problems, and issues to be resolved. The future work will consider how to present more advanced mathematical concepts to engineering students. It will lead to defining many open research problems. For example, one can define attached different numerical attributes to different components of mathematical objects, and then consider signal processing over these objects.

REFERENCES

[1] N. J. Higham, *Functions of Matrices*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2008, ISBN: 9780898716467.

[2] B. V. S. Kumar and H. Dutta, *Discrete Mathematical Structures, A Succinct Foundation*. Boca Raton, FL: CRC Press, 2020, ISBN: 9780429624742.

[3] D. A. Fleisch, *A Student's Guide to Vectors and Tensors*. Cambridge, UK: Cambridge University Press, 2012, ISBN: 9780521171908.

[4] J. H. Heinbockel, "Introduction to tensor calculus and continuum mechanics", Lecture Notes, 2001, [Online]. Available: https://www.academia.edu/105055636/.

[5] M. Tabatabaian, *Tensor Analysis for Engineers*, 3rd ed. Dulles, VA: Mercury Learning And Information LLC, 2023, ISBN: 1683929628.

[6] J. Tromp, *A Geometrical Introduction to Tensor Calculus*. Princeton, NJ: Princeton University Press, 2025, ISBN: 9780691269221.

[7] N. Islam, *Tensors and Their Applications*. New Delhi, India: New Age International (P) Ltd., Publishers, 2006, ISBN: 8122418385.

[8] A. Karagila, "Axiomatic Set Theory", Lecture Notes, 2025, [Online]. Available: https://karagila.org/files/set-theory-2017.pdf.

[9] M. Bays, "B1.1 Logic", Lecture Notes, 2025, [Online]. Available: https://people.maths.ox.ac.uk/bays/teaching/b1logic/B1.1.pdf.

[10] M. Bays, "B1.2 Set Theory", Lecture Notes, 2025, [Online]. Available: https://courses.maths.ox.ac.uk/pluginfile.php/128327/mod_resource/content/1/B1.2-HT25.pdf.

[11] P. Smith, *An Introduction to Gödel's Theorems*, 2nd ed. Cambridge, UK: Cambridge University Press, 2013, ISBN: 9781107606753.

[12] L. Alcock, *How to Think About Abstract Algebra*. Oxford, UK: Oxford University Press, 2021, ISBN: 9780198843382.

[13] J. A. Gallian, *Contemporary Abstract Algebra*, 10th ed. Boca Raton, FL: CRC Press, 2021, ISBN: 9780367651787.

[14] A. W. Knapp, *Basic Algebra and Advanced Algebra*, 2nd ed. East Setauket, New York: Published by the Author, 2016, ISBN: 9781429799928.

[15] S. Shahriari, *Algebra in Action: : A Course in Groups, Rings, and Fields*. Providence, Rhode Island: American Mathematical Society, 2017, ISBN: 9781470428495.

[16] J. Scherk, *Algebra, A Computational Introduction*, 2nd ed. Toronto, Canada: Creative Commons Copyright, 2009.

[17] A. Macdonald, *Linear and Geometric Algebra*. CreateSpace Independent Publishing Platform, 2025, ISBN: 9781453854938.

[18] D. Chatterjee, *Topology, General & Algebraic*. New Delhi, India: New Age International Pvt. Ltd., Publishers, 2007, ISBN: 9788122419436.

[19] T. Leinster, "General Topology", Lecture Notes, 2015, [Online]. Available: https://www.maths.ed.ac.uk/~tl/topology/topology_notes.pdf.

[20] T. Needham, "Introduction to Applied Algebraic Topology", Lecture Notes, 2019, [Online]. Available: https://tgda.osu.edu/math-4570-applied-algebraic-topology/.

[21] C. Perwass, *Geometric Algebra with Applications in Engineering*. Berlin Heidelberg: Springer-Verlag, 2009, ISBN: 9783540890676.

[22] D. Hildenbrand, *Foundations of Geometric Algebra Computing*. Berlin, Heidelberg: Springer-Verlag, 2013, ISBN: 9783642317941.

[23] L. Dorst, D. Fontijne, and S. Mann, *Geometric Algebra for Computer Science*. San Francisco, CA: Morgan Kaufmann, 2007, ISBN: 9780123749420.

[24] U. Görtz and T. Wedhorn, *Algebraic Geometry I: Schemes*, 2nd ed. Wiesbaden, Germany: Springer Fachmedien Wiesbaden GmbH, 2020, ISBN: 9783658307325.

[25] D. I. Spivak, *Category Theory for the Sciences*. Cambridge, MA: MIT Press, 2014, ISBN: 9780262028134.

[26] E. Riehl, "Category theory in context", Lecture Notes, 2023, [Online]. Available: https://math.jhu.edu/~eriehl/161/context.pdf.

[27] F. W. Lawvere and S. H. Schanuel, *Conceptual Mathematics*, 2nd ed. Cambridge, UK: Cambridge University Press, 2009, ISBN: 9780521719162.

[28] B. Fong and D. I. Spivak, *An Invitation to Applied Category Theory*. Cambridge, UK: Cambridge University Press, 2019, ISBN: 9781108711821.

[29] S. Roman, *An Introduction to the Language of Category*. Cham, Switzerland: Birkhäuser, 2017, ISBN: 9783319419176.

[30] J. Gallier and J. Quaintance, "Algebra, Topology, Differential Calculus, and Optimization Theory For Computer Science and Machine Learning", Lecture Notes, 2025, [Online]. Available: https://www.cis.upenn.edu/~jean/math-deep.pdf.

[31] S. Serovajsky, *Architecture of Mathematics*. Boca Raton, FL: CRC Press, 2021, ISBN: 9780429893544.

[32] M. M. Bronstein, J. Bruna, T. Cohen, and P. Veličković, *Geometric deep learning grids, groups, graphs, geodesics, and gauges*, 2021. arXiv: 2104.13478v2 [cs.LG].

[33] D. Simovici, *Mathematical Analysis for Machine Learning and Data Mining*. Singapore: World Scientific Publishing Co. Pte. Ltd., 2018, ISBN: 9789813229709.

[34] M. Ghayoumi, *Mathematical Foundations for Deep Learning*. Boca Raton, FL: CRC Press, 2026, ISBN: 9781040389072.

[35] S. Warner, *Pure Mathematics for Beginners*. Self-published: Get 800, 2018, ISBN: 9780999811757.

[36] R. M. Golden, *Statistical Machine Learning, A Unified Framework*. Boca Raton, FL: CRC Press, 2020, ISBN: 9781138484696.

[37] M. P. Deisenroth, A. A. Faisal, and C. S. Ong, *Mathematics For Machine Learning*. Cambridge, UK: Cambridge University Press, 2021, ISBN: 9781108455145.