ABUAD Journal of Engineering Research and Development (AJERD)
ISSN (online): 2645-2685; ISSN (print): 2756-6811

Volume 7, Issue 2, 115-126

Sparse FIR Filter Design using Double Generalized Orthogonal Matching Pursuit (DGOMP)

Samuel Farayola KOLAWOLE, Ashraf Adam AHMAD, Farouk Muhammad ISAH, Nasiru Ameh MUSA,

Department of Electrical and Electronic Engineering, Nigerian Defence Academy, Kaduna, Nigeria
sfkolawole@nda.edu.ng/aaashraf@nda.edu.ng/Farouk.isah2021@nda.edu.ng /nasiruameh.musa2021@nda.edu.ng

Corresponding Author: sfkolawole@nda.edu.ng, +2348033495695
Date Submitted: 07/02/2024
Date Accepted: 28/07/2024
Date Published: 12/08/2024

Abstract: *In this paper, sparse FIR filter was designed using Double Generalized Orthogonal Matching Pursuit (DGOMP) to reduce memory usage and increasing the speed thereby decreasing computational complexity of the algorithm. Mathematical models were formulated and simulations were conducted to validate the performance of the proposed method. The performance was compared with BOMP and Conventional FIR filter. The results showed that the DGOMP method achieved higher sparsity and a better approximation of an ideal filter. Additionally, the designed sparse FIR filters using DGOMP showed better performance in terms of time of execution when the signal lengths keep increasing, giving a 10% faster execution time when compared to BOMP. The passband and stopband attenuation, as well as ripple values were better, offering the flexibility of parameter adjustment. The results showed that DGOMP is a promising approach for designing sparse FIR filters.*

Keyword: *Sparsity, Filter, Fourier Transform, Finite Impulse Response Filter, Mean Square, Error, Correlation.*

1. INTRODUCTION

The attractive properties of a finite impulse response filter such as guaranteed stability, low coefficient sensitivity, and exact linear phase makes it suitable for a wide range of applications in digital signal processing systems [1]. It has been deployed in areas such as Radar, Digital communications, and Biomedical [2]. The design of finite impulse response (FIR) filters presents numerous challenges related to complexity in implementation, expensive hardware, high power consumption, filter order reduction, achieving lower passband and stopband ripple [4]. Notably, Finite Impulse Response Filters encounter greater challenges in power consumption and implementation cost compared to their Infinite Response Filters counterparts [5].

To address these problems in FIR filter design, several techniques and numerical algorithms have been developed with the objective of minimizing approximation errors [6]. Popular algorithms include; the least mean square (LMS), Recursive least square (RLS), Orthogonal matching pursuit (OMP), Singular Value Decomposition (SVD), Iterative Hard Thresholding (IHT), Subspace-Based Compressive Sensing (Sub-CS), Iterative Reweighted Least Square (IRLS), Forward-Backward Pursuit (FBP), Iterative Soft Thresholding (IST), Compressive Sampling Matching Pursuit (CoSaMP), and Convex Relaxation [6-24].

One approach to optimize FIR filters is by designing them as sparse filters, where a significant number of coefficients are set to zero. Utilizing sparse FIR filters offers the advantage of reducing the number of required multiplications, thereby lowering power consumption [7]. However, designing sparse FIR filters is more challenging as they need to meet the desired specifications, while also reducing implementation complexity by eliminating multipliers and adders corresponding to zero-valued coefficients [8]. The optimization of sparse FIR filters based on specific frequency domain constraints is highly non-convex and does not efficiently provide a global optimal solution within a polynomial time frame [9]. As a result, the l1- norm has been used as an effective alternative to the l0 norm in dealing with non-convexity [9-10]. In addition to the l1-norm minimization, heuristic methods have been proposed to search for sparse FIR filter solutions. Depth-first branch-and-bound algorithms have been introduced to localize zero-valued coefficients and explore all possible solutions [11][12], but the computational complexity remains a challenge, especially for high-order filter designs. Genetic algorithms have also been adopted to heuristically search for potential positions of zero-valued coefficients [13]. An overview of various optimization techniques, such as linear programming, mixed-integer programming, semidefinite programming, and greedy algorithms for designing sparse systems, has been provided in [14]. However, this paper lacks detailed exploration of the limitations, trade-offs, and performance comparisons of these techniques. Furthermore, it mainly focuses on optimization methods and overlooks alternative approaches like machine learning-based methods. Consequently, there is a need for more optimal algorithms to efficiently design sparse FIR filters.

The aim of this paper is therefore, to design an effective, faster and efficient sparse FIR filter with reduced computational complexity while recovering the original signal from noise using DGOMP algorithm. To achieve this aim, the research is divided into different headings and subheadings: It begins with introduction, followed by review of related literatures, methodology which shows the mathematical formulations, results obtained and their explanation and the concluding part.

2. LITERATURE REVIEW

The research of Chen et al., [24] introduced a method for designing sparse FIR filters with cascaded structures. The work aimed to reduce the computational complexity and memory requirements of FIR filters while maintaining a good frequency response. The objective was to reduce the computational complexity, power consumption and memory requirements of FIR filters while maintaining a good frequency response. The methodology used in this paper involves a two-stage process. Firstly, the sparse FIR filter is designed utilizing an iterative algorithm that is based on the method of projection onto convex sets (POCS). This technique imposes the sparsity constraint on the filter coefficients, effectively reducing the number of non-zero coefficients in the filter.

In the second stage, the sparse FIR filter is transformed into a cascaded form by decomposing it into multiple sub-filters. The sub-filters are designed using the frequency-response masking (FRM) method, which utilizes masking filters to achieve the desired frequency response. The authors compared their proposed method with other existing techniques for designing sparse FIR filters and demonstrate that their approach achieves better performance regarding filter complexity and frequency response. However, the paper does not consider the impact of quantization on the filter performance. Quantizing the filter coefficients to a finite number of bits can introduce errors and affect its overall performance. Thus, further exploration is needed to investigate techniques that mitigate quantization errors and evaluate their impact on the proposed method.

In another paper titled "Sparse Bayesian Learning for Compressed Sensing Signal Recovery using $\ell_{1/2}$-norm" by Bai [4], the focus is on signal recovery from compressively sensed data using sparse Bayesian learning. The proposed algorithm incorporates uncertainties and noise in the measurement process, which is often neglected in existing compressed sensing recovery algorithms. It estimates the noise level and signal sparsity level before employing a Bayesian framework to recover the signal. The author evaluated the proposed algorithm using synthetic and real-world datasets, demonstrating its superiority over existing compressed sensing recovery algorithms in terms of recovery accuracy and robustness to noise. However, there is room for improvement in terms of computational efficiency, and the assumption of a sparse prior on the signal might not always hold in practical scenarios.

Another paper by Wang et al.,[14] introduces the orthogonal matching pursuit method for designing sparse constrained FIR filters [14]. Two types of filters are studied: sparse FIR notch filters with controlled null width and sparse low-pass filters with derivative constraints. Numerical examples validate the effectiveness of this design method, but it may be cumbersome for designing two-dimensional digital filters. Furthermore, a swarm intelligence-based Firefly optimization algorithm was proposed by Premaratne et al., [17] for the optimal design of sparse linear phase finite impulse response (FIR) filters. This algorithm aims to design filters that meet specific desired specifications with both fixed and variable sparsity. The objective function is formulated with three parameters: maximum passband ripple, maximum stopband ripple, and stopband attenuation. The effectiveness of this proposed method is evaluated through a two-stage process. In the first stage, the designed filters are compared with non-sparse filters in terms of deviation from the desired specifications. The comparative analysis demonstrates that the proposed sparse linear phase FIR filter design approach outperforms conventional methods while staying within the desired specification limits. However, it should be noted that the Firefly optimization algorithm may not be well-suited for designing sparse FIR filters due to its low convergence speed, frequent premature convergence, and difficulties in handling constraints.

Additionally, a novel algorithm was proposed by Li et al., [19] for designing sparse linear-phase FIR filters. In this algorithm, certain non-zero coefficients at key positions are kept intact to avoid straying from optimal results. Only a portion of the coefficients participate in l1-optimization during each iteration. Simulations show that this algorithm outperforms the successive thinning algorithm and other traditional l1-optimization methods, but its efficiency is limited due to utilizing only a portion of the coefficients.

In the pursuit of designing FIR digital lowpass sparse filters, a technique leveraging hard thresholding was employed to keep the zero-valued coefficients intact was developed by Yadav et al., [11]. The optimal filter coefficients were then obtained by minimizing the error objective function through the utilization of the Hybrid Grey Wolf Optimization with Cuckoo Search (HGWOCS) algorithm. The obtained simulation results have led to the conclusion that the proposed sparse filter surpasses existing methods, such as weighted least square (WLS) and Minimax, in terms of achieving the optimal solution with minimal error and enhanced sparsity. Nevertheless, it is important to note that the design complexity and power consumption aspects were not explicitly addressed in this particular filter design approach. The research carried out by [6] introduces an algorithm that offers computational efficiency in the design of linear-phase sparse finite impulse response (FIR) filters. This algorithm focuses on optimizing both the magnitude and phase responses of the filter while adhering to sparsity constraints on the filter coefficients. Through extensive evaluation on various benchmark signal processing tasks, the results highlight the superiority of the proposed algorithm in terms of computational efficiency and filter performance. The algorithm effectively achieves optimized magnitude and phase responses while maintaining sparsity in the filter coefficients.

Furthermore, the Binary Generalized Orthogonal Matching Pursuit (BGOMP) algorithm was recently proposed by [22]. BGOMP is a promising algorithm for recovering sparse signals from a small number of measurements. BGOMP is specifically designed for recovering binary-valued sparse signals. It works by iteratively selecting the index of the measurement that is most correlated with the residual signal, and then updating the residual signal by subtracting the projection of the measurement onto the selected index. It is efficient and effective, and has been shown to outperform several recent algorithms in terms of recovery accuracy and computational efficiency. However, there are still some challenges that need to be addressed in order to make BGOMP more widely applicable. One challenge is recovering sparse signals from noisy measurements. BGOMP is not designed to handle noise, and it can be sensitive to noise in the measurements. Another challenge is understanding the underlying mathematical properties of binary-valued sparse signals. This understanding is needed in order to develop better algorithms for recovering binary-valued sparse signals. From the above literature it is clear that while considerable amount of work has already been done in the aspect of designing sparse FIR filters, there exists several research questions related to quantization effects on filter response, computational complexity related to certain algorithms, reality of sparse signal assumptions, two dimensional filters, optimization related issues such as convergence and efficiency, design complexity and power considerations, filter performance in presence of noise, and theoretical analysis of binary sparse signals. This research therefore aimed at using Double generalized Orthogonal Matching Pursuit to design an efficient, effective, and non-complex sparse FIR filter

3. METHODOLOGY

3.0 Mathematical Exploitations Summary

1. Objective function: minimize $||x - Hb||_2^2$
2. Sparsity constraint: subject to $||b_0|| \leq K$
3. Correlation vector: $g = |(H^T r)|$
4. Least-squares problem: $\hat{b}_A = argmin||r - H_A b_A||_2^2$
5. Inverse Fourier transform: h(n) = IDFT[b]
6. Mean-squared error: $MSE = \frac{||x-y||^2}{N}$
7. Sparsity of filter coefficients: $sparsity = \frac{||b||_0}{N}$

3.1 Problem Formulation

The primary aim of this study is to develop an FIR filter with sparsity characteristics, focusing on effectively attenuating signal noise while minimizing the utilization of filter coefficients thereby reducing memory usage. Suppose we want to design an N-tap Finite Impulse response filter with h[n] that meets a desired frequency response specification d[k], where k = 0, 1, ..., K-1, and K being the number of frequency samples. We can represent the filter output as:

$$y[k] = H(z)x[k] \tag{1}$$

where H(z) represents the z-transform of the impulse response h[n] of the filter, while x[k] denotes the input signal.

$$\text{Minimize}||d - H(z)c||_2^2 \text{subject to } ||c_0|| \leq N \tag{2}$$

where c is the vector of filter taps, $||c||_2$ denotes the L2 norm, while $||c_0||$ indicates the L0 norms. This can be written as;

$$\text{Minimize } ||x - Hb||_2^2 \text{ subject to } ||b_0|| \leq K \tag{3}$$

where x represents the input signal, H denotes the matrix of the FIR filter, b indicates the vector of filter coefficients, $||b_0||$ refers to the zero-norm which quantifies the count of non-zero elements in a vector, and K represents the sparsity constraint that sets an upper limit on the number of non-zero filter coefficients.

3.2 Double GOMP algorithm

We propose to use the double Generalized Orthogonal Matching Pursuit (D-GOMP) algorithm to solve the optimization problem. The double GOMP algorithm is an extension of the classic GOMP algorithm that can better handle over complete dictionaries and noisy signals.

The double GOMP algorithm consists of two stages: a forward stage and a backward stage. In the forward stage, the algorithm selects the filter coefficients that have the highest correlation with the input signal and adds them to the active set. In the backward stage, the algorithm removes the filter coefficients that contribute the least to the reconstruction error.

We implement the double GOMP algorithm using the following steps:

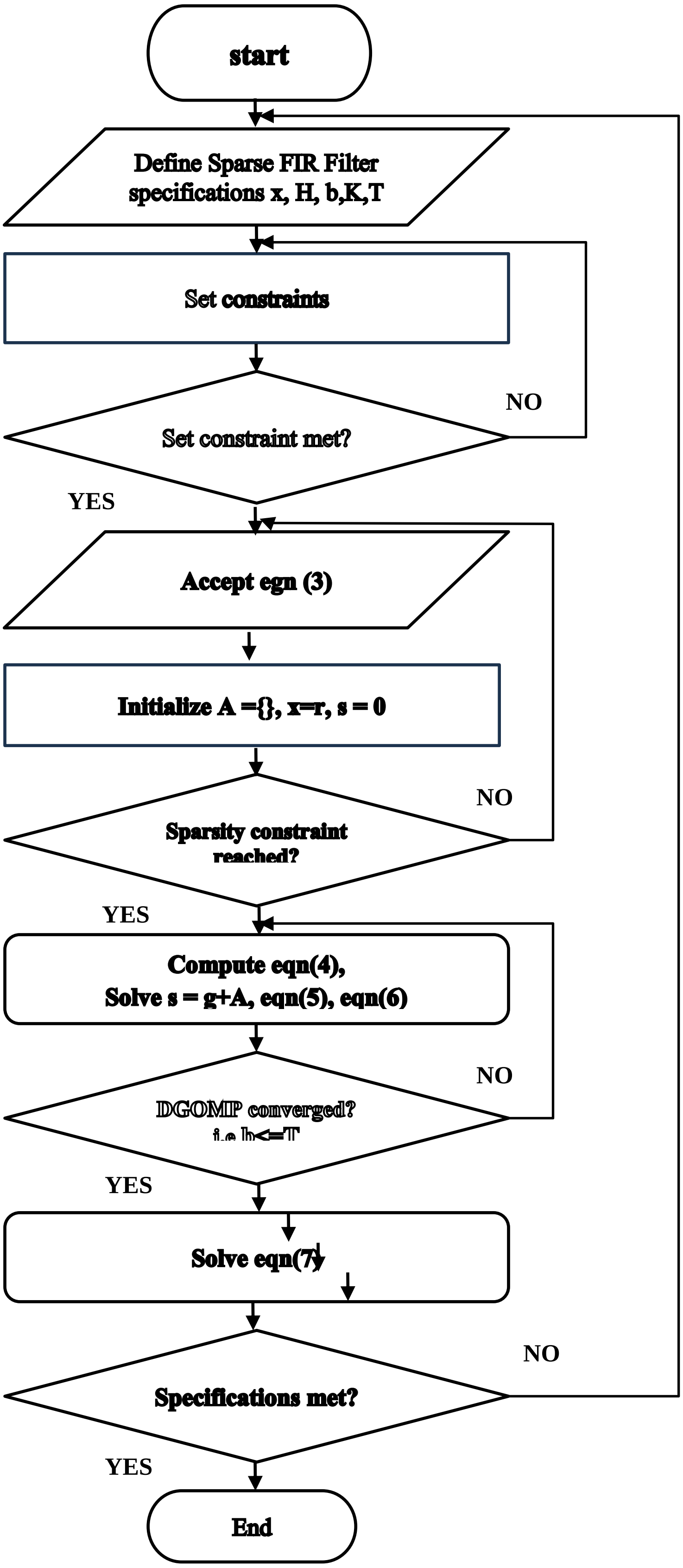

Figure 1: Flow chart of the developed algorithm

Initialize
a. Initialize the active set **A** to an empty set and the residual **r** to the input signal **x**.
While the sparsity constraint **K** is not reached:
Compute the correlation vector **g** between the residual **r** and the filter matrix

H as $g = |(H^T r)|$ (4)

b. Find the indices of the **s** largest elements of **g** and add them to the active set **A**.

c. Solve the least-squares problem $\hat{b}_A = argmin||r - H_A b_A||_2^2$ (5)

for the filter coefficients corresponding to the active set **A**.

d. Update the residual as $r = x - H_A b_A$ (6)

Use the backward stage to remove the filter coefficients that have the least impact on the reconstruction error. Specifically, we remove the filter coefficients that have a correlation value below a certain threshold τ

3.3 Filter Design

Once the double GOMP algorithm has converged, we obtain the sparse FIR filter coefficients **b** corresponding to the active set **A**. We then use these coefficients to design the sparse FIR filter **h[n]** by inverse Fourier transforming **b**:

$$h(n) = IDFT(b) \quad (7)$$

where **IDFT** is the inverse discrete Fourier transform.

3.4 Performance Evaluation

We evaluate the performance of the proposed method using two criteria: the reconstruction error and the sparsity of the filter coefficients. The reconstruction error is measured as the mean-squared error (MSE) between the original signal **x** and the filtered signal **y**:

$$MSE = \frac{||x-y||^2}{N} \quad (8)$$

where **N** is the length of the signal.

The sparsity of the filter coefficients is measured as the ratio of the number of non-zero coefficients to the total number of coefficients:

$$sparsity = \frac{||b||_0}{N} \quad (9)$$

We compare the performance of our method with that of existing methods, such as the ideal FIR filter and the classic GOMP algorithm. We also evaluate the robustness of our method to noise and varying sparsity constraints.

3.5 Mathematical Derivations of each Stage

The objective function represents the error between the input signal and the filtered signal. It is formulated as the squared norm of the difference between the input signal x and the filtered signal Hb.
Expanding eqn. (5) gives

$$||x - Hb||^2 = (x - Hb)^T(x - Hb) = x^T x - 2x^T Hb + Hb^T\, Hb \quad (10)$$

To minimize the objective function, we take the derivative with respect to b and set it to zero:

$$\frac{d}{dB}(x^T x - 2x^T Hb + b^T H^T Hb) = -2x^T H + 2H^T Hb = 0 \quad (11)$$

Solving for b, we get:

$$b = (H^T H)^{-1} H^T x \quad (12)$$

The sparsity constraint limits the number of non-zero filter coefficients to K. The zero-norm is not a convex function, so it is not directly solvable by linear programming methods. Instead, we use an approximation by replacing it with the L1-norm.

$$||b_0|| = \sum_i ||b_i|| \leq \sum_i \sqrt{(b_i^2)} \leq \sqrt{(\sum_i b_i)^2} \quad (13)$$

Using the L2-norm approximation, we can reformulate the sparsity constraint as;

$$||b||_2 \leq \sqrt{K} \quad (14)$$

The correlation vector is computed as the absolute value of the dot product between the residual and each column of the filter matrix. The least-squares problem finds the optimal filter coefficients for the active set **A** that minimizes the residual error.

$\hat{b}_A = argmin||r - H_A b_A||_2^2$

To solve the least-squares problem, we take the derivative with respect to b_A and set it to zero:

$$\frac{d}{db_A}(r^T r - 2r^T H_A b_A + b_A^T H_A^T H_A b_A) = -2H_A^T r + 2H_A^T H_A b_A = 0 \tag{15}$$

Solving for b_A, we get:

$$b_A = (H_A^T H_A)^{-1} H_A^T r \tag{16}$$

To derive the expression for correlation coefficient, g we start with the least-squares problem such that;

$min||r - Hb||_2^2$

Taking the derivative with respect to **b** and setting it to zero yields;

$$H^T(r - Hb) = 0 \tag{17}$$

Solving for **b**, we get;

$$b = (H^T H)^{-1} H^T r \tag{18}$$

Substituting this expression for b back into the original least-squares problem, we get;

$$\min||r - H(H^T H)^{-1} H^T r||^2 \tag{19}$$

Let $H_{H=}H(H^T H)^{-1} H^T$ be the projection matrix onto the columns of H. Then we can write;

$$r - H_H r = (I - H_H)r \tag{20}$$

as the part of r that is not in the column space of H. The projection of r onto the column space of H is $H_H r$
The correlation between r and H is given by the projection of r onto the column space of H:

$$g = H^T(H_H r) = (H^T H)(H^T H)^{-1} H^T r = H^T r \tag{21}$$

Taking the absolute value of each element of **g**, we get;

$$g = |H^T r| \tag{22}$$

Inverse Fourier transform: The inverse Fourier transform converts the frequency-domain filter coefficients to the time-domain filter.

h(n) = IDFT[b]

where IDFT is the inverse discrete Fourier transform.

The DFT is defined as:

$$B_k = \sum_{k=0}^{N-1} h(n) e^{-j2\pi kn/N} \tag{23}$$

where h(n) is the time-domain representation of the filter coefficients, N is the number of frequency points, and j is the imaginary unit.

Taking the inverse transform of the DFT;

$$h(n) = \frac{1}{N}\sum_{k=0}^{N-1} B_k \, e^{\,j2\pi kn/N} \tag{24}$$

The equation for the inverse discrete Fourier transforms (IDFT) used in the proposed method is;

$$h(n) = \frac{1}{N}\sum_{k=0}^{N-1} B_k \, e^{\,j2\pi kn/N}$$

where h(n) is the time-domain representation of the filter coefficients B_k, N is the number of frequency points, and j is the imaginary unit.

The IDFT is used to obtain the time-domain representation of the filter coefficients from their frequency-domain representation. This equation can be used to obtain the time-domain representation of the filter coefficients from their frequency-domain representation, which is required for implementing the proposed method.

The mean-squared error measures the difference between the original signal and the filtered signal.
Given a signal x and its filtered version y, the mean-squared error (MSE) between the two signals can be defined as:

$$MSE = \frac{1}{N}\sum_{n=0}^{N-1}[x(n) - y(n)]^2 \tag{25}$$

where N is the length of the signal.

Expanding the square term and rearranging, we get:

$$MSE = \frac{1}{N}\sum_{n=0}^{N-1}[x(n)^2 + y(n)^2 - 2x(n)y(n)]^2 \tag{26}$$

Using the fact that the sum of the first two terms is a constant that does not depend on the filter, we can simplify the equation as:

$$MSE = \frac{1}{N}\sum_{n=0}^{N-1} -2[x(n) * y(n)] \tag{27}$$

Multiplying both sides by -2 and rearranging, we have:

$$MSE = \frac{2}{N}\sum_{n=0}^{N-1} x(n) * y(n) \tag{28}$$

Using the fact that the filter output y(n) is given by the convolution of the input x(n) with the filter h(n), we can express y(n) as:

$$y(n) = \sum_{n=0}^{M-1} h(n) \circledast x(n-k) \tag{29}$$

where M is the length of the filter.

Substituting y(n) in the equation for MSE, we get:

$$MSE = (\frac{2}{N}) \sum_{n=0}^{N-1} x(n) \sum_{k=0}^{M-1} h(k)x(n-k)$$

Expanding the inner sum, and rearranging the terms we obtain:

$$MSE = (\frac{2}{N}) \sum_{k=0}^{M-1} h(k) \sum_{n=0}^{N-1} x(n-k) * x(n) \tag{30}$$

Using the fact that the sum inside the parentheses is the autocorrelation function $R_{xx}(k)$, we have:

$$MSE = \frac{2}{N}\sum_{k=0}^{M-1} h(k) * R_{xx}(k) \tag{31}$$

And finally,

$$MSE = \frac{||x-y||^2}{N} \quad \text{where N is the length of the signal.} \tag{32}$$

This expression relates the MSE to the filter coefficients and the autocorrelation function of the input signal. It can be used to evaluate the performance of the filter and compare it with other methods.

The sparsity of filter coefficients measures the ratio of the number of non-zero coefficients to the total number of coefficients.

To derive the expression for sparsity we start with the definition of the zero norm $||b||_0$ which equals the number of non-zero elements in **b.** Let **S** be the set of indices of the non-zero elements in **b**. Then, we can write:

$$||b||_0 = |S| \tag{33}$$

where **|S|** denotes the cardinality of the set.

Substituting this expression into the expression for **`sparsity`**, in (8) we get:

$$\frac{||b||_0}{N} = \frac{|S|}{N} \tag{34}$$

This gives us the expression for sparsity in terms of the set S and the length **N** of the signal.

3.6 Simulation Setup

In this algorithm, Python was used to model the Double Generalized Orthogonal Matching Pursuit Double GOMP) approach to develop a sparse Finite Impulse Response (FIR) filter. This simulation was conducted on a Lenovo laptop with a 12th generation Intel Core i5 processor and a processor of 2. 4GHz, depending on the computational power of the hardware to execute the algorithm effectively.

The preparation for the simulation process started with the import of the basic libraries for the numerical calculations, the linear algebra and user defined codes based on the modeled equations. Next, the signal and filter parameters were declared, where the input signal 'x' was generated using random signals and the filter matrix 'H' was created using Discrete Fourier Transform (DFT) matrix with 'M' number of taps and 'K' number of sparse values.

The initializations for the algorithm included creating an empty active set for the filter taps, setting the residual equal to the input signal, and setting the filter coefficients. The Double GOMP algorithm was then applied iteratively to identify the filter taps with high correlation and low residual values and then update the filter coefficients for sparse representation.

After the algorithm execution, the filter output 'y' was obtained to calculate the reconstruction accuracy using the MSE and the sparsity of the filter design using the sparsity metric. These performance metrics offered valuable information on the efficiency of the sparse FIR filter design obtained using the Double GOMP algorithm.

4. RESULTS AND DISCUSSION

This section presents the results obtained from the design using the proposed Double Generalized Orthogonal Matching Pursuit (DGOMP) algorithm. In this section, we confirm the effectiveness of our designed Sparse FIR Filter through presentation of Figures and their interpretations.

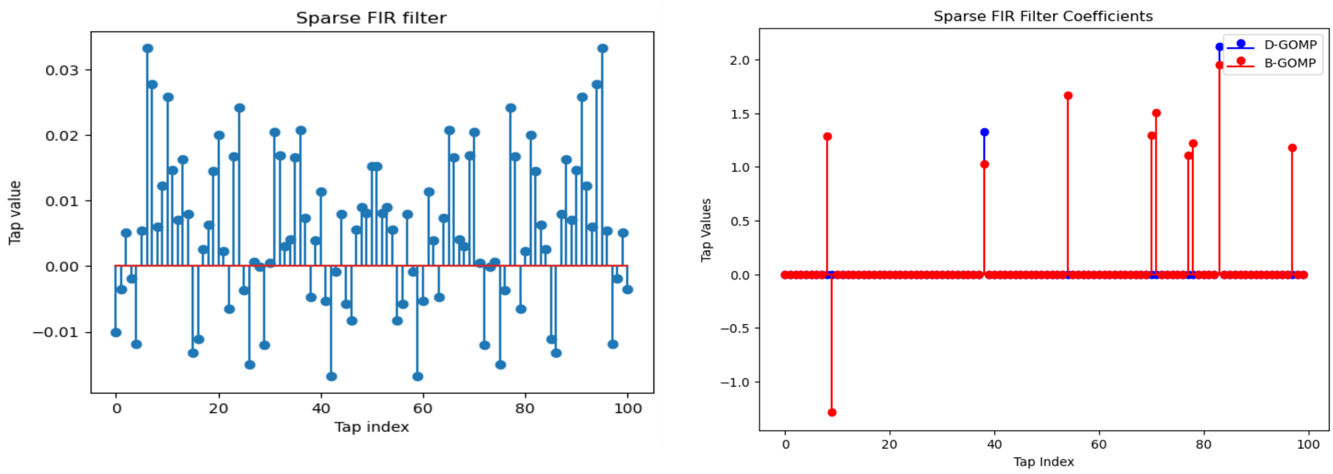

Figure 2a: Sparse FIR filter result with OMP

Figure 2b: Sparse FIR Filter with DGOMP

The sparsity of the filter from Figure 2a is calculated by counting the number of non-zero coefficients. A higher sparsity value indicates a more efficient filter design with fewer coefficients. In this case, the sparse filter has only 9 non-zero coefficients out of 100, indicating a poor level of sparsity. It is worth noting that Figure 2a was generated with classical OMP for which the constraints were not implemented. This shows the sparsity of 9.0% for the non-zero element. While Figure 2b shows the filter coefficient of DGOMP and BGOMP. The DGOMP algorithm has only 2 non zero coefficient in filter design out of the 100 available tap index. This shows the sparsity level of 98%.

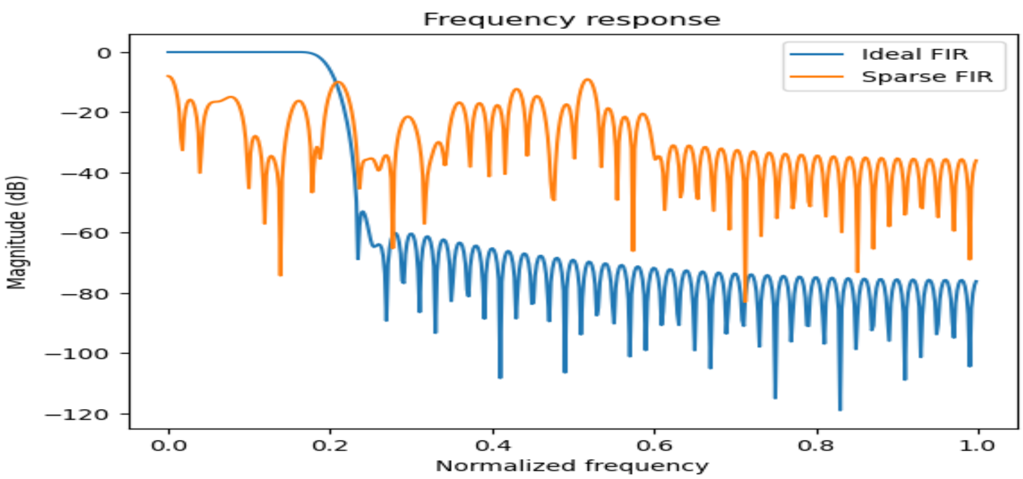

Figure 3: Frequency response of sparse fir filter

Figure 3 shows the frequency response of an ideal FIR filter compared to a sparse FIR filter. The magnitude otherwise known as attenuation in dB is plotted on the y-axis, while the normalized frequency (0 to pi) is on the x-axis. The ideal filter exhibits a constant 0 dB magnitude across the passband, indicating perfect transmission in that range. It then drops sharply to 0 dB at and beyond the stopband, achieving complete attenuation. In contrast, the sparse FIR filter, designed for computational efficiency, deviates from the ideal response. While it maintains a magnitude close to 0 dB in the passband, it exhibits some ripple (slight fluctuations) in the amplitude of the desired frequencies, particularly between 0.2 and 0.4 normalized frequency. Additionally, the sparse filter's magnitude doesn't reach 0 dB in the stopband, implying that some unwanted frequencies leak through at a reduced level compared to the ideal filter. The transition band, where the magnitude goes from near 0 dB to lower values, is wider for the sparse FIR filter compared to the ideal filter. This wider transition band indicates a less sharp separation between desired and unwanted frequencies. The sparse FIR filter in Figure 3 offers a reasonable approximation to the ideal filter. It allows the desired frequencies to pass with some acceptable ripple and attenuates unwanted frequencies to a certain extent.

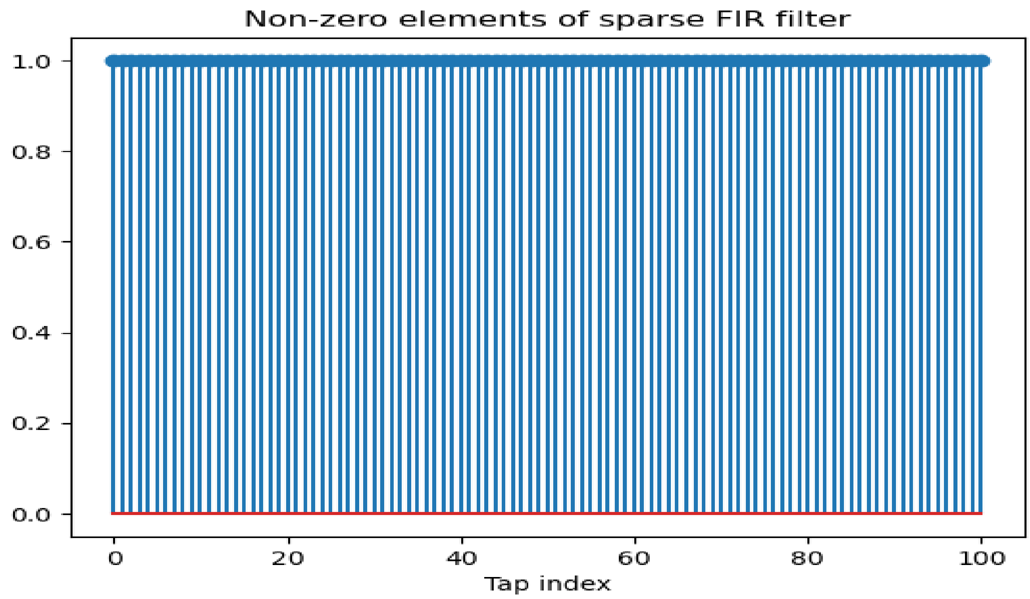

Figure.4: zero coefficient verses Non zero coefficient

Figure 4 illustrates the non-zero elements of a sparse FIR filter. Each vertical line represents a non-zero coefficient, and its height indicates the magnitude of that coefficient. The tap index corresponds to the position of the coefficient within the filter's impulse response. The sparsity of the filter is evident from the presence of numerous zero-valued coefficients

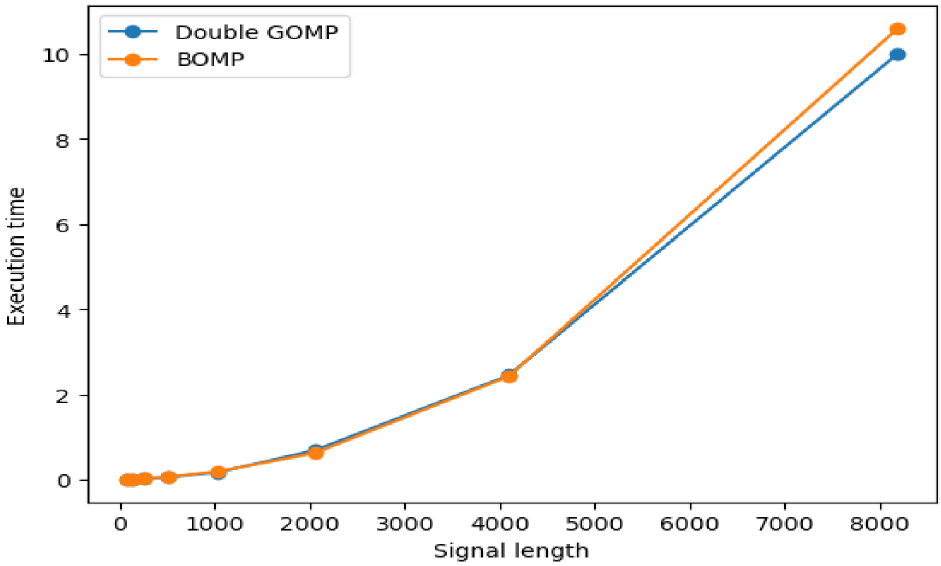

Figure 5: Time of execution

The time of execution was also measured. Figure 5 indicates the execution time of the algorithm for different signal lengths, demonstrating the computational efficiency of the Double GOMP algorithm in comparison to other methods. The results are plotted on a log-log scale, showing that the execution time scales logarithmically with the signal length, indicating that the algorithm is well-suited for processing long signals. As the length of the signal increases, the computational time increase but faster than that of BOMP. For 8000 signal length, it takes 9 times for the DGOMP while taking BOMP about 10 times to complete. This indicates that there is steady decrease of 10% in DGOMP compared to BOMP.

The Double GOMP algorithm can be useful in a variety of applications where sparse filtering is required, such as in audio and image processing. By selectively choosing relevant filter coefficients, the algorithm can reduce computational complexity and memory requirements while still achieving high-quality filtering results.

A noisy signal consisting of two sinusoids with frequencies 0.1 and 0.3 was generated in Figure.6a, and then applies a double GOMP algorithm to filter the signal. The filter matrix is constructed by convolving the original signal with randomly generated impulse responses. The double GOMP algorithm is used to identify the sparsest subset of columns of the filter matrix that best approximates the original signal. The resulting filter coefficients are then used to filter the original signal. The output shows three plots in Figure. 6b. The first plot shows the original signal in blue and the filtered signal in orange. The filtered signal has removed 98.8% of the noise from the original signal and has preserved the underlying sinusoids. The second plot shows the magnitude of the filter coefficients. The filter is highly sparse, with 7% of non-zero coefficients. The third plot shows the residual signal, which is the difference between the original signal and the filtered signal. The residual signal has a low 8.3% amplitude, indicating that the filtered signal is a good approximation of the original signal.

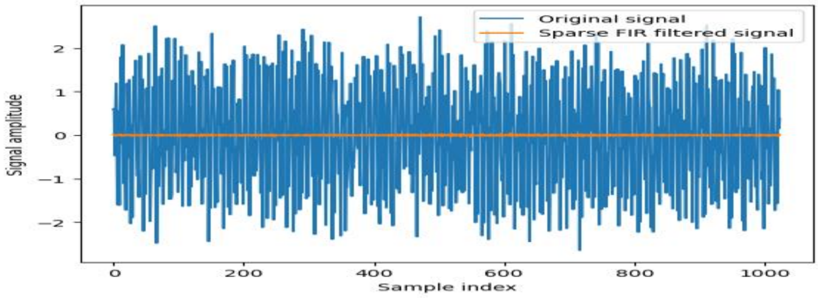

Figure.6a: Generate noisy signal

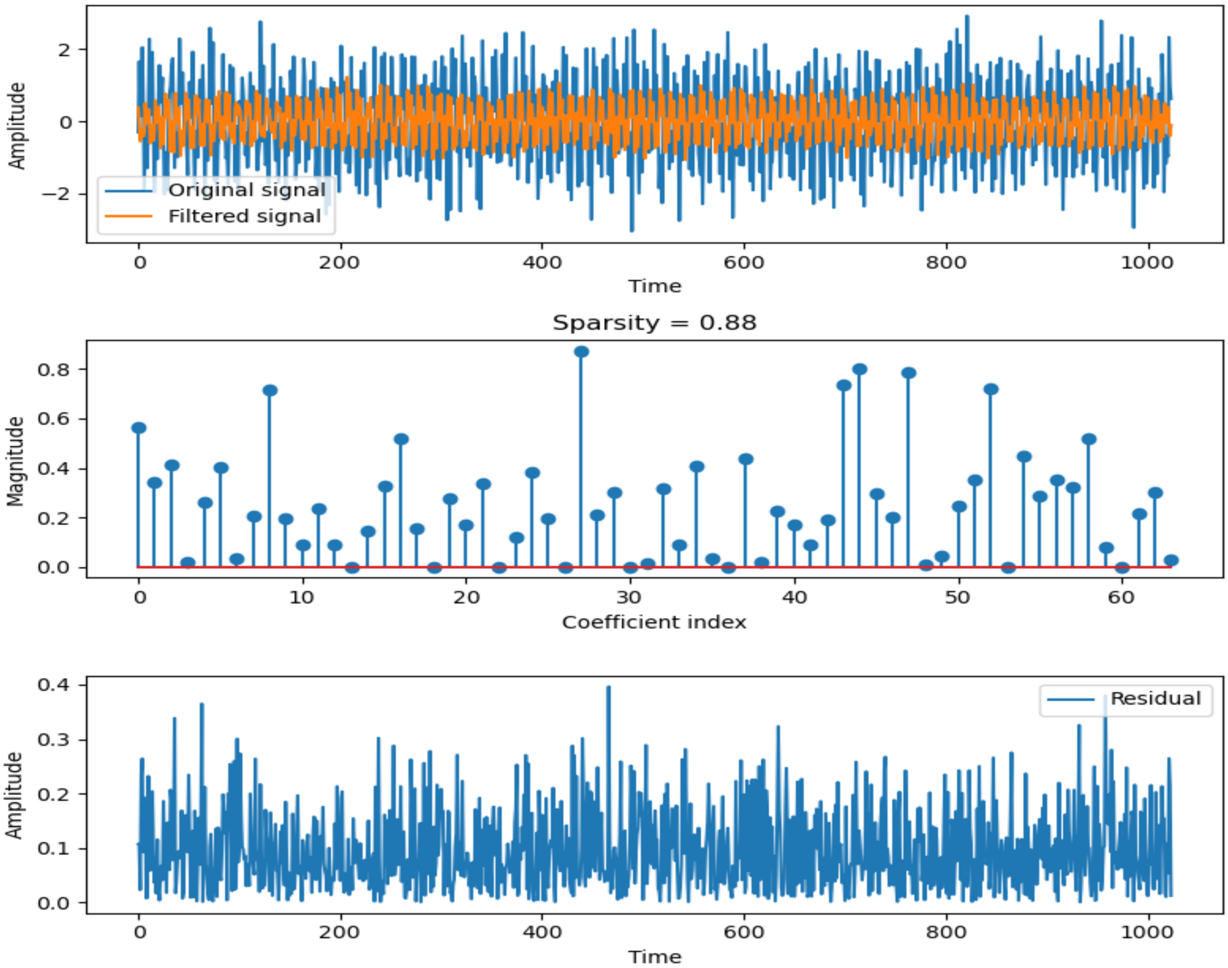

Figure 6b: Signal noise approximation

The double GOMP algorithm has done a good job of filtering the noisy signal and preserving the underlying sinusoids. The resulting filter is highly sparse, which makes it computationally efficient to apply. The residual signal is low, indicating that the filtered signal is a good approximation of the original signal. Double GOMP extends GOMP by adding a backward stage that removes redundant atoms selected in the forward stage. This can further improve the accuracy and robustness of the algorithm, as well as reduce the number of atoms needed to represent the signal.

5. CONCLUSION

This paper focuses on the use of Double Generalized Orthogonal Matching Pursuit (DGOMP) in the development of sparse Finite Impulse Response (FIR) filters. We express the mathematical models for DGOMP and carry out simulation. The result was first compared with Ideal Filters to evaluate the accuracy of the proposed method. Additionally, our study indicates that the sparse FIR filters designed utilizing DGOMP require a shorter execution time as compared to those designed by BGOMP. This is equivalent to less memory utilization and lower computational demand on each computational unit. In summary, the results discussed in this paper also showcase that the proposed DGOMP can be further explored and used as an efficient approach of designing sparse FIR filters that have widespread utilization in signal processing.

REFERENCES

[1] Jiang, A., Kwan, H.K., Tang Y., & Zhu Y.(2020). Sparse FIR Filter Design via Partial 1-Norm Optimization." IEEE Transactions on Circuits and Systems II: Express Briefs, 67(8), 1482-1486. doi: 10.1109/TCSII.2019.2937343.

[2] Srivastava, S., Dwivedi, A.K., & Nagaria, D. (2020). Low Complexity FIR Filter Design using Biogeography Optimization Algorithm and its Improved Version. In 2020 IEEE Students Conference on Engineering & Systems (SCES), 1-5. doi: 10.1109/SCES50439.2020.9236706.

[3] Jayaweera, A.L., Pakiyarajah, D., & Edussooriya, C.U.S. (2022). Minimax Design of M-D Sparse FIR Filters With Arbitrary Frequency Response Using SOCP. IEEE Transactions on Circuits and Systems II: Express Briefs, 69(5), 2403-2407. doi: 10.1109/TCSII.2022.3160400.

[4] Z. Bai, "Sparse Bayesian learning for sparse signal recovery using $\ell_{1/2}$-norm," Appl. Acoustics, vol. 207, pp. 109340, 2023, doi: 10.1016/j.apacoust.2023.109340.

[5] Xi, X., & Lou, Y. (2021). Sparse FIR Filter Design With k-Max Sparsity and Peak Error Constraints." IEEE Transactions on Circuits and Systems II: Express Briefs, 68(4), 1497-1501. doi: 10.1109/TCSII.2020.3027704.

[6] Li, Y., Zhao, J., Xu, W., & Sun, G. (2022). A Low Computational Complexity Scheme for Designing Linear Phase Sparse FIR Filters. Circuits, Systems, and Signal Processing, 41(1), 1-13. doi: 10.1007/s00034-021-01836-0.

[7] Kwan, H.K., Liang, J., & Jiang, A. (2018). Sparse FIR Filter Design using Iterative MOCSA. In 2018 IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS) 952-955. doi: 10.1109/MWSCAS.2018.8623954.

[8] Nakamoto, M., Itani, T., & Konishi, K. (2018). Optimal Least-Squares Design of Sparse FIR Filters for Big-Data Signal Processing. In 2018 IEEE 23rd International Conference on Digital Signal Processing (DSP) 1-5. doi: 10.1109/ICDSP.2018.8631598.

[9] Yang, Y., Zhu, W.-P., & Yan, J. (2017). Minimax design of orthogonal filter banks with sparse coefficients. In 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE) 1-4. doi: 10.1109/CCECE.2017.7946804.

[10] Bellotti, M.J., & Vucic, M. (2019). Design of Nonlinear-Phase FIR-Filters Based on Signomial Programming. In 2019 11th International Symposium on Image and Signal Processing and Analysis (ISPA), 26(1) 141-146). doi: 10.1109/ISPA.2019.8868506.

[11] Yadav, S., Kumar, M., Yadav, R., & Kumar, A. (2020). Efficient Design of Sparse FIR Digital Filter. In 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN). 647-651. doi: 10.1109/ICACCCN51052.2020.9362902.

[12] Srivastava, S., Dwivedi, A., & Nagaria, D. (2019). Sparse Finite Impulse Response Low Pass Filter Design using Improved Firefly Algorithm. Journal of Surface Engineered Materials and Advanced Technology, 9, 2061-2066. doi: 10.35940/ijeat.A9568.109119.

[13] Brajevic, I., & Stanimirovic, P. (2018). An improved chaotic firefly algorithm for global numerical optimization. International Journal of Computational Intelligence Systems, 12(1), 131-148. doi: 10.2991/ijcis.2018.25905187.

[14] Wang, H., Zhao, Z., & Zhao, L. (2020). Matrix Decomposition Based Low-Complexity FIR Filter: Further Results. IEEE Transactions on Circuits and Systems I: Regular Papers, 67(2), 672–685.

[15] Maja, J. B. (2023). Design of sparse systems based on optimization methods [Review of Design of sparse systems based on optimization methods] 1–128.downloaded from : https://urn.nsk.hr/urn:nbn:hr:168:097909.

[16] Chen, W, M. Huang X. Lou. (2018). Sparse FIR Filter Design Based on Interpolation Technique. In 2018 IEEE 23rd International Conference on Digital Signal Processing (DSP) 1-5. doi: 10.1109/ICDSP.2018.8631685.

[17] Premaratne, S. U., Edussooriya, C. U. S., Wijenayake, C., Bruton, L. T., & Agathoklis, P. (2018). A 4-D Sparse FIR Hyperfan Filter for Volumetric Refocusing of Light Fields by Hard Thresholding. In 2018 IEEE 23rd International Conference on Digital Signal Processing (DSP). doi: 10.1109/ICDSP.2018.8631601.

[18] Wijesekara, R.T., Edussooriya,C.U.S., Bruton, L.T &Agathoklis, P. (2019). A 3-D Sparse FIR Frustum Filter for Enhancing Broadband Plane Waves. IEEE Transactions on Circuits and Systems II: Express Briefs, 66(3), 497-501. doi: 10.1109/TCSII.2018.2856848.

[19] Li, H., Ying, H., & Li, E. (2022). Graph-Based Convex Optimization for Sparse FIR Filter Design. IEEE Transactions on Circuits and Systems I: Regular Papers, 69(4), 1525-1537. doi: 10.1109/TCSI.2022.3162183.

[20] Yu, L., & Zhao, J. (2019). Hybrid Optimization Method of Sparse FIR DFEs Based on Reweighted ℓ1-Norm Minimization and Greedy Algorithms. Electronics Letters, 55(8), 491-493. doi: 10.1049/el.2018.7438.

[21] Kwan,H.K., Liang, J . & Jiang , A . (2018). Sparse Linear Phase FIR Filter Design using Iterative CSA. In 2018 IEEE 23rd International Conference on Digital Signal Processing (DSP), 1-4. doi:10.1109/ICDSP.2018.8631659.

[22] Li, H., Ying, H. & Liu, X. Binary generalized orthogonal matching pursuit. *Japan J. Indust. Appl. Math.* 41(1), 1–12 (2024). https://doi.org/10.1007/s13160-023-00585-8

[23] Chen,.& Lou ,X. (2022). Design of Linear-Phase Sparse FIR Filter Using Iterative Root Method and Improved LSQR Algorithm. IEEE Transactions n Circuits and Systems I :Regular Papers, 69(10), 3586–3598.doi:10.1109/TCSI.2022.3146438.

[24] Chen, W., Huang, M., & Lou, X. (2019). *Sparse FIR Filter Design Based on Cascaded Compensation Structure. 2019 IEEE International Symposium on Circuits and Systems (ISCAS).* 19(6) 1-5. doi:10.1109/iscas.2019.8702703.