IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 16 November 2025, accepted 13 December 2025, date of publication 16 December 2025,
date of current version 22 December 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3645152

RESEARCH ARTICLE

Decomposition-Free DOA Estimation Method for Multiple Coherent Sources

AHMED A. HUSSAIN[1], (Student Member, IEEE), NIZAR TAYEM[2], (Member, IEEE), AND ABDEL HAMID SOLIMAN[3]

[1]Department of Electrical Engineering, Prince Mohammad Bin Fahd University, Al Khobar 31952, Saudi Arabia
[2]Department of Engineering and Technology, East Texas A&M University, Commerce, TX 75428, USA
[3]School of Digital, Technology, Innovation, and Business, University of Staffordshire, ST4 2DE Stoke-on-Trent, U.K.

Corresponding author: Ahmed A. Hussain (ahussain1@pmu.edu.sa)

ABSTRACT Direction of Arrival (DOA) estimation of highly correlated or coherent sources remains a challenging problem, particularly for subspace-based methods such as MUSIC and Root-MUSIC, which require spatial smoothing or eigenvalue decomposition. This work introduces a decomposition-free DOA estimation algorithm that exploits the inherent Toeplitz structure of uniform linear arrays (ULAs). The method constructs a structured Toeplitz matrix directly from the received data vector, forms a real, symmetric compound matrix to enable real-valued processing, and extracts structured sub-blocks without matrix decomposition. A polynomial matrix is then generated, and the DOAs are obtained by solving its roots, thereby bypassing covariance decomposition entirely. This design reduces computational burden, improves numerical stability, and inherently accommodates coherent sources without the need for spatial smoothing. Computational complexity analysis shows significant savings compared to subspace methods, while MATLAB simulations across various SNR, snapshot, and array size conditions validate estimation accuracy. Furthermore, a hardware realization on the NI PXIe-7993 FPGA platform demonstrates that the decomposition-free structure requires minimal hardware resources and execution cycles, making the algorithm highly suitable for real-time embedded applications.

INDEX TERMS DOA estimation, coherent sources, FPGA implementation, Toeplitz matrix, decorrelation, matrix decomposition-free.

I. INTRODUCTION

Direction-of-arrival (DOA) estimation plays a central role in array signal processing with broad applications across radar and electronic warfare, sonar and underwater acoustics, wireless communications (including beamforming for 5G/6G and massive MIMO), audio/speech capture, and navigation/localization systems [1], [2]. Accurate and low latency DOA estimates enable spatial filtering, interference suppression, target localization and tracking, and adaptive beamforming which are functions central to both civilian and defense sensor systems.

Classical subspace-based algorithms such as MUSIC, Root-MUSIC [3], and ESPRIT [4] provide high-resolution DOA estimates and serve as standard benchmarks [5]. They rely on covariance estimation and eigen/singular value decomposition (EVD/SVD) to separate signal and noise subspaces, offering statistical efficiency under ideal conditions [2]. However, the cubic complexity and high memory demand of EVD/SVD limit their scalability for large arrays and real-time systems. Moreover, coherent sources cause covariance rank deficiency, breaking subspace orthogonality unless preprocessing (e.g., spatial smoothing or forward–backward averaging) [6] is used, which reduces aperture and increases computational load.

To adrress these limitations, several approaches have been proposed that eliminate full EVD/SVD by exploiting algebraic or linear relations within the array manifold. Propagator and partial-propagator methods form subarray mappings to estimate DOAs without eigen analysis, offering major computational savings but reduced robustness to noise and mismatch [7], [8]. Yan's two-step Root-MUSIC

The associate editor coordinating the review of this manuscript and approving it for publication was Zaharias D. Zaharis.

(TS-Root-MUSIC) [9], [10] similarly uses subarray cross-correlations and polynomial rooting to lower complexity, trading some optimality for efficiency and real-time suitability.

Another important research direction exploits the Toeplitz or centro-Hermitian structure of ULA covariances [11], [12], [13], [14] to enhance estimation and reconstruct full covariances from partial or corrupted data. Low-rank Toeplitz reconstruction and completion via rank minimization or nuclear norm relaxation achieve high-resolution DOA and handle coherent sources without smoothing, but rely on iterative convex optimization that limits real-time use. Robust extensions, such as correntropy-based and fused-covariance formulations [13], [15] improve resistance to impulsive noise and outliers, albeit with added iteration and tuning overhead.

Sparse and coarray techniques, including nested and coprime arrays, extend the virtual aperture and degrees of freedom, enabling more sources to be resolved with fewer sensors. Combined with covariance reconstruction and sparse recovery [16], [17], they effectively support subspace processing on virtual arrays, though they remain sensitive to model mismatch and add computational cost in covariance interpolation.

A separate research approach leverages tensor decompositions [18] to exploit multiway structures in multi-snapshot or multifrequency data. Tensor methods, such as PARAFAC [19], can identify sources under relaxed conditions but involve iterative least-squares or higher-order SVD operations with high computational and memory demands, making them sensitive to initialization and less suited for real-time FPGA implementation [20].

More recently, machine learning (ML) and deep learning (DL) approaches [21], [22], [23] have been applied to DOA estimation, offering robustness to array imperfections, non-Gaussian noise, and low snapshot conditions. However, they often require large, well-matched datasets and are challenging to deploy on resource-constrained hardware. To address this, recent studies have developed lightweight and hardware-aware ML/DL architectures, including compact CNNs for array calibration and DOA estimation [24], temporal-fusion transformers with attention compression [25], and sparse or quantized models for efficient inference [26]. Hybrid and interpretable frameworks such as deep unfolding networks [27] and self-supervised contrastive learning models [28] further improve transparency and generalization. Collectively, these advances mark a clear trend toward efficient and deployable ML-based DOA estimators.

While most of the existing algorithms sufficed with numerical simulations for validation, a few have also looked at hardware implementations and real-time experiments. FPGA and SoC designs employing pipelined, fixed-point, and streaming architectures have accelerated unitary-MUSIC [29], [30], [43], [44], [45], [46], QR/LU/LDL/Cholesky pipelines [31], [32], [33], [34], [35], and CORDIC-based EVD approximations [36], [45], though at high resource cost. In contrast, decomposition-free or algebraic methods better align with DSP and streaming architectures, minimizing memory transfers and improving efficiency on platforms such as the NI PXIe-7993 FlexRIO [37], [38].

In summary, despite recent progress in fast subspace, sparse, and ML-based DOA algorithms, a fundamental trade-off remains between estimation robustness, computational efficiency, and hardware feasibility. Accelerated variants of MUSIC [43], [47], [48], orthogonal matching pursuit (OMP) extensions [49], and tensor decompositions [50] have reduced complexity, while lightweight and hybrid machine learning models [24], [25], [26], [27], [28] enhance data efficiency. Yet, achieving simultaneous coherence resilience, low arithmetic cost, numerical stability at low SNR, and real-time FPGA suitability remains an open challenge. This motivates the development of DOA algorithms that (a) handle coherent sources without spatial smoothing, (b) avoid full EVD/SVD, (c) maintain stability under moderate/low SNR and few snapshots, and (d) map efficiently to FPGA and real-time platforms.

Motivated by these challenges, the proposed work develops a decomposition-free DOA algorithm tailored for coherent sources. The proposed method (i) constructs a Toeplitz-structured matrix directly from ULA data, (ii) applies a unitary centro-Hermitian transform to form a real, symmetric compound matrix with reduced arithmetic, (iii) extracts low dimensional sub-blocks without eigen decomposition, and (iv) assembles a structured matrix to form a polynomial, with DOAs recovered via root finding. This pipeline blends Toeplitz/centro-Hermitian processing with the low complexity advantages of propagator and two-step rooting methods, making it well suited for FPGA/SoC implementation [30], [37].

The proposed method employs a Toeplitz-structured matrix instead of a covariance matrix to enhance resolution and estimation accuracy of coherent sources while reducing computational complexity, making the proposed algorithm highly suitable for hardware implementation. Unlike conventional spatial smoothing [6] or tensor decompositions [18], our method preserves the full array aperture and avoids the exponential complexity of higher order algebraic operations. To the best of our knowledge, this is among the first works to demonstrate a Toeplitz-driven subspace framework that jointly addresses coherence, complexity, and hardware feasibility.

The key contributions of this paper are summarized as follows:

1. Decomposition-free and covariance-free DOA estimation: The proposed algorithm eliminates the need for matrix decomposition (EVD/SVD/QR) as well as for computing the covariance matrix, thereby reducing complexity and enhancing scalability.
2. Partitioning-based polynomial rooting: The proposed algorithm partitions the transformed Toeplitz matrix into sub-blocks that naturally preserve rotational invariance, enabling direct polynomial formation and rooting for DOA estimation without decomposition.

3. Hardware friendly design: Due to its low arithmetic complexity and decomposition-free nature, the proposed method is particularly suited for FPGA implementation, making it attractive for real-time and resource constrained applications.
4. Support for coherent and non-coherent sources: The proposed method reliably estimates DOAs for both coherent and non-coherent signals with improved efficiency and scalability.

The proposed method is validated in MATLAB and LabVIEW FPGA simulations confirming its accuracy and computational advantages over classical and recent algorithms, and implemented on NI PXIe-7993 hardware. Compared with decomposition-based methods (such as [9], [14], [40]), the proposed algorithm offers lower complexity, robustness to coherence, and suitability for fixed-point FPGA implementation.

The remainder of this paper presents the system model and proposed algorithm, analyzes computational and implementation complexity, presents simulation results for coherent and uncorrelated scenarios, presents FPGA implementation details, calculates computation time for the proposed algorithm, and finally draws conclusions.

II. SYSTEM MODEL AND PROPOSED ALGORITHM

The system setup in Fig. 1 employs a uniform linear array (ULA) consisting of $2N + 1$ antenna elements. The array is arranged such that the central element, indexed as 0, acts as the reference point, while the remaining $2N$ elements are symmetrically distributed to its left and right. The inter-element spacing d is fixed at half the wavelength of the impinging signals to avoid spatial aliasing and ensure optimal sampling of the wavefront. Furthermore, the incoming signals are assumed to arrive from the far-field, guaranteeing planar wavefronts across the array aperture and thereby justifying the use of the narrowband array model.

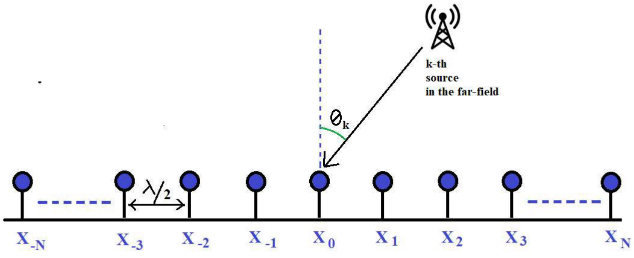


FIGURE 1. Receiver setup employing a ULA with $2N + 1$ elements.

Assume K narrowband source signals arrive at the array from directions $\theta_1\theta_2, , \ldots \theta_K)$, where $K < 2N + 1$. All sources are assumed to share the same carrier frequency. The received signal at the n-th element of the ULA for a snapshot at time t can then be expressed as:

$$x_n(t) = \sum_{i=1}^{K} e^{-j\left(\frac{2\pi}{\lambda}(n)\right) d\cos\theta_i} s_i(t) + n_n(t) \tag{1}$$

where

$s_i(t)$ = represents the i^{th} incoming source signal received by the ULA,

$\mathrm{n_n(t)}$ = noise term modeled as AWGN at the $\mathrm{n^{th}}$ array element of the ULA,

$\mathrm{x_n(t)}$ = signal observed at the n^{th} array element at time t.

With the central array element designated as the reference, the collective output of the $(2N + 1)$ antenna elements positioned along the linear axis can be expressed in vector form, representing the received data across the array, as:

$$\hat{\boldsymbol{X}}(t) = \begin{bmatrix} X_{-N}(t) \\ \vdots \\ X_0(t) \\ \vdots \\ X_N(t) \end{bmatrix} = \boldsymbol{A}(\theta)\, s(t) + n(t) \tag{2}$$

where $\boldsymbol{X}(t)$ is of dimension $(2N+1)$ x 1, and

$$\boldsymbol{A}(\theta) = \left[a(\theta_1)\; a(\theta_2)\; \ldots\; a(\theta_K) \right] \tag{3}$$

is the array manifold matrix of soze $(2N+ 1) \times K$, where

$$a(\theta_k) = \left[(u_k^*)^N \;\cdots\; 1 \;\cdots\; u_k^N \right]^T \tag{4}$$

is the corresponding array steering vector of size $(2N+1) \times 1$, where

$$u_k = e^{-j\left(\frac{2\pi}{.\lambda}\right) d\cos(\theta_k)} \tag{5}$$

s(t) is the vector of K received signals

$$s(t) = [s_1(t)\; s_2(t)\; \cdots\; s_K(t)]^{\,T} \tag{6}$$

and,

$$\mathrm{n}(t) = \left[\left(n_{-N}(t) \cdots\; n_0(t) \;\cdots n_N(t) \;\right)\right]^T \tag{7}$$

where $\mathrm{n}(t)$ represents the noise vector of dimension $(2N + 1)\times 1$. Note: $(\ )^{\mathrm{H}}$ denotes the Hermitian operation, $(\ )^{\mathrm{T}}$ denotes the transpose, and $()^*$ denotes the complex conjugate operation.

In the proposed approach, the received signal vectors $\hat{\boldsymbol{X}}(t)$ are first averaged over L snapshots, as:

$$y(t) = \frac{1}{L}\sum_{i=1}^{L} \hat{X}_i(t) = \begin{bmatrix} y_{-N}(t) \\ \vdots \\ y_0(t) \\ \vdots \\ y_N(t) \end{bmatrix} \tag{8}$$

In the next step, the Toeplitz-structured data matrix $\boldsymbol{Y}$ of size $(N + 1) \times (N + 1)$ is constructed from the sample-averaged data vector y(t) as:

$$\boldsymbol{Y} = \begin{bmatrix} y_0 & y_{-1} & & y_{-N} \\ y_1 & y_0 & \cdots & y_{-(N-1)} \\ y_2 & y_1 & & y_{-(N-2)} \\ \vdots & \vdots & \ddots & \vdots \\ y_N & y_{N-1} & \cdots & y_0 \end{bmatrix}$$

$$= \begin{bmatrix} s_0 & s_{-1} & & s_{-N} \\ s_1 & s_0 & \cdots & s_{-(N-1)} \\ s_2 & s_1 & & s_{-(N-2)} \\ \vdots & \vdots & \ddots & \vdots \\ s_N & s_{N-1} & \cdots & s_0 \end{bmatrix} + \begin{bmatrix} n_0 & n_{-1} & & n_{-N} \\ n_1 & n_0 & \cdots & n_{-(N-1)} \\ n_2 & n_1 & & n_{-(N-2)} \\ \vdots & \vdots & \ddots & \vdots \\ n_N & n_{N-1} & \cdots & n_0 \end{bmatrix} \tag{9}$$

where s_i denotes the source signal component received at the i-th element of the ULA, and n_i represents the corresponding noise component.

For simplicity, in the absence of noise, the Toeplitz matrix $\boldsymbol{Y}$ can be expressed using the Vandermonde decomposition theorem [41] as:

$$\boldsymbol{Y} = \tilde{\boldsymbol{A}}\boldsymbol{S}_p\tilde{\boldsymbol{A}}^H$$

While under noisy conditions, we obtain:

$$\boldsymbol{Y} = \boldsymbol{Y}_s(t) + \boldsymbol{Y}_n(t) = \tilde{\boldsymbol{A}}\boldsymbol{S}_p\tilde{\boldsymbol{A}}^H + \tilde{\boldsymbol{A}}\boldsymbol{N}_p\tilde{\boldsymbol{A}}^H \tag{10}$$

where $\tilde{A} = [\tilde{a}(\theta_1)\ \tilde{a}(\theta_2)\ \cdots\ \tilde{a}(\theta_K)]$ is the $(N+1)$ x K Vandermonde array steering matrix with $\tilde{a}(\theta_k) = \left[1e^{-j\left(\frac{2\pi}{.\lambda}\right)dcos(\theta_k)} \ldots e^{-j\left(\frac{2\pi}{.\lambda}\right)Ndcos(\theta_k)}\right]^T$, $\boldsymbol{S}_p = \text{diag}[s_1, \ldots, s_K]$ is a denotes a $(K \times K)$ diagonal matrix with strictly positive entries. where $s_{\text{k}} > 0$ ($\text{k} = 1, \ldots, K$) represent the incident signal sources, and $\boldsymbol{N}_p = \text{diag}[n_1,\ldots, n_K]$ denotes the diagonal noise matrix.

The Hermitian Toeplitz matrix $\boldsymbol{Y}$ inherits its rank from the Vandermonde steering matrix $\tilde{\boldsymbol{A}}$, which is limited to the number of sources K. Thus, the rank of $\boldsymbol{Y}$ equals the number of incident sources, ensuring that DOAs can be accurately estimated even for coherent sources. In practice, noise disrupts the conjugate symmetry of $\boldsymbol{Y}$, as off-diagonal noise elements are not exact conjugates. For instance, the noise elements n_{-2} and n_2 in (9) are not equal ($n_{-2} \neq n_2{}^*$). To address this, we add $\boldsymbol{Y}$ to $\boldsymbol{Y^H}$, which preserves the Toeplitz structure while enabling a real-valued transformation of the matrix.

$$\hat{\boldsymbol{Y}} = (\boldsymbol{Y} + \boldsymbol{Y^H}) = (\boldsymbol{Y} + \boldsymbol{JY}^*\boldsymbol{J}) \tag{11}$$

where $\boldsymbol{J}$ is an $(N+1) \times (N+1)$ matrix whose off-diagonal entries are ones and rest of the elements are zeros.

$$\begin{aligned} \text{Also, } \hat{\boldsymbol{Y}} &= \boldsymbol{Y} + \boldsymbol{Y}^H = \tilde{\boldsymbol{A}}\boldsymbol{S}_p\tilde{\boldsymbol{A}}^H + (\tilde{\boldsymbol{A}}\boldsymbol{S}_p\tilde{\boldsymbol{A}}^H)^H \\ &= \tilde{\boldsymbol{A}}\boldsymbol{S}_p\tilde{\boldsymbol{A}}^H + \tilde{\boldsymbol{A}}\boldsymbol{S}_p^H\tilde{\boldsymbol{A}}^H \\ &= \tilde{\boldsymbol{A}}(\boldsymbol{S}_p\tilde{\boldsymbol{A}}^H + \boldsymbol{S}_p^H\tilde{\boldsymbol{A}}^H) = \tilde{\boldsymbol{A}}(\boldsymbol{S_p} + \boldsymbol{S_p^H})\tilde{\boldsymbol{A}}^H \\ \therefore \hat{\boldsymbol{Y}} &= \boldsymbol{Y} + \boldsymbol{Y}^H = \tilde{\boldsymbol{A}}(2\boldsymbol{Re}(\boldsymbol{S}_p))\tilde{\boldsymbol{A}}^H \end{aligned}$$

It should be mentioned here that the conjugate smoothing operation in (11) is effectively equivalent to applying forward–backward averaging to the data matrix $\boldsymbol{Y}$, yielding $\hat{\boldsymbol{Y}}$. This process enhances the decorrelation of strongly correlated or coherent signals, thereby improving the accuracy of DOA estimation in challenging multipath environments.

Next, $\hat{\boldsymbol{Y}}$ in (11) which is a complex-valued matrix is converted to a real-valued matrix as follows:

$$\hat{\boldsymbol{Y}}_{\boldsymbol{r}} = \text{real}\left(\hat{\boldsymbol{Y}}\right) + \text{imag}\left(\hat{\boldsymbol{Y}}\right) * \boldsymbol{J} \tag{12}$$

where $\boldsymbol{J} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & \vdots & \cdots & \vdots & 0 \\ \vdots & 1 & \ddots & 0 & \vdots \\ 1 & 0 & \cdots & 0 & 0 \end{bmatrix}$ is of size $(N+1) \times (N+1)$.

The proposed strategy of converting complex-valued matrices into equivalent real-valued representations offers notable efficiency advantages over conventional unitary transformation approaches. Specifically, this conversion reduces computation time by approximately a factor of four and decreases memory requirements by half. Furthermore, in the subsequent stages of the DOA estimation algorithm, the use of real-valued matrices further lowers the computational burden by at least fourfold, thereby enhancing the overall efficiency of the proposed method.

In the next step, the real-valued matrix $\hat{\boldsymbol{Y}}_{\boldsymbol{r}}$ is partitioned into structured sub-blocks as:

$$\widehat{\mathbf{Y}}_{\mathbf{r}} = \begin{bmatrix} - & \mathbf{R}_{12} & \mathbf{R}_{13} \\ \mathbf{R}_{21} & - & \mathbf{R}_{23} \\ \mathbf{R}_{31} & \mathbf{R}_{32} & - \end{bmatrix} \tag{13}$$

where $\mathbf{R}_{12}$, $\mathbf{R}_{13}$, $\mathbf{R}_{21}$, etc., are submatrices of $\widehat{\mathbf{Y}}_{\mathbf{r}}$ defined as:

$$\begin{aligned} \mathbf{R}_{21} &= \widehat{\mathbf{Y}}_{\text{r}}(K+1:2K,\ 1:K), \\ \mathbf{R}_{32} &= \widehat{\mathbf{Y}}_{\text{r}}(2K+1:\text{end},\ K+1:2K), \\ \mathbf{R}_{12} &= \widehat{\mathbf{Y}}_{\text{r}}(1:K,\ K+1:2K), \\ \mathbf{R}_{31} &= \widehat{\mathbf{Y}}_{\text{r}}(2K+1:\text{end},\ 1:K) \end{aligned}$$

where K is the number of sources impinging on the ULA.

Partitioning the real-valued matrix $\hat{\boldsymbol{Y}}_{\boldsymbol{r}}$ into sub-blocks provides a structured way to capture cross-relations between array elements, preserving the rotational invariance needed for DOA estimation without relying on eigenvalue or singular value decomposition. These sub-blocks encode the phase shifts introduced by different source directions, enabling polynomial-based DOA estimation where the polynomial roots correspond directly to the DOAs. This avoids costly decompositions while maintaining estimator accuracy.

In addition, working with smaller structured submatrices improves computational efficiency and hardware suitability. The reduced dimensionality accelerates processing and maps naturally onto FPGA pipelines for matrix operations and polynomial formation, lowering latency and resource usage. By avoiding large scale matrix factorizations, the approach also enhances numerical stability in noisy or low SNR scenarios, making partitioning a practical and robust foundation for decomposition-free DOA algorithms.

Next, define the steering matrix as:

$$\mathbf{A}(\theta) = \begin{bmatrix} \mathbf{A}_1(\theta) \\ \mathbf{A}_2(\theta) \\ \mathbf{A}_3(\theta) \end{bmatrix} \tag{14}$$

where $\mathbf{A}_1(\theta) \in \mathbb{C}^{k\times k}$, $\mathbf{A}_2(\theta) \in \mathbb{C}^{k\times k}$, and $\mathbf{A}_3(\theta) \in \mathbb{C}^{(N-2k)\times k}$

$$\begin{aligned}
\text{Define } \mathbf{R}_{ij} &= \mathbf{A}_i(\theta)\mathbf{S}_p\mathbf{A}_j^H(\theta) \\
\text{Let } \boldsymbol{\psi}_1 &= \mathbf{R}_{32}\mathbf{R}_{12}^{-H} \qquad (15)\\
\boldsymbol{\psi}_1 &= [\mathbf{A}_3(\theta)\mathbf{S}_p\mathbf{A}_2^H(\theta)][\mathbf{A}_1(\theta)\mathbf{S}_p\mathbf{A}_2^H(\theta)]^{-H} \\
&= \mathbf{A}_3(\theta)\mathbf{S}_p\mathbf{A}_2^H(\theta)\mathbf{A}_2^{-1}(\theta)\mathbf{S}_p^{-H}\mathbf{A}_1(\theta)^{-H} \\
&= \mathbf{A}_3(\theta)\mathbf{A}_1^{-1}(\theta) \qquad (16)
\end{aligned}$$

$$\text{Similarly, define } \boldsymbol{\psi}_2 = \mathbf{R}_{31}\mathbf{R}_{21}^{-H} = \mathbf{A}_3(\theta)\mathbf{A}_2^{-1}(\theta) \tag{17}$$

The block-partition plus the $\boldsymbol{\psi}_i$ construction yields a matrix $\mathbf{W}$ whose nullspace captures the signal structure (so the phase factors can be recovered by standard root methods applied to a derived matrix).

Define the block matrix $\mathbf{W}$, as:

$$\begin{aligned}
\mathbf{W} &= \begin{bmatrix} \boldsymbol{\psi}_1^H & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\psi}_2^H \\ -\mathbf{I} & -\mathbf{I} \end{bmatrix} \\
&= \begin{bmatrix} (\mathbf{A}_3(\theta)\mathbf{A}_1^{-1}(\theta))^H & \mathbf{0} \\ \mathbf{0} & (\mathbf{A}_3(\theta)\mathbf{A}_2^{-1}(\theta))^H \\ -\mathbf{I} & -\mathbf{I} \end{bmatrix} \\
\mathbf{W}^H\mathbf{A} &= \begin{bmatrix} \mathbf{A}_3(\theta)\mathbf{A}_1^{-1}(\theta) & \mathbf{0} & -\mathbf{I} \\ \mathbf{0} & \mathbf{A}_3(\theta)\mathbf{A}_2^{-1}(\theta) & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{A}_1(\theta) \\ \mathbf{A}_2(\theta) \\ \mathbf{A}_3(\theta) \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{A}_3(\theta)\mathbf{A}_1^{-1}(\theta)\mathbf{A}_1(\theta) - \mathbf{A}_3(\theta) \\ \mathbf{A}_3(\theta)\mathbf{A}_2^{-1}(\theta)\mathbf{A}_2(\theta) - \mathbf{A}_3(\theta) \end{bmatrix} = \mathbf{0} \qquad (18)
\end{aligned}$$

Hence, it is shown that the source subspace represented by the steering matrix $\mathbf{A}(\theta)$ lies in the null space of $\mathbf{W}$.

Next, compute polynomial $\boldsymbol{G}_n$

$$= \boldsymbol{U}^H\boldsymbol{W}\boldsymbol{W}^H\boldsymbol{U} \tag{19}$$

$$\begin{aligned}
\text{where, } \mathbf{W}\mathbf{W}^H &= \begin{bmatrix} \boldsymbol{\psi}_1^H & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\psi}_2^H \\ -\mathbf{I} & -\mathbf{I} \end{bmatrix} \\
&\times \begin{bmatrix} \boldsymbol{\psi}_1^H & \mathbf{0} & -\mathbf{I} \\ \mathbf{0} & \boldsymbol{\psi}_2^H & -\mathbf{I} \end{bmatrix} \\
&= \begin{bmatrix} \boldsymbol{\psi}_1^H\boldsymbol{\psi}_1^H & \mathbf{0} & -\boldsymbol{\psi}_1^H \\ \mathbf{0} & \boldsymbol{\psi}_2^H\boldsymbol{\psi}_2^H & -\boldsymbol{\psi}_2^H \\ -\boldsymbol{\psi}_1^H & -\boldsymbol{\psi}_2^H & 2\mathbf{I} \end{bmatrix} \qquad (20)
\end{aligned}$$

and $\boldsymbol{U}$ is a $(N+1)\times(N+1)$ unitary matrix given as:

$$\boldsymbol{U} = \boldsymbol{I}j + \boldsymbol{J} = \begin{bmatrix} j & 0 & 0 & 0 & 1 \\ 0 & j & \cdots & 1 & 0 \\ 0 & \vdots & \cdots & \vdots & 0 \\ \vdots & 1 & \ddots & j & \vdots \\ 1 & 0 & \cdots & 0 & j \end{bmatrix}$$

where $\boldsymbol{I}$ is identity matrix and $\boldsymbol{J}$ is as shown in (12).

Note that all required operations for $\boldsymbol{\psi}_i$ and $\mathbf{W}$ are multiplications, additions and small inverses which map well to fixed-point, pipelined FPGA logic. Root-finding can be relegated to a host CPU where high precision arithmetic is available.

The polynomial roots w_k of **Gn** in (19) capture the spatial frequency characteristics of the incoming signals. Like the Root-MUSIC algorithm, the DOA estimates are obtained by identifying the roots nearest to the unit circle and mapping their arguments to angular values using the array steering relationship.

$$\theta_k = arccos\left(\frac{\lambda angle\,(w_k)}{2\pi d}\right) \tag{21}$$

Now, the proposed algorithm can be summarized as follows:

1: Form the Toeplitz structured data matrix $\boldsymbol{Y}$ from the data vector x(t) averaged over L samples as in (8) and (9)

2: Estimate $\hat{\boldsymbol{Y}}$ as given in (11)

3: Transform the complex matrix $\hat{\boldsymbol{Y}}$ to a real matrix $\hat{\boldsymbol{Y}}_{\boldsymbol{r}}$ as given by (12)

4: Divide $\hat{\boldsymbol{Y}}_{\boldsymbol{r}}$ into submatrices as expressed in (13)

5: Construct the matrix $\mathbf{W}$ according to (18) with $\boldsymbol{\psi}_1$ and $\boldsymbol{\psi}_2$ computed as in (16) and (17), respectively

6: Generate the polynomial **Gn** as in (19)

7: Compute the roots of **Gn** through polynomial rooting techniques

8: Estimate the DOAs using the formulation in (21)

III. ANALYSIS OF COMPUTATIONAL COST

This section analyses the computational cost of the proposed method (PM) and compares it with that of representative methods [9], [14], [40] from the literature. The aim is to demonstrate the efficiency of the PM in reducing the computational burden of DOA estimation. The analysis is carried out for $2N+1$ antenna elements, L snapshots, and K signal sources, which serve as the key parameters in formulating the cost expressions. Table 1 summarizes the computational requirements of the principal processing steps for the proposed method.

For a quantitative assessment of computational complexity, the proposed method is compared with the two-step Root-MUSIC algorithm (TS_DOA) in [9], the single snapshot coherent source estimator (SS_DOA) in [14], and the enhanced Toeplitz–Khatri–Rao approach (TK_DOA) in [40].

TABLE 1. Computational cost for proposed method.

Algorithm Step	Computation Cost
Averaging the data vector over L samples	$(2N+1)(L-1)$
Toeplitz structure preservation	$(N+1)^2$
Unitary transformation	$2(N+1)^2$
Computing W	$2(N-2K)K^2 + 2K^3$
Matrix multiplication	$(N-2K)[2K(N-2K) + 2K + 1] + 4N^2$
Compute Polynomial Roots	$2N(N+1)^2$
Compute DOAs	$2N$

The computational load is assessed by calculating the total number of arithmetic operations for $K = 3$ incident sources and $L = 10$ snapshots, while varying the number of antenna elements N from 8 to 256. The corresponding results are summarized in Table 2. It should be noted that the actual number of antenna elements is given by $2N + 1$. The computational cost expressions for SS_DOA, TS_DOA and TK_DOA are given below:

$$\text{SS\_DOA\_cost} = (2N+1)(L-1) + (N+1)^2 + 2(N+1)^2 + 2(N+1)^3/3 + N^2(N-K) + 4N^2 + 2N(N+1)^2 + 2N$$

$$\text{TS\_DOA\_cost (step 2)} = LN^2 + 32K^3 + 40K^2(N-2K) + K^3 + 16K^2(N-2K) + 8N(N-2K)^2 + 8(N-K)^2(N-2K) + K^3 + (2N-1)^3$$

$$\text{TK\_DOA\_cost} = (2M+1)^2 L + (2M+1)^2(M+1)^2 + (2M+1)^3 + 3(2M+1)K^2 + 2K^3$$

TABLE 2. Computational cost with varying N.

N	PM	SS [14]	%PM/SS	TK [40]	%PM/TK	TS [9]	%PM/TS
8	2092	3166	66.08	31725	6.59	13194	15.86
16	12372	19499	63.45	362493	3.41	129258	9.57
24	37116	60281	61.57	1643661	2.26	490570	7.57
32	82468	136774	60.30	4919709	1.68	1244586	6.63
48	259572	441953	58.73	23600445	1.10	4520554	5.74
64	592836	1025147	57.83	72624861	0.82	11136810	5.32
128	4470532	7940326	56.30	1116763485	0.40	94597674	4.73
256	34663812	62509499	55.45	17519700573	0.20	780466218	4.44

Table 2 clearly demonstrates the efficiency of the proposed method, requiring substantially fewer operations compared to the existing approaches SS_DOA, TK_DOA, and TS_DOA. The proposed method (PM) achieves substantial computational savings compared with recent decomposition-free and subspace-based DOA estimators. When benchmarked against SS_DOA, PM consistently requires fewer arithmetic operations, consuming only 55–66% of SS_DOA's cost across all tested values of N. Importantly, the relative advantage grows with array size: at $N = 8$, PM costs 66.1% of SS_DOA, while at $N = 256$, it drops to 55.5%. This indicates that PM scales more efficiently, making it better suited for large ULAs and real-time scenarios where computational resources are limited.

Compared with TK_DOA, the efficiency difference is dramatic. Although TK_DOA excels in resolving coherent sources and extending virtual aperture, its reliance on covariance reconstruction, vectorization, and subspace processing leads to prohibitive computational growth. At $N = 8$, TK_DOA already requires ~31,725 operations (15× more than PM), and at $N = 256$, the complexity exceeds 17.5 billion operations, compared to only ~34 million for PM. Consequently, PM achieves similar goals with just 0.2–6.6% of TK_DOA's cost, highlighting its suitability for FPGA or embedded deployment where TK_DOA would be impractical.

Relative to TS_DOA, PM again demonstrates superior efficiency. TS_DOA avoids explicit EVD/SVD but still incurs heavy costs from polynomial rooting and intermediate transformations. The results show that PM consumes only 4.4–15.86% of TS_DOA's complexity, with the margin widening at larger array sizes (from 15.86% at $N = 8$ to under 4.4% at $N = 256$). This confirms that PM not only maintains decomposition-free operation but also scales more favorably than existing propagator-type alternatives.

Overall, PM grows with linear-to-quadratic complexity in N, while SS_DOA, TK_DOA, and TS_DOA exhibit much steeper increases. This consistent efficiency across small and large arrays underscores PM's practical advantage for real-time DOA estimation, FPGA acceleration, and embedded low power platforms. Reduced computational demand also translates into lower latency and memory requirements, which are critical in high snapshot or multisource environments. In summary, PM provides a balanced solution: it is lighter than TK_DOA, more scalable than SS_DOA, and more efficient than TS_DOA, making it an attractive candidate for practical high performance DOA estimation systems.

IV. NUMERICAL SIMULATIONS AND PERFORMANCE ANALYSIS

This section presents the outcomes of extensive MATLAB simulations conducted to assess the effectiveness of the proposed method (PM) and validate its estimation accuracy across different scenarios. The performance is evaluated in terms of root mean square error (RMSE) under varying conditions, including changes in signal-to-noise ratio (SNR), number of snapshots, and array size. Simulations are conducted for up to four sources to be estimated, considering a mix of both coherent and non-coherent signals, with snapshots ranging from 1 to 500 and SNR values from 0 dB to 30 dB. The array under study is a 25-element ULA (N = 12), and AWGN noise model is assumed in generating the received data. Each configuration is simulated over 100 Monte Carlo runs to ensure statistical reliability. Performance of the proposed method is compared with SS_DOA [14], TS_DOA [9], and TK_DOA [40] and benchmarked against the CRB (Cramer-Rao bound).

A. VERIFICATION OF PROPOSED ALGORITHM

Fig. 2 and Fig. 3 show the graphs for RMSE values vs SNR for two coherent sources located at 40° and 60°, for the case of 5 and 100 snapshots, respectively. It is clear from the graphs that PM performs well even at low snapshot counts and has superior performance when sufficient number of snapshots are available.

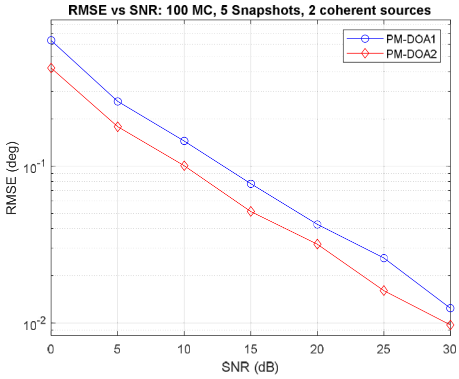


FIGURE 2. RMSE vs SNR: two coherent sources estimated with 5 snapshots.

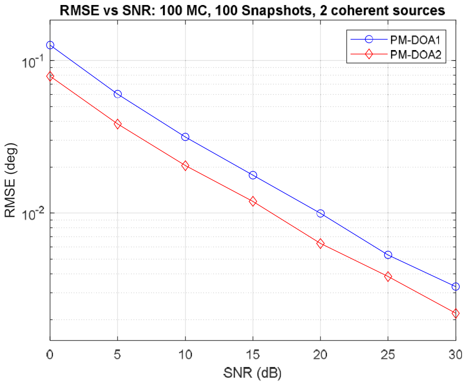


FIGURE 3. RMSE vs SNR: two coherent sources estimated with 100 snapshots.

Fig. 4 and Fig. 5 present the RMSE versus SNR performance for DOA estimation of four sources positioned at 10°, 30°, 60°, and 80°, using 200 snapshots. Fig. 4 demonstrates the effectiveness of the proposed method when estimating a combination of coherent and non-coherent sources, while Fig. 5 illustrates its performance when all four sources are coherent.

The above graphs demonstrate the robustness and versatility of the proposed method in estimating multiple non-coherent and coherent signals with high accuracy.

The performance of the proposed method (PM) is further examined under different numbers of snapshots and antenna elements. Fig. 6 presents the DOA estimation results for two coherent sources as the number of snapshots varies from 1 to 50 and until 500 at an SNR of 10 dB. The results demonstrate that the proposed method achieves reliable estimation accuracy even under low snapshot conditions, while its performance progressively improves with larger snapshot counts, highlighting its scalability and effectiveness in practical scenarios.

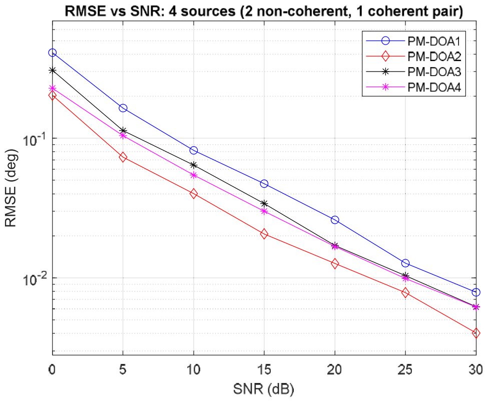


FIGURE 4. RMSE vs SNR: four sources (2 non-coherent and 1 pair of coherent signals) estimated with 200 snapshots.

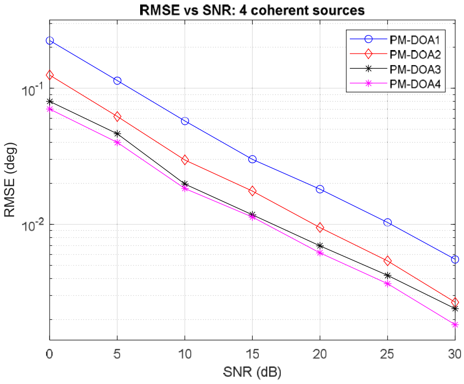


FIGURE 5. RMSE vs SNR: four coherent sources estimated with 200 snapshots.

Fig. 7 shows the DOA performance of PM in the estimation of two coherent sources against varying number of antenna elements ($N = 5{:}4{:}64$) at 10 dB SNR with a single snapshot. It can be noted that even with a single snapshot, PM performs well and its performance improves with increasing number of antenna elements (as expected).

Fig. 8 shows the DOA performance of PM in the estimation of two coherent sources with 10 snapshots. It can be noted that there is a marked improvement in performance.

The simulation results shown in Fig. 6, Fig. 7, and Fig. 8 demonstrate that the method is capable of effectively resolving coherent signals even under limited data conditions. This underscores the robustness of PM in scenarios where snapshot availability is constrained, which is often the case in real-time or resource limited applications.

The histogram in Fig. 9 illustrates the estimation accuracy of PM for four coherent sources positioned at 10°, 30°, 60°, and 80° relative to the ULA. The DOA estimates are obtained

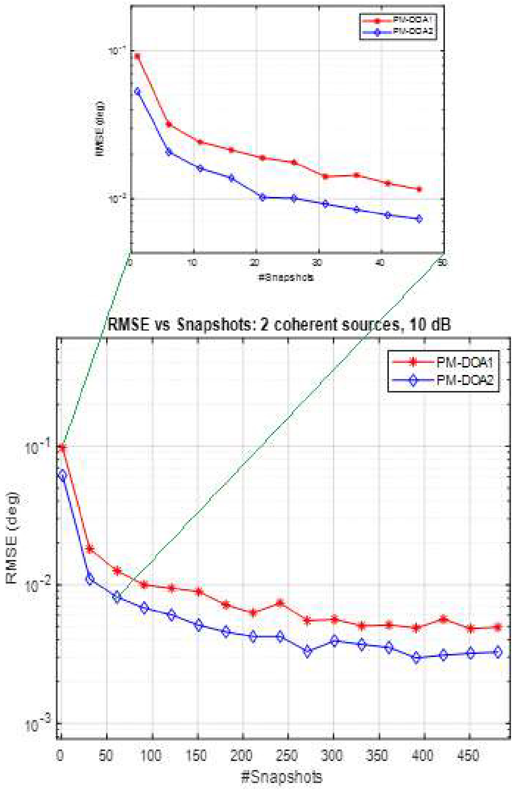


FIGURE 6. RMSE vs snapshots: two coherent sources (located at 40° and 60°) estimated with SNR = 10 dB.

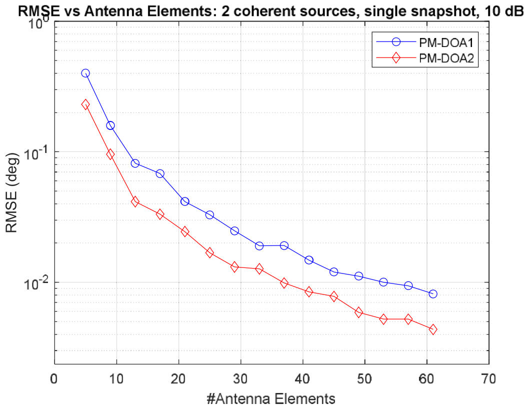


FIGURE 7. RMSE vs antenna elements: two coherent sources (located at 40° and 60°) estimated with a single snapshot at SNR = 10 dB.

using 200 snapshots with 25 antenna elements at an SNR of 20 dB.

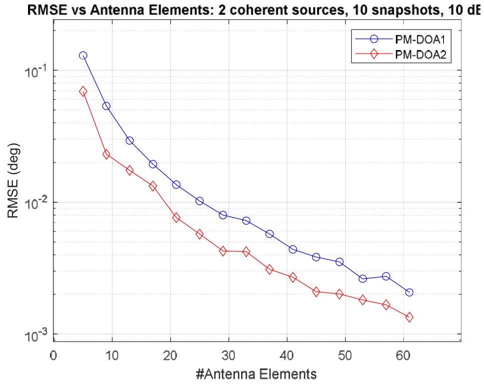


FIGURE 8. RMSE vs antenna elements: two coherent sources (located at 40° and 60°) estimated with 10 snapshots at SNR = 10 dB.

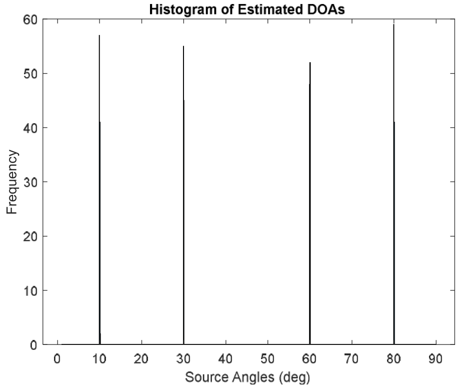


FIGURE 9. Histogram of DOA estimation using the proposed method for four coherent sources located at 10°, 30°, 60°, and 80° with 200 snapshots.

Next, the angular resolution of the proposed decomposition-free DOA estimation algorithm was evaluated using four test scenarios for the case of three coherent sources separated by 2°, 3°, 5°, and 10°, respectively. These estimates are obtained using 29 antenna elements with 200 snapshots at an SNR of 15 dB. The corresponding histograms, shown in Fig. 10, clearly illustrate that the proposed method accurately resolves all three sources even under closely spaced conditions. For a larger number of sources, high estimation accuracy and fine angular resolution can be maintained by varying the SNR, number of antenna elements, and/or snapshots, appropriately.

Finally, the robustness of the proposed algorithm under challenging SNR regimes (below 0 dB) is evaluated. Fig. 11 illustrates the RMSE vs SNR performance for three fully coherent sources positioned 10° apart at 20°, 30°, and 40°, respectively. Curves are shown for −5 to +15 dB to emphasize the target operating regime. All three coherent sources are fully resolvable at −5 dB and the algorithm maintains sub-degree accuracy even at −5 dB, with RMSE values in the range of 0.2° to 0.6°, demonstrating strong robustness in practical low SNR environments. Performance can be further improved with increasing number of antenna elements and/or snapshots.

B. PERFORMANCE COMPARISON

Fig. 12 shows the average RMSE vs SNR graphs for estimating two coherent sources located at 40° and 60° for the proposed method (PM_DOA) and its performance is compared with Root-MUSIC (RM_DOA), SS_DOA [14],

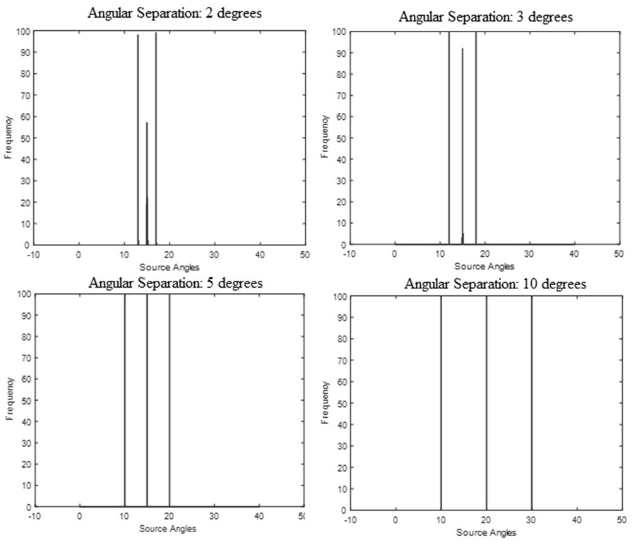


FIGURE 10. Histogram of DOA estimation using the proposed method for three coherent sources separated by 2°, 3°, 5°, and 10°, respectively.

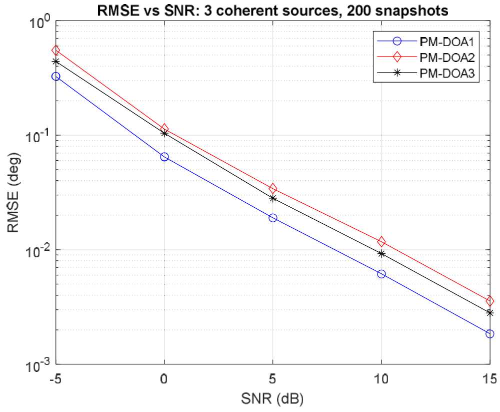


FIGURE 11. RMSE vs SNR: three coherent sources estimated with a 25-element ULA located at 20°, 30°, and 40°, with 200 snapshots.

TS_DOA [9] and benchmarked against the CRB. Simulations were performed with 25 antenna elements, 200 snapshots, and 100 Monte Carlo runs. The proposed method outperforms RM_DOA and SS_DOA, only TS_DOA has better performance but which comes at a much greater cost as analyzed above in section III.

Fig. 13 shows the average RMSE curves for three coherent sources located at 10°, 30°, and 60° for the proposed method and its performance is compared with RM_DOA, SS_DOA [14], EE_DOA [42] and benchmarked against the CRB. Simulation conditions remain the same as in the case above. PM outperforms all the other methods, while Root-MUSIC suffers from estimation failure.

Fig. 14 below shows the average RMSE curves for estimation of four sources with a combination of non-coherent and coherent signals. A pair of coherent signals are located at (10°, 30°), while the two non-coherent sources are located at 60° and 80° from the ULA. A performance comparison is made with RM_DOA, SS_DOA [14], TK_DOA [40] and benchmarked against the CRB. Simulation conditions remain the same as in the cases above. The proposed algorithm is clearly superior to all the other methods, while TK_DOA shows poor estimation accuracy.

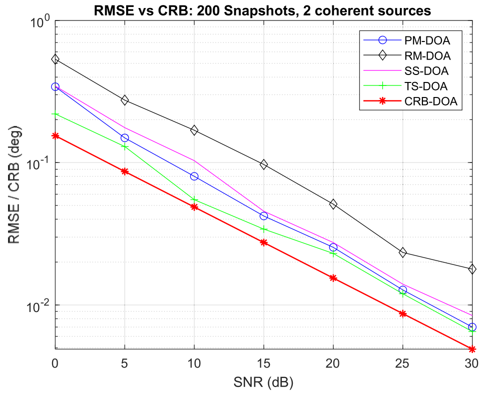


FIGURE 12. Performance comparison: average RMSE values for DOA estimation of two coherent sources located at 40° and 60° with 200 snapshots.

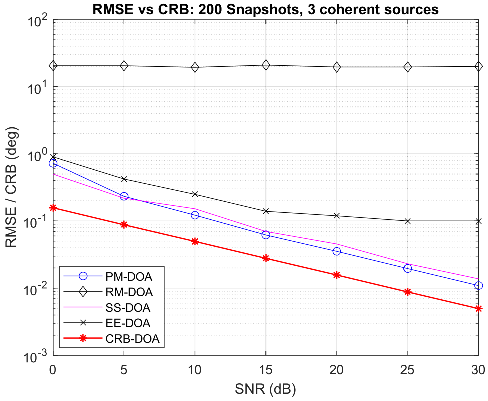


FIGURE 13. Performance comparison: average RMSE values for DOA estimation of three coherent sources located at 10°, 30°, and 60° with 200 snapshots.

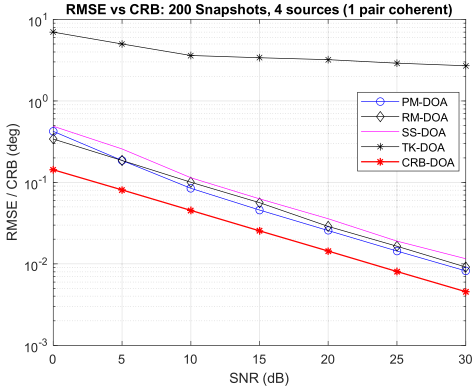


FIGURE 14. Performance comparison: average RMSE values for DOA estimation of two non-coherent (60°, 80°) and one pair of coherent sources(10°, 30°) with 200 snapshots.

In the next graph (Fig. 15), the average RMSE curves for two pairs of coherent signals are shown. The first pair of coherent signals is located at (10°, 30°), and the other one at (60°, 80°) from the ULA. The performance of PM is compared with RM_DOA, SS_DOA [14], TK_DOA [40] and benchmarked against the CRB. Simulation conditions remain the same as in the cases above. PM outperforms all the other methods, while TK_DOA shows poor estimation accuracy.

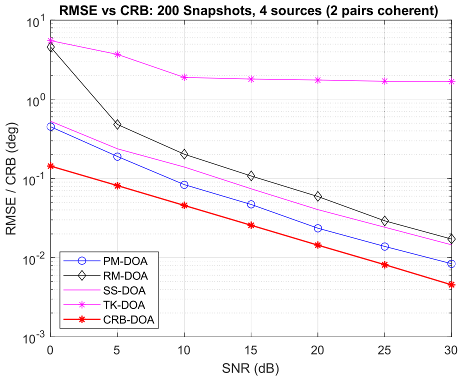


FIGURE 15. Performance comparison: average RMSE values for estimating two pairs of coherent sources (10°, 30°) and (60°, 80°) with 200 snapshots.

Finally, we examine the case of four coherent sources in Fig. 16. The four coherent sources are located at 10°, 30°, 60°, and 80° from the ULA. PM outperforms all the other methods, while Root-MUSIC suffers from estimation failure.

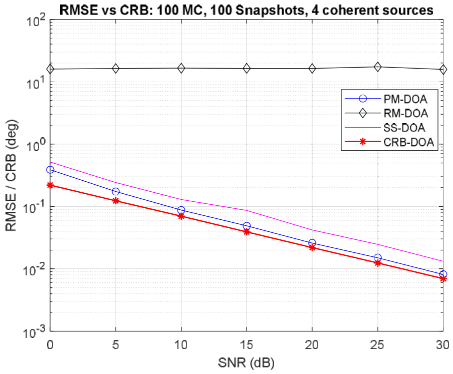


FIGURE 16. Performance comparison: average RMSE values for estimating four coherent sources located at 10°, 30°, 60°, and 80° with 200 snapshots.

The above graphs establish the effectiveness of the proposed method in estimating the DOAs of multiple coherent and non-coherent signals and also show its superior performance when compared with the existing methods ([9], [14], [40], [42]) over a range of simulation scenarios.

V. FPGA IMPLEMENTATION OF PROPOSED ALGORITHM

To demonstrate the suitability of the proposed algorithm for an efficient and low cost hardware implementation, it has been implemented on the PXIe-7993 [38] which is a high-performance FlexRIO FPGA module housing Xilinx Kintex UltraScale KU11P FPGA. The implementation was realized using high throughput LabVIEW FPGA modules (with LabVIEW version 21.0.1f1 (64-bit)) for arithmetic operations for estimating up to two incident sources using a ULA with nine antenna elements ($N = 4$) resulting in matrices of size 5×5.

The FPGA implementation presented in this section is intended to validate the hardware friendliness, resource efficiency, and real-time throughput potential of the proposed decomposition-free algorithm. DOA estimation accuracy was also verified on the FPGA using LabVIEW FPGA simulation mode, as shown in Fig. 25.

A. HYBRID IMPLEMENTATION

A hybrid approach was taken for the implementation with Steps 1 through 6 of the proposed algorithm implemented on the FPGA and the final two steps of computing the DOAs from polynomial rooting implemented on the Host Processor. This model is shown in Fig. 17 below:

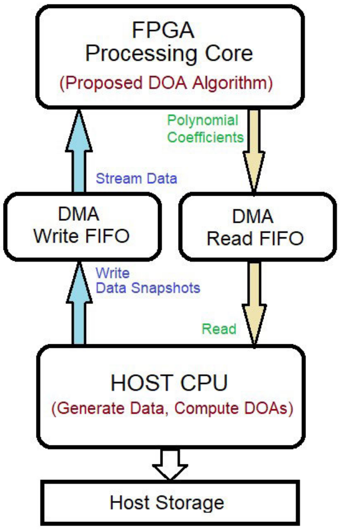


FIGURE 17. Hybrid implementation model for the proposed algorithm.

The proposed FPGA implementation follows a pipeline architecture consisting of six stages, as shown in Fig. 18 below. Each stage of the pipeline was created as a separate LabVIEW virtual instrument (VI) [39]. Programming of each of the VIs shown in Fig. 19 through Fig. 24 utilized LabVIEW FPGA modules with fixed-point arithmetic (16-bit word, 8-bit integer), chosen for its balance of accuracy, speed, and efficient resource usage, unlike floating-point which provides higher accuracy but demands more FPGA resources and reduces performance.

Received data vector at the ULA with nine antenna elements is generated by the Host Processor and written to the DMA FIFO which in turn is read by the FPGA for further processing. The FPGA generates the coefficients of the polynomial whose roots are utilized for computing the DOAs by

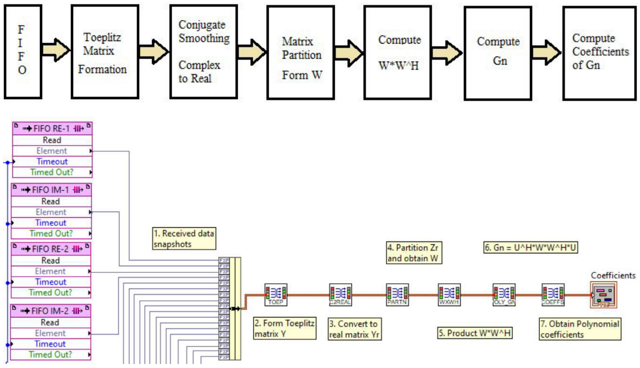


FIGURE 18. Pipeline architecture block diagram (top) and its implementation in LabVIEW FPGA (bottom).

the Host Processor according to (21). The six stages of the pipeline correspond to Step 1 through Step 6 of the proposed algorithm, respectively. The final two steps in the proposed algorithm are implemented on the Host Processor.

This hybrid design, where the FPGA performs all pre-processing and outputs only polynomial coefficients while the host handles root finding and DOA estimation, balances efficiency and flexibility. The FPGA delivers fast, parallel matrix operations with minimal data transfer, while the host provides precision and adaptability for the sensitive root-ing step. This approach saves FPGA resources, simplifies updates, and leverages the host's flexibility for accurate DOA estimation, though it introduces some dependency on host-side latency.

Fig. 19 shows the formation of the Toeplitz structure matrix from the received data vector at the ULA. It is implemented without the need for any arithmetic operations.

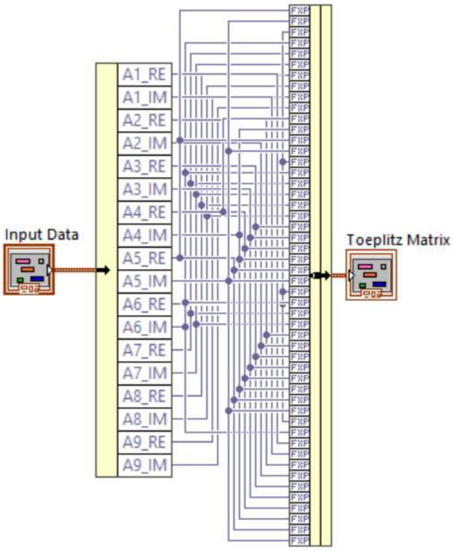


FIGURE 19. LabVIEW FPGA schematic for constructing the Toeplitz structure matrix as in (8) and (9).

Fig. 20 below shows the LabVIEW FPGA implementation of the operation to convert the complex-valued data matrix $\hat{\boldsymbol{Y}}$ (of size 5 × 5) to a real-valued matrix $\hat{\boldsymbol{Y}}_{\boldsymbol{r}}$ as in (12). It converts all complex-valued elements to real-valued ones in parallel, consuming 25 ADD modules and requiring only 2 clock cycles for the conversion.

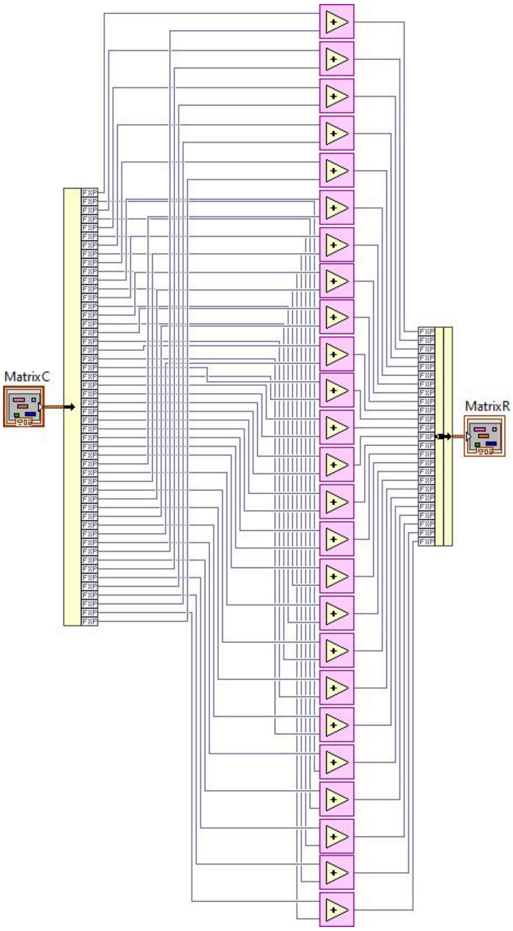


FIGURE 20. LabVIEW FPGA schematic for matrix conversion from complex to real as shown in (12).

Fig. 21 below shows the LabVIEW FPGA implementation of Step 4 and Step 5 of the proposed algorithm using Add/Subtract, Multiply/Divide, and Negation modules consuming 33 clock cycles.

Fig. 22 shows a very efficient implementation for computing **Gn** as in (19), requiring only 11 Negation and 11 Add/Subtract modules and executing in only 2 clock cycles. Fig. 23 illustrates how the product $\mathbf{WW}^{\mathrm{H}}$ is implemented as in (20) with only 6 Multiply and 4 Negation modules and can be executed in 2 clock cycles.

Fig. 24 shows the implementation for computing the coefficients of **Gn**. The coefficients are computed by finding the sum of each diagonal of the matrix **Gn**, ordered from the bottom-left diagonal to the top-right diagonal. These coefficients are required for finding the roots of the polynomial **Gn** to finally compute the DOAs of the incident signals on the

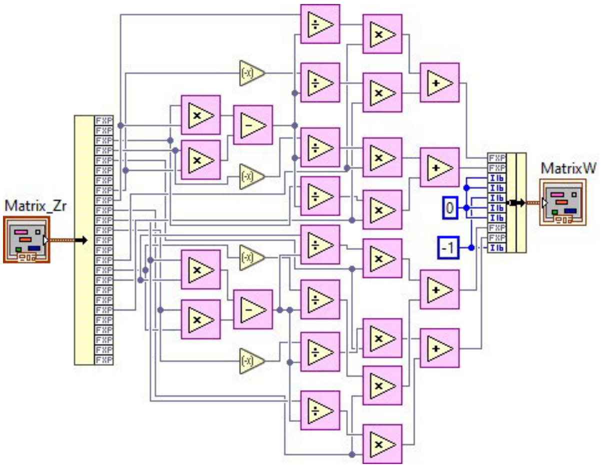


FIGURE 21. LabVIEW FPGA schematic for matrix partitioning stage of the algorithm and formation of matrix W.

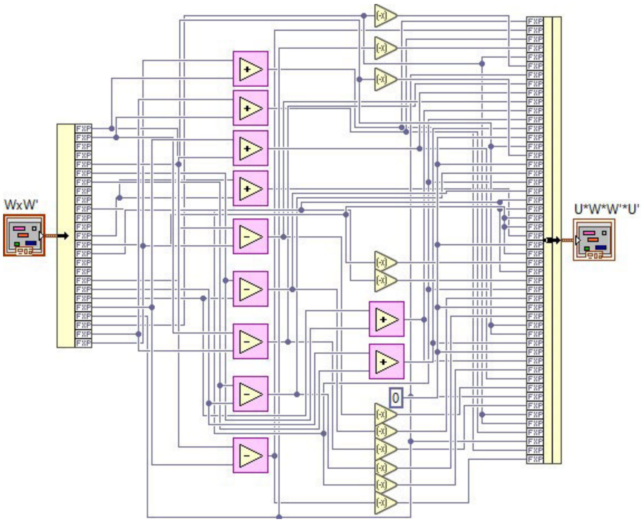


FIGURE 22. LabVIEW FPGA schematic for computing Gn as in (19).

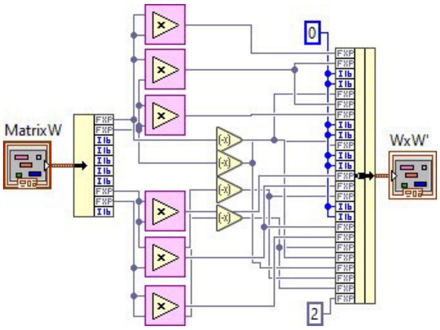


FIGURE 23. LabVIEW FPGA schematic for computing $W^{*}W^{H}$ as in (20).

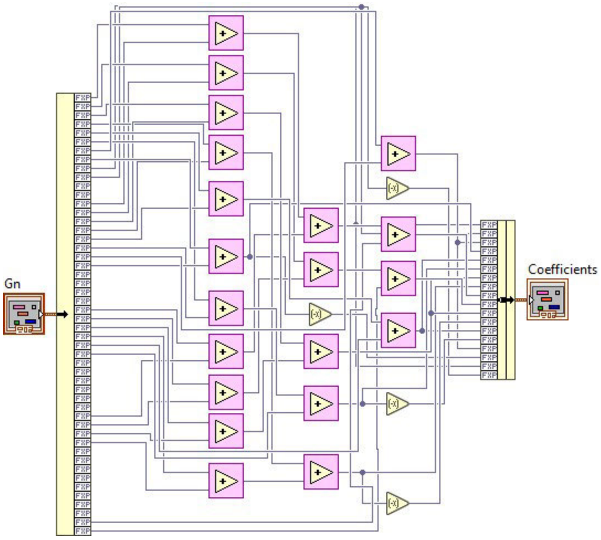
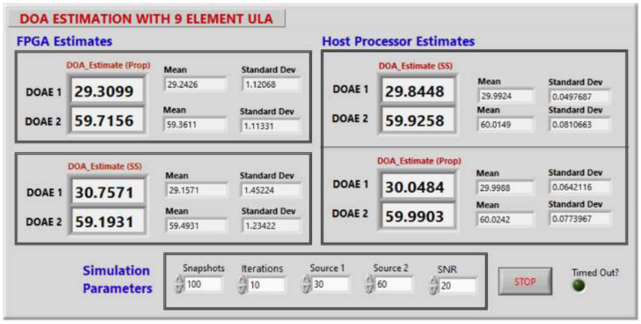


FIGURE 24. LabVIEW FPGA schematic for computing the coefficients of the polynomial Gn.


DOA ESTIMATION WITH 9 ELEMENT ULA

FPGA Estimates

DOA_Estimate (Prop)
DOAE 1 29.3099 Mean 29.2426 Standard Dev 1.12068
DOAE 2 59.7156 Mean 59.3611 Standard Dev 1.11331

DOA_Estimate (SS)
DOAE 1 30.7571 Mean 29.1571 Standard Dev 1.45224
DOAE 2 59.1931 Mean 59.4931 Standard Dev 1.23422

Host Processor Estimates

DOA_Estimate (SS)
DOAE 1 29.8448 Mean 29.9924 Standard Dev 0.0497687
DOAE 2 59.9258 Mean 60.0149 Standard Dev 0.0810663

DOA_Estimate (Prop)
DOAE 1 30.0484 Mean 29.9988 Standard Dev 0.0642116
DOAE 2 59.9903 Mean 60.0242 Standard Dev 0.0773967

Simulation Parameters
Snapshots 100 Iterations 10 Source 1 30 Source 2 60 SNR 20

FIGURE 25. GUI for LabVIEW FPGA simulation of proposed algorithm.

Host Processor and display them on the front panel GUI as seen in Fig. 25.

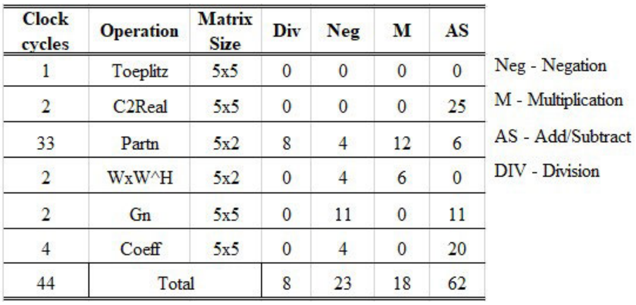
A GUI built for LabVIEW FPGA simulation is shown below in Fig. 25 which depicts simulation results of running the proposed algorithm on the FPGA as well as the Host Processor for two sources placed at 30° and 60° respectively from the ULA with SNR = 20 dB. The simulation is run with 100 snapshots and repeated for 10 iterations.

B. FPGA RESOURCE CONSUMPTION AND LATENCY

Table 3 lists the number of clock cycles consumed and arithmetic modules required in implementing each of the LabVIEW FPGA VIs in the pipelined execution of the proposed algorithm. It is clear from this table that the proposed algorithm is efficient in terms of computation time and resources consumption as it takes only 44 clock cycles and requires very few arithmetic modules relying exclusively on elementary arithmetic operations. By avoiding higher-cost operations such as complex-valued multiplications and square-root, etc., the new approach reduces computational demand and enhances overall efficiency.

Resource consumption data extracted from a successful FPGA compilation report of the proposed algorithm is shown in the Table 4 below. The compilation tool employed is Xilinx Vivado 2019.1 (64-bit). Table 4 also provides a quantitative comparison of FPGA resource utilization and latency between the proposed method (PM) and existing DOA

TABLE 3. Clock cycles and arithmetic operations for proposed DOA.



Clock cycles	Operation	Matrix Size	Div	Neg	M	AS
1	Toeplitz	5x5	0	0	0	0
2	C2Real	5x5	0	0	0	25
33	Partn	5x2	8	4	12	6
2	WxW^H	5x2	0	4	6	0
2	Gn	5x5	0	11	0	11
4	Coeff	5x5	0	4	0	20
44	Total		8	23	18	62

Neg - Negation
M - Multiplication
AS - Add/Subtract
DIV - Division

estimation implementations reported in [43], [44], [45], and [46] (which deploy an 8-element ULA whereas PM deploys a 9-element ULA).

TABLE 4. FPGA device utilization and latency.

Device Utilization	PM	[43]	[44]	[45]	[46]
Total Slices	11615	NA	NA	NA	NA
Slice Registers	72248	27700	95185	28160	30200
Slice LUTs	61397	34200	84658	49601	48100
Block RAMs	83	55	7	23	117
DSP48s	56	268	1311	216	96
Latency (clock cycles)	44	15900	5396	7800	25500
Latency (µs, 100 MHz clock)	0.44	159	53.96	78	255

The results presented in Table 4 demonstrate that PM achieves a favorable balance between resource usage and processing speed.

- Slices, Registers, and LUTs: PM requires 11,615 slices, while comparative data for slices is not reported in [43], [44], [45], and [46]. For registers and LUTs, PM consumes more registers and LUTs than [43], [45] and [46] but lower than [44]. This reflects the cost of implementing decomposition-free Toeplitz processing while still remaining competitive in overall logic usage.
- Block RAMs (BRAMs): PM uses 83 BRAMs, which is higher than [43], [44], and [45], but significantly lower than [46]. The additional BRAM demand in PM comes from structured data buffering, yet it remains manageable for modern FPGA architectures.
- DSP48s: A major strength of PM is its very low DSP consumption (56), which is drastically lower than [43], [44], and [45], and even [46]. This efficiency makes PM particularly attractive for platforms with constrained DSP availability, such as mid-range or embedded-class FPGAs.
- Latency: PM achieves a latency of only 44 clock cycles, outperforming [43], [44], [45], and [46]. This dramatic reduction in latency highlights the decomposition-free architecture's suitability for real-time DOA estimation in time-sensitive applications. Latency values in seconds is also calculated considering a 100 MHz clock and listed in the table.

Overall, the proposed method (PM) exhibits exceptionally low latency and DSP usage compared to existing FPGA-based DOA implementations, while maintaining competitive levels of registers, LUTs, and BRAMs. These characteristics position PM as a highly efficient, hardware-friendly solution for real-time DOA estimation in both coherent and non-coherent source environments.

C. COMPUTATION TIME IN HYBRID MODE

In this section, we calculate the computation time of the proposed algorithm in both hybrid as well as fully software mode, while considering the following simulation parameters:

- $K = 2$ sources, 100 snapshots, 9 antenna elements, 100 iterations

With a clock speed of 100 MHz for PXIe-7993 FPGA, the execution time for computing the coefficients of the polynomial for the proposed algorithm averaged over 100 iterations was found to be ∼0.53 µs. The execution time for computing the DOAs on the Host Processor (3 GHz, 16 GB RAM) by polynomial rooting of **Gn** averaged over 100 iterations was found to be ∼61 µs. The net execution time is shown in Table 5 below.

The execution time of the proposed algorithm when executed entirely on the Host Processor under the same simulation conditions was found to be ∼1.03 ms.

The hybrid implementation (FPGA + Host) is ∼17 times faster than the entirely software one. In contrast, the execution time for TS_DOA [9] is ∼91 ms which makes the proposed method ∼88 times faster when executed entirely on the Host Processor and ∼1481 times faster when executed in hybrid mode.

TABLE 5. Computation time for proposed method in hybrid mode.

Execution stage	Proposed Method (PM)		TS_DOA
	Hybrid	Software	Software
FPGA	0.44 µs	NA	NA
Host Proc.	61 µs	1.03 ms	91 ms
Total	61.44 µs	1.03 ms	91 ms

The FPGA portion of PM requires only 44 clock cycles (0.44 µs), making it by far the fastest among all compared methods. Although polynomial rooting and DOA computation are performed on the Host Processor, leading to a total latency of 61.44 µs, the hybrid design still outperforms [43] and [46], and remains competitive with [44] and [45].

D. HYBRID IMPLEMENTATION: SOME OBSERVATIONS

A key advantage of the proposed hybrid implementation lies in its scalability when applied to larger arrays and multiple source scenarios. Pure FPGA-only approaches often suffer from rapid growth in resource demand as the number of antenna elements and incident sources increases. In particular, eigenvalue decomposition (EVD), singular value decomposition (SVD), or matrix inversions used in classical FPGA-based designs have computational complexity on the order of $O(N^3)$ and require substantial FPGA resources such as DSP48s, LUTs, and block RAMs. As a result, the latency and resource usage of these methods scale poorly, making them impractical for large-scale ULAs or scenarios involving many sources.

In contrast, the hybrid method is specifically designed to minimize FPGA workload by offloading the most computationally intensive tasks (polynomial rooting and DOA computation) to the host processor. The FPGA fabric is responsible only for lightweight, highly parallelizable operations, such as covariance-free Toeplitz matrix formation and basic arithmetic, whose complexity scales as O(NL), where N is the number of array elements and L is the number of snapshots. This ensures that FPGA resource consumption grows linearly with the array size, avoiding saturation even on mid-range devices.

The final stage, involving polynomial rooting and angle computation, scales with the number of antenna elements and has complexity on the order of $O(N^3)$. Importantly, this step is executed on the host processor, which can leverage optimized numerical libraries and floating-point units that are far better suited for handling higher-order polynomials than FPGA hardware. This division of labor provides two clear scalability benefits:

- Resource Efficiency: Since decomposition-heavy tasks are avoided on the FPGA, the method requires far fewer LUTs, DSP slices, and block RAMs compared to existing approaches, making it feasible for larger ULAs.
- Flexible Expansion: As K increases, the rooting step grows in cost, but because it runs on the host CPU, scalability is achieved without requiring additional FPGA resources.

Therefore, the hybrid design combines the low-latency and deterministic parallelism of FPGA with the computational flexibility of the host, ensuring that the overall system can handle larger numbers of antenna elements and impinging sources while keeping both FPGA utilization and latency under control. This approach is well-suited for practical real-time implementation scenarios, although certain bottlenecks still need to be addressed.

The main bottleneck of the hybrid implementation stems from the reliance on the host processor for polynomial rooting and DOA computation, which introduces an additional 61 μs latency compared to the FPGA-side latency of only 0.44 μs. Data transfer overhead between FPGA and host can further impact timing. Pure FPGA designs avoid this transfer delay and offer deterministic execution, but they incur high resource consumption and limited scalability due to the decomposition and rooting stages, which demand extensive DSPs, BRAMs, and long latencies. Thus, hybrid designs trade determinism for efficiency, flexibility, and scalability, while pure FPGA approaches face challenges in supporting larger arrays and higher source counts.

A second challenge is the effect of host-side execution on overall system throughput. When processed serially, the throughput is limited to ∼16.3k estimates per second, far below the FPGA's potential. However, this bottleneck can be alleviated by employing pipelining, batching, and parallelization strategies such as double buffering, vectorized rooting, or distributing tasks across CPU cores and GPUs. These optimizations can significantly improve throughput without increasing FPGA complexity, making the hybrid method both resource-efficient and scalable for practical real-time DOA estimation.

VI. CONCLUSION AND FUTURE WORK

This paper presented a decomposition-free direction-of-arrival (DOA) estimation method that exploits the inherent Toeplitz structure of uniform linear arrays (ULAs) and eliminates the need for matrix decompositions such as EVD, SVD, or QR. By directly constructing a structured Toeplitz matrix, transforming it into a real-valued compound form, and applying polynomial rooting for DOA estimation, the proposed approach achieves low arithmetic complexity and avoids the computational bottlenecks of subspace-based techniques.

A hybrid hardware-software implementation was developed, wherein all preprocessing and matrix construction steps were executed on the FPGA while polynomial rooting and final DOA computation were performed on the host processor. This architecture demonstrated very low FPGA resource utilization, fast execution, and robustness to coherent as well as non-coherent sources. The hybrid design achieved a total latency of 61.44 μs, with only 0.44 μs incurred on the FPGA and the remainder on the host processor, thereby offering a highly efficient trade-off between hardware complexity and processing speed.

In summary, the high estimation accuracy of the proposed method for both non-coherent and coherent signals across diverse conditions, and its decomposition-free structure combined with the hybrid execution model makes the proposed method particularly well-suited for real-time, resource-constrained applications such as wireless communications, radar, and electronic warfare systems. Future work will incorporate real-time experimental characterization under challenging real-world conditions as well as focus on extending the proposed approach to non-uniform and sparse array geometries.

ACKNOWLEDGMENT

The authors extend their appreciation to Prince Mohammad Bin Fahd University for its support in carrying out this research.

REFERENCES

[1] H. L. Van Trees, *Optimum Array Processing: Part IV of Detection, Estimation, & Modulation Theory*. Hoboken, NJ, USA: Wiley, 2002.

[2] P. Stoica and R. L. Moses, *Spectral Analysis of Signals* Upper Saddle River, NJ, USA: Pearson Prentice-Hall, 2005.

[3] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. Antennas Propag.*, vol. AP-34, no. 3, pp. 276–280, Mar. 1986.

[4] R. Roy and T. Kailath, "ESPRIT-estimation of signal parameters via rotational invariance techniques," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, no. 7, pp. 984–995, Jul. 1989.

[5] M. Wax and T. Kailath, "Detection of signals by information theoretic criteria," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-33, no. 2, pp. 387–392, Apr. 1985.

[6] S. U. Pillai and B. H. Kwon, "Forward/backward spatial smoothing techniques for coherent signal identification," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, no. 1, pp. 8–15, Jan. 1989.

[7] J. Li, P. Stoica, and Z. Wang, "On robust Capon beamforming and diagonal loading," *IEEE Trans. Signal Process.*, vol. 51, no. 7, pp. 1702–1715, Jul. 2003, doi: 10.1109/TSP.2003.812831.

[8] S. Liu, L. Yang, J. H. Huang, and Q. P. Jiang, "Generalization propagator method for DOA estimation," *Prog. Electromagn. Res. M*, vol. 37, pp. 119–125, 2014, doi: 10.2528/pierm14051701.

[9] F.-G. Yan, S. Liu, J. Wang, and M. Jin, "Two-step root-MUSIC for direction of arrival estimation without EVD/SVD computation," *Int. J. Antennas Propag.*, vol. 2018, pp. 1–8, Aug. 2018.

[10] J. Li, B. Halder, P. Stoica, and M. Viberg, "Computationally efficient angle estimation for signals with known waveforms," *IEEE Trans. Signal Process.*, vol. 43, no. 9, pp. 2154–2163, Sep. 1995, doi: 10.1109/78.414778.

[11] X. Wu, W.-P. Zhu, and J. Yan, "A Toeplitz covariance matrix reconstruction approach for direction-of-arrival estimation," *IEEE Trans. Veh. Technol.*, vol. 66, no. 9, pp. 8223–8237, Sep. 2017, doi: 10.1109/TVT.2017.2695226.

[12] S. Liu, Z. Mao, Y. D. Zhang, and Y. Huang, "Rank minimization-based Toeplitz reconstruction for DOA estimation using coprime array," *IEEE Commun. Lett.*, vol. 25, no. 7, pp. 2265–2269, Jul. 2021, doi: 10.1109/LCOMM.2021.3075227.

[13] J. Dai, T. Qiu, S. Luan, Q. Tian, and J. Zhang, "An improved Toeplitz approximation method for coherent DOA estimation in impulsive noise environments," *Entropy*, vol. 25, no. 6, p. 960, Jun. 2023, doi: 10.3390/e25060960.

[14] A. A. Hussain, N. Tayem, and A.-H. Soliman, "Low complexity DOA estimation of multiple coherent sources using a single direct data snapshot," *IEEE Access*, vol. 12, pp. 2371–2388, 2024, doi: 10.1109/ACCESS.2023.3348414.

[15] S. Wu, Y. Yuan, L. Huang, K. Cui, and N. Yuan, "Grid-less DOA estimation of coherent sources based on the covariance matrix recovery," *Phys. Commun.*, vol. 46, Jun. 2021, Art. no. 101345, doi: 10.1016/j.phycom.2021.101345.

[16] P. Pal and P. P. Vaidyanathan, "Nested arrays: A novel approach to array processing with enhanced degrees of freedom," *IEEE Trans. Signal Process.*, vol. 58, no. 8, pp. 4167–4181, Aug. 2010.

[17] P. Pal and P. P. Vaidyanathan, "Coprime sampling and the music algorithm," in *Proc. Digit. Signal Process. Signal Process. Educ. Meeting (DSP/SPE)*, Sedona, AZ, USA, Jan. 2011, pp. 289–294, doi: 10.1109/DSP-SPE.2011.5739227.

[18] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, Aug. 2009.

[19] N.-J. Ruan, F.-Q. Wen, L. Ai, and K. Xie, "A PARAFAC decomposition algorithm for DOA estimation in colocated MIMO radar with imperfect waveforms," *IEEE Access*, vol. 7, pp. 14680–14688, 2019, doi: 10.1109/ACCESS.2019.2894747.

[20] F. Xu, H. Zheng, and S. A. Vorobyov, "Tensor-based 2-D DOA estimation for L-shaped nested array," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 60, no. 1, pp. 604–618, Feb. 2024, doi: 10.1109/TAES.2023.3326793.

[21] M.-Y. You, A.-N. Lu, Y.-X. Ye, K. Huang, and B. Jiang, "A review on machine learning-based radio direction finding," *Math. Problems Eng.*, vol. 2020, Aug. 2020, Art. no. 8345413, doi: 10.1155/2020/8345413.

[22] S. Zheng, Z. Yang, W. Shen, L. Zhang, J. Zhu, Z. Zhao, and X. Yang, "Deep learning-based DOA estimation," *IEEE Trans. Cognit. Commun. Netw.*, vol. 10, no. 3, pp. 819–835, Jun. 2024, doi: 10.1109/TCCN.2024.3360527.

[23] D. H. Shmuel, J. P. Merkofer, G. Revach, R. J. G. van Sloun, and N. Shlezinger, "SubspaceNet: Deep learning-aided subspace methods for DoA estimation," *IEEE Trans. Veh. Technol.*, vol. 74, no. 3, pp. 4962–4976, Mar. 2025, doi: 10.1109/TVT.2024.3496119.

[24] W. Fang, Z. Cao, D. Yu, X. Wang, Z. Ma, B. Lan, C. Song, and Z. Xu, "A lightweight deep learning-based algorithm for array imperfection correction and DOA estimation," *J. Commun. Inf. Netw.*, vol. 7, no. 3, pp. 296–308, Sep. 2022, doi: 10.23919/JCIN.2022.9906943.

[25] C. M. Mylonakis, P. Velanas, and Z. D. Zaharis, "On the implementation of temporal fusion transformers for target recognition and tracking," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Milan, Italy, Mar. 2025, pp. 1–6.

[26] A. Liu, J. Guo, Y. Arnatovich, and Z. Liu, "Lightweight deep neural network with data redundancy removal and regression for DOA estimation in sensor array," *Remote Sens.*, vol. 16, no. 8, p. 1423, Apr. 2024, doi: 10.3390/rs16081423.

[27] Z. Gong, X. Su, P. Hu, S. Liu, and Z. Liu, "Deep unfolding sparse Bayesian learning network for off-grid DOA estimation with nested array," *Remote Sens.*, vol. 15, no. 22, p. 5320, Nov. 2023.

[28] Z. Zhou, T. Jin, Y. Li, C. Wang, Z. Zhou, Y. Huang, and Y. Sun, "Enhanced coherent DOA estimation in low SNR environments through contrastive learning," *IEEE Trans. Instrum. Meas.*, vol. 74, pp. 1–21, 2025, doi: 10.1109/TIM.2025.3547111.

[29] R. Ghayoula, W. Amara, I. E. Gmati, A. Smida, and J. Fattahi, "An efficient FPGA implementation of MUSIC processor using cyclic Jacobi method: LiDAR applications," *Appl. Sci.*, vol. 12, no. 19, p. 9726, Sep. 2022, doi: 10.3390/app12199726.

[30] H. Zhou, X. Jing, B. Li, Z. Zhou, and B. Li, "Implementation of DOA estimation algorithm based on FPGA," in *Signal and Information Processing, Networking and Computers*. Singapore: Springer, 2022, pp. 46–52.

[31] A. Alhamed, N. Tayem, T. Alshawi, S. Alshebeili, A. Alsuwailem, and A. Hussain, "FPGA-based real-time implementation for direction-of-arrival estimation," *J. Eng.*, vol. 2017, no. 6, pp. 260–265, Jun. 2017, doi: 10.1049/joe.2017.0165.

[32] A. A. Hussain, N. Tayem, M. O. Butt, A.-H. Soliman, A. Alhamed, and S. Alshebeili, "FPGA hardware implementation of DOA estimation algorithm employing LU decomposition," *IEEE Access*, vol. 6, pp. 17666–17680, 2018, doi: 10.1109/ACCESS.2018.2820122.

[33] A. A. Hussain, N. Tayem, and A.-H. Soliman, "LDL decomposition-based FGPA real-time implementation of DOA estimation," in *Proc. 52nd Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, USA, Oct. 2018, pp. 1163–1168, doi: 10.1109/ACSSC.2018.8645387.

[34] A. A. Hussain, N. Tayem, A.-H. Soliman, and R. M. Radaydeh, "FPGA-based hardware implementation of computationally efficient multi-source DOA estimation algorithms," *IEEE Access*, vol. 7, pp. 88845–88858, 2019.

[35] A. A. Hussain, N. Tayem, and A.-H. Soliman, "Matrix decomposition methods for efficient hardware implementation of DOA estimation algorithms: A performance comparison," in *Proc. 4th Int. Conf. Workshops Recent Adv. Innov. Eng. (ICRAIE)*, Penang, Malaysia, Nov. 2019, pp. 1–7.

[36] S. Zhou and L. Zhou, "Field programmable gate array (FPGA) implementation of parallel Jacobi for eigen-decomposition in direction of arrival (DOA) estimation algorithm," *Remote Sens.*, vol. 16, no. 20, p. 3892, Oct. 2024, doi: 10.3390/rs16203892.

[37] *FPGA-Based Real-Time Implementation for Direction of Arrival Estimation Algorithm Using QR Decomposition*, National Instruments, Austin, TX, USA, 2023.

[38] *NI FlexRIO FPGA Module Installation Guide & Specifications*, document 373047B-01, National Instruments, Austin, TX, USA, 2024.

[39] *NI LabVIEW Software Platform*. Accessed: Aug. 8, 2025. [Online]. Available: http://www.ni.com/labview/

[40] B. Qi, X. Liu, D. Dou, Y. Zhang, and R. Hu, "An enhanced DOA estimation method for coherent sources via Toeplitz matrix reconstruction and Khatri–Rao subspace," *Electronics*, vol. 12, no. 20, p. 4268, Oct. 2023, doi: 10.3390/electronics12204268.

[41] Z. Yang, L. Xie, and P. Stoica, "Vandermonde decomposition of multilevel Toeplitz matrices with application to multidimensional super-resolution," *IEEE Trans. Inf. Theory*, vol. 62, no. 6, pp. 3685–3701, Jun. 2016, doi: 10.1109/TIT.2016.2553041.

[42] J. Cui, W. Pan, and H. Wang, "Direction of arrival estimation method based on eigenvalues and eigenvectors for coherent signals in impulsive noise," *Mathematics*, vol. 12, no. 6, p. 832, Mar. 2024, doi: 10.3390/math12060832.

[43] Z. Li, W. Wang, R. Jiang, S. Ren, X. Wang, and C. Xue, "Hardware acceleration of MUSIC algorithm for sparse arrays and uniform linear arrays," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 69, no. 7, pp. 2941–2954, Jul. 2022.

[44] L. Lutao, C. Ying, and Z. Yu, "FPGA implementation of direction of arrival estimation method for polarization sensitive array," *J. Electron. Inf. Technol.*, vol. 45, no. 9, pp. 3340–3349, 2023, doi: 10.11999/JEIT221146.

[45] S. Lahti, T. Aaltonen, E. Rastorgueva-Foi, J. Talvitie, B. Tan, and T. D. Hämäläinen, "An efficient high-level synthesis implementation of the MUSIC DoA algorithm for FPGA," in *Proc. 27th Int. Symp. Design Diag. Electron. Circuits Syst. (DDECS)*, Apr. 2024, pp. 142–147.

[46] H. Chen, K. Chen, K. Cheng, Q. Chen, Y. Fu, and L. Li, "An efficient hardware accelerator for the MUSIC algorithm," *Electronics*, vol. 8, no. 5, p. 511, May 2019.

[47] H. Nan, X. Ma, Y. Han, and W. Sheng, "A computationally efficient MUSIC algorithm with an enhanced DOA estimation performance for a crossed-dipole array," *Sensors*, vol. 25, no. 11, p. 3469, May 2025, doi: 10.3390/s25113469.

[48] J. Huang and S. Xu, "A low-complexity DOA estimation method for nested array via root-MUSIC," in *Proc. 8th Int. Conf. Electron. Inf. Technol. Comput. Eng.*, New York, NY, USA, Oct. 2024, pp. 462–466, doi: 10.1145/3711129.3711208.

[49] L. Nagaraju and P. K. Kumar, "A novel approach for optimal number of antenna elements for 1D-DOA based on compressive sensing framework," in *Proc. Int. Conf. Commun., Circuits, Syst. (IC3S)*, May 2023, pp. 1–5, doi: 10.1109/IC3S57698.2023.10169600.

[50] D. Meng, X. Li, and W. Wang, "Robust tensor decomposition approach for DOA estimation with EMVS-MIMO radar," *IEEE Sensors J.*, vol. 24, no. 19, pp. 31262–31271, Oct. 2024, doi: 10.1109/JSEN.2024.3441839.

AHMED A. HUSSAIN (Student Member, IEEE) received the M.S. degree in electrical and computer engineering from the University of Florida, Gainesville, in 1998. He is a Lecturer with the Department of Electrical Engineering, Prince Mohammad Bin Fahd University, Saudi Arabia. He has more than 22 years of university teaching experience in electrical engineering. He was with Motorola, as an Embedded Software Engineer. His research interests include wireless communications, array signal processing, digital and embedded systems, and engineering programs assessment and accreditation. He has several research publications and U.S. patents in his areas of interest.

NIZAR TAYEM (Member, IEEE) received the Ph.D. degree in electrical engineering from Wichita State University, Kansas, USA. He is an Associate Professor with the Electrical Engineering Department, East Texas A&M University, Commerce, TX, USA. He worked on research projects for Aerospace Sensor Networks Technology Thrust, Minority Leader's Program, and AFRL/Clarkson Aerospace. He is the author/co-author of more than 60 research publications in recognized international journals and conferences. His research interests include signal processing algorithm for wireless communication systems, array signal processing, source localization, MIMO systems, channel estimation, and OFDM and OFDMA communication systems. He is a Senior Member of the Institute of Doctors Engineers and Scientists (IDES). He is a regular reviewer of many well-known journals. He served as the Editor-in-Chief for *International Journal on Electrical and Power Engineering* (ACEEE, USA).

ABDEL HAMID SOLIMAN received the B.Sc. degree in electronics and telecommunications from the Arab Academy for Science and Technology and Maritime Transport, the M.Sc. degree in smart data acquisition systems from Alexandria University, and the Ph.D. degree in image/video processing from the University of Staffordshire. He is a Professor in telecommunication and signal processing with the School of Digital, Technology, Innovation, and Business, University of Staffordshire. He has a multi-disciplinary academic/research experience in digital signal processing, telecommunications, data acquisition systems, wireless sensor networks (WSN), and image/video processing. His research focused on harnessing these technologies to contribute in the smart cities applications. He has more than 27 years' experience of working in higher education institutes. In addition to his research and teaching/learning activities, he is involved in several enterprise projects and consultancy activities for national and international companies. He is leading and involved in several European funded projects in education, industry automation, and health care applications.