Comparison of Efficient FPGA-Oriented Direction-of-Arrival Algorithms on an RFSoC 4x2 Board

István Váradi[1]

Supervisor(s): Arash Asadi[1]**, Florian Kosterhon**[1]

[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfillment of the Requirements
For the Bachelor of Computer Science and Engineering
June 21, 2026

Name of the student: István Váradi
Final project course: CSE3000 Research Project
Thesis committee: Arash Asadi, Florian Kosterhon, Georgios Iosifidis

***Abstract*—Reliable UAV navigation in GNSS-challenged environments often requires complementary sensing methods that can operate when satellite positioning is degraded or unavailable. Radio-frequency signals of opportunity can support this type of navigation, but their practical use depends on intermediate processing stages such as direction-of-arrival estimation. This work compares three FPGA-oriented DoA estimation approaches, MUSIC, ESPRIT, and an LU-decomposition-based estimator, within a common RFSoC 4x2 and Vitis HLS implementation framework. Instead of proposing a new estimator, the study evaluates existing hardware-oriented methods under shared assumptions about the receiver, array, fixed-point, signal model, and measurements. All methods use the same four-element ULA configuration, broadside angle convention, 256-snapshot frame structure, and captured RF data. The evaluation compares angular accuracy, HLS kernel latency, update interval, and FPGA resource usage. MUSIC provides the strongest full-range angular robustness, while the LU-based estimator gives the lowest latency and resource cost but produces larger edge-angle error tails. ESPRIT provides intermediate accuracy while retaining timing and resource characteristics close to those of MUSIC, since both methods rely on EVD-based subspace processing.**

***Index Terms*—direction of arrival, DoA, FPGA, RFSoC 4x2, MUSIC, ESPRIT, Vitis HLS, passive positioning**

I. Introduction

Global navigation satellite systems (GNSS) are the backbone of modern unmanned aerial vehicle (UAV) positioning. In most conditions, they are accurate, widely available, and easy to integrate. However, because many navigation systems depend on them, GNSS is an attractive target for spoofing and jamming attacks [1], [2]. Their performance also degrades in dense urban areas due to multipath effects and in indoor environments where signals may be attenuated or unavailable. Therefore, it is important to develop resilient alternatives and complementary methods for when GNSS navigation is unavailable.

Several alternative navigation approaches have been investigated, but each introduces its own constraints and limitations. For example, camera-based methods can support localization by recognizing environments or tracking motion [3], but in low-visibility conditions, visual sensors may become unreliable. These methods can also be computationally intensive, increasing computational load and power consumption [4], [5]. Another approach is to use on-board inertial measurement units (IMUs) to track vehicle movement, integrating motion from a previously known position. IMUs are self-contained and largely insensitive to external interference. Their main drawback is drift. Since their measurements can be noisy, the positioning error grows unless another source of information corrects the inertial solution [6], [7]. Therefore, reliable UAV navigation in difficult environments often requires a sensor-fusion approach rather than relying on a single GNSS replacement.

Radio-frequency signals of opportunity (SoOps) are a useful candidate class for such a fused navigation system. These signals are not necessarily designed for positioning, but are often already present in the environment through communication, broadcast, cellular, WiFi, or satellite systems. Previous studies have shown that they can support positioning and tracking in GNSS-challenged conditions [8], [9]. At the same time, they pose practical challenges, including unknown and often disadvantageous transmitter geometry, imperfect synchronization, non-line-of-sight propagation, and questions about signal integrity [2], [10], [11]. Since no single RF source is consistently available or reliable across all environments, practical systems often need to exploit multiple signals or combine RF measurements with other sensing modalities.

A key measurement obtainable from RF SoOps is the direction of arrival (DoA). By estimating the incoming angle, a receiver can track relative movement over time or estimate its own position through triangulation when multiple transmitters with known positions are available [12]. DoA estimation is therefore an important intermediate step for RF-based UAV navigation. The aim of this work is not to implement a complete navigation system, but to examine the DoA estimation stage on which such a system could depend.

A wide range of DoA estimation algorithms has been proposed, with different trade-offs between complexity and accuracy. Simpler methods, such as phase-interferometry or beamforming, estimate the arrival angle from inter-antenna phase differences [13], [14] or from the spatial power response [15]. High-resolution subspace methods such as MUSIC and ESPRIT can provide better angular resolution but require more demanding operations, including covariance estimation, eigenvalue or matrix decompositions, and, for MUSIC, spectral searches [16], [17]. For an embedded UAV platform, accuracy alone is not sufficient, and the algorithm must also be evaluated for latency, resource consumption, and update rate.

Field-programmable gate arrays (FPGAs) are well-suited to this type of workload because they can exploit parallelism and can be equipped with capable RF processing resources. Recent work has demonstrated FPGA implementations of MUSIC, ESPRIT, beamforming, and decomposition-based DoA methods [18], [19], [20], [21], [22], [23], [24]. These studies show that real-time DoA estimation on FPGA hardware is feasible. However, they are not always easy to compare. Reported performance is influenced not only by the algorithm, but also by other variables such as the array configuration and snapshot length. The implementation approach also affects the comparison. Designs written directly in hardware description languages and designs generated through high-level synthesis may prioritize different performance metrics, depending on the tools and optimization settings used. When reported performance differs only by small margins, as is the case for many of these algorithms, the targeted hardware can introduce ambiguity about whether the algorithm is inherently more efficient or whether the performance gains are due to improvements in the platform. As a result, it remains difficult to determine which DoA algorithms are intrinsically suitable for low-latency embedded deployment.

The main contribution of this work is a common RFSoC- and HLS-based comparison of three FPGA-oriented DoA estimators: MUSIC, ESPRIT, and an LU-decomposition-based estimator. The algorithms are evaluated under the same four-element ULA geometry, snapshot structure, fixed-point word-

length variants, implementation workflow, and evaluation metrics. The comparison focuses on angular accuracy, latency, update rate, and FPGA resource utilization rather than proposing a new estimator.

II. System Model and Comparison Framework

Direction-of-arrival estimation requires a signal model that relates the received antenna samples to the angle of an incoming RF signal. In this work, the model is used primarily to define the common assumptions under which the selected algorithms are implemented and compared. This section introduces the notation, array geometry, signal representation, and the scope of comparison used throughout the paper. The RFSoC data path is described in Section IV, while the measurement procedure and evaluation metrics are described in Section V.

A. Array Geometry and Angle Convention

The receiver is modeled as a uniform linear array (ULA) with four antenna elements. This array size is determined by the four receive channels used in the RFSoC 4x2-based implementation. The antenna elements are placed along a single axis with approximately equal spacing. The experiments use a carrier frequency of 2.480 GHz, which is comparable to real-world WiFi signals while providing a controlled test environment. For this frequency, the physical array spacer is designed with an inter-element spacing of 60.44 mm, approximately half the signal wavelength.

All angles are defined relative to the array broadside. In this convention, an angle of 0° corresponds to a signal arriving perpendicular to the array axis. The evaluated angular range is

$$-90^\circ \leq \theta \leq 90^\circ. \tag{1}$$

The reference angle, denoted by θ, represents the physical direction of the dominant transmitter during a measurement. Some DoA algorithms and reference implementations use different internal angle definitions, signs, or reference axes. To correct for this, all algorithm outputs are converted to the same broadside convention before comparison.

B. Snapshot and Covariance Model

The signal model assumes a narrowband plane wave arriving from the far field. The intended experimental case uses a single dominant stationary transmitter, so the received data are modeled as a single-source case. At sample index n, the four-channel snapshot is written as

$$\mathbf{x}[n] = \begin{bmatrix} x_0[n] & x_1[n] & x_2[n] & x_3[n] \end{bmatrix}^T, \tag{2}$$

where each entry contains a single complex baseband sample from one receive channel. Under the single-source model, the same snapshot is modeled as

$$\mathbf{x}[n] = \mathbf{a}(\theta)s[n] + \mathbf{w}[n], \tag{3}$$

where $s[n]$ is the received source signal, $\mathbf{a}(\theta)$ is the array steering vector, and $\mathbf{w}[n]$ contains receiver noise, interference, reflections, and other unmodeled disturbances. The disturbance term is not restricted to an idealized simulated noise model, as the comparison is based on a practical laboratory implementation rather than a purely analytical or synthetic study.

For the four-element ULA, the steering vector is written as

$$\mathbf{a}(\theta) = \begin{bmatrix} 1 \\ e^{-j2\pi \frac{d}{\lambda} \sin(\theta)} \\ e^{-j2\pi 2 \frac{d}{\lambda} \sin(\theta)} \\ e^{-j2\pi 3 \frac{d}{\lambda} \sin(\theta)} \end{bmatrix}, \tag{4}$$

where d is the inter-element spacing and λ is the wavelength. In the implemented measurement setup, d is approximately $\lambda/2$. The sign of the phase progression follows the convention used in this model. In practice, the sign depends on the ordering of the antenna channels, the chosen reference element, and the direction assigned as positive rotation. This convention is therefore treated as part of the implementation interface, and estimated angles are converted before comparison when required.

All selected algorithms operate on the same sample covariance estimate. The sample covariance matrix is computed as

$$\hat{\mathbf{R}}_{xx} = \frac{1}{N} \sum_{n=0}^{N-1} \mathbf{x}[n]\mathbf{x}^H[n]. \tag{5}$$

Unless otherwise stated, the covariance matrix is estimated using $N = 256$ snapshots per DoA estimate. The snapshot length affects the stability of the covariance estimate, as well as buffering, latency, and update rate. It is therefore part of the common comparison framework rather than only a statistical parameter.

C. Comparison Scope

The comparison is limited to the DoA estimation stage rather than a complete navigation or passive positioning system. MUSIC, ESPRIT, and LU-DOA are evaluated under the same four-element ULA geometry, broadside angle convention, single-source narrowband signal model, 256-snapshot covariance input, Vitis HLS workflow, and accuracy, timing, and resource metrics.

Fixed-point arithmetic is included as word length affects numerical robustness and hardware cost in an FPGA datapath. Four internal kernel formats are evaluated: q16.8, q18.8, q20.8, and q24.8, where qW.I denotes W total bits and I integer bits. The evaluated formats refer to internal kernel arithmetic; the streaming input and output interfaces are described in Section IV.

III. Related Work and Algorithm Selection

Several FPGA-focused DoA estimators have been proposed, including implementations based on subspace methods, matrix decomposition, phase interferometry, and beamforming. Rather than reviewing individual papers in isolation, this section groups prior work by method category and highlights the implementation trade-offs most relevant to this thesis. It concludes by motivating the selection of the algorithms reproduced in this work.

A. FPGA-Based Subspace DoA Methods

Subspace-based DoA methods are widely used as high-resolution reference algorithms. MUSIC and ESPRIT are especially relevant because they are well established in the DoA literature and are commonly used as reference methods for high-resolution array processing [16], [17]. They have also both been implemented on FPGA hardware. MUSIC estimates the DoA by separating the signal and noise subspaces and evaluating an angular spectrum. At the same time, ESPRIT uses the shift-invariance property of subarrays to avoid the angular spectrum search. This gives the two methods different implementation trade-offs. MUSIC retains the spectral search, whereas ESPRIT removes it in favor of a shift-invariance relation, but still relies on covariance and subspace estimation.

Recent FPGA-oriented MUSIC implementations show that the algorithm can be made practical on hardware, but also highlight the cost of the required matrix operations. Lahti et al. present an HLS implementation of single-source MUSIC for a four-antenna ULA with one-degree angular resolution [25]. That work is especially relevant to this thesis because it uses an HLS-based workflow, an iterative Jacobi method for the eigenvalue decomposition, and a four-element ULA. Other FPGA MUSIC implementations also demonstrate the continued relevance of MUSIC as a hardware reference algorithm, including designs that modify the conventional processing chain or accelerate MUSIC for different array configurations [18], [20]. Zhou and Zhou focus specifically on FPGA-based parallel Jacobi eigen-decomposition for DoA estimation and apply the method to MUSIC, showing that covariance matrix decomposition remains a major implementation issue in subspace-based methods [21].

ESPRIT provides a related but different subspace-based trade-off. Jung et al. present a scalable FPGA ESPRIT processor for FMCW radar, supporting a variable number of antennas and separating the covariance, eigenvalue decomposition, least-squares, and angle-estimation stages [19]. Their work is relevant because it identifies the EVD stage as a major implementation cost and reduces execution time by using a simplified cyclic Jacobi structure. At the same time, not all assumptions in that work directly map to the receiver used in this thesis. In particular, the radar-oriented processing model and scalable antenna configuration differ from the fixed four-channel RFSoC 4x2 frame structure used here. The work is therefore used as an implementation reference for ESPRIT, but not as a directly comparable performance baseline.

The subspace literature shows that MUSIC and ESPRIT are suitable reference algorithms for FPGA-based DoA estimation, especially when angular resolution is important. However, the same literature also shows that their practical cost is strongly affected by covariance estimation, eigenvalue decomposition, and, for MUSIC, angular spectrum evaluation [19], [21], [26]. These stages influence latency, resource usage, and implementation complexity, making them important for an RFSoC/HLS comparison where accuracy and hardware cost must be evaluated together.

B. Decomposition-Based Alternatives

Because matrix operations are a major cost in subspace DoA estimation, several hardware-oriented methods try to reduce or restructure the decomposition stage. Hussain et al. present an FPGA implementation of DoA estimation algorithms employing LU decomposition [26]. Their work implements both lower- and upper-triangular LU-based forms on a Virtex-5 FPGA and validates the methods using an NI PXI receiver platform with a four-element ULA. The reported comparison against QR-based alternatives is relevant here because it treats LU factorization as a lower-complexity alternative to more expensive matrix decomposition approaches.

More recent decomposition-based work also shows the continued interest in reducing matrix-processing cost. Tayem et al. use LU decomposition in a real-time 2D DoA estimation method for an L-shaped array, reducing computational complexity compared to EVD- or SVD-based approaches [27]. Although the L-shaped 2D array model does not match the four-element ULA used in this thesis, the work supports the broader motivation to consider LU-based methods for real-time, hardware-oriented DoA estimation.

Decomposition-based approaches are therefore useful in this thesis because they directly target one of the main hardware bottlenecks in subspace-based DoA estimation. Among these methods, LU-DOA provides a suitable comparison point because it is covariance-based, has a prior FPGA-oriented implementation, and is compatible with the four-element ULA model used in this work.

C. Phase-Based and Beamforming Methods

Lower-complexity FPGA DoA approaches have also been explored through phase-based and beamforming methods. Phase-interferometry designs estimate angle from measured phase differences and are often attractive for hardware because they can be organized around regular phase-processing structures [13], [14]. Beamforming-based methods instead evaluate the array response over candidate directions. Recent digital beamforming work shows that reconfigurable scan-based FPGA architectures can support real-time DoA estimation with adjustable angular resolution [24]. Bartlett-type estimators provide another hardware-friendly baseline because the computation can be arranged as repeated steering-vector evaluations over an angular grid [15].

Adaptive beamforming methods are also relevant to this category. A related real-valued Capon-like approach was considered in the algorithm selection process [22]. That method is relevant because it aims to reduce computational cost by avoiding EVD and source-number estimation. However, its array assumptions were not aligned with the final receiver setup used in this thesis. In particular, the method relies on a different array spacing from that of the approximately half-wavelength four-element ULA used by the RFSoC 4x2 receiver. Therefore, it was not included in the final implementation comparison.

These methods are an important context for embedded DoA estimation because they can offer lower implementation

complexity than subspace-based methods. However, they represent a different estimator class from the three covariance-based methods reproduced in this thesis. Phase-interferometry, beamforming, and Capon-like methods do not require the same estimator structure as MUSIC, ESPRIT, and LU-DOA, and including them would broaden the study into a comparison of different receiver-processing approaches. For this reason, they are treated as relevant lower-complexity references rather than as part of the final implemented comparison.

D. Summary of Selected Algorithms

MUSIC, ESPRIT, and LU-DOA were selected because they represent three FPGA-oriented trade-offs within a covariance-based DoA framework. MUSIC retains EVD and angular search, serving as the accuracy-oriented reference. ESPRIT removes the spectral search but still depends on EVD-based subspace processing. LU-DOA avoids the full EVD in the implemented single-source path, providing a lower-complexity comparison point. The selection is therefore not intended to cover all possible DoA estimators, but to compare three reproducible methods under the same RFSoC/HLS receiver assumptions. The detailed implementation of the selected kernels is described in the next section.

IV. Implementation

The selected DoA estimators are implemented within a common Vitis HLS framework to reduce implementation-dependent differences between the algorithms. This section describes the RFSoC data path, the shared snapshot and covariance interface, and the algorithm-specific processing stages.

A. Implementation Overview

MUSIC, ESPRIT, and LU-DOA are implemented as three separate Vitis HLS 2022.2 top-level kernels targeting the RFSoC 4x2 programmable logic. Each kernel uses the same project structure, top-level streaming interface style, and synthesis workflow. In the integrated block design, the three kernels operate in parallel on the same input frame, producing separate DoA estimates for software readout. The DoA datapath is intended to run in the RFSoC fabric, while the processing system is used to configure the design and read back completed estimates.

Figure 1 summarizes the implemented RFSoC/HLS processing flow.

The complete receiver and DoA processing chain operates at 125 MHz, beginning at the RFSoC receiver interface and continuing through the exported DoA IP cores. Separate bitstreams were generated for the filtered and unfiltered receiver paths, as well as for each evaluated internal fixed-point format. This results in eight implemented configurations: two input-conditioning choices and four internal kernel word lengths.

B. RFSoC Data Path and Snapshot Interface

The kernels are intended to operate on live streamed RFSoC receiver data. The RF Data Converter provides complex I/Q

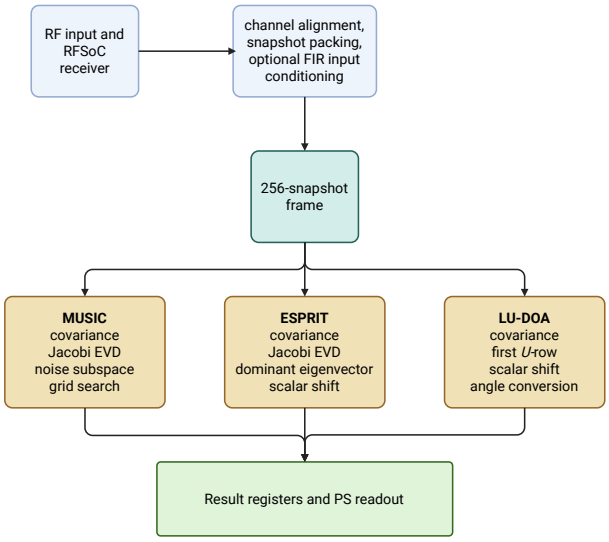


Fig. 1. Block-level overview of the implemented RFSoC/HLS DoA processing chain. The RFSoC receiver path aligns and repacks the four-channel input stream, optionally applies FIR input conditioning, and forms 256-snapshot frames. The same frame is provided to the MUSIC, ESPRIT, and LU-DOA kernels, which run in parallel and produce q16.8 broadside DoA estimates for register-based software readout.

samples from the four receive channels, and channel phase alignment is handled using the RFSoC multi-tile synchronization flow [28]. The receiver configuration uses a 4 GS/s ADC with 4× decimation, producing an effective 1 GS/s sample stream before the DoA-specific repacking. The RFDC stream provides multiple samples per AXI clock cycle, and a preprocessor converts the receiver stream into the common four-channel snapshot format expected by the DoA kernels.

Each estimator receives complex antenna snapshots in frames, using the four-channel snapshot definition in (2). The streaming input format is fixed across all evaluated configurations. Each snapshot contains four complex antenna samples, and both the real and imaginary parts are represented using q18.8 fixed-point values. The evaluated q16.8, q18.8, q20.8, and q24.8 formats therefore refer to the internal arithmetic used inside the DoA kernels, not to changes in the RFSoC snapshot interface.

All three kernels use 256 snapshots per DoA estimate. The input stream packs the four complex antenna samples into a common snapshot format, and the output stream produces one broadside DoA estimate $\hat{\theta}$ for each processed frame. The output angle is represented as a q16.8 fixed-point value in degrees for all three kernels. Frame identifiers are propagated through the stream sideband information so that estimates can be associated with the corresponding input frame. The result streams are also connected to a small register block that stores the most recent MUSIC, ESPRIT, and LU-DOA angles, frame identifiers, and update counters for software readout.

To evaluate the effect of input conditioning, a hardware low-pass filter was added before the DoA preprocessing stage

for the filtered-bitstream variants. The filter was implemented using AMD's FIR Compiler IP block [29]. The same filter was applied independently to each of the eight real-valued receiver streams, corresponding to the I and Q components of the four antenna channels. The filter coefficients were generated in MATLAB for a 20 MHz low-pass bandwidth, using a 1 GS/s sample rate, a 10 MHz passband edge, a 30 MHz stopband edge, and 70 dB stopband attenuation, with 18-bit quantized coefficients. The resulting filter is a 219-tap symmetric linear-phase low-pass filter with approximately unity DC gain.

C. Common Covariance and Fixed-Point Processing

All three selected algorithms operate on the same sample covariance definition in (5). The covariance code is implemented as a common source-level function and reused by MUSIC, ESPRIT, and LU-DOA. Each HLS top-level kernel calls this function inside its own processing pipeline. Therefore, the covariance stage is common to both the algorithm and the source structure, but it is not a single shared hardware block that feeds all three kernels. In the integrated design, each parallel kernel contains its own instantiated covariance logic.

The covariance function accumulates the antenna-pair products for the four-channel array over one frame and normalizes the result by the frame length. Since covariance estimation is a common prerequisite for the selected subspace and decomposition-based estimators, using the same covariance implementation improves the fairness of the comparison. Differences after this stage are therefore more directly associated with the decomposition, subspace extraction, search, and angle-estimation steps of each algorithm.

The covariance matrix is treated as a small fixed-size matrix. This allows loops over antenna indices to be unrolled or partially unrolled, and it allows small arrays to be partitioned so that independent products can be evaluated in parallel. Snapshot ingestion is pipelined so that the covariance stage can efficiently accept the streaming frame. These implementation choices are common to all three kernels.

The HLS designs use fixed-point arithmetic rather than floating-point arithmetic. Complex values are represented by separate fixed-point real and imaginary components, and wider intermediate values are used where required by accumulation or matrix operations. The four evaluated internal kernel formats are q16.8, q18.8, q20.8, and q24.8, as introduced in Section II. The designs use truncation and wrap-around fixed-point behavior, with no additional rounding, saturation, or scaling optimizations.

D. MUSIC Kernel

The MUSIC kernel follows the subspace-and-search structure motivated in Section III. After covariance estimation, an iterative Jacobi eigenvalue decomposition is used to obtain the eigenvectors of $\hat{\mathbf{R}}_{xx}$. For the single-source case, the eigenvector associated with the largest eigenvalue is treated as the signal subspace, and the remaining eigenvectors form the noise subspace, denoted by $\mathbf{E}_n$. The MUSIC estimate is then computed as a search over the common angular grid Θ,

$$\hat{\theta}_{\mathrm{MUSIC}} = \arg\min_{\theta\in\Theta} \left\|\mathbf{E}_n^H \mathbf{a}(\theta)\right\|_2^2. \tag{6}$$

This is equivalent to finding the peak of the MUSIC pseudospectrum, but the denominator form is more directly connected to the hardware implementation because the reciprocal does not need to be explicitly computed. The implemented grid covers -90° to 90° with one-degree resolution. The steering vectors are stored in a precomputed table, so the kernel evaluates the denominator for each grid angle and selects the minimum value.

E. ESPRIT Kernel

The ESPRIT kernel keeps the covariance and EVD stages but removes the angular spectrum search. After covariance estimation, the kernel performs a Jacobi eigenvalue decomposition and extracts the dominant eigenvector as the single-source signal subspace. Adjacent ULA subarrays are then formed from this eigenvector, and their shift-invariance relation is reduced to a scalar phase estimate.

In the implemented single-source form, the shift estimate is computed from adjacent entries of the dominant eigenvector rather than by explicitly solving a general least-squares matrix problem. If the dominant eigenvector is denoted by $\mathbf{e}_s$, the scalar relation is accumulated as

$$\psi_{\mathrm{ESPRIT}} = \sum_{i=0}^{2} e_s^*[i] e_s[i+1]. \tag{7}$$

The phase of ψ_{ESPRIT} is then converted to the common broadside angle convention before output.

F. LU-DOA Kernel

The LU-DOA kernel is implemented as a lower-complexity alternative to the EVD-based subspace methods. It follows the U-based LU-DOA formulation, where a signal-space representation is extracted from the covariance matrix and then used in a shift-invariance relation [26]. In the general formulation, the covariance matrix is factorized as

$$\hat{\mathbf{R}}_{xx} = \mathbf{L}\mathbf{U}, \tag{8}$$

where $\mathbf{L}$ is lower triangular and $\mathbf{U}$ is upper triangular. For the single-source case used in the evaluated implementation, the required U-based representation reduces to one row vector. Under the Doolittle LU structure, the first row of $\mathbf{U}$ is equal to the first row of $\hat{\mathbf{R}}_{xx}$. The implemented kernel, therefore, extracts this row directly and does not compute an EVD or a full LU factorization along the single-source path.

The rotational factor is estimated from adjacent entries of the extracted row using the scalar relation implemented in the HLS kernel. If this row is denoted by $\mathbf{u}_0$, the estimate is computed as

$$\psi_{\mathrm{LU}} = \frac{\sum_{i=0}^{2} u_0[i] u_0^*[i+1]}{\sum_{i=0}^{2} u_0[i] u_0^*[i]}. \tag{9}$$

The phase of this scalar estimate is then converted to the common broadside-angle convention before being output.

G. Validation Method

Captured binary playback was used for repeatable HLS testbench validation. Stored complex snapshots were read in the antenna-major order expected by the MUSIC, ESPRIT, and LU-DOA testbenches, and C simulation and C/RTL cosimulation outputs were compared against Python reference calculations before hardware integration. This playback path was used only for functional debugging and did not replace the streamed RFSoC datapath used for physical measurements.

V. Evaluation

This section reports the measurement setup and evaluates MUSIC, ESPRIT, and LU-DOA in terms of captured-data accuracy, HLS kernel timing, and post-implementation FPGA resource usage. The combined interpretation of these results is left for Section VI.

A. Measurement Setup and Captured Data

The RF measurements were collected in an indoor laboratory environment. The room was approximately $7\,\mathrm{m}$ by $10\,\mathrm{m}$, and the transmitter and receiver were placed with a clear line-of-sight path between them. The approximate room layout, measurement geometry, and physical experiment setup are shown in Figure 2. No additional nearby reflectors were intentionally placed around the measurement geometry, although normal laboratory objects such as tables and other equipment remained in the environment.

The transmitter was an Ettus USRP transmitting a narrowband sinusoid centered at 2.480 GHz, generated at 1 MS/s with a 200 kHz waveform frequency and 30.5 dB transmit gain. The receiver was the RFSoC 4x2-based four-channel ULA described earlier, placed approximately 4 m from the transmitter. The transmitter remained fixed, while the receiver antenna spacer was rotated to set each reference angle. The same setup and captured data were used for all three algorithms.

The reference angle was set using a level vertical laser line as the physical alignment reference. The laser line passed through the protractor at the center of the antenna spacer and was also aligned with the transmitter antenna. For each new reference angle, the receiver antenna spacer was rotated and then realigned using the laser line and protractor. This procedure fixed the reference angle with respect to the broadside convention used in Section II.

The evaluated reference angles were $-60°$, $-45°$, $-30°$, $-15°$, $0°$, $15°$, $30°$, $45°$, and $60°$. The manual alignment uncertainty was not quantified experimentally but was estimated at approximately $\pm 1°$. All algorithm outputs were converted to the common broadside angle convention before the angular error was calculated.

The accuracy evaluation used captured RF measurement data from both the filtered and unfiltered receiver branches. The filtered branch applied the FIR-based input-conditioning stage described in Section IV. The unfiltered branch bypassed this stage before the DoA preprocessor. The main accuracy results combine the filtered and unfiltered captures, while branch-specific results are used only where the effect of input conditioning is considered.

TABLE I
Central- and Large-Angle Captured-Data Accuracy. Errors are in degrees.

	$\lvert\theta\rvert < 45°$			$\lvert\theta\rvert \geq 45°$		
Algorithm	MAE	P99	RMSE	MAE	P99	RMSE
MUSIC	0.87	3.00	1.11	5.12	23.00	10.06
ESPRIT	0.97	3.36	1.23	5.85	101.29	14.98
LU-DOA	1.07	4.06	1.37	17.21	117.55	32.20

The four fixed-point formats defined in Section II were evaluated for all algorithms. The captured dataset contained 145 capture runs and 3,589,597 evaluated output rows. The row counts differed slightly between algorithms because incomplete captures were retained and contributed only their available samples. MUSIC contributed 1,185,057 rows, ESPRIT contributed 1,201,887 rows, and LU-DOA contributed 1,202,653 rows. No evaluated angle, filtering condition, or fixed-point format had too few rows to be included in the comparison.

B. Accuracy Results

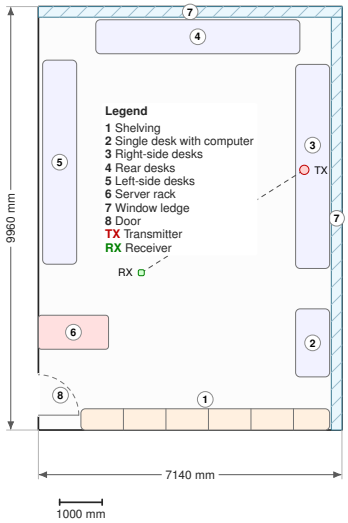
Accuracy was evaluated using the angular error definition in Section II. RMSE was used as the main failure-sensitive metric as it weights larger angular errors more strongly. Mean absolute error was used to describe the average absolute deviation, median absolute error was used to describe typical behavior, and the 99th percentile absolute error was used to describe the error tail.

The results were also separated into central-angle and large-angle subsets. Central angles were defined as $|\theta| < 45°$, which includes $-30°$, $-15°$, $0°$, $15°$, and $30°$. Large angles were defined as $|\theta| \geq 45°$, which includes $-60°$, $-45°$, $45°$, and $60°$. This separation is used because the large-angle cases produced the dominant error tails in the captured data.

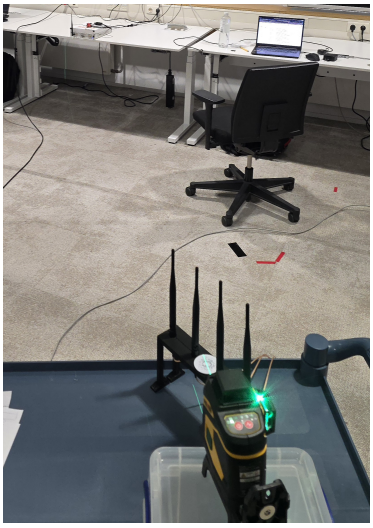
1) Central- and Large-Angle Accuracy: Table I separates the captured-data accuracy into central- and large-angle subsets. Across the combined captures, MUSIC had the lowest overall RMSE and P99 absolute error, followed by ESPRIT and LU-DOA: $6.85°$ and $15.00°$, $10.17°$ and $23.10°$, and $21.62°$ and $82.47°$, respectively. The central-angle results are similar across algorithms, whereas the large-angle subset shows the main robustness difference.

Figure 3(a) shows RMSE as a function of reference angle. The largest errors occur near the edges of the evaluated angular range, while the central-angle cases remain close across the three algorithms.

2) Fixed-Point Accuracy: Figure 3(b) compares central-angle RMSE across the four fixed-point formats. Increasing word length did not yield a monotonic improvement for all algorithms: ESPRIT and LU-DOA improved most clearly, while MUSIC had its lowest central-angle RMSE at q20.8. This indicates that q20.8 was not intrinsically inaccurate; its less favorable full-range behavior came from large-angle outliers.

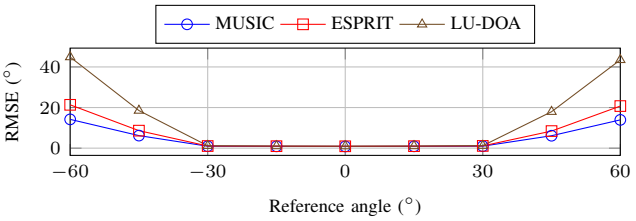


(a) Laboratory room layout and approximate TX/RX placement.

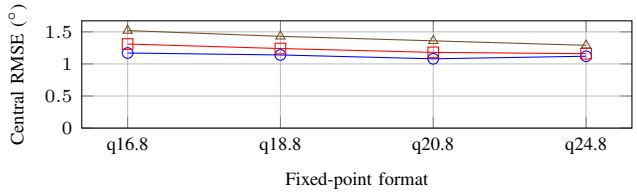


(b) Physical experimental setup.

Fig. 2. Indoor RF measurement setup. Panel (a) shows the approximate room layout, dimensions, and TX/RX placement. Panel (b) shows the physical setup; the vertical guideline is partly obscured by lighting.



(a) RMSE as a function of reference angle.



(b) Central-angle RMSE for the evaluated fixed-point formats.

Fig. 3. Captured-data RMSE trends for MUSIC, ESPRIT, and LU-DOA. Panel (a) shows the RMSE by reference angle, while panel (b) shows the central-angle RMSE for q16.8, q18.8, q20.8, and q24.8.

3) Effect of Input Conditioning: Table II compares the filtered and unfiltered receiver branches. Filtering reduced the MAE, P99 absolute error, and RMSE for MUSIC and ESPRIT, but increased the MAE and RMSE for LU-DOA. The FIR path is therefore treated as an estimator-dependent receiver-level trade-off rather than as a general estimator improvement.

C. HLS Timing Results

Algorithm timing was evaluated using Vitis HLS synthesis results with an 8 ns target clock period, corresponding to the 125 MHz RFSoC receiver pipeline. Maximum latency and initiation interval were used for comparison, and cycle counts were converted using the common clock period. These results describe DoA kernel behavior rather than full end-to-end receiver latency.

TABLE II
FILTERED AND UNFILTERED CAPTURED-DATA ACCURACY. ERRORS ARE IN DEGREES.

Algorithm	Branch	MAE	Median AE	P99 AE	RMSE
MUSIC	Filtered	2.13	1.00	9.00	4.32
	Unfiltered	3.46	1.00	21.00	8.59
ESPRIT	Filtered	2.75	1.37	12.38	7.23
	Unfiltered	3.63	1.41	70.18	12.35
LU-DOA	Filtered	12.69	1.82	74.41	27.53
	Unfiltered	4.06	1.48	83.94	13.52

1) Kernel Latency and Initiation Interval: Table III reports the HLS timing results for each algorithm and fixed-point format. All synthesized kernels met the 8 ns target clock period, with estimated clock periods around 5.8 ns. MUSIC and ESPRIT had almost identical latency and initiation interval across all fixed-point formats, consistent with their shared covariance and EVD-based processing structure. LU-DOA had substantially lower latency and a shorter initiation interval across all evaluated formats.

The latency and initiation interval were nearly equal across kernel configurations, so the reported interval primarily determines the kernel update rate. MUSIC and ESPRIT produced approximate update rates from 23.6 kHz to 46.2 kHz, depending on fixed-point format. LU-DOA produced approximate update rates from 229 kHz to 230 kHz.

D. FPGA Resource Usage

FPGA resource usage was evaluated using Vivado post-implementation results. The reported values include the implemented DoA kernels and exclude shared receiver-path resources, so the comparison reflects MUSIC, ESPRIT, and LU-DOA rather than the complete board design. Absolute LUT, FF, DSP, and BRAM counts are reported.

TABLE III
VITIS HLS KERNEL TIMING RESULTS. LATENCY TIME IS COMPUTED USING THE 125 MHZ SYSTEM CLOCK.

Algorithm	Format	Clock est. (ns)	Latency (cycles)	Latency (μs)	II (cycles)
MUSIC	q16.8	5.834	4966	39.728	4967
	q18.8	5.797	5302	42.416	5303
	q20.8	5.822	2707	21.656	2708
	q24.8	5.815	3475	27.800	3476
ESPRIT	q16.8	5.834	4967	39.736	4968
	q18.8	5.835	5303	42.424	5304
	q20.8	5.827	2708	21.664	2709
	q24.8	5.827	3476	27.808	3477
LU-DOA	q16.8	5.827	545	4.360	546
	q18.8	5.835	545	4.360	546
	q20.8	5.827	542	4.336	543
	q24.8	5.839	542	4.336	543

TABLE IV
VIVADO POST-IMPLEMENTATION DOA KERNEL RESOURCE USAGE. SHARED RECEIVER-PATH RESOURCES ARE EXCLUDED.

Algorithm	Format	LUT	FF	DSP	BRAM36	BRAM18
MUSIC	q16.8	19609	21034	141	2	14
	q18.8	26042	25667	141	2	14
	q20.8	42616	25908	243	10	6
	q24.8	42747	37830	341	10	6
ESPRIT	q16.8	25172	23143	108	5	0
	q18.8	29649	27625	108	5	0
	q20.8	46028	27599	198	5	0
	q24.8	49239	40446	260	5	0
LU-DOA	q16.8	8640	7397	51	5	0
	q18.8	9587	8696	51	5	0
	q20.8	12587	9552	51	5	0
	q24.8	12107	10546	99	5	0

The FIR filters in the filtered receiver branch are excluded from Table IV because they belong to the input-conditioning path rather than to the DoA kernels. The eight FIR Compiler instances together used 34,640 LUTs, 243,744 FFs, and 880 DSPs. Since Table II shows estimator-dependent accuracy effects, this cost is treated as a receiver-level trade-off rather than part of the estimator resource ranking.

1) Implemented Kernel Resources: Table IV reports the implemented resource usage for each DoA kernel and fixed-point format. LU-DOA used the fewest LUTs, FFs, and DSPs across all evaluated formats. ESPRIT generally used the most LUTs and FFs, while MUSIC generally used the most DSPs. This is consistent with the algorithm structures described earlier: MUSIC and ESPRIT retain the EVD-oriented processing stages, while LU-DOA avoids the full EVD in the implemented single-source case.

VI. DISCUSSION & FUTURE WORK

This section interprets the measured accuracy, latency, update-rate, and resource trade-offs under the shared RFSoC/HLS conditions.

A. Interpretation of the Results

The central-angle results show that all three algorithms can provide similar accuracy near broadside, while the full-range metrics expose failure-sensitive behavior. As shown in Figure 3(a), the main accuracy differences appear at larger steering angles, especially near the edge of the evaluated range.

The large-angle results indicate different robustness properties rather than a uniform accuracy gap. MUSIC is less affected at the angular extremes, while ESPRIT and LU-DOA degrade more strongly. This is consistent with phase-derived shift relations, where small phase perturbations can produce larger angular changes after conversion to the broadside convention. In the captured data, the worst cases often appeared as sign-mirror estimates near the angular extremes, increasing the tail metrics while leaving the median errors comparatively low.

The fixed-point sweep shows that the observed ordering is not due to any single numerical format, while the filtered and unfiltered captures show that input conditioning is an estimator-dependent receiver-level trade-off.

B. Limitations and Future Work

The comparison is limited to three implemented DoA algorithms, a four-element ULA, a single dominant source, and controlled indoor measurements. These constraints match the RFSoC 4x2 setup used in this work, but limit generalization to larger arrays, other geometries, multi-source channels, and outdoor signals of opportunity. The timing results also describe kernel-level behavior rather than full end-to-end navigation latency.

Future work should evaluate larger arrays, dedicated phase and gain calibration, multi-source scenarios, and outdoor or less controlled SoOp measurements. Additional HLS optimization variants could help separate algorithmic cost from implementation choices, while integration into a complete passive-positioning pipeline would show how the measured estimator-level trade-offs affect navigation-level performance.

VII. RESPONSIBLE RESEARCH

Although the focus of this thesis is hardware implementation and evaluation, the investigated RF sensing techniques may also be applicable in systems with dual-use potential. This thesis does not implement a complete navigation, tracking, or surveillance system. The evaluation is limited to the DoA estimation stage under a controlled single-source laboratory setup, using non-confidential signals and a common receiver model. The results should therefore be interpreted as a hardware-oriented comparison of estimation methods rather than as an operational deployment of an autonomous RF sensing system.

Reproducibility is supported by defining a common model and evaluation procedure before comparing the algorithms. The array geometry, angle convention, signal assumptions, snapshot structure, covariance estimate, and error metrics are defined in Section II. The implementations use the same RFSoC 4x2 receiver assumptions, the same Vitis HLS 2022.2 and Vivado 2022.2 workflow, the same streaming format, and the same shared covariance stage described in Section IV.

Generative AI tools were used as supporting tools during the research and writing process. The tools included ChatGPT, Claude, and Grammarly. They were mainly used to support writing flow, grammar, and structure; debugging; developing MATLAB and Python code; and generating plot scripts. They were not used as sources of technical evidence or as a replacement for implementation, analysis, or interpretation. AI-assisted text was reviewed and rewritten, and AI-assisted code was manually checked and tested before being used for implementation support or result processing. Thesis text, code, results, and figures were entered into these tools only when no confidential, proprietary, personal, or sensitive information was involved. The final technical claims, results, code, and conclusions remain the responsibility of the author.

VIII. Conclusion

This work compared MUSIC, ESPRIT, and LU-DOA for FPGA-oriented DoA estimation on an RFSoC 4x2 platform. A common RFSoC/HLS framework was used to isolate algorithm-level trade-offs in angular accuracy, latency, update rate, and FPGA resource usage.

The results show a clear trade-off. MUSIC provides the strongest full-range angular robustness, especially at larger steering angles. ESPRIT removes the angular search but remains close to MUSIC in timing and resource usage because both retain EVD-based processing. LU-DOA gives the lowest latency, shortest initiation interval, and lowest resource usage, but its large-angle error tails limit full-range robustness. Thus, MUSIC is preferable when robustness dominates, while LU-DOA is preferable when latency, update rate, and resources are the main constraints.

References

[1] Y. J. Morton, F. v. Diggelen, J. J. S. Jr, B. W. Parkinson, S. Lo, and G. Gao, *Position, Navigation, and Timing Technologies in the 21st Century: Integrated Satellite Navigation, Sensor Systems, and Civil Applications, Volume 2*. John Wiley & Sons, Dec. 2020, google-Books-ID: PNoPEAAAQBAJ.

[2] Z. M. Kassas, J. Khalife, A. A. Abdallah, and C. Lee, "I Am Not Afraid of the GPS Jammer: Resilient Navigation Via Signals of Opportunity in GPS-Denied Environments," *IEEE Aerospace and Electronic Systems Magazine*, vol. 37, no. 7, pp. 4–19, Jul. 2022.

[3] S. Yeom, "Vision-Only Localization of Drones with Optimal Window Velocity Fusion," *Electronics*, vol. 15, no. 3, p. 637, Feb. 2026. [Online]. Available: https://www.mdpi.com/2079-9292/15/3/637

[4] S. Alghamdi, S. Alahmari, S. Yonbawi, K. Alsaleem, F. Ateeq, and F. Almushir, "Autonomous Navigation Systems in GPS-Denied Environments: A Review of Techniques and Applications," in *2025 11th International Conference on Automation, Robotics, and Applications (ICARA)*, 2025.

[5] N. D. Almalis, G. A. Tsihrintzis, M. Mitsios, and T. Tilemachos, "Learning to Navigate and Track: A Decentralized RL Framework for UAV Operations in GNSS-Denied Terrain," in *Artificial Intelligence XLII*, M. Bramer and F. Stahl, Eds. Cham: Springer Nature Switzerland, 2026, pp. 300–307.

[6] M. J. Wright, L. Anastassiou, C. Mishra, J. M. Davies, A. M. Phillips, S. Maskell, and J. F. Ralph, "Cold atom inertial sensors for navigation applications," *Frontiers in Physics*, vol. 10, Oct. 2022. [Online]. Available: https://www.frontiersin.org/journals/physics/articles/10.3389/fphy.2022.994459/full

[7] A. Shurin and I. Klein, "QDR: A Quadrotor Dead Reckoning Framework," *IEEE Access*, vol. 8, pp. 204 433–204 440, 2020. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9256293

[8] Z. Kassas, J. Morales, and J. Khalife, "New-Age Satellite-Based Navigation – STAN: Simultaneous Tracking and Navigation with LEO Satellite Signals," *Inside GNSS Magazine*, vol. 14, no. 4, pp. 56–65, Aug. 2019. [Online]. Available: https://escholarship.org/uc/item/2cf7c651

[9] H. Benzerrouk, Q. Nguyen, F. Xiaoxing, A. Amrhar, A. V. Nebylov, and R. Landry, "Alternative PNT based on Iridium Next LEO Satellites Doppler/INS Integrated Navigation System," in *2019 26th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS)*, May 2019, pp. 1–10. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8769440

[10] C. Yang, T. Nguyen, and E. Blasch, "Mobile positioning via fusion of mixed signals of opportunity," *IEEE Aerospace and Electronic Systems Magazine*, vol. 29, no. 4, pp. 34–46, Apr. 2014. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6819122

[11] J. A. del Peral-Rosado, R. Raulefs, J. A. López-Salcedo, and G. Seco-Granados, "Survey of Cellular Mobile Radio Localization Methods: From 1G to 5G," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 2, pp. 1124–1148, 2018. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8226757

[12] H.-C. Chen, T.-H. Lin, H. T. Kung, C.-K. Lin, and Y. Gwon, "Determining RF angle of arrival using COTS antenna arrays: A field evaluation," in *MILCOM 2012 - 2012 IEEE Military Communications Conference*, Oct. 2012, pp. 1–6, iSSN: 2155-7586. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6415851

[13] M. N. E. I. Rouabhia, A. Bellabas, M. L. Bencheikh, and M. E. M. Abdelaziz, "On the phase interferometry direction finding: Performance comparison and FPGA implementations," in *2017 Seminar on Detection Systems Architectures and Technologies (DAT)*, Feb. 2017, pp. 1–5. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7889153

[14] A. Florio, G. Avitabile, C. Talarico, and G. Coviello, "A Reconfigurable Full-Digital Architecture for Angle of Arrival Estimation," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 71, no. 3, pp. 1443–1455, Mar. 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10376051

[15] F. M. Unlersen, E. Yaldiz, and S. T. Imeci, "FPGA Based Fast Bartlett DoA Estimator for ULA Antenna Using Parallel Computing," *Applied Computational Electromagnetics Society Journal (ACES)*, pp. 450–459, 2018. [Online]. Available: https://journals.riverpublishers.com/index.php/ACES/

[16] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, Mar. 1986. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/1143830/references

[17] R. Roy, A. Paulraj, and T. Kailath, "Estimation of Signal Parameters via Rotational Invariance Techniques - ESPRIT," in *MILCOM 1986 - IEEE Military Communications Conference: Communications-Computers: Teamed for the 90's*, vol. 3, Oct. 1986, pp. 41.6.1–41.6.5. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/4805850

[18] U. M. Butt, S. A. Khan, A. Ullah, A. Khaliq, P. Reviriego, and A. Zahir, "Towards Low Latency and Resource-Efficient FPGA Implementations of the MUSIC Algorithm for Direction of Arrival Estimation," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 8, pp. 3351–3362, Aug. 2021. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9445729

[19] Y. Jung, H. Jeon, S. Lee, and Y. Jung, "Scalable ESPRIT Processor for Direction-of-Arrival Estimation of Frequency Modulated Continuous Wave Radar," *Electronics*, vol. 10, no. 6, p. 695, Jan. 2021. [Online]. Available: https://www.mdpi.com/2079-9292/10/6/695

[20] 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 Transactions on Circuits and Systems I: Regular Papers*, vol. 69, no. 7, pp. 2941–2954, Jul. 2022. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9745799

[21] 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 Sensing*, vol. 16, no. 20, 2024.

[22] Y. Chen, Z. Fu, and J. Li, "A Priori-Knowledge-Free Real-Valued Capon-Like Method and Implementation on FPGA," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 71, no. 12, pp. 6532–6543, Dec. 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10598232

[23] J. Li, G. Yang, X. Ren, Z. Fu, and S. Qiu, "Modified ESPRIT and Unitary-Transformation Form for Computationally Efficient DOA Estimation on FPGA," *IEEE Transactions on Instrumentation and Measurement*, vol. 74, pp. 1–9, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10902624

[24] Y. Wang, L. Zhu, T. Wu, and S. Ma, "A Reconfigurable Digital Beamformer Implemented on a Field-Programmable Gate Array for Real-Time and Resource-Efficient Direction-of-Arrival Estimation," *Sensors*, vol. 25, no. 8, p. 2497, Jan. 2025. [Online]. Available: https://www.mdpi.com/1424-8220/25/8/2497

[25] 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 *2024 27th International Symposium on Design & Diagnostics of Electronic Circuits & Systems (DDECS)*, Apr. 2024, pp. 142–147, iSSN: 2473-2117. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10508912

[26] 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. 17 666–17 680, 2018. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8327576

[27] N. Tayem, A. Burns, K. Hansen, S. Pearson, N. Blount, D. Reyes, and A. A. Hussain, "Hardware Implementation of Real-Time 2D DOA Estimation Algorithm Using an L-Shaped Array," in *2024 IEEE 15th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, Oct. 2024, pp. 0656–0661. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10754692

[28] Xilinx, "RFSoC-MTS: A PYNQ Overlay Demonstrating AMD RFSoC Multi-Tile Synchronization (MTS)." [Online]. Available: https://github.com/Xilinx/RFSoC-MTS

[29] "FIR Compiler." [Online]. Available: https://www.amd.com/en/products/adaptive-socs-and-fpgas/intellectual-property/fir_compiler.html