IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 11, 2018, accepted March 22, 2018, date of publication March 28, 2018, date of current version April 23, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2820122

FPGA Hardware Implementation of DOA Estimation Algorithm Employing LU Decomposition

AHMED A. HUSSAIN[1], NIZAR TAYEM[1], (Member, IEEE), MUHAMMAD OMAIR BUTT[1], ABDEL-HAMID SOLIMAN[2], ABDULRAHMAN ALHAMED[3], AND SALEH ALSHEBEILI[3]

[1]Department of Electrical Engineering, Prince Mohammad Bin Fahd University, Alkhobar 31952, Saudi Arabia
[2]School of Engineering, Staffordshire University, Stoke-on-Trent, ST4 2DE, U.K.
[3]KACST-TIC in RF and Photonics for the e-Society (RFTONICS), Department of Electrical Engineering, King Saud University, Riyadh 11421, Saudi Arabia

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

This work was supported by the Deanship of Scientific Research at King Saud University through the Research Group under Grant RG-1438-092.

ABSTRACT In this paper, authors present their work on field-programmable gate array (FPGA) hardware implementation of proposed direction of arrival estimation algorithms employing LU factorization. Both L and U matrices were considered in computing the angle estimates. Hardware implementation was done on a Virtex-5 FPGA and its experimental verification was performed using National Instruments PXI platform which provides hardware modules for data acquisition, RF down-conversion, digitization, etc. A uniform linear array consisting of four antenna elements was deployed at the receiver. LabVIEW FPGA modules with high throughput math functions were used for implementing the proposed algorithms. MATLAB simulations of the proposed algorithms were also performed to validate the efficacy of the proposed algorithms prior to hardware implementation of the same. Both MATLAB simulation and experimental verification establish the superiority of the proposed methods over existing methods reported in the literature, such as QR decomposition-based implementations. FPGA compilation results report low resource usage and faster computation time compared with the QR-based hardware implementation. Performance comparison in terms of estimation accuracy, percentage resource utilization, and processing time is also presented for different data and matrix sizes.

INDEX TERMS FPGAs, LU factorization, NI PXI platform, pipelined architecture.

I. INTRODUCTION

With the rapid advances in the different fields of communication technologies, DOA estimation finds important practical applications in areas such as channel estimation and equalization, echo and interference cancellation, source localization in radar and sonar systems, beam forming 'smart' adaptive antenna arrays in wireless mobile communications systems, and multiple-input-multiple-output (MIMO) systems [1]–[4]. Majority of the research work reported in these areas has focused primarily on numerical simulations of the algorithms for DOA estimation to establish their accuracy and efficacy [5]–[11]. However, due to the practical significance of these problems, these algorithms are required to be implemented and tested on real hardware to validate their viability in terms of computational speed, memory requirements, and implementation cost in hardware. In addition, most applications require the DOA estimates to be computed in real-time (with computation speeds of the order of a few microseconds or even nanoseconds) such as in tracking a very fast moving target using a radar or sonar.

The performance of a DOA algorithm is determined by several factors such as the size, number of elements and spacing of the antenna array as well as different configurations of impinging signals. Many DOA techniques exist [5]–[10], which are based on analysis of covariance matrix using Eigen Value Decomposition (EVD) or analysis of received data matrix using Singular Value Decomposition (SVD). Both EVD and SVD based algorithms involve separating noise and signal subspaces that can be used to infer angles of arrival of impinging signals.

Matrix operations are at the heart of computations in array signal processing, in general, and DOA estimation, in particular. The complexity of the DOA estimation algorithm is determined by the matrix operations and the size of the matrices involved. The following paragraphs provide an overview of some of the important matrix computation techniques applied to DOA estimation algorithms.

QR decomposition algorithm factorizes a matrix into two matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ as $\boldsymbol{A} = \boldsymbol{QR}$, where $\boldsymbol{Q}$ is orthogonal and $\boldsymbol{R}$ is upper triangular matrix, and the process can be inverted simply by multiplying the two matrices. There are three different methods to calculate $\boldsymbol{R}$ and $\boldsymbol{Q}$ matrices: Gram-Schmidt procedure Givens Rotations, and Householder Reflections, and Modified Gram-Schmidt. QR decomposition is a subspace scheme that applies to data received from multiple antenna array configurations to calculate the signal and noise spaces [12], [13]. Compared with either SVD or EVD, which are widely used in subspace techniques such as ESPRIT and MUSIC [14]–[18], QR is computationally less complex and is less expensive in terms of resource requirements. The QR factorization for $(N \times N)$ requires $O((4N^3/3))$ flops.

LU factorization factors a matrix $\boldsymbol{A}$ as a product of two matrices $\boldsymbol{L}$ and $\boldsymbol{U}$ such that $\boldsymbol{A} = \boldsymbol{LU}$ where $\boldsymbol{L}$ is lower triangular matrix and $\boldsymbol{U}$ is upper triangular matrix. In $\boldsymbol{L}$, diagonal elements are all one (1) and elements located above the diagonal are all zero. In $\boldsymbol{U}$, elements below the diagonal are zero. LU factorization is used for decomposing the data correlation matrix into signal and noise subspaces [19]. The LU factorization has much less complexity compared to QR factorization. LU factorization requires $O(2N^3/3)$ flops, which are half the number of flops required for QR. Low number of flops will reduce the memory storage and the processing time.

For hardware implementation of DOA estimation algorithms, it is important to consider the computational complexity of the algorithm besides its speed and accuracy in calculating the DOA estimates, and the suitability of the chosen hardware platform for real-time implementation in terms of speed, memory requirements, scalability, and development cost.

In [20] and [21], a hardware implementation is presented of novel DOA estimation methods, which are based on QR decomposition. A least squares (LS) approach or a total least squares technique (TLS) is applied and finally EVD of an $L \times L$ matrix is calculated to estimate the DOAs where L is the number of sources. QR schemes are unlike the other existing schemes where EVD is applied on the spectral cross correlation matrix and SVD is applied on the data matrix. In both cases the dimension is M which is the case in most of the real-world applications - the number of antenna elements M is much greater than the number of sources L.

The methods presented in [20-21] were implemented in LabVIEW software and tested on a prototype built using National Instruments (NI PXI) platform. These methods require less computational time compared to well-known DOA methods MUSIC and ESPRIT [10], [16]. The experimental results verified the successful implementation of the proposed DOA estimation methods. However, real-time implementation on a hardware platform such as FPGA (field programmable gate array) was not studied. The following paragraphs will describe why the FPGA platform is suitable for hardware prototyping.

The silicon area consumed (and in turn power and cooling requirements) and execution time of the hardware implementation for complex signal processing algorithms have often been a bottleneck in the practical deployment of these algorithms in modern mobile communication systems [22]. Real-time implementation of sophisticated DOA estimation algorithms is no exception [23], [24]. For example, DOA estimation such as for a smart antenna system requires orders of magnitude of MAC (multiply and accumulate) operations which are beyond the processing capabilities of currently available DSPs (digital signal processors). However, massively parallel computational devices such as FPGAs are well suited for these challenges especially with inherently parallel algorithms such as DOA estimation algorithms. FPGAs employ various reconfigurable processing elements such as Complex Programmable Logic Devices (CPLDs), memory-based Look-Up-Table (LUTs), and high-speed Digital Signal Processing elements (DSPs) that are optimized for implementation of complex signal processing algorithms.

Abusultan *et al.* [25] and LaMeres *et al.* [26] propose two FPGA implementations of Minimum Variance Distortionless Response (MVDR) and Bartlett methods for DOA estimation, one using Xilinx MicroBlaze soft processor and the other using full custom VHDL programming. This work reports several orders of magnitude improvement in performance in terms of computation time and resource utilization for implementation in FPGA hardware, in comparison with software implementation of the said DOA estimation algorithms. For a circular antenna array of size eight (8), the designs were able to estimate DOA in the order of seconds in the case of soft processor and microseconds in the case of optimized VHDL design using Xilinx Virtex-5 FX70 FPGA chip [27], [28].

Alhamed *et al.* [29] present a very recent work on real-time FPGA implementation of DOA estimation algorithms based on QR decomposition. The performance of the proposed algorithms is compared with existing hardware implementations reported in the literature of unitary-MUSIC [30], [31], MUSIC [32] and ESPRIT [33] algorithms. The proposed implementation compares favorably with existing implementations.

In this paper, we propose DOA estimation algorithms based on LU factorization: one method considering partial L matrix (LU-L), and the other considering partial U matrix (LU-U). These methods have been verified through Matlab simulations before being implemented on a Xilinx Virtex-5 FPGA [34] using LabVIEW FPGA high throughput modules [35]. Experimental validation of the proposed DOA estimation algorithms has been performed through real-time testing on a hardware prototype built using NI PXI

platform [36], as well as through LabVIEW FPGA hardware simulations. The performance of the proposed algorithms in terms of estimation accuracy, resource utilization, and processing time has been compared with QR decomposition-based DOA estimation methods (QR-R, QR-Q). Both simulations and real-time experiments establish LU-U to be superior to others in all performance parameters. However, QR-R has been found to have slightly better estimation accuracy (compared with LU-U) which comes at a much higher cost in terms of FPGA resources consumption and processing time. LU-U consumes the least amount of FPGA resources whereas QR-R consumes the highest. In addition, LU-U has been also found to be the fastest in computing the DOA estimates.

This paper is organized as follows: Section II presents the system model; section III describes the hardware implementation of the proposed DOA estimation algorithms using a pipelined architecture; section IV discusses the FPGA resources utilization for the proposed algorithms as well as LU and QR factorization; Section V presents Matlab and FPGA simulation results; Section VI describes the experimental setup for the real-time FPGA DOA estimation and presents the experimentation results; and conclusions are presented in section VII.

II. SYTEM MODEL

A uniform linear array (ULA) consisting of four omni-directional antennas is shown in Fig. 1. The distance between the adjacent antennas is 16 cm, which is equivalent to having the wavelength of 900 MHz. Single source $K = 1$ and multiple narrowband sources $K = 2$ are considered for testing using real hardware, LabVIEW software [37], and LabVIEW FPGA modules [35].

FIGURE 1. A uniform linear array (ULA) and a single source in the far-field region of the ULA.

We consider the cases of $K = 1$ and multiple narrowband sources $K = 2$ present in the far-field region of a ULA consisting of $M = 4$ elements. The sources are assumed to be lying at angles θ_1 *and* θ_2. At any time instant t, the snapshot of the signal received at the ULA can be expressed as:

$$\boldsymbol{x}_m(t) = \sum_{i=1}^{K} \boldsymbol{s}_i(t) e^{-j(2\pi/\lambda)dm\cos\theta_i} + \boldsymbol{n}_m(t);\ (m = 1, 2, \cdots, 4) \quad \text{and } K = 1, 2 \tag{1}$$

where $s_i(t)$ is the signal from the i-th incident source, λ is the wavelength, $(d = \lambda/2)$ the spacing distance of ULA, and $n_m(t)$ is the noise at the m-th element.

The received data can be expressed as:

$$\boldsymbol{X}(t) = \boldsymbol{A}(\theta)\,\boldsymbol{S}(t) + \boldsymbol{N}(t), \tag{2}$$

where $\mathbf{A}(\theta)$ is the $(M \times K)$ array response matrix given as:

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

Where $\boldsymbol{a}(\theta_i)$ for $i = 1, 2, \ldots, K$ is the corresponding array response vector.

$$\boldsymbol{a}(\theta_K) = \left[1\ \ \cdots\ \ u_K^M\right]^T, \quad \textit{where } u_k = \exp\left(-j2\pi d\cos(\theta_k)/\lambda\right) \tag{4}$$

S(t) is the vector of received signals given by:

$$\boldsymbol{S}(t) = \left[\boldsymbol{s}_1(t)\ \ \boldsymbol{s}_2(t)\ \ \ldots\ \ \boldsymbol{s}_K(t)\right]^T, \tag{5}$$

and,

$$\boldsymbol{N}(t) = \left[\boldsymbol{n}_1(t)\ \ \cdots\ \ \boldsymbol{n}_M(t)\right], \tag{6}$$

is the $(M \times 1)$ additive white Gaussian noise (AWGN) vector. Here and in the following, the superscripts T and $*$ denote the transpose and conjugate operations, respectively.

A. PROPOSED DOA ESTIMATION METHODS

In the proposed methods, we employ LU decomposition to find the DOAs of multiple RF incident sources. The DOA information can be extracted from either the signal space of the lower triangular matrix $\boldsymbol{L}$ or the signal space of the upper triangular matrix $\boldsymbol{U}$. Least square (LS) approach of finding the direction matrix is applied. Detailed information about the proposed methods is given in the following subsections.

Method 1: Extract DOAs from L matrix employing shift invariant property of the array.

In this method, LU factorization is employed to estimate the lower triangular matrix $\boldsymbol{L}$. The following steps show the proposed method in details for multiple sources $K = 2$ and the number of antennas $M = 4$.

Step 1: Apply LU factorization on data matrix $\boldsymbol{R}$.

$$\boldsymbol{R} = LU(R) = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ l_{21} & 1 & 0 & 0 \\ l_{31} & l_{32} & 1 & 0 \\ l_{41} & l_{42} & l_{43} & 1 \end{bmatrix}}_{\boldsymbol{L}} \times \underbrace{\begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix}}_{\boldsymbol{U}} \tag{7}$$

Step 2: Extract the first two columns of $\boldsymbol{L}$ which span the same signal space as the columns of the steering vectors in $\boldsymbol{A}(\theta)$.

So, the signal space for the two sources $\boldsymbol{L_s} \in \mathrm{C}^{M\times 2}$ can be obtained as:

$$\boldsymbol{L}_s = \begin{bmatrix} 1 & 0 \\ l_{21} & 1 \\ l_{31} & l_{32} \\ l_{41} & l_{42} \end{bmatrix} \tag{8}$$

The data matrix $\boldsymbol{L}_s$ with dimension $(M \times 2)$ will be used to estimate the DOAs. Doolittle's method [38] can be applied to find the entries of $\boldsymbol{L}$ and $\boldsymbol{U}$ as:

$$\begin{aligned} l_{ji} &= \frac{r_{ji} - \left(\sum_{q=1}^{i-1} l_{jq}u_{qi}\right)}{u_{ii}}; \; i \prec j \quad \text{and } i = 1, 2, \ldots, M \\ u_{ij} &= r_{ij} - \left(\sum_{q=1}^{i-1} l_{iq}u_{qj}\right); \; j \geq i \quad \text{and } j = 1, 2, \ldots, M \end{aligned} \tag{9}$$

Step 3: Partition $\boldsymbol{L}_s$ data matrix into two sub-matrices of size (3×2), such that:

$$\begin{aligned} \boldsymbol{L}_{s1} &= \boldsymbol{L}_s(1 : M-1, 1 : 2), \\ \boldsymbol{L}_{s2} &= \boldsymbol{L}_s(2 : M, 1 : 2) \end{aligned} \tag{10}$$

Since range of $\Re[l_s] = \Re[A]$, there must exist a unique matrix T, such that:

$$\boldsymbol{L}_s = \begin{bmatrix} \boldsymbol{l}_{s1} \\ \boldsymbol{l}_{s2} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1(\theta)\boldsymbol{T} \\ \boldsymbol{A}_1(\theta)\boldsymbol{\Phi}(\theta)\boldsymbol{T} \end{bmatrix}, \tag{11}$$

where $\boldsymbol{A}_1(\theta) = \begin{bmatrix} \boldsymbol{a_1}(\theta_1) & \boldsymbol{a_1}(\theta_2) \end{bmatrix}$ is the (3×2) array response matrix, $\boldsymbol{a}_1(\theta_1) = \begin{bmatrix} 1 & \cdots & u_1^3 \end{bmatrix}^T$, and $\boldsymbol{\Phi}(\theta)$ is a (2×2) diagonal matrix containing information about the DOAs of incident sources.

$$\boldsymbol{\Phi}(\theta) = diag\begin{bmatrix} e^{-\frac{j2\pi d\cos(\theta_1)}{\lambda}} & \cdots & e^{-\frac{j2\pi d\cos(\theta_2)}{\lambda}} \end{bmatrix}$$

It can be easily seen that $\Re[l_{s1}] = \Re\left[l_{s2}\right] = \Re[A_1]$ since $\boldsymbol{l}_{s1}$ *and* $\boldsymbol{l}_{s2}$ span the same signal space. This leads to both spaces being related by a nonsingular transform $\boldsymbol{\Lambda}$ as follows:

$$\boldsymbol{l}_{s2} = \boldsymbol{l}_{s1}\boldsymbol{\Lambda} \tag{12}$$

Since $\boldsymbol{A}$ is a full rank for uncorrelated sources, (12) can be expressed as:

$$\boldsymbol{\Lambda} = \boldsymbol{T}^{-1}\boldsymbol{\Phi}(\theta)\boldsymbol{T} \tag{13}$$

The eigenvalues of the matrix $\boldsymbol{\Lambda}$ are the diagonal elements of $\boldsymbol{\Phi}(\theta)$. Finding the eigenvalues of $\boldsymbol{\Lambda}$ will lead to obtaining the DOAs for the incident sources.

$$\begin{aligned} \boldsymbol{\Lambda} &= \arg\ \min_{(\boldsymbol{\Lambda})} \|\boldsymbol{l}_{s2} - \boldsymbol{l}_{s1}\boldsymbol{\Lambda}\|_F^2 \\ &= \arg\min_{(\boldsymbol{\Lambda})} tr\left\{[\boldsymbol{l}_{s2} - \boldsymbol{l}_{s1}\boldsymbol{\Lambda}]^H[\boldsymbol{l}_{s2} - \boldsymbol{l}_{s1}\boldsymbol{\Lambda}]\right\} \end{aligned} \tag{14}$$

The least square solution of (14) can be found as:

$$\boldsymbol{\Lambda} = \left[\boldsymbol{l}_{s1}^H\boldsymbol{l}_{s1}\right]^{-1}\boldsymbol{l}_{s1}\boldsymbol{l}_{s2} \tag{15}$$

Step 4: Compute the eigenvalues Γ_K of the matrix $\boldsymbol{\Lambda}$ in (15).

Step 5: Estimate the DOAs of multiple incident sources using the following expression:

$$\theta_K = \cos^{-1}\left(\frac{angle\left((\Gamma_K)\right)}{2\pi d}\right); K = 1, 2 \tag{16}$$

where Γ_K is the k^{th} eigenvalue.

Method 2: Extract DOAs from U data matrix employing the shift invariant property of the array.

The output data matrix $\boldsymbol{U}$ from (7) and ESPRIT shift invariant rotational property of the array will be used to estimate the DOAs of incident sources as follows:

Step 1: Extract the signal space from the data matrix $\boldsymbol{U}$. The signal space $\boldsymbol{U}_s$ for the $K = 2$ sources can be obtained by selecting the first 2 rows of $\boldsymbol{U}$ as:

$$\boldsymbol{U}_s = \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \end{bmatrix} \tag{17}$$

Step 2: Perform the Hermitian operation $(\cdot)^H$ on the data matrix in (17).

$$\boldsymbol{U}_{ss} = \boldsymbol{U}_s^H = \begin{bmatrix} 0 & u_{11}^* \\ u_{22}^* & u_{12}^* \\ u_{23}^* & u_{13}^* \\ u_{24}^* & u_{14}^* \end{bmatrix} \tag{18}$$

where $(\cdot)^*$represents conjugate operation.

Step 3: Partition the $\boldsymbol{U}_{ss}$ matrix into two sub-matrices of size (3×2), $\boldsymbol{U}_{s1}$ *and* $\boldsymbol{U}_{s2}$ as follows:

$$\begin{aligned} \boldsymbol{U}_{s1} &= \boldsymbol{U}_{ss}(1 : M-1, 1 : 2), \\ \boldsymbol{U}_{s2} &= \boldsymbol{U}_{ss}(2 : M, 1 : 2) \end{aligned} \tag{19}$$

Since range of $\Re[U_{ss}] = \Re[A]$, there must exist a unique matrix T, such that:

$$\boldsymbol{U}_{ss} = \begin{bmatrix} \boldsymbol{U}_{s1} \\ \boldsymbol{U}_{s2} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1(\theta)\boldsymbol{T} \\ \boldsymbol{A}_1(\theta)\boldsymbol{\Phi}(\theta)^*\boldsymbol{T} \end{bmatrix}, \tag{20}$$

Since $\boldsymbol{U}_{s1}$ *and* $\boldsymbol{U}_{s2}$ span the same signal space, they are related by a nonsingular transform $\boldsymbol{\Omega}$ as follows:

$$\boldsymbol{U}_{s2} = U_{s1}\Omega \tag{21}$$

The LS (least square) solution of (21) can be found as:

$$\boldsymbol{\Omega} = \left[\boldsymbol{U}_{s1}^H\boldsymbol{U}_{s1}\right]^{-1}\boldsymbol{U}_{s1}\boldsymbol{U}_{s2} \tag{22}$$

Step 4: Compute the eigenvalues Ω_k of the matrix $\boldsymbol{\Omega}$ in (22).

Step 5: Estimate the DOAs of multiple incident sources using the following expression:

$$\hat{\theta}_k = -\cos^{-1}\left(\frac{angle\left((\Omega_K)\right)}{2\pi d}\right) \tag{23}$$

where $\widehat{\theta_K}$ is the estimated DOA of the k^{th} source for $K = 1, 2$.

III. HARDWARE IMPLEMENTATION OF PROPOSED DOA ALGORITHMS

For hardware implementation of the proposed DOA estimation algorithms, we selected Xilinx Virtex-5 SXT FPGA [34] target hardware and programmed it using LabVIEW software [37]. LabVIEW graphical software facilitates configuring NI-certified hardware modules in a block diagram fashion, which is suitable for fast prototyping designs.

The hardware implementation model is as shown in Fig. 2. Signals received from the ULA are down-converted, digitized, and stored in a FIFO (first-in first-out queue). These steps are executed on the host (PC) while the DOA estimation algorithm is executed on the FPGA target. Signal data is transferred to the FPGA through the FIFO using direct memory access for speedy transfer.

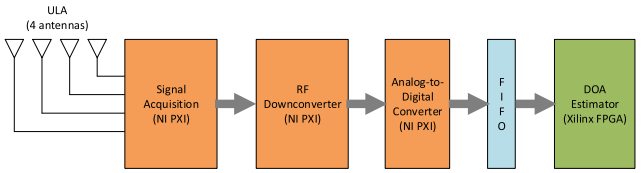

FIGURE 2. Hardware implementation model.

To achieve high throughput, a pipelined architecture is used for the FPGA implementation of the proposed DOA estimation algorithms, as shown in Fig. 3. The different stages of the pipeline represent the major operations of the algorithm. Data flows from one stage of the pipeline to the next one permitting high throughput implementation for the chosen algorithm.

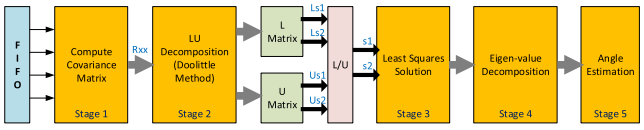

FIGURE 3. Pipelined execution of DOA estimation algorithm based on LU factorization.

Stage 1: The covariance matrix $\boldsymbol{R}_{xx}$ is estimated based on the data received from the four-antenna array of the ULA. The estimated covariance matrix from a number of snapshots can be calculated as:

$$\hat{R}_{xx} = E\left[\boldsymbol{x}(t)\boldsymbol{x}(t)^H\right] = \frac{1}{N}\sum_{t=1}^{N}\boldsymbol{x}(t)\boldsymbol{x}(t)^H \tag{24}$$

where N is the number of snapshots, and $x(t)$ is the column vector from the ith antenna element. The entries of the covariance matrix $\boldsymbol{R}_{xx}$ can be expressed as:

$$\hat{R}_{xx} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{23} & r_{24} \\ r_{31} & r_{32} & r_{33} & r_{34} \\ r_{41} & r_{42} & r_{43} & r_{44} \end{bmatrix}, rij = x_i x_j^H$$

and

$$i, j = 1, 2, 3, 4 \tag{25}$$

For hardware implementation, signal data is first retrieved from the FIFO and the covariance matrix $\hat{R}_{xx}$ is generated through multiply and accumulate operations.

Stage 2: In the second stage, the LU factorization is performed using the Doolittle method. LU factorization factors the correlation matrix $\boldsymbol{R}_{xx}$ as a product of two matrices $\boldsymbol{L}$ and $\boldsymbol{U}$ such that $\boldsymbol{R}_{xx} = \boldsymbol{LU}$ where $\boldsymbol{L}$ is lower triangular matrix and $\boldsymbol{U}$ is upper triangular matrix. In $\boldsymbol{L}$, diagonal elements are all one (1) and elements located above the diagonal are all zero. In $\boldsymbol{U}$, elements below the diagonal are zero. Signal space can be extracted from the $\boldsymbol{L}$ and $\boldsymbol{U}$ matrices, which can be determined using the following steps:

Step 1: The elements of first row of the matrix $\boldsymbol{U}$ can be calculated from the first row of $\boldsymbol{R}_{xx}$ as:

$$u_{1j} = r_{1j} \text{ for } 1 \le j \le M \tag{26}$$

Step 2: The element of the first column of $\boldsymbol{L}$ can be calculated from the first column of $\boldsymbol{R}_{xx}$ and the element u_{11} as:

$$L_{i1} = \frac{r_{i1}}{u_{11}}, \quad for\ 2 \le i \le M \tag{27}$$

Step 3: Since the first row of $\boldsymbol{U}$ and first column of $\boldsymbol{L}$ are known, the elements of the second row of $\boldsymbol{U}$ can be calculated as:

$$u_{2j} = r_{2j} - l_{21} \cdot u_{1j} \text{ for } 2 \le j \le M \tag{28}$$

Step 4: Now, the second column of matrix $\boldsymbol{L}$ can be calculated as:

$$L_{i2} = \frac{r_{i2} - l_{i1} \times u_{12}}{u_{22}} \text{ for } 3 \le i \le M \tag{29}$$

Step 5: Following the same procedure above, the remaining columns of $\boldsymbol{L}$ and rows of $\boldsymbol{U}$ can be calculated. Table 1 below shows the matrix operations for computing the elements of $\boldsymbol{L}$ and $\boldsymbol{U}$ matrices.

TABLE 1. Matrix operations for computing elements of L and U matrices.

Us Matrix		Ls Matrix	
First row	Second row	First column	Second column
$u_{11} = r_{11}$	$u_{21} = 0$	$l_{11} = 1$	$l_{12} = 0$
$u_{12} = r_{12}$	$u_{22} = r_{22} - l_{21}\ u_{12}$	$l_{21} = \frac{r_{21}}{u_{11}}$	$l_{22} = 1$
$u_{13} = r_{13}$	$u_{23} = r_{23} - l_{21}\ u_{13}$	$l_{31} = \frac{r_{31}}{u_{11}}$	$l_{32} = \frac{r_{32} - l_{31}\ u_{12}}{u_{22}}$
$u_{14} = r_{14}$	$u_{24} = r_{24} - l_{21}\ u_{14}$	$l_{41} = \frac{r_{41}}{u_{11}}$	$l_{42} = \frac{r_{42} - l_{41}\ u_{12}}{u_{22}}$

For hardware implementation of the proposed methods, it is required to compute only the first two rows of $\boldsymbol{U}$ matrix and the first two columns of $\boldsymbol{L}$ since the case of two incident sources ($K = 2$) is considered. Table 1 lists the operations for this partial LU factorization.

Fig. 4 below illustrates the sequence of operations and data flow for the partial LU factorization listed in Table 1. r_{ij} are elements of the covariance matrix $\boldsymbol{R}_{xx}$. Column 1 of

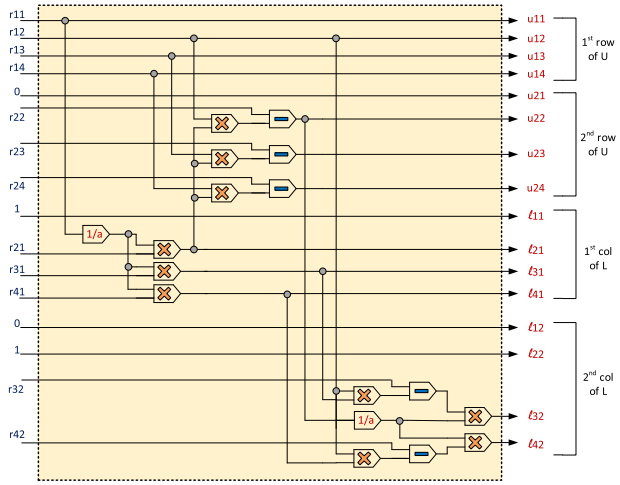

FIGURE 4. Sequence of operations and data flow of partial LU factorization.

the $\boldsymbol{L}$ matrix is computed first, followed by the second row of matrix $\boldsymbol{U}$. Finally, the second column of $\boldsymbol{L}$ matrix is computed. As can be seen in the diagram, operations under each column in Table 1 can be computed in parallel. For data size 16/8 (word length 16 bits, integer length 8 bits) and fixed-point data representation, the number of clock cycles required for division operation is nineteen and only one cycle for multiplication and subtraction operations.

For further processing (in Stage 3), matrix $\boldsymbol{L}$ is partitioned into two submatrices as given by (10), and matrix $\boldsymbol{U}$ is also partitioned into two submatrices as given by (19). The implementation of these steps using LabVIEW FPGA module is shown in Fig. 5 and Fig. 6 below. It is worth pointing out here, that for Stage 3, either $\boldsymbol{L}$ matrix or $\boldsymbol{U}$ matrix is used but not both at the same time.

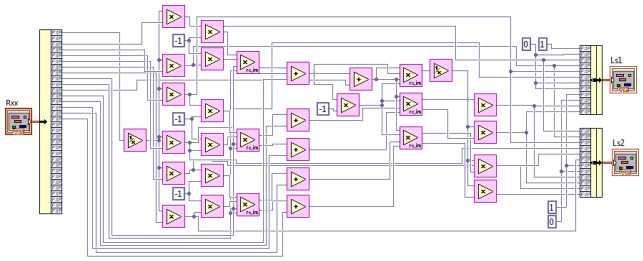

FIGURE 5. Generation of Ls matrix and its partitioning in LabVIEW FPGA.

Fig. 7 below shows the FPGA implementation of QR-Q decomposition. It can be observed that QR-Q requires significantly higher number of operations compared with LU-L and LU-U. In addition, it needs to implement the vector norm operation required in QR decomposition, as shown in Fig. 8. For QR-Q decomposition shown in Fig. 7, the rectangular box with red vertical lines is the sub-VI that implements the vector norm operation shown in Fig. 8. The implementation of QR-R is not shown here due to its huge size.

Stage 3: The least square (LS) solution of $\Lambda = \left[\boldsymbol{l}_{s1}^{H}\boldsymbol{l}_{s1}\right]^{-1}\boldsymbol{l}_{s1}\boldsymbol{l}_{s2}$ in (15) (if matrix $\boldsymbol{L}$ is chosen) or that of

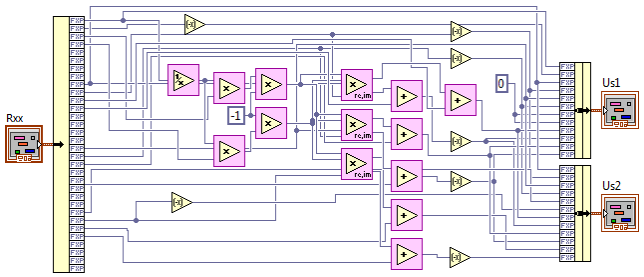

FIGURE 6. Generation of Us matrix and its partitioning in LabVIEW FPGA.

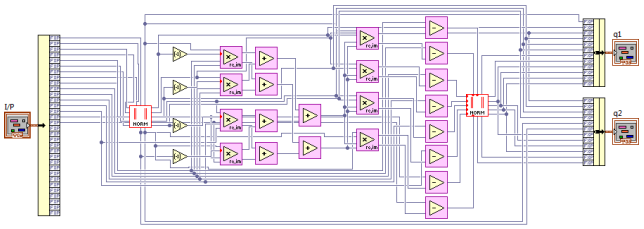

FIGURE 7. Generation of Qs matrix and its partitioning in LabVIEW FPGA.

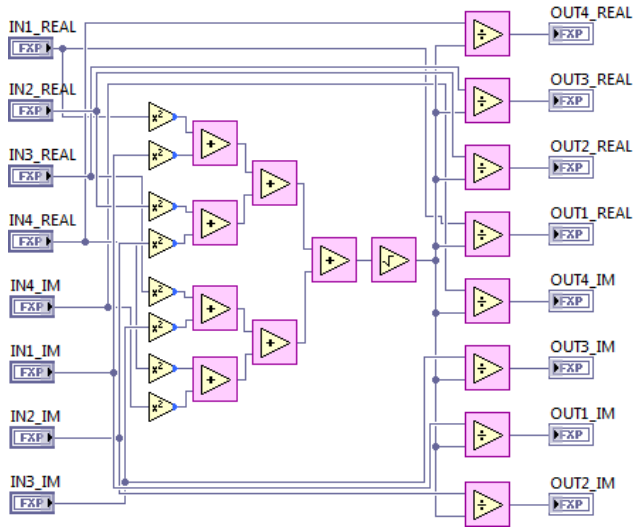

FIGURE 8. Generation of vector norm operation in LabVIEW FPGA.

$\boldsymbol{\Omega} = \left[\boldsymbol{U}_{s1}^{H}\boldsymbol{U}_{s1}\right]^{-1}\boldsymbol{U}_{s1}\boldsymbol{U}_{s2}$ in (22) (if matrix $\boldsymbol{U}$ is chosen) is implemented in this stage. The implementation of LS solution in LabVIEW FPGA requires a matrix inversion operation and complex-number multiplication operations. As shown in Fig. 9, the implementation of matrix inverse operation in LabVIEW FPGA requires six complex-number multipliers, two subtractions, one addition, one division, and eight real multipliers. The complex-number multiplication operation is different from real multiplication operation in terms of complexity and processing time. In finding the LS in (15) and (22), complex-number multiplication operations are required. Similarly, the implementation of complex-number multiplier for the inner product of a row vector with dimension (1×4) and a column vector with dimension of

(4×1) in LabVIEW FPGA requires four complex-number multipliers and six additions.

This stage may present a bottleneck for larger number of antennas (M>4) since matrix inverse and matrix multiplication operations for larger sized matrices will be computationally intensive and will consume a significantly large number of FPGA resources and clock cycles.

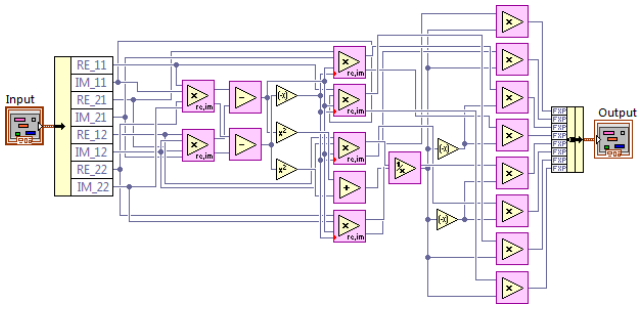

FIGURE 9. Implementation of matrix inverse operation for a 2×2 matrix in LabVIEW FPGA.

Stage 4: In this stage, eigen value decomposition (EVD) is performed to obtain the eigenvalues of matrix $\boldsymbol{\Lambda}$ (given in (15)) for the $\boldsymbol{L}$ matrix, and those of matrix $\boldsymbol{\Omega}$ (given in (22)) for the $\boldsymbol{U}$ matrix. Several approaches have been proposed for implementation of the eigen decomposition using FPGA Hardware. The most popular algorithms are Jacobi based Rotation, cyclic Jacobi rotation, Approximate Jacobi Method, and Algebraic Method. Calculating the eigen value decomposition for symmetric matrices with small dimension such as 3×3 or less, the Algebraic Method achieves a high throughput with much smaller number of slices as compared to Jacobi Methods [39]. Algebraic method is the appropriate choice since 2×2 matrix is considered for the case of $K = 2$ sources. In our method, only hardware implementation for the eigenvalues is required. For a given matrix $\boldsymbol{A}$, the eigenvalues can be calculated as $determinant(A - \lambda I) = 0$.

For a 2×2 matrix, the eigenvalues for A are the solution of quadratic equation, which involves the computation of the complex square root. Consider a complex number $z = x + jy$, where $x = Re(z) \quad and \ y = Im(z)$, the square root of z can be calculated as:

$$\sqrt{x + jy} = a + jb \tag{30}$$

$$a = \sqrt{\frac{x + \sqrt{x^2 + y^2}}{2}}, \quad b = \frac{y}{2a} \tag{31}$$

Fig. 10 below shows the implementation of the complex square root in LabVIEW FPGA. It requires four multipliers, two additions, one divisions, and two square roots.

As shown in Fig. 11, the eigen value decomposition for the case of a 2×2 matrix can be implemented in LabVIEW FPGA using three complex multipliers, two real multipliers, four divisions, and six additions.

This stage is the most computationally intensive in the pipeline due to the complex square root and division operations required in computing the EVD (in addition to complex multipliers, real multipliers, etc.). For estimating DOA of RF sources greater than two (K>2), this stage may also present a bottleneck in the pipeline.

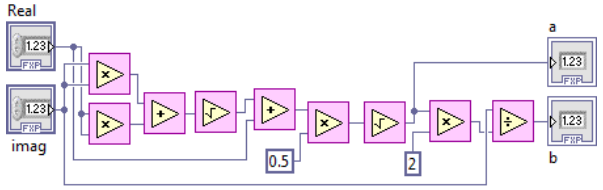

FIGURE 10. Computation of square root of a complex number in LabVIEW FPGA.

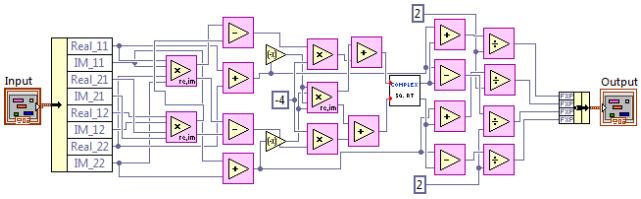

FIGURE 11. Implementation of eigen value decomposition in LabVIEW FPGA.

Stage 5: This is the final stage in the pipeline in which angle estimates are computed according to (16) for the $\boldsymbol{Ls}$ matrix and (23) for the $\boldsymbol{Us}$ matrix. Its implementation in LabVIEW FPGA is shown in Fig. 12 below. A look-up table (LUT) stores the pre-computed values of $\cos^{-1}()$ for speedy computation. The ACOS module shown in the figure is used for this purpose.

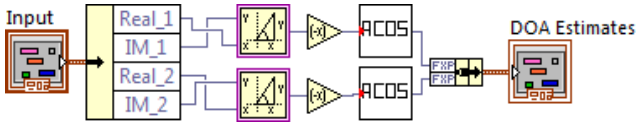

FIGURE 12. Computation of angle estimates in LabVIEW FPGA.

IV. FPGA RESOURCES UTILIZATION AND PROCESSING TIME

The proposed algorithms have been implemented in hardware on NI PXIe-7965R FlexRIO FPGA module [40], which features DSP-focused Xilinx Virtex-5 SXT FPGA with 40 MHz of onboard base clock and 512 MB of onboard RAM. Table 2 lists the available resources in the FPGA [34].

Programming was done using LabVIEW FPGA modules with high throughput mathematical operations available for implementation on FPGAs. Fixed-point data type was selected and three different data sizes were used. The data sizes used are 16/8, 20/10, and 24/12 where the first number indicates word length in bits and the second number indicates integer length in bits. Fixed-point data type was chosen for the implementation because it offers acceptable accuracy with much less resource usage and higher speeds. On the other hand, floating-point data type offers higher accuracy at the cost of significantly more FPGA resources and lower speeds.

TABLE 2. FPGA Resources Available in Xilinx Virtex-5 SXT FPGA.

Resource	Max. Available
Total Slices	14,720
Logic Cells	94,208
Slice Registers	58,880
Slice LUTs	58,880
Max. distributed RAM (Kb)	1,520
Block RAMs (36 Kb each)	244
Total block RAM (Kb)	8,784
DSP48E Slices	640
Configuration Memory (Mb)	35.8

Separate LabVIEW codes (called virtual instruments or VIs) were developed that implement the proposed DOA algorithms employing LU-U and LU-L factorization. LabVIEW FPGA codes employing QR-Q and QR-R factorization were also developed for comparison. All these VI code files were compiled for testing and performance evaluation of the DOA estimation algorithms in real-time. A successful compilation produces a report on the FPGA resources consumed and processing time required (in MHz). Implementations of QR-Q, QR-R, LU-L, and LU-U factorization were also separately compiled for performance evaluation of these methods of factorization, which are at the heart of DOA estimation. LabVIEW FPGA VIs for three different data sizes were compiled and information on resources and timing requirements were recorded.

TABLE 3. Count of mathematical operations for QR and LU factorization.

Operations	QR-Q		QR-R		LU-L		LU-U	
	(4x4)	(8x8)	(4x4)	(8x8)	(4x4)	(8x8)	(4x4)	(8x8)
Addition	20	44	69	224	6	14	6	15
Subtraction	0	16	0	16	0	0	0	0
Multiplication	0	32	16	32	17	41	4	4
Complex Multiplication	8	16	41	112	6	14	3	7
Division	16	2	0	0	2	2	1	1
Square Root	2	2	2	2	0	0	0	0
Total # of Operations	46	112	128	386	31	71	14	27
# of Clock Cycles taken	59	60	75	93	22	25	20	23

A. LU AND QR FACTORIZATION

Table 3 below shows the count of various mathematical operations required and time taken (in number of clock cycles) to implement QR and LU factorization for (4 × 4) and (8 × 8) sized matrices using LabVIEW FPGA. It can be observed from the table that QR-R and QR-Q consume the highest amount of resources compared to the proposed methods LU-U and LU-L, with LU-U consuming the least amount of resources. For example, LU-U for an 8 × 8 matrix requires only 7 complex multipliers compared with 112 for QR-R, 16 for QR-Q, and 14 for LU-L. In terms of performance, the proposed LU methods provide an accurate estimation for the DOA but the QR-R has slightly better performance at low signal-to-noise ratio (SNR) coming at the cost of higher processing time and larger number of resources. It can also be noted that computational complexity and resource requirements for QR will increase significantly with increase in size of the matrix. Moreover, LU-U is the fastest in execution and QR-R is the slowest as is evident by the number of clock cycles taken. The number of clock cycles has been calculated based on the longest propagation path.

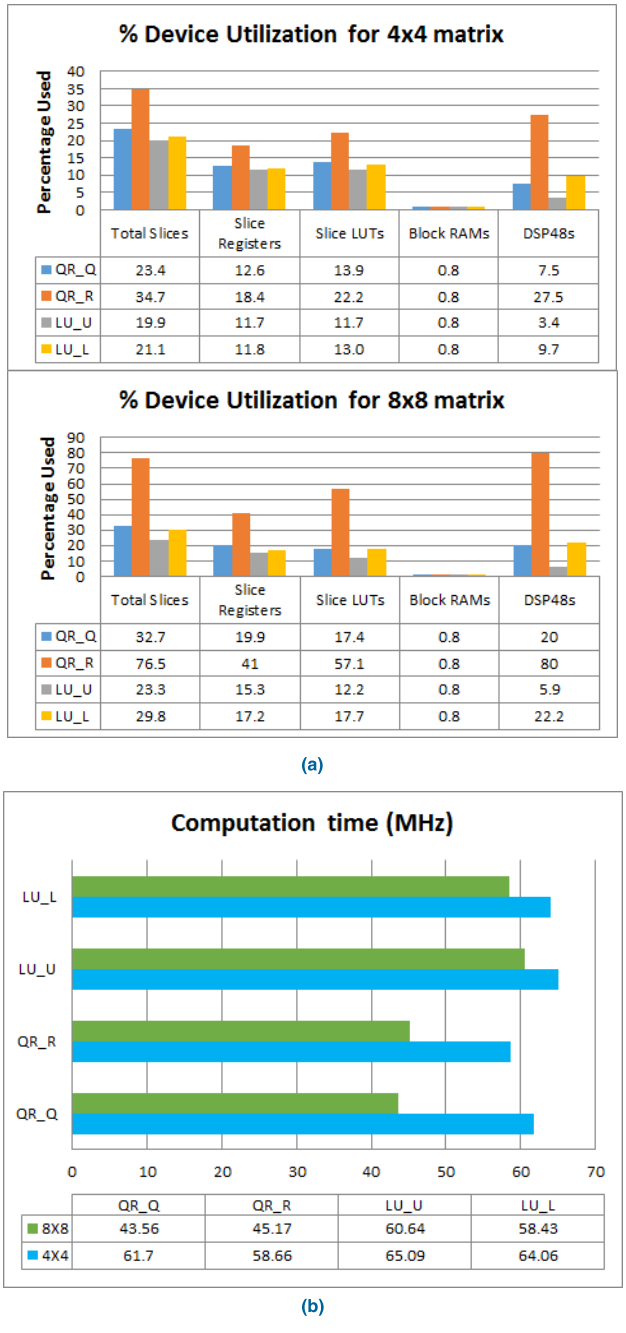

FIGURE 13. (a) % Device utilization for QR and LU factorization of a 4 × 4 matrix (top) and 8 × 8 matrix (bottom). (b) Computation time (in MHz) for QR and LU factorization for a 4 × 4 and 8 × 8 matrices.

Fig.13 (a) shows the percentage device utilization for each of QR-Q, QR-R, LU-U, and LU-L decomposition methods

for a 4 × 4 and 8 × 8 matrix, respectively. The computation time in MHz is shown in Fig. 13 (b). The percentage device utilization and computation time (in MHz) information presented here have been extracted from the successful LabVIEW FPGA compilation reports. It follows from the above discussion that LU-U stands out as the winner as it uses the least amount of resources and completes the decomposition of a matrix in the fastest time.

TABLE 4. FPGA resources consumed for DOA estimation using QR and LU.

Resource	DOA_QR_Q	DOA_QR_R	DOA_LU_U	DOA_LU_L
Total Slices	9555	10846	8867	8914
Slice Registers	18778	22840	16710	16763
Slice LUTs	24820	30568	22936	23438
Block RAMs	10	10	10	10
DSP48s	270	418	240	265

B. DOA ESTIMATION

Table 4 below shows the count of FPGA resources consumed (for word length of 16 bits and integer size of 8 bits) in the implementation of DOA estimation algorithm employing QR-Q, QR-R, LU-U, and LU-L. It can be clearly seen that DOA estimation employing LU-U consumes the least amount of resources while QR-R consumes the highest amount of resources.

Fig. 14 (a) shows the percentage device utilization for DOA estimation for three different data sizes of 16/8, 20/10, and 24/12, respectively. Fig. 14 (b) shows the computation times in MHz. It can be observed that overall LU-U outperforms all other methods in terms of resource utilization as well as processing time. For example, the processing time for the data size 24/12 is 51.08 MHz for LU-U, 47.33 MHz for LU-L, 44.78 MHz for QR-Q, and 42.75 MHz for QR-R. The percentage of resource utilization for slice registers for the data size 24/12 is 33.6% for LU-U, 34.2% for LU-L, 38.1% for QR-Q, and 47.3% for QR-R. It can also be observed that increase in data size results in higher amount of resources required and higher processing time (indicated by a decrease in frequency), without appreciable improvement in performance or estimation accuracy. Performance evaluation from simulations and real-time experiments of the proposed DOA estimation algorithms is presented and discussed in section V and VI, respectively.

The total computation time for DOA estimation in terms of clock cycles has also been calculated for the proposed LU-based methods as well as for QR. Table 5 shows the number of clock cycles taken by the estimator for each stage of the pipeline to calculate the DOA angle employing QR-Q, QR-R, LU-L, and LU-U. The number of clock cycles for each stage was calculated based on the longest path for data size 16/8. We observe that the proposed methods LU-U and LU-L consume less number of clock cycles compared with QR-Q and QR-R. It is also clear that Stage 4 (EVD) is the bottleneck in the pipeline as it requires the highest number of clock cycles due to complexity of matrix inverse and complex square root operations. The computational complexity of EVD will increase significantly as number of RF sources increase.

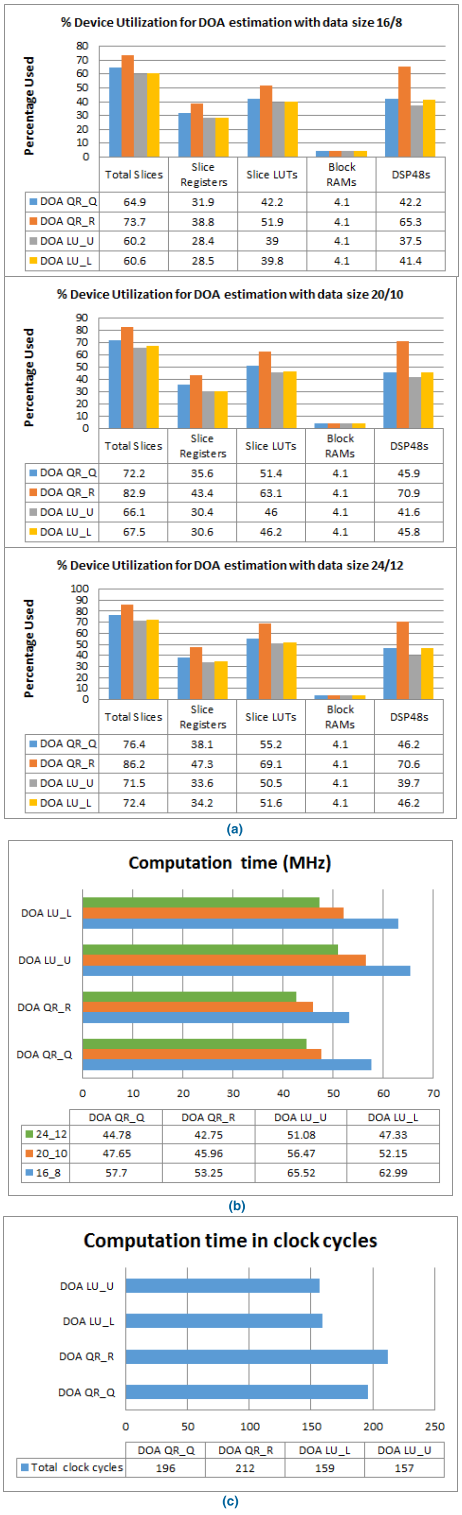

FIGURE 14. (a) % Device utilization for DOA estimation with data size 16/8, 20/10, and 24/12. (b) Computation time (in MHz) for DOA estimation with data size 16/8, 20/10, and 24/12. (c) Computation time in clock cycles for DOA estimation using LU and QR decomposition.

TABLE 5. Clock cycles consumed for DOA estimation using QR and LU.

#	Pipeline Stage	DOA QR-Q	DOA QR-R	DOA LU-L	DOA LU-U
1	Covariance Matrix computation	3	3	3	3
2	Matrix Decomposition (4x4)	59	75	22	20
3	Least square solution	28	28	28	28
4	Eigen value decomposition (EVD)	82	82	82	82
5	Angle Estimation	24	24	24	24
	Total clock cycles	196	212	159	157

Fig. 14 (c) below shows the total computation time in terms of clock cycles for DOA estimation employing QR-Q, QR-R, LU-L, and LU-U methods. With the 40 MHz onboard clock, the FPGA is estimated to take 3.925 μs for estimating DOA angles using LU-U, which is the fastest time among the four methods.

V. SIMULATION RESULTS

The performance of the proposed DOA estimation methods is compared with QR decomposition method. It is verified through Matlab simulations, simulations in LabVIEW FPGA, and by conducting experiments in real-time. Two separate cases are considered with a single source $K = 1$, and two sources $K = 2$ placed at arbitrarily selected angles from the array reference.

A. MATLAB SIMULATION RESULTS

The performance is measured in terms of root mean square error (RMSE) for the direction of arrival angle estimation. Four antenna elements are considered in total for a single source and multiple sources experiments. The distance between the adjacent elements is taken to be half the wavelength of the incoming signal, and the number of uncorrelated sources is taken as two non-coherent sources $K = 1$ for the first and $K = 2$ for the second experiment. Monte-Carlo trials are considered. The RMSE for the DOA estimation for multiple sources is defined as:

$$RMSE = \frac{1}{K}\sum_{k=1}^{K}\sqrt{E\left[(\hat{\theta}_k - \theta_k)^2\right]} \tag{32}$$

where k represents the source index, $E[Q]$ represents the expectation value of a random variable Q.

1) Single RF Incident Source

Consider a single source with direction of arrival angle $\theta = 75°$, SNR range set from -5 to 30 dB, the number of snapshots 500, and Monte-Carlo trials of 300 are used. Fig. 15 shows the RMSE values using the proposed methods LU-L and LU-U, and QR-R and QR-Q methods versus SNR. It can be observed in Fig. 15 that the proposed method has good performance even at low SNR. On the other hand, the QR-R method has slightly better performance but high computational complexity compared with the proposed methods.

2) Two RF Incident Sources

The case of two uncorrelated sources is also considered with direction of arrival angles at 65° and 85° from the array reference. SNR range is set from -5 to 30 dB, and the number of snapshots is 500. Monte-Carlo trials of 300 are used. The combined RMSE values for the two sources is shown in Fig. 16 versus SNR for the proposed methods LU-L and LU-U, and QR-R and QR-Q methods. It can be seen from Fig. 16 that the proposed method has very good estimation accuracy which is indicated through lower RMSE especially at low SNR. The QR-R method has slightly better performance at low SNR but similar performance at higher SNR such as 10 dB. However, QR-R Method has higher complexity and computational time.

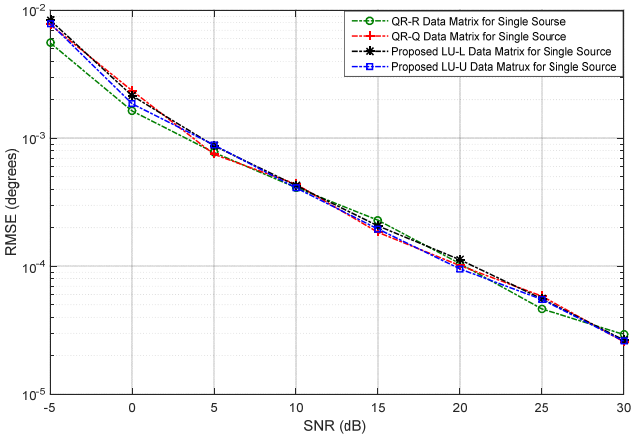

FIGURE 15. Simulated DOA estimates of the proposed methods (LU-L, LU-U) and (QR-Q, QR-R) method for single source lying at 75° from the array reference.

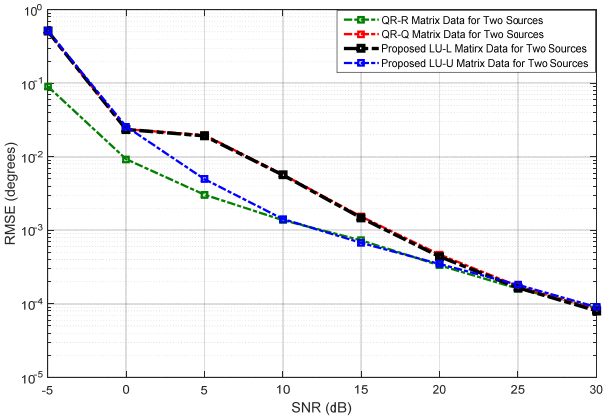

FIGURE 16. Simulated DOA estimates of the proposed methods (LU-L, LU-U) and (QR-Q, QR-R) method for two sources lying at 65° and 85° from the array reference.

B. LABVIEW FPGA SIMULATION RESULTS

Verification of the FPGA implementation of proposed DOA estimation algorithms based on LU decomposition has been also done using LabVIEW simulations. The implementation of these algorithms using LabVIEW FPGA modules has been discussed in Section 3 above. Simulation results of proposed algorithms have been compared with QR-based algorithms.

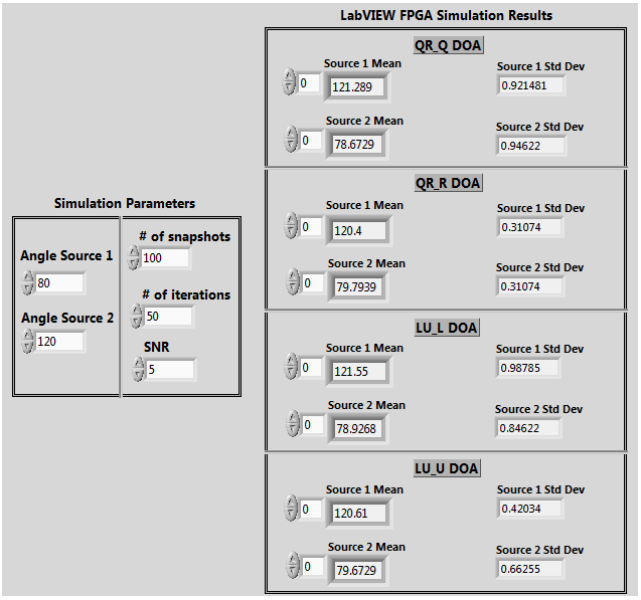

FIGURE 17. LabVIEW FPGA Simulation results for DOA estimates of the proposed methods (LU-L, LU-U) and (QR-Q, QR-R) method for two sources lying at 80 o and 120o from the array reference.

Fig. 17 shows the results at the end of LabVIEW simulation for DOA estimates (using QR and LU methods) performed with two sources placed at angles 80° and 120°, respectively.

Simulations were conducted with SNR ranging from 0 dB to 25 dB. DOA estimates were obtained through simulations running for 50 iterations with 100 snapshots in each iteration. Simulations were performed with both a single source and two sources placed at different angles. Fig. 18 shows performance graphs for DOA estimation accuracy for LU and QR based methods measured in RMSE for both a single source and two sources for different values of SNR in the range 0 dB to 25 dB.

It is clear from these graphs that LU-U has higher DOA estimation accuracy compared with QR-Q and LU-L, with QR-R slightly better than LU-U. However, the higher accuracy for QR-R also comes at a higher cost in terms of FPGA resource requirements and processing time. Therefore, considering all performance parameters, LU-U is found to be the optimum method for hardware real-time implementation of DOA estimation algorithms.

VI. REAL-TIME EXPERIMENTAL VERIFICATION

Real-time experimental verification of the proposed algorithms was carried out using NI PXI platform which houses a data acquisition module, digitizers, RF downconverters, RF up-converters, local oscillators, arbitrary waveform generators, and an FPGA module FlexRIO with Xilinx Virtex-5.

A. EXPERIMENTAL SETUP

The experimental setup with two transmitters and a uniform linear array with four antenna elements deployed at the receiver is shown in Fig. 19. The inter element spacing between the receiver antennas is half wavelength ($\lambda/2$).

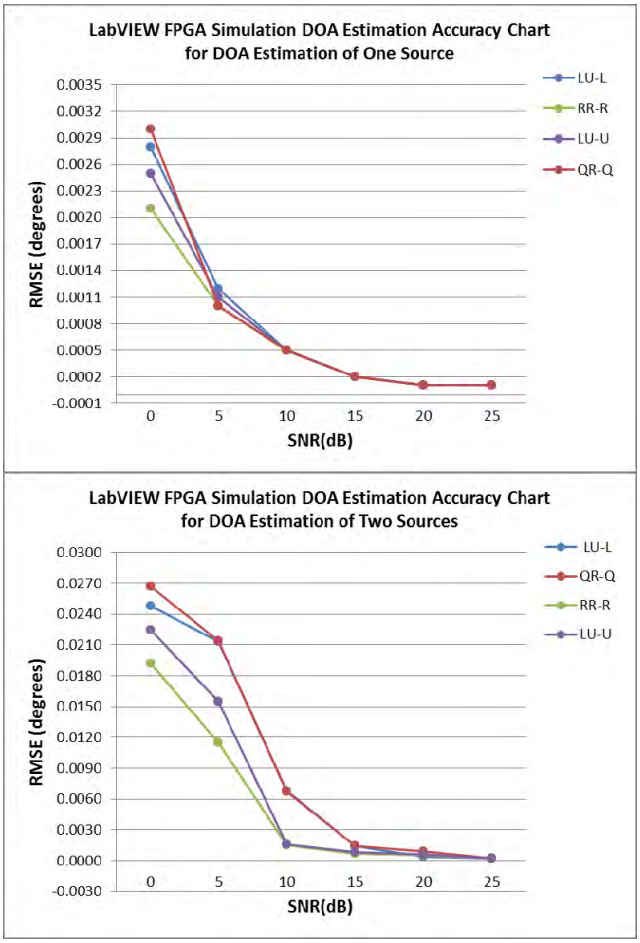

FIGURE 18. Performance Comparison of DOA estimation of the proposed methods (LU-L, LU-U) and (QR-Q, QR-R) method for both one and two sources.

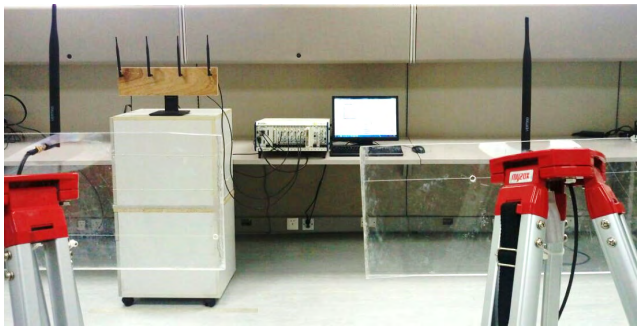

FIGURE 19. Experimental setup showing two transmitters (in the foreground) and a 4-element antenna array and PXI system (in the background).

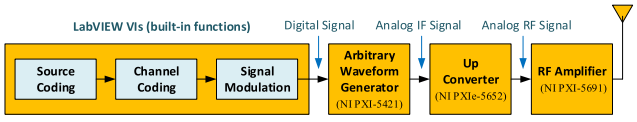

FIGURE 20. Transmitter unit block diagram.

The NI PXI transmitter is implemented as shown in Fig. 20. LabVIEW built-in functions for source coding, channel

coding, and modulation are used to first generate a signal in the digital domain. This digital signal is then converted to an intermediate frequency (IF) analog signal using an arbitrary waveform generator (AWG) module (NI PXI-5421). Next, the analog signal is converted to a radio frequency (RF) signal using an up-converter module (NI PXIe-5652). Finally, the signal is amplified before transmission using RF amplifier module (NI PXI-5691). All these modules are housed in the PXI chassis as shown in Fig. 21. The transmitter unit acts as a source lying in a far field region of the receiver.

FIGURE 21. NI PXI transmitter modules in the NI PXI platform chassis.

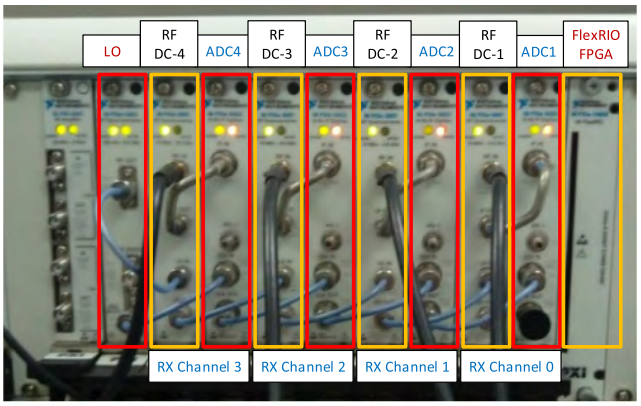

FIGURE 22. NI PXI receiver modules in the NI PXI platform chassis.

The AWG runs at a maximum sampling rate of 100 million samples per second. The IF signal has a frequency of 25 MHz and the maximum frequency of the RF signal generated by the up-converter is 2.7 GHz.

The receiver units on the NI PXI chassis are shown in Fig. 22 below. Each receiver unit is composed of an RF downconverter (PXIe-5601) and a high-speed digitizer (PXIe-5622). The NI PXI chassis shown in Fig. 22 houses four receiver units (each connected to an antenna in the 4 element ULA), a local oscillator, and FlexRIO FPGA module. All the receiver units share the same clock generated by the local oscillator (LO).

The downconverter operates at a maximum frequency of 2.7 GHz and a bandwidth of 15 MHz. The received signal is downconverted to an IF signal of 15 MHz which is then fed to a digitizer operating at a maximum sampling frequency of 64 Mega Samples/s. The outputs of the digitizers are modulated signals in (I, Q) form, from which the amplitude and phase information of the message signal is extracted.

B. REAL-TIME EXPERIMENTS FOR DOA ESTIMATION

The real-time experiments conducted for the validation of the proposed DOA estimation algorithms followed the procedure described below:

Step 1: Compile the LabVIEW FPGA codes for DOA estimation algorithms.

LabVIEW FPGA codes for DOA estimation algorithms employing LU and QR decomposition methods are compiled separately to run on the target FPGA. These cannot be combined to run in one code as they cannot fit in the limited resources available on the Xilinx Virtex-5 FPGA. The implementation of the proposed algorithms has been discussed in detail in Section III. The FPGA resource utilization and processing time information generated after successful compilation of the LabVIEW codes has been presented in Section IV. LabVIEW FPGA codes for data size 16/8 only was considered for the real-time experiment since it is optimum in terms of resource usage when compared with higher data sizes (20/10, 24/12) while the higher data sizes provide marginally higher estimation accuracy but at the cost of significantly higher resource usage and processing time.

Step 2: Set up the transmitter and receiver units and check signal reception.

Set up the transmitter and receiver units as mentioned in Section VI-A above. Send a 1 GHz sine wave signal from the transmitter unit and check signal reception at the receiver unit. This is done to check signals are received at the receiver with acceptable signal strength. Fig. 23 shows signal reception at the four receivers in the NI PXI receiver unit.

FIGURE 23. The received signal strength from source 1 (1 GHz sinewave) at the four element ULA at the receiver (seen in NI-RFSA Soft Front Panel on each of the four RF Downconverters (RF DC1 – DC4)).

Step 3: Run the DOA FPGA LabVIEW code and configure the front panel for real-time data acquisition.

On the transmitter side, each source is configured to transmit a sine wave of 1 GHz with an IQ rate of 1 Mega Samples/s,

and an SNR of 10 dBm. On the receiver side, RF down-converters for each channel are selected (under NI-RFSA Devices) and IQ carrier frequency and sampling rate are set using the front panel user interface shown in Fig. 24 (left). The figure (right) also shows a snapshot of I and Q signals acquired in real-time on each of the four receive channels.

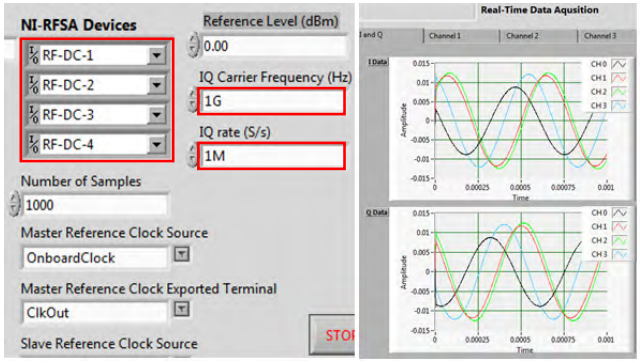

FIGURE 24. Hardware settings for the receiver as seen on the LabVIEW user interface (left) and real-time I and Q data signals received on each of the 4 channels of the ULA.

Step 4: Perform co-phase synchronization to calibrate the phase differences of all RF receiver channels.

The direction of arrival information of the RF source signal impinging on the receiver antennas of the ULA is extracted from the phase-shifted copies of the source signal received at the antennas. The phase shift is due to the time delay of the signal arriving at the other antennas of the ULA with respect to one antenna treated as the reference antenna. Thus, in order to avoid estimation errors, it is extremely important to co-phase the receiver antennas with respect to the reference antenna.

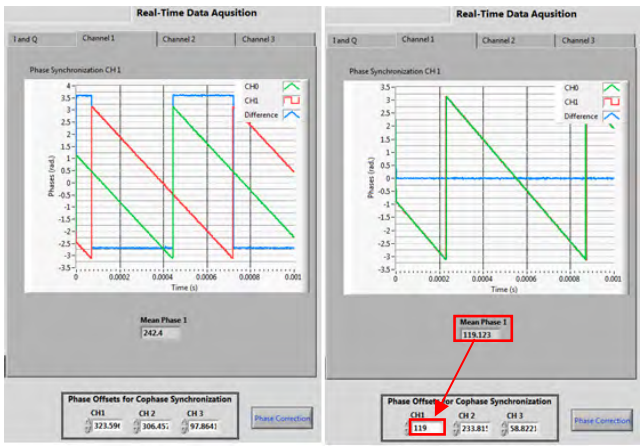

FIGURE 25. Co-phase synchronization of Channel 1 with Channel 0; before (left) and after synchronization (right).

The front panel view of the phase synchronization module developed in LabVIEW is shown in Fig. 25. The co-phase synchronization is done separately for each channel with respect to the reference channel. The figure shows the measured mean initial phase difference between the signals received from Channel 0 antenna (reference) and Channel 1 antenna. This phase-offset value is introduced in the received signals to compensate for this phase difference and make Channel 1 in co-phase with Channel 0. Fig. 25 shows the phase difference between the two channels before and after introducing the phase offset. The other channels (2 and 3) are co-phased with Channel 0 in the same manner.

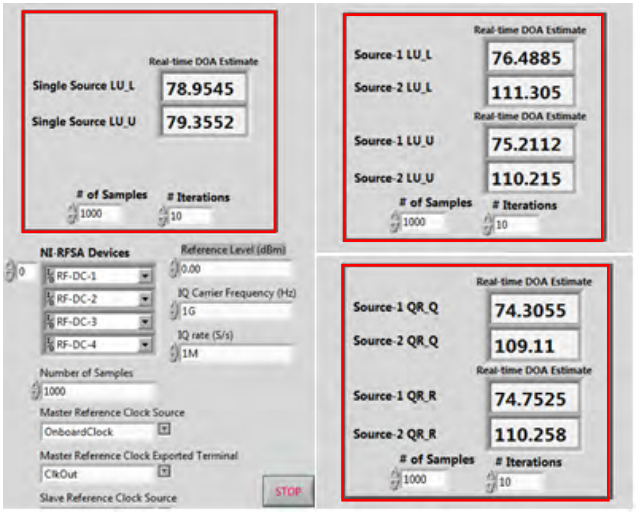

FIGURE 26. Real-time FPGA DOA estimates: for a single source located at 80° using LU-L and LU-U methods (top left), for two sources located at 75° and 110° using LU-L and LU-U methods (top right) and using QR-Q and QR-R methods (bottom right).

Step 5: Run the LabVIEW FPGA code for DOA estimation and record the estimated angles.

DOA estimates are obtained for both a single source and two sources placed at arbitrary angles with respect to the ULA. Fig. 26 shows DOA estimates using LU-L and LU-U methods for a single source placed at an angle of 80°, and DOA estimates using LU-L and LU-U methods for two sources placed at an angle of 75° and 110°, respectively. For comparison, DOA estimates employing QR factorization method were also obtained. Fig. 26 also shows the DOA estimates using QR methods for two sources placed at an angle of 75° and 110°, respectively. It can be seen that the DOA estimates with LU-U are more accurate and closer to the actual angle(s).

C. REAL-TIME DOA ESTIMATION RESULTS

Experimental verification for a single source and two sources placed at arbitrary angles was performed with 20 trials (1000 snapshots and 10 iterations in each trial) and the mean values of DOA estimates were calculated. This step validated the real-time performance of the proposed DOA estimation algorithms. The results of these trials are shown in Table 6 and Table 7 for one source and two sources, respectively.

It is clear from the test results shown in the tables above that LU-U and QR-R offer higher accuracy in DOA estimation compared with LU-L and QR-Q. Although the estimation

TABLE 6. Mean DOA estimate of 20 successful trials from real-time experimental verification with one source.

Actual location: One Source	Real-time DOA Estimation			
	Proposed LU-L	Proposed LU-U	QR-Q	QR-R
90°	89.34°	89.63°	89.32°	89.83°
80°	79.45°	79.78°	79.26°	79.80°
75°	75.31°	75.12°	74.65°	75.12°
60°	60.54°	60.31°	60.65°	60.86°
55°	55.75°	54.87°	55.82°	55.42°

TABLE 7. Mean DOA estimate of 20 successful trials from real-time experimental verification with two sources.

Actual location: Two sources	Real-time DOA Estimation			
	Proposed LU-L	Proposed LU-U	QR-Q	QR-R
(75°, 110°)	(75.42°, 109.31°)	(75.12°, 109.56°)	(75.72°, 109.41°)	(75.22°, 109.83°)
(70°, 95°)	(68.81°, 94.45°)	(70.52°, 94.32°)	(69.03°, 94.62°)	(70.12°, 94.81°)
(50°, 65°)	(48.71°, 64.62°)	(49.23°, 65.67°)	(48.62°, 65.82°)	(49.83°, 64.35°)
(100°, 125°)	(99.03°, 123.89°)	(99.52°, 124.41°)	(99.23°, 124.12°)	(99.31°, 124.56°)

accuracy of both LU-U and QR-R is comparable, LU-U is better overall as it consumes fewer resources and executes faster.

VII. CONCLUSIONS

In this paper, we presented the FPGA hardware implementation of two proposed DOA estimation algorithms based on LU decomposition. We evaluated the performance of these algorithms through software simulations, FPGA hardware simulations, and through real-time experiments. Experimental validation was done on a hardware prototype built using NI PXI platform, which allowed for real-time testing of the proposed algorithms. Performance was compared with QR decomposition-based algorithms. LU-U was found to be the optimum method for DOA estimation in terms of FPGA resource utilization, processing time, computational complexity, and estimation accuracy.

ACKNOWLEDGEMENT

The authors extend their appreciation to the Wireless Communications & Signal Processing lab at Prince Mohammad University for accessing its Facility.

REFERENCES

[1] R. T. Compton, Jr., *Adaptive Antennas: Concepts and Performance*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1988.
[2] R. A. Monzingo and T. W. Miller, *Introduction to Adaptive Arrays*. New York, NY, USA: Wiley, 1980.
[3] J. E. Hudson, *Adaptive Array Principles*. London, U.K.: Peter Peregrinus, 1981.
[4] J. C. Liberti and T. S. Rappaport, *Smart Antennas for Wireless Communications*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1999.
[5] A. Paulraj, R. Roy, and T. Kailath, "Estimation of signal parameters via rotational invariance techniques—ESPRIT," in *Proc. 19th Asilomar Conf. Circuits, Syst. Comput.*, 1985, pp. 83–89.
[6] P. Yang, F. Yang, and Z.-P. Nie, "DOA estimation with sub-array divided technique and interporlated esprit algorithm on a cylindrical conformal array antenna," *Prog. Electromagn. Res.*, vol. 103, pp. 201–216, 2010.
[7] Y.-S. Kim and Y.-S. Kim, "Improved resolution capability via virtual expansion of array," *Electron. Lett.*, vol. 35, no. 19, p. 1596, 1999.
[8] G.-M. Park and S.-Y. Hong, "Resolution enhancement of coherence sources impinge on a uniform circular array with array expansion," *J. Electromagn. Waves Appl.*, vol. 21, no. 15, pp. 2205–2214, Jan. 2007.
[9] A. J. Barabell, "Improving the resolution performance of eigenstructure-based direction-finding algorithms," in *Proc. ICASSP*, 1983, pp. 336–339.
[10] 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.
[11] L. Osman, I. Sfar, and A. Gharsallah, "Comparative study of high-resolution direction-of-arrival estimation algorithms for array antenna system," *Int. J. Res. Rev. Wireless Commun.*, vol. 2, no. 1, pp. 72–77, 2012.
[12] R. A. Horn and C. Johnson, *Matrix Analysis*. Cambridge, MA, USA: Cambridge Univ. Press, 1985.
[13] G. Golub and C. F. van Loan, *Matrix Computations*, 3rd ed. Baltimore, MD, USA: The John Hopkins Univ. Press, 1996.
[14] R. O. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. Antennas Propag.*, vol. 34, no. 3, pp. 276–280, Mar. 1986.
[15] G. Bienvenu and L. Kopp, "Adaptively to background noise spatial coherent for high resolution passive methods," in *Proc. ICASSP*, vol. 1. Denver, CO, USA, Apr. 1980, pp. 307–310.
[16] G. Bienvenu and L. Kopp, "Optimality of high resolution array processing using the eigen system approach," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 31, no. 5, pp. 1235–1248, Oct. 1983.
[17] B. D. Rao and K. V. S. Hari, "Performance analysis of root-music," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, no. 12, pp. 1939–1949, Dec. 1989.
[18] M. D. Zoltowski, G. M. Kautz, and S. D. Silverstein, "Beamspace root-MUSIC," *IEEE Trans. Signal Process.*, vol. 41, no. 1, pp. 344–364, Jan. 1993.
[19] M. Bouri and S. Bourennane, "High resolution methods based on rank revealing triangular factorizations," *Trans. Eng., Comput. Technol.*, vol. 2, pp. 35–38, Dec. 2004.
[20] N. Tayem, "Real time implementation for DOA estimation methods on NI-PXI platform," *Prog. Electromagn. Res. B*, vol. 59, pp. 103–121, 2014.
[21] N. Tayem, M. Omer, M. El-Lakki, S. A. Raza, and J. F. Nayfeh, "Hardware implementation of a proposed QR-TLS DOA estimation method and MUSIC, ESPRIT algorithms on NI-PXI platform," *J. Prog. Electromagn. Res. C*, vol. 45, pp. 203–221, Nov. 2013.
[22] D. Boppana, "FPGA-based WiMAX system design," Altera Corp., San Jose, CA, USA, Appl. Note CP-WIMAX-1.0, Oct. 2005.
[23] C. Dick, F. Harris, M. Pajic, and D. Vuletic, "Implementing a real-time beamformer on an FPGA platform," *Xcell J.*, vol. 86, no. 60, pp. 36–40, 2nd Quart., 2007.
[24] H. Arai and K. Ichige, "Hardware implementation of smart antenna systems for high speed wireless communication," in *Proc. General Assemblies Int. Union Radio Sci.*, 2005, Art. no. 01157.
[25] M. Abusultan, S. Harkness, B. J. LaMeres, and Y. Huang, "FPGA implementation of a Bartlett direction of arrival algorithm for a 5.8 GHz circular antenna array," in *Proc. IEEE Aerosp. Conf.*, Big Sky, MT, USA, Mar. 2010, pp. 1–10.
[26] B. J. LaMeres, R. J. Weber, Y. Huang, M. Abusultan, and S. Harkness, "Design and test of FPGA-based direction-of-arrival algorithms for adaptive array antennas," in *Proc. IEEE Aerosp. Conf.*, Big Sky, MT, USA, Mar. 2011, pp. 1–8.
[27] J. L. Tripp, A. A. Hanson, M. Gokhale, and H. Mortveit, "Partitioning hardware and software for reconfigurable supercomputing applications: A case study," in *Proc. ACM/IEEE Conf. Supercomput. (SC)*, Washington, DC, USA, Nov. 2005, p. 27.
[28] J. Williams, A. George, J. Richardson, K. Gosrani, and S. Suresh, "Characterization of fixed and reconfigurable multi-core devices for application acceleration," presented at the Reconfigurable Syst. Summer Inst. (RSSI), Urbana, IL, USA, Jul. 2008. [Online]. Available: http://rssi.ncsa.illinois.edu/
[29] 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, 2017, doi: 10.1049/joe.2017.0165.

[30] M. Kim, K. Ichige, and H. Arai, "Real-time smart antenna system incorporating FPGA-based fast DOA estimator," in *Proc. Veh. Technol. Conf.*, 2004, pp. 160–164.
[31] M. Kim, K. Ichige, and H. Arai, "Implementation of FPGA based fast DOA estimator using unitary MUSIC algorithm [cellular wireless base station applications]," in *Proc. Veh. Technol. Conf.*, vol. 1. Oct. 2003, pp. 213–217.
[32] J. Yan, Y. Huang, H. Xu, and G. A. E. Vandenbosch, "Hardware acceleration of MUSIC based DoA estimator in MUBTS," in *Proc. 8th Eur. Conf. Antennas Propag. (EuCAP)*, 2014, pp. 2561–2565.
[33] P. Boonyanant and S. Tan-a-Ram, "FPGA implementation of a subspace tracker based on a recursive unitary ESPRIT algorithm," in *Proc. TENCON*, 2004, pp. 547–550.
[34] *Xilinx Virtex-5 SXT FPGA Data Sheet*. Accessed: Nov./Dec. 2017. [Online]. Available: https://www.xilinx.com/support/documentation/selection-guides/virtex5-product-table.pdf
[35] *NI LabVIEW FPGA Module*. Accessed: Nov./Dec. 2017. [Online]. Available: http://www.ni.com/labview/fpga/
[36] *NI PXI Platform*. Accessed: Nov./Dec. 2017. [Online]. Available: http://www.ni.com/pxi/
[37] *NI LabVIEW Software Platform*. Accessed: Nov./Dec. 2017. [Online]. Available: http://www.ni.com/labview/
[38] M. H. Doolittle, "Method employed in the solution of normal equations and in the adjustment of a triangularization," in *Report of the Superintendent of the Coast and Geodetic Survey*. Washington, DC, USA: Government Printing Office, Jun. 1878, pp. 115–120.
[39] Y. Liu, C.-S. Bouganis, P. Y. K. Cheung, P. H. W. Leong, and S. J. Motley, "Hardware efficient architectures for eigenvalue computation," in *Proc. Design Autom. Test Eur. Conf.*, Munich, Germany, 2006, pp. 1–6.
[40] *PXIe-7965 PXI FPGA Module for FlexRIO*. Accessed: Nov./Dec. 2017. [Online]. Available: http://www.ni.com/en-lb/support/model.pxie-7965.html

AHMED A. HUSSAIN received the M.S. degree in electrical and computer engineering from the University of Florida, Gainesville, in 1998. He has over 15 years of university teaching experience in electrical engineering. He has also with Motorola as an Embedded Software Engineer. He is currently a Lecturer with the Department of Electrical Engineering, Prince Mohammad Bin Fahd University. His research interests include wireless communications, array signal processing, digital and embedded systems, computing, and program assessment and accreditation. He has several publications in the above areas to his credit.

NIZAR TAYEM (M'15) received the Ph.D. degree in electrical engineering from Wichita State University, Wichita, KS, USA. He joined the Electrical Engineering Department, Prince Mohammad Bin Fahd University, as an Assistant Professor, where he has been the Chair since 2011. He has authored/co-authored over 60 research publications in recognized international journals and conferences. He is a regular reviewer for many well-known journals. He was involved in research projects for Aerospace Sensor Networks Technology Thrust, Minority Leader's Program, and AFRL/Clarkson Aerospace. 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. He served as the Editor-in-Chief of the *International Journal on Electrical and Power Engineering* (ACEEE, USA).

MUHAMMAD OMAIR BUTT received the B.S. degree in telecom engineering from FAST University, Islamabad, Pakistan, in 2010, and the M.S. degree in electrical engineering from King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia, in 2014. He is currently a Lab Instructor with the Electrical Engineering Department, Prince Mohamed Bin Fahd University, and has contributed in ongoing researches in areas related to signals processing and wireless communication.

ABDEL-HAMID SOLIMAN received the B.Sc. degree in electronic and telecommunications from Arab Academy for Science, 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 Staffordshire University. He has over 27 years' experience of working in Higher Education Institutes. He is currently an Associate Professor in telecommunication and signal processing with the Department of Engineering, Staffordshire University. He has a multi-disciplinary academic/research experience in digital signal processing, telecommunications, data acquisition systems, wireless sensor networks, and image/video processing, and is focusing in his research on harnessing these technologies to contribute in the smart cities applications. 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 also leading and involved in several European funded projects in education, industry automation, and health care applications.

ABDULRAHMAN ALHAMED received the B.S. degree in electrical engineering from King Saud University (KSU), Riyadh, Saudi Arabia, in 2016. He is currently pursuing the master's degree with the University of California, San Diego. He joined the Department of Electrical Engineering, KSU, as a Teaching Assistant. His research interests include field programmable gate arrays applications, digital, analog, and RF integrated circuit design, and VLSI implementation of communications, and signal processing algorithms.

SALEH ALSHEBEILI has over 25 years of teaching and research experience in the area of communications and signal processing. He is currently a Professor with the Department of Electrical Engineering, King Saud University. He is a member of the Board of Directors of the King Abdullah Institute for Research and Consulting Studies from 2007 to 2009, a member of the Board of Directors of the Prince Sultan Advanced Technologies Research Institute (PSATRI) from 2008 to 2017, the Managing Director of PSATRI from 2008 to 2011, the Director of Saudi-Telecom Research Chair from 2008 to 2012. Since 2011, he has been the Director of the Technology Innovation Center, RF and Photonics in the e-Society (RFTONICS), funded by King Abdulaziz City for Science and Technology (KACST). He has been on the Editorial Board of the *Journal of Engineering Sciences of King Saud University* from 2009 to 2012. He has also an active involvement in the review process for a number of research journals, KACST general directorate grants programs, and national and international symposiums and conferences.