Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

FPGA-based Hardware Implementation of Computationally Efficient Multi-Source DOA Estimation Algorithms

Ahmed A. Hussain[1], Nizar Tayem[2], Abdel-Hamid Soliman[3], Redha M. Radaydeh[2]

[1]Department of Electrical Engineering, Prince Mohammad bin Fahd University, Al Khobar 31952, Saudi Arabia

[2]Department of Engineering and Technology, Texas A & M University, Commerce, TX, USA

[3]School of Engineering, Staffordshire University, Stoke-on-Trent, ST4 2DE, UK

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

ABSTRACT Hardware implementation of proposed direction of arrival (DOA) estimation algorithms based on Cholesky and LDL decomposition is presented in this paper. The proposed algorithms are implemented for execution on an FPGA (field programmable gate array) as well as a PC (running LabVIEW) for multiple non-coherent sources located in the far-field region of a uniform linear array (ULA). Prototype testbeds built using National Instruments (NI) Universal Software Radio Peripheral (USRP) software defined radio (SDR) platform and Xilinx Virtex-5 FPGA are originally constructed for the experimental validation of the proposed algorithms. Results from LabVIEW simulations and real-time hardware experiments demonstrate the effectiveness of the proposed algorithms. Specifically, the implementation of proposed algorithms on a Xilinx Virtex-5 FPGA using LabVIEW software clarifies their efficiency in terms of computation time and resource utilization, which make them suitable for real-time practical applications. Moreover, performance comparison with QR decomposition-based DOA algorithms as well as similar FPGA-based implementations reported in the literature is conducted in terms of estimation accuracy, computation speed, and FPGA resources consumed.

INDEX TERMS Cholesky and LDL decomposition, pipelined architecture, hardware implementation, Xilinx Virtex-5 FPGA, uniform linear array, LabVIEW, direction of arrival estimation, software defined radio, NI USRP-2901

I. INTRODUCTION

Source localization or direction of arrival (DOA) estimation of a radio frequency (RF) signal is a very important component in many practical applications such as channel estimation, beamforming, radar and sonar tracking, multiple-input multiple-output (MIMO) systems, etc. However, performing numerical simulations of DOA estimation algorithms to compute estimation accuracy and other performance parameters and to establish the effectiveness of the algorithms [1-7] is not sufficient. To establish the efficacy of an algorithm for real-time practical implementation, experimental validation on a hardware prototype is essential.

Sub-space DOA estimation techniques such as MUSIC [1] and ESPRIT [2] have been widely reported in the literature to have high estimation accuracy. However, these techniques and their several variants [3-7] require either eigenvalue decomposition or singular value decomposition of the received data matrix. These operations have a high computational cost (of the order of $O(N^3)$), making them unsuitable for real-time hardware implementation due to significantly higher processing time and hardware resources required.

Experimental validation of a DOA estimation algorithm requires a prototype testbed be built consisting of an antenna array for signal reception, and communication modules for down-conversion and digitization of the received signal. Subsequent signal processing may be done on a desktop processor running an operating system or on a hardware platform such as an FPGA. Building a prototype testbed could be an expensive and time consuming endeavor. Two popular commercial off-the-shelf (COTS) platforms which have been reported in the literature are ideal for rapid prototyping - one is the National Instruments (NI) PXI

platform [8] and the other is based on the software defined radio platform USRP [9] also from NI.

A few works have been reported in the literature on the hardware implementation and experimental validation of DOA estimation algorithms. A hardware implementation of DOA estimation methods based on QR decomposition on the NI PXI platform has been reported in [10-11], with signal processing carried out on a desktop processor. FPGA implementations of a Bartlett DOA estimator have been presented in [12-13], and implementations of MUSIC-based DOA algorithms are reported in [14-15]. The Bartlett DOA estimator in [12] is shown to be an efficient implementation in terms of computation time. FPGA real-time implementation based on QR and LU decompositions have been reported in [16] and [17], respectively. These methods [16-17] have been shown to be superior in performance (in terms of estimation accuracy, processing time, and resources utilization) to those of MUSIC and ESPRIT-based algorithms reported in the literature. For this reason, the QR-based algorithm has been taken as a benchmark for performance comparison.

One drawback of the NI PXI platform is that it is not easily scalable and has significantly higher cost when compared with the USRP SDR platform. Furthermore, USRPs are ideal for easy and quick deployment. In [18], a USRP-2921 implementation of AOA-based (angle of arrival) localization using MUSIC algorithm is presented. As mentioned earlier, subspace estimation techniques for DOA estimation are not amenable to efficient hardware implementation.

A COTS SDR platform comprising USRP-N200 units used in the testbed for determining the angle of arrival of RF incident signals is presented in [19]. It uses a maximum likelihood method to find the angle estimates which are computed on a desktop PC. Other works have been reported in the literature that use SDR platform for building an experimental testbed for DOA estimation [20-23] and for MIMO applications [24-25]. The focus of these works [19-25] was on establishing the benefits of deploying a COTS platform over other approaches. No new estimation algorithms were proposed for efficient hardware implementation.

In this paper, we propose two DOA estimation techniques based on LDL and Cholesky factorization for hardware implementation. Both Cholesky and LDL have been shown [26-28] to have low computational cost as they do not require either EVD or SVD. They require $O(N^3/6)$ flops while EVD/SVD-based methods require $O(N^3)$ flops, where N is the dimension of the data matrix. The lower the complexity of an algorithm, the lower the memory requirements and processing time. This makes LDL and Cholesky preferable over EVD/SVD-based methods for hardware implementation. For the experimental validation of the proposed algorithms, a testbed using NI USRP-2901 SDR platform [31] was built. Each USRP-2901 can support up to 2 receive channels, hence only two are required for building a 4-element uniform linear array (ULA) system for DOA estimation. The proposed algorithms have been implemented using LabVIEW software [29] for computing the DOA estimates on a desktop PC. These algorithms have been also implemented in a pipelined architecture (consisting of 5 stages) using LabVIEW FPGA high throughput modules [30] for computing the DOA estimates on a target FPGA.

Performance of the proposed algorithms has been measured in terms of estimation accuracy, count of FPGA resources consumed, and computation time, and has been compared with QR[1] decomposition-based DOA estimation methods (QR-Q, QR-R). The proposed DOA estimation algorithms have superior performance characteristics compared to QR-based methods. The proposed methods also compare favorably with similar FPGA-based implementations of DOA estimation algorithms reported in the literature [12-16].

The main contributions of this paper are summarized as follows:

- Propose two computationally efficient DOA estimation algorithms based on Cholesky and LDL decomposition suitable for FPGA hardware implementation. For these algorithms, only the lower triangular matrix needs to be computed for extracting angle information estimates.
- Implement efficient FPGA hardware realization of proposed algorithms employing a pipelined architecture. The proposed algorithms are superior to QR-based algorithms as well as others reported in the literature in terms of lower FPGA resources consumption and lower computation time, while their estimation accuracy compares favorably with QR-based algorithms.
- Conduct experimental validation of the proposed algorithms on a testbed built using NI USRP SDR platform. These algorithms are validated experimentally on an FPGA as well as a desktop processor with 4-element and 8-element ULAs.
- Construct separate testbeds for real-time experimental validation of proposed algorithms for: 1) estimation of up to two sources with a 4-element ULA on a desktop processor, 2) estimation of up to two sources with a 4-element ULA on an FPGA, and 3) estimation of up to three sources with an 8-element ULA on a desktop processor.
- Leverage the unique advantages and flexibility of USRPs and an FPGA combined in building a prototype testbed for experimental validation of real-time DOA estimation, which are performed for the first time herein to the best of the authors' knowledge.

[1]It is worth recalling here that QR decomposition is of the form $\boldsymbol{A} = \boldsymbol{QR}$, where $\boldsymbol{Q}$ is an orthogonal matrix and $\boldsymbol{R}$ is an upper triangular matrix. In the context of DOA estimation in this paper, QR-Q refers to matrix $\boldsymbol{Q}$ being used to extract the angle estimates, while QR-R refers to matrix $\boldsymbol{R}$ used for computing the angle estimates.

This paper is organized as follows: Section II presents the system model and the proposed methods based on LDL and Cholesky decomposition; section III describes the LabVIEW programming for computing DOA estimates on a desktop PC; section IV describes the LabVIEW FPGA implementation of the proposed DOA estimation algorithms on an FPGA; section V presents the USRP SDR testbeds for 4-element as well as 8-element ULAs; section VI presents results of real-time experimental validation of the proposed methods on the prototype testbeds; conclusions are presented in section VII.

II. SYSTEM MODEL

The system model in Fig. 1 shows a uniform linear array (ULA) of eight omni-directional antennas (M=8) placed 15 cm apart $(d = \lambda/2)$ which is equivalent of having the wavelength of a signal with frequency 1 GHz. Multiple non-coherent sources in the same plane as the ULA are considered for real-time testing using the NI USRP SDR platform. Up to two sources (K = *1, 2*) are considered in the case of data processing performed on the FPGA (due to resource and timing constraints) while up to three sources (K = *1, 2, 3)* are considered in the case of a desktop processor. The two RF sources lying in the far-field region of the ULA are assumed to be located at angles θ_1 and θ_2 from the ULA, respectively.

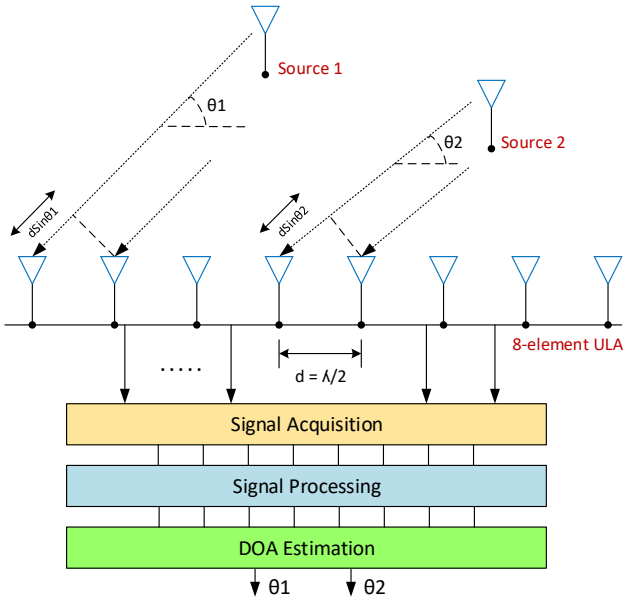


FIGURE 1. System model showing two sources in the far-field of an 8-element ULA.

Signals received at the ULA are acquired, downconverted, and digitized before being processed. The DOA estimates are then computed using the proposed algorithms implemented using a pipelined architecture as shown in Fig. 2.

The snapshot of the signal received at the ULA, at any time instant ***t***, can be expressed as:

$$x_m(t) = \sum_{i=1}^{K} s_i(t)\, e^{-j(2\pi/\lambda)dm\cos\theta_i} + n_m(t); (m = 1,2,\text{L}\ ,4)\ and\ K = 1,2 \quad (1)$$

where $s_i(t)$ is the i-th incident source signal, 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:

$$X(t) = A(\theta)S(t) + N(t), \quad (2)$$

where $A(\theta)$ is the (M x K) array response matrix given as:

$$\text{A}(\theta) = \left[\mathbf{a}(\theta_1) \quad \mathbf{a}(\theta_2) \quad \text{K} \quad \mathbf{a}(\theta_K)\right], \quad (3)$$

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

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

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

$$S(t) = \left[s_1(t) \quad s_2(t) \quad \text{K} \quad s_K(t)\right]^T, \quad (5)$$

and

$$N(t) = \left[n_1(t) \quad \text{L} \quad n_M(t)\right], \quad (6)$$

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

A. PROPOSED ALGORITHMS FOR EFFICIENT HARDWARE IMPLEMENTATION

The proposed DOA estimation algorithms are based on LDL and Cholesky decomposition methods which are suitable for efficient hardware implementation owing to their low computational complexity. Cholesky decomposition factors a Hermitian positive-definite matrix A into a lower triangular matrix L (with real and positive diagonal entries) such that $A = LL^*$, where L^* denotes the conjugate transpose of L**.** In LDL decomposition, which is a close variant of Cholesky, matrix A is factored into a lower triangular matrix L (with 1's on the diagonal), and a diagonal matrix D such that $A = LDL^*$.

For hardware implementation, one distinct advantage of the proposed methods is that it is sufficient to compute only the lower triangular matrix L for determining the DOA estimates of incident RF sources. This reduces processing time as well as memory storage requirements. The DOA information is extracted from the signal space contained in the lower triangular matrix L, and the least squares (LS) approach is used to obtain the direction matrix.

B. PIPELINED ARCHITECTURE IMPLEMENTATION

The proposed algorithms are implemented for execution in a pipelined architecture consisting of five (5) stages, as shown in Fig. 2.

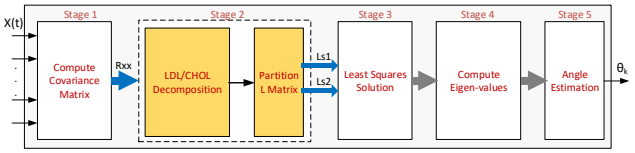


FIGURE 2. Pipelined execution of DOA estimation using proposed methods.

Details of each stage of the pipeline are presented below. In its implementation, up to two sources ($K = 2$) are considered for the two cases of a ULA consisting of four and eight antenna elements (*M=4 or 8*), respectively. The case of $M = 4$ is presented below.

Stage 1: Computation of Covariance Matrix $\boldsymbol{R}_{xx}$

In this stage, the N snapshots of the signal data received from the antenna array of the ULA is retrieved and used to compute the covariance matrix $\boldsymbol{R}_{xx}$ according to the equation below:

$$\boldsymbol{R}_{xx} = E\left[\boldsymbol{x}(t)\boldsymbol{x}(t)^H\right] = \frac{1}{N}\sum_{t=1}^{N}\boldsymbol{x}(t)\boldsymbol{x}(t)^H \tag{7}$$

where $x(t)$ is the column vector from the ith antenna element. The matrix $\boldsymbol{R}_{xx}$, thus obtained, is shown below:

$$\boldsymbol{R}_{xx} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{23} & r_{24} \\ r_{31} & r_{32} & r_{33} & r_{34} \\ r_{41} & r_{42} & r_{43} & r_{44} \end{bmatrix} \tag{8}$$

Stage 2: Matrix Decomposition

The covariance matrix $\boldsymbol{R}_{xx}$ computed in Stage 1 is decomposed by applying LDL (or Cholesky) factorization. Matrix decomposition using LDL factorization is performed as shown below:

$$\boldsymbol{LDL}^H(\boldsymbol{R}_{xx}) = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ l_{21} & 1 & 0 & 0 \\ l_{31} & l_{32} & 1 & 0 \\ l_{41} & l_{42} & l_{43} & 1 \end{bmatrix}}_{\boldsymbol{L}} \underbrace{\begin{bmatrix} D_{11} & 0 & 0 & 0 \\ 0 & D_{22} & 0 & 0 \\ 0 & 0 & D_{33} & 0 \\ 0 & 0 & 0 & D_{44} \end{bmatrix}}_{\boldsymbol{D}} \underbrace{\begin{bmatrix} 1 & l_{21}^* & l_{31}^* & l_{41}^* \\ 0 & 1 & l_{32}^* & l_{42}^* \\ 0 & 0 & 1 & l_{43}^* \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\boldsymbol{L}^H} \tag{9}$$

The entries of L and D are calculated as follows:

$$\begin{aligned} D_j &= r_{jj} - \sum_{k=1}^{j-1} L_{jk}L_{jk}^* D_k, \\ L_{ij} &= \frac{1}{D_j}\left(r_{ij} - \sum_{k=1}^{j-1} D_k L_{jk}L_{jk}^*\right); \text{for } i \text{ f } j. \end{aligned} \tag{10}$$

In case of Cholesky factorization, matrix $\boldsymbol{R}_{xx}$ is decomposed as follows:

$$\boldsymbol{R}_{xx} = \boldsymbol{L}\boldsymbol{L}^H \tag{11}$$

where $\boldsymbol{L}$ is a unique lower triangular matrix with positive diagonal entries. $\boldsymbol{L}$ is given by:

$$\boldsymbol{L} = \begin{bmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} \end{bmatrix} \tag{12}$$

where $l_{ij} > 0$ *for* $j \geq i$ can be found as:

$$\begin{aligned} l_{ii} &= \sqrt{\left(r_{ii} - \sum_{k=1}^{i-1} l_{ik}^2\right)}, \text{ for } i = j \\ l_{ij} &= \frac{r_{ij} - \sum_{k=1}^{j-1} l_{ik}l_{jk}}{l_{jj}}, \text{ for } i > j \text{ and } i = j+1, \text{L } ,3N \end{aligned} \tag{13}$$

For two sources, only the first two columns of $\boldsymbol{L}$ need to be extracted to compute the DOA estimates. The submatrix $\boldsymbol{L}_s$ of size *M* x *2* is obtained as:

$$\boldsymbol{L}_s = \begin{bmatrix} 1 & 0 \\ l_{21} & 1 \\ l_{31} & l_{32} \\ l_{41} & l_{42} \end{bmatrix} \quad \boldsymbol{L}_s = \begin{bmatrix} l_{11} & 0 \\ l_{21} & l_{22} \\ l_{31} & l_{32} \\ l_{41} & l_{42} \end{bmatrix} \tag{14}$$

(LDL) (Cholesky)

Stage 3: Least Squares Solution

In this stage, the least squares (LS) approach is used to obtain the direction matrix. First, the $\boldsymbol{L}_s$ matrix is further partitioned into two sub-matrices of size *(M-1)* x *2* as follows:

$$\begin{aligned} \boldsymbol{L}_{s1} &= \boldsymbol{L}_s(1:M-1,1:2), \quad \boldsymbol{L}_{s2} = \boldsymbol{L}_s(2:M,1:2) \\ \boldsymbol{L}_{s1} &= \boldsymbol{L}_s(1:3,1:2), \qquad \boldsymbol{L}_{s2} = \boldsymbol{L}_s(2:4,1:2) \text{ ; } M = 4 \end{aligned} \tag{15}$$

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

$$\boldsymbol{L}_s = \begin{bmatrix} l_{s1} \\ l_{s2} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1(\theta)\boldsymbol{T} \\ \boldsymbol{A}_1(\theta)\Phi\boldsymbol{T} \end{bmatrix}, \tag{16}$$

where $\boldsymbol{A}_1(\theta) = \begin{bmatrix} \mathbf{a}_1(\theta_1) & \mathbf{a}_1(\theta_2) \end{bmatrix}$ is the array response matrix of size (3×2), $a_1(\theta_1) = \begin{bmatrix} 1 & \text{L} & u_1^3 \end{bmatrix}^T$, and Φ is a diagonal matrix of size (2×2) containing information about the DOA angle estimates of the incident sources.

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

Both l_{s1} and l_{s2} span the same signal space and their ranks are same. They are related by a nonsingular transform Λ as follows:

$$l_{s2} = l_{s1}\Lambda \tag{17}$$

Equation (17) can be solved using the least squares (LS) approach which minimizes the difference between l_{s2} and $l_{s1}\Lambda$.

$$\begin{aligned}\Lambda &= \arg\ \min_{(\Lambda)} \left\| l_{s2} - l_{s1}\Lambda \right\|_F^2 \\ &= \arg\min_{(\Lambda)} tr\left\{ \left[l_{s2} - l_{s1}\Lambda \right]^H \left[l_{s2} - l_{s1}\Lambda \right] \right\}\end{aligned} \tag{18}$$

The LS solution of (18) can be found as:

$$\mathbf{\Lambda} = \left[l_{s1}^H l_{s1} \right]^{-1} l_{s1} l_{s2} \tag{19}$$

Stage 4: Computation of Eigenvalues

In this stage, the eigenvalues Γ_k of the matrix $\mathbf{\Lambda}$ in (19) are computed by performing EVD. The eigenvalues, for a given matrix A, can be calculated as: $determinant(A - \lambda I) = 0$

$$\mathbf{\Lambda} = \left[l_{s1}^H l_{s1} \right]^{-1} l_{s1} l_{s2} \tag{19}$$

Stage 5: Computation of DOA estimates

In the final last stage, the DOA angle estimates of multiple incident sources are computed using the following expression:

$$\theta_K = \cos^{-1}\left(\frac{angle\left((\Gamma_K)_\Lambda \right)}{2\pi d} \right);\ K = 1, 2 \tag{20}$$

where Γ_K is the k[th] eigenvalue.

III. LABVIEW SIMULATION OF PROPOSED ALGORITHMS

The proposed algorithms were first implemented in LabVIEW for theoretical validation, following the pipelined architecture illustrated in Fig. 2. Linear algebra math functions provided in LabVIEW were used in implementing the proposed algorithms. Received data *x(t)* is generated according to (2) which is then passed on to the first stage of the pipeline for computation of the covariance matrix. Fig. 3 shows part of the LabVIEW code implemented using linear algebra functions for DOA estimation using LDL method.

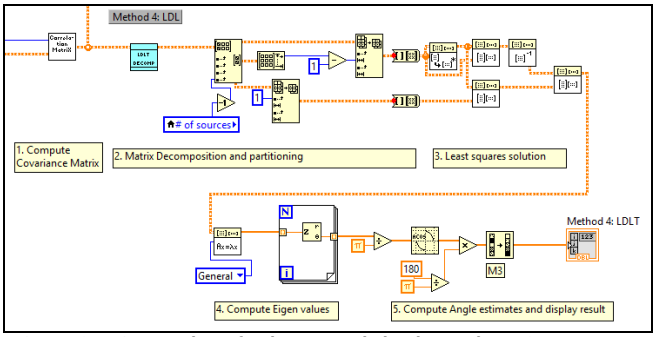


FIGURE 3. Screenshot of LabVIEW code implementing DOA estimation using proposed LDL method.

The user interface (UI) of the LabVIEW simulation program is show in Fig. 4. The UI allows for selecting the number of sources to be localized, source signal angles, number of receivers, SNR, number of snapshots, and related parameters. DOA estimates are computed for proposed algorithms as well as for QR-decomposition based algorithms for comparison.

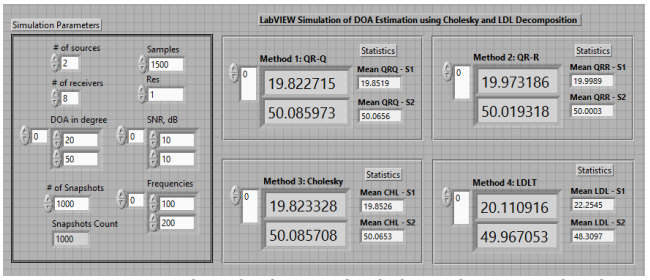


FIGURE 4. Screenshot of LabVIEW simulation UI for DOA estimation using proposed methods for two sources.

Fig. 5 shows the RMSE (root mean square error) vs. SNR curves for the case of a single source located at 20°, 500 snapshots, and four receivers. SNR is varied from 0 dB to 25 dB. It is clear from the figure that the estimation accuracy of the proposed methods matches that of QR-based methods, and, as expected, it improves significantly with incease in SNR value.

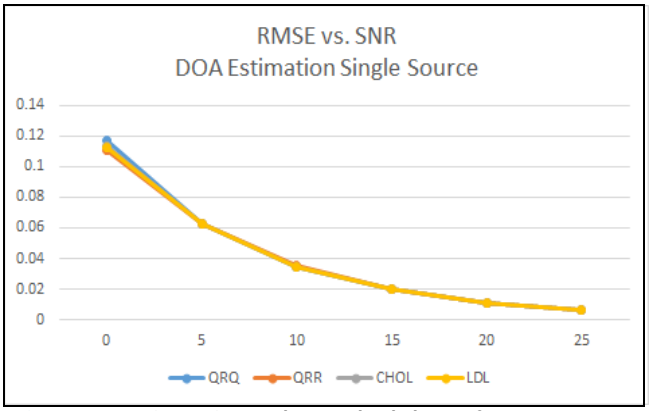
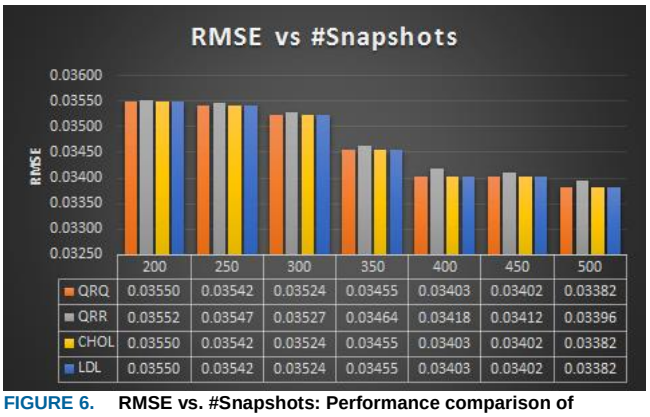


FIGURE 5. RMSE vs. SNR: LabVIEW simulation performance comparison of proposed methods with QR for DOA estimation of a single source (at 20°) and M=4.



	200	250	300	350	400	450	500
QRQ	0.03550	0.03542	0.03524	0.03455	0.03403	0.03402	0.03382
QRR	0.03552	0.03547	0.03527	0.03464	0.03418	0.03412	0.03396
CHOL	0.03550	0.03542	0.03524	0.03455	0.03403	0.03402	0.03382
LDL	0.03550	0.03542	0.03524	0.03455	0.03403	0.03402	0.03382

FIGURE 6. RMSE vs. #Snapshots: Performance comparison of proposed methods with QR for DOA estimation of a single source (at 20°) and M=4.

The effect of number of snapshots used for computation on the estimation accuracy of the proposed methods is also analysed. Fig. 6 shows the RMSE vs. #snapshots chart for the case of a single source located at 20°, 10 db SNR, and four receivers. Number of snapshots is varied from 200 to 500 in steps of 50. Performance of the proposed methods can be seen to improve with increasing number of snapshots.

Performance comparison of the proposed methods for two sources in terms of RMSE is also made, as shown in Fig. 7. The two sources are located at 70° and 120°, respectively. SNR value is varied from 0 dB to 25 dB. Number of receivers is $M = 4$. The performance of the proposed methods is slightly better than that of QR-Q. At low SNR, QR-R clearly has better performance.

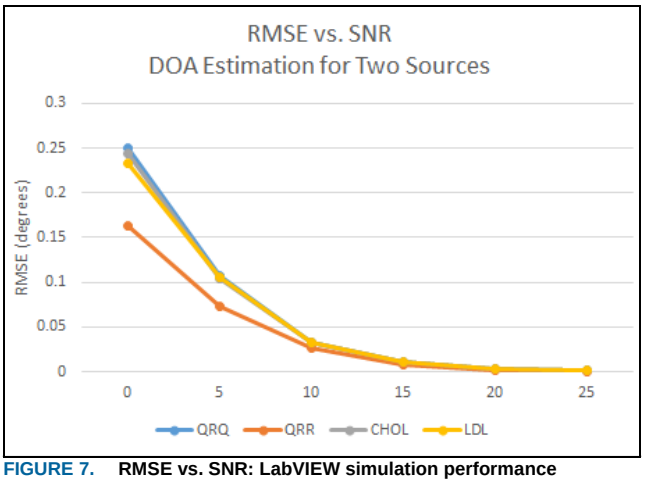


FIGURE 7. RMSE vs. SNR: LabVIEW simulation performance comparison of proposed methods with QR for DOA estimation of two sources (at 70° and 120°) and M=4.

From a comparison of Fig. 5 and Fig. 7, it can be observed that system performance deteriorates as the number of sources to be localized increase without an increase in the number of receiver antennas.

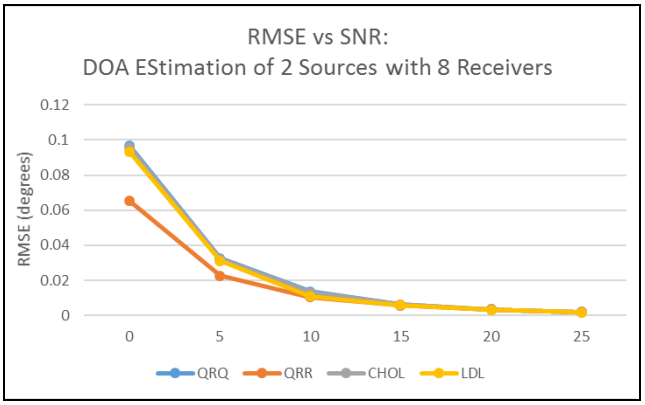


FIGURE 8. RMSE vs. SNR: LabVIEW simulation performance comparison of proposed methods with QR for DOA estimation of two sources (at 70° and 120°) and M=8.

Fig. 8 shows RMSE vs. SNR chart for DOA estimation of two sources using eight receivers ($M = 8$). The two sources are located at 70° and 120°, respectively. Upon comparing Fig. 7 and Fig. 8, we can observe signficant improment in system performance from the lower RMSE values for the case of eight receiver antennas ($M = 8$). This shows that estimation accuracy of the proposed DOA estimation algorithms improves when number of receivers are increased.

Improvement in estimation accuracy with increase in the number of receivers comes at the cost of significantly higher amount of FPGA resources consumed. Up to two sources can be estimated with minimum four receivers while estimation of three sources requires a minimum of eight receivers.

LabVIEW simulation of DOA estimation of three source employing eight receivers (M=8) using the proposed Cholesky and LDL decomposition methods is also considered. Simulation results are shown in Fig. 9 for the three sources located at 40°, 70°, and 110°, respectively. SNR = 10 dB.

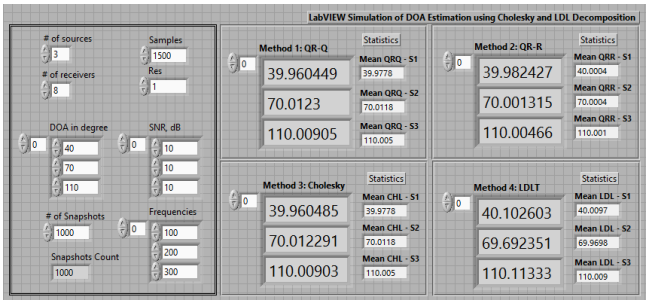


FIGURE 9. Screenshot of LabVIEW simulation UI for DOA estimation using proposed methods for three sources located at 40°, 70°, and 110°.

IV. IMPLEMENTATION OF PROPOSED ALGORITHMS ON FPGA

An FPGA platform is highly suitable for rapid prototyping and experimental validation of DOA estimation algorithms on real hardware. Moreover, an FPGA allows parallel execution of multiple operations unlike a single processor system. The proposed algorithms have been implemented for execution on NI FlexRIO 7965R module [32] featuring a Xilinx Virtex-5 SXT FPGA [33], using the pipelined architecture illustrated in Fig. 2. High throughput FPGA modules provided in LabVIEW were used for coding the proposed algorithms. Data size used in the implementation is fixed-point 16-bits/8-bits (word length/integer length) which has been found to be optimum in terms of resource consumption and computation time. LabVIEW implementation of Stage 2 of the pipeline in Fig. 2 for DOA estimation employing LDL and Cholesky decomposition is shown in Fig. 10 and Fig. 11, respectively. It can be observed that matrix L elements are computed in parallel during the matrix decomposition phase. Fig. 10 shows the computation of elements of matrix L for LDL decomposition according to (10) while Fig. 11 shows the computation of elements of matrix L for Cholesky decomposition according to (13).

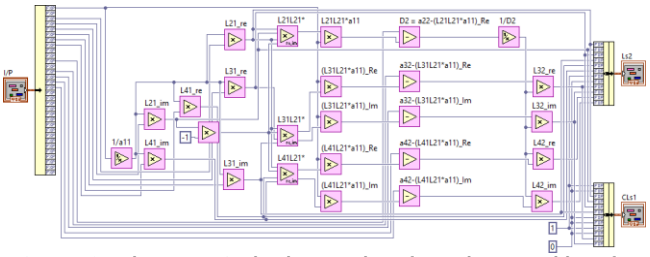


FIGURE 10. LabVIEW FPGA implementation of LDL decomposition of a 4x4 matrix and its partitioning into two submatrices.

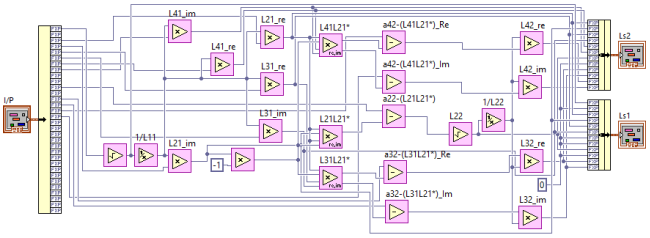


FIGURE 11. LabVIEW FPGA implementation of Cholesky decomposition of a 4x4 matrix and its partitioning into two submatrices.

A. FPGA RESOURCES UTILIZATION

LabVIEW FPGA VIs created for implementing proposed algorithms for computing the DOA estimates by executing the pipeline illustrated in Fig. 2 were compiled for the case of *M=4*. Table I shows the count of FPGA resources consumed for data size 16/8 and this is illustrated in Fig. 12 in terms of percentage device utilization of the maximum count available in Xilinx Virtex-5 FPGA device. These numbers are taken from a successful FPGA compilation report.

TABLE I
COUNT OF FPGA RESOURCES CONSUMED FOR DOA ESTIMATION USING PROPOSED LDL AND CHOLESKY, AND QR METHODS FOR DATA SIZE 16/8

FPGA Resource	QR-Q	QR-R	LDL	CHOL
Total Slices	9555	10846	8520	8757
Slice Registers	18778	22840	17344	17362
Slice LUTs	24820	30568	21870	21956
Block RAMs	10	10	10	10
DPS48s	270	418	233	230

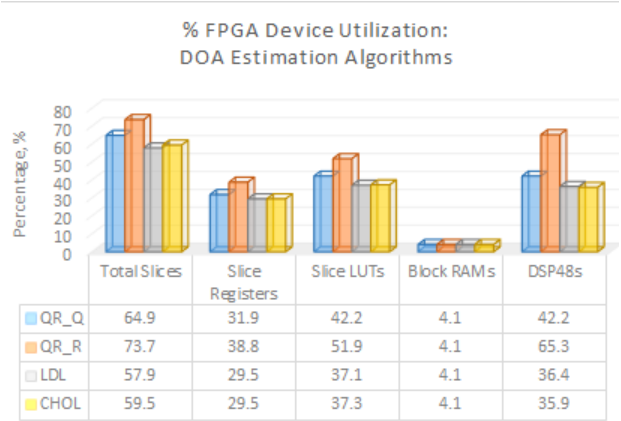


FIGURE 12. % FPGA Device Utilization for DOA estimation using proposed LDL and Cholesky methods for data size 16/8.

As seen in Fig. 12 for data size 16/8, LDL-based method consumes the least amount of resources. For example, it consumes 15.8% less Total Slices in the FPGA than QR-R and 7% less than QR-Q. It is clear that both proposed Cholesky and LDL-based methods are superior to QR in terms of resource requirements, with LDL holding a slight edge over Cholesky.

To study the effect of data size (word length) on resources consumption, FPGA codes were compiled for data sizes 12/6 and 20/10 as well. Table II and Fig. 13 show resources consumption as a percentage for three data sizes 12/6, 16/8, and 20/10.

TABLE II
% DEVICE UTILIZATION FOR DOA ESTIMATION USING PROPOSED LDL AND CHOLESKY FOR DATA SIZES 12/6, 16/8, AND 20/10

FPGA Resource	LDL 12/6	LDL 16/8	LDL 20/10	CHOL 12/6	CHOL 16/8	CHOL 20/10
Total Slices	55	57.9	62.4	54.8	59.5	62.9
Slice Registers	25	29.5	32.3	25	29.5	32.3
Slice LUTs	32.6	37.1	41.7	32.4	37.3	41.8
DPS48s	37.8	39.4	43.6	37.2	38.9	42.3

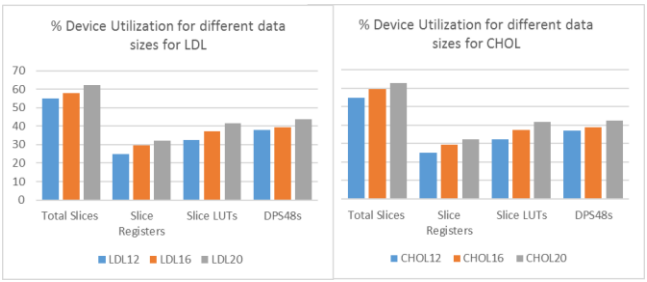


FIGURE 13. % FPGA Device Utilization for DOA estimation using proposed LDL and Cholesky methods for data sizes 12, 16, and 20

It can be observed from Table II and Fig. 13 that increasing the data size or word length results in significant increase in FPGA resources consumption for both LDL and Cholesky-based DOA estimation methods.

The computation speed in MHz estimated by the FPGA compiler Xilinx 14.7 (after a successful compilation) for the proposed methods as well as QR is shown in Fig 12. Cholesky is the fastest, followed by LDL. The onboard clock speed for FlexRIO 7965R is 40 MHz.

B. DOA COMPUTATION TIME

Computation time for the execution of the pipeline of Fig. 2 for the case *M=4* has been calculated for the proposed algorithms as well as QR for data size 16/8 as shown in Table III and for data size 20/10 as shown in Table IV. The tables show clock cycles consumed by each stage of the pipeline during runtime execution on the FPGA, and the computation speed in MHz. (taken from successful FPGA compilation report for each of the algorithms; with respect to the onboard base clock of 40 MHz). The time consumed during signal acquisition, phase calibration, FIFO read/write operations, and for other overheads has not been

considered in these tables. The computation time is calculated as:

Computation time = (Total No. of clock cycles)*(1/f_{max})

TABLE III
CLOCK CYCLES AND COMPUTATION TIME FOR DOA ESTIMATION USING LDL, CHOLESKY, AND QR FOR DATA SIZE 16 BITS

#	Pipeline Stage	QR-Q	QR-R	LDL	CHOL
1	Covariance Matrix computation	3	3	3	3
2	Matrix Decomposition	59	75	44	63
3	Least square solution	28	28	28	28
4	Eigen value decomposition (EVD)	76	76	76	76
5	Angle Estimation	24	24	24	24
	Total clock cycles	190	206	175	194
	Maximum frequency, f_{max}	57.7	53.3	59.4	63.0
	Computation time (µs)	3.29	3.86	2.95	3.08

TABLE IV
CLOCK CYCLES AND COMPUTATION TIME FOR DOA ESTIMATION USING LDL, CHOLESKY, AND QR FOR DATA SIZE 20 BITS

#	Pipeline Stage	QR-Q	QR-R	LDL	CHOL
1	Covariance Matrix computation	3	3	3	3
2	Matrix Decomposition	83	87	52	74
3	Least square solution	31	31	31	31
4	Eigen value decomposition (EVD)	88	88	88	88
5	Angle Estimation	24	24	24	24
	Total clock cycles	229	233	198	220
	Maximum frequency, f_{max}	51.56	48.46	52.5	54.33
	Computation time (µs)	4.44	4.81	3.77	4.05

It can be observed in the tables above that LDL is the fastest in computing the DOA estimates followed closely by Cholesky while QR-R is the slowest. Fig. 14 shows a comparison plot of computation time for data sizes 16 and 20 bits. It can be clearly seen that computation times for data size 20 bits are significantly higher than those for data size 16 bits.

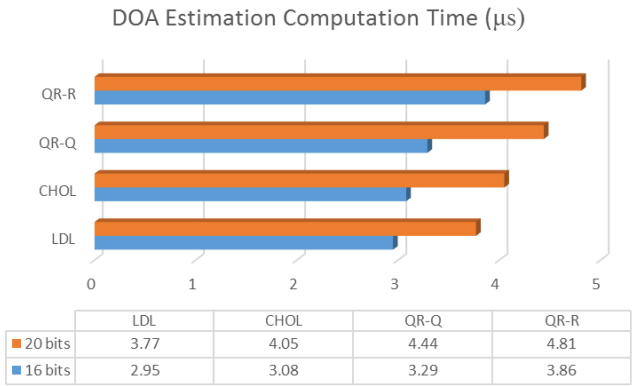


FIGURE 14. Computation time for DOA estimation using proposed LDL and Cholesky methods for data sizes 16 and 20 bits.

The effect of data size on computation speed as a function of maximum frequency (with respect to the base clock frequency of the FPGA) for the proposed LDL and Cholesky methods is shown in Fig. 15. It can be observed that while computation speed is decreasing with increase in data size, there is a significant decrease in computation speed going from data size 16/8 to 20/10.

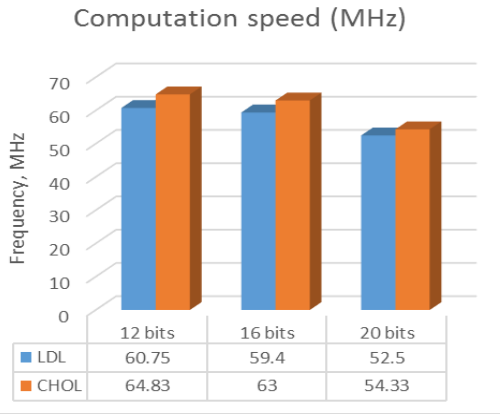


FIGURE 15. Computation speed in MHz for DOA estimation using proposed LDL and Cholesky methods for data sizes 12, 16, and 20 bits.

Overall, the proposed LDL and Cholesky-based DOA estimation algorithms have been found to be superior to QR-based algorithms in terms of resources utilization as well as computation speed. It has also been shown that data size 16/8 is optimum for implementation taking resources consumption and computation time into consideration. Effect of data size on estimation accuracy is presented in section VI.B where it will be shown that there is no appreciable increase in estimation accuracy beyond data size 16/8.

While FPGA allows for parallel execution of multiple operations, LabVIEW graphical programming is inherently parallel. These two factors make the implementation of proposed algorithms efficient and allow for fast computation of DOA estimates. It is worth mentioning here that due to the parallel nature of the implementation, the number of clock cycles required for DOA estimation of up to two sources ($K=2$) on the FPGA for the proposed algorithms for the case of 8-element ULA ($M=8$) were found to be same as those for the 4-element ULA ($M=4$) listed in Tables III and IV.

TABLE V
HARDWARE IMPLEMENTATION PERFORMANCE COMPARISON

Parameter	References					Proposed	
	[12]	[13]	[14]	[15]	[16]	Method 1	Method 2
Algorithm	Bartlett	Bartlett	MUSIC	MUSIC	QR	LDL	CHOL
Antenna elements	4 ULA	8 UCA	8 ULA	8 ULA	4 ULA	4 ULA	4 ULA
Target Hardware Device	Altera Cyclone IV	Virtex-5	Artix-7	Virtex-6	Virtex-5	Virtex-5	Virtex-5
Data size (bits)	8	-	-	16	16	16	16
Logic Elements Utilized	8467	3420	-	-	11317	9192	9253
Base Clock Frequency (MHz)	225	40	650	160	40	40	40
Number of clock cycles	181	1840*	-	-	-	175	194
Computation time (µs)	0.804	46	2560	93.4	24.38	2.95	3.08

*-estimated, UCA: uniform circular array

C. PERFORMANCE COMPARISON

Performance comparison of the proposed LDL and Cholesky-based FPGA implementation of DOA estimation with similar implementations reported in the literature [12-16] is presented in Table V. It can be deduced from the table that the proposed methods are superior to the implementations in [13-16] in terms of computation time. The implementation in [13] scores over the proposed methods in terms of resource consumption but it is 15 times slower with computation time of 46 µs.

It can also be seen that the proposed methods compare favorably with the fast Bartlett implementation in [12] which consumes only 181 clock cycles. With a 225 MHz base clock, the proposed LDL implementation will have a computation time of 0.78 µs which is slightly lower than that of [12]. The DOA estimator in [12] is shown to be an efficient implementation of the Bartlett algorithm on the FPGA. However, its drawback is that it is not truly real-time as it has a physically separate data collection unit which collects received signal data and saves it on a laptop before it is transferred serially to an antenna simulator and eventually to a DOA estimator. This implementation uses three FPGA boards and two laptop PCs for computing the DOA estimates.

V. USRP SDR TESTBED FOR REAL-TIME EXPERIMENTAL VALIDATION

A prototype testbed built using USRP SDR platform is employed for real-time experimental validation of the proposed DOA estimation algorithms, as shown in Fig. 16. The receiver setup shown consists of two USRP-2901 units used for receiving signals from the 4-element ULA and another USRP-2901 unit is used to generate a reference signal for phase synchronization. A multi-clock signal generation device CDA-2990 is used for generating timing signals for time synchronization.

Target signal (lying in the far-field region of the ULA) to be localized is generated using a USRP-2901 unit (not shown in Fig. 16). One USRP-2901 is required for each target signal. Target signal characteristics are: non-coherent source, minimum distance from ULA is 2 meters, frequency = 1 GHz, IQ rate = 500k, gain = 20 dB. Target data signals are received through the 4-element ULA connected to the USRP TX1/RX1 channels via USB 3.0 ports on a desktop PC.

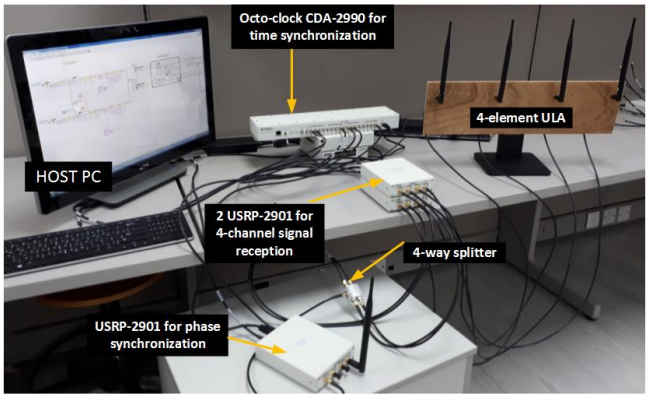


FIGURE 16. **Testbed for real-time experimental validation of proposed methods using a 4-element ULA.**

A. DOA ESTIMATION ON HOST PROCESSOR

Fig. 17 shows the receiver and signal processing block diagram using USRP-2901 for a 4-element ULA. The USRP unit first amplifies the received signal, downconverts it to baseband signals (I and Q), filters out noise and high frequency signals, and digitizes the signals (I and Q) before being passed on to the data/signal processor (Host PC).

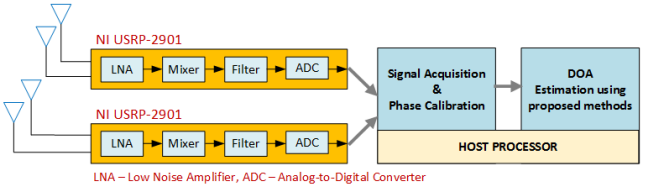


FIGURE 17. **Receiver block diagram for 4-element ULA with signal processing performed on host processor.**

Fig. 18 shows the hardware connections of the receiver testbed for a 4-element ULA (comprising of three USRP-2901 SDR units [31]) with signal processing performed on

a host processor (PC). Each antenna on the 4-element ULA is connected to four TX1/RX1 channels on the two USRP-2901 units. Each USRP-2901 unit supports two ports (one RX2 and another TX1/RX1 port) on each of the two channels. A third USRP-2901 is used to generate a reference signal for phase calibration which is fed into the RX2 ports on the two USRPs via a 4-way RF splitter. Connections are made using SMA cables of equal length to minimize phase difference between the receive channels.

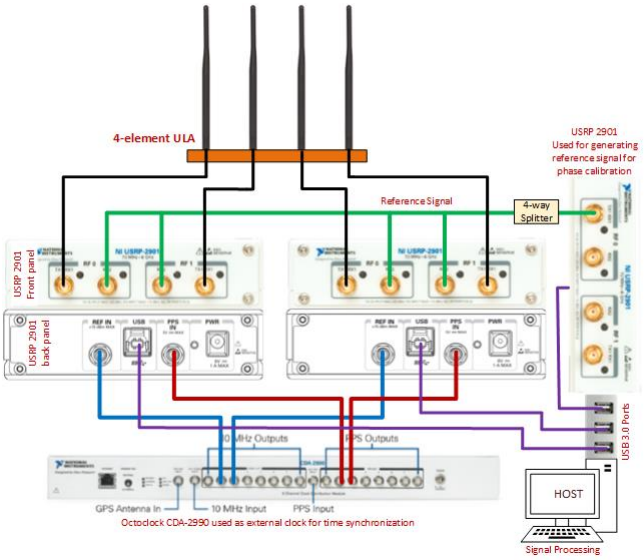


FIGURE 18. Connection diagram of USRP SDR receiver testbed with a 4-element ULA for real-time DOA estimation on the host processor.

B. DOA ESTIMATION ON FPGA

When FPGA is used for data/signal processing, the digitized signals are passed on to the FPGA by a real-time host controller through a FIFO (first-in-first-out) queue, as shown in Fig. 19.

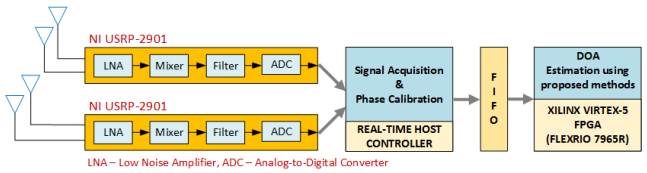


FIGURE 19. Receiver block diagram for 4-element ULA with signal processing performed on Xilinx Virtex-5 FPGA.

Fig. 20 shows the hardware connections of the receiver testbed for a 4-element ULA with signal processing performed on a Xilinx Virtex-5 FPGA. The USRPs are connected to the USB 2.0 ports on the real-time controller. Data signals from the ULA are transferred to the controller at a rate of 8M samples/second.

Increasing the number of channels would require more USRPs. However, due to limited number of USB ports available on the real-time controller shown in Fig. 20, the additional USRPs cannot be directly connected to the controller. A USB hub may be used but the data rate is reduced by a factor equal to the number of USRP devices connected to the USB hub. This problem also exists for a desktop PC with limited number of USB ports. Other models of USRPs may be used (such as USRP-2920) which have an Ethernet port allowing for high speed Gigabit Ethernet switches to be used for connecting multiple USRPs to the host computer.

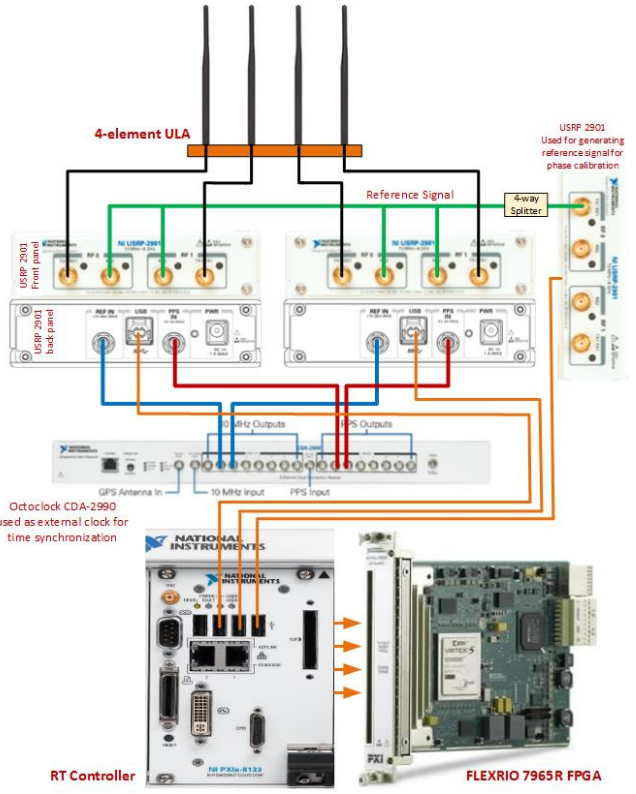


FIGURE 20. Connection diagram of USRP SDR receiver testbed with a 4-element ULA for real-time experiments on the Xilinx Virtex-5 FPGA.

C. TIME AND PHASE SYNCHRONIZATION

Before the target data signals can be acquired for computing DOA estimates which relies on the phase delay between receive channels, each USRP must be time and phase synchronized. Time synchronization is achieved through CDA-2990 module [34] which is a high accuracy 8-channel timing reference system. A 10 MHz REF signal (cyan color line) and a PPS (pulse per second) signal (maroon color line) generated by the CDA-2990 is connected to the REF IN and PPS inputs on each USRP-2901 in order to synchronize this 4-channel system to a common timing source.

Achieving phase synchronization is a non-trivial operation with USRPs. They do not share a local oscillator (LO), and this causes the phase to drift over time. For this reason, a phase calibration must be performed every time before data signals are acquired for processing. Phase synchronization can begin only after USRPs have been first synchronized in time successfully.

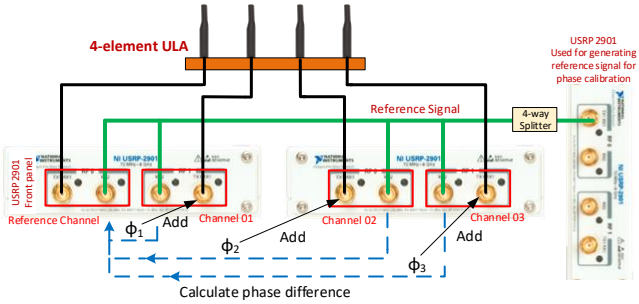


FIGURE 21. Phase Synchronization using a reference signal.

In the testbed shown in Fig. 16, phase synchronization is achieved through one USRP-2901 module which is used to generate a 10 kHz reference signal (up-converted to 1 GHz). As shown in Fig. 18 and Fig. 20, this reference signal (green color line) is fed into the RX2 channels on the USRPs. A LabVIEW VI (virtual instrument) code reads the reference signal and calculates the phase offset between the reference channel and each of the other receive channels. This phase offset is then added to the data signals received from the 4-channel ULA to achieve phase synchronization, as shown in Fig. 21. The system is now ready to compute the DOA estimates of the source signals. Fig. 22 shows the reference signals before and after synchronization for a 4-element ULA.

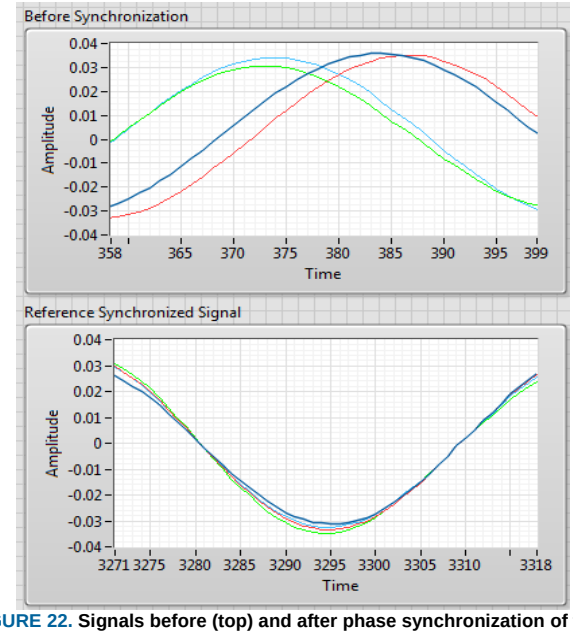


FIGURE 22. Signals before (top) and after phase synchronization of the reference signals for a 4-channel system.

Several problems were encountered during the process of phase synchronization of the USRPs, such as "command stream error" and "overflow error". Phase synchronization could not be performed without troubleshooting these errors, which were eventually resolved, as shown in Fig. 23, by appropriately setting the trigger time and trigger levels during task initiation of the USRP and before data fetch could begin.

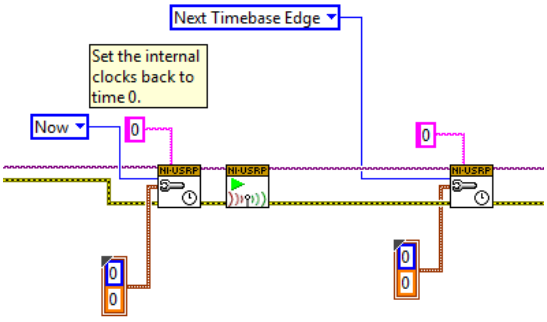


FIGURE 23. Managing triggering of USRPs before data fetch.

D. REAL-TIME DOA ESTIMATION OF MORE THAN TWO SOURCES ON HOST PROCESSOR

A testbed built for DOA estimation using an 8-element ULA is shown in Fig. 24. An 8-element ULA can be used for estimating more than two source signals. Since each USRP supports 2 channels, four USRP-2901 units are required for receiving the 8 signals from the 8-element ULA. The reference phase synchronization signal is fed to the four USRP-2901 units using one 2-way splitter and two 4-way splitters.

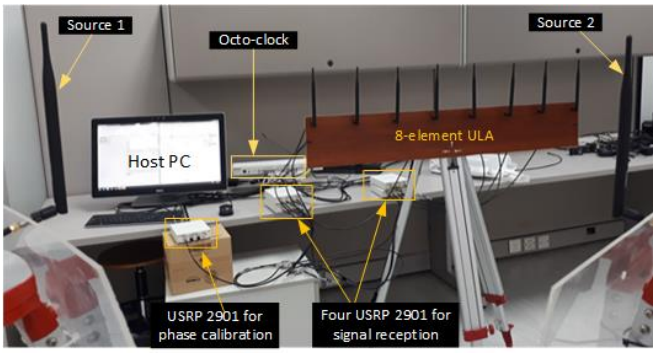


FIGURE 24. Testbed for real-time experimental validation of proposed methods using an 8-element ULA.

Logic resources available on the Xilinx Virtex-5 FPGA were found to be insufficient for implementation of DOA estimation algorithms for more than two sources and using a ULA of more than 4 elements. However, we have been able to successfully compile for DOA estimation of up to two sources using an 8-element ULA on the FPGA with the available resources almost maxed out (with 97.9% of Total Slices consumed for Cholesky-based DOA estimation algorithm).

VI. REAL-TIME DOA ESTIMATION RESULTS

The proposed algorithms implemented in LabVIEW are executed on the two USRP SDR prototype testbeds discussed in Section V above. Data signals acquired from the USRPs are first phase synchronized before being passed on to the execution pipeline illustrated in Fig. 2. The proposed algorithms are executed on the host PC as well as on Xilinx Virtex-5 FPGA (FlexRIO 7965R).

A. REAL-TIME DOA ESTIMATION ON HOST PROCESSOR

Fig. 25 shows a screenshot of the UI of the real-time DOA estimation program. Real-time reference signals received before synchronization are shown in the top left chart, and signals after synchronization are shown in the bottom left chart. Synchronized target (source) signals are shown in the top right chart. In the bottom right corner, real-time DOA estimates are shown for one source signal located at 55º from the 4-element ULA. DOA estimates are computed for LDL, Cholesky, and QR-based algorithms.

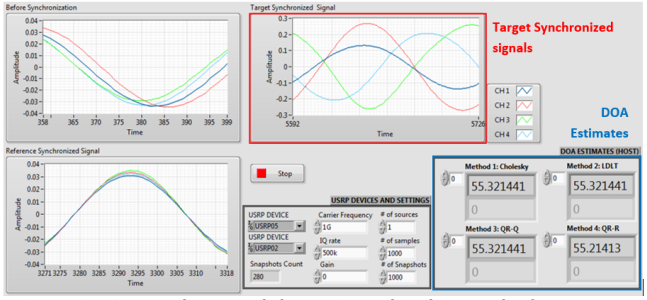


FIGURE 25. Screenshot - Real-time DOA estimation results for one source located at 55º from the 4-element ULA. Proposed algorithms are executed on the host processor (PC).

Fig. 26 shows the results for two sources located at 55º and 130º, respectively. Computations were performed for 1000 snapshots.

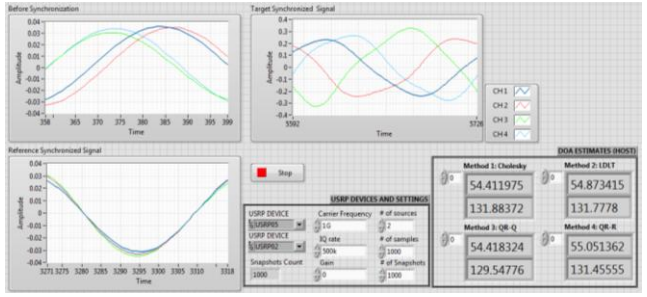


FIGURE 26. Screenshot - Real-time DOA estimation results for two sources located at 55º and 130º, respectively, from the 4-element ULA. Proposed algorithms are executed on the host processor (PC).

Table VI shows real-time DOA estimates on the host processor for two sources located at different angles from the ULA reference. Average and standard deviation values are calculated for each DOA estimate for 10 successful trials with 100 snapshots in each trial. The standard deviation value is calculated offline. To get an accurate value of standard deviation as a measure of estimation accuracy, the standard deviation is calculated with respect to the actual location of the source angle and not the average of the sample of DOA estimates obtained.

The estimation accuracy of proposed algorithms compares favorably with QR. Both Cholesky and LDL are better compared with QR-Q while QR-R has a slight edge over the proposed methods. but it consumes significantly higher number of FPGA resources and takes longer for computation of DOA estimates as indicated in Table I and Fig. 12, respectively.

The USRP testbed shown in Fig. 24 for real-time DOA estimation using an 8-element ULA can be used for the estimation of up to three sources. Results of real-time computation of DOA estimates of three source signals located at 50º, 90º, and 110º, repectively, is shown in Fig. 27.

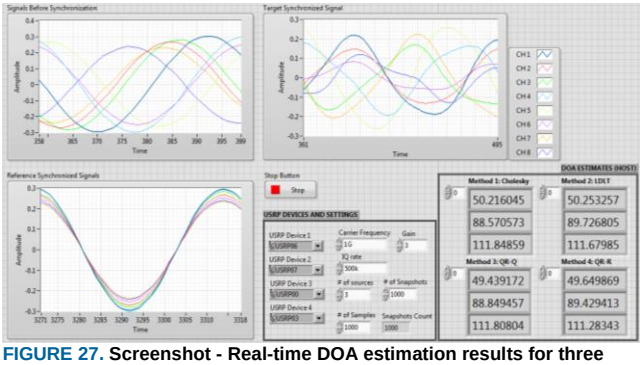


FIGURE 27. Screenshot - Real-time DOA estimation results for three sources located at 50º, 90º, and 110º, respectively, from the 8-element ULA. Proposed algorithms are executed on the host processor (PC).

As seen in Fig. 27, Cholesky and LDL-based algorithms fare better than QR-Q, but QR-R has higher estimation accuracy coming at a higher cost in terms of resources as well as computation time.

B. REAL-TIME DOA ESTIMATION ON FPGA

DOA estimates are also computed on the target FPGA for data size 16/8. After signals are acquired from the ULA and phase calibrated, they are passed on to the FPGA via a FIFO queue using direct memory access. Fig. 28 shows the DOA estimation results for proposed algorithms running on the FPGA. The two sources are located at 105º and 150º, respectively. Computations were performed for 10 iterations with 100 snapshots in each iteration.

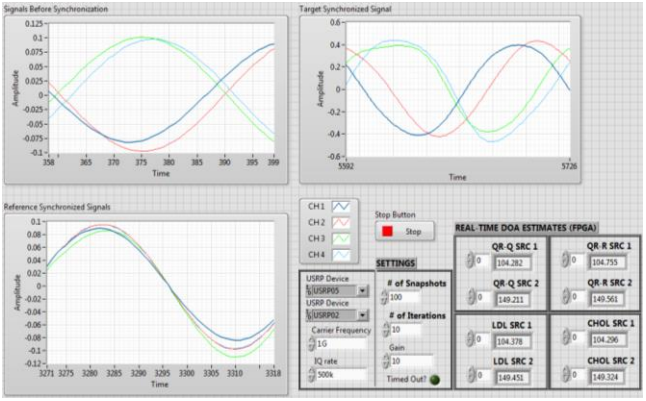


FIGURE 28. Screenshot - Real-time DOA estimation results for two sources located at 105º and 150º, respectively, from the 4-element ULA. Proposed algorithms are executed on the FPGA.

The performance comparison chart of proposed methods with QR in terms of RMSE vs. SNR is shown in Fig. 29. It can be seen that the proposed algorithms have better performance than QR-Q. Estimation accuracy of proposed algorithms running on the FPGA can be further improved by implementing the algorithms with a bigger data size such as 20/10. However, this improvement would come at

the cost of significant increase in resources consumption as well as computation time, as discussed in sections IV.A and IV.B above. In fact, FPGA compilation may fail due to resource and timing constraints.

TABLE VI
REAL-TIME DOA ESTIMATES OF TWO SOURCES COMPUTED ON HOST PROCESSOR USING PROPOSED AND QR-BASED METHODS. MEAN VALUES FOR 10 SUCCESSFUL ITERATIONS AND 100 SNAPSHOTS IN EACH.

Actual location: SRC1/SRC2	Real-time DOA Estimation							
	QR-Q		QRR		LDL		CHOL	
	Avg.	Std. Dev	Avg	Std. Dev	Avg	Std. Dev	Avg	Std. Dev
55°/130°	54.43°/129.37°	±0.56/±0.65	54.79°/129.65°	±0.22/±0.33	55.49°/129.41°	±0.50/±0.60	55.55°/130.45°	±0.54/±0.47
70°/110°	69.35°/109.44°	±0.64/±0.57	69.62°/110.35°	±0.39/±0.33	69.40°/109.47°	±0.63/±0.53	69.41°/110.43°	±0.60/±0.43
90°/120°	90.47°/119.55°	±0.55/±0.53	90.33°/119.57°	±0.32/±0.45	89.55°/120.55°	±0.46/±0.54	89.49°/120.45°	±0.53/±0.43
100°/135°	99.21°/134.49°	±0.81/±0.50	99.42°/135.29°	±0.61/±0.28	99.31°/134.46°	±0.70/±0.55	99.25°/134.53	±0.75/±0.49

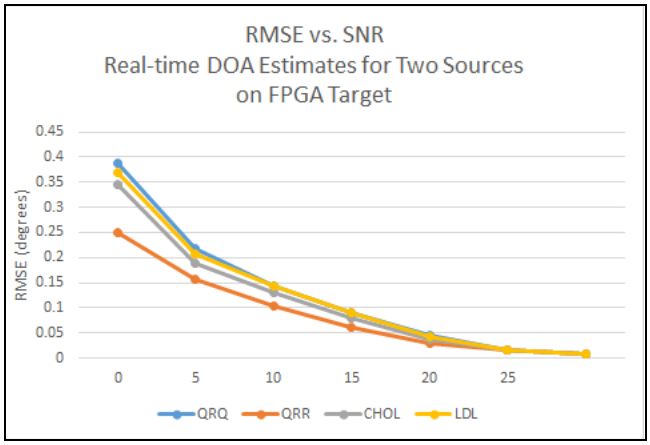


FIGURE 29. RMSE vs. SNR: FPGA Real-time DOA estimation performance comparison of proposed methods with QR for two sources (at 105° and 150°) and 4-element ULA.

Real-time experiments on FPGA target were also conducted to study the effect of varying data sizes on estimation accuracy for the proposed LDL and Cholesky-based implementations. Fig. 30 shows a chart depicting average relative error for data sizes 12/6, 16/8, and 20/10 in the DOA estimation of two sources located at 105° and 150° at an SNR of 10 dB. Relative error was calculated for DOA estimates for 10 iterations with 100 snapshots in each iteration. Experiments were conducted separately for each data size.

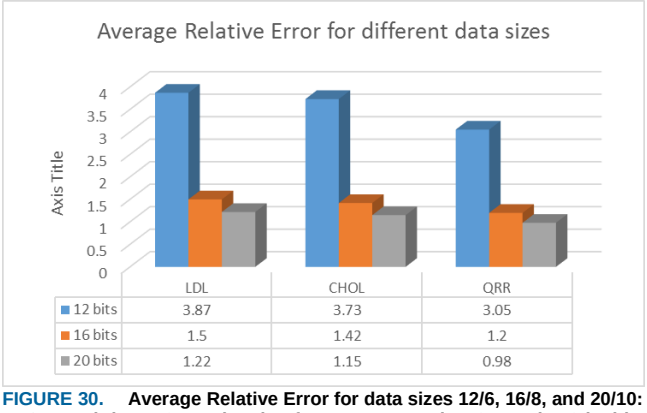


FIGURE 30. Average Relative Error for data sizes 12/6, 16/8, and 20/10: FPGA Real-time DOA estimation for two sources (at 105° and 150°) with 4-element ULA.

It can be noticed from the chart that there is significant improvement in estimation accuracy when data size increases from 12 to 16 bits. However, there is only a slight reduction in error going from 16 to 20 bits. This shows that data size 16/8 is optimum when considering that FPGA resources consumption and computation time increases with increase in data size (as discussed in section IV.A and IV.B).

VII. CONCLUSIONS

The work presented in this paper establishes the superior performance of proposed LDL and Cholesky-based DOA estimation algorithms for FPGA hardware implementation over existing methods reported in the literature. The proposed algorithms have been shown to be efficient for real-time hardware implementation in terms of resource requirements and computation time. The proposed algorithms have been also experimentally validated on a prototype testbed built using USRP SDR platform which is a low cost and scalable commercial off-the-shelf platform allowing rapid prototyping of systems for source localization, MIMO, etc. Overall, the proposed algorithms have been shown to be better for real-time practical applications when compared with QR-based estimation algorithms and other DOA methods reported in the literature.

ACKNOWLEDGEMENT

This research work was carried out at the Wireless Communications & Signal Processing Research Lab at Prince Mohammad bin Fahd University, Al Khobar, KSA.

REFERENCES

[1] R. O. Schmidt, "Multiple emitter location and signal parameter estimation," in IEEE Transactions on Antennas and Propagation, 1986, 34(3), pp.276-280.

[2] A. Paulraj, R. Roy, and T. Kailath, "Estimation of Signal Parameters Via Rotational Invariance Techniques- Esprit," in Nineteeth Asilomar Conference on Circuits, Systems and Computers, 1985., 1985, pp. 83–89.

[3] Roy, R., Kailath, T., "ESPRIT Estimation of Signal parameters via Rotational Invariance Techniques," IEEE Tran on Acoustics, Speech, and Signal Processing, vol. 29, no. 4, pp. 984-995, July 1989.

[4] 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," Progress In Electromagnetics Research, vol. 103, pp. 201–216, 2010.
[5] G.-M. Park and S.-Y. Hong, "Resolution Enhancement of Coherence Sources Impinge on a Uniform Circular Array with Array Expansion," Journal of Electromagnetic Waves and Applications, vol. 21, no. 15, pp. 2205–2214, Jan. 2007.
[6] Barabell, A.J., "Improving the Resolution Performance of Eigenstructure Based Direction Finding Algorithms," Proceedings of the ICASSP-83, pp. 336-339, 1983.
[7] L. Osman, I. Sfar, and A. Gharsallah, "Comparative Study of High-Resolution Direction-of-Arrival Estimation Algorithms for Array Antenna System," vol. 2, no. 1, pp. 72–77, 2012.
[8] NI PXI platform, http://www.ni.com/pxi/
[9] NI USRP SDR platform, http://www.ni.com/en-lb/shop/select/usrp-software-defined-radio-device
[10] N. Tayem, "Real time implementation for DOA estimation methods on NI-PXI platform," Progress In Electromagnetics Research B, Vol. 59, 103-121, 2014.
[11] N. Tayem, M. Omer, M. El-Lakkis, S. A. Raza , J. Nayfeh, "Hardware Implementation of a Proposed QR-TLS DOA Estimation Method and Music, Esprit Algorithms on NI-PXI Platform," Journal of Progress In Electromagnetics Research C, Vol. 45, 203-221, November 2013.
[12] Unlersen, Fahri M., Yaldiz, Ercan; Imeci, Sehabeddin T., "FPGA Based Fast Bartlett DoA Estimator for ULA Antenna Using Parallel Computing," Applied Computational Electromagnetics Society Journal, April 2018, Vol. 33 Issue 4, p450-459. 10p.
[13] 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," 2010 IEEE Aerospace Conference, pp. 1-10, 6-13 Mar. 2010.
[14] M. Devendra and K. Manjunathachari, "Direction of arrival estimation using MUSIC algorithm in FPGA: Hardware software co-design," International Journal of Applied Engineering Research, vol. 11, no. 5, pp. 3112-3116, 2016.
[15] J. Yan, Y. Huang, H. Xu, G. A. E. Vandenbosch, "Hardware acceleration of MUSIC based DoA estimator in MUBTS," in the 8th European Conference on Antennas and Propagation (EuCAP 2014), 2014, pp. 25612565
[16] Abdulrahman Alhamed, Nizar Tayem, Tariq Alshawi, Saleh Alshebeili, Abdullah Alsuwailem, Ahmed Hussain, "FPGA-based Real Time Implementation for Direction-of-Arrival Estimation," The Journal of Engineering, 2017, 13 pp., doi: 10.1049/joe.2017.0165
[17] 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," in *IEEE Access*, vol. 6, pp. 17666-17680, 2018. doi: 10.1109/ACCESS.2018.2820122
[18] Donggu Kim, Seongah Jeong, Kwang Eog Lee, and Joonhyuk Kang, "Performance Analysis of AOA-based Localization with Software Defined Radio," in International Global Navigation Satellite Systems Society (IGNSS) Symposium. Gold Coast, QLD, Australia, July 2015.
[19] Chen, H., Lin, T., Kung, H.T., Lin, C., & Gwon, Y., "Determining RF angle of arrival using COTS antenna arrays: A field evaluation," *MILCOM 2012 - 2012 IEEE Military Communications Conference*, 1-6.
[20] B. Rares *et al.*, "Experimental Evaluation of AoA Algorithms using NI USRP Software Defined Radios," *2018 17th RoEduNet Conference: Networking in Education and Research (RoEduNet)*, Cluj-Napoca, 2018, pp. 1-6. doi: 10.1109/ROEDUNET.2018.8514133
[21] A.D. Redondo, T. Sanchez, C. Gomez, L. Betancur, R.C. Hincapie, "MIMO SDR-based implementation of AoA algorithms for Radio Direction Finding in spectrum sensing activities," in IEEE Colombian Conference on Communications and Computing (COLCOM), pp.1-4, 13-15, May 2015.
[22] V. Goverdovsky, D. C. Yates, M. Willerton, C. Papavassiliou, E. Yeatman, "Modular software-defined radio testbed for rapid prototyping of localization algorithms," in IEEE *Trans. Instrum. Meas.*, vol. 65, pp. 1577-1584, Jul. 2016.
[23] A. Akindoyin, M. Willerton, A. Manikas, "Localization and array shape estimation using software defined radio array testbed," in IEEE 8th Sensor Array and Multichannel Signal Processing Workshop (SAM), A Coruna, pp. 189-192, June 2014.
[24] Ettus Research Application Note – Synchronization and MIMO Capability with USRP Devices. https://kb.ettus.com/Synchronization_and_MIMO_Capability_with_USRP_Devices
[25] NI White paper - Building an Affordable 8x8 MIMO Testbed with NI USRP. April 2015. http://www.ni.com/white-paper/14311/en/
[26] Golub, G.H., Van Loan, C.F.: Matrix computations (Johns Hopkins University Press, London, 2013, 4th edn.)
[27] Saleh O. Al-Jazzar, "Angle of Arrival Estimation Using Cholesky Decomposition," International journal of antenna and propagation,pp.1-6, 2012
[28] Nizar Tayem, "Cholesky Factorization Based Parallel Factor for Azimuth and Elevation Angles Estimation," accepted at Arabian Journal for Science and Engineering, June 2017. doi: 10.1007/s13369-017-2678-9
[29] NI LabVIEW software platform, http://www.ni.com/labview/
[30] NI LabVIEW FPGA Module, http://www.ni.com/labview/fpga/
[31] USRP SDR 2901: http://www.ni.com/en-lb/support/model.usrp-2901.html
[32] PXIe-7965 PXI FPGA Module for FlexRIO, http://www.ni.com/en-lb/support/model.pxie-7965.html
[33] Xilinx Virtex-5 SXT FPGA data sheet: https://www.xilinx.com/support/documentation/selection-guides/virtex5-product-table.pdf
[34] Octo-clock CDA-2990 8-Channel Clock Distribution Module: https://www.ettus.com/product/details/OctoClock-G