Matrix Decomposition Methods for Efficient Hardware Implementation of DOA Estimation Algorithms: A Performance Comparison

Ahmed A. Hussain   
Deptartment of Electrical Engineering  
Prince Mohammad bin Fahd UniversityAlkhobar, KSA  
ahussain1@pmu.edu.sa Nizar Tayem  
Department of Engg.pandpTechnology  
TexaspAp& MpUniversity *-pCommerce*Texas,pUSA  
[Nizar.Tayem@tamuc.edu](mailto:Nizar.Tayem@tamuc.edu)Abdel-Hamid Soliman  
School of Creative Arts & Engineering  
Staffordshire Univ. – Stoke-on-TrentUnited Kingdom  
[a.soliman@staffs.ac.uk](mailto:a.soliman@staffs.ac.uk)

*Abstract*— Matrix operations form the core of array signal processing algorithms such as those required for direction of arrival (DOA) angle estimation of radio frequency signals incident on an antenna array. In this paper, we present a performance comparison of matrix decomposition methods for efficient FPGA hardware implementation of DOA estimation algorithms. These methods are very important in subspace-based DOA estimation algorithms as they are used for signal space extraction. DOA estimation algorithms employing LU, LDL, Cholesky, and QR decomposition methods are implemented on a Xilinx Virtex-5 FPGA. These DOA estimation algorithms are simulated in LabVIEW as well as experimentally validated in real-time on a prototype testbed constructed using Universal Software Radio Peripheral (USRP) Software Defined Radio (SDR) platform from National Instruments. Performance comparison of these algorithms is made in terms of resources consumption, computation speed, and estimation accuracy.

Keywords— matrix decomposition, array signal processing, field programmable gate arrays (FPGAs), DOA estimation, LabVIEW, LabVIEW FPGA, USRP

# Introduction

Array signal processing forms an integral part of myriad applications including mobile communications, multiple-input and multiple-output (MIMO) systems, source localization and DOA estimation, radars, beamforming, etc. Matrix operations such as inversion, factorization/decomposition, and multiplication with complex-valued signal data are at the core of these signal processing algorithms.

Most practical applications require the DOA estimates to be computed in real-time and, hence, their hardware implementation and their subsequent experimental validation is essential. Computational complexity of a DOA estimation algorithm is an important consideration in the hardware implementation of the algorithm, directly impacting computation time and resource requirements. The computation cost and efficiency of these algorithms is driven by the complexity of the matrix operations used and the size of the matrices.

Subspace-based DOA estimation algorithms widely cited in the literature such as multiple signal classification (MUSIC) [1], estimation of signal parameters via rotational invariant techniques (ESPRIT) [2] and their different variants [3-6] use either eigen value decomposition (EVD) or singular value decomposition (SVD) to separate the signal space from the noise space. However, both EVD and SVD methods are compute intensive as they require O(N3) operations where N is the size of the received data matrix. A majority of the work reported in the literature in the area of DOA estimation relies on numerical analysis for validation; only a few works have reported hardware implementation and experimental validation.

Real-time implementation of a DOA estimation algorithm based on QR decomposition using FPGA presented in [7] was shown to be superior to EVD/SVD-based methods [8-9], the performance indicators being accuracy of estimation, speed of computation, and consumption of resources. Similar works reported in [10], [11], and [12] use LU, LDL, and Cholesky factorization, respectively, for signal space extraction in the real-time FPGA implementation of DOA estimation algorithms. These methods were shown to be superior to QR-based algorithms (QR-Q and QR-R) in terms of the performance indicators mentioned above. However, QR-R was shown to have higher estimation accuracy.

This paper focuses on the following: 1) matrix decomposition methods such as LU, LDL, Cholesky, and QR employed in DOA estimation algorithms; 2) FPGA hardware implementation of DOA estimation algorithms employing these matrix decomposition methods; and 3) performance comparison of these DOA estimation algorithms in terms of (a) estimation accuracy, (b) computation speed, and (c) resources consumption. It will be shown in the following sections that LU is the most efficient with respect to (b) and (c). While QR-R is the most accurate, it requires a significantly large amount of resources and is also the slowest in computing the DOA estimates.

# SYSTEM MODEL FOR DOA ESTIMATION

The system model depicted in Fig. 1 shows an array of uniformly spaced antenna elements called a uniform linear array (ULA). This ULA consists of eight antennas (*M=8*) which are omni-directional. These antenna elements are placed 15 cm apart  which is equivalent to the wavelength of a 1 GHz signal. Up to two sources (*K = 1, 2*) located at angles , respectively, are assumed to be lying in the far-field region of the ULA. Received RF signals at the antenna array are first down-converted to baseband frequency, filtered, and then converted to digital signals before being sent for further processing. The DOA estimates are computed using a DOA estimation algorithm employing one of the several matrix decomposition methods considered in this paper.



Fig. 1. System model for DOA estimation of up to two sources in the far-field region of an antenna array.

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

**** (1)

where  is the *i-*th incident source signal,  is the wavelength,  the spacing distance of ULA, and  is the noise at the *m*-th element.

The received RF data signal can be expressed as:

, (2)

where  is the (*M* x *K*) array response matrix given by:

, (3)

and for is the corresponding array response vector.

 (4)

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

, (5)

and , (6)

where N(t) is the  additive white Gaussian noise (AWGN) vector.

# Overview of Matrix Decomposition Methods

Matrix operations and linear algebra techniques form the core of array signal processing operations such as those used in algorithms for DOA estimation. These algorithms are inherently complex due to complex-valued matrix operations and other compute intensive operations such as division, multiplication, and square root, etc. As the number of receive antennas are increased, the size of the matrices increases proportionally. In subspace-based DOA estimation algorithms, matrix decomposition techniques are used to separate the noise space from the signal space. Commonly used matrix decomposition methods reported in the literature are described in the following paragraphs.

Singular Value Decomposition (SVD) factorizes a matrix  as , where  are orthonormal columns, are singular values of A, and  are orthogonal columns. ESPRIT technique [13-15], a subspace technique based on either SVD or EVD, is an efficient and robust technique of DOA estimation. Instead of exhaustive search over possibility space, ESPRIT reduces computational complexity and storage requirements [13] compared with MUSIC. However, it suffers from scalability issues, complexity, and high computational time which makes it unsuitable for hardware implementation for high data speed wireless communication system.

Eigen Value decomposition (EVD) factorizes a matrix A with dimension M x M such that where are the M eigenvalues andare the corresponding M eigen vectors. In array signal processing, EVD is used in decomposing the data correlation matrix for separating noise subspace from the signal subspace. These signal and noises subspaces can be used for parameter estimation such as Direction of arrival estimation (DOA). MUSIC [1], [13] is a subspace algorithm based on EVD which is commonly used in the area of array signal processing. In addition, it requires an extensive search procedure for DOA estimation over the space of elevation angle which tends to be computationally expensive. Even efficient variants of MUSIC such as ROOT-MUSIC [6], [13] involve searching for polynomial roots, which is still too expensive for real-time hardware implementation.

QR decomposition [15] which is a popular method decomposes a matrix *A* as *A=QR* , *Q* being an orthogonal matrix while *R* an upper triangular matrix. In DOA estimation algorithms, QR decomposition is applied to the received data correlation matrix to separate the noise space from the signal space. The Modified Gram-Schmidt [16] is an efficient numerical method used for calculating the elements of the *R* and *Q* matrices. QR decomposition-based methods offer a computationally efficient alternative to SVD or EVD-based methods such as ESPRIT and MUSIC [13-15]. QR also consumes less resources, relatively, making it suitable for hardware implementation.

LU factorization [15] is another matrix decomposition method that decomposes a matrix *A* as *A = LU*. *L* is a lower triangular matrix in which diagonal elements are all one (1) while U is upper triangular. In DOA estimation algorithms, LU factorization applied to the data correlation matrix is used for separating the noise space from the noise space [17]. An efficient hardware implementation of LU-based DOA estimation algorithm has been reported in [10].

Cholesky decomposition factorizes a matrix A, into a matrix, G and its conjugate transpose, such that, where matrix A should be a positive-definite. A positive definite matrix is one that is always nonsingular and its determinant |A| is always greater than zero. In comparison with Gaussian elimination, Cholesky decomposition technique only requires half of the number of operations and half the memory space [15], [18-19].

LDL decomposition decomposes a matrix A as *A = LDL\** (D is diagonal and L is lower triangular matrix). LDL is a close variant of Cholesky decomposition The efficiency of Cholesky decomposition is also about two times that of LU decomposition for solving a set of linear equations [19]. LU is particularly useful in implementation of DOA estimation algorithms due to its lower complexity compared with the conventional subspace techniques.

# FPGA Hardware Implementation

This section presents FPGA hardware implementation of DOA estimation algorithms employing aforementioned matrix decomposition methods (viz. QR, LU, LDL, and Cholesky). The FPGA platform selected for implementation is an NI FlexRIO 7965R module [20] with a Xilinx Virtex-5 SXT FPGA [21]. The pipelined architecture depicted in Fig. 2 was used for implementing the DOA algorithms. The implementation was carried out using a word length of 16-bits/8-bits (word length/integer length) with fixed-point data.

Fig. 2.  Five-stage execution pipeline for DOA estimation

The five stages of the pipeline are listed below:

Stage 1: Compute covariance matrix

Stage 2: Extract signal space from the covariance matrix by performing matrix decomposition

Stage 3: Find the direction matrix using the least squares approach

Stage 4: Compute eigen values of the direction matrix

Stage 5: Compute angle estimates using the eigen values

In the first stage, the covariance matrixis computed, using the N snapshots of the RF signal received from the ULA, according to the following equation:

 (7)

where is the column vector from the ith antenna element. The computed matrix is shown below:

 (8)

In Stage 2, the covariance matrix computed in Stage 1 is decomposed to extract the signal space by applying QR, LU, LDL or Cholesky factorization. For estimation of DOA angles of up to two sources, it suffices to compute only the first two columns of ***Q*** or ***L*** (or the first two rows of R in (9) or the first two rows of U in (10)). Matrix decomposition for each method was implemented in LabVIEW [22] using LabVIEW FPGA [23] high throughput functions.

QR decomposition of is performed as shown below:

 (9)

where the elements of ***Q*** and ***R*** are computed using the Modified-Gram-Schmidt procedure [16]. Implementation of QR decomposition of a 4x4 matrix using LabVIEW FPGA and computation of the first two columns of matrix ***Q*** is shown in Fig. 3.

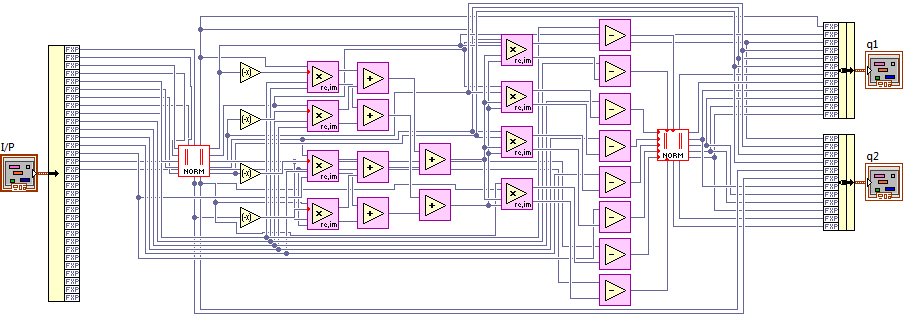


Fig. 3.  Schematic showing QR decomposition of a 4x4 matrix and generation of *Q* matrix using LabVIEW FPGA.

LU factorization of matrix ***Rxx*** is performed as shown below:

 (10)

The entries of ****** and are computed as given below:

 (11)

Implementation of LU decomposition of a 4x4 matrix using LabVIEW FPGA and computation of the first two columns of matrix ***L*** is shown in Fig. 4.

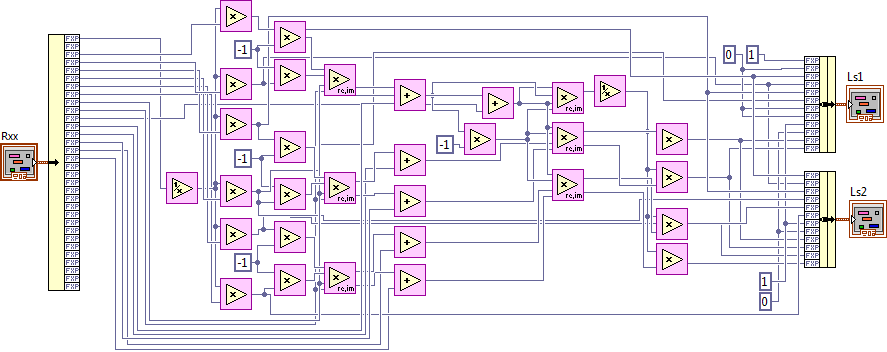


Fig. 4.  Schematic showing LU decomposition of a 4x4 matrix and generation of *L* matrix using LabVIEW FPGA.

LDL decomposition of matrix  is performed as shown below:

 (12)

Elements of matrices ***D*** and ***L*** are computed according to the following equations:

 (13)

LDL factorization of a 4x4 matrix using LabVIEW FPGA and computation of the first two columns of matrix ***L*** is shown in Fig. 5.

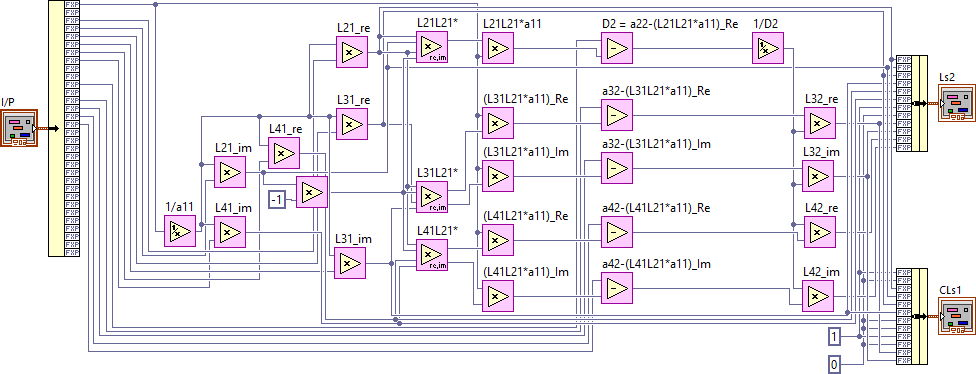


Fig. 5 Schematic showing LDL decomposition of a 4x4 matrix using LabVIEW FPGA.

Cholesky factorization of matrix  is performed as shown below:

 (14)

where  is given by:

 (15)

The entries of ***L***, where , are computed according to the following equations:

 (16)

The Cholesky decomposition of a 4x4 matrix using LabVIEW FPGA and computation of the first two columns of matrix L is shown in Fig. 6.

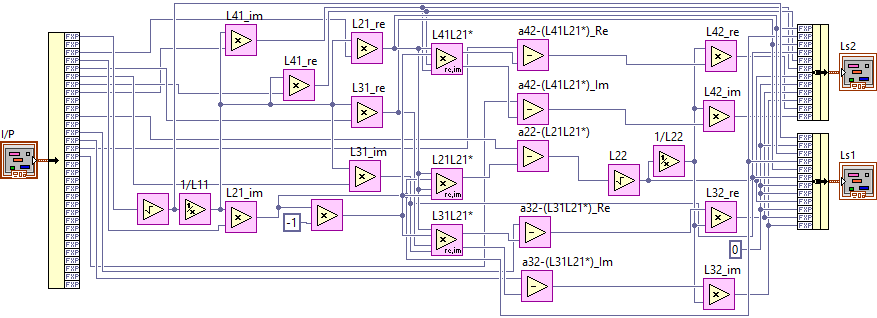


Fig. 6.  Schematic showing Cholesky decomposition of a 4x4 matrix using LabVIEW FPGA.

Tables I and II show the count of arithmetic operations required for FPGA implementation of different matrix decomposition methods considered here for DOA estimation algorithms for the two cases of M=4 and M=8, respectively. It can be observed from the table that LU-U requires the least number of arithmetic operations for computing the DOA estimates while QR-R requires the most.

TABLE I. COUNT OF ARITHMETIC OPERATIONS FOR M=4

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Arithmetic**  **Operations** | **4 x 4 Matrix (M = 4 antennas)** | | | | | |
| **QRQ** | **QRR** | **LUL** | **LUU** | **LDL** | **CHOL** |
| Addition | 20 | 69 | 6 | 6 | 0 | 0 |
| Subtraction | 0 | 0 | 0 | 0 | 5 | 5 |
| Multiplication | 0 | 16 | 17 | 4 | 16 | 11 |
| Complex-valued  Multiplication | 8 | 41 | 6 | 3 | 3 | 3 |
| Division | 16 | 0 | 2 | 1 | 2 | 2 |
| Square Root | 2 | 2 | 0 | 0 | 0 | 2 |
| **Total # of Operations** | 46 | 128 | 31 | 14 | 26 | 23 |

TABLE II. COUNT OF ARITHMETIC OPERATIONS FOR M=8

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Arithmetic**  **Operations** | **8 x 8 Matrix (M = 8 antennas)** | | | | | |
| **QRQ** | **QRR** | **LUL** | **LUU** | **LDL** | **CHOL** |
| Addition | 44 | 224 | 14 | 15 | 0 | 0 |
| Subtraction | 16 | 16 | 0 | 0 | 13 | 13 |
| Multiplication | 32 | 32 | 41 | 4 | 40 | 27 |
| Complex-valued  Multiplication | 16 | 112 | 14 | 7 | 7 | 7 |
| Division | 2 | 0 | 2 | 1 | 2 | 2 |
| Square Root | 2 | 2 | 0 | 0 | 0 | 2 |
| **Total # of Operations** | 112 | 386 | 71 | 27 | 62 | 51 |

## FPGA Resources Consumption

Tables III and IV show the resources consumption in the FPGA implementation QR, LU, LDL, and Cholesky matrix decomposition methods. Fig. 7 depicts device utilization as a percentage of the total count available in the FPGA.

TABLE III. DEVICE UTILIZATION FOR M=4

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **4 x 4 Matrix (M = 4 antennas)** | | | | | |
| **FPGA Resource** | **QRQ** | **QRR** | **LUL** | **LUU** | **LDL** | **CHOL** |
| Total Slices | 3119 | 5168 | 2302 | 2304 | 2469 | 2027 |
| Slice Registers | 7426 | 10830 | 5503 | 5208 | 5325 | 5247 |
| Slice LUTs | 8194 | 13074 | 5432 | 4590 | 5006 | 4891 |
| Block RAMs | 2 | 2 | 2 | 2 | 2 | 2 |
| DSP48s | 48 | 176 | 41 | 16 | 28 | 23 |

TABLE IV. DEVICE UTILIZATION FOR M=8

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **8 x 8 Matrix (M = 8 antennas)** | | | | | |
| **FPGA Resource** | **QRQ** | **QRR** | **LUL** | **LUU** | **LDL** | **CHOL** |
| Total Slices | 4725 | 12040 | 3756 | 3653 | 3564 | 3572 |
| Slice Registers | 12026 | 26659 | 10105 | 9865 | 9808 | 9617 |
| Slice LUTs | 10724 | 35700 | 8386 | 7567 | 7805 | 7645 |
| Block RAMs | 2 | 2 | 2 | 2 | 2 | 2 |
| DSP48s | 128 | 512 | 95 | 30 | 66 | 53 |

It can be observed from the tables above that LU-U and Cholesky consume the least amount of FPGA resources while the highest amount of resources is consumed by QR-R. For M=8 case, Cholesky consumes 3572 Total Slices out of 14720 and 9617 Slice Registers out of 58880, and LU-U consumes 7567 Slice LUTs (look-up tables) out of 58880 and 30 DSP48s out of 640. In contrast (on the higher side), QR-R consumes 12040 Total Slices, 26659 Slice Registers, 35700 Slice LUTs, and 512 DSP48s.

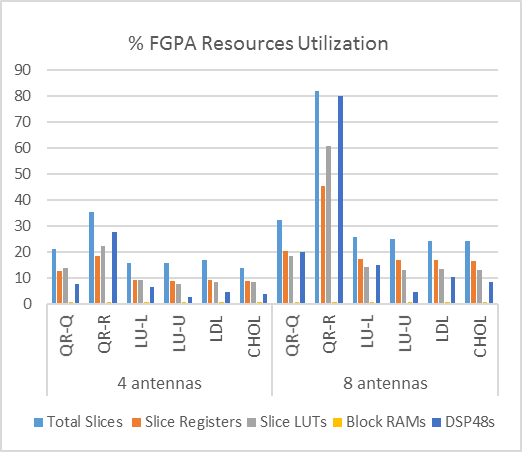


Fig. 7. % Device Utilization for M=4 and M=8 cases.

It can be deduced from Fig. 7 that, overall, LU, LDL, and Cholesky methods fare better in resources consumption compared with QR decomposition.

## DOA Estimates Computation Time

Tables V and VI show the computation time (in **μs)** for matrix decomposition using QR, LU, LDL, and Cholesky methods for the two cases of M=4 and M=8, respectively. The computation time is calculated as the product of number of clock cycles and the maximum frequency. The longest propagation path in the LabVIEW Virtual Instrument (VI) schematic is used in calculating the number of clock cycles taken by each of the decomposition methods. Maximum computation speed in MHz shown in the tables is extracted from the report of a successfully completed FPGA compilation for each method. It is also worth noting here that the number of clock cycles for both M=4 and M=8 are almost same for each method, respectively. This is due to the fact that the matrix operations were implemented to be executed in parallel.

TABLE V. COMPUTATION TIME M=4

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **4 x 4 Matrix (M = 4 antennas)** | | | | | |
| **QRQ** | **QRR** | **LUL** | **LUU** | **LDL** | **CHOL** |
| # of clock cycles | 59 | 75 | 22 | 20 | 44 | 63 |
| Max. freq. (MHz) | 76.6 | 65.5 | 78.6 | 67.7 | 73.1 | 71.8 |
| **Computation time (μs)** | 0.77 | 1.15 | **0.28** | **0.30** | 0.60 | 0.88 |

TABLE VI. COMPUTATION TIME M=8

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **8 x 8 Matrix (M = 8 antennas)** | | | | | |
| **QRQ** | **QRR** | **LUL** | **LUU** | **LDL** | **CHOL** |
| # of clock cycles | 60 | 75 | 25 | 23 | 44 | 64 |
| Max. freq. (MHz) | 65.0 | 40.4 | 60.7 | 55.5 | 69.0 | 64.9 |
| **Computation time (μs)** | 0.92 | 1.86 | **0.41** | **0.41** | 0.64 | 0.99 |

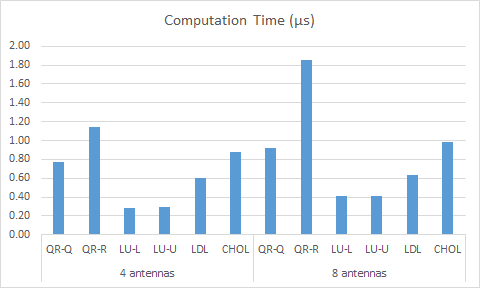


Fig. 8. Computation time for matrix decomposition for M=4 and M=8 cases.

It can be observed from the tables V and VI and Fig. 8 that LU decomposition method is the fastest in extracting the signal space for both M=4 and M=8 cases while QR-R is the slowest.

# DOA ESTIMATION RESULTS

## The DOA estimation results obtained through LabVIEW simulations as well as through real-time experiments on a prototype testbed are presented in this section.

## LabVIEW Simulation Results

The DOA estimation algorithms employing QR, LU, LDL, and Cholesky decomposition methods were implemented using linear algebra functions in LabVIEW prior to FPGA hardware implementation. Fig. 9 shows the screenshot of simulation results of DOA estimation algorithms. The LabVIEW user interface (UI) shows two sources located at 55o and 90o, respectively. SNR = 15 dB, M=4, and snapshots=500.

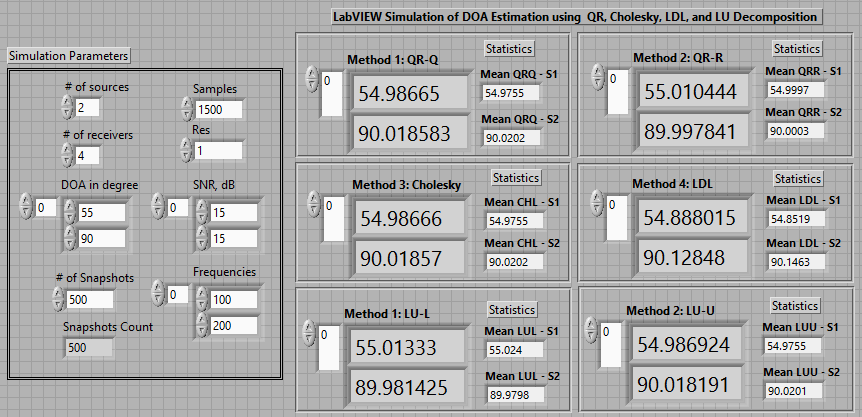


Fig. 9 Screenshot of UI showing LabVIEW simulation results for two sources located at 55o and 90o, respectively.

A performance comparison of the DOA estimation algorithms in terms of root mean square error (RMSE) vs. signal-to-noise ratio (SNR) is shown in Fig. 10 for estimating two sources located at 55o and 90o, respectively. SNR values range from 0 dB to 25 dB and 500 snapshots are used. M=4 receivers are considered. It is clear from this graph that QR-R has significantly lower RMSE values compared with the other methods especially at low SNR values up to 10 dB. The rest of the methods have similar performance characteristics throughout with Cholesky have a slight edge over them.

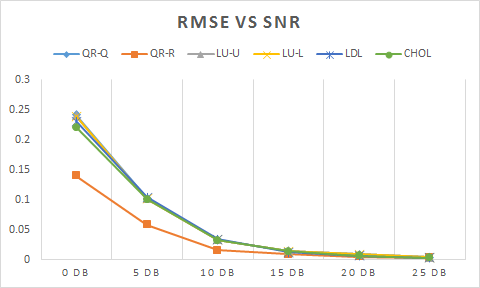


Fig. 10. RMSE vs. SNR performance comparison of DOA estimation.

## Real-time Experimental Results

Real-time experimental validation of the DOA estimation algorithms was performed on a prototype testbed constructed using USRP-2901 units [24-25] from NI, a real-time host controller, and NI FlexRIO 7965R FPGA. The prototype testbed for estimating up to two sources using a ULA of 4-elements is shown in Fig. 11. Two USRP 2901 units are needed for signal acquisition while another unit is used for phase synchronization. An Octo-clock CDA 2990 [26] unit is used for time synchronization. Testbed setup and phase/time synchronization details can be seen in [12].

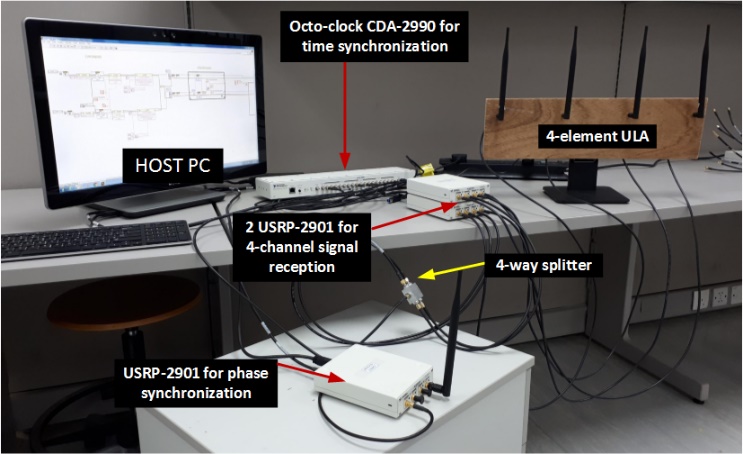
Fig. 11. Prototype testbed built using NI USRP 2901 units for real-time experiments using a 4-element ULA.

Fig. 12 shows real-time estimation results of a single source located at 55o.

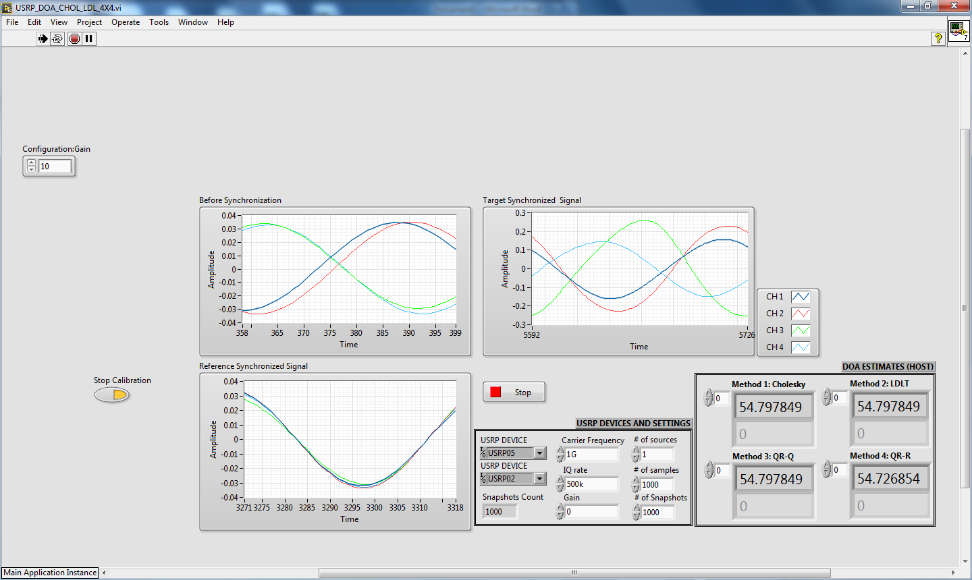
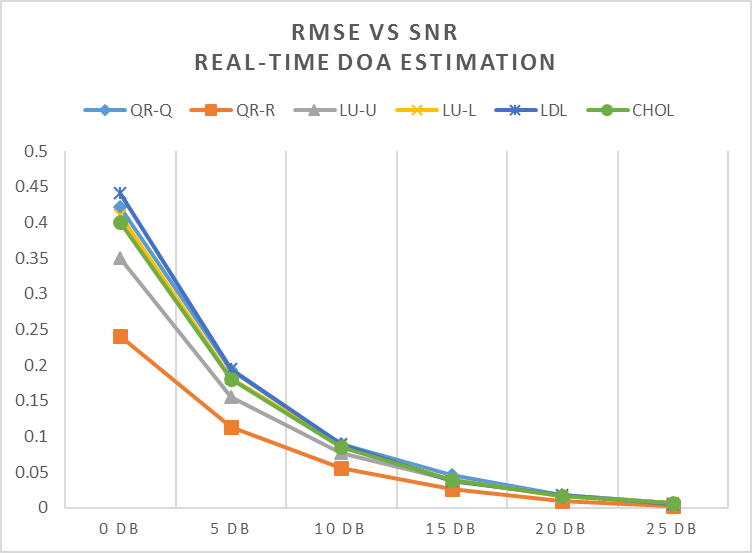


Fig. 12. Screenshot of LabVIEW UI showing real-time estimation of DOA of a single source located at 55o.

Experimental results of real-time DOA angle estimation of a single RF source placed at arbitrary angles are shown in Table VII. Experiments were performed with 20 trials consisting of 10 iterations with 500 snapshots in each trial. SNR was fixed at 10 dB. The values shown in the table are calculated mean values of DOA estimates. It can be observed that DOA estimates using QR-R are closest to the actual source angle. DOA estimates using other methods are reasonably accurate with standard deviation in the range ±0.35 to ±0.60.

TABLE VII. REAL-TIME DOA ESTIMATES

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Actual location:  Single Source | Real-time DOA Estimation | | | | | |
| **QRQ** | **QRR** | **LUL** | **LUU** | **LDL** | **CHOL** |
| 55° | 54.46° | 54.79° | 55.45° | 55.32° | 55.49° | 55.35° |
| 90° | 90.47° | 89.83° | 89.45° | 89.64° | 89.55° | 89.59° |
| 110° | 110.44° | 110.20° | 109.47° | 109.65° | 109.47° | 110.43° |
| 130° | 129.55° | 129.78° | 130.55° | 130.40° | 130.45° | 129.53° |

Fig. 13. Performance comparison: RMSE vs SNR for DOA estimation in real-time.

A performance comparison graph of RMSE vs. SNR for estimating the DOA angle of a single source located at 110o is shown in Fig. 13. It is clear from the graph that QR-R is the most accurate in computing the DOA estimates especially at low SNR values while LU-U is the next best. The rest of other methods have acceptable estimation accuracy. Beyond SNR of 15 dB, all methods demonstrate high estimation accuracy.

1. CONCLUSIONS

In this paper, a performance comparison of matrix decomposition methods used in signal space extraction in the DOA estimation of RF incident sources was presented. The DOA estimation algorithms employing QR, LU, LDL, and Cholesky were implemented in hardware on an FPGA and their performance was compared with respect to resources consumption, computation speed as well as estimation accuracy. These algorithms were validated through both LabVIEW simulations as well as real-time experiments on a prototype testbed constructed using USRPs, the software defined radio platform from NI. An analysis of resources consumption and computation time revealed the superiority of LU-based algorithms. QR-R is a cut above the rest, in terms of estimation accuracy, but its performance comes at the cost of significantly large resource consumption and high computation time. Hence, it can be concluded that LU-based method provides acceptable estimation accuracy while being the most efficient for FPGA hardware implementation.

##### Acknowledgment

The support of Prince Mohammad bin Fahd University (Al Khobar, KSA) and the use of its facilities in carrying out this research work is acknowledged and appreciated by the authors.

##### 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. 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.
3. P. Yang, F. Yang, and Z.-P. Nie, “DOA Estimation with Sub-array Divided Technique and Interpolated ESPRIT Algorithm on a Cylindrical Conformal Array Antenna,” Progress In Electromagnetics Research, vol. 103, pp. 201–216, 2010.
4. C. Qian, L. Huang, H. C. So, “Computationally efficient ESPRIT algorithm for direction-of-arrival estimation based on Nyström method,” Signal Process., vol. 94, no. 1, pp. 74-80, 2014.
5. Zhao Q., Liang W., “A Modified MUSIC Algorithm Based on Eigen Space,” Advances in Computer Science, Intelligent System and Environment. Advances in Intelligent and Soft Computing, vol 104. Springer, Berlin, Heidelberg, 2011.
6. K. V. Rangarao and S. Venkatanarasimhan, "gold-MUSIC: A Variation on MUSIC to Accurately Determine Peaks of the Spectrum," in IEEE Transactions on Antennas and Propagation, vol. 61, no. 4, pp. 2263-2268, April 2013.
7. 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
8. 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.
9. 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
10. 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
11. A. A. Hussain, N. Tayem and A. Soliman, "LDL Decomposition-based FGPA Real-time Implementation of DOA Estimation," 2018 52nd Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, USA, 2018, pp. 1163-1168. doi: 10.1109/ACSSC.2018.8645387
12. A. A. Hussain, N. Tayem, A. Soliman and R. M. Radaydeh, "FPGA-Based Hardware Implementation of Computationally Efficient Multi-Source DOA Estimation Algorithms," in IEEE Access, vol. 7, pp. 88845-88858, 2019.doi: 10.1109/ACCESS.2019.2926335
13. N. P. Waweru, D. B. O. Konditi, P. K. Langat, “Performance Analysis of MUSIC, Root-MUSIC and ESPRIT DOA Estimation Algorithm”, World Academy of Science, Engineering and Technology International Journal of Electrical, Computer, Electronics and Communication Engineering Vol:8 No:1, 2014
14. 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.
15. Golub, G.H., Van Loan, C.F.: Matrix computations (Johns Hopkins University Press, London, 2013, 4th edn.
16. Chang R.C.H., Lin C.H., Lin K.H., ET AL.: ‘Iterative QR decomposition architecture using the modified Gram Schmidt algorithm for MIMO systems’, IEEE Trans. Circuits Syst. I, Regul. Pap., 2010, 57, (5), pp. 1095–1102.
17. M. Bouri and S. Bourennane," High resolution methods based on rank revealing triangular factorization," Transactions on Engineering, Computing and Technology, vol. 2, pp. 35-38, Dec 2004.
18. Saleh O. Al-Jazzar, “Angle of Arrival Estimation Using Cholesky Decomposition,” International journal of antenna and propagation,pp.1-6, 2012
19. 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
20. PXIe-7965 PXI FPGA Module for FlexRIO, http://www.ni.com/en-lb/support/model.pxie-7965.html
21. Xilinx Virtex-5 SXT FPGA data sheet: https://www.xilinx.com/support/documentation/selection-guides/virtex5-product-table.pdf
22. NI LabVIEW software platform, http://www.ni.com/labview/
23. NI LabVIEW FPGA Module, http://www.ni.com/labview/fpga/
24. NI USRP SDR platform, <http://www.ni.com/en-lb/shop/select/usrp-software-defined-radio-device>
25. USRP SDR 2901: <http://www.ni.com/en-lb/support/model.usrp-2901.html>
26. Octo-clock CDA-2990 8-Channel Clock Distribution Module: https://www.ettus.com/product/details/OctoClock-G