e-ISSN 2320 –7876 www.ijfans.org
Vol.11,S Iss 1, 2022
Research Paper , **UGC CARE Listed (Group -I) Journal**

Simulation of Systolic Arrays for QR Decomposition

Rahil Khan †
Research Scholar, Sage University, Indore, MP, India, rahilhkhanpathan@gmail.com

Dr. Mukesh Yadav,
Associate Professor, Sage University, Indore, MP, India, Dr.mukesh71073@gmail.com

ABSTRACT

Adaptive filters have gained popularity over the years due to their ability to adapt themselves to different environmental scenarios without the user taking much action. Here, an adaptive noise cancellation filter technique is put into practice. The Recursive Least Square (RLS) algorithm is used in the filter design because it is computationally straightforward, exhibits robust performance when implemented in hardware with finite precision, and exhibits well-understood convergence performance. Through the use of the MATLAB/Simulink tool, the RLS algorithm can verify the accuracy and responsiveness of the adaptive noise cancellation filter. Utilizing the Xilinx Tool Box, this algorithm is implemented using the Simulink model as a guide. The System Generator ("SysGen") tool in the Xilinx block set is used to build the bit file that can be downloaded onto the FPGA through hardware co-simulation in order to construct the adaptive filter on Xilinx. The study proposes an adaptive noise cancellation filter that uses the noise-cancelling RLS algorithm, and the outcomes are verified by visualising the output using MATLAB.

Keywords : Cancelling noise, adaptive filtering process, Recursive Least Square (RLS) algorithm

1. Introduction

Beamforming Adaptive weight calculation (AWC) has Numerous wireless communication applications, such as adaptive beamforming, equalization, predistortion, and multiple-input multiple-output (MIMO) systems, call for the use where A is the observations matrix which is assumed to be noisy, b is a known training sequence and x is the vector to be computed by using least squares method. This is described more compactly in matrix notation as Ax = b + e. If there is the same number of equations as there are unspecified, i.e., n = m, this system of equations has a distinctive solution, x = A−1b. Applications requiring high sample rates are frequently over-determined, or, i.e., n > m. By reducing the residuals, the least squares method—min n e2 n—helps to overcome the issue. Generally speaking, the least squares method (LMS), Normalized LMS (NLMS) and Recursive Least Squares (RLS), is used to find an approximate solution to these kinds of system of equations. Among them, RLS is most commonly used due to its good numerical properties and fast convergence rate. However, it requires matrix inversion which is not efficient in terms of precision and hardware implementation.

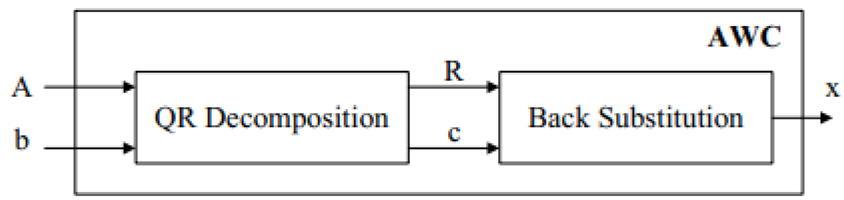

Figure 1. Adaptive Weight Calculation (AWC) using QRD-RLS method

Figure 1 shows the Adaptive Weight Calculation (AWC) using QRD-RLS method which consists of two different parts to calculate the weights, QR decomposition and back-substitution. It is recommended to use QR decomposition to carry out adaptive weight computation based on RLS as it produces more precise outcomes and effective architectures. In order to create an adaptive weight calculation (AWC) core that uses QR decomposition in its solution steps, we use our tool GUSTO. After transforming b into a different column matrix, c, so that it becomes the solution of the QR decomposition, the resulting upper triangular matrix, R, is used to obtain the coefficients of the system using back substitution. Rx = c

2. Algorithm Description

QR decomposition is an elementary operation, which decomposes a matrix into an orthogonal and a triangular matrix. QR decomposition of a real square matrix A is a decomposition of A as A = Q×R, where Q is an orthogonal matrix (QT × Q = I) and R is an upper triangular matrix. And Factor (m × n) matrices (with m ≥ n) of full rank as the product of an (m × n) orthogonal matrix where QT × Q = I and (n× n) upper triangular matrix.

Different techniques can be used to calculate QR decomposition. The techniques for QR decomposition are householder transformation and the givens rotations. But above all method, the Coordinate Rotation Digital Computer "CORDIC" algorithm is effective technique and using for QR decomposition.

Consider the estimation of the N-dimensional parameter vector $\hat{\theta}$ for the following linear model:

$$d(k) = x^T(n)\,\hat{\theta} + v(n) \qquad (1)$$

e-ISSN 2320 –7876 www.ijfans.org
Vol.11,S Iss 1, 2022
Research Paper , **UGC CARE Listed (Group -I) Journal**

where $d(k)$ and x^T (n) are the desired signal and input vector respectively, and

$$e(j) = d(j) - x_N^T(j)\theta(n) \qquad (2)$$

In least squares parameter estimation, the following time-averaged squared magnitude error is:

$$\xi_N(n) = \sum_{j=0}^{n} \lambda^{n-j}|e(j)|^2 \qquad (3)$$

where the constant ƛ is the forgetting factor with a value between 0 and 1. Equation (1) can be written more compactly in matrix form as:

$$e(n) = d(n) - x_N(\mathrm{n})\,\Theta(n) \qquad (4)$$

Where

$$d(n) = [d(0), \quad d(1) \dots\dots, d(n)]$$
$$x_N(n) = [x_1(n),\ x_2(n) \dots\dots, x_N(n)]^T$$
$$X_N(n) = [x_N(0),\ x_N(1) \dots\dots, x_N(n)]^T$$

X_N(n) and X_N(n) are the received signal vector and the data matrix, respectively. Then, the least squares objective function £(k) in becomes
£(k) = $e^H w^2(k)e(k)$ = $|w(k)e(k)|^2$
Where w(n) is a diagonal weighting matrix given by

$$W(n) = diag.(\sqrt{ƛ^n}, \sqrt{ƛ^{n-1}}, \dots\dots, \sqrt{ƛ},\ 1) \qquad (5)$$

The optimum value of Θ()can be obtained by solving the normal equation:

$$R_N\,(n)\Theta(n) = P_N\,(n)$$
$$R_N\,(n) = \sum_{i=0}^{n} ƛ^{n-i}\, x_N(i)x_N^H(i)$$
$$P_N\,(n) = \sum_{i=0}^{n} ƛ^{n-i}\, d(i)x_N^H(i)$$

are, respectively, the weighted autocorrelation matrix of $x_N(k)$, and the weighted cross correlation vector of $x_N(k)$ and $d(k)$. Due to the lower numerical accuracy in solving the normal equation, a better method, calledthe QR-LS method, is employed.
The following QRD of w(k)$X_M(k)$ is performed

(k)w(k)$X_N(k)$ = $[^{RN(k)}]$

where $Q(k)$is some $(k + 1)(k + 1)$ unitary matrix and R_N (k) is (N x N) anupper triangular matrix. Using can be rewritten as

$$Q(n)W(n)e(n) = \begin{bmatrix} d_N(n) \\ C_{n+1-N} \end{bmatrix} - \begin{bmatrix} R_N(n) \\ 0 \end{bmatrix} \Theta(n) \qquad (6)$$

where $Q(k)W(k)d(k)$ =[$dN(k)Cn$+1−N] T. Since (k) is a unitary matrix, the square of the Euclidean norm is equal to £(k). The two-norm achieves its minimum value when $\Theta(k)$ is chosen as $RN(k)\Theta(k)$ = $dN(k)$, and $mik\Theta(k)£N(k)$ = £∗ (k) = ‖CN+1−N‖2. Since (k) is an upper triangular matrix $\Theta(k)$, can be obtained by back-substitution. There are several methods to perform the QRD of the weighted data matrix W(k)X$N(k)$.

The inverting a matrix [A] using Gaussian elimination has a complexity of (k^3). For real-time applications and high values of n, the complexity of matrix inversion in hardware becomes unmanageable. Our objective is to physically invert a matrix of dimensions (12×12) in hardware. The results for inverting an 8 x 8 matrix can be shown in this publication. For bigger matrix sizes, the same principle can be applied with a minor hardware adjustment. Systolic arrays and QR decomposition were used in the hardware design. Figure (2).

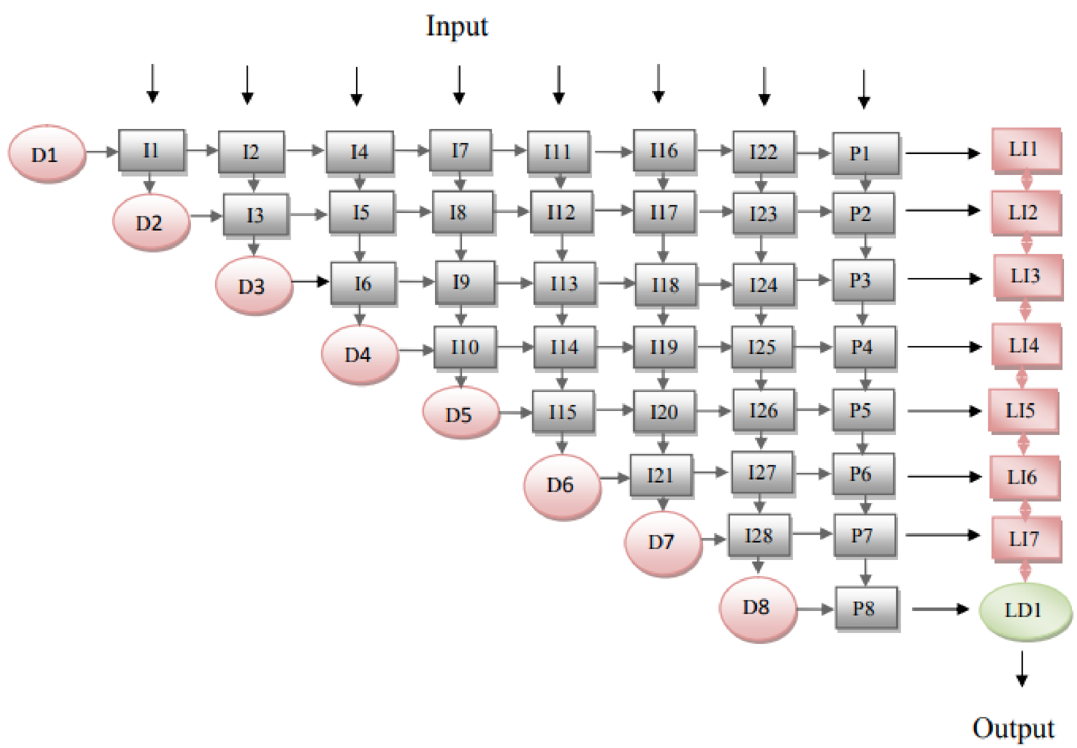

Figure 2. Systolic Array for QR Decomposition

3. Matlab simulation result of Adaptive Beamforming:

In this section, An example of the MVDR beamformer from [6] can be shown in Figure 3 where the correlation/covariance matrix is built from the snapshot of K samples of data from each channel, and passed to an MVDR processor which calculates the adaptive

 ,

beamforming weights which are applied across each channel, then coherently summed to form the output beam y(n). Note, in, the channel count is denoted by M instead of N.

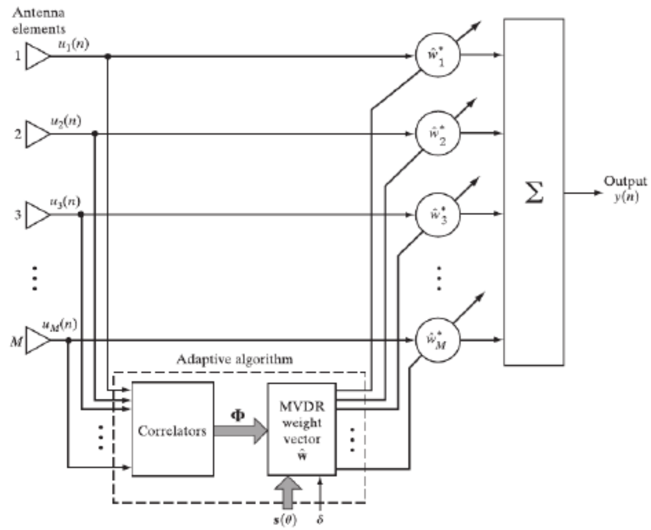

Figure 3. Block Diagram of a MVDR Beamformer for a ULA

Using the same example from the deterministic beamforming section, we can apply the MVDR-calculated beamforming weights and compare the spectrum to the quiescent beamforming spectrum from Figure 3. To repeat the scenario, a desired signal at 300 MHz is impinging the array at θd = 5◦and an interference source at 270 MHz is impinging the array at θ in f = 30◦. The MVDR narrowband beamformer output is thus given by equation below

$$y(k) = \hat{\mathbf{w}}^H \mathbf{x}(k) \tag{7}$$

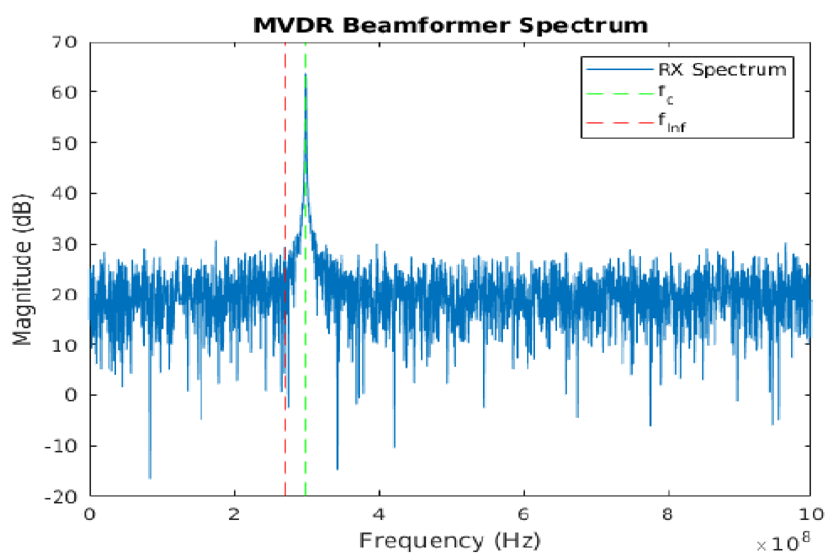

Figure 4. Output Spectrum from the MVDR Beamformer for a ULA

e-ISSN 2320 –7876 www.ijfans.org
Vol.11,S Iss 1, 2022
Research Paper , **UGC CARE Listed (Group -I) Journal**

As seen in Figure 4, the desired SOI has a visibly high SINR, due to the applied beamforming gain, and the interference signal is no longer present in the output spectrum at all. Further demonstration that the interference source has in-fact been nulled can be seen by examining the MVDR output weights in sine space. This can be seen in Figure 3 where the interference angle sees a deep null, and as such, the interference signal is attenuated.

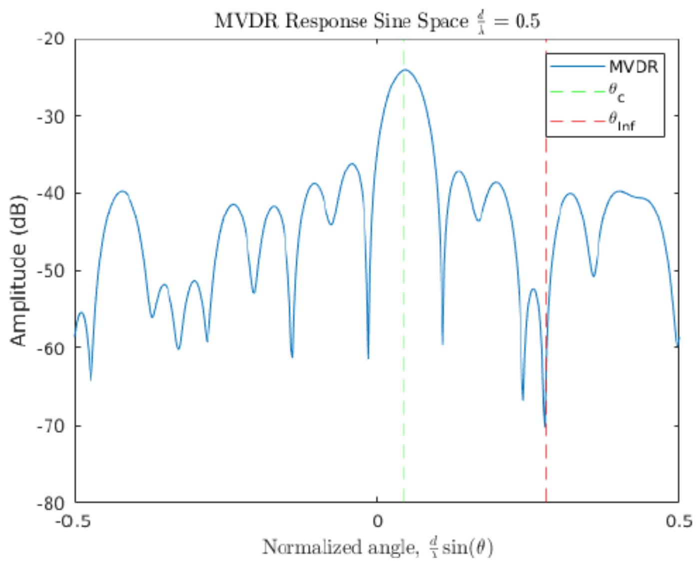

Figure 5. Radiation Plot of the MVDR Beamforming Weights

Based on the spatial response of a given number of antenna elements, the degrees of freedom (DOF) of an N element array is fundamentally driven by the number of independent nulls that can be produced by the MVDR algorithm, as defined in Equation 8.

$$DOF = N - 2 \qquad (8)$$

In the context of interference mitigation, this means that up to N − 2 interference sources can be cancelled out using MVDR [6]. To show this in practice an 8-element ULA can be shown to have 6 interference sources, at varying narrowband frequencies and varying incident angles, which completely muddies the output spectrum in the quiescent beamforming

4. Conclusion :

In this research work, we've covered the background knowledge of RF array processing in the context of current, and next generation, MIMO systems. We then explored the current state-of-the-art in Adaptive Beamforming processes for embedded FPGA devices. After creating a baseline implementation forperformance and resource comparisons, we explored a novel Deep Learningmodel to solve Adaptive Beamforming weights in a more efficient process then the current closed-form, statistical solution

5. References :

[1] B. D. V. Veen and K. M. Buckley, "Beamforming: A versatile approachto spatial filtering," IEEE ASSP Magazine, 1998.

[2] "Phased array antenna patterns- part 1: Linear array beam characteristicsand array factor," Analog Devices, Tech. Rep., 2020. [Online]. Available: https://www.analog.com/en/analog-dialogue/articles/ phased-array-antenna-patterns-part1.html#.

[3] S. Rangan, T. S. Rappaport, and E. Erkip, "Millimeter-wave cellular wireless networks: Potentials and challenges," Proceedings of the IEEE, vol. 102, no. 3, pp. 366–385, 2014. DOI: 10.1109/JPROC.2014.2299397.

[4] J. Guerci, Space-Time Adaptive Processing for Radar, 2nd ed. Artech House,2015.

[5] R. Mailloux, Phased Array Antenna Handbook, 2nd ed. Artech House,2005.

[6] S. Haykin, Adaptive Filter Theory, 5th ed. pearson, 2014, ISBN: 978-0-132-67145-3.

[7] E. P. Tsakalaki, L. Ã. M. R. de TemiÃs´o, T. Haapala, J. L. RomÃa¸n, and M. A. Arauzo, "Deterministic beamforming for enhanced vertical sectorization and array pattern compensation," in 2012 6th European Conference on Antennas and Propagation (EUCAP), 2012, pp. 2789–2793. DOI: 10.1109/EuCAP.2012.6206403.

[8] M. A. Richards, J. A. Scheer, and W. A. Holm, Principles of Modern Radar: Basic Principles. IET, 2010, ISBN: 9781891121524.

[9] "Phased array antenna patterns- part 2: Grating lobes and beam squint," Analog Devices, Tech. Rep., 2020. [Online]. Available: https://www. analog.com/en/analog-dialogue/articles/phased-array-antennapatterns- part2.html. 42

[10] D. Li, Q. Yin, P. Mu, and W. Guo, "Robust mvdr beamforming using the doa matrix decomposition," IEEE-ISAS, 2011.