Design Of Smart Antenna Array With Interference Supression Capability

Er. Aashish, Assistant Professor, SVIET Ramnagar, Banur, India
Er. Rishav Dewan, Assistant Professor, BGIET Sangrur, India

Abstract
Matrix decomposition refers to the transformation of a matrix into a canonical form. Matrix decomposition has very important applications in scientific computing because of its scientific and engineering significance. The purposes of matrix decomposition are analytic simplicity and computational convenience. The choice of right decomposition technique depends on the problem we want to solve and the matrix to be decomposed. In this paper, a Quadratic rotation decomposition (QRD) technique which decomposes a matrix into an orthogonal and a triangular matrix using Gram-Schmidt orthonormalization method has been used with recursive least squares (RLS) algorithm for implementing a smart antenna. Simulation results have been presented.

Keywords: Matrix decomposition, Quadratic rotation decomposition, Recursive least squares, Smart antenna.

Introduction
Spatially propagating signals encounter the presence of interfering signals and noise signals. If the desired signal and the interferers occupy the same temporal frequency band, then temporal filtering cannot be used to separate the signal from the interferers. However the desired and the interfering signals generally originate from different spatial locations. At the same time, some detrimental effects in randomly varying mobile communication environment like multipath fading, co-channel interference and Doppler effects need to be addressed. Smart antenna is a recent design technology that known to offer the solution for the above mentioned problems.

Research on applications of adaptive antenna arrays have been an interesting subject over past 40 years [1] contributing to the invention of smart antenna design methods. By taking advantage of the fact that users collocated in frequency domain are typically separated in spatial domain, the smart antenna is used to direct the antenna beams towards the desired user while canceling signal from other users [2]. The smart antenna electronically steer a phased array by weighting the amplitude and phase of signal at each array element in response to changes in the propagation environment. Capacity improvement is achieved by effective co-channel interference cancellation and multipath fading mitigation. The beam pattern produced by a phased array antenna can be steered electronically to place the region of greatest sensitivity towards a source of interest, and nulls in the directions of interferers. Modern communication systems require adaptive filters which converge much more quickly than Least Mean Squares (LMS) algorithm.

The QR algorithm is a Recursive Least Squares (RLS) technique which meets this need and is particularly suited to Field Programmable Gated Array (FPGA) implementation. The LMS has been widely used for more than 40 years. It is well known that the LMS is an approximation to the least squares (LS) solution for adaptive filtering, however its convergence and simple implementation have made it the algorithm of choice for applications such as echo control, and wire line channel equalisation. One key point regarding wire line channel equalisation is that there enough time to train the adaptive filter, and once adapted the channel is stationary and does not change. However for wireless applications includes equalisation, and smart antenna beamformers, MIMO systems etc. The time available for training the system is very small, and further the channel will change and a complete re-training of the system is required. Now, the faster the channel changes, the shorter the time available for training, and the training is required more frequently. Hence faster adaptive algorithms are re-

Figure 1: Array of Antennas
quired. This is the simple motivation and drive for the move real time LMS to real time LS algorithms.

In its direct form, the RLS algorithm would require floating point precision, or very long fixed point word lengths, due to its numerical ill-conditioning. In addition to Multiply/Add standard RLS implementation also requires divide operations. Hence the consequences of overflow and underflow can cause serious problems such as Divide-by-zero errors, etc. Hence for FPGA fixed point implementation, RLS must be carefully implemented. Therefore long fixed point word length is likely to provide the dynamic range demanded by the RLS algorithm. This motivates the QR-RLS algorithm method which is the most numerically robust method of RLS implementation and aims to keep the dynamic range of values low.

II. QRD-RLS Algorithm

Recursive least squares algorithm attempts to solve for the coefficient vector \( \mathbf{c} \) from \( \mathbf{X} \) and \( \mathbf{y} \). To realize this, the QR-decomposition algorithm [3] is first used to transform the matrix \( \mathbf{X} \) into an upper triangular matrix \( \mathbf{R} \) (\( N \times N \) matrix) and the vector \( \mathbf{y} \) into another vector \( \mathbf{u} \) such that \( \mathbf{R} \mathbf{c} = \mathbf{u} \). The coefficients vector \( \mathbf{c} \) is then computed using a procedure called back substitution, which involves solving the equations shown below

\[
c_N = \frac{\mathbf{u}_N}{R_{NN}} \\
c_i = \frac{1}{R_{ii}} \left( \mathbf{u}_i - \sum_{j=i+1}^{N} R_{ij} c_j \right) \quad \text{for } i=N-1, \ldots, 1
\]

The QRD-RLS algorithm flow is shown as below:

![Fig 2 QR Decomposition Based Least Square](image)

The QR-decomposition [4] of the input matrix \( \mathbf{X} \) can be performed, as illustrated in Fig 3, using the well-known systolic array architecture [5]. The rows of matrix \( \mathbf{X} \) are fed as inputs to the array from the top along with the corresponding element of the vector \( \mathbf{y} \). The \( \mathbf{R} \) and \( \mathbf{u} \) values held in each of the cells once all the inputs have been passed through the matrix are the outputs from QR-decomposition. These values are subsequently used to derive the coefficients using the back-substitution technique. Each of the cells in the array can be implemented as a coordinate-rotation digital computer (CORDIC) block.

![Fig 3 Systolic Array Architecture for QR Decomposition](image)

CORDIC describes a method to perform a number of functions, including trigonometric, hyperbolic and logarithmic functions. The algorithm is iterative, and uses only additions, subtractions and shift operations. This makes it very attractive for hardware implementations. The number of iterations depends on the precision required, which correlates directly with the amount of bits needed. For real inputs, only one CORDIC block is required per cell. Many applications involve complex inputs and outputs to the algorithm, for which three CORDIC blocks are required per cell. In such cases, a single CORDIC block can be efficiently timeshared to perform the complex operations [6,7]. Direct mapping of the CORDIC blocks onto the systolic array is shown in Fig 3. The resources required to implement the array can be reduced by trading throughput for resource consumption via mixed and discrete mapping schemes.

a) Mixed mapping: In mixed mapping schemes, the bottom rows in the systolic array are moved to the end of the top rows, to ensure the same number of cells in each row. A single CORDIC block can then be used to perform the operations of all the cells in a row, with the total number of CORDIC blocks required being equal to the total number of rows. Because each CORDIC block has to operate in both victories and rotating modes, the scheme is called mixed mapping [8].
b) Discrete mapping: In this scheme, at least two CORDIC blocks are required. One block is used purely for victories operations, while the other is used for rotate operations [9]. This single functionality of the processors allows any gains from hardware optimization to be realized.

III. Proposed Smart Antenna
The adaptive smart antenna has been implemented for four antennas. Code has been written in MATLAB. For the proposed design input signal frequency has been taken as 6 GHz and has been sampled at a rate of 100 GHz. The angle of incidence for desired input signal has been kept as 0 degree and amplitude has been taken as 1V. Fig 6.1 and 6.2 show the input signal and its spectrum respectively. Whereas, for the interfering signal, frequency has been taken as 20 GHz and has been sampled at a rate of 100 GHz. The angle of incidence for the interfering signal has been kept as pi/4 degree and amplitude has been taken as 1V. Fig 4 and 5 show the interfering signal and its spectrum respectively.

Fig 4 Input Signal

Fig 5 Spectrum of Input Signal

Fig 6 interference signal

Fig 7 Spectrum of Interference Signal

Fig 8 and 9 shows the desired signal corrupted with the interfering signal.

Fig 8 Input Signal Corrupted with Interfering signal

Fig 9 Spectrum of Interference Signal
Comparing Fig 5 and 9, it has been concluded that the interfering signal has resulted in an undesired peak in the spectrum, which should be removed by using adaptive algorithm. When adaptation has been applied with FFT, the output becomes as smooth as was the input. Fig 10 show the ability of the proposed adaptive beamformer in suppressing the undesired peak.

**Fig 10 Smart Antenna Output**

**IV. Conclusion**

Research on applications of smart antenna arrays have been an interesting subject over past 40 years contributing to the invention of adaptive beamforming method. By taking advantage of the fact that users collocated in frequency domain are typically separated in spatial domain, the beamformer is used to direct the antenna beams toward the desired user while canceling signal from other users. Adaptive beamformer uses adaptive algorithm for adjustment of its weights. This paper presents the design of a smart antenna array using DRD-RLS algorithm. Simulation results show that the proposed antenna array was able to suppress the undesired peak in an effective manner.

---

**References**


