Matlab Simulation of Cordic based Adaptive Filtering for Noise Reduction using Sensors Array

Anamika Gupta
ME. Student
Department of ECE
NITTTR,Chandigarh

Rajesh Mehra
Associate Professor
Department of ECE
NITTTR, Chandigarh

ABSTRACT

The process of suppressing noise and interference in any type of signal for transmission has created challenges in the area of digital signal processing. Apart from mobile communication devices, there are a huge number of applications, in which it is difficult to have a good acoustic interface for accurate voice control or smooth audio communication. So for this it is very essential to enhance the signal by removing its noise.For this a cordic based adaptive algorithm has been designed and simulated for digital signal processing. When the input signals statistics to the filter are time varying or unknown and the design requirements for fixed filters cannot easily be specified then adaptive filter is useful**.** Examples of such applications are system identification, channel, equalization /identification, noise reduction in digital signal processing. These algorithms are also used in case of beamforming. Array of sensors (microphones) play an important role in noise reduction and speech enhancement. In this paper a cordic based qrd_rls adaptive (CQR_RLS) algorithm is developed and simulated using MATLAB. Instead of taking signal from one sensor here array of sensor is used for signal as well as interference.Matlab simulation results are calculated and compared for different iteration.

General Terms

Adaptive algorithm, Matlab

Keywords

Qrd_rls, cqr_rls

1. INTRODUCTION

In the last many years the day by day advances in digital circuit design have been the main key technological development that increased a growing interest in the field of digital signal processing. The main attraction of digital signal processing systems is their accuracy, less cost, reliability, small physical sizes, and flexibility. The most efficient example of DSP is filter. The main aim of filtering is filter out the noise and interference contained in the signal. A filter that processes discrete-time signals represented in digital format is called digital filter. In some cases when using digital filters, signals or systems may undergo some changes with time or some noise and interference occurs but the exact nature of change is not predictable then in such cases it is highly required to design a filter that can learn from the process itself, way that can be adapted to handle the situation. To resolve many of these problems, it is proposed to use adaptive filters [1].Today adaptive systems have found their way into many applications where learning capacity of the system is a factor important. An adaptive filter is a filter that self-adjusts its transfer function according to an optimization algorithm driven by an error signal [2].

Today adaptive systems have found their way into many applications where learning capacity of the system is a factor important. There are several algorithms to achieve the calculation of coefficients in a given system, which vary in complexity. Among the most simple is the Least Mean Square algorithm (LMS). This algorithm is widely used because of its ease of implementation and low utilization of computer resources. When the medium is highly dynamic, requires algorithms that adapt quickly to changes, for these cases the LMS algorithm do not provide a good Performance [3].

If LMS algorithms represent the simplest and most easily applied adaptive algorithms, the recursive least squares (RLS) algorithms represents increased complexity, computational cost, and fidelity. In performance, RLS approaches the Kalman filter in adaptive filtering applications, at somewhat reduced required throughput in the signal processor. Compared to the LMS algorithm, the RLS approach offers faster convergence and smaller error with respect to the unknown system, at the expense of requiring more computations. In contrast to the least mean squares algorithm, from which it can be derived, the RLS adaptive algorithm minimizes the total square error between the desired signal and the output from the unknown system.

Suppose in a room an array of sensor is places In the case of speech signal, a person who is near to first sensor of the sensor array take voice signal as a useful signal while all other sensor takes this signal as interference signal, because that signal is not in the frequency range of the sensors. If person moves from one place to another, then the sensor which is near to the person takes person’s voice as a useful signal. So for this reason an array of sensor is placed in place of one sensor.The paper is organized as follows: Section 2 explains the basic concepts adaptive algorithms in general. Section 3 shows the architecture and design platform. This paper shows the implementation of RLS algorithm which is based on cordic processing. The proposed algorithm is developed in MATLAB and after that the simulated results for different iterations are compared.

2. ADAPTIVE ALGORITHM

An adaptive system can be modeled as shown in Figure 1. As can be seen there is a plant with defined characteristics, the output is input to the adaptive mechanism after being subtracted of a desired signal, whereby adaptive mechanism can calculate the new coefficients needed to adapt the response of the plant to the desired. The operation model equations showed an adaptive system [4].

$$\mathbf{y(n)=x(n)*w(n)} \quad (1)$$
$$\mathbf{e(n)=d(n) - y(n)} \quad (2)$$
$$\mathbf{e(n)=d(n) - [x(n)*w(n)]} \quad (3)$$

Where x(n) is the input signal, y (n) is the filter output, d (n) is the desired output signal and e(n) is the error between d(n) and y(n). In this case, the signal input x(n) moves into the plant that contains the coefficients w (n) (FIR filter) and returns a signal y(n) whose result is shown

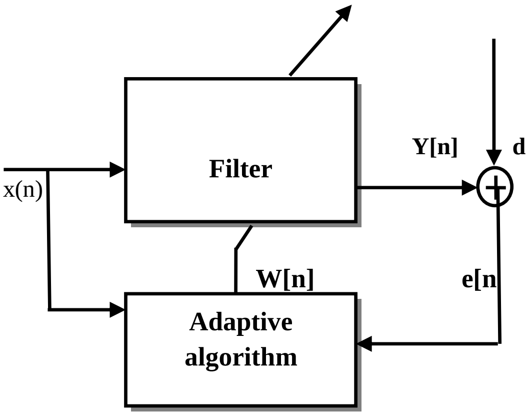

Fig 1: Diagram of an adaptive system

in equation 1. Then an error signal e(n) is computed by the subtraction of result y(n) from a signal d(n) ,whose result is shown in equation (2), which is the parameter that tells the adaptive mechanism how "far" is the mechanism to have a response similar to the desired signal d(n). With the help of this error signal and the input signal new coefficients w (n) are calculated for the Filter using an Adaptive algorithm.

In the noisy environment speech signal is affected by the presence of noise signal (acoustic noise).To solve this problem one possible solution of obtaining a better recording of desired signal is simple sensor array system with adaptive filter as shown in figure 2.

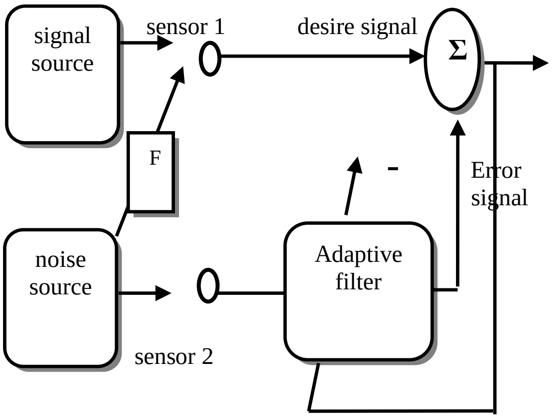

Fig 2: adaptive noise cancelation system

In the above figure 2 the path at which signal coming from the noise source to the sensor 1,which is primary sensor is as unknown FIR channel F.If an adaptive filtering is applied to the noise source at the sensor 2,then it is possible to employ an adaptive algorithm to train the adaptive filter.

There are four major types of adaptive filtering configurations, adaptive system identification, adaptive noise cancellation, adaptive linear prediction, and adaptive inverse system. All of the above systems are similar in the implementation of the algorithm, but different in system configuration [5]. Figure 3 shows array of sensor based adaptive algorithm system in which 3 sensors are used for interference signal in the place of one.

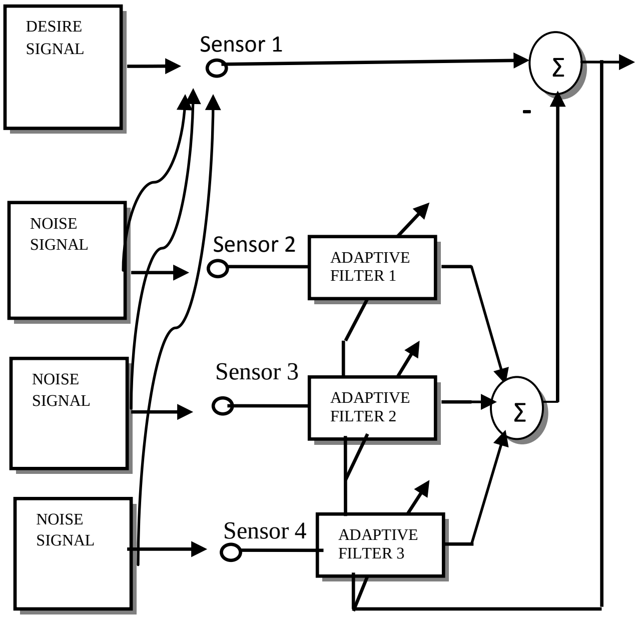

Fig 3:sensor array based adaptive noise cancelation system

2.1 LMS Algorithm

The traditional LMS filtering algorithm is an approximation to using gradient descent to find the optimal filter coefficients by finding the minimum mean square error (MMSE) between the filter output and some desired output. It is an iterative procedure where the coefficients can be updated according to the gradient of the MSE .Least Mean Square (LMS) is the most common and popular algorithm The LMS algorithm is very popular and has been widely used due to its extreme simplicity [5].

2.2 RLS Algorithm

On the other hand RLS (Recursive Least Square), is generally preferred for its fast convergence. The direct calculation of the new vector of coefficients involves matrix inversion, which is usually unwanted in implementations of hardware due to the high consumption of resources. The based matrix decomposition schemes such as least squares, SVD (Singular Value Decomposition) and explicitly avoiding QR matrix inversion are more robust and more affordable implementation in hardware [6].Table 1 shows the comparison between LMS and RLS.

Table 1 : Comparison Table of LMS and RLS

Algorithm	Computational complexity	Convergence speed	**Mult.**	**Add.**	**Sub.**
LMS	O(N)	LOW	**N(2M+1)**	**N(2M)**	**2N**
RLS	O(N^2).	FAST	**N(6M)**	**N(3M)**	**N(2M)**

2.3 QRD_RLS Algorithm

QR-RLS algorithm is used to solve linear least square problems. The decomposition is the basis for the QR algorithm. As all methods are iterative, their development and constant improvement aim for reduction of computational complexity, increased speed of convergence, and robustness against round-off errors. QR matrix decomposition (QRD) is the decomposition of a matrix (*A*) into an orthogonal matrix (*Q*) and an upper triangular matrix (*R*). QRD is useful for solving least squares' problems and simultaneous equations.

3. QR DECOMPOSITION

The QR algorithm, which is based on the QR decomposition of **A**, is still considered one of the most important methods developed so far. the QR algorithm decomposes **A** into a unitary matrix **Q** and a (different!) upper triangular matrix **R**, instead of the lower and upper triangular matrices from elimination. The QR algorithm uses successive unitary transformations, which render the method superior with respect to numerical stability and computational requirements

$$A = QR \qquad \textbf{(4)}$$

where

$$Q = \begin{bmatrix} q1 & q2\cdots & \cdots qN \end{bmatrix} \qquad \textbf{(5)}$$

And

$$R = \begin{bmatrix} r11 & r12\cdots & r1N \\ 0 & r22\cdots & r2N \\ 0 & & rNN \end{bmatrix} \qquad \textbf{(6)}$$

There are three methods for factorization of matrix.The first classical and the modified versions of the Gram–Schmidt orthogonalization method based on projections. Next is Householder orthogonalization method based on reflections and the last one is Givens orthogonalization method based on rotations. Among all three Gives rotation is the most suitable and economical [7]. The QR decomposition method starts from the data matrix using unitary transformation. An error vector is defined as

$$e(n) = d(n) - A(n)w(n) \qquad \textbf{(7)}$$

Now the cost function can be define as

$$\mathrm{E}(n) = \sum\nolimits_{t=1}^{n} |e(i)|^2 \qquad \textbf{(8)}$$

For a given matrix Q(n), the cost function may be expressed as[8]:

$$E(n) = \| Q(n)\Lambda^{1/2}(n)e(n) \|^2 \qquad \textbf{(9)}$$

$$E(n) = \| Q(n)\Lambda^{1/2}(n)d(n) - Q(n)\Lambda^{1/2}(n)A(n)w(n) \|^2 \qquad \textbf{(10)}$$

Where λ is a forgetting factor slightly less than 1, and $\Lambda^{1/2} = diag(\lambda(n-1), \lambda(n-2)....\lambda(0))$.The minimization problem defined by the cost function, the unit matrix Q (n) is chosen to triangular matrix data exponentially weighted such that

$$Q(n)\Lambda^{1/2}(n)d(n) = \begin{bmatrix} R(n) \\ 0 \end{bmatrix} \qquad \textbf{(11)}$$

Where R (n) is an upper triangular matrix dimension k x k and 0 is a zero matrix of dimension (n-K) x K. The desired signal vector, after being converted, is defined by:

$$Q(n)\Lambda^{1/2}(n)d(n) = \begin{bmatrix} p(n) \\ v(n) \end{bmatrix} \qquad \textbf{(12)}$$

Where p (n) is a vector of elements K X 1, v (n) is a vector (n-K) x 1 element, then we can rewrite the cost function as follows:

$$E(n) = \| \begin{bmatrix} p(n) \\ v(n) \end{bmatrix} - \begin{bmatrix} R(n) \\ 0 \end{bmatrix} w(n) \|^2 \qquad \textbf{(13)}$$

$$E(n) = \| \begin{bmatrix} p(n) & R(0)w(n) \\ v(n0 & \end{bmatrix} \|^2 \qquad \textbf{(14)}$$

Obviously, the least squares estimation for the weight vector must satisfy that:

$$w'(n) = R^{-1}(n)p(n) \qquad \textbf{(15)}$$

The unitary matrix Q (n), the upper triangular matrix R (n), and the vector p (n) can be calculated recursively using

$$\begin{bmatrix} R(n) & p(n) \\ 0(n-K-1)xK & 0(n-K-1) \\ 01xK & \alpha(N) \end{bmatrix}$$

$$= Q'(n)\begin{bmatrix} \lambda^{1/2}R(n-1) & \lambda^{1/2}p(n-1) \\ 0(n-K-1)xK & 0(N-K-1)x1 \\ u^T(n) & d(n) \end{bmatrix}$$

$$Q(n) = Q'(n)\begin{bmatrix} Q(n-1) & 0 \\ 0 & 1 \end{bmatrix} \qquad \textbf{(16)}$$

Therefore, the optimal vector of coefficients can be obtained. But in some applications such as noise reduction and linear prediction e (n) is the signal output. Developing the previous equation can obtain e (n) directly without removing the weight vector explicitly. This result can be summarized following equation:

$$\begin{bmatrix} R(n) & p(n) & R^{-1}(n)u(n) \\ 0_{1xk} & \alpha(n) & (\gamma^{1/2}(n))^{*} \end{bmatrix} \quad \textbf{(17)}$$

$$Q''(n)\begin{bmatrix} \lambda^{1/2}R(n-1) & \lambda^{1/2}p(n-1) & 0_{kx1} \\ u^{T}(n) & d(n) & 1 \end{bmatrix} \quad \textbf{(18)}$$

4. CQRD_RLS ALGORITHM

Efficient generation of trigonometric as well as exponential functions without much increase in hardware complexity has always been a challenge, owing mainly to their importance and widespread use in Digital Signal Processing applications besides other areas.

One such algorithm which is very much effective for the calculation of trigonometric, hyperbolic, exponential, linear and logarithmic functions is the CORDIC algorithm .CORDIC (Coordinate Rotation Digital Computer) is an iterative algorithm for the calculation of the rotation of a two-dimensional vector, in linear, circular and hyperbolic coordinate systems, using only add and shift operations. A CORDIC describes a method to perform a number of functions, including trigonometric, hyperbolic and, multiplication with the help of addition and shifting only. The algorithm is very much hardware efficient because it omits the dependence on multipliers and is rather a combination of shift-add operations [9]. The CORDIC is a shift-and-add technique for computing a large class of mathematical functions in hardware. The algorithm can be derived from the rotation transform.

$$x' = x\cos\phi - y\sin\phi \qquad \textbf{(19)}$$

$$y' = y.\cos\phi - x.\sin\phi \qquad \textbf{(20)}$$

On rearrangement of the terms, this can be given as

$$x' = \cos\phi[x - y.\tan\phi] \qquad \textbf{(21)}$$

$$y' = \cos\phi[y + x.\tan\phi] \qquad \textbf{(22)}$$

The implementation of these equations is still complex due to the presence of the trigonometric functions. If the rotation angles are restricted to values such that tan ϕ = ± 2^{-i} , the multiplication by the tangent can be greatly simplified as it can be implemented using simple shift and addition operations. Normally the calculation of an FFT consists complex multiplications. These requirements can be minimized by replacing the complex multipliers by applying CORDIC algorithm. For implementing QR decomposing RLS algorithm there are three methods for factorization of matrix. Among these the last one is Givens orthogonalization method based on rotations is the most suitable and economical. But the hardware consumption is somewhat high. So in the proposed algorithm the Givens rotation is implemented by CORDIC algorithm and shown in figure 4.

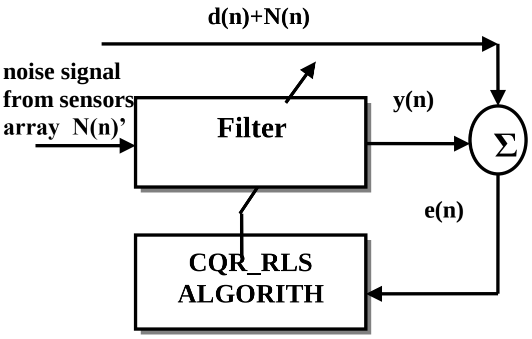

Fig 4:sensor array based cqr_rls adaptive system

5. SIMULATION RESULT

In this paper we have designed and simulated proposed algorithm using MATLAB. Then the results are compared for different adaptive iteration. Figure 5 shows the input signal which is useful signal and the interference signal used in cqr_rls adaptive system.

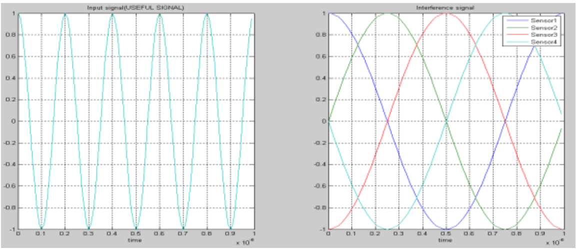

Fig 5:Input signal and interference signal

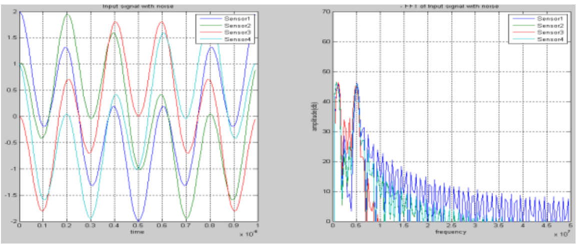

Fig 6:Input signal contaminated with inteference signal

Figure 6 show the input signal contaminated with interference signal and its frequency domain. Now we will discuss the desire signal and output signal for different iterations.

Figure 7, 8, 9 and 10 shows the result of the proposed algorithm for 60, 80,150 and 250 iterations, which shows that after 60 to 80 iterations we are able to get the output signal very close to the desire one but still have some noise signal. After 150 iterations output is very similar to the desire one almost negligible noise as shown in fig. And when this algorithm is computed for 250 iteration then the output is almost a replica of the desire signal as shown in figure 10.So we can see as the number of iteration increases the output moves toward the desire signal.

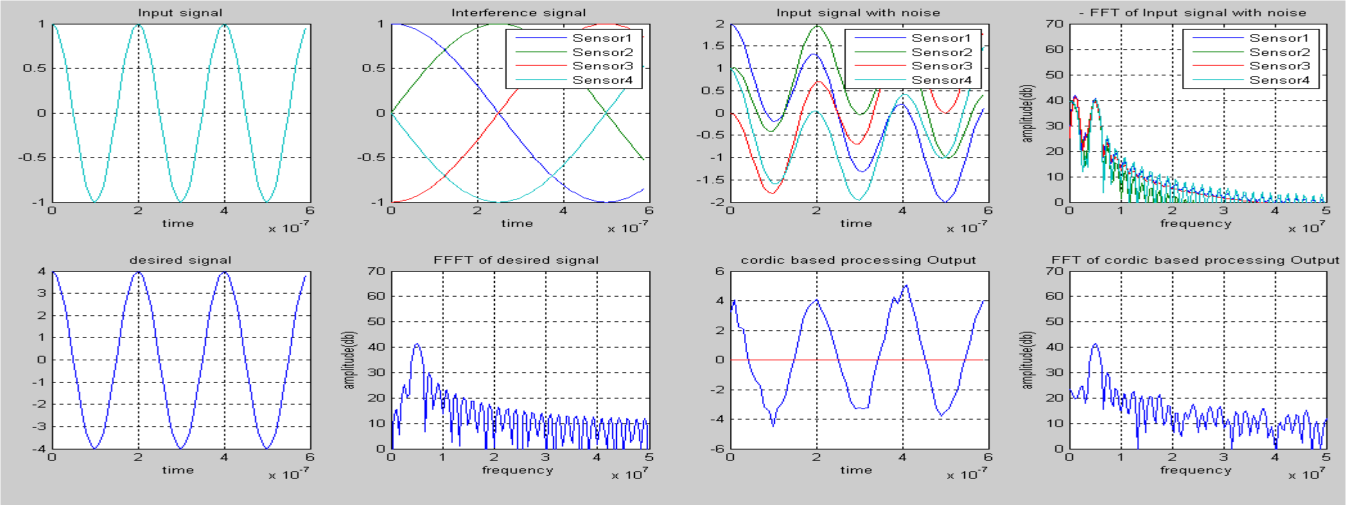

Fig 7 Result of CQR_RLS algorithm for 60 iteration

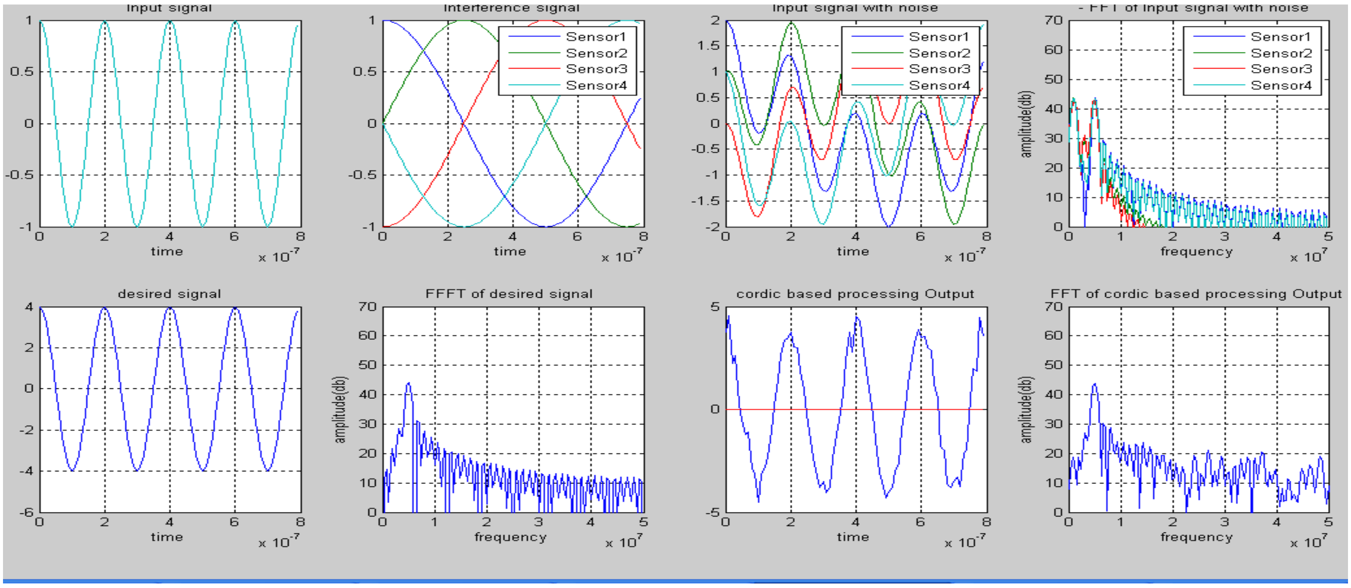

Figure 8 Result of CQR_RLS algorithm for 80 iteration

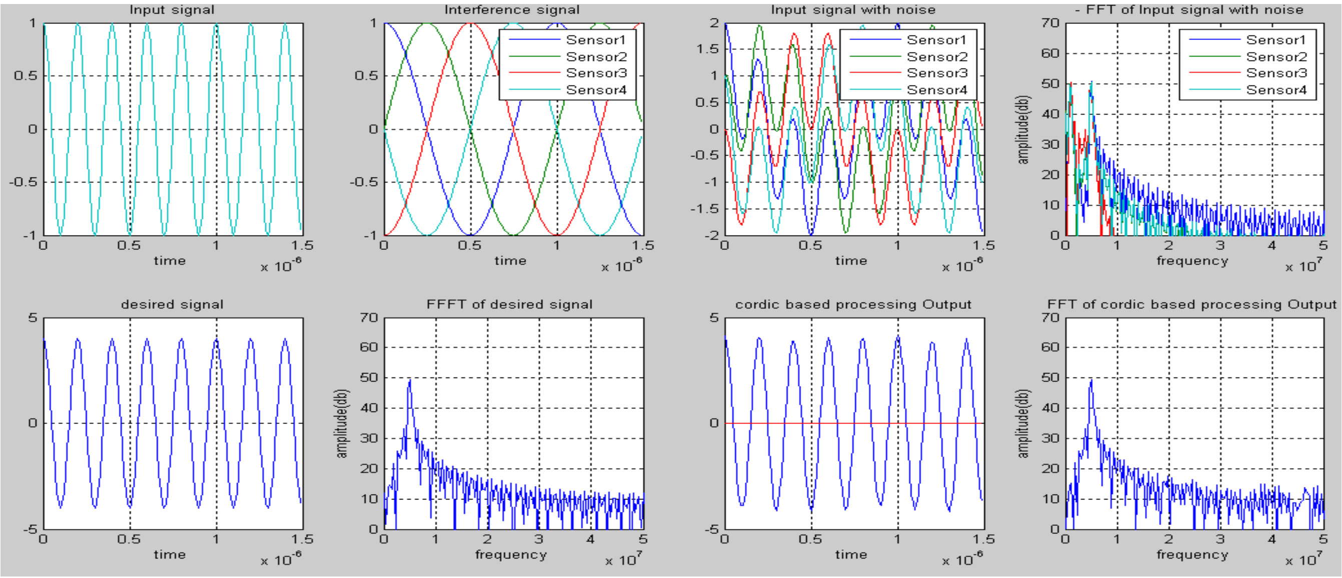

Figure 9 Result of CQR_RLS algorithm for 150 iteration

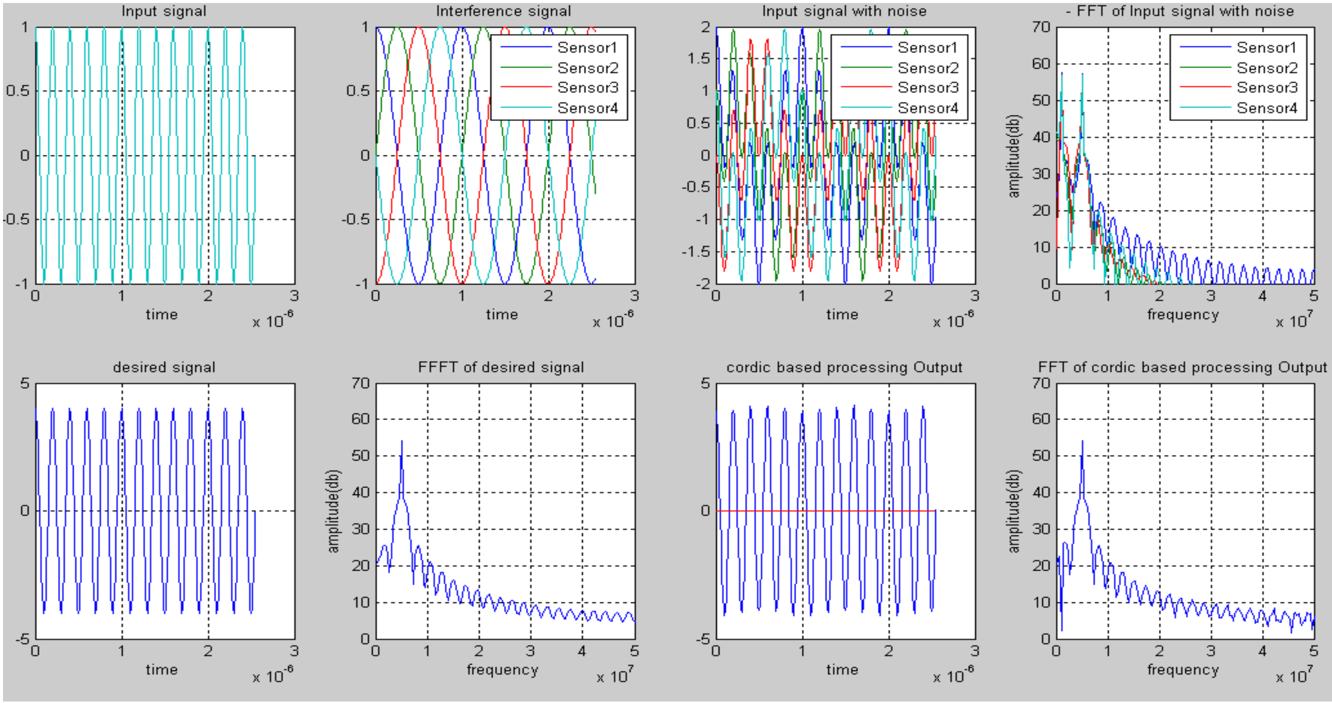

Figure 10 Result of CQR_RLS algorithm for 250 iteration

6. CONCLUSION

The CQR-RLS algorithm is an excellent way to filter out any signal using a signal reference d(n) as a model,. The proposed algorithm in which decomposition is based on Cordic algorithm is more efficient in terms of resource utilization. This adaptive algorithm is also used in beamforming for signal enhancement Array of sensors (microphones) play an important role in noise reduction and speech enhancement.Instead of taking signal from one sensor here array of sensor is used for signal as well as interference.Matlab simulation results are calculated and compared for different iteration.

7. REFERENCES

[1] S. Haykin, "Adaptive Filter Theory", 3rd ed. Upper Saddle River, NJ: Prentice-Hall, 1994.

[2] Emanuël A. P. Habets, Jacob Benesty, and Patrick A. Naylor.”A speech distortion and interference rejectionconstraint beamformer”, IEEE transactions on Audio, Speech and Language Processing, vol 20, no. 3, pp. 854-867, March 2012.

[3] Saeed V. Vaseghi, Advanced Digital Signal Processing and Noise Reduction, Third Edition, pp.45-46, 2006.

[4] Yiu,K.F.C.;HoKHOChun,Grrric,Nedelko;Lu;Yao;Shi,Xiaoxiang;Luk,Wayne S., “Reconfigurable Acceleration of Microphone Array Algorithms for Speech Enhancement”, IEEE Conference on Application Specific Systems, Architecture and processors, pp.203-208, 2008.

[5] Jing Dai and Yanmei Wang,”NLMS Adaptive Implement Based on FPGA”, IEEE Conference on Intelligent Networks and Intelligent System, pp-422-425, October 2010.

[6] B. Yang and JF Bohme, "Rotation-based RLS algorithms: Unified derivations, numerical properties, and Parallel Implementations, "IEEE Transactio on Signal Processing, vol. 40, pp. 1151-1167, May 1992.

[7] Ka Fai Ceric and Yiu,Chun Hok Ho,”Reconfigurable Acceleration of Microphone Array Algorithms for speech Enhancement”,IEEE Confrence,2008

[8] M. E. I. Martínez, “ Implementation of QRD-RLS algorithm on FPGA. Application to Noise Canceller System “, IEEE latin america transactions, vol. 9, no. 4, pp.458-462, july 2011.

[9] Leena Vachhani, K. Sridharan and Pramod Kumar Meher,” Efficient FPGA Realization of CORDIC With Application to Robotic Exploration” IEEE Transactions on Industrial Electronics, vol. 56, no.12,pp.4915-4919, December 2009.