FPGA IMPLEMENTATION OF AREA EFFICIENT ADAPTIVE FILTER USING ARRAY OF SENSORS

Anamika Gupta[1] and Rajesh Mehra[2]
[1]ME Student, Department of ECE, NITTTR, Chandigarh, India
[2]Associate Professor, Department of ECE, NITTTR, Chandigarh, India

ABSTRACT

In this paper an area efficient Cordic based adaptive algorithm has been designed and simulated for Digital Signal Processing. An adaptive filter is useful whenever the statistics of the input signals to the filter are not known. Many adaptive algorithms like LMS, NLMS and RLS are used for adaptive filtering. An efficient QR decomposition based RLS algorithm is an efficient way to filter out noise signal but its area consumption is high. So for improving area a Cordic based approach is used.Cordic algorithm is very much hardware efficient, it omits the dependence on multipliers because it implement various operations with the help of shift-add operation. Instead of taking signal from one sensor, here array of sensor (microphones) is used, whichs play an important role in noise reduction and speech enhancement. The proposed Cordic QR decomposition based adaptive algorithm is designed using MATLAB and Xilinx ACCELDSP ,synthesized with Xilinx Synthesis Tool (XST), and implemented on SPARTAN-3an(xc3s700an-5fgg484) FPGA device. The proposed algorithm has been compared with conventional QRD based recursive least square in terms of area. The results show that the performance is almost similar, but area consumption is low. The proposed design can operate at an estimated frequency of 93.7 MHz along with the minimum period of 10.6710 ns the Spartan 3an device.

KEYWORDS: *Adaptive Filter, DSP, FPGA, Matlab, Sensor*

I. INTRODUCTION

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 [1].Signals captured by a set of sensors in a communication system are mixtures of desired and undesired signals as well as noise also. Filtering algorithms are supposed, ideally, to reject the undesired signal and reduce the ambient noise [2].

In some cases when using digital filters, signals or systems may undergo some changes with time, and the exact nature of change is not predictable in such cases it is highly desirable 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 [2].Today adaptive systems have found their way into many applications where learning capacity of the system is a factor important [3].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. 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 [4].

An adaptive filter may be understood as a self-modifying digital filter that adjusts its coefficients in order to minimize an error function. This error function, also referred to as the cost function, is a distance measurement between the reference and desired signal and the output of the adaptive filter.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 firstly developed in MATLAB and after that HDL code is generated using Accel DSP software. Section 4 shows the results obtained from the design. Sections 5 and 6 cover the conclusions of this work and references consulted.

II. BASIC ADAPTIVE ALGORITHM

There is lots of algorithm which are used for adaptive filtering. An adaptive system is shown in Figure 1. As can be seen there is a filter with defined characteristics, the output is input to the adaptive algorithm system after being subtracted of a desired signal. Adaptive algorithm system can calculate the new coefficients needed to adapt the response of the filter.

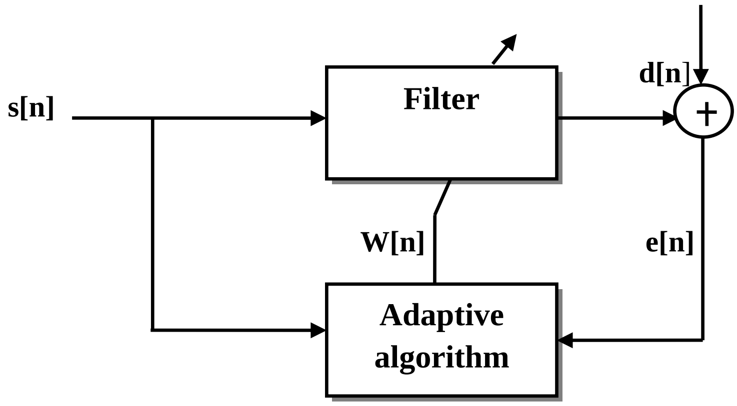

Figure 1 Diagram of an adaptive system

The operation model equations are

$$y(n) = s(n) * w(n) \quad (1)$$
$$e(n) = d(n) - y(n) \quad (2)$$
$$e(n) = d(n) - [s(n) * w(n)] \quad (3)$$

Where s(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 s(n) moves into the filter block that contains the coefficients w(n) (FIR filter) and returns a signal y(n) whose result is shown in (1). Then the result y(n) is subtracted from a signal d(n) and produces an error signal e(n) whose result is shown in equation (2), which is the parameter that tells the adaptive algorithm that algorithm response is how far to the desired signal d(n)[1]. With the help of this error signal and the input signal new coefficients w(n) are calculated for the filter using an adaptive algorithm.
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]. All 4 systems have the same general parts, an input x(n), a desired result d(n), an output y(n), an adaptive transfer function w(n), and an error signal e(n) which is the difference between the desired output u(n) and the actual output y(n).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.

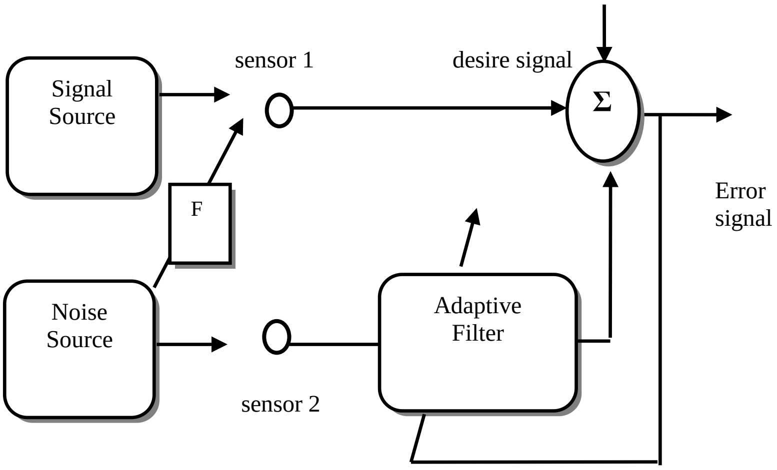

Figure 2 Adaptive Noise Cancellation Configuration

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.

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 [6].An improved version of LMS is presented in paper[7] which shows that ILMS has faster convergence rate.On the other hand RLS (Recursive Least Square), is generally preferred for its fast convergence. The demand for fast convergence and less MSE level cannot be met by conventional adaptive filtering algorithms such as LMS. The best choice is the block recursive least squares (RLS) algorithm. Block Recursive Least Squares algorithms are known to exhibit better performances [8].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 are least squares, SVD (Singular Value Decomposition) and QR decomposition [9]. QR based adaptive algorithm is used to solve linear least square problems.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. A common problem encountered in many systems is the presence of echoes and noise. Removal of these echoes and requires the precise knowledge of the impulse response of the noisy path, which may be time varying. In recent years, an adaptive filter is widely used for cancellation of noise component which is overlap with unrelated signal in the same frequency range [10].

III. QR DECOMPOSITION

The QR algorithm, which is based on the QR decomposition of **A**, is still considered one of the most important methods. The algorithm based on QR decomposition decomposes **A** into a unitary matrix **Q** and a upper triangular matrix **R**, instead of the lower and upper triangular matrices from elimination [4]. The QR algorithm uses successive unitary transformations, which render the method superior to its predecessor with respect to numerical stability and computational requirements

$$A = QR \tag{4}$$

Where

$$Q = \left[q1 \quad q2 \cdots \quad \cdots qN \right] \tag{5}$$

$$R = \begin{bmatrix} r11 & r12\cdots & r1N \\ 0 & r22\cdots & r2N \\ 0 & & rNN \end{bmatrix} \tag{6}$$

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.
The QR decomposition method starts from the data matrix using unitary transformation [9]. An error vector is defined as

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

Cost function can be define as

$$\mathrm{E}(n) = \sum_{t=1}^{n} |e(i)|^2 \tag{8}$$

The cost function may be expressed as [9] for a given matrix Q(n),

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

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

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

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

Where 0 is a zero matrix of dimension (n-k) x k and R(n) is an upper triangular matrix dimension k x k [9] .The desired signal vector, after being converted, is defined by:

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

Where p (n) is a vector of elements k x1, v (n) is a vector (n-k) x 1 element, then we can rewrite the cost function as follows:

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

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

Basically the main aim of the adaptive algorithm is to minimise the cost function [9]. The least squares estimation for the weight vector must satisfy that:

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

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

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

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

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

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. The purpose of using array of sensors is that, single-sensor noise reduction technique is found not to be able to achieve speech intelligibility improvements[11].

IV. CORDIC RLS ALGORITHM

A CORDIC describes a method to perform a number of functions, including trigonometric, hyperbolic, exponential, linear and logarithmic functions.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 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 [12]. The CORDIC is a shift-and-add technique for computing a large class of mathematical functions in hardware. It is a special purpose computer meant for the real-time calculation of trigonometric and exponential functions by the use of iterative vector rotations. The algorithm can be derived from the rotation transform.

$$x' = x\cos\phi - y\sin\phi \tag{17}$$

$$y' = y.\cos\phi - x.\sin\phi \tag{18}$$

On rearrangement of the terms, this can be given as

$$x' = \cos\phi[x - y.\tan\phi] \tag{19}$$

$$y' = \cos\phi[y + x.\tan\phi] \tag{20}$$

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 $\phi = \pm 2^{-i}$, the multiplication by the tangent can be greatly simplified as it can be implemented using simple shift and addition operations. The CORDIC method can be employed in two different modes, known as the “rotation” mode and the “vectoring” mode[13].For implementing QR decomposing RLS algorithm there are three methods for factorization of matrix. Among these the 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 3.

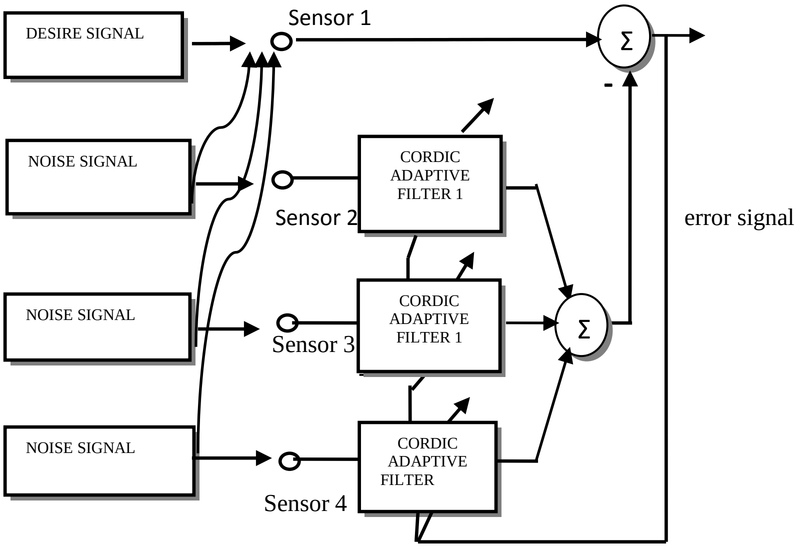

Figure 3 Sensor array based adaptive noise cancelation

V. DESIGN PLATFORM

Matlab and Accel DSP tool are used to implement this algorithms. AccelDSP Synthesis Tool, the only DSP (Digital Signal Processing) synthesis tool that allows transforming a MATLAB floating-point design into a hardware module that can be implemented in a Xilinx FPGA. The AccelDSP Synthesis Tool features an easy-to-use Graphical User Interface that controls an integrated environment with other design too such as MATLAB and Xilinx ISE tools [14].The AccelDSP synthesis flow is shown in figure 4.

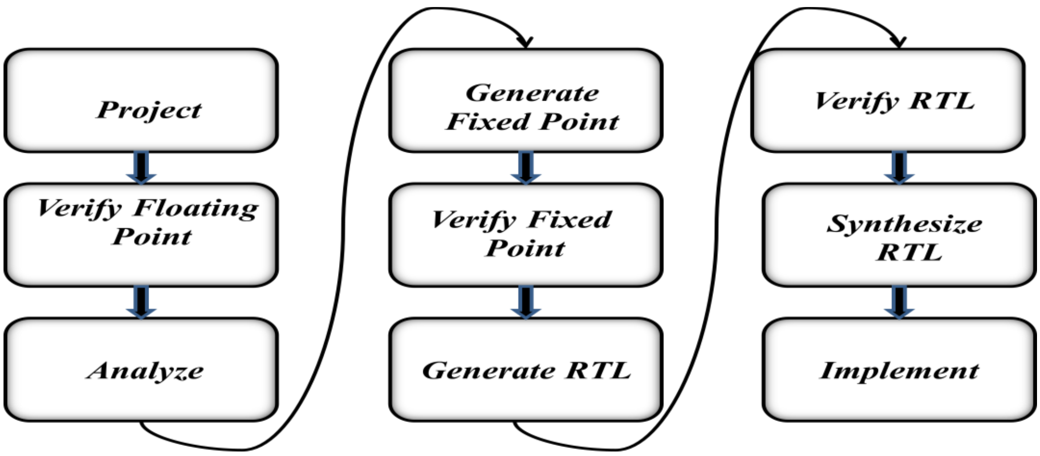

Figure 4 ACCEL DSP Synthesis Flow

VI. RESULTS

In order to compare the results obtained in the simulation of cordic based qr decomposition rls algorithm a series of graphs are developed for different adaptive iteration. Figure 5 shows the input signal and interference signal which are useful for the adaptive system. Figure 6 shows the Signal contaminated with interference and its frequency response. Simulation is done for different iteration, after 150 iterations the output signal is almost similar as the desire one with almost negligible noise. Figure 7 shows the desire signal and its frequency response and figure 8 shows the output of the cordic based adaptive system.

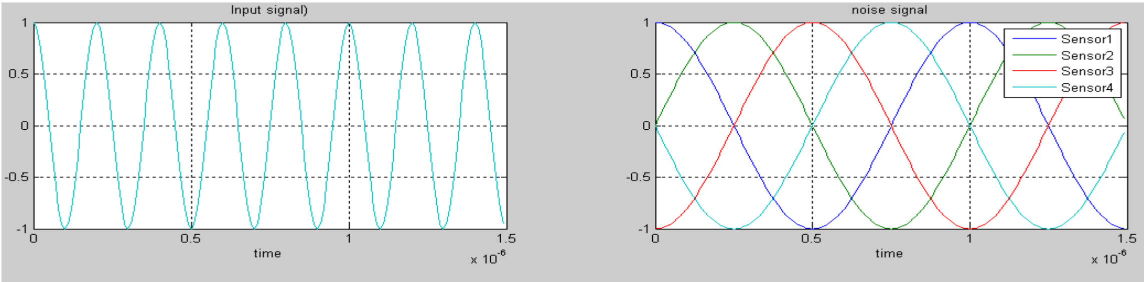

Figure 5 Input signal and interference signal

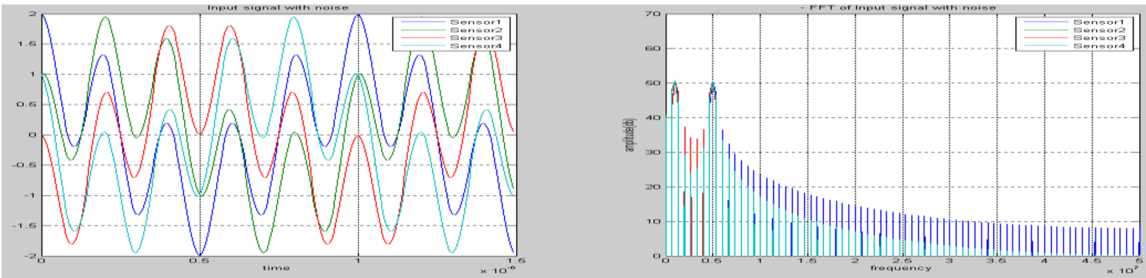

Figure 6 signal with interference and its frequency response

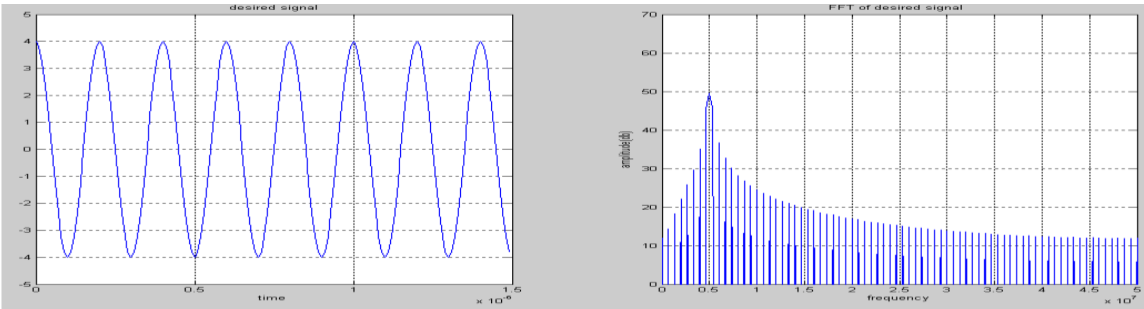

Figure 7 Desire signal and its frequency response

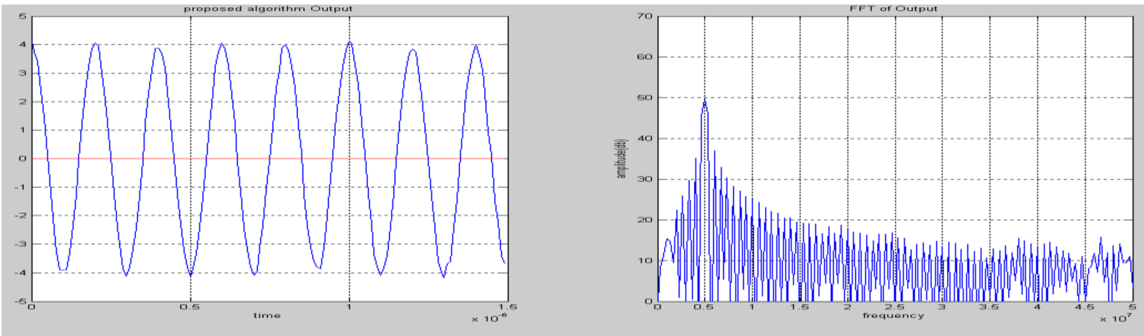

Figure 8 Result of proposed algorithm for 150 iterations

As for the resources used in implementation of the algorithm CQR-RLS the matlab code is firstly converted in VHDL code with the help of AccelDsp. Table 1 shows the consumption of resources. The Whole system is implemented on the SPARTAN3an (xc3s700an-5fgg484) FPGA architecture, which has 11,776 flip flop, and the proposed algorithm is using 6% of the same. The proposed algorithm has been compared with conventional qrd based recursive least square in terms of area. Instead of taking signal from one sensor here array of sensor is used. From table 1 we can conclude that the proposed structure shows 33% of reduction in LUTS and almost 33% reduction in slices as compared to conventional qr decomposition based recursive least square, while the consumption of flip flop is almost same. Figure 8 shows the comparison between the proposed and existing result.

Table 1. Area Comparison for QRD_RLS and proposed QRD_RLS

	Proposed algorithm	QR decomposition based algorithm[8]	% saving
Number of Slice Flip Flops	812	789	0%
Number of 4 input LUTs	1,944	5819	33%
Number of occupied Slices	1,054	2987	33%

Table 2. Performance evaluation

Clock Name	Requested Frequency	Estimated Frequency	Estimated Period	Max Throughput	Input Sampling
Clock	130.0 MHz	93.7 MHz	10.6710 ns	141	664.624 KSPS

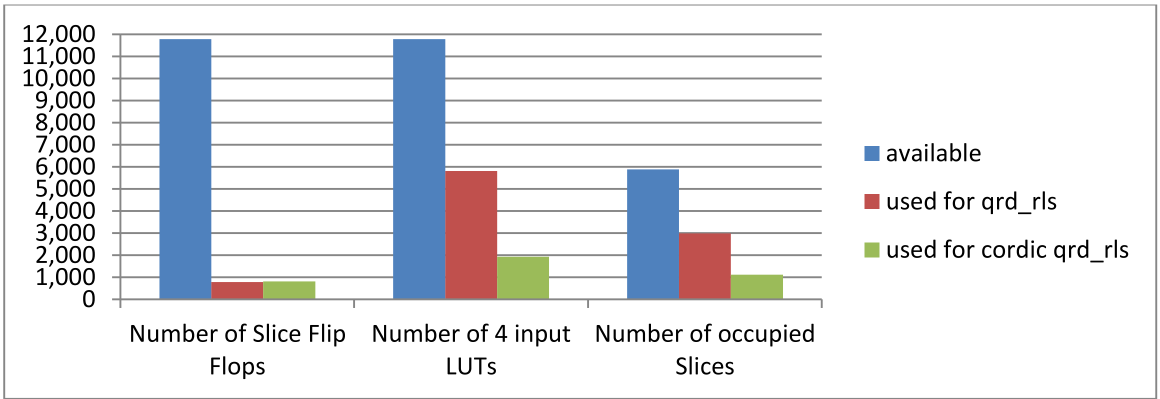

Figure 9 Comparison between the proposed algorithm and existing result

VII. CONCLUSION

Least mean-square (LMS) algorithm is commonly used in adaptive filtering but RLS (Recursive Least Square), is generally preferred for its fast convergence. In this paper an area efficient Cordic based QR decomposition RLS adaptive algorithm has been designed and simulated with the help of Matlab. This algorithm is implemented on FPGA using ACCEL DSP software. The result shows that Cordic based adaptive algorithm is more area efficient in comparison to conventional QR decomposition based algorithm, because is consuming fewer resources in comparison to conventional QR RLS algorithm [9]. The proposed structure shows 33% of reduction in memory (LUTS) and almost 33% reduction in slices as compared to conventional one. Cordic based QR-RLS algorithm is an excellent way to filter out any signal using a signal reference d(n) as a model. The total CPU time to execution completion is 10.6710 ns and total estimated frequency is 93.7 MHz .

VIII. FUTURE WORK

QRD algorithm is also used in beamforming for signal enhancement.Multi-sensors noise reduction techniques combine and filter different sensor signals in order to achieve an SNR improvement. Future work can be done in the area of SNR improvement. For hearing aid applications, it is widely accepted that multi-sensors noise reduction or beamforming can achieve significant speech intelligibility improvements.

REFERENCES

[1]Yiu ,K.F.C.; HoK HO Chun; 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.

[2] S. Haykin, "Adaptive Filter Theory", 3rd edition. Upper Saddle River, NJ Prentice-Hall, 1994.
[3] Emanuël A. P. Habets, Jacob Benesty, and Patrick A. Naylor. "A speech distortion and interference rejection constraint beamformer", IEEE transactions on Audio, Speech and Language Processing, Volume 20, Issue No.3, pp. 854-867, March 2012.
[4] Saeed V. Vaseghi, "Advanced Digital Signal Processing and Noise Reduction", Third Edition, pp.45-46, 2006.
[5] Jos´e Antonio Apolin´ario Jr., "QRD-RLS Adaptive Filtering", Springer Science, Business Media, pp.25-27, September 2008.
[6] Jing Dai and Yanmei Wang, "NLMS Adaptive Implement Based on FPGA", IEEE Conference on Intelligent Networks and Intelligent System, pp. 422-425, October 2010.
[7] Ma Shengqian, Xu Guowei, Ma Zhifeng, Wei Shuping, Fan Manhong, "Research on adaptive noise canceller of an improvement LMS algorithm" International Conference on Electronics, Communication and Control, pp.1161-1164, October 2011.
[8] S.VijayaLakshmi1, K.Raghuram, "FPGA Implementation of the Block RLS Algorithm", International Journal of Modern Engineering Research, Volume 2, Issue No. 5, pp. 3479-3472, September 2012.
[9] M. E. I. Martínez, "Implementation of QRD-RLS algorithm on FPGA. Application to Noise Canceller System ", IEEE Latin America transaction, Volume 9, pp.458-462, Issue No.4, July 2011.
[10] M.A.Raja, M. Somiya, K.Divya," Adaptive Noise Cancellation for Speech processing", International Journal of communication Engineering, Volume 3, No.3, PP. 33-39, March 2012.
[11] P. C. Loizou and G. Kim. "Reasons why current speech-enhancement algorithms do not improve speech intelligibility and suggested solutions"IEEE Transaction on Audio, Speech, Lang. Processing, Volume 19,Issue No.1, pp. 47–56, Jan. 2011.
[12] Leena Vachhani, K. Sridharan and Pramod Kumar Meher," Efficient FPGA Realization of CORDIC With Application to Robotic Exploration" IEEE Transactions on Industrial Electronics, Volume 56, Issue No.12, pp. 4915-4919, December 2009.
[13] Sukhpreet Kaur, Kulbir Singh, "Implementation of High Speed Fixed Point CORDIC Techniques", International Journal of Computer Applications , Volume 56, Issue No.3, pp.35-41, October 2012.
[14]. Accel DSP Synthesis Tool, Release 10.1.1, April 2008:http://www.xilinx.com/acceldsp_user.pdf.

Rajesh Mehra received the Bachelors of Technology degree in Electronics and Communication Engineering from National Institute of Technology, Jalandhar, India in 1994, and the Masters of Engineering degree in Electronics and Communication Engineering from National Institute of Technical Teachers Training & Research, Panjab University, Chandigarh, India in 2008. He is pursuing Doctor of Philosophy degree in Electronics and Communication Engineering from National Institute of Technical Teachers Training & Research, Panjab University, Chandigarh, India. He is an Associate Professor with the Department of Electronics & Communication Engineering, National Institute of Technical Teachers‟ Training & Research, Ministry of Human Resource Development, Chandigarh, India. His current research and teaching interests are in Advanced Signal Processing, VLSI System Design and Embedded System Design. He has authored more than 100 research publications including more than 50 are in Journals. Mr. Mehra is member of IEEE and ISTE.

Anamika Gupta received the Bachelors of Technology degree in Electronics and Communication Engineering from Rohilkhand University, Bareilly, Uttar Pradesh, India in 2007. She is shortly finishing her M.E. from National Institute of Technical Teachers Training and Research, Ministry of Human Research Development, Chandigarh, India. Her area of interest is Digital Signal Processing, VLSI System Design and Digital design.