Implementation of RLS AlgorithmbyAdaptive Beam Forming and its Comparison for Active Phased Array Unit AirBorne Application Radar

M.C. Narasimhamurthy[1], G V Jayaramaiah[2]

[1] DRDO, Ministry of Defence, Govt of India, Bengaluru, India

[2]Professor, Dean R&D, Head of EEE Department, Dr AIT, Bengaluru, India

ranjithasres@gmail.com

Keywords: Uniform Linear Array, Radar, Sample Matrix Inversion, Recursive Least Squares, Least Mean Squares, Differential Steepest Descent, Xilinx System Generator,hardware description language

Abstract

The use of array of sensors has become an attractive option in radars and communication because of the host of advantages it offers for instance, in terms of directivity and beam width as compared to a single sensor. An adaptive array is a system consisting of an array and a real time adaptive signal processor. It automatically adjusts the array beam sensitivity pattern so that a measure of the quality of the array performance is improved as compared to that of a conventional array.

Several algorithms have been proposed and studied for adaptive beam forming. Least mean squared (LMS) and Recursive least squares (RLS) are two recognized algorithms. SMI algorithm is the most basic which relies on matrix inversion. The goal of the LMS algorithm is to adaptively produce weights that minimize the mean squared error between a desired signal and the arrays output. The RLS algorithm on the other hand offers better convergence rate, steady-state mean square error, and the parameter tracking capability over the LMS-based algorithms.

This paper deals with the signal processing algorithms to be used in the adaptive processor unit of the antenna array. The simulation of well-known algorithms such as SMI, LMS, NLMS and RLS have been carried out and compared with respect to certain parameters. Quantization effects have been applied to the inputs through mat lab (fixed point conversion) and the results of the algorithm to these values have been compared to non-quantized inputs. Finally, the most frequently used LMS algorithm has been implemented on hardware.

Introduction

This introduces the concepts about adaptive array processing. A brief introduction is given to our problem statement and the objective is mentioned.

1.1 Adaptive Array Antenna and Beamforming.

A system consisting of an antenna array and an adaptive processor that can perform filtering in both the space and the frequency domains, thus reducing the sensitivity of the signal receiving system to interfering directional noise sources.

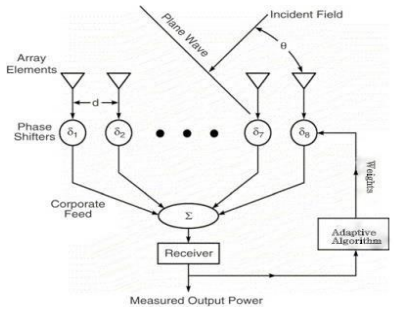


Figure: 1

An adaptive beam former is a system which performs adaptive spatial signal processing with an array in order to transmit or receive signals in different directions without having to mechanically steer the array. The main distinction between an adaptive beamformer and a conventional beamforming system is the ability of the former to adjust its

performance to suit differences in its environment. A particularly important feature, in military applications, is the potential for an adaptive beamformer to reduce sensitivity to certain directions of arrival so as to counteract jamming by hostile transmissions.

For adaptive beamforming, schemes such as LMS, SMI and RLS are used.

1.2 The Complete Problem Picture

The fundamental problem facing the adaptive array designer is to improve the reception of a certain desired signal in the presence of undesired interfering signals. The terms "desired signal" and "interfering signals" imply that the characteristics of these two signal classes are different in some respect, and that this difference may be exploited in order to realize the desired signal reception improvement. For example, if the direction of arrival of the desired signal is known (or can be deduced), then any signals arriving from different directions can be suppressed by the introduction of array pattern nulls in those directions.

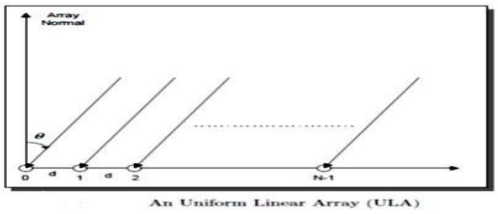
Statement: Consider an antenna array that receive various signals from several directions in space. The spatial filtering problem is to optimize the antenna array pattern such that maximum possible gain is placed on the direction on the desired signal and nulls are placed on the directions of the interferers.

Problem Setup: Let s(t) denote the desired communications signal arriving at the array at an angle $\boldsymbol{\theta}_s$ and $\boldsymbol{u_i}$(t) denote the $N_u$interfering signals arriving at the array at angle of incidences$\boldsymbol{\theta_i}$respectively. It is assumed here that the adaptive processor has no a-priori knowledge of thedirection of arrivals (DOA's) of both the signal and the interferers.

The generalized spatial filtering problem is to estimate the signal s(t) from the received signal x(t)such some measure of error between the estimate ˆs(t) and the original signal s(t) is minimized, based on certain (optional) constraints such as constant modulus, exact position of nulls, etc.

The generalized spatial filtering problem is to estimate the signal s(t) from the received signal x(t)such some measure of error between the estimate ˆs(t) and the original signal s(t) is minimized, based on certain (optional) constraints such as constant modulus, exact position of nulls, etc.

The goal is to develop a general framework for the simulation of the LMS and SMI algorithms and to simulate their performance for various configurations of the antenna array, the number of interferers (Nu), the statistical properties of the signal and the interferers and the statistical relations between the signal and the interferers (i.e.,correlated or uncorrelated, etc.).



An Uniform Linear Array (ULA)

Figure: 2

The Adaptive Processor: The adaptive array used in this project is a Uniform Linear Array (ULA) of N elements. Each of the elements in the ULA is scaled by their corresponding weights and is summed to form the output signal. The weights are in turn controlled by an adaptive algorithm that operates on the received signal and the desired signal. The antenna array and the adaptive processor configurations are shown below

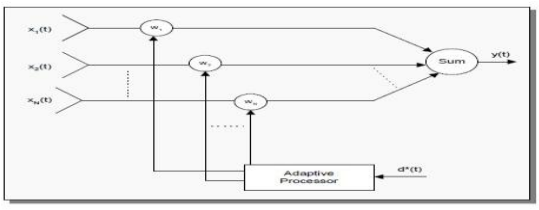


Figure: 3

In the case of an ULA, the array steering vectors are given by,

$$\mathbf{a}(\theta) = \left[1, e^{jkdsin(\theta)}, e^{j2kdsin(\theta)}, \ldots, e^{j(N-1)kdsin(\theta)}\right]$$

The output of the adaptive processor y(t) is given by,

$$y(t) = \mathbf{w}^H \mathbf{x}(t)$$

where, w represents the N-element weight vector. The reference signal d(t) generated at the receiver is usually assumed to have similar statistical (first and second order) properties as the transmitted signal s(t). Various methods are described in literature for generating this reference signal d(t).

The project deals with Adaptive beamforming and Interference cancellation. Simulation of several algorithms which perform interference cancellation is carried out, their results compared and the pros and cons discussed. This has following specific objectives:

i) To compare steady state errors and convergence rate for the adaptive algorithms and to bring out the constraints in implementing these algorithms on hardware.

ii) To implement LMS algorithm on an FPGA.

2.2 Two Approaches to Problem Solving.

The adaptive array processing unit which essentially has the algorithm takes in two inputs depending on the array configuration being used. They are called the traditional configuration and the side lobe canceller configuration which is more of a subset of the first configuration.

1) Traditional Adaptive Array Configuration

The traditional array configuration is as shown in the Figure below, the first input which is fixed in both the configurations is

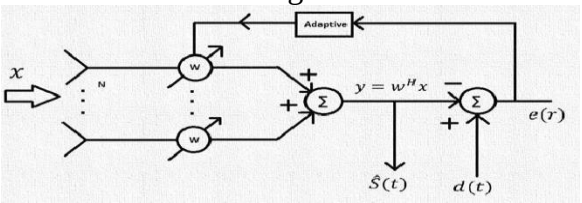


Figure:4

the final summed up output after it passes through the weights. The other input is an estimate of the desired signal which we must actually receive. This estimate can be taken in through a highly directional antenna or can be present beforehand. As clearly seen, the error signal e(r) can easily be isolated by a simple subtraction operation and this can be used to update the weights. The problem with this configuration is that the desired output will not preexist and cannot be determined, hence this approach is mainly for theory sake and not very practical.

2)Side lobe Canceller Configuration

Unlike the traditional approach this configuration doesn't assume the desired signal to preexist. As seen in the diagram below there exists two antennas, a main antenna which is directed (having main lobe) tow

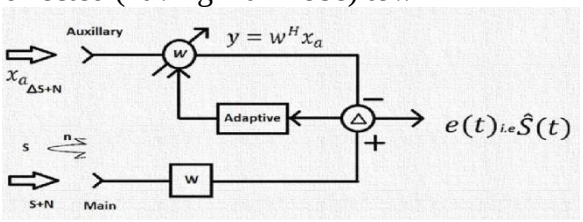


Figure:5

The main antenna therefore picks up signal and noise, but the auxiliary antenna picks up more of noise and a bit of signal as well since it is isotropic. An effective subtraction of the signals from these antennas is done which will give $(S + N) - (\Delta S + N) = S - \Delta S$.

The adaptive processor makes sure that the weights are adjusted so that N in both channels is equally subtracted and hence the desired signal S can be obtained.

2.3: Ideal Solution

2.3.1: Weiner Solution

The ideal solution to the adaptive filter problem lies in mathematics in the wiener hopf equations, which takes in the minimum-square value of the estimate error and considers this the cost function and tries to reduce this. Therefore, the cost function will have a distinct minimum that uniquely defines the optimum value of the filter [9].

2.3.2: Sample Matrix Inversion (SMI) algorithm.

The SMI algorithm [9] has a faster convergence rate since it employs direct inversion of the covariance matrix R. Let us consider the equations for the covariance matrix R and the correlation matrix r. .

2.4 Differential Steepest Descent (DSD) algorithm.

The method of steepest descent is the simplest of the gradient methods. Imagine that there's a function F(x), which can be defined and differentiable within a given boundary, so the direction it decreases the fastest would be the negative gradient of F(x). To find the local minimum of F(x), The Method of the Steepest Descent is employed, where it uses a zig-zag like path from an arbitrary point Xo and gradually slide down the gradient, until it converges to the actual point of minimum [9]. To put this step into a function, one can get:

$$x_{k+1} = x_k - \lambda_k \nabla F(x_k) = x_k - \lambda_k g(x_k)$$

2.5 Least Mean Squares (LMS) algorithm.

Least mean squares (LMS) algorithms are a class of adaptive filter used to mimic a desired filter by finding the filter coefficients that relate to producing the least mean squares of the error signal (difference between the desired and the actual signal). It is a stochastic gradient descent method in that the filter is only adapted based on the error at the current time [9].

The algorithm is described by the following equations:

$y(n)=w^T(n)x(n)$ $e(n)=d(n)-y(n)$ $w(n+1) =w(n)+2\mu e(n)x(n)$

2.7: Recursive Least Squares (RLS) Algorithm

Least-squares algorithms aim at the minimization of the sum of the squares of the difference between the desired signal and the model filter output. When new samples of the incoming signals are received at every iteration, the solution for the least-squares problem can be computed in recursive form resulting in the recursive least-squares (RLS) algorithms [9].The RLS algorithms are known to pursue fast convergence even when the Eigen value spread of the input signal correlation matrix is large. These algorithms have excellent performance when working in time-varying environments. All these advantages come with the cost of an increased computational complexity and some stability problems, which are not as critical in LMS-based algorithms.

2.8: Quantization Effects

In digital signal processing, **quantization** is the process of approximating a continuous range of values (or a very large set of possible discrete values) by a relatively-small set of discrete symbols or integer values [9].

A common use of quantization is in the conversion of a discrete signal (a sampled continuous signal) into a digital signal by quantizing. Both of these steps (sampling and quantizing) are

2.9: Hardware Implementation

A field-programmable gate array (FPGA) is an integrated circuit designed to be configured by a customer or a designer after manufacturing—hence "field-programmable". The FPGA configuration is generally specified using a hardware description language (HDL), similar to that used for an application-specific integrated circuit (ASIC) (circuit diagrams were previously used to specify the configuration, as they were for ASICs, but this is increasingly rare). Contemporary FPGAs have large resources of logic gates and RAM blocks to implement complex digital computations.

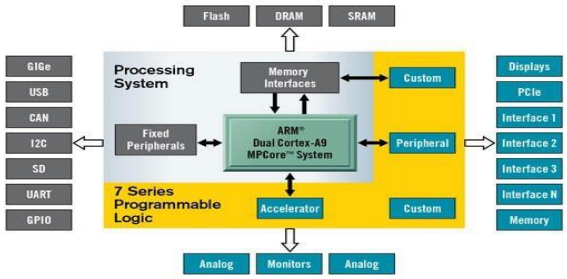


Figure:6

Applications of FPGAs include digital signal processing, software-defined radio, ASIC prototyping, medical imaging, computer vision, speech recognition, cryptography, bioinformatics, computer hardware emulation, radio astronomy, metal detection and a growing range of other areas.

Methodology

The algorithms discussed in the literature survey were coded in matlab (version R2009b) and simulation results were studied. Xilinx System Generator (XSG) was used to generate the hardware description language (HDL) code through which hardware implementation was achieved.

3.1: Data Acquisition

To test the algorithms, we used input data consisting of a sine wave of 100 Hz frequency which was generated using sine () function and noise was added to it using randn () function in matlab.

Later, recorded RADAR data was applied as inputs to all the algorithms which were given as. mat files to us. The results obtained and comparisons are shown in the results section.

The RADAR data supplied to us consisted of two jammer signals and the target. Specifications were

3.2: Simulation on Matlab (version R2009b)

Over the course of the project we worked together and implemented all of the algorithms mentioned in the literature survey mainly Sample Matrix Inversion (SMI), Least Mean Squares (LMS), Normalized Least Mean Squares (NLMS) and Recursive Least Squares (RLS).

These algorithms were coded in matlab (version R2009b) which is a numerical computing environment and fourth-generation programming language developed by Math Works. The mathematical expressions of each algorithm were directly written in matlab (version R2009b) and input data supplied in matrix form for computation. The codes written were independent of the number of input samples or number of jammer signals present (although only 2 jammers were considered throughout the scenario). The outputs of each of these algorithms contained plots of the input compared to the output, output compared to the ideal value, weight updation and the mean square error.

To understand the entirety of the problem of beamforming through every step, we generated our own pulse radar signal which consisted of sine wave pulses at regular intervals as shown in Figure 3.1.

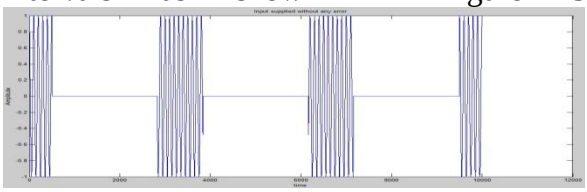


Figure 7

We added thermal noise to this, which is inevitably present in any electronic equipment. We then generated the antenna pattern which looks as shown in Figure 3.2.

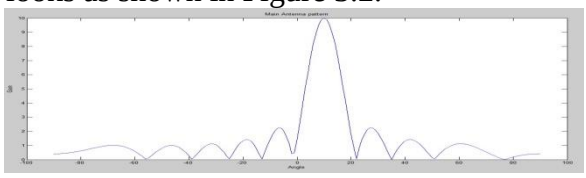


Figure 8

We took the SNR (signal to noise ratio), JNR (Jammer to noise ratio) and angle of incidence of the jammers (angle in degrees) as inputs to the code and then added the jammer noise to the main signal. This noisy signal was then run through the adaptive algorithm to get back the originally generated pulse signal with of course thermal noise. The results to this approach can be seen in the appendix (Figure A1 and Figure A2).

Quantization:

he quantization effects were tested on the matlab (version R2009b) signals generated by us and not the actual radar data because the

acquired RADAR signals were already quantized to a certain precision.
The effect of quantization on these algorithms has also been tested through matlab. Embedded fi blocks have been used for conversion of inputs to fixed point. These converted values were then supplied to the algorithms to test their output. The fixed point conversion was done for many values such as word length – 16, fraction part – 15, and the word length – 10 and fraction part – 8.
3.3: Hardware Implementation

3.3.1: Steps in VHDL Design Process
VHDL is a general-purpose hardware description language which can be used to describe and simulate the operation of a wide range of digital systems starting with those containing only a few gates up to systems formed by the interconnection of many complex integrated circuits. VHDL can describe a system at the behavioral, data flow and structural levels. At the behavioral level, the digital system is described by the functions that it performs. It provides no details as to how the design is implemented. Complex hardware units are first described at the behavioral level to simulate and test the design ideas. At the data flow level, the system is described in terms of the flow of control and the movement of data. This involves the architecture of busses and control hardware. The structural description is the lowest and most detailed level of description and is the simplest to synthesize into hardware. It includes a list of concurrently active components and their interconnections.
The design flow is described in the Figure

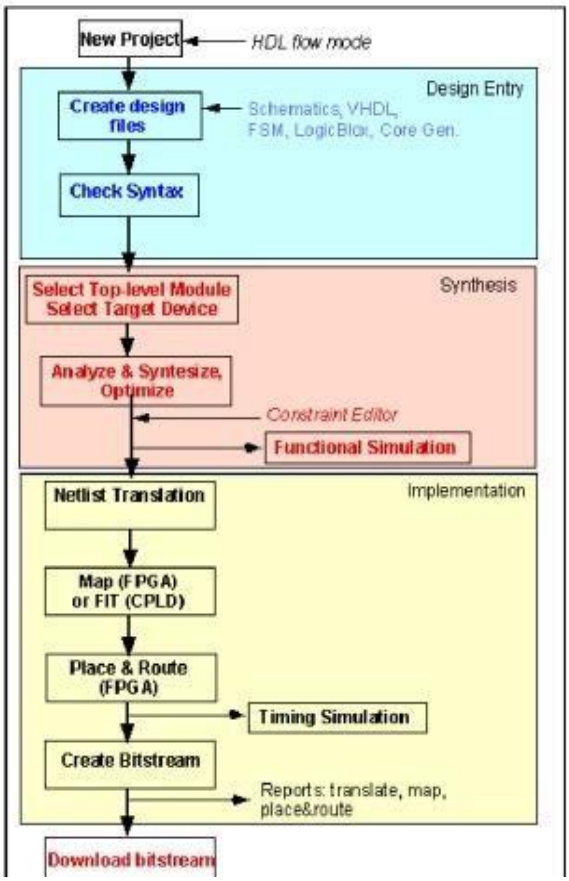


Figure: 9

3.3.2: Xilinx System Generator Model
The standard LMS algorithm according to the equationsy(n)=w^T(n)x(n) e(n)=d(n)-y(n) w(n+1)=w(n)+2μe(n)x(n) is implemented as shown in Figure 3.2.2, the model which is implemented using the Xilinx System Generator takes in the inputs of RADAR data directly from the BRAM (Block Random Access Memory). The other blocks are used to realize the three equations of the LMS algorithm. The outputs which are mainly the resultant filtered output and weights are sent back to the workspace.

Results

4.1: SMI Results
A simple sine wave with 1khz frequency and amplitude 1 was generated on matlab and then noise added to it by random generation. This mixed wave with signal and noise was supplied to the ideal

algorithm. The result obtained is as shown. Similarly, the same algorithm was applied to a radar signal with artificial noise applied to it.

Simulation results for pure Sine Wave:

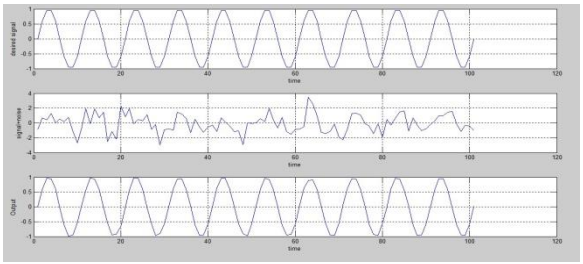


Figure: 10

Simulation results for radar data:

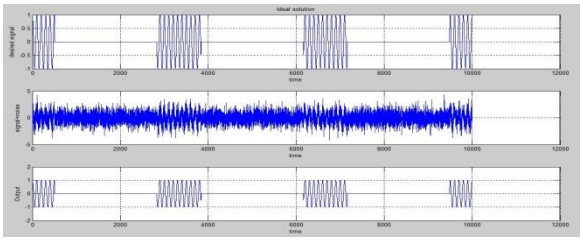


Figure: 11

4.2: DSD Results

The DSD algorithm is very close to the ideal algorithm. Hence the output obtained is as good as the best obtainable results (i.e desired signals).

4.2.1: Sine Wave Results

Output Red Compared to Input Blue signal

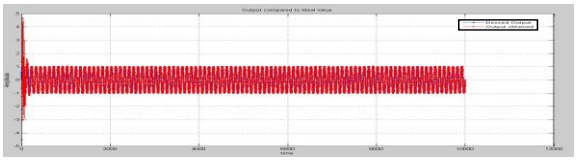


Figure 12

Output (Red) compared to the ideal output with thermal noise (Blue) which needs to be obtained:

4.2.2: Radar Data Results

Simulation results for RADAR data:

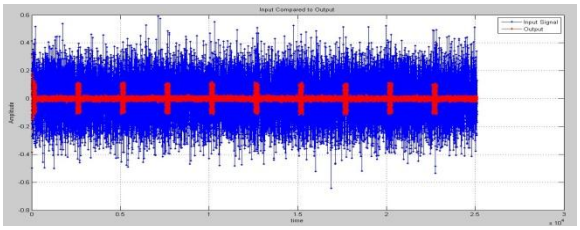


Figure13

4.3: LMS Results

In this section we evaluate the performance of LMS algorithms in noise cancellation. Input signal is radar data which has the noisy signal as well. As it converges to the correct filter model, the filtered noise is subtracted and the error signal should contain only the original signal. The desired signal is composed of thermal noise and the actual radar signal coming back from the target. The parameter μ is varied. Various outputs are obtained for various step size i.e. μ = 0.0009, 0.009 system reaches steady state faster when the step size is larger.

4.3.1 Sine Wave Results

Output (Red) compared to input (Blue) Signal.

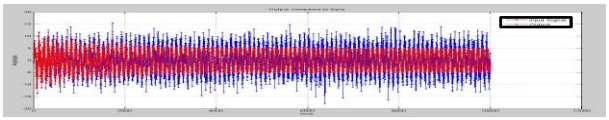


Figure 14

Output (Red) compared to the ideal output with thermal noise (Blue) which needs to be obtained:

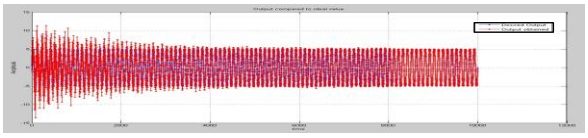


Figure15

. The greatest advantage however of the NLMS algorithm is that the power difference between the inputs does not affect the output as much as it does in the LMS algorithm. The simulation results below clearly show that. When applied to RADAR data with large

differences between the noise signals, the LMS algorithm shows more error as compared to the NLMS algorithm in the areas of signal presence.

Error obtained for LMS algorithm:

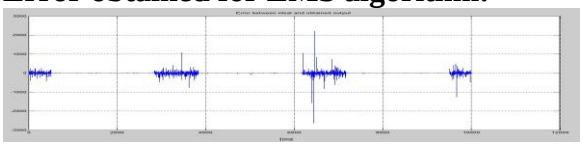


Figure: 16

4.6: RLS Results

The RLS algorithm was applied to recorded RADAR data. The simulation results obtained are as shown in the *Figure*s. The output is compared to the ideal desired result. The weight updation which takes place and the error difference between ideal and obtained output are also displayed. Clearly the simulation results show the faster convergence rate of the RLS algorithm.

Output (Red) compared to the ideal output with thermal noise (Blue) which needs to be obtained:

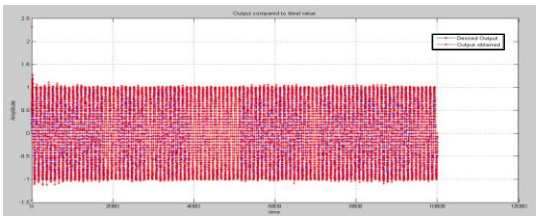


Figure: 17

4.6.2: Radar Data Results

lOutput obtained (Red) when an input signal (Blue) is applied with noise:

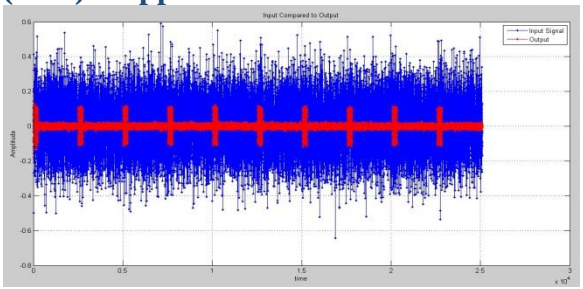


Figure:18

Output (Red) compared to the ideal output with thermal noise (Blue) which needs to be obtained:

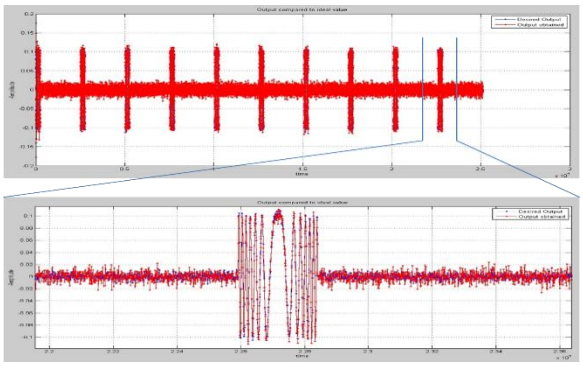


Figure: 19

Weight Updation :

4.7: Comparison between LMS and RLS

In the LMS algorithm, the correction that is applied in updating the old estimate of the coefficient vector is based on the instantaneous sample value of the tap-input vector and the error signal. On the other hand, in the RLS algorithm the computation of this correction utilizes all the past available information**.**

The major difference between the LMS and RLS algorithms is therefore the presence of

RLS is more complex to implement because of the squared number of unknown system parameters.

Setting the value of the step size parameter to a certain value and running both the LMS and RLS algorithm to the same input, we simulated outputs such that the mean square error after a certain period was almost similar (better for RLS). The results are shown below, the square error performance is slightly better than LMS for the RLS but what is noticeable is the convergence rate. Clearly the LMS takes a lot of time to adapt.

Parameter	**DSD**	**LMS**	**NLMS**	**RLS**
Convergence Rate	80 samples	1100 samples	650 samples	20 samples
Steady state MSE	$4.1 * 10^{-4}$	$5.5 * 10^{-4}$	$5.4 * 10^{-4}$	$4.9 * 10^{-4}$
Computation Time	16.82s	18.03s	18.10s	25.89s

LMS Algorithm:

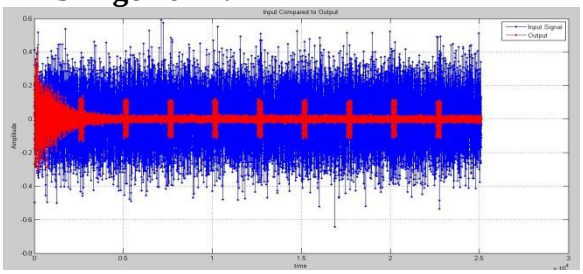


Figure: 20

4.10: Comparison of Algorithms

The performance of these algorithms can be compared according to three parameters which are rate of convergence, steady state MSE and computation time. The parameter values which we obtained in our simulation are tabulated as follows.

The DSD, LMS and NLMS algorithms were simulated with a step size of 0.009 and the RLS algorithm was simulated with $\lambda = 0.999$ and $\delta = 0.0001$

Conclusion

The work completed during this mainly focused on learning about existing algorithms present for adaptive signal processing such as the DSD, LMS, NLMS, RLS. Based on the results obtained we can conclude that the algorithm to be used for any application solely depends on the application at hand. Even though RLS algorithm produced the best results with minimum convergence rate and good mean square error performance, the computation complexity is high. LMS on the other hand has very low computational complexity but has a trade-off of slower convergence rate and higher mean square error. Based on the comparison of the algorithm done in this project (Table 4.10.1), it is evident that the RLS algorithm is the best suitable algorithm for RADAR data processing.

FPGA implementation of the LMS algorithm was achieved on a Virtex-5 kit. The output obtained was not as accurate as of that obtained on matlab (version R2009b).

References

1) Adaptive Antenna Systems, PROCEEDINGS OF THE IEEE, VOL. 55, NO. 12, DECEMBER 1967. B. Widrow, MEMBER, IEEE, P. E. Mantey, Memberie, EE, L. J. Griffiths, Student Member, IEEE, and B. B. Goode, Student Member, IEEE.

2) Adaptive Noise Cancelling: Principles and Applications, Bernard Widrow, Senior Member, IEEE, John R. Glover, JR., Member, IEEE, John M. Mccool, Senior Member, IEEE, John Kaunitz, Member, IEEE, Charles s. Williams, Student Member, IEEE, Robert H. Hean, James R. Zeidler, Eugene Dong, JR., and Robert C. Goodlin.

3) Echo Cancellation, Mathieu C. Hans and Thomas M. Levergood, CRL 94/7, Digital Equipment Corporation Cambridge Research Lab, October 13, 1994.

4) Beamforming: A Versatile Approach to Spatial Filtering, IEEE ASSP magazine April 1988. Barry D. Van Veen, Kevin M. Buckley.

5) A Novel LMS Algorithm Applied to Adaptive Noise Cancellation, J. M. Górriz, Javier Ramírez, S. Cruces-Alvarez, Carlos G. Puntonet, Elmar W. Lang, and Deniz Erdogmus, Senior Member, IEEE.
6) RLS Algorithm for Acoustic Echo Cancellation, Proceedings of 2nd National Conference on Challenges & Opportunities in Information Technology , Amit Munjal, Vibha Aggarwal, Gurpal Singh
7) Performance Analysis of the Forgetting Factor RLS Algorithm, International Journal of Adaptive Control and Signal Processing, Lei Guo, Lennari Ljung, Pierre Priouret
8) Effects of Quantization and Overflow in Recursive Digital Filters, IEEE Transactions of Acoustics, Speech and Signal Processing, Claasen, Thepo ACM.
9) Simon Haykin, "Adaptive Filter Theory", Third Edition pages 365-439, pages 562587
10) Hardware Implementation of Adaptive Algorithms for Noise Cancellation,
International Journal of Information and Electronics Engineering, Vol. 2, No. 2, March 2012 Raj Kumar Thenua and S. K. Agrawal, Member, IACSIT.

11) Herbert M Aumann , Alan J Fenn and Frank G Willwerth, "Phased Array Antenna Calibration and Pattern Prediction using Mutual Coupling Measurements" IEEE Transaction on Antennas and Propagation, vol 31. no 7, July 1989.

BIO-DATA OF AUTHORS

C. ammurthyreceived his in Electronics and ation from S. J. t Polytechnic, Bangalore, Karnataka, in 1996. He received his B. Tech Degree in Electronics and Communication from U.B.D.T.College of Engineering, Kuvempu University, Davanagere, in 2001. He received his M. Tech degree in VLSI and Embedded Systems from UTL, VTU, Bangalore, in 2008. He started his professional career as a Lecturer and is working in DRDO Bangalore since 2005 pursuing PhD from Dr. Ambedkar Institute of Technology, Bengaluru. His areas of interest are RF, Microwave and FPGA based digital sub-systems. His interests include VLSI Systems, Programmable Controllers, Adaptive Beam Forming Techniques and Embedded systems.

Dr.G.V. Jayaramaih is an Professor in the Department of Electrical and Electronics Engineering of the Dr. Ambedkar Institute of Technology, Bengaluru. He works in the area of Power Electronics, Interfacing of Power electronics with Non-renewable energy. He graduated with a Ph.D. from Indian Institute of Technology, Bombay in Apr.2008.He obtained his B.E degree in Electrical Engineering from the Siddaganga Institute of Technology, Tumkur, in 1990.