DESIGN AND OPTIMIZATION OF LMS ADAPTIVE FILTER

Nishi Singh Patel[1], Sunil Kumar[2], Gagan Minocha[3]

[1,2,3] *Department of Electronics and Communication Engineering*

[1,2,3] *AMITY School of Engineering and Technology, AMITY UNIVERSITY, Noida, U.P. (India)*

ABSTRACT

This paper presents our experience in implementing the Adaptive filter using Least Mean Square algorithm on MATLAB. Adaptive filters track the dynamic nature of a system and allow to eliminate time varying signals. If the signal statistics are unknown then adaptive filtering algorithms can be implemented to estimate the signals statistics iteratively. There are many adaptive algorithms but the most popularly used is the Least Mean Square (LMS) algorithm. These filters minimize the difference between the output signal and the desired signal by altering their filter coefficients. Adaptive filters are commonly used in a wide range of applications such as Echo cancellation, Noise cancellation, Prediction, Adaptive interference cancelling, system modeling or system identification, radar signal processing, equalizations of communication channels, biomedical signal enhancements, navigational systems, digital communication receiver, adaptive antenna systems beam forming and many more.

Key words: Adaptive, LMS.

I. INTRODUCTION

The design of a wiener filter requires a prioi information about the statistics of the data to be processed. The filter is optimum only when the statistical characteristics of the input data match the priori information on which the design of the filter is based. When this information is not known completely, it may not be possible to design the wiener filter or else the design may no longer be optimum.

1.1 Why is Adaptive Filtering important?

A more efficient method is to use an adaptive filter. By such a device we mean one that is self-designing in that the adaptive filter relies for its operation on recursive algorithm, which makes it possible for the filter to perform satisfactorily in an environment where complete knowledge of the relevant signal characteristics is not available. The algorithm starts from some predetermined set of initial conditions, representing whatever we know about the environment. Yet in a stationary environment, we find that after successive iterations of the algorithm it converges to the optimum wiener solution in some statistical sense. In a non-stationary environment, the algorithm offers a tracking ability, in that it can track time variations in the statistics of the input data, provided that the variations are sufficiently slow.

An adaptive filter is said to be linear if the estimate of a quantity of interest is computed adaptively (at the output of the filter) as a linear combination of the available set of observations applied to the filter input. Otherwise, the adaptive filter is said to be non-linear. The choice of one algorithm over another is determined by following factors-

- **Rate of convergence-** It is defined as the number of iterations required for the algorithm, in response to stationary inputs, to converge "close enough" to the optimum wiener solution in the mean-square sense. A fast rate of convergence allows the algorithm to adapt rapidly to a stationary environment of unknown statistics.
- **Tracking-** When an adaptive filtering algorithm operates in a non-stationary environment, the algorithm is required to track statistical variations in the environment. The tracking performance of the algorithm is influenced by two contradictory features such as rate of convergence and steady-state fluctuation due to algorithm noise.
- **Robustness-** For an adaptive filter to be robust, small disturbances can only result in small estimation errors. The disturbances may arise from a variety of factors, internal or external to the filter.
- **Computational requirements-** Here the issues of concern include the number of operations (i.e. multiplications, divisions and additions /subtractions) required to make one complete iteration of the algorithm, the size of memory locations required to store the data and program and the investment required to program the algorithm on a computer.
- **Numerical properties-** When an algorithm is implemented numerically, inaccuracies are produced due to quantization errors. The quantization errors are due to analog-to-digital conversion of the input data and digital representation of internal calculations. In particular, there are two basic issues of concern: numerical stability and numerical accuracy. Numerical stability is an inherent characteristic of an adaptive filtering algorithm. Numerical accuracy is determined by the number of bits used in the numerical representation of data samples and filter coefficients. An adaptive filtering algorithm requires is said to be numerically robust when it is insensitive to variations in the word-length used in its digital representation.

1.2 Basic Operation

An adaptive filter is a filter that self-adjusts its transfer function according to an optimization algorithm driven by an error signal. Because of the complexity of the optimization algorithms, most adaptive filters are digital filters. The adaptive filter uses feedback in the form of an error signal to refine its transfer function to match the changing parameters.

The basic operation involves two processes:

- A filtering process, which produces an output signal in response to a given input signal.
- An adaptation process, which aims to adjust the filter parameters (filter transfer function) to the (possibly time-varying) environment.
- Here square value of the error signal is used as the optimization criterion.

II. ADAPTIVE FILTER

For the designing of the adaptive filter, Least Mean Square (LMS) algorithm has been selected due to its stability and best performance. LMS algorithm is simple to implement because it does not require matrix inversion. LMS algorithm was developed by Windrow and Hoff in 1959. The algorithm uses a gradient descent to estimate a time varying signal. By taking steps in the direction negative of the gradient, this method finds a minimum (if exists). This is done by adjusting the filter coefficients so as to minimize the error. LMS algorithm is defined by following equations-

The filter output equation-

$$y(n)=\Sigma_{k=0}^{M-1} w_k^*(n).u(n-k) \quad \ldots\ldots(1)$$

The error signal equation-

$$e(n)=d(n)-y(n) \quad \ldots\ldots(2)$$

The weight-updation equation-

$$w_k(n+1) = w_k(n)+\mu^*u(n)^*e(n) \quad \ldots\ldots(3)$$

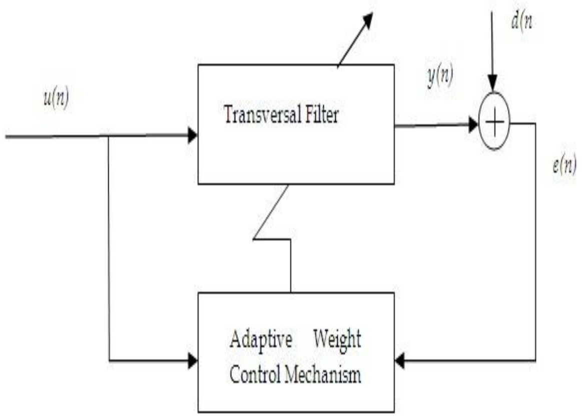


Fig.1: Adaptive Filter System

2.1 Selection of step-size

The basic idea behind LMS filter is to approach the optimum filter weights, by updating the filter weights in a manner to converge to the optimum filter weight. The algorithm starts by assuming a small weights (zero in most cases), and at each step, by finding the gradient of the mean square error, the weights are updated. That is, if the mean square error (MSE) gradient is positive, the error would keep increasing positively, if the same weight is used for further iterations, which means we need to reduce the weights. In the same way, if the gradient is negative, we need to increase the weights.

If the LMS algorithm does not use the exact values of the expectations, the weights would never reach the optimal weights in the absolute sense, but a convergence is possible in mean. That is even-though, the weights may change by small amounts, it changes about the optimal weights. However, if the variance, with which the weights change, is large, convergence in mean would be misleading. This problem may occur, if the value of step-size μ is not chosen properly.

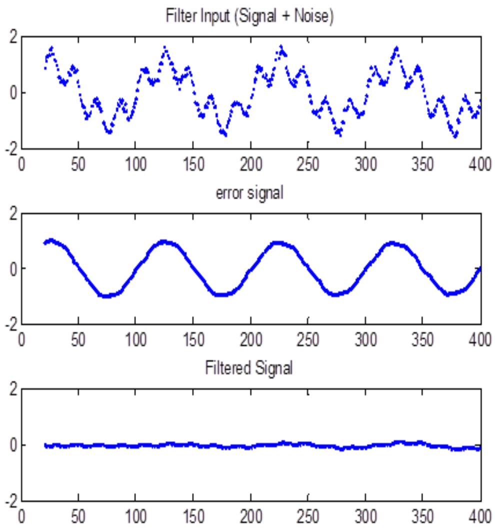
If μ is chosen to be large, the amount with which the weights change depends heavily on the gradient estimate, and so the weights may change by a large value so that gradient which was negative at the first instant may now become positive. And at the second instant, the weight may change in the opposite direction by a large amount because of the negative gradient and would thus keep oscillating with a large variance about the optimal weights. On the other hand, if μ is chosen to be too small, time to converge to the optimal weights will be too large.

Thus, an upper bound on μ is needed which is given as $0 < \mu < (2/\lambda_{max})$ where λ_{max} is the greatest eigenvalue of the autocorrelation matrix $R=E\{x(n).x(n)\}$. If this condition is not fulfilled, the algorithm becomes unstable. Maximum convergence speed is achieved when $\mu=[2/(\lambda_{max} + \lambda_{min})]$ where λ_{min} is the smallest eigenvalue of R. If μ is less than or equal to this optimum, the convergence speed is determined by λmin, with a larger value producing faster convergence. This means that faster convergence can be achieved when λ_{max} is close to λ_{min}, that is the maximum achievable convergence speed depends on the eigenvalue spread of R.

Adaptive filters are successfully applied in various fields such as control, radar, sonar, seismology, communications and biomedical engineering.

III. IMPLEMENTATION RESULTS

The design has been implemented in MATLAB. We present the implementation results of a size 20 taps (M) and maximum numbers of epochs are 400 for different values of step size (μ). Here μ is a positive parameter controlling the stability and the convergence speed. The MATLAB simulation results are obtained for the different values of step size.



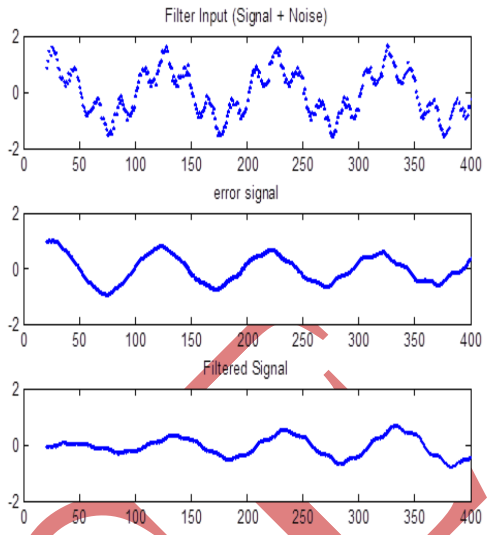
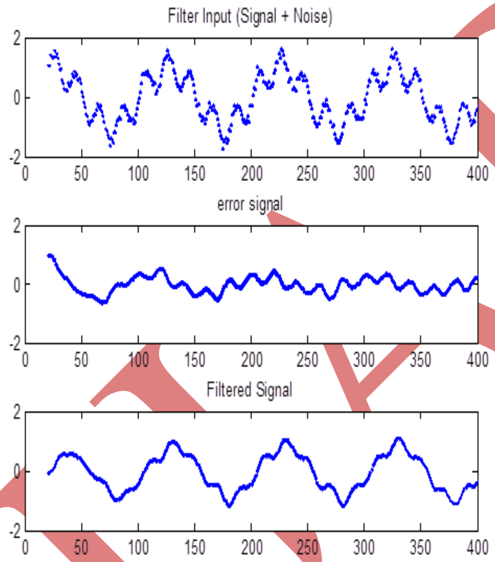


Fig.2(a):When μ (step-size parameter) = 0.00005 **Fig.2(b):When μ (step-size parameter) = 0.0005**



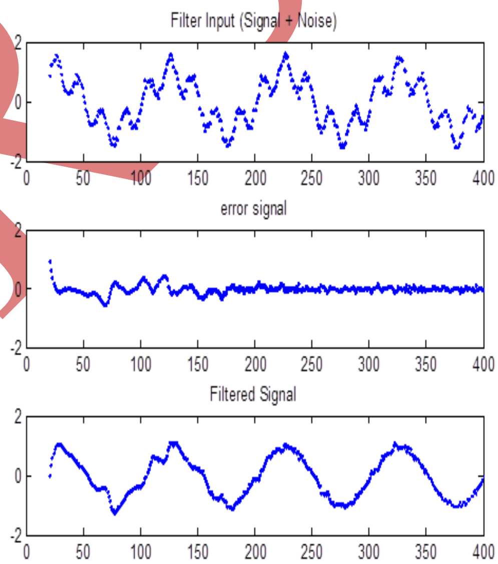


Fig.2(c): When μ (step-size parameter) =0.005 **Fig.2 (d): When μ (step-size parameter) = 0.05**

IV. CONCLUSION AND FUTURE SCOPE

Successful results are obtained with various step sizes. The increase in step size (μ) results in a significant improvement for an error to converge. The conclusion is that the Least Mean Square algorithm is the most efficient algorithm for adaptive filters. For designing purpose, the direct-form method is preferred.

LMS algorithm is famous for its stable performance. The algorithm speed can be further enhanced by following its pipelined versions FPGAs provide a perfect solution. The reprogram ability and register richness of the FPGA is best solution for implementing LMS.

REFERENCES

[1]. S. Haykin, "*Adaptive Filter Theory*". Pearson Education, 4th edition, 2002.

[2]. www.latticesemi.com

[3]. S.S.Godbole, P.M.Palsodkar, V.P.Raut, "*FPGA Implementation of Adaptive LMS filter*" Proceedings of SPIT-IEEE Colloquium and International Conference, Mumbai, India.

[4]. Mohamed Salah, Abdel-Halim Zekry, Mohammed Kamel,"*FPGA Implementation of Adaptive filter*",28th NRSE, 26-28 april,2011.

[5]. Mihir Narayan Mohanty, Biswajit Mishra, Aurobind Routray, "*FPGA Implementation of Constrained LMS algorithm*", Proceedings IEEE,2011.