

Detailed Analysis and Implementation of LMS Algorithm Using Matlab

Marthy Siva Sai Krishna¹, Murali Banoth², Manda Suhas Priyatham³

¹Electronics and Communication Engineering, GITAM University, Rudraram, Hyderabad

² Scientist 'D', DRDO Kanchanbhag

³Electronics and Communication Engineering, VASAVI College of Engineering, Ibrahim Bagh, Hyderabad

Abstract: Adaptive filters are the major components in a digital Beam forming technique which can be utilized in implementing smart antennae which can adaptively change the direction of beam towards the user. This can be achieved with a feedback mechanism used in the filters named adaptive filters and the filter transfer function is implemented with an algorithm called the Least Mean Square algorithm. This paper describes the detailed working of the Least Mean Square Algorithm and its simulation in MATLAB..

Keywords: LMS algorithm, Adaptive filters, beamforming, LMS algorithm, wiener filter

1. Introduction

I. Adaptive filter techniques have numerous applications in the field of Signal processing such as smart antennae systems, RADAR, SONAR etc. If any such application is considered and looked into, one would find that the antenna radiates an omni-directional beam whose power is radiated in all the directions. Now, any user located in a particular direction to a reference point on the antenna receives this Omni-directional radiation and the rest of the power that is radiated goes in vain such as in the places A, B and C as shown in the figure 1. In order to save energy and focus all the energy into a particular direction such that only the user gets all the power undivided, and optimal power and thus we need an antenna that would provide a constructive interference pattern to the user and a destructive interference pattern to the interferer or any other user other than the desired user.

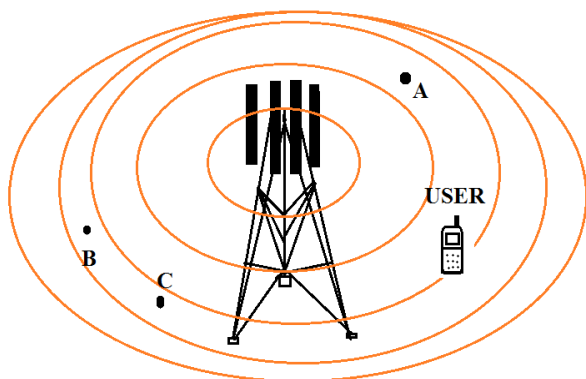


Figure 1: Antenna radiating in all directions

So, in order to achieve this particular beam to a particular user, we need an antenna array that would provide constructive interference to the desired user and destructive interference to the interferer such that all the power is concentrated towards the user as shown in figure 2.

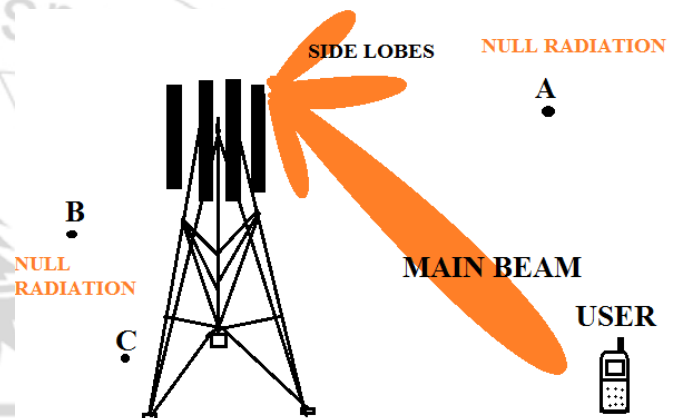


Figure 2: Main beam towards the desired user and null towards other sides

This constructive and destructive interference in the desired directions can be achieved by maintaining proper combination of effective filter weights to the signals, different time delays, phase delays and distances between different antennas in the antenna array system. All of this set and done, still there is an issue with the user that is not stationary in position and thus, the direction of beam should vary continuously in such a way that constant constructive and destructive interference pattern of the signals is generated in the direction of user. The weights assigned in the filter equation should also constantly vary according to the user's movement. The position of the user can be accurately determined by using direct line of sight information from the user as an acknowledgment signal. A solution to perform such an operation is needed and has been developed by Dr. Bernard Widrow of Stanford University, in the form of an algorithm called the Least Mean Square (LMS) Algorithm which is famous for smart antennas.

2. Description of LMS Algorithm

This LMS algorithm can be described in a stepwise procedure as follows:

First, consider the random sequence of inputs $x(n)$ which will be given to an FIR filter whose output is $y(n)$. Let us say that $d(n)$ is the reference/target signal and there always exists a difference between $y(n)$ and $d(n)$ which is taken as an error $e(n)$. This error signal $e(n)$ is given to an adaptive filter which constantly adjusts the filter weights such that the output of the FIR filter produces results close to $d(n)$ as desired.

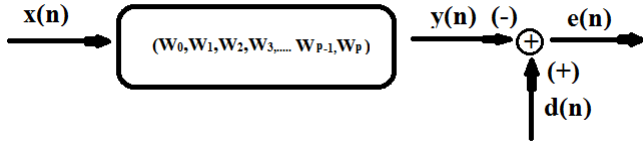


Figure 3: Wiener Filter

Set your page as A4, width 210, height 297 and margins as follows [3]:

Input sequence is $x(n)$

$$x^T(n) = [x(0) \ x(1) \ x(2) \ \dots \ x(N)]$$

Output sequence is $y(n)$

$$y^T(n) = [y(0) \ y(1) \ y(2) \ \dots \ y(N)]$$

and

$$y(n) = w_0x_0 + w_1x_1 + w_2x_2 \ \dots + w_px_N \\ = w^T x(n)$$

Target sequence is $d(n)$

$$d^T(n) = [d(0) \ d(1) \ d(2) \ \dots \ d(N)]$$

Error sequence is $e(n)$

$$e^T(n) = [e(0) \ e(1) \ e(2) \ \dots \ e(N)]$$

$x(n), y(n), d(n), e(n)$ are zero mean random processes. Here $W_0, W_1, W_2, \dots, W_N$ are filter weights/coefficients.

The equation following the above filter is $e(n) = d(n) - y(n)$

Let R_{xx} be a correlation matrix where

$$R_{xx} = E[x(n) x^H(n)]$$

R_{xx} : Hermitian

Choose $k \neq 0$:

$$\text{Where } k^T = [k_0 \ k_1 \ k_2 \ \dots \ k_N]$$

Consider

$$k^H R_{xx} k = k^H E[x(n) x^H(n)] k \\ = E[k^H x(n) \cdot x^H(n) k]$$

Here k^H is row vector, $x(n)$ is column vector therefore

The product of row and column vector is a scalar.

i.e. $k^H x(n)$ and $x^H(n) k$ are scalars.

$$k^H x(n) = a \implies x^H(n) k = a^H$$

then

$$k^H R_{xx} k = E[|a|^2]$$

Where $E[|a|^2] > 0$

Consider

$$R = E[x(n) x^T(n)]$$

$$P(n) = E[x(n) d(n)]$$

Consider

$E[e^2(n)]$ which is mean square error of the output.

$$E[e^2(n)] = E[(d(n) - w^T x(n))(d(n) - w^T x(n))^T]$$

$$= E[d^2(n)] - 2E[w^T x(n) d(n)] + E[w^T x(n) x^T(n) w]$$

$$E[e^2(n)] = E[d^2(n)] - 2w^T p + w^T R w \quad \text{----- equation(1)}$$

Here $E[d^2(n)]$ is scalar.

As the optimum error value for a filter is required to be 0.

Derivation of equation 1 on both sides for minimum value

$$d(E[e^2(n)]/dw) = (d(E[d^2(n)]/dw) - (d(w^T p)/dw) + 2(d(w^T R w)/dw))$$

$$0 = 0 - 2p + 2Rw \quad \text{(from equation (1))}$$

$$W = R^{-1} p \quad \text{where}$$

$$(w = w_{\text{optimum}}) \quad \text{----- equation (2)}$$

This is the filter equation of Wiener filter.

In order to get minimal average error power the filter weights has to be satisfying the wiener filter equation.

The mean square error equation is generally a quadratic equation in this case. i.e. the graph would be in parabolic form as shown in the figure 4.

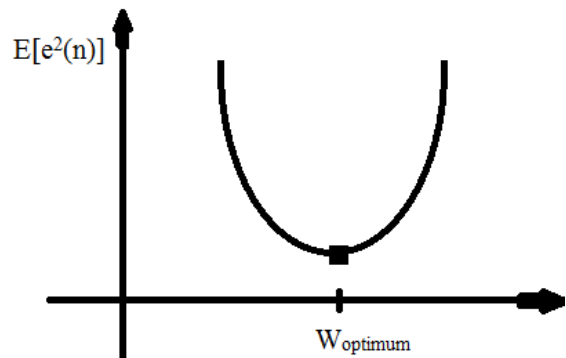


Figure 5: Graph of mean square error quadratic equation

It is not possible to converge the filter weights to the optimum value in real time, but we can get the filter weights close to it by using steepest descent procedure. The adaptive filter uses the steepest descent procedure to converge the filter weights to the optimum values. The system undergoes much iteration to minimize the error.

The steepest descent procedure can be described as follows: Consider $W(i)$ be the filter weights of the FIR filter at i^{th} iteration.

CASE 1: $W(i)$ is right to the W_{optimum}

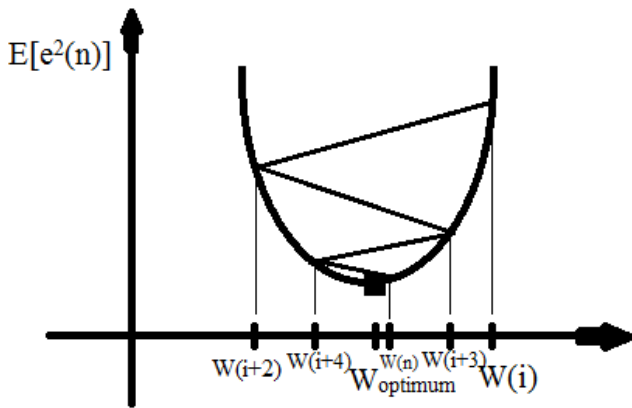


Figure 6: Converging the filter weight value to optimum value

If the value of the $w(i)$ is right it should get subtracted by a value such that the $w(i+1)$ is close to the $W_{optimum}$ as compared with $w(i)$ and the $w(i+2)$ value should be further close to $W_{optimum}$ as compared to $w(i+1)$ and the procedure follows till it converges the $W_{optimum}$. If the tangent passing to the point on the parabola has narrow slope small value has to be selected either to add or subtract or else large values should be taken. So that the convergence will takes place within less number of iteration.

CASE 2: $W(i)$ is Left to the $W_{optimum}$

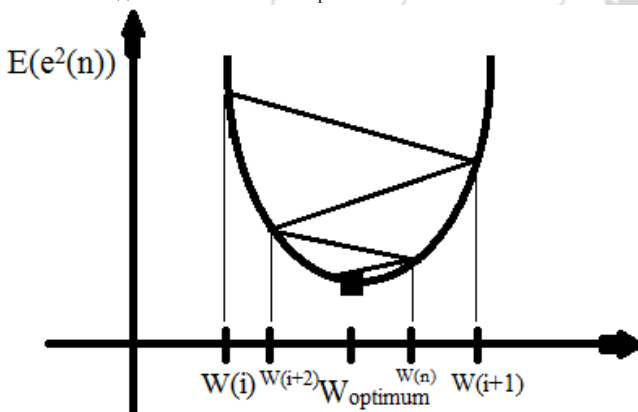


Figure 7: Converging the filter weight value to optimum value

If the value of the $w(i)$ is Left it should get added by a value such that the $w(i+1)$ is close to the $W_{optimum}$ as compared with $w(i)$ and the $w(i+2)$ value should be further close to $W_{optimum}$ as compared to $w(i+1)$ and the procedure follows till it converges the $W_{optimum}$. If the tangent passing to the point on the parabola has narrow slope small value has to be selected either to add or subtract or else large values should be taken. So that the convergence will takes place within less number of iteration.

The value that has to be added or subtracted can be chosen by the following expression:

Consider μ be a constant

$$w(i+1) = w(i) - \frac{\mu}{2} \frac{d}{dw} (E(e^2(n))) \Big|_{w=w(i)}$$

$$w(i+1) = w(i) - (\mu/2) d(E[e^2(n)])/dw \Big|_{w=w(i)}$$

$$w(i+1) = w(i) + (\mu x(n)[d(n)-y(n)])$$

$$w(i+1) = w(i) + \mu x(n)e(n) \text{ ----- equation (3)}$$

This is the equation to determine the value that has to be either added or subtracted during the iterative process. Generally

$\lim_{i \rightarrow \infty} W(i) \rightarrow W_{optimum}$ from the above equation 3. But in real time equation 3 is an approximate. Considering equation 3 as standard the real convergence will takes place by the following expression.

$$\lim_{i \rightarrow \infty} E(w(i)) \rightarrow W_{optimum}$$

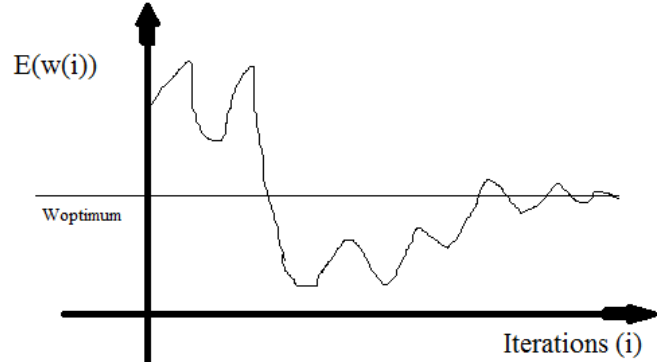


Figure 8: $E(w(i))$ converging to $W_{optimum}$ through iterative process

This is called LMS Algorithm. In order to satisfy the wiener filter equation the filter weights should be optimum. This can be achieved by using adaptive filter to the system. This LMS technique is used to implement the adaptive filter.

The system block diagram with adaptive filter will be like

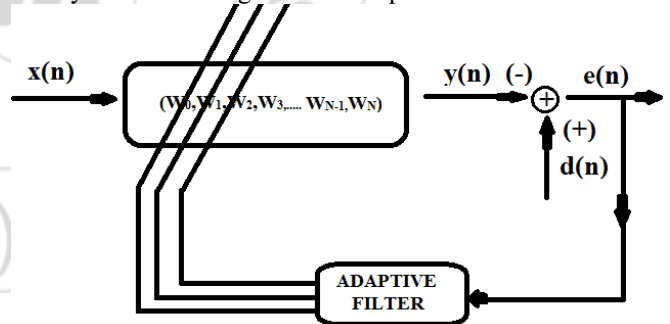


Figure 9: Block diagram of system with adaptive filter driving the system.

LMS technique is the most famous technique implemented for the adaptive filter. RLS technique is also used to implement the adaptive filter which has some advantages over LMS adaptive technique but consists of complex hardware when compared with LMS. Thus LMS is highly implemented technique.

3. Matlab Implementation of LMS

A. Rough code

- W_0 = some initial value.
- For $n=0:N$ (final value)
- $y(n) = w^T x(n)$

- $e(n) = d(n) - y(n)$
- $w(i+1) = w(i) + \mu x(n)e(n)$
- Plotting the graphs

B.Simulation results

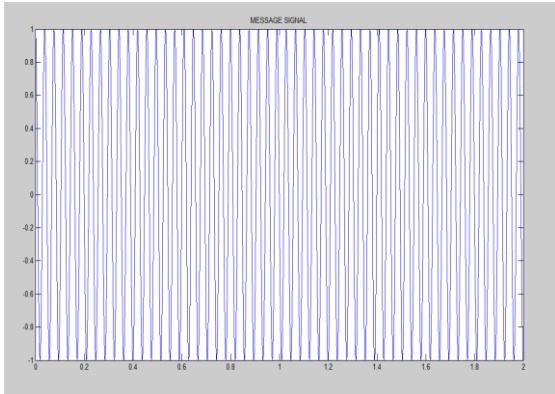


Figure 10: Input message signal to FIR filter $x(n)$

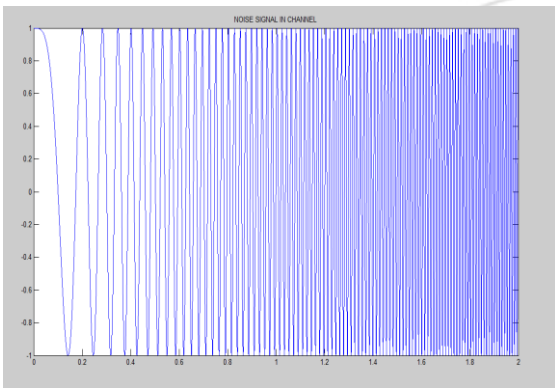


Figure 11: Noise signal

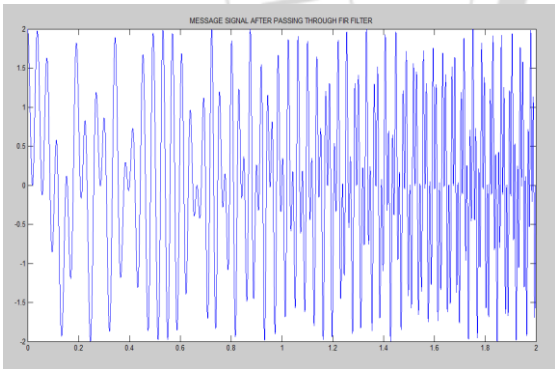


Figure 12: Signal after passing through filter (message signal + noise)

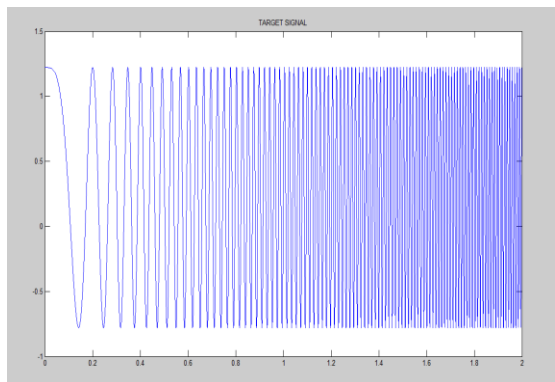


Figure 13: Target signal $d(n)$

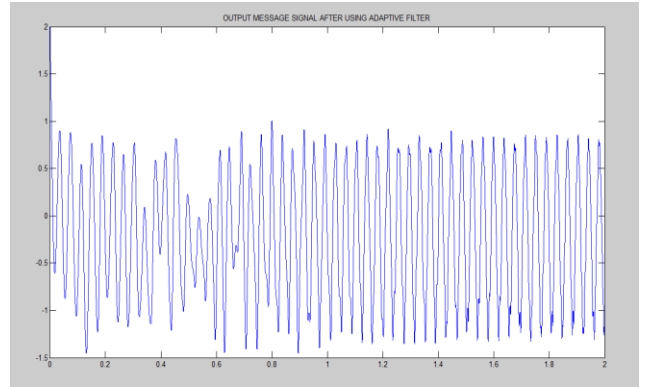


Figure 13: Output message signal after using Adaptive filter (hopefully similar to the message signal).

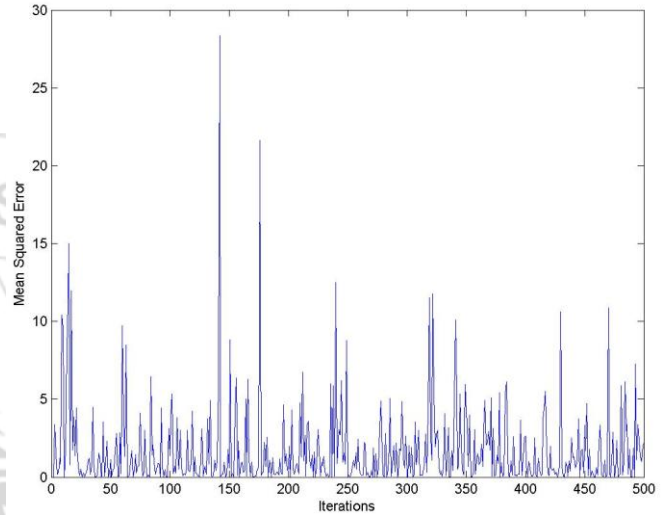


Figure 14: Mean squared error vs Iterations (That changes for every compilation)

From fig.14 we can see that the output generated signal is almost similar to the input message signal. By this way we can restore the message signal from noise and other attenuation effects. Here noise in the channel includes both the internal and external noises.

4. Conclusion

The Adaptive filters based on LMS algorithm have a wide range of applications. Adaptive beam forming techniques, beam forming, echo cancellation are some of its main applications. Implementation of this algorithm is simple as compared with other techniques in terms of hardware. Still RLS algorithm has better convergence procedure than LMS but due to its complex hardware implementation LMS is widely implemented over many Adaptive systems

References

- [1] Anupama Senapati, Kaustabh Ghatak, and Jibendu Sekhar Roy, "A Comparative Study of Adaptive Beamforming Techniques in Smart Antenna Using LMS Algorithm and Its Variants," 2015 International Conference on Computational Intelligence & Networks, IEEE transaction.
- [2] S.Arunkumar, P.Parthiban, S.Aravind Kumar, "Implementation of Least Mean Square Algorithm For

Sinusoidal and Audio Denoising Using FPGA”
IJAREEIE, Vol. 2, Issue 12, December 2013.

Author Profile



Marthy Siva Sai Krishna received the Bachelor degree in Electronics and communications engineering from GITAM University in 2016 and have an interest in Signal processing and communications.



Murali Banoth received the degree from IIT Kharagpur and presently working as Scientist ‘D’ at Defence research and development organization.



Suhas priyatham manda received the Bachelor degree in Electronics and communications Engineering from Vasavi college of engineering in 2016 and have an interest in Signal processing and communications.

