Adaptive Filter Identification
Using Genetic with LMS (GALMS) Algorithm

Asst. Lect. Emad A. Hussein
Electrical Engineering Department
Al-Mustansiriya University- Iraq-Baghdad

Abstract

Conventional non-adaptive filters that are used for extracted the information from an input signals, are normally linear and time invariant. In this case the restriction of time invariance is removed. This is done by allowing the filter to change the coefficients used in the filtering operations according to some predetermined optimization criteria. This has the important effect that the adaptive filters may be applied in areas where the exact filtering operation required may be known a priori and further, this filtering operation may be non-stationary. System modeling or system identification is one of the wide applications of the adaptive filtering that have a great importance in the fields of communication systems and signal processing .The main object of this paper is to find a best optimization algorithm that gives a minimum Mean Squared Error (MSE) between the desired and the actual signal to identify the unknown system. Many algorithms will be studied, such as the Least Mean Squared (LMS) algorithm, Adaptive Linear Neuron Network (ADALINE) and Genetic Algorithm (GA). Then we will produce a new improvement algorithm (we called it GALMS) that uses the LMS algorithm with optimized learning coefficient using genetic algorithm. Optimal weights (coefficients) will also be found to be concentrated with the actual weights.

Key words : ***Adaptive filters, identification, LMS, ADALINE network, and GAs.***

الخلاصـــــــة

المرشحات غير التكيفية التقليدية التي تستخدم في استخلاصِ المعلومات من إشارة إدخال معينة تكون عادة خطية وذات معاملات ثابتة مع الزمن. في هذه الحالةِ التقيد بالزمن يمكن التخلص منه عن طريق السَماح للمرشِحَ بتَغيير معاملاتِه اثناء عملياتِ التَرْشيح طبقاً لبعض المعايير التي تحقق الأمثلية المحدّدة مسبقاً. هذا لَهُ التأثيرُ المهمُ الذي يجعل استخدام المرشحات التكيفية في التطبيقات التي تحتاج تكيُّف ودقة في عملية الترشيح. إن تمثيل الأنظمة (system identification) أحد تطبيقات المَرْشحات التكيفية الكثيرةِ والتي لَها أهمية كبيرة في حقولِ أنظمة الاتصالاتِ ومعالجةِ الإشارة. الهدف الأساسي من هذا البحث هو إيجاد أفضل خوارزميةِ تحقق أمثلية المنظومة أي أنها تَعطي أدنى معدل مربع خطأ (MSE) بين الإشارةِ المطلوبةِ والإشارة الفعليةِ لتَمثيل الأنظمة المجهولة. العديد مِنْ الخوارزميات سَتَدرس في هذا البحث مثل خوارزمية اقل معدل مربع خطأ (LMS) ، شبكة الخلية العصبيةِ الخطيّةِ التكيفية (ADALINE) و الخوارزمية الوراثية (GA). ثمّ نعرض خوارزمية جديدة أسميناها (GALMS) والتي تستخدم أل (LMS) وال (GA) في أدائها. ستقوم هذه الخوارزمية بإيجاد احسن معامل تعلم (يعطي اقل (MSE)) عن طريق أل(GA) من ثم تطبيق هذا المعامل على أل (LMS) للحصول على أوزان (معاملات) المرشح المثالية والتي تماثل الأوزان الأصلية للمنظومة المجهولة .

1. Introduction

An adaptive filter is a system whose structure is adjustable in such a way that its behavior or performance (according to some desired criterion) improves through contact with its environment. The adaptive filter coefficients adjust themselves to achieve the desired result, such as identifying an unknown filter or canceling noise in the input signal. In Figure-1, the shaded box represents the adaptive filter, comprising the adaptive filter and the adaptive algorithm [1] [2].

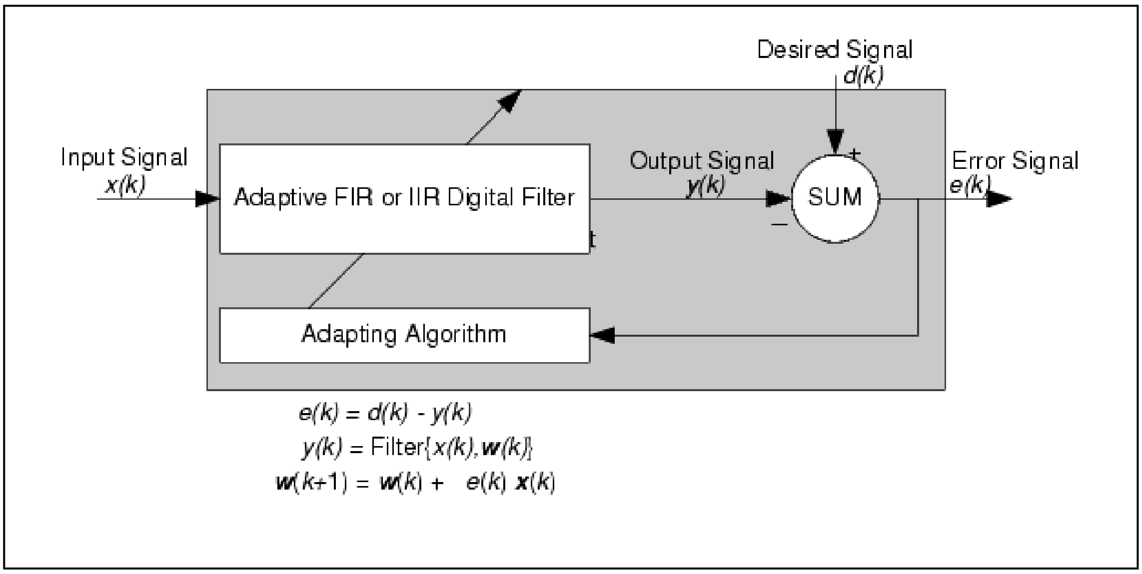

Figure-1: Block Diagram Defining General Adaptive Filter Algorithm Inputs and Outputs

An adaptive FIR or IIR filter designs itself based on the characteristics of the input signal to designing the filter does not require any other frequency response information or specification. To define the self learning process the filter uses, you select the adaptive algorithm used to reduce the error between the output signal *y(k)* and the desired signal *d(k)*. Now the output from the adaptive filter matches closely the desired signal *d(k)*. When we change the input data characteristics, sometimes called the filter environment, the filter adapts to the new environment by generating a new set of coefficients for the new data. Notice that when *e(k)* goes to zero and remains there you achieve perfect adaptation; the ideal result but not likely in the real world [2].

2. System Identification

The System Identification problem is to estimate a model of a system based on observed input-output data. Several ways to describe a system and to estimate such descriptions exist. One of these ways is the Least Mean Squares error (LMS) method.

When the Least Mean Squared error (LMS) performance criterion has achieved its minimum value through the iterations of the adapting algorithm, the adaptive filter is finished and its coefficients have converged to a solution [3].

System Identification allows us to build mathematical models of a system based on measured data. This is done essentially by adjusting parameters within a given model until its output coincides as well as possible with the measured output.

One common application is to use adaptive filters to identify an unknown system, such as the response of an unknown communications channel or the frequency response of an auditorium, to pick fairly divergent applications.

Other applications include echo cancellation and channel identification. In figure-2, the unknown system is placed in parallel with the adaptive filter.

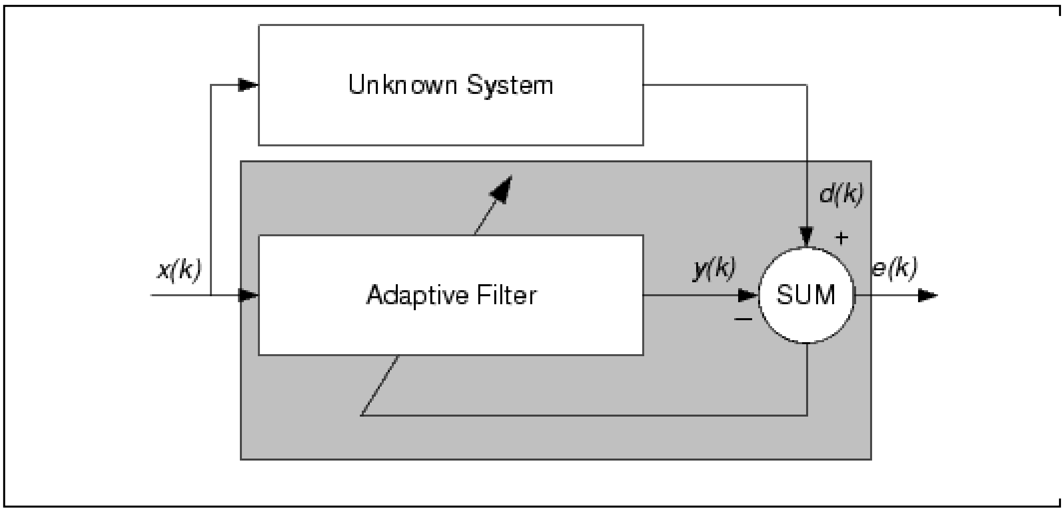

Figure-2: Using an Adaptive Filter to Identify an Unknown System

Clearly, when *e(k)* is very small, the adaptive filter response is close to the response of the unknown system. In this case the same input feeds both the adaptive filter and the unknown.

When the unknown system is a modem, the input often represents white noise, and is the sound we hear from our modem when we log in to our Internet service provider [4][5].

3. Least Mean Squares error (LMS) algorithm

The LMS Algorithm introduced by Widrow and Hoff can be applied to find the adaptive filter weights (w_i) (shown in figure-3) so that the error ($\varepsilon(k)$) be equal or converges to zero and the mean square error ($e(k)$) be minimum as possible. In this case the adaptive filter weights will be the optimal weights that give the solution of identification problem of the identified system [3].

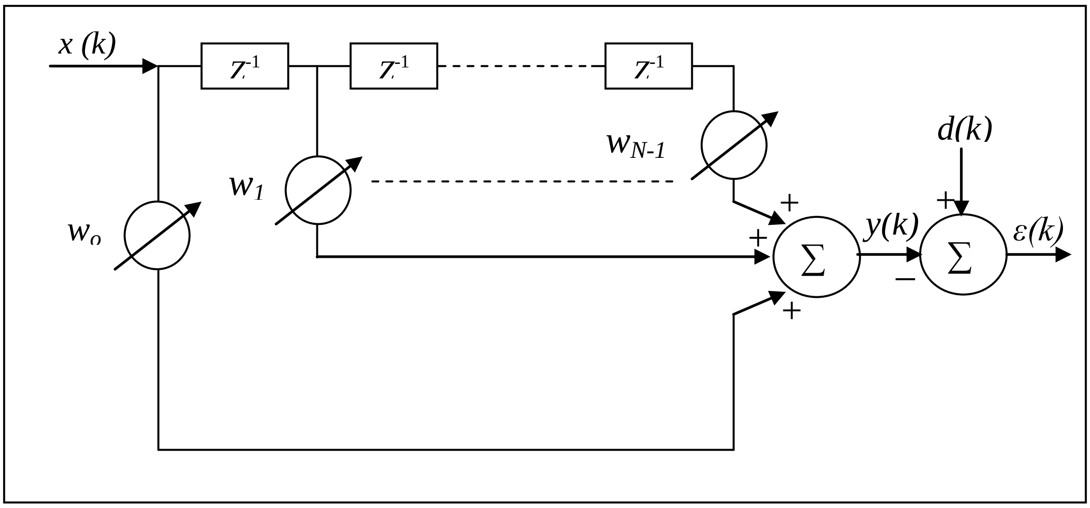

Figure-3: The adaptive linear filter implementation

The input output relationship is:

$$y(k) = \sum_{n=0}^{N-1} w_i(k)x(k-n) \qquad \ldots\ldots(1)$$

Where $y(k)$ the actual output, $w_i(k)$ the i-th filter weight, $x(k)$ the applied input for the k-th sample and N is the order of the filter.

Now the adaptation of the weights (w_i^{new}) depending on the previous weight values (w_i^{old}) according to the relation:

$$w_i^{new} = w_i^{old} + \Delta w_i \qquad : i = 1,2,.....,N-1 \qquad \ldots\ldots(2)$$

Now the following derivation is to find an expression for the weight change (Δw_i). Hence, the corresponding error term $\varepsilon(k)$ is:

$$\varepsilon(k) = d(k) - y(k) \qquad \ldots\ldots(3)$$

where *d(k)* and *y(k)* are the desired and actual output for the *k-th* sample respectively. The sum squared error, or expectation value of the error (*e(k)*), is defined by:

$$e(k) = \sum_{k=1}^{R} \varepsilon^2(k) \qquad \ldots\ldots(4)$$

Where *R* is the number of input patterns in the training set. The error can be reduced by adjusting the weights in proportion to the negative of the gradient, the direction of must rapid decrease in the error function, *e(k)* with respect to each weight change, Thus, we want an expression for the weight change Δw_i proportional to the negative gradient of the error, that is

$$\Delta w_i = -\eta \frac{\partial e(k)}{\partial w_i} = -\eta \sum_{k=1}^{R} \frac{\partial \varepsilon^2(k)}{\partial w_i} \qquad \ldots\ldots (5)$$

where η is a positive constant. Taking partial derivatives of each term in the summation gives the following,

$$\frac{\partial \varepsilon^2(k)}{\partial w_i} = \frac{\partial}{\partial w_i}(d(k) \quad \sum_{i=1}^{N-1} w_i\, x_i(k))^2 \qquad \ldots\ldots (6)$$

$$= 2(d(k) \quad y(k))\, x_i(k)$$

So, we have the simple LMS rule given by:

$$\Delta w_i = 2\, \eta\, (d\,(k)\text{-}y\,(k))\, x_i\,(k)$$

$$= 2\, \eta\, \varepsilon\,(k)\, xi\,(k)$$

$$= \mu\, \varepsilon\,(k)\, x_i\,(k) \qquad \ldots\ldots (7)$$

where μ is the step size or learning coefficient that has a significant effect on training. If μ is too large, convergence will never take place, no meter how long is the training period. Finally, we conclude that our used formula that adapts the weights of the adaptive filter (with known μ) is:

$$w_i^{new} = w_i^{old} + \mu\, \varepsilon(k)\, x_i(k) \qquad \ldots\ldots (8)$$

4. Adaptive Linear Neuron Element (ADALINE) Network

The ADALINE network discussed in this section is similar to the perceptron, but their transfer function is linear rather than hard limiting [6].

This allows their outputs to take on any value, whereas the perceptron output is limited to either 0 or 1. Both the ADALINE and the perceptron can only solve linearly separable problems. However, here the LMS algorithm will be used as a learning rule which is much more powerful than the perceptron learning rule. The LMS or Widrow-Hoff learning rule minimizes the mean square error and thus moves the decision boundaries as far as it can from the training patterns. The linear network produces outputs of corresponding target vectors that when presented with a set of given input vectors. The difference between an output vector and its target vector is the error [7]. The complete ADALINE consists of the Adaptive Linear Combiner (ALC) and a pure-linear transfer function (as shown in Figure-4)). ALC is adaptive in the sense that their exist a well-defined procedure for modifying the weights in order to allow the device to give the correct output value for the given input. What output value is correct depends on the particular processing function being performed by the device. The ADALINE (or the ALC) is linear because the output is a simple linear function of the input values. The ADALINE could also be said to be a linear Element, avoiding the neuron issue altogether [8].

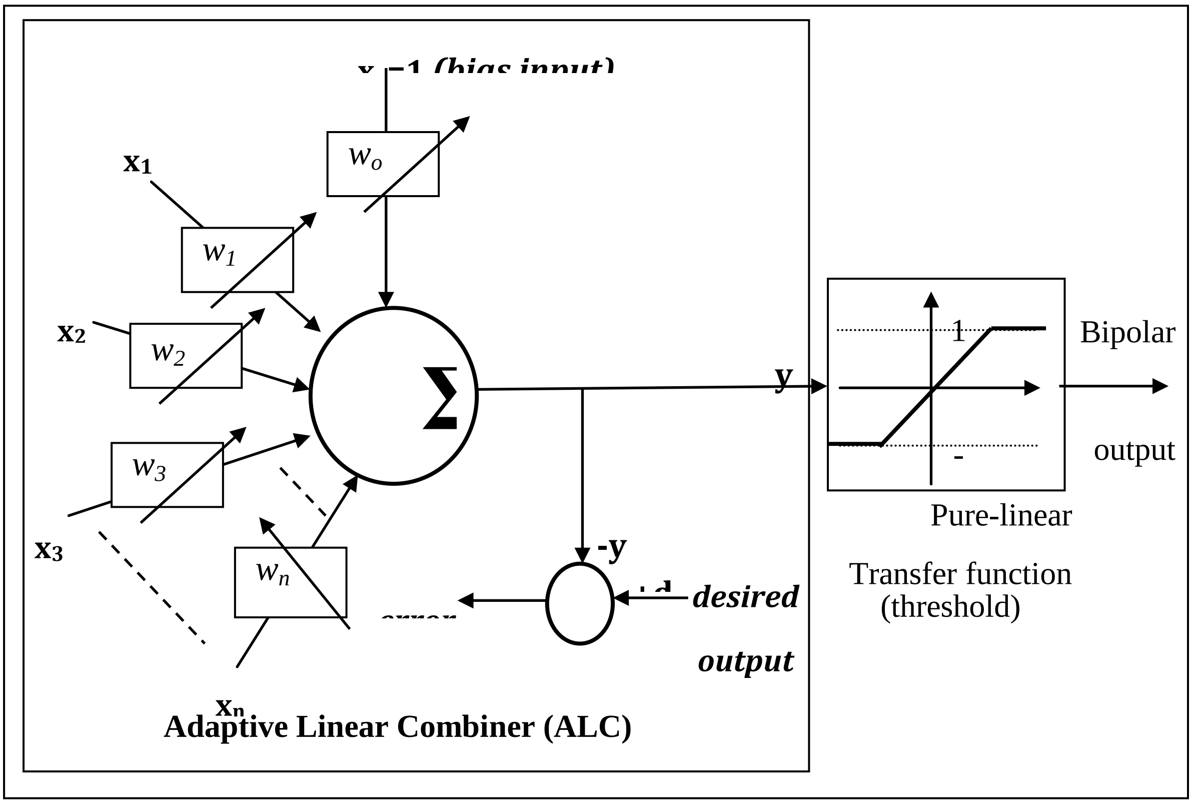

Figure-4: ADALINE network

The ALC performs a sum-of-products calculation using the input and weight vectors, and applies an output function to get a single output value. Using the notation in Figure-4, the network consists of n inputs and one output equal to:

$$y = \sum_{i=0}^{n} w_i \, x_i \qquad \ldots\ldots(9)$$

or, in vector notation,

$$y = w^T x \qquad \ldots\ldots(10)$$

where w^T is the transpose of the weight vector on the connection between the i-th input pattern element and the single output neuron. The linear output from the neuron is sent to a threshold logic unit to obtain the final bipolar output. The output function in this case is the identity function, as is activation function. The use of the identity function as both output and activation functions means that the output is the same as the activation, which is the same as the net input to the unit.

As we illustrate previously, ADALINE network uses the LMS algorithm or Widrow-Hoff learning algorithm based on approximate steepest descent procedure. So the adaptation of the weights and bias will depend on Equ.8 as shown

$$w_i(j+1) = w_i(j) + \mu \, \varepsilon(j) \, x_i(j) \qquad \ldots\ldots(11)$$

$$b_i(j+1) = b_i(j) + \mu \, \varepsilon(j) \qquad \ldots\ldots(12)$$

Where $w_i(j)$ the *i-th* weight for the j-th iteration and $b_i(j)$ the *i-th* bias weight. The other variables are defined previously.

5. Genetic Algorithm (GA)

A genetic algorithm is an optimization method that manipulates a string of numbers in a manner similar to how chromosomes are changed in biological evolution.

An initial population made up of strings of numbers is chosen at random or is specified by the user. Each string of numbers is called a "chromosome" or an "individual," and each number slot is called a "gene." A set of chromosomes forms a population.

Each chromosome represents a given number of traits which are the actual parameters that are being varied to optimize the "fitness function". The fitness function is a performance index that we seek to maximize [9]. The operation of the GA proceeds in steps. Beginning with the initial population, "selection" is used to choose which chromosomes should survive to form a "mating pool." Chromosomes are chosen based on how fit they are (as computed by the fitness function) relative to the other members of the population.

More fit individuals end up with more copies of themselves in the mating pool so that they will more significantly effect the formation of the next generation. Next, several operations are taken on the mating pool. First, "crossover" (which represents mating, the exchange of genetic material) occurs between parents. To perform crossover, a random spot is picked in the chromosome, and the genes after this spot are switched with the corresponding genes of the other parent. Following this, "mutation" occurs. This is where some genes are randomly changed to other values. After the crossover and mutation operations occur, the resulting strings form the next generation and the process is repeated. A termination criterion is used to specify when the GA should end (e.g., the maximum number of generations or until the fitness stops increasing) [10]. In adaptive filtering, the set of filter coefficients is encoded into a chromosome as a list of real numbers and the genetic operators are used as a searching tool for the gradient method to find a faster short-cut route in learning the optimal solution during the adaptation phase [11].

6. The New GALMS Algorithm

Whenever the LMS algorithm gets stuck at a local minimum or the convergence rate is relatively slow depending on the value of the learning coefficient (μ), we start the genetic search by finding the optimal learning coefficient with the best fitness i.e. the smallest (MSE) as the survived chromosome. In other words, the learning coefficient will be the GA chromosomes in which a fitness value is assigned to each chromosome. The fitness value for the j-th chromosome is inversely proportional to the mean squared error ε_j^{-2}, where (ε_j^{-2}) is given by

$$\varepsilon_j^2 = \frac{1}{R}\sum_{k=1}^{R}[d_j(k)\text{-}y_j(k)]^2 \qquad \text{...... (13)}$$

Where R is the window size over which the errors will be accumulated; $d(k)$ is the desired output; $y(k)$ is the estimated output associated with the j-th estimated chromosome. The survived chromosome will be the optimized learning coefficient that gets us a minimum MSE (or optimal solution).

So we can summarize our new algorithm in the following steps:

Step-1: Apply an input signal $x(k)$ to the unknown system and evaluate the desired output ($d(k)$).

Step-2: Initialize the filter weight (make it all zeros) and compute the actual output (y(k)) according to Equ.1.

Step-3: Put a random number for the learning rate μ (ranged 0 to 1), this will be the GA chromosome.

Step-4: Find the corresponding error term ε(k) according to Equ.3.
Step-5: Compute the new weights according to LMS algorithm (Equ. (8)).
Step-6: Evaluate the fitness function for the GA according to the inverse of Equ.13.
Step-7: If the population size of the GA does not end, go to step-3.
Step-8: Begun the GA search for minimum MSE according to the fitness function.
Step-9: Print the chromosome (learning rate μ) that gives minimum MSE and print the filter weights.

7. Case Study

In this section we will implement an FIR-Low pass filter that has the frequency and impulse response shown in Figure-5 & -6 respectively. This will be the unknown system that we discussed it previously in section (2) an illustrated in Figure-2.This unknown system may be any type of filters or any communication or control system.

We will use three methods to identify this system; Adaptive filter method with LMS algorithm, ADALINE network, and GA only. Then we will compare the results with our new approach (that we called it GALMS algorithm) uses the genetic algorithm in the design process. A normally distributed random input will be used as an input signal and the filter output will be the desired output.

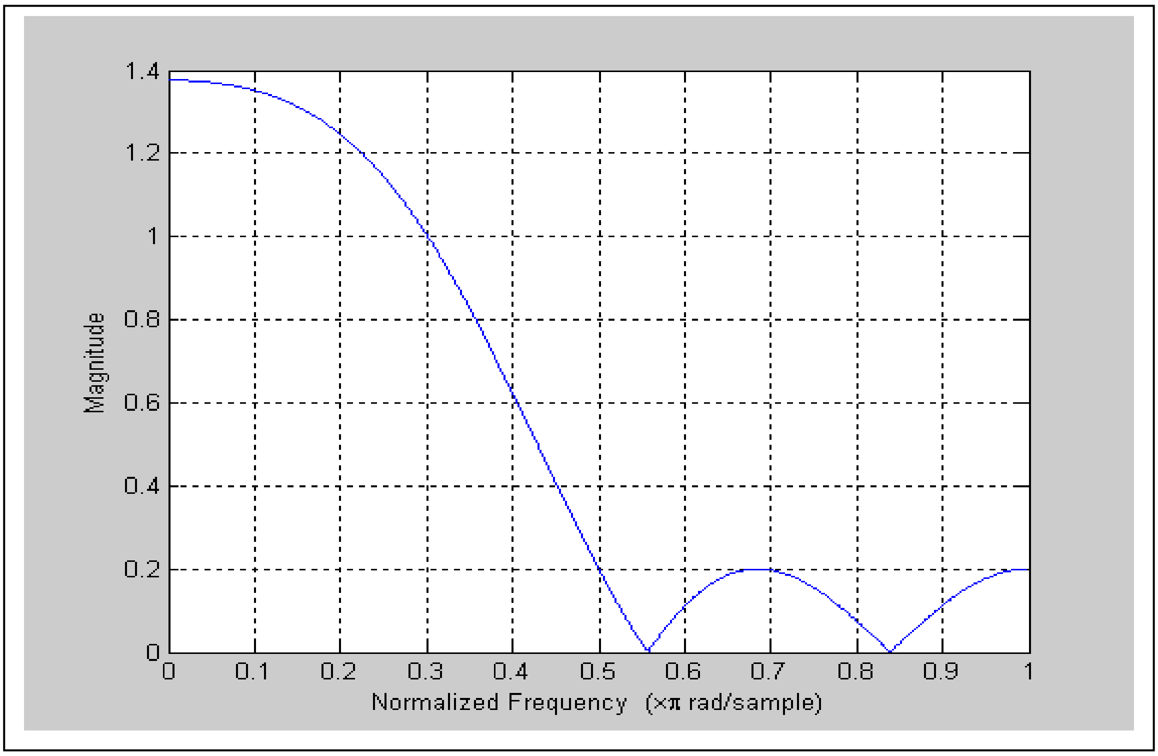

Figure-5: Frequency response for the identified system (FIR- Low pass filter)

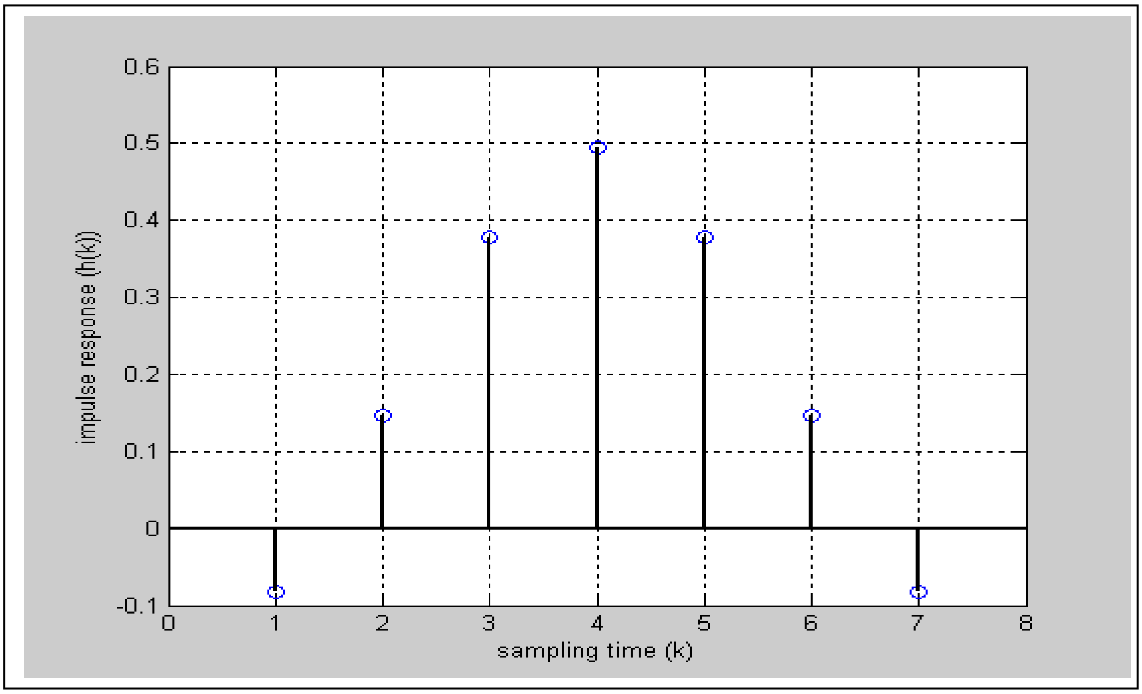

Figure-6: Impulse response for the identified system (FIR- Low pass filter)

8. Simulation Results

The unknown system with the above characteristics will be identified using our algorithms in order to find which of these algorithms suitable for getting minimum MSE in a reduced number of iterations. Note that the following results are obtained after applying some parameters to each algorithm using MATLAB programming language simulated in a computer having a 2.7 GHz pentum-4 processor.

8-1. Least Mean Squared error (LMS) algorithm

The learning curves (error versus number of iterations) for three different values of step size, μ, (equal to 0.2, 0.5 & 0.8) are shown in figures (-7, -8 & -9) respectively. These curves obtained when we apply these parameters in Equ.1 to Equ.8

8-2. Adaptive Linear Neuron Network (ADALINE)

Figure (10) shows the ADALINE learning curve for 500 iteration based on Equ.(9,10,11 and 12).

8-3. Genetic Algorithm (GA) only

The filter weights will be adapted by the use of genetic algorithm with the bellow parameters :

Population size = 20.

Number of bits in each chromosome = 10 bits.

Probability of crossover = 0.9.

Probability of mutation = 0.05.

Maximum number of generations = 100.

Figure-11 shows a relation between the maximum and average fitness vs. generations and Figure-12 shows the learning curve. Note that these parameters simulated in a MATLAB program depending on reference [10].

8-4. GA and LMS (GALMS) Algorithm

The learning coefficient will be adapted using GA with the following specifications:

Population size = 40.

Number of bits in each chromosome = 5 bits.

Probability of crossover = 0.9.

Probability of mutation = 0.05.

Maximum number of generations = 20.

After applying these parameters the results are cleared in Figure-13 (the relation between the maximum and average fitness vs. generations) and Figure-14 (the learning curve). The optimal learning coefficient founds in this method equal to two.

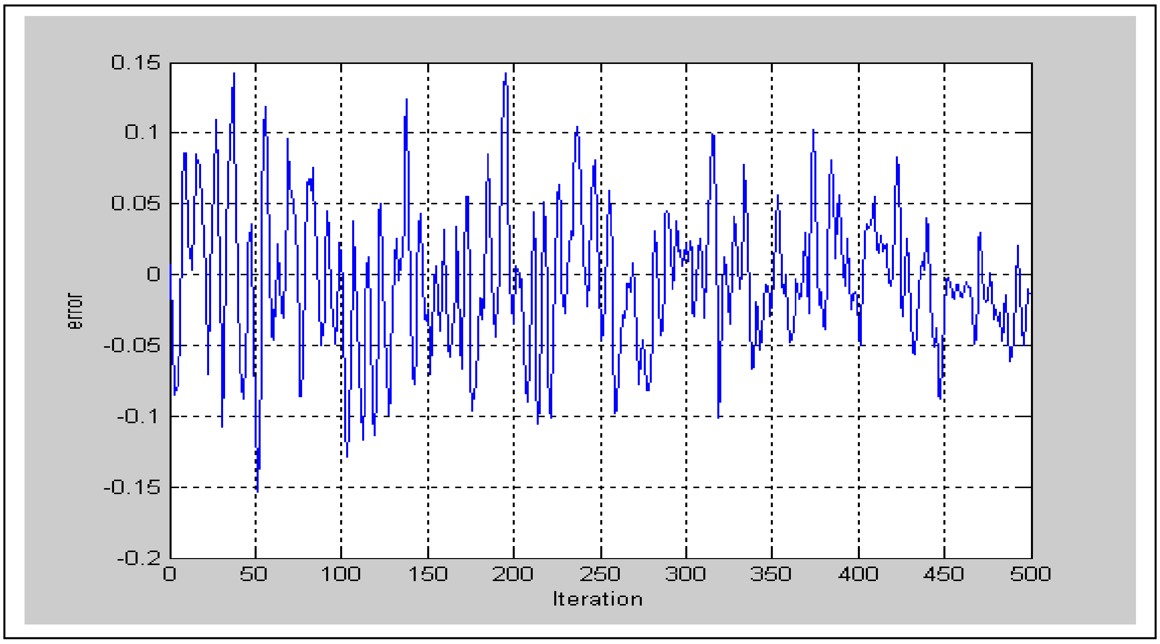

Figure-7: The learning curve for the LMS algorithm (μ = 0.2).

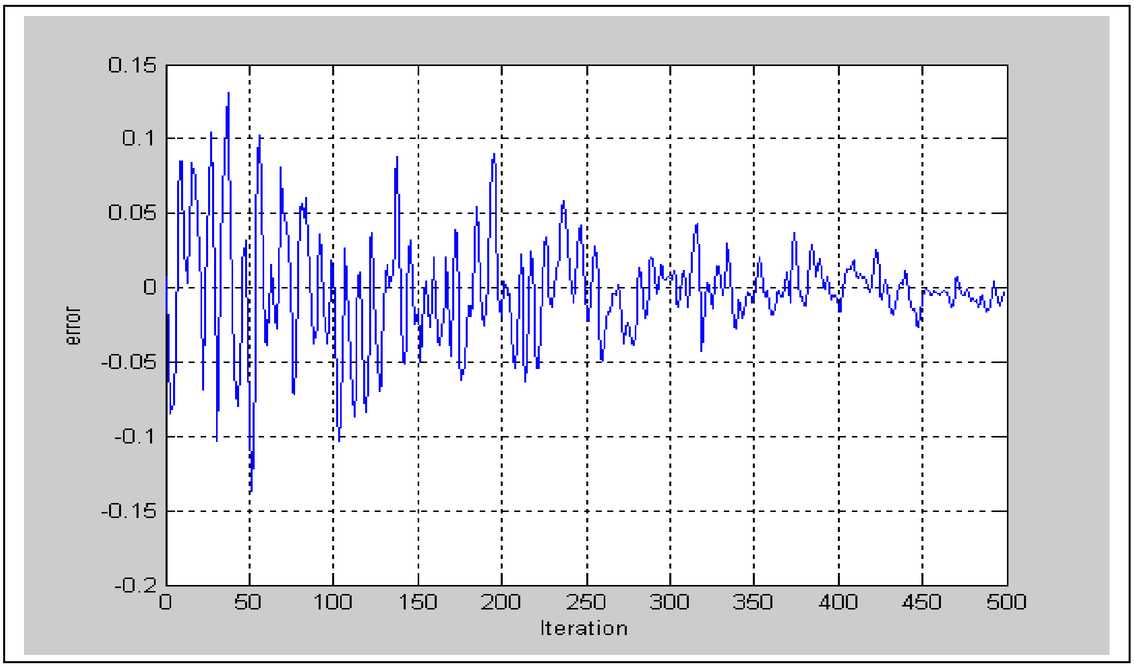

Figure-8: The learning curve for the LMS algorithm (μ = 0.5).

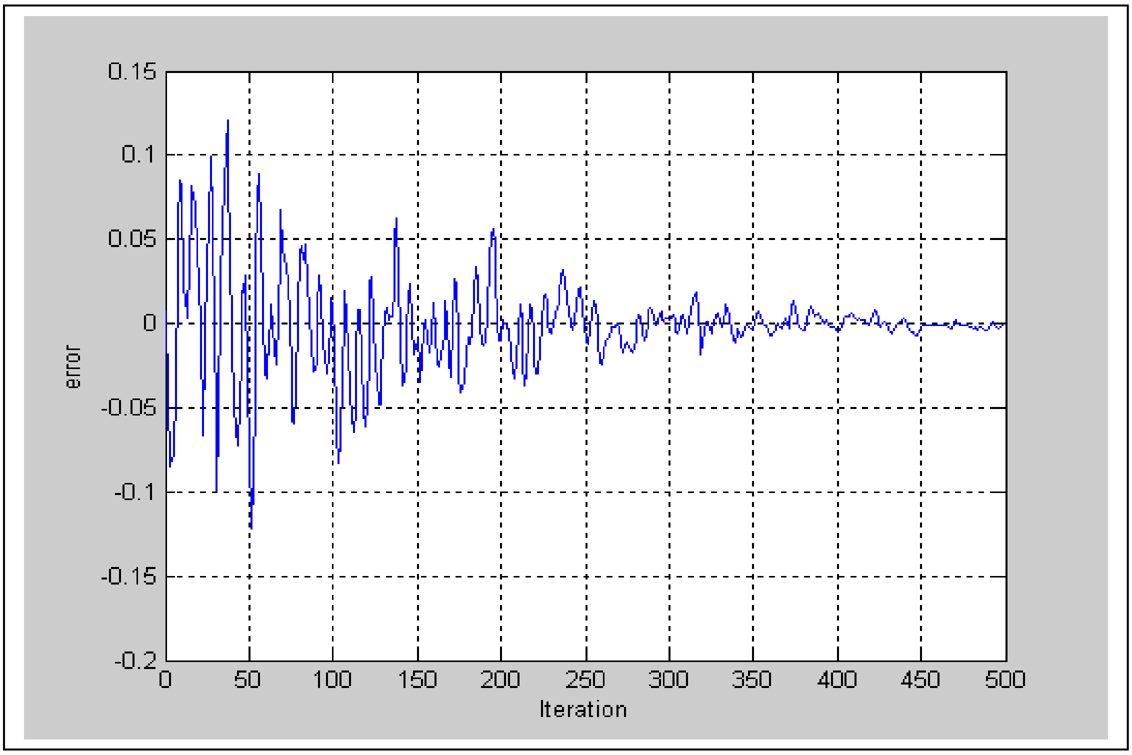

Figure-9: The learning curve for the LMS algorithm (μ = 0.8).

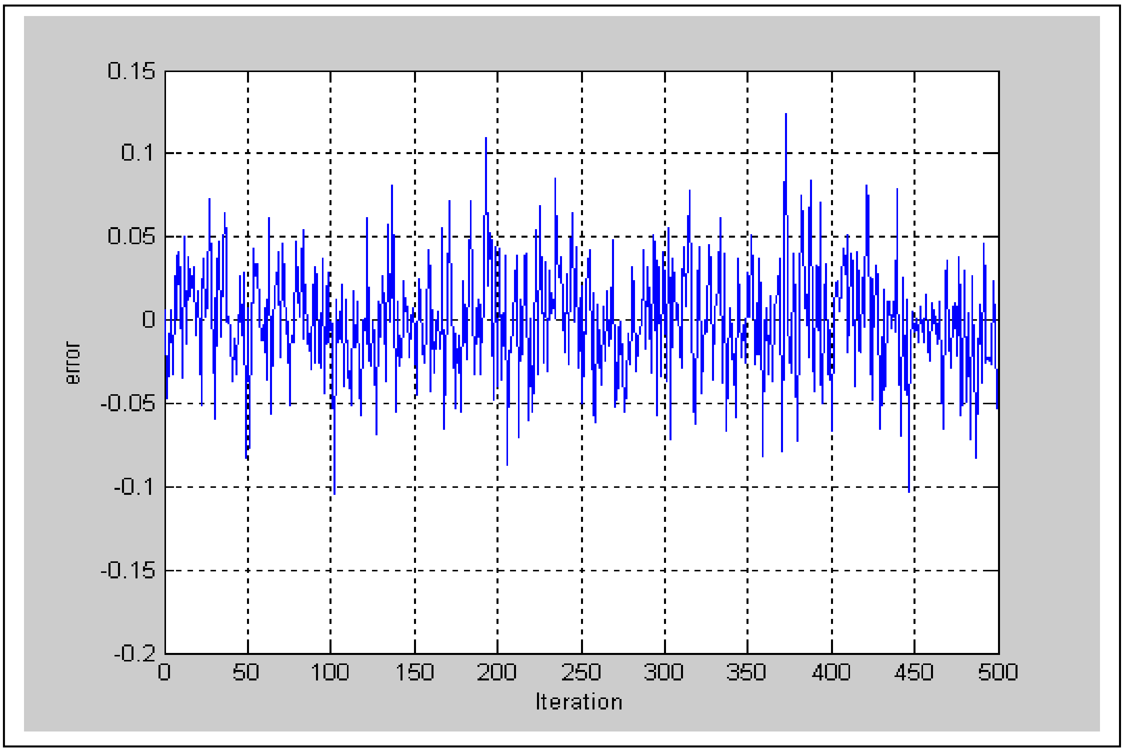

Figure-10: The learning curve for the ADALINE network.

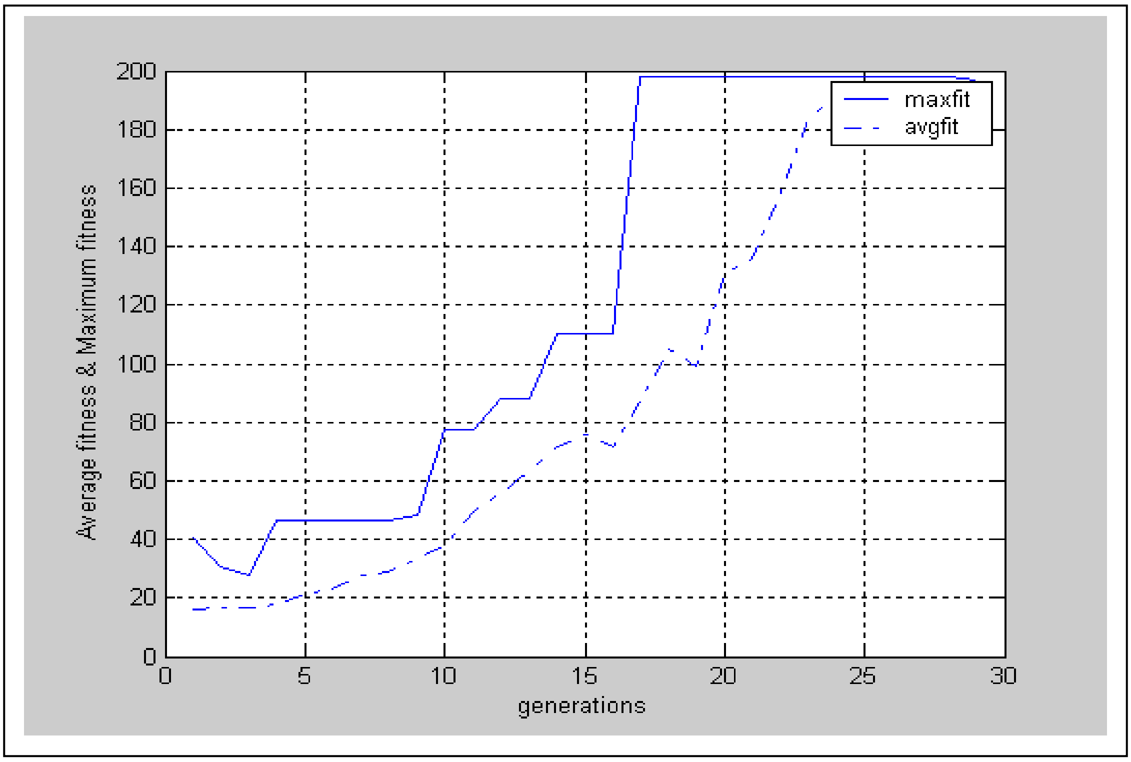

Figure-11: Average & maximum fitness vs. generations during GA operations

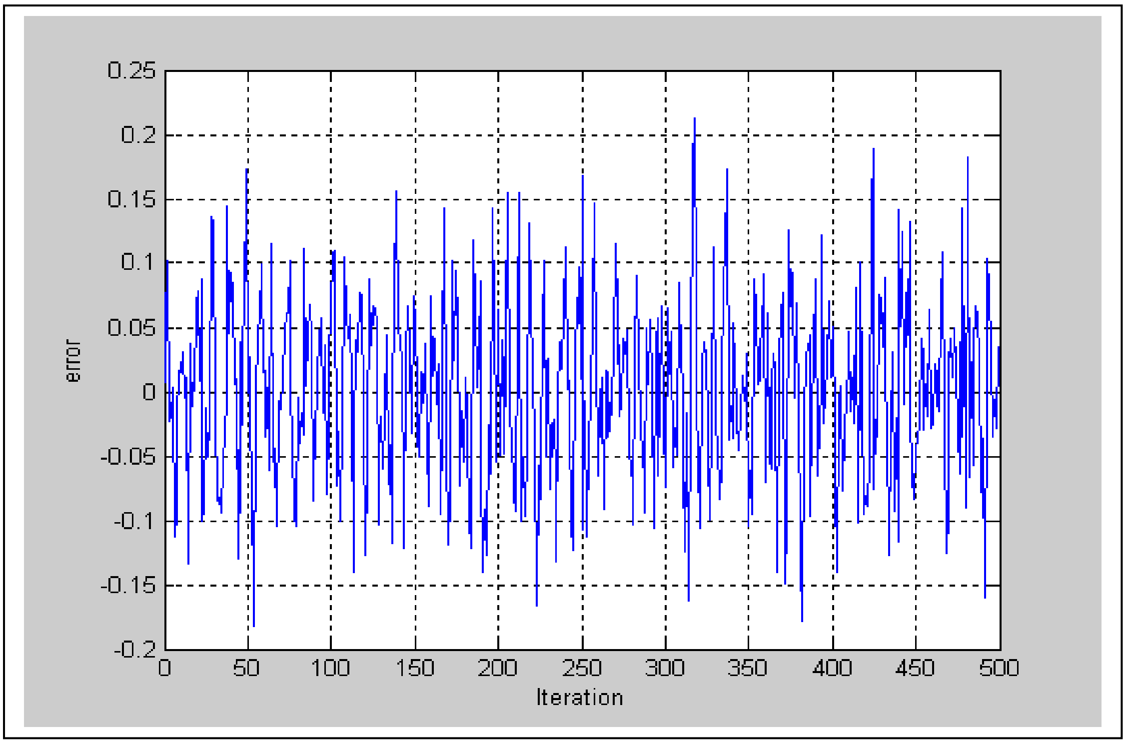

Figure-12: The learning curve according to GA operations.

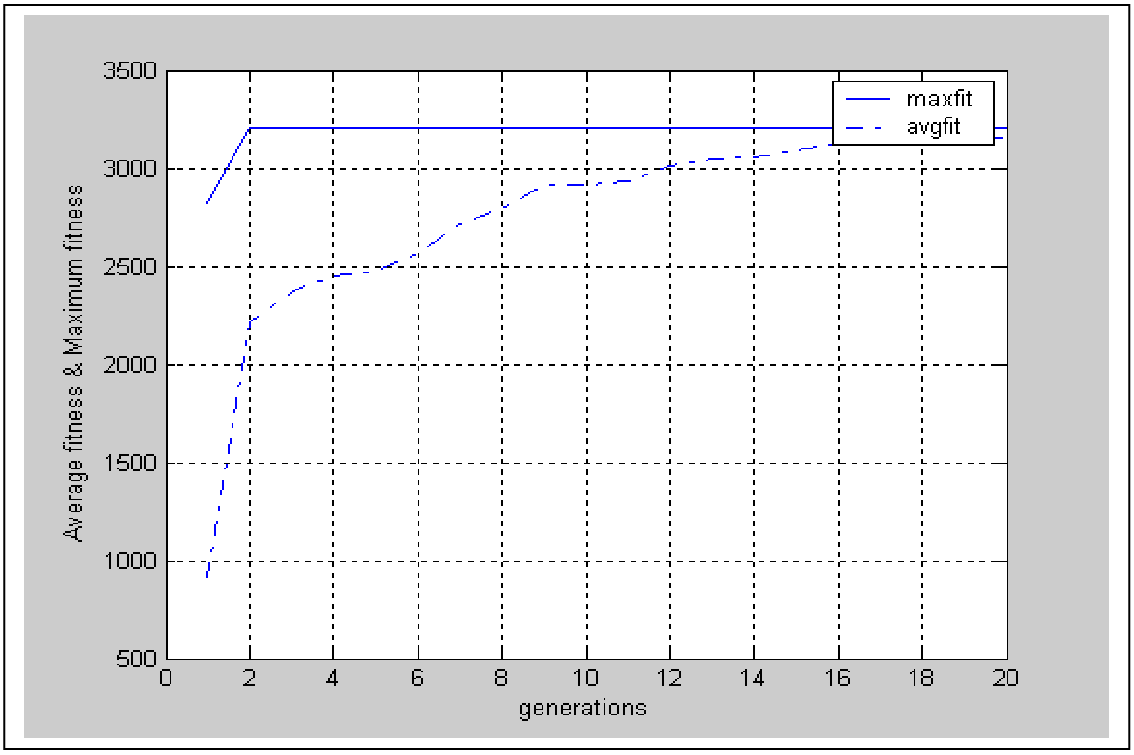

Figure-13: Average & maximum fitness vs. generations during GALMS operations.

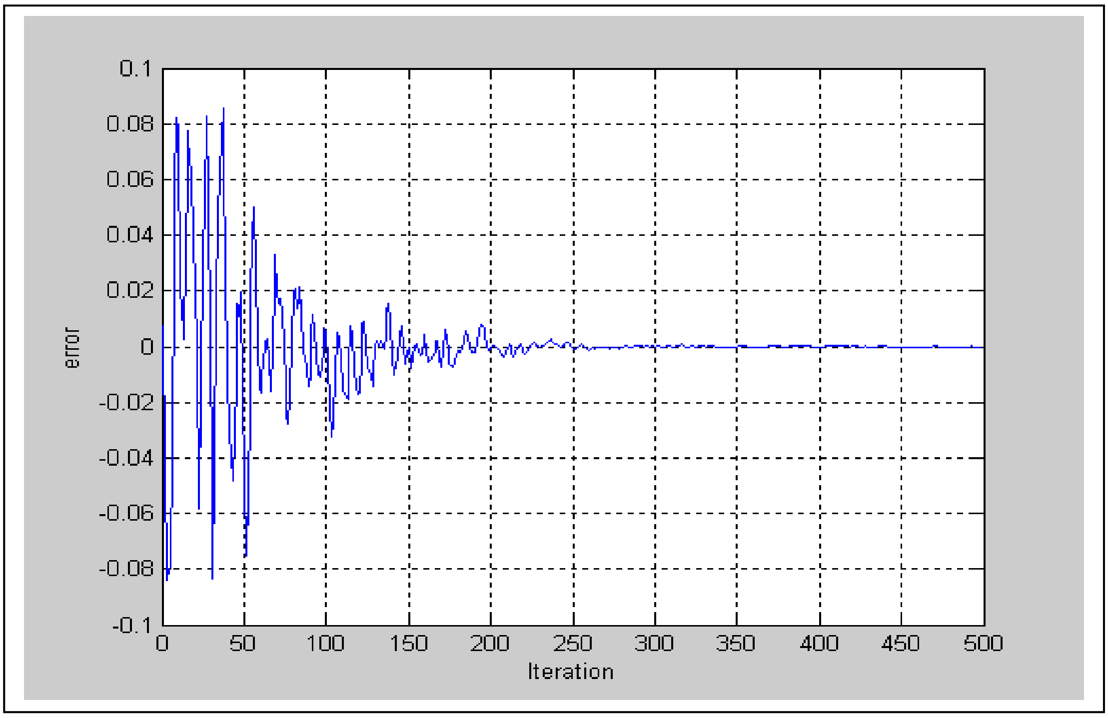

Figure-14: The learning curve according to GALMS operations.

Table-1 shows a comparison between our new GALMS algorithm and the known algorithms. This comparison depends on the obtained values for the MSE, the elapsed time and the filter coefficients between these methods.

Table-1: comparison between the methods result.

Algorithm	MSE	Elapsed Time/ Sec	Weights (Coefficients)						
			W_1	W_2	W_3	W_4	W_5	W_6	W_7
1- LMS	7.506e-004	1.332	-0.082	0.143	0.368	0.481	0.368	0.143	-0.083
2- ADALINE	0.0012	1.172	1.331	-0.693	-0.076	0.190	-0.023	0.113	-0.002
3- GA	0.0051	5.127	0.076	0.780	0.330	0.444	0.076	0.088	-0.268
4- GALMS	3.117e-004	9.904	-0.083	0.147	0.377	0.494	0.377	0.147	-0.083

9- Summary and Conclusion

An adaptive filter is essentially a digital filter with self-adjusting characteristics. It adapts automatically its weights to any change in the input signal. Many algorithms are used to adjust these weights depending on some parameters, such that error signal, minimized according to some criterion.

The common algorithm that has found wide-spread application is the LMS algorithm. The LMS algorithm is the most efficient algorithm because it does not suffer from the numerical instability problem inherent with the other algorithms, so the LMS algorithm does not require

The new algorithm (GALMS algorithm), discussed in this paper, uses the genetic search embedded into the LMS algorithm in order to improve this method by searching for optimal learning coefficient.

So this will give more accuracy and more convergence results with the identical filter.

According to the results presented previously in the figures and the comparison in the above table, we conclude that our new approach is the best method that gives minimum MSE and gets big concentration with the unknown system and the GALMS algorithm coefficients.

The only limitation of this algorithm that it takes a few greater elapsed time than the other methods. So our algorithm is suitable in the places that the elapsed time does not affect on the whole process else pure LMS algorithm can be used with some limitations.

Some of these limitations summarized in finding the suitable learning coefficient and this may make the LMS algorithm takes a greater elapsed time than our algorithm.

References

[1] Hayes, M.H., “Statistical Digital Signal Processing and Modeling”, John Wiley and Sons, 1996

[2] Haykin, S., “Adaptive Filter Theory”, Third Edition, Prentice-Hall, Inc., 1996.

[3] Widrow, B., and M. E. Hoff, “Adaptive switching circuits,” 1960 IRE WESCON Convention Record, New York IRE, pp. 96-104, 1960.

[4] Jackson, L., “Digital Filtering and Signal Processing with MATLAB Exercises”, Third edition, Kluwer Academic Publishers, 1996.

[5] Widrow, B., and S. D. Sterns, “Adaptive Signal Processing”, New York: Prentice-Hall, 1985.

[6] Hagan, M. T., H. B. Demuth, and M. H. Beale, “Neural Network Design”, Boston, MA: PWS Publishing, 1996.

[7] L. C. Jain & R. K. Jain *“Hybrid Intelligent Engineering systems”,* World Scientific Publishing, Co. Pte. Ltd., 1997.

[8] J. A. Freeman, & D. M. Skapura, *“ Neural Networks: Algorithms, Applications, and Programming Techniques”*, Addison-Wesley Publishing Company, Inc.,1992

[9] K. –F. Man, K.-S.Tang, S. Kwong & W. A. Halang., *“ Genetic Algorithm: Concepts and Design”* City University of Hong Kong, Tachee Aveneue, Kowloon, maito: email: %20k. nan @ ieee. Org. (2000).

[10] D. E. Goldberg, *“Genetic Algorithms in search, optimization, and Machine Learning,”* Addison-Wesley Publishing Company, Inc, U.S. (1989).

[11] S. C. NG., S. H. Leung, C. Y. Chung, A. Luk & W. H. Lan, *“The Genetic Search Approach (a new learning algorithm for adaptive IIR filtering ”* Addison-Wesley Publishing Company, Inc, U.S. (1995).