

Optimization of LMS Algorithm for Adaptive Filtering using Global Optimization Techniques

Shikha Tripathi

Assistant professor, Department of Electronics & Communications Engineering, JIT, Barabanki, India

Mohammad Asif Ikbal

Assistant professor, Department of Electronics & Communications Engineering, JIT, Barabanki, India

ABSTRACT

Adaptive filtering is a growing area of research due to its vast no of application in many fields and its numerous advantages over non adaptive filters. In fact there are many areas where the use of adaptive filters is becoming mandatory. Few of them are System Identification, Inverse Modeling, Linear Prediction, Feedforward Control etc. although enough work has been carried out on adaptive filters, still there are many fields where we can make significant contribution .One is the developing adaptive filtering for systems which are having a multimodal error surface, like IIR filters as gradient based optimization techniques, which are used so far in the designing of these type of system get stuck to The multi-modal error surface of these system and causes the gradient based algorithms to be stuck at local minima and not converge to the global optimum, resulting in an unstable system. In this work, we have combined the advantages of both gradient based algorithm and global optimizations algorithm to make the adaptive filters capable of efficiently working for the system having multimodal error surface. In this new method we use LMS as gradient based algorithm and Ant Colony Optimization (ACO) & Particle swarm optimization (PSO) as global optimization algorithm. In which ACO take inspiration from the behavior of real ant colonies to solve this type of optimization problems and PSO is a population based stochastic optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995, inspired by social behavior of bird flocking or fish schooling.

The algorithm is implemented using MATLAB, and the simulation results obtained shows that the proposed approaches is quite efficient, accurate and has a fast convergence rate. The results obtained also demonstrate that the proposed method can be efficiently used in designing and identification of systems having multimodal error surface.

Keywords

IIR, LMS, Ant Colony Optimization, Particle Swarm Optimization, System Identification

1. INTRODUCTION

Digital signal processing is a major area of research because of its never ending scope of development and it's a field which is never going to absolute. The backbone of digital signal processing is the filters; which may be classified on different basis according to our requirement or the problem under consideration. Major developments have been made in the area of non-adaptive filters but they are having their limitations. It is not possible for non-adaptive filters to

process signals which are time-varying and non-stationary as they require a prior knowledge of the statistics of the signal to be processed. This problem is solved by virtue of adaptive filters, which may be used very efficiently for these type of time varying and non stationary signals.

Now the digital signal processing systems may also be classified on the basis of error surfaces. Systems having uni-model error surface and systems having multi-modal error surface.

The applications of adaptive filters are very effectively done for the systems having uni-modal error surface. It uses gradient based algorithms for that purpose as the intrinsic stable behavior of these algorithm is effective for the application of adaptive filter on such systems. These algorithms are based on the process of the least-mean-square (LMS) and normalized least-mean square (NLMS) errors. However, since the gradient based algorithms try to find the global minimum of the error surface by moving in the direction of the negative gradient, approaches based on these algorithms may lead the filter to a local minimum when the error surface is multi-modal as such in IIR filters. Systems having multi-modal error surface can provide a much better performance than the uni-modal systems having the same number of coefficients.

1.1 Problem Definition

Despite of the advantages of systems having multi-modal error surfaces over systems having uni-model error surface we can't use these systems in the application of adaptive filters because of the following reasons.

- As discussed earlier gradient based algorithms (LMS) try to find the global minimum of the error surface by moving in the direction of the negative gradient, which may lead the system to a local minimum resulting into the instability during the process of adaptation. Figure 1 shows the systems having uni-modal & multi-modal error surface.
- The convergence speed of LMS algorithm decreases as the Eigen-value spread of the correlation matrix R , which is the ratio of the maximum to minimum Eigen value of the autocorrelation matrix, increases.
- Further, there is still another yet serious problem associated with LMS and NLMS algorithms, which is the choice of step-size of the search parameter that needs a trade-off between steady state miss adjustment and the speed of convergence.

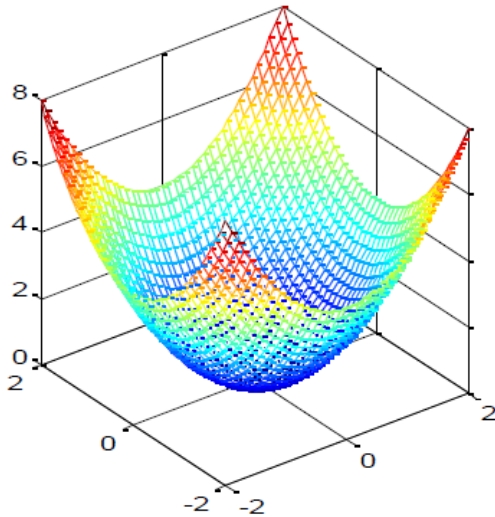
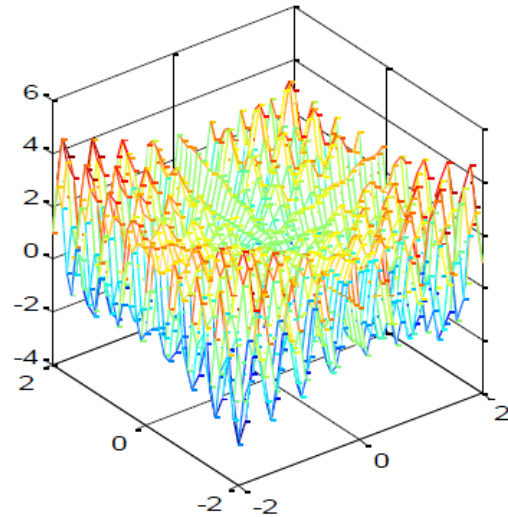


Fig 1: (a) Uni-model error surface



(b) multi-model error surface

1.2 Formulation of Problem

In this work, a new approach based on global optimizations techniques is presented for the systems having a multi-modal error surface. In this approach we have combined the advantages of both gradient based algorithm and global optimizations algorithm. This approach is very efficient for finding the optimal step size value of LMS-type algorithms.

When initialized in the global optimum valley, the LMS algorithm can be tuned to provide an optimal rate of convergence without fear of encountering a local minimum. Therefore, by using a global optimization techniques, such as ACO, PSO to quickly focus the population on regions of interest, an optimally tuned LMS algorithm can take over and provide better results than standard LMS.

Thus to summarize we can say that to apply the parameter vector, we use LMS algorithm and for implementation of LMS algorithm, we need to select appropriate value of the step size, which affects the stability and performance. For which we use ACO, PSO.

2. ANT COLONY OPTIMIZATION

Ant colony algorithm is the probabilistic technique to compute the computational problem. This algorithm is based on the food finding technique of ant. Ant is seeking for path between their colony and food. So ant searching for the food and search which food is nearer to their colony then establish shortest path between colony and food with the help of pheromones trial. The inspiring source of ACO is the foraging behavior of real ants.

ACO technique:

- probabilistic technique
- Searching for the optimal path in the graph based on behavior of ants seeking a path between their colony and source of food.
- Meta-heuristics optimization

2.1 Basic configuration of ACO

Ant moves random in path because they are blind. They select shortest path between colony and food source via pheromone trails. Each ant moves randomly and pheromone is deposited on the path. Ants move towards the maximum pheromones on

path. more the pheromone on the path increases the probability of the path followed by the ants. Pheromones get evaporated time to time so where, less no of ant are there pheromones is less to each ant attracted time to time strongest pheromones

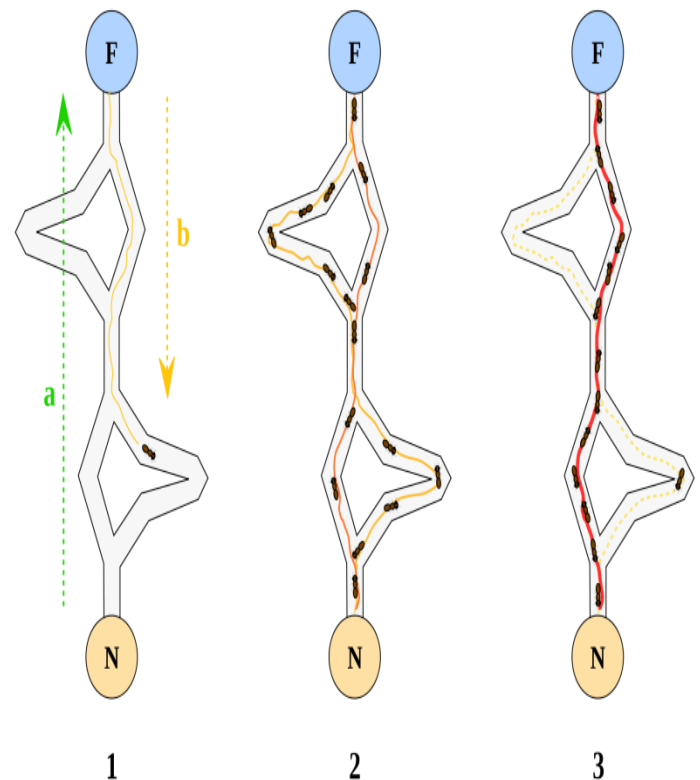
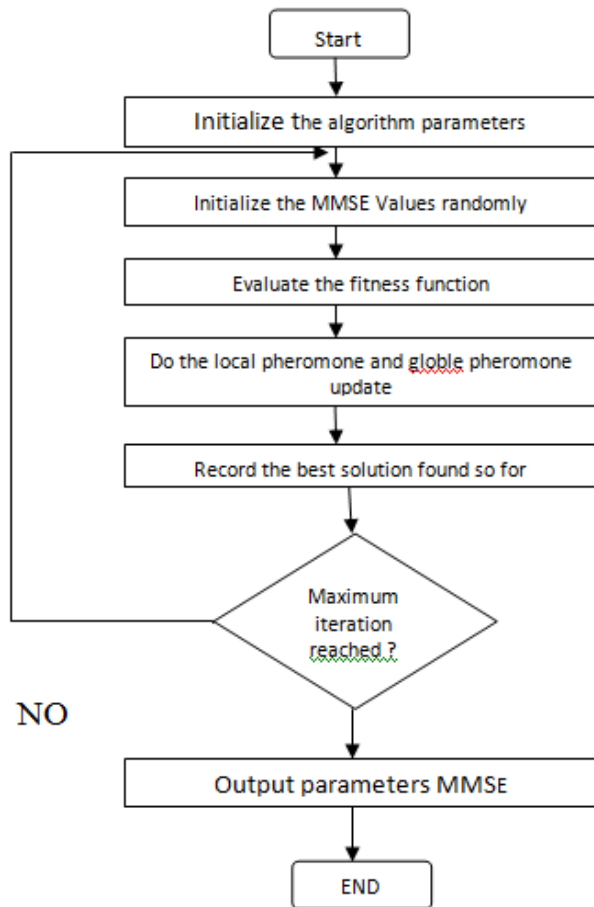


Fig 2: Showing the movement of ant from their nest to food source.

2.2 Flow chart representation of ACO



2.3 Implementation of ACO

Here we have shown the implementation of algorithm that adapts the behavior of real ants to solve the optimization problems. The information about these solutions makes an allusion to the communication system of real ants.

$$p_{ij}(t) = \frac{\tau_{ij}(t) \alpha \left(\frac{1}{d_{ij}} \right)^\beta}{\sum_{j \in \text{nodes}} \tau_{ij}(t) \alpha \left(\frac{1}{d_{ij}} \right)^\beta} \quad (1)$$

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij} + \sum_{k \in \text{colony}} \sum_{(i,j)} \frac{Q}{L_m} \quad (2)$$

Equation 1 represents the probability of ant to move between the two nodes i and j . Equation 2 represents the local updates of pheromone after travelling from node to node.

- Set a concentration of pheromone τ_{ij} to each link (i, j) .
- Assigned a number $k=1, 2, \dots, n$ in the nest.
- Iteratively build a path to the food source, using Eq. (1) for every ant.

Remove cycles and compute each route weight $f(x^k(t))$. A cycle could be generated when there are no feasible candidates' nodes then the predecessor of that node is included as a former node of the path.

- Update of the pheromone concentration using Eq. (2).
- Finally, finish the algorithm in any of the three different ways:
 - When a maximum number of approaches has been reached.
 - When it has been found an acceptable solution, with $f(x^k(t)) < \epsilon$.
 - When all ants follow the same path.

3. PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization (PSO) is a biologically inspired computational search and optimization method developed in 1995 by Eberhart and Kennedy based on the social behaviors of birds flocking or fish schooling.

Theory of particle swarm optimization (PSO) has been growing rapidly. PSO has been used by many applications of several problems. The algorithm of PSO emulates from behavior of animals societies that don't have any leader in their group or swarm, such as bird flocking and fish schooling. Typically, a flock of animals that have no leaders will find food by random, follow one of the members of the group that has the closest position with a food source (potential solution). The flocks achieve their best condition simultaneously through communication among members who already have a better situation. Animal which has a better condition will inform it to its flocks and the others will move simultaneously to that place. This would happen repeatedly until the best conditions or a food source discovered. The process of PSO algorithm in finding optimal values follows the work of this animal society. Particle swarm optimization consists of a swarm of particles, where particle represent a potential solution.

Recently, there are several modifications from original PSO. It modifies to accelerate the achieving of the best conditions. The development will provide new advantages and also the diversity of problems to be resolved. Study on the development of PSO is necessary to do to know how far its development, its advantages and disadvantages and how much use this method to settle a problem. Tutorial and theoretical of PSO has made about what is PSO, those describe about what PSO is, simple data tested, and comparison with others evolutionary computations.

3.1 Implementation of PSO

Exploration is the ability of a search algorithm to explore different region of the search space in order to locate a good optimum. Exploitation, on the other hand, is the ability to concentrate the search around a promising area in order to refine a candidate solution. With their exploration and exploitation, the particle of the swarm fly through hyperspace and have two essential reasoning capabilities: their memory of their own best position - local best (lb) and knowledge of the global or their neighborhood's best - global best (gb). Position of the particle is influenced by velocity. Let $x_i(t)$ denote the position of particle i in the search space at time step; unless otherwise stated, t denotes discrete time steps. The position of the particle is changed by adding a velocity $v_i(t)$, to the current position :

$$X_i(t+1) = x_i(t) + v_i(t+1) \quad (3)$$

Where,

$$v_i(t) = v_i(t-1) + c_1r_1 (\text{local best}(t) - x_i (t-1)) + c_2r_2 (\text{global best}(t) - x_i (t-1))$$

With $x_i \sim U(x_{\min}, x_{\max})$, acceleration coefficient c_1 and c_2 and random vector v_1 and v_2 Simple example of PSO, there is a function :Min $f(x)$

Where $x(B) \leq x < x(A)$

Denote $x(B)$ as a lower limit and $x(A)$ as an upper limit.

So, PSO procedure can be described by the following steps:

First, Assume that the size of the group of particle is N . It is necessary that the size N is not too large, but also not too small, so that there are many possible positions toward the best solution or optimal.

Second, generate initial population x with range $x(B)$ and $X(A)$ by random order to get the $x_1, x_2, x_3, \dots, x_n$. It is necessary if the overall value of the particle is uniformly in the search area. After that, the particle j and the velocity at iteration i are denoted as $x_j(i)$ and $v_j(i)$ thus, these initial particles will be $x_1(0), x_2(0), x_3(0), \dots, x_n(0)$. Vector $x_j(0)$, ($j = 1, 2, \dots, n$) is called a particle or vector coordinates of the particle. (Such as: chromosomes in genetic algorithms). Evaluation of the objective function value for each particle and expressed by $f[x_1(0)], f[x_2(0)], f[x_3(0)], \dots, f[x_n(0)]$. Then calculate the speed of all particles. All particles move towards the optimal point with a velocity. Initially all of the particle velocity is assumed to be zero. Set iteration $i = 1$, at the i^{th} iteration, find the three important parameters for each particle j that is:

- The best value of $x_j(i)$ (the coordinates of particle j at iteration i) and declare as $P_{\text{best}}(j)$, with the lowest value of objective function (minimization case) $f[x_j(i)]$, which found a particle j at all previous iteration. The best value for all particles $x_j(i)$ which found up to the i^{th} iteration, G_{best} with the value function the smallest goal / minimum among all particles for all the previous iterations, $f[x_j(i)]$.
- Calculate the velocity of particle j at iteration i using the following formula :

$$v_i(t) = v_i(t-1) + c_1r_1 (\text{local best}(t) - x_i (t-1)) + c_2r_2 (\text{global best}(t) - x_i (t-1))$$

Where c_1 and c_2 , respectively, are learning rates for individual ability (cognitive) and social influence (group) r_1 , and r_2 and uniformly random numbers are distributed in the interval 0 and 1. So the parameters c_1 and c_2 represent weight of memory (position) of a particle towards memory (position) of the groups (swarm). The value of c_1 and c_2 is usually 2, so multiply c_1r_1 and $c_2 r_2$ ensure that the particles will approach the target about half of the difference.

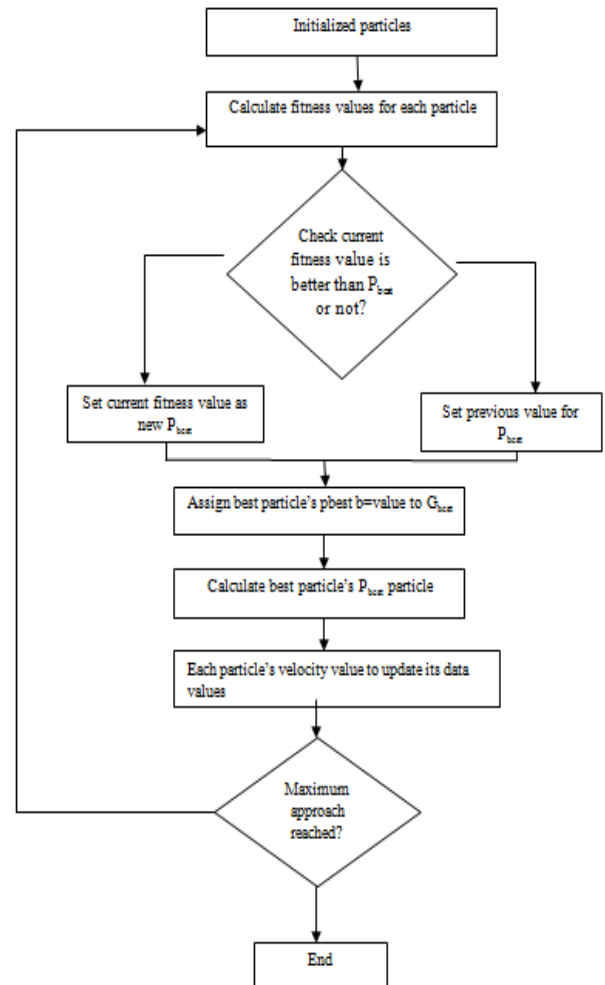
- Calculate the position or coordinates of particle j at i^{th} iteration by Equation 3

Evaluation the objective function value for each particle and expressed as: $f[x_1(i)], f[x_2(i)], f[x_3(i)], \dots, f[x_n(i)]$

The last step, check whether the current solution is convergent. If the positions of all particles leading to an equal value, then this is called convergence. If not convergent then step 4 is repeated by updating iterations $i = i + 1$, by calculating new values from $P_{\text{best}}(j)$ and G_{best} . This iteration process continues until all particles convergence the same solution. Usually be determined by the termination criteria (Stopping criterion), for example the amount of the excess solution with a solution now previously been very small.

If the current solution is convergent, then the iteration will stop. We do not know whether the final value is the best value. Below are the stopping criteria conditions for the iteration: First, terminate when a maximum number of iterations, or FEs, has been exceeded. Second, Terminate when an acceptable solution has been found, Third, Terminate when no improvement is observed over a number of iteration. Fourth, terminate when the normalized swarm radius is close to zero. Fifth, terminate when the objective function slope is approximately zero. Although the particle has stopped, we do not know whether the particle will pitch on local optima, local minima, global optima or global optima.

3.2 Flow chart representation of PSO



4. PROPOSED APPROACH

So far in our discussion we have concluded that an adaptive algorithm can't be applied to the systems with multimodal error surface because of problems listed in sec...

In the available method for systems having uni-modal error surface we adjust the parameters of an adaptive filter to minimize a cost function chosen for the task at hand using LMS adaptive algorithm. In this approach we have combine LMS algorithm with PSO & ACO. The general form of proposed algorithm is.

$$W(n + 1) = W(n) + \mu(n)G(e(n), X(n), \Phi(n)), \quad (4)$$

Where $G(\cdot)$ is a particular vector-valued nonlinear function, $\mu(n)$ is a step size parameter, $e(n)$ and $X(n)$ are the error signal

and input signal vector, respectively, and $\Phi(n)$ is a vector of states that store important information about the characteristics of the input and error signals and/or the coefficients at previous time instants. The step size is so called because it determines the magnitude of the change or “step” that is taken by the algorithm in iteratively determining a useful coefficient vector. In this approach it is optimized by ASO, PSO.

4.1 The Mean-Squared Error Cost Function

The form of $G(\cdot)$ in (4) depends on the cost function chosen for the given adaptive filtering task. We now consider one particular cost function that yields a popular adaptive algorithm. Define the mean-squared error (MSE) cost function as

$$J_{MSE}(n) = \frac{1}{2} \int_{-\infty}^{\infty} e^2(n) p_n(e(n)) de(n) \quad (5)$$

$$= \frac{1}{2} E\{e^2(n)\},$$

Where $p_n(e)$ represents the probability density function of the error at time n and $E\{\cdot\}$ is shorthand for the expectation integral on the right-hand side of (5). The MSE cost function is useful for the systems having multimodal error surface because

- $J_{MSE}(n)$ has a well-defined minimum with respect to the parameters in $W(n)$;
- The coefficient values obtained at this minimum are the ones that minimize the power in the error signal $e(n)$, indicating that $y(n)$ has approached $d(n)$; and
- $J_{MSE}(n)$ is a smooth function of each of the parameters in $W(n)$, such that it is differentiable with respect to each of the parameters in $W(n)$.

The third point is important in that it enables us to determine both the optimum coefficient values given knowledge of the statistics of $d(n)$ and $x(n)$ as well as a simple iterative procedure for adjusting the parameters of an FIR filter.

5. SIMULATION & THE ANALYSIS OF RESULT

The control parameter values of the ACO based and other evolutionary algorithms used in the simulations are given in following table.

Table 1. Control parameter values used in the simulations

ACO :	PSO :
Num Of Ants = 20	Swarm size = 5
Pheromone = 0.2	Inertia factor, $\omega= 0.5$
Evaporation Parameter = 0.1	Cognitive factor, $c1=0.5$
Positive Pheromone = 0.2	Social factor, $c2= 0.5$
Negative Pheromone = 0.3	LB = 0.02
Max Tour = 600	UB = 0.05
Min Value = -1000	-

Max Value = 1000	-
LB = 0.02	-
UB = 0.05	-

Table 2. ACO Simulation for different values of μ & α parameters and their Analysis

S.No.	α	μ	MMSE
1	0.9	0.02342	0.01737
2	0.8	0.03979	0.03515
3	0.7	0.0333	0.03121
4	0.6	0.04815	0.03399
5	0.5	0.03676	0.04058
6	0.4	0.03722	0.03187
7	0.3	0.03652	0.02025
8	0.2	0.02718	0.01798
9	0.1	0.04828	0.03090

Table 3. PSO Simulation for different values of μ & α parameters and their Analysis

S.No.	α	μ	MMSE
1	0.9	0.02787	0.01471
2	0.8	0.04588	0.04027
3	0.7	0.02634	0.02277
4	0.6	0.02797	0.01716
5	0.5	0.04708	0.0889
6	0.4	0.04148	0.0341
7	0.3	0.04743	0.02126
8	0.2	0.03675	0.02288
9	0.1	0.0468	0.02767

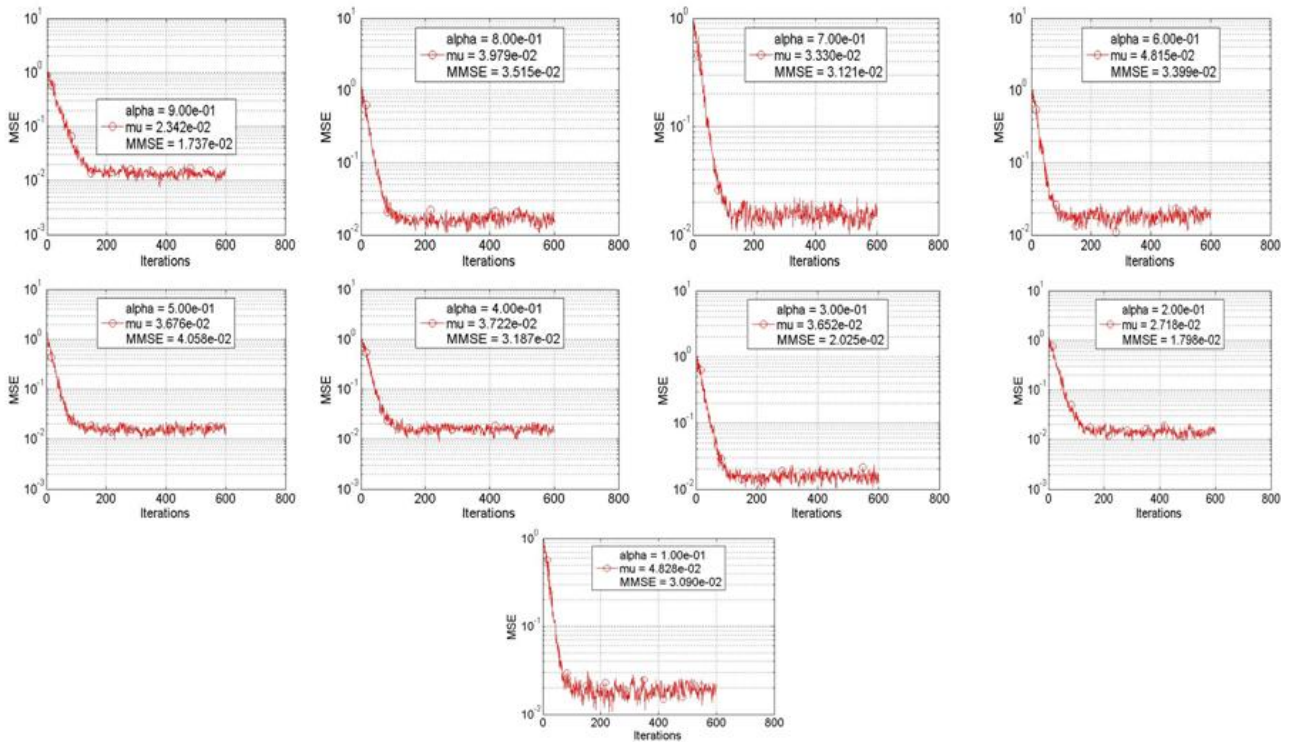


Fig 2: simulation results based on ACO

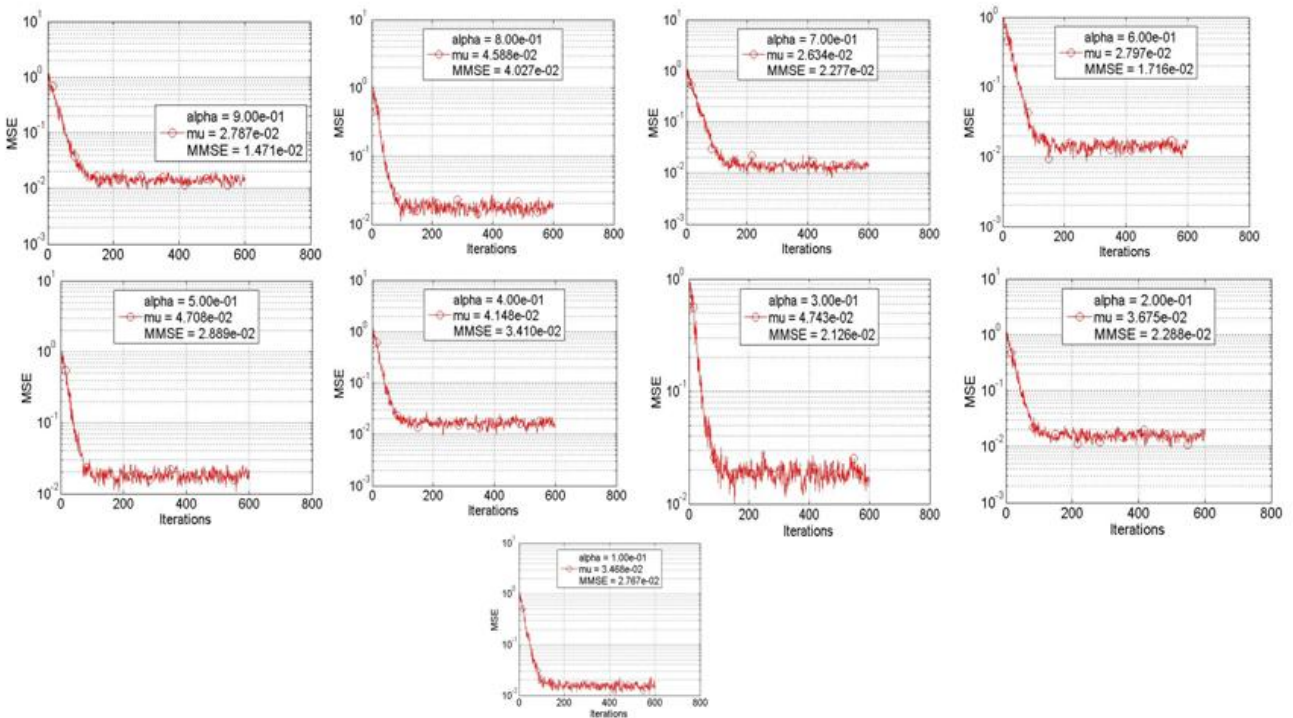


Fig 3: simulation results based on PSO

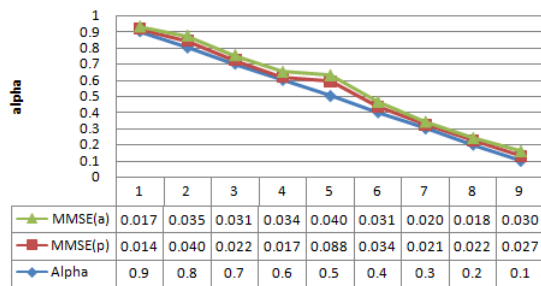


Fig 4: showing a comparative study of ACO & PSO, algorithm based calculation of MMSE.

6. CONCLUSION

In this work application of the Ant Colony Optimization technique and Particle Swarm Optimization Technique is presented for Step Size Optimization of LMS Algorithm for systems having multi-modal error surfaces. It has been found that ACO & PSO have significantly improved the adaptive algorithms to be used for systems with multi-modal error surfaces. We can say that it is a powerful and robust algorithm as compared with the previous methods for finding the optimal step size value of LMS-type algorithms proposed in the design of adaptive filters. From the simulation result, it can be easily seen that the proposed algorithm has a noticeable performance over PSO in terms of the averaged mean squared error and PSO is fast convergent with respect to ACO.

7. REFERENCES

- [1] Paulo S. R. Diniz, Adaptive Filtering Algorithms and Practical Implementations, Springer, USA, 2008.
- [2] S. Haykin, Adaptive Filter Theory, Prentice Hall, USA, 2002.
- [3] D. J. Krusienski, W. K. Jenkins, Design and performance of adaptive systems based on structured stochastic optimization strategies, IEEE Circuits Systems Magazine 5 (2005), pp. 8-20.
- [4] S. C. Ng, S. H. Leung, C. Y. Chung, A. Luk, W. H. Lau, The genetic search approach: A new learning algorithm for adaptive IIR filtering, IEEE Signal Processing Magazine 13 (1996), pp. 38-46.
- [5] N. Karaboga, Digital IIR filter design using differential evolution algorithm, EURASIP Journal on Applied Signal Processing 8 (2005), pp. 1-9.
- [6] A. Kalinli, N. Karaboga, A parallel tabu search algorithm for digital filter design, COMPEL-The International Journal for Computation and Mathematics in Electrical and Electronic Engineering 24 (2005), pp. 1284-1298.
- [7] N. Karaboga, B. Cetinkaya, Design of digital FIR filters using differential evolution algorithm, Circuits Systems and Signal Processing Journal 25 (2006) , pp. 649-660.
- [8] <https://www.scribd.com/doc/279933360/Optimized-Variable-Step-Size-Normalized-LMS-Adaptive-Algorithm-for-Echo-Cancellation>. LMS Algorithm.
- [9] D. J. Krusienski, W. K. Jenkins, Particle swarm optimization for adaptive IIR filter structures, Congress on Evolutionary Computation, 2004, pp. 965-970.
- [10] A. Kalinli, N. Karaboga, A new method for adaptive IIR filter design based on tabu search algorithm, International Journal of Electronics and Communication 59 (2004), pp. 1-7.
- [11] S. Chen, B. L. Luk, Adaptive simulated annealing for optimization in signal processing applications, Signal Processing 79 (1999), pp. 117-128.
- [12] N. Karaboga, A new design method based on artificial bee colony algorithm for digital IIR filters, Journal of the Franklin Institute-Engineering and Applied Mathematics 346 (2009), pp. 328-348.
- [13] N. KARABOGA, A. KALINI, D. KARABOGA, Designing digital IIR filter using ant colony optimization algorithm. Engineering applications of artificial intelligence april 2004. Vol.17 (3)
- [14] Dissanayake, S.D. Performance analysis of noise cancellation in a diversity combined ACO-OFDM system. ICTON, 2012
- [15] P. S. R. Diniz, Adaptive Filtering: Algorithms and Practical Implementations, Kluwer Academic Publishers, Boston, 1997.
- [16] P. S. R. Diniz, Adaptive Filtering: Algorithms and Practical Implementations, Kluwer Academic Publishers, Boston, 1997.
- [17] Ioan Tabus, Stochastic gradient based adaptation: Least Mean Square (LMS)Algorithm, SGN 21006 Advanced Signal Processing:Lecture 5
- [18] P. Visu and E. Kannan, Traffic Parameterized ACO for Ad-Hoc Routing
- [19] Ali M. and Babak, A. "A new clustering algorithm based on hybrid global optimization based on a dynamical systems approach algorithm", Expert Systems with Applications (Elsevier), Vol. 37,pp. 5645-5652, 2010
- [20] Dorigo, M. and Stutzle, T. "Ant Colony Optimization", MIT Press, Cambridge MA, 2004
- [21] Dorigo, M., Maniezzo, V. and Coloni, A., "Ant System: Optimization by a colony of cooperating agents," IEEE Transactions on Systems, Man, and Cybernetics—Part B, Vol. 26, No. 1, pp. 29 – 41, 1996
- [22] Frank, N. and Carsten, W. "Ant Colony Optimization and the minimum spanning tree problem", Theoretical Computer Science (Elsevier), Vol. 411, pp. 2406-2413, 2010
- [23] Goss, Aron, Deneubourg, and Pasteels, "Selforganized shortcuts in the Argentine ant," Naturwissenschaften, Vol. 76, pp. 579–581, 1989
- [24] Hsin-Yun, L., Hao-Hsi, T., Meng-Cong, Z. and Pei-Ying, L. "Decision support for the maintenance management of green areas", Expert Systems with Applications (Elsevier), Vol. 37, pp. 4479- 4487, 2010
- [25] KwangMongSim and Weng Hong Sun, "Ant Colony Optimization for Routing and Load-Balancing: Survey and New Directions IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems And Humans", Vol. 33, No. 5, 2003
- [26] Li-Ning Xing, Ying-Wu Chen, Peng Wang, Qing-Song Zhao, JianXiong, "A Knowledge-Based Ant Colony Optimization for Flexible Job Shop Scheduling Problems", Applied Soft Computing 10 (2010),pp. 888–896