International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 13 (2017) pp. 3988-3995

Soft Computing Techniques to Estimate FIR Filter Weights in an Adaptive Channel Equalizer: A Comparative Study

Rashmi Sinha
Associate Professor,
Department of Electronics and Communication Engineering,
National Institute of Technology, Jamshedpur
Jharkhand, India.
ORCID: 0000-0002-8130-6129

Arvind Choubey
Professor, Department of Electronics and Communication Engineering,
National Institute of Technology, Jamshedpur
Jharkhand, India.

Abstract

An adaptive channel equalizer is ubiquitous in combating the effect of Inter-Symbol Interference (ISI) and other impairments caused by cross talk, additive noise, electronic components present in transceivers and the nature of medium, on the digital data transmitted in a communication system. Techniques to estimate the weights of finite impulse response (FIR) filter, which forms an integral part of the equalizer, using few popular soft computing techniques are presented here. Parameterized expression of the filter cost function under adaptation and corresponding training set is created. Thereafter, global minimization process is executed using Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Bacterial Foraging Optimization (BFO), Wind Driven Optimization (WDO) and the benchmark Least Mean Square (LMS) algorithm. A design example is undertaken to verify the effectiveness of each technique in estimating the weights of FIR filter based adaptive channel equalizer. Extensive simulation is carried out on linear and nonlinear channel using the five popular and successful algorithms at 10dB additive noise. WDO achieves minimum error of -9.574dB at 35 iterations as compared to -5.153dB at 20 iterations, -5.993dB at 33 iterations, -8.66dB at 23 iterations for GA, PSO and BFO respectively. It also shows better performance in terms of bit error rate (BER), which is 800 bits as compared to 5717, 13640, 3630 and 8508 bits in 10^6 samples for LMS, GA, PSO and BFO respectively. This paper presents a good and comprehensive set of results and states arguments for the merits and demerits of each of the technique. On careful observation, it is revealed that WDO emerges as the fastest optimization algorithm for nonlinear channel.

Keywords: Bit Error Rate (BER); Equalizer; Finite Impulse Response (FIR); Inter-Symbol Interference (ISI); Optimization

INTRODUCTION

The communication channel through which data is transmitted is often subjected to practical considerations such as restricted bandwidth allocated to the channel, presence of multipath effects in the transmission medium, nonlinear and time varying characteristics of the channel itself and noise present in the ambience, to name a few. A serious consequence of these practical limitations is the occurrence of ISI and other impairments like cross-talk, noise etc. which affects the accuracy of the data at the receiver side [1]. Adaptive channel equalizer employed in series with the channel is an effective solution to reduce the error in number of bits received. FIR filter, commonly referred as linear transversal filter is a widely implemented form of adaptive equalizer whose response can be adjusted by changing the weights of the filter through some optimization techniques to meet specific measured channel characteristics. The task of an adaptive equalizer can be subdivided into three parts. First, filter weights need to be estimated or initialized, then the filtering of distorted received data is performed and finally, filter parameters or coefficients are adapted to a changed environment using suitable optimization techniques [2].

Traditionally, there are two major classes of optimization algorithms, calculus based technique and enumerative technique. Calculus-based optimization technique employs the gradient directed searching mechanism to solve the error surface or differentiable surface of an objective function [3]. However, in signal processing ill-defined or multimodal objective function is encountered since the signal can be noisy, fuzzy, vague and discontinuous. Local optima are frequently obtained for such objective functions. Least Mean Square (LMS), Recursive Least Square (RLS) and Artificial Neural Network (ANN) are few popular algorithms belonging to this category. Local optima can be handled by enumerative techniques in operation search such as Dynamic Programming (DP) [4]. But its simplicity, robustness and popularity is overshadowed for high computational complexity. In addition, DP may break down on complex problems faced in channel equalizers due to its large dimensions, a situation that is widely known as the "curse of dimensionality". Henceforth, these anomalies of traditional algorithms have paved way for biological and nature inspired soft computing approaches. Many problems that were unsolvable in the past can now be tackled with ease. Many soft computing tools have been

successfully implemented in designing FIR based channel equalizer such as BFO [5,6], GA [7], PSO [8], WDO [9], Artificial Immune System (AIS) [10] etc. In this article, weights of FIR filter of channel equalizer are estimated based on training data that is known a priori at the receiver. Four soft computing techniques, PSO, GA, WDO and BFO are presented as optimization algorithms to estimate the weights. All of them are compared to LMS method which is the benchmark algorithm for designing channel equalizer. A comparative study on their performance in terms of convergence characteristics and BER is summarized at the end.

PROBLEM FORMULATION

An eight tap FIR filter based equalizer for a channel modeled on 3- tap FIR filter in a communication system scenario is depicted in Fig. 1, where the block labeled 'NL' represents the nonlinearity introduced by the channel.

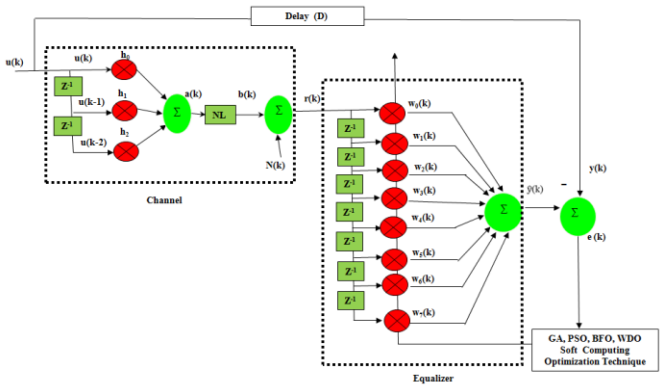

Figure 1: FIR filter based channel equalizer in a communication system scenario

The equalizer models the inverse transfer function of the channel and the order of the inverse transfer function is equal to or higher than that of the forward transfer function. The input to the channel is considered to be uniformly distributed random binary data u(k). The output of the channel at k^{th} instant is given by

$$a(k) = \sum_{i=0}^{L} h_i u(k-i) \qquad 0 < k \le n \tag{1}$$

where L is the length of channel, h_i is the channel taps and n is the number of transmitted symbols. The nonlinearly distorted output b (k) corresponding to a (k) can be written as b (k) =g (a(k)), where g (·) is the nonlinear function associated with the block NL. It is assumed that the channel is affected by AWGN, N (k). So, the received signal at the equalizer is $r(k) = b(k) + N(k)$. The compensated output $\hat{y}(k)$ from the equalizer is compared with the desired signal y (k) = u (k-D), which is the delayed version of the input signal u (k). The time delay of the signal transmitted through the physical channel is represented by D. The error signal defined by e (k) is used to modify the internal parameters of the equalizer according to the optimization algorithm adopted for training, where e (k) is represented by

$$e(k) = y(k) - \hat{y}(k) \tag{2}$$

The equalization process is viewed as an optimization problem where cost function or the fitness function is given by following mean square error (MSE) criterion

$$J = \frac{1}{n}\sum_{k=1}^{n} e^2(k). \tag{3}$$

SOFT COMPUTING TECHNIQUES

A brief description of the soft computing algorithms used to estimate the filter weights and steps for their implementation in adaptive channel equalizer is presented in this section.

A. Wind Driven Optimization

WDO is a population based iterative heuristic global optimization technique similar to other nature inspired optimization algorithms, aiming to improve the best candidate solution over time. It is highly correlated with the actual physical equations describing the trajectory of an air parcel in our atmosphere. The populations of air parcels are ranked in descending order based on their pressure values such that its new velocity u_{new} and new position x_{new} can be represented as [11]

$$u_{new} = (1-\alpha)u_{cur} - gx_{cur} + \left(RT\left|\frac{1}{i} - 1\right|(x_{opt} - x_{cur})\right) + \frac{cu_{cur}^{otherdim}}{r} \tag{4}$$

$$x_{new} = x_{cur} + u_{new}\Delta t \tag{5}$$

where u_{cur} is the velocity at the current iteration, α and g being friction coefficient and gravitational constant respectively, x_{cur} is the current location, RT defines universal gas constant and temperature, C is the Coriolis force, x_{opt} being the optimum location, r is the ranking among all parcels and $cu_{cur}^{otherdim}$ is the replaced velocity vector from another randomly chosen dimension to represent the influence of coriolis force. A time step, Δt equal to 1 is assumed. Coriolis force and gravitational pull in WDO provides a favorable contribution which prevents air parcels from remaining trapped at the boundary for long period and pulls them back into the search space. WDO coefficients can be fine-tuned for different optimization topologies to provide potential benefits.

The air parcel population is ranked based on their pressure value (cost function) and velocity is updated according to (4) with following limitation

$$u_{new}^{*} = \begin{cases} u_{max} & if\ u_{new} > u_{max} \\ -u_{max} & if\ u_{new} < -u_{max} \end{cases} \tag{6}$$

where the direction of motion is preserved but the magnitude is limited to $|u_{max}|$ at any dimension and u*new represents the adjusted velocity after it is limited to the maximum speed. The lowest ranked fitness function is taken as x_{opt}value.

Detail procedure for implementing WDO algorithm for estimating FIR filter weights of equalizer is given in Table 1.

Table 1: WDO algorithm for estimating the weights of FIR filter of equalizer

Step#	Description
1	Determination of output of the channel: The input is a random binary signal drawn from a uniform distribution. The number of input samples is n, which is passed through the channel to produce output, a (k) given by (1).
2	Equalizer input: The output of the channel r(k)is passed through 'Q' number of tap delay elements of the equalizer to produce the error vector using (3).
3	Initialization of air parcels: Since it is an evolutionary algorithm, we begin with a group of random solutions, which is a group of weight vectors of equalizer. Each weight vector consists of Q number of elements and is represented by air parcel which is basically a binary string of definite length. So, a set of binary strings equal to the population of air parcels is initialized to represent corresponding weight vector. Each weight vector in the set can be a probable solution.
4	Calculation of desired output of the equalizer: The desired signal y(k) is formed by delaying the input sequence u(k) by m samples, where $m = Q/2$ or $(Q + 1)/2$.
5	Fitness evaluation: For each i^{th} weight vector, MSE is determined using (3) and is used as a cost/fitness function.
6	Ranking of air parcel: Air parcels are ranked according to (4) and (6).
7	Weight updating: Weight vector is updated using (5).
8	Steps 5, 6 and 7 are repeated until maximum number of iterations is reached.
9	At the end of stipulated iterations, almost all air parcels occupy same position and the best solution i.e. least ranked weight vector is considered as optimum solution.

B. Genetic Algorithm

GA is a powerful global optimization searching process based on the mechanics of natural selection and evolutionary genetics [12]. The algorithm begins with a randomly generated population of chromosomes. A typical GA cycle is shown in Fig. 2.

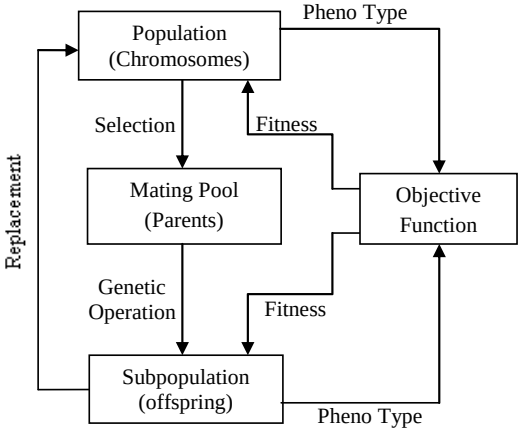

Figure 2: Genetic Algorithm Cycle

The population is updated after each learning cycle through three evolutionary processes: selection, crossover and mutation. These create a new generation of solution variables. The selection function creates a mating pool of parent solution strings based upon the "survival of fittest" criterion. From the mating pool, the crossover operator exchanges gene information. This essentially crosses the more productive genes from within the solution population to create an improved productive generation. Mutation alters selected genes randomly and helps prevent premature convergence by pulling the population into unexplored areas of the solution surface.

It further adds new gene information into the population. The algorithm to implement in channel equalizer is stated in Table 2.

Table 2: GA for estimating the weights of FIR filter of equalizer

Step #	Description
1.	Randomly generate an initial population $w(k) = [w_0, w_1, \ldots . w_Q]$
2.	Compute the fitness function of each chromosome w_i in the current population w(k) according to (3).
3.	Create new chromosomes $w_{new}(k)$ by mating current chromosomes, applying mutation and recombination as the parent chromosomes mate.
4.	Delete numbers of the population to make room for the new chromosomes.
5.	Compute the fitness of $w_{new}(k)$ and insert these into population.
6.	k := k+1, if not (end test) go to step 3 or else stop and return the best chromosome.

Although GA is a powerful optimization tool, it does have certain weaknesses and barriers which have to be overcome before it can be applied to real world implementations. Online system performance becomes unpredictable due to the randomness of the GA operation.

C. Particle Swarm Optimizaton

PSO is an evolutionary computation technique rooted on the notion of swarm intelligence of insects, birds, bees, ants etc. [13]. Similar to GA, a PSO system is initiated with a population of random solutions and searches for optima by updating generations. However, it has no evolution operators such as crossover and mutation. Instead, the potential solutions called particle fly through the problem space and the trajectory of each particle is influenced in a direction determined by the previous velocity and the location of global best position *gbest*, which is the best solution achieved by the swarm so far and previous personal best position called *pbest*. Since, the particles have memory, corresponding fitness value is also remembered. The position of i^{th} particle in d^{th} dimension represented by $X_i(t) = \left(X_{i1}(t), X_{i2}(t) \ldots . X_{id}(t)\right)$ and the corresponding velocity $V_i(t) = (V_{i1}(t), V_{i2}(t), \ldots \ldots . V_{id}(t))$ is updated according to [14]

$$V_i(k+1) = \omega * V_i(k) + C1 * \text{rand}(.) * \left(P_i(k) - X_i(k)\right) + C2 * \text{rand}(.) * \left(P_g(k) - X_i(k)\right) \quad (7)$$

$$X_i(k+1) = X_i(k) + V_i(k+1). \quad (8)$$

where Pg= (Pg1,Pg2….Pgd) and Pi= (Pi1,Pi2….Pid) are the d^{th} dimensional positions of the previous *gbes*t and *pbest* respectively, rand(.) is the random number in the range [0,1] which is different in different dimensions, C1 and C2 are acceleration coefficients and ω is the inertia weight which plays the role of balancing the local search and global search. Personal best position of each particle is updated using the condition

$$P_{id}(k+1) = \begin{cases} P_i(k); \text{ if } f\left(X_i(k+1)\right) \geq f(P_i(k) \\ X_i(k+1); \text{if } f\left(X_i(k+1)\right) \leq f(P_i(k) \end{cases} \quad (9)$$

and gbest found by any particle during all previous iterations is defined as

$$P_g(k+1) = \arg\min_{P_i} f(P_i(k+1)), \quad 1 \leq i \leq M \quad (10)$$

The PSO algorithm for FIR filter channel equalizer is described in Table 3.

Table 3: The PSO algorithm for estimating the weights of FIR filter of equalizer

Step #	Description
1.	Define the problem space and set the boundaries i.e. inequality constraints of the tap weights defined by their maximum and minimum limits.
2.	Initialize an array of particles with random positions (tap weights of FIR filter) and their associated velocities inside the problem space.
3.	Check if the current position is inside the problem space or not. If not, adjust the positions so as to be inside the problem space.
4.	Evaluate the fitness value (MSE) of each particle according to (3).
5.	Compare the current fitness value with pbesti(k). If better, then assign the current value to it and assign current coordinate to $X_{id}(k+1)$.
6.	Determine the current global minimum among particle's best position according to (10).
7.	If, current global minimum is better than gbest, then assign it to gbesti (k) and the current coordinates to $X_{id}(k+1)$ using (9).
8.	Change the velocities according to (7).
9.	Move each particle to the new position according to (8) and return to step 3.
10.	Repeat step 3- step 9 until maximum number of iterations is achieved.

Convergence of PSO is faster, since we are defining additional parameters in the search space. It is not a global convergence guaranteed algorithm because the particle is restricted to a finite sampling space for each of the iterations. This restriction weakens the global search ability of the algorithm and may lead to premature convergence in many cases.

D. Bacterial Foraging Optimization

BFO is a derivative free optimization scheme in which foraging behaviour of E. Coli bacteria present in man's intestine is mimicked. Foraging refers to methods for locating, handling and ingesting food by the bacteria [15]. In order to do so, they undergo different stages such as chemotaxis, swarming, reproduction, elimination and dispersal.

Chemotaxis:

Movement patterns generated by the bacteria in presence of attractants and repellents are called chemotaxis. E. Coli adopts two different ways to move in search of nutrients- run or tumble. The bacterium sometime tumbles after a tumble or tumbles after a run i.e. alternates between these two modes of operation in its entire lifetime. Suppose position of each member in the population of S bacteria at the j^{th} chemotactic step, kth reproduction step and lth elimination is represented by $\theta^i(j,k,l)$.

The direction of movement after a tumble is defined by a unit length random direction $\Phi(j)$, which also denotes the cost function

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\Phi(j), \quad i = 1,2,\ldots\ldots S \quad (11)$$

where, $C(i)$ is the size of the step taken in the random direction specified by the tumble (run length time).

Swarming:

Swarming refers to the phenomenon when bacteria congregate into groups and move as concentric patterns of groups along the nutrient gradients created as a result of consumption by the group. The outward movement of the ring and local releases of the attractants decides the spatial order. The cells provide an attraction signal to each other so they swarm together.

Reproduction:

To keep the population constant, the bacteria with high health value splits into two bacteria which are placed in the same location, others die.

Elimination-Dispersal:

Some influence such as noxious substance or the increased temperature causes all the bacteria in a region to either die off or disperse to a new location in search of good food source. Elimination and dispersal are used to guarantee diversity of individuals and to strengthen the ability of global optimization.

Weights of FIR filter of equalizer are updated according to training rule [5] outlined in Table 4.

As the BFO scenario involves number of parameters for searching the total solution space, possibility of avoiding the local minima is higher. However, the performance decreases rapidly with an increase in the search space, in dealing with complex problems.

Table 4: BFO algorithm for estimating the weights of FIR filter

Step #	Description
1.	Initialization: S: Dimension of the search space. (No. of bacteria used) n: No. of input samples. p: No. of parameters to be optimized. NS: Swimming length. NC: No. of iterations to be undertaken in a chemotaxis loop. (NC>NS) N_{re}: No. of reproduction steps. N_{ed}: No. of elimination-dispersal events. P_{ed}: Probability with which the elimination and dispersal will continue. C(i): The size of the step taken in the random direction specified by the tumble. (i=1,2,…S) P(j,k,l): $P(j,k,l) = \{\Phi(j,k,l) \,/\, i = 1,2,\ldots S\}$. Location of each bacterium Φ is specified by random nos. [0,1].
2.	Generation of training signal: Binary input is delayed by 4 (half the order of equalizer) samples to act as the desired signal d(k).
3.	Iterative algorithms for weight updating: Assuming j=k=l=0 initially, this section models the bacterial population, chemotaxis, reproduction, elimination, and dispersal. (i)Elimination dispersal loop: $l = l + 1$ (ii)Reproduction loop: $k = k + 1$ (iii)Chemotaxis loop: $j = j + 1$ (a)Take a chemotactic step for every bacterium (i). (b)Compute $J(i,j,k,l)$ for each bacterium using (3), then let $J_{last} = J(i,j,k,l)$. (c)Tumble: Generate a random vector $\Phi(j)$ with each element in the range [1, -1]. (d)Move: Let $P^i(j+1,k,l) = P^i(j,k,l) + C(i) \times \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$ (12) This results in an adaptable step size in the direction of tumble. The cost function (MSE)$J(i,j+1,k,l)$ is computed. (e)Swim: Let $C = 0$ (Counter for swim length) While $C < N_S$, Let $C = C + 1$ and If$J(j) < J(j-1)$, then use (12) to compute new $J(i,j+1,k,l)$ ELSE Let $C = N_S$ (f)Go to the next bacterium $(i+1)$ if $i \neq S_b$ to process the next bacterium. If J(min) {minimum value of J among all the bacteria} is less than the tolerance limit, then break all loop.
4.	If $J < N_C$, go to step 3 i.e. continue chemotaxis loop.
5.	Reproduction: (a)Sort bacteria (i=1,2,…Sb) in ascending order of their health (higher cost means lower health) for given k and l. (b)Sr = Sb/2 bacteria with highest J value die and other Sr bacteria with the best value split and the copies that are made are placed at the same location as their parent.
6.	If$k < N_{re}$ go to step 2 to start the next generation in the chemotactic loop.
7.	Elimination-Dispersal: If elimination-dispersal probability is above a pre-set value P_{ed} for a bacterium, eliminate them. The total population is maintained constant in this way.

COMPARISION RESULTS AND DISCUSSIONS

Weights of FIR filter of channel equalizer were estimated using LMS, GA, PSO, BFO and WDO algorithm. Following linear and nonlinear channels were assumed as an example to compare the performance of these soft computing approaches under noisy condition (AWGN=10dB).

Linear channel:

The channel was modelled as a three tap FIR filter with transfer function $h(z) = 0.2600 + 0.9300z^{-1} + 0.2600z^{-2}$.

Nonlinear channel:

Nonlinear function represented by block NL in Fig. 1 is assumed to be tanh(a(k)), which approximately models the nonlinear channel.

All algorithms were initialized with the same population of real valued parameters and allowed to evolve. Each algorithm was tuned such that the population converged before the last iteration. The population sizes and algorithms were chosen to experimentally provide the best results for each case. Number of input samples and population size taken for each case was 500 and 60 respectively. Convergence characteristics were plotted for a maximum of 200 iterations in all cases. A step size of0.08 was assumed for LMS. Rest of the parameters are recorded in Table 5.

Table5: Parameters used for simulation of different algorithms

GA	PSO	BFO	WDO
No. of binary bits = 50 Crossover probability = 0.75 Mutation Probability =0.025	c1=1.5 c2= 1.5 v_{max}=1 ω=0.08	Sb=8 NS=3 NC=5 (NC>NS) N_{re}=30 P_{ed}=0.25 N_{ed}=10 C(i)=0.075	RT =2.6 g =0.1 α = 0.1 c = 0.4; u_{max} = 0.25 dimMin=-dim. dimMax= dim

Comparison of convergence characteristics of all five algorithms are summarized in Table 6-7 and plotted in Fig. 3-4.

Table 6: Comparative convergence data of FIR based channel equalizer at 10dBnoise

Algorithm	Linear channel		Nonlinear channel	
	NMSE (in dB)	No. of iterations	NMSE (in dB)	No. of iterations
LMS	-5.368	47	Convergence beyond 200	
GA	-7.403	14	-5.153	20
PSO	-9.658	15	-5.993	33
BFO	-9.960	28	-8.660	23
WDO	-8.053	26	-9.574	35

Table 7: Comparison of probability of error ($\times 10^{-6}$) at 15dB signal to noise ratio

Channel type	LMS	GA	PSO	BFO	WDO
Linear channel	3098	6982	880	150	400
Nonlinear channel	5717	13640	3630	8508	800

Careful observation reveals that BFO performs extremely well for a linear channel achieving NMSE of -9.96dB at 28 iterations as compared to -5.368dB at 47 iterations, -7.403dB at 14 iterations, -9.658dB at 15 iterations and -8.053dB at 26 iterations for LMS, GA, PSO and WDO respectively. Performance of PSO is close to BFO in terms of NMSE and better than BFO in terms of convergence time. The considered soft computing algorithms outperform the traditional LMS method. Although GA converges fastest among the five but its NMSE is poor suggesting that it might be a local minimum. WDO is a relatively new soft computing tool which is very efficient for equalizing nonlinear channel, as is evident from Fig. 3 and Fig. 4. It achieves minimum error of -9.574dB at 35 iterations as compared to -5.153dB at 20 iterations, -5.993dB at 33 iterations, -8.66dB at 23 iterations for GA, PSO and BFO respectively. LMS converges beyond 200 iterations.

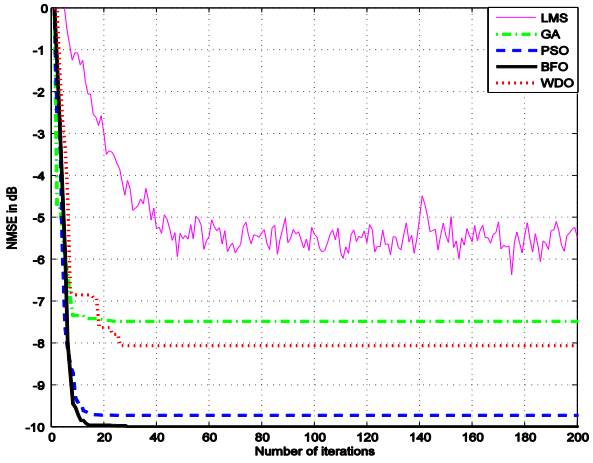

Figure 3: Convergence plot of linear channel at 10dB

BER performance comparison is shown in Table 7 and Fig. 5-6. BFO shows minimum BER of 150 bits in 10^6 samples for linear channel. In case of nonlinear channel, WDO again outperforms other algorithms with error of 800 bits as compared to 5717, 13640, 3630 and 8508 bits in 10^6 samples for LMS, GA, PSO and BFO respectively.

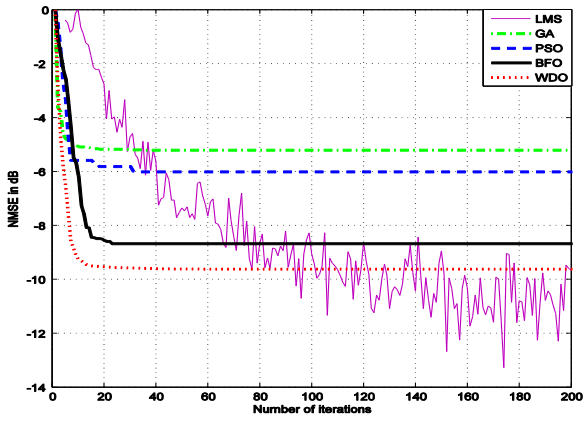

Figure 4: Convergence plot of non-linear channel at 10dB

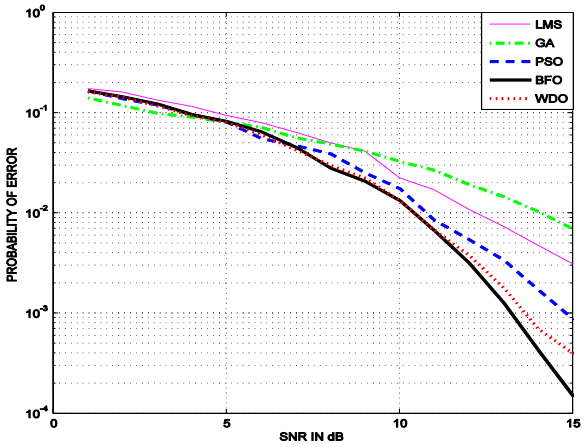

Figure 5: BER plot of linear channel at 10dB

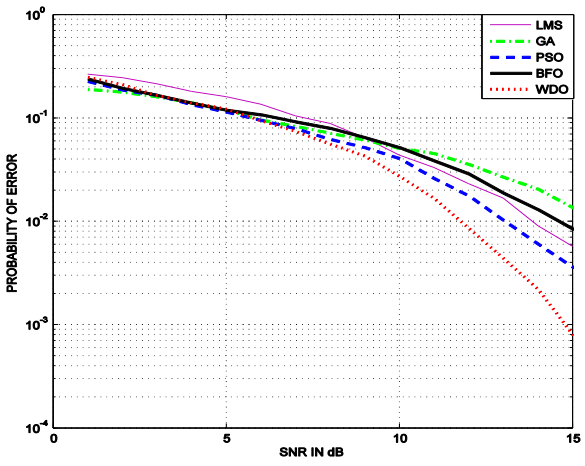

Figure 6: BER plot of non-linear channel at 10dB

CONCLUSION AND FUTURE SCOPE

Being motivated by the distinct advantage of each soft computing algorithm, this work presents the performance comparison of popular GA, PSO, BFO and WDO algorithms in estimating the weights of FIR filter in a channel equalizer. The benchmark adaptive algorithm LMS is also used to bring about the substantial improvement of soft algorithm, in general, over hard computing methods.

In engineering applications such as reliable data transmission in communication systems, the goal is to estimate the weights of equalizers installed at the front end of the receiver, as quickly as possible instead of good average potential solution. Future services demand high data rate and quality. Thus, it is necessary to define new and robust algorithms to estimate equalizer weights, so as to reduce the effect of noise in the communicated data. Population based soft computing algorithms such as these are envisioned to receive increasing attention due to its reliability and accuracy.

This work shows that WDO is, by far the most appropriate algorithm to estimate the weights of FIR based equalizer. Simulation results suggest that WDO achieves minimum error and best BER performance among the five considered algorithm. In future, WDO can be clubbed with other soft algorithms to further improve its performance such as developing a hybrid BFO-WDO methodology.

REFERENCES

[1] John G. Proakis, Intersymbol interference in digital communication systems, John Wiley & Sons, Inc., 2003.

[2] Simon S Haykin, Adaptive filter theory. Pearson Education India, 2008.

[3] Philip E Gill, Walter Murray and Margaret H. Wright, Practical optimization, Emarald, 1981.

[4] Richard E Bellman and Stuart E. Dreyfus, Applied dynamic programming, Princeton university press, 2015.

[5] Babita Majhi, G. Panda, and A. Choubey. "On the development of a new adaptive channel equalizer using bacterial foraging optimization technique,"2006 Annual IEEE India Conference. IEEE, 2006.

[6] Babita Majhi, G. Panda and A. Choubey. "On the development of a new adaptive channel equalizer using bacterial foraging optimization technique."2006 Annual IEEE India Conference. IEEE, 2006.

[7] Hocine Merabti, and Daniel Massicotte, "Nonlinear adaptive channel equalization using genetic algorithms," New Circuits and Systems Conference (NEWCAS), 2014 IEEE 12th International. IEEE, 2014.

[8] D. J., Krusienski and W. K. Jenkins. "Adaptive filtering

via particle swarm optimization." Signals, Systems and Computers, Conference Record of the Thirty-Seventh Asilomar Conference on. Vol. 1. IEEE, 2003.

[9] R. Sinha, A. Choubey, S. K. Mahto, "A Novel Efficient Transversal Channel Equalizer based on Nature Inspired Wind Driven Optimization (WDO) Technique," Kasmera Journal, vol. 44, no. 1, pp. 423-436, 2016.

[10] Satyasai Jagannath Nanda, Ganapati Panda, and Babita Majhi. "Development of Novel Digital Equalizers for noisy nonlinear channel using artificial Immune System." 2008 IEEE Region 10 and the Third International Conference on Industrial and Information Systems. IEEE, 2008.

[11] Zikri Bayraktar et al. "The wind driven optimization technique and its application in electromagnetics." IEEE transactions on antennas and propagation, vol. 61, no. 5, pp. 2745-2757, 2013.

[12] D. E. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Publishing Company, Inc., New York." (1989).

[13] Eberhart, Russ C., and James Kennedy. "A new optimizer using particle swarm theory." Proceedings of the sixth international symposium on micro machine and human science, vol. 1, 1995.

[14] Eberhart, Russ C., and Yuhui Shi. "Comparing inertia weights and constriction factors in particle swarm optimization." Evolutionary Computation, 2000. Proceedings of the 2000 Congress on. Vol. 1. IEEE, 2000.

[15] Kevin M Passino, "Biomimicry of bacterial foraging for distributed optimization and control." IEEE control systems, vol. 22, no. 3, pp. 52-67, 2002.