Journal of Engineering Science and Technology
Vol. 13, No. 8 (2018) 2271 - 2286

PERFORMANCE COMPARISON OF ADAPTIVE CHANNEL EQUALIZERS USING DIFFERENT VARIANTS OF DIFFERENTIAL EVOLUTION

SWATI SWAYAMSIDDHA[1,*], H. PAL THETHI[2]

[1]Kalinga Institute of Industrial Technology (KIIT), Bhubaneswar, Odisha, India
[2]Lovely Professional University, Jalandhar, Punjab, India
*Corresponding Author: swayamsiddhafet@kiit.ac.in

Abstract

In this paper, Differential Evolution (DE) based channel equalization is proposed and an in-depth comparison of the performance of different variants of DE is made. Adaptive equalization involves training of parameters such that the transmitted data is faithfully received. The equalization task is viewed as an optimization problem where the mean square error between the delayed transmitted signal and the equalizer output is minimized iteratively. In this paper, the equalizer coefficients are achieved using different variants of DE and the performance is compared in terms of convergence rate, optimality of solution and Bit Error Rate. Thus, the DE-based learning technique is an efficient method for adaptive nonlinear channel equalization.

Keywords: Differential evolution and its variants, Nonlinear channel equalization, Optimization.

1. Introduction

Channel equalization is a channel impairment improvement technique, which compensates for the signal distortion and noise caused due to multipath in time-dispersive channels. The channel equalization is an important aspect in high-speed digital communication required for efficient and reliable data recovery and reception when the data is transmitted over band-limited channel subjected to noise and interference. The digital data is fed into a channel, which can be modelled as an adaptive delay-tapped transversal filter having certain filter coefficients [1]. Due to band-limited, dispersive channel and multipath fading, the transmitted symbols overlap with each other and is distorted termed as inter-symbol interference (ISI).

In a wireless communication channel when the modulation bandwidth is exceeding the coherence bandwidth ISI takes place as the transmitted pulses are spread into the adjacent symbols [2, 3]. To combat the effects of ISI and noise and to reconstruct the signal and minimize Bit Error Rate (BER), the adaptive channel equalizer is used at the receiver end [4]. When the training is complete transfer function of the equalizer becomes inverse to that of the channel and the filter coefficients are adaptively optimized using adaptive optimization techniques so that the output of the equalizer (estimated signal) matches to that of the delayed version of the transmitted signal (desired signal) [5, 6]. Thus, adaptive channel equalization can be viewed as an iterative optimization problem where the objective is to minimize the mean-square error (MSE) such that an estimate of equalizer coefficients is obtained which nullifies the effects of ISI and noise on the signal transmitted through the channel.

Adaptive channel equalization is required as the wireless communication channels are unknown, non-stationary and time-varying channels. Since the adaptive channel equalizer compensates for the effects of the non-linear time-varying channel, a suitable adaptive optimization algorithm is to be applied for updating the equalizer coefficients and thus tracking the variations of the channel. In the recent past, the adaptive channel equalization is developed using soft computing approaches such as evolutionary and swarm intelligence algorithms compared to conventional learning techniques such as Least Mean Squares (LMS), Least Mean Fourth (LMF) and Recursive Least Squares (RLS) and their variants where there is possibility of solution being trapped by local optima. Moreover, there is performance degradation of gradient-based algorithms for non-linear channels [7-9]. The artificial neural network is employed for adaptive non-linear channel equalization [10-16] where the computationally efficient, single layered functional link artificial neural network is proposed and compared with multi-layer perceptron (MLP) and polynomial perceptron network (PPN).

The performance of the neural network based equalizer using a Genetic algorithm (GA) is studied in [17, 18] where the convergence speed is improved. Also, the neural network based equalizer is trained using swarm intelligence techniques such as Particle Swarm Optimization (PSO), Firefly Algorithm (FA) and their variants like hybrid GA-PSO algorithms [19, 20]. The Differential Evolution (DE) algorithm compared to gradient-based algorithms is studied in [21-23] where the performance of DE is shown to be superior in terms of convergence rate, quality of solution and the BER. Robust non-linear channel equalizers are developed based on Bacteria Foraging Optimization (BFO), which gives better performance in terms of optimality of the solution, convergence speed and BER computation [24, 25]. However, with an increase in search space and the complexity the convergence rate diminishes using BFO based training. Thus, a modified BFO called self-adaptation BFO (SA-BFO) is proposed for the design of an adaptive channel equalizer [26].

The SA-BFO algorithm strikes a balance between exploitation and exploration by adaptively changing the size of run length, hence giving good results. Also, Cat Swarm Optimization (CSO) is proposed for enhanced non-linear channel equalization where the optimal key parameters for the algorithm are determined [27]. The nonlinear time-varying channel equalization has been dealt with using fuzzy adaptive filters in [28]. Although DE-based channel equalizers and their superior performance compared to conventional optimization techniques and their counterparts such as BFO has been demonstrated in the literature, yet there exists a gap where channel equalizers based on different schemes of DE is studied. In this paper, adaptive channel equalization based on different variants of DE has been employed and a detailed comparison of performance has been established based on convergence plots and bit-error-rate. Such type of performance comparison has not been attempted so far.

The paper is organized as follows: Section 2 presents the non-linear adaptive channel equalizer model where the problem is formulated as an optimization problem. In section 3, the DE-based channel equalization is described where the different variants of DE are dealt. Section 4 gives the simulation results and discusses the performance of different variants of DE in terms of convergence rate and BER. In section 5, the conclusion is drawn and the relevant future research direction is mentioned.

2. Adaptive Channel Equalization Model

Channel equalization is a key area in a digital communication system where the objective is compensation for the channel distortion, which can be achieved by minimization of squared error between the equalizer output and the delayed version of the transmitted signal. The equalization in digital communication scenario is illustrated in Fig. 1, where $x(k)$ represents the symbol sequence transmitted through the non-linear channel. The Additive White Gaussian Noise (AWGN) is the channel noise contaminated to the channel output. This output of the channel acts as input to the adaptive non-linear equalizer. The output of the channel equalizer $y(k)$ is subtracted from the delayed version of the desired signal $d(k)$ to compute the error $e(k)$. The square of error $e^2(k)$ is considered as cost function, which is to be minimized such that the equalizer output matches with delayed transmitted source signal. The coefficients of the equalizer are iteratively updated using DE algorithm to achieve the best possible minimum squared error.

Digital communication channels are often modelled as low pass FIR filter. Figure 2 shows the digital channel as a 3-tapped delay filter whose output is associated with nonlinearities and noise, hence it is highly distorted. Therefore, to restore, the transmitted signal the output of the channel is passed as input to the equalizer, which is also modelled as an adaptive delay tapped filter.

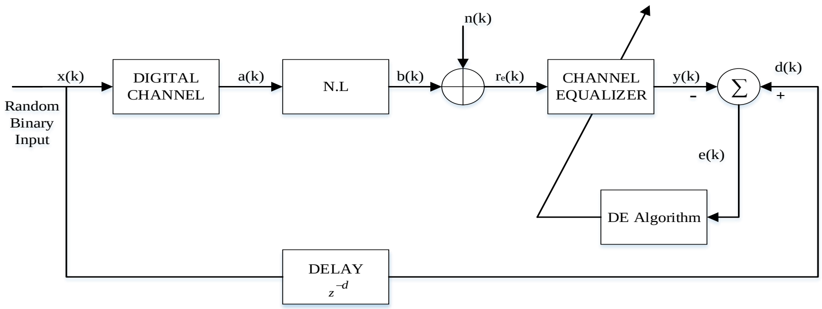

Fig. 1. Schematic block diagram of channel equalization.

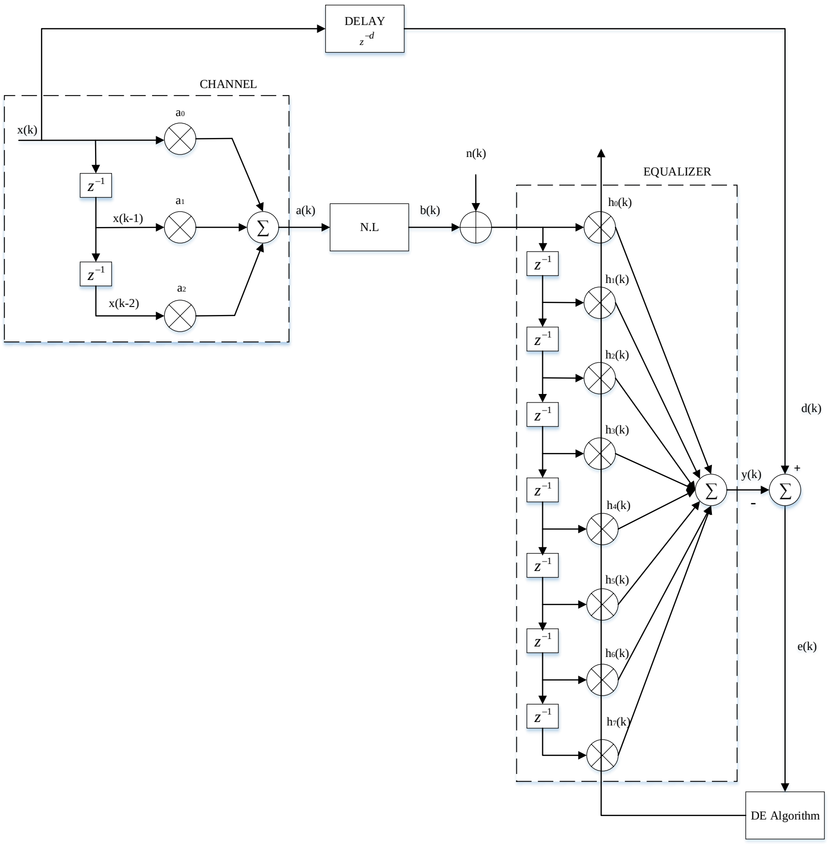

Fig. 2. The adaptive channel equalizer model.

Then, the delayed version of the input signal which is the desired signal, is compared with the output of the equalizer to evaluate the mean square error (MSE). The channel equalization problem is viewed as a minimization problem of the MSE using different variants of DE such that the estimated output of the equalizer progressively matches with the source input signal. The output of channel $r_e(k)$, which is fed as input to the equalizer, is given by Eq. (1) as:

$$r_e(k) = \sum_{i=0}^{k} a(i)x(k-i) + n(k) \tag{1}$$

where, $a(i)$=channel coefficients, $i = 0,1,2, \dots N - 1$ for N number of taps (N=3 for the given illustration) $x(k)$=binary input sample, and $n(k)$=AWGN noise. The output of the equalizer which is the estimated signal is given by Eq. (2):

$$y(k) = h^T(k) r_e(k) \tag{2}$$

where, $h(k) = [h(0), h(1), .., h(M - 1)]^T$ represents weight vector or filter coefficients of the equalizer having M number of delay taps which are adaptively updated by the optimization algorithm and $r_e(k) = [r_e(k), r_e(k - 1), \dots, r_e(k - M + 1)]^T$ is the input vector to the equalizer.The output of the non-linear channel is obtained by passing the above output signal $y(k)$ through a non-linear function. The desired signal is the delayed version of the source signal denoted as Eq. (3):

$$d(k) = x(k - m) \tag{3}$$

where m being the number of delays. The output of the equalizer, which is the estimated signal is expected to match with the desired signal and the difference between the two gives the error signal $e(k)$ represented in Eq. (4) as:

$$e(k) = d(k) - y(k) \tag{4}$$

The MSE is computed over S number of samples. This is the cost function which is to be minimized iteratively using the adaptive optimization algorithm is given by Eq. (5)

$$MSE = \frac{\sum_{k=0}^{S-1} e^2(k)}{S} \tag{5}$$

3. DE Based Channel Equalization

The general steps of Differential Evolution (DE) algorithm followed by adaptive equalization using DE is focussed below:

3.1. DE steps

DE algorithm proposed by Storn and Price [29] is an efficient population-based bio-inspired meta-heuristic and derivative-free optimization technique used for complex real-world engineering applications. This algorithm is very much similar to Genetic Algorithm (GA) except that the mutation precedes the crossover operation. The DE is known to have a faster convergence rate and the algorithms follow four steps: initialization, differential mutation, crossover and selection. The principal operators of this tool are population size, scaling factor and probability of crossover.

3.1.1. Initialization

The initialization step involves generation of NP number of initial parameter vector solutions. Each vector has p number of parameters and the lower and upper bounds of each parameter are fixed. Each of the parameters are random numbers within the specified range. The initial i^{th} vector for j^{th} parameter in generation g is denoted as $v_{i,j}(g)$ and given by Eq. (6) as follows:

$$v_{i,j}(g) = rand_j(0,1).(p_U - p_L) \tag{6}$$

where p_U and p_L are upper and lower bounds.

3.1.2. Differential mutation

Let us consider the first vector of the population as the target vector. With respect to this targetting vector, three random vectors $(v_{r1,j}, v_{r2,j}, v_{r3,j})$are chosen. Then the difference between the corresponding elements of the last two vectors is taken and each element of the difference vector is multiplied by scaling factor F. The resultant vector becomes the mutant vector of first target vector. This process is continued until the last number of population. Thus, for each number of target vector the corresponding mutant vectors $m_{i,j}(g+1)$ are generated. The equation used to generate the mutant vector is given Eq. (7):

$$m_{i,j}(g+1) = v_{r1,j}(g) + F.(v_{r2,j}(g) - v_{r3,j}(g)) \tag{7}$$

This variant of differential mutation is referred to as DE/rand/1. Based on different mutation strategies the other variants of DE are DE/rand/2, DE/best/1 and DE/best/2. The mutation operation carried out in various variants of DE are:

$$m_{i,j}(g+1) = v_{r5,j}(g) + F.(v_{r1,j}(g) + v_{r2,j}(g) - v_{r3,j}(g) - v_{r4,j}(g)) \tag{8}$$

$$m_{i,j}(g+1) = v_{best,j}(g) + F.(v_{r2,j}(g) - v_{r3,j}(g)) \tag{9}$$

$$m_{i,j}(g+1) = v_{best,j}(g) + F.(v_{r1,j}(g) + v_{r2,j}(g) - v_{r3,j}(g) - v_{r4,j}(g)) \tag{10}$$

where, $F$is a real constant ∈ [0, 2]. This control parameter amplifies the differential variation. $v_{best,j}$ is the best member of the population.

3.1.3. Crossover

The crossover operation involves the exchange of parameters between the initially chosen target vector and the mutant vector based on the probability of the crossover ratioCR. The resultant vector called the trial vector $u_{i,j}(g+1)$ is obtained as given in Eq. (11).

$$u_{i,j}(g+1) = \begin{cases} m_{i,j}(g+1)\ if\ rand(0,1) \le CR\ or\ j = j_{rand} \\ v_{i,j}(g) if\ rand(0,1) > CR\ or\ j \ne j_{rand} \end{cases} \tag{11}$$

The crossover ratio (CR) lies between 0 and 1 and decides the probability of parameters from a mutant vector that is to be copied to trial vector. A random number $rand(0,1)$ is generated and if its value is less than or equal to CR then the parameter from mutant vector is inherited to trial vector, otherwise, the trial vector takes the parameter from the target vector. This process is repeated for all pairs of target and mutant vectors.

3.1.4. Selection

The cost function is evaluated for the resultant trial vector $u_{i,j}(g)$. If the cost of trial vector is better compared to that of the target vector, then the trial vector survives and replaces the target vector in the next generation otherwise the target vector is retained for another generation. Mathematically, the expression for the process of selection is presented in Eq. (12):

$$v_{i,j}(g+1) = \begin{cases} u_{i,j}(g)\ if\ f(u_{i,j}) \le f(v_{i,j}) \\ v_{i,j}(g)\ otherwise \end{cases} \tag{12}$$

where, $f(u_{i,j})$ and $f(v_{i,j})$ represents the cost of trial and target vector respectively.

3.2. Channel equalization using DE

The channel equalization using DE is discussed through the following steps:

Step 1: The channel coefficients are initialized. Random binary input (k samples) is generated and passed through the channel.

Step 2: The output of the channel added with AWGN of certain *SNR* is passed through a nonlinear channel.

Step 3: The population of parameter vectors corresponding to equalizer coefficients are initialized randomly. First target vector is taken from NP number of vectors, which consists of p no. of parameters.

Step 4: The nonlinear channel output subject to noise and distortion is passed as input to the equalizer. Thus, the estimated output of the equalizer is computed.

Step 5: The delayed transmitted signal is considered as the desired signal.

Step 6: The difference between the estimated output of the channel equalizer and the desired signal gives the error signal. Thus, k no. of error signals are generated and the mean of the squared error gives the MSE and this process is repeated for NP no. of times.

Step 7: The mutation, crossover and fitness evaluation and selection processes are carried out (discussed in sub-section 3.1).

Step 8: The above steps are repeated iteratively until MSE decreases gradually. Once the MSE further ceases to decrease and attains the lowest level all the parameters become identical and the stopping criterion is met. At this stage, the final equalizer coefficients are obtained.

Table 1 and Fig. 3 illustrate the flow diagram and pseudo-code of nonlinear channel equalization using the DE algorithm respectively.

Table 1. Pseudo-code of DE algorithm for adaptive nonlinear channel equalization.

1:	Generate random binary input x(k) and;
2:	Compute the output of channel A(z)
3:	Population initialization: $v_{i,j}(g)$ parameter vector of equalizer
4:	Evaluate the cost function MSE for each individual solution
5:	While (stopping condition not satisfied) {
6:	Choose a target vector
7:	Randomly select two vectors $v_{r1,j}(g)$ and $v_{r2,j}(g)$
8:	Compute weighted difference vector $F(v_{r1,j}(g) - v_{r2,j}(g))$
9:	Mutation: Evaluate mutant vector $m_{i,j}(g+1)$ for different schemes of DE
10:	Crossover: Evaluate trial vector by computing parameters from mutant vector and target vector based on probability of CR
11:	Fitness Evaluation and Selection: Select target vector or trial vector, the one with lower cost survives for the next generation}

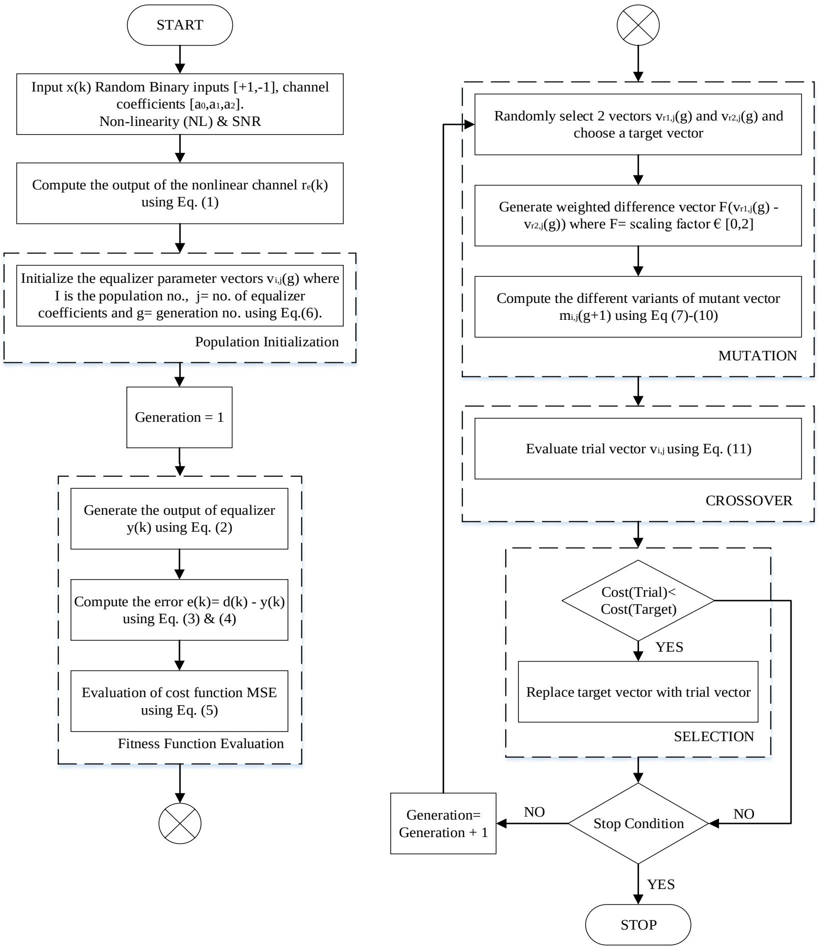

Fig. 3. Flow-diagram of DE algorithm for adaptive nonlinear channel equalization.

4. Results and Discussion

The simulation results are obtained for the DE-based channel equalization problem where two different linear channels are considered for the simulation purpose as given in Eq. (13):

$$A1 = 0.2600 + 0.9300z^{-1} + 0.2600z^{-2}$$

$$A2 = 0.3482 + 0.8769z^{-1} + 0.3482z^{-2} \quad (13)$$

In order to simulate the non-linear condition, the output of the linear channel is passed through three types of non-linearity functions given in Eq. (14)

$$NLF1 = \tanh\big(y(k)\big)$$

$$NLF2 = y(k) + 0.2y^2(k) - 0.1y^3(k)$$

$$NLF3 = y(k) + 0.2y^2(k) - 0.1y^3(k) + 0.5\cos(\pi y(k)) \quad (14)$$

The AWGN noise of 30 dB is added to the channel output which serves as the input to the adaptive channel equalizer.

The typical values of key parameters of DE algorithm used in the computer simulation study are population size NP=40, Scaling factor F=0.9, Cross-over ratio CR=0.9. The number of input samples $k = 100$ and the number of iterations $N1$=100.

The convergence characteristics of the MSE and BER plot using different variants of DE are presented in Figs. 4-6 for channel 1 corresponding to three different nonlinearities. Similarly, Figs. 7-9 presents the learning curves for the second channel using the same three nonlinear channels.

The BER is plotted for different variants of DE based equalization using channel 1 corresponding to three nonlinearities given in Figs. 10-12. Similarly, for channel 2 the BER plot is presented in Figs. 13-15 for four variants of DE corresponding to Eqs. (7)-(10).

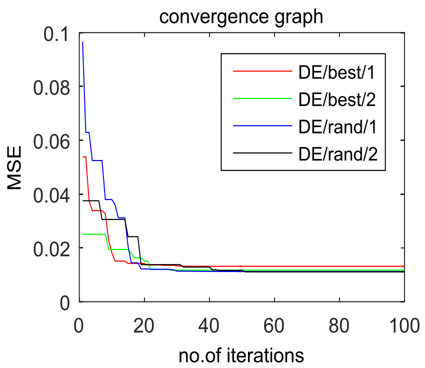

Fig. 4. Learning curves for CH-1, NLF-1.

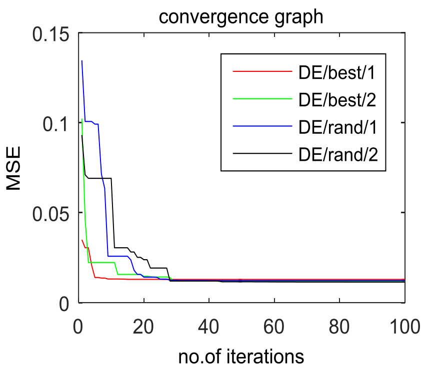

Fig. 5. Learning curves for CH-1, NLF-2.

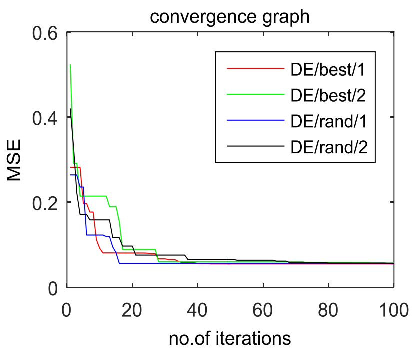

Fig. 6. Learning curves for CH-1, NLF-3.

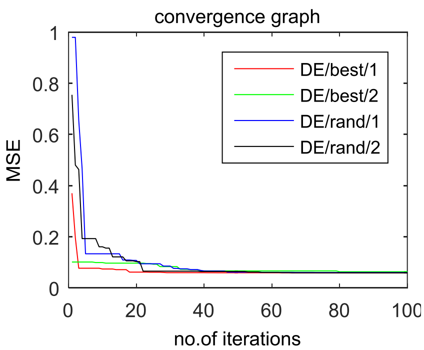

Fig. 7. Learning curves for CH-2, NLF-1.

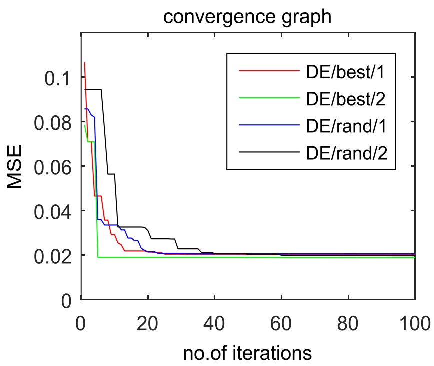

Fig. 8. Learning curves for CH-2, NLF-2.

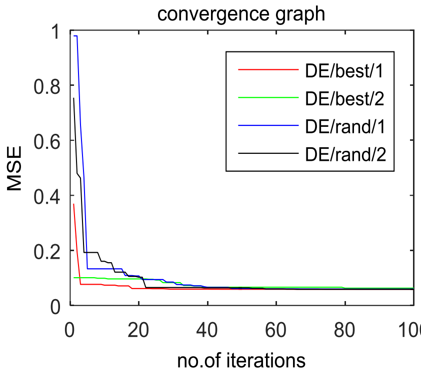

Fig. 9. Learning curves for CH-2, NLF-3.

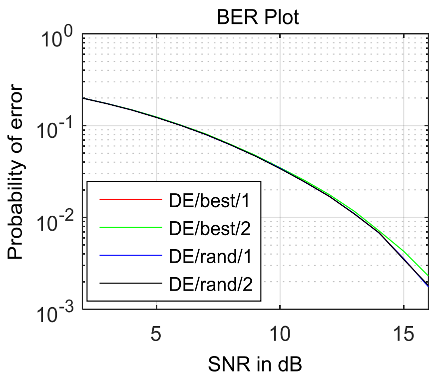

Fig. 10. BER vs. SNR plot for CH-1, NLF-1.

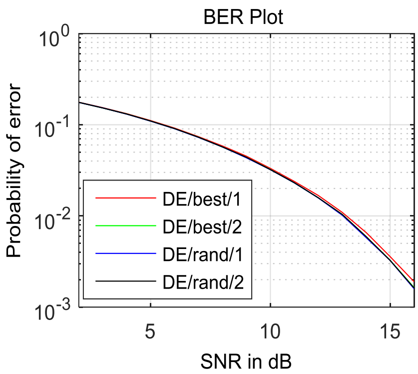

Fig. 11. BER vs. SNR plot for CH-1, NLF-2.

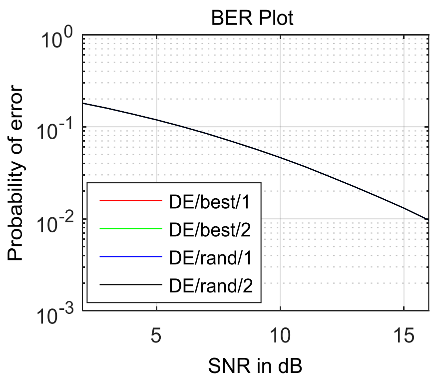

Fig. 12. BER vs. SNR plot for CH-1, NLF-3.

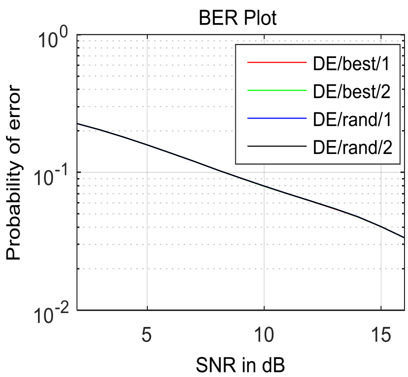

Fig. 13. BER vs. SNR plot for CH-2, NLF-1.

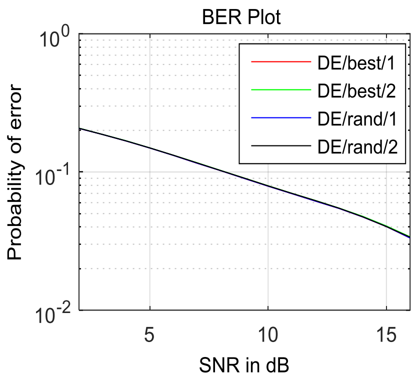

Fig. 14. BER vs. SNR plot for CH-2, NLF-2.

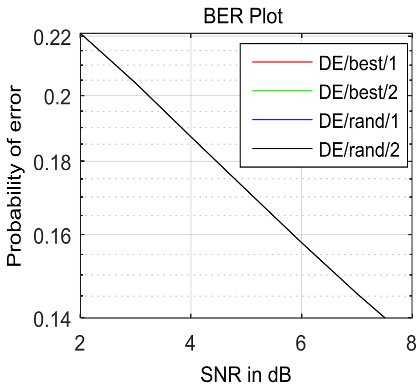

Fig. 15. BER vs. SNR plot for CH-2, NLF-3.

The minimum MSE (MMSE) attained at the convergence using different variants of DE are shown in Table 2. From the MMSE values, it is evident that the DE/Rand/2 performs the best in terms of providing the least MSE as 0.011089(NL-1), 0.011422(NL-2), 0.055107(NL-3) for channel 1 and 0.019072(NL-1), 0.018898(NL-2), 0.058644(NL-3) for channel 2. Based on MMSE the order of various variants based equalizers are DE/rand/2 < DE/rand/1 < DE/best/1 < DE/best/2. Also, it is observed that as the nonlinearity present in the channel becomes mild to severe the MMSE accordingly increases.

Table 2. Convergence MSE Value attained using different variants of DE.

Minimum MSE performance		**NL-1**	**NL-2**	**NL-3**
CH-1	**DE/Best/1**	0.011712	0.012271	0.056235
	DE/Best/2	0.013164	0.012884	0.057641
	DE/Rand/1	0.011207	0.011882	0.05619
	DE/Rand/2	0.011089	0.011422	0.055107
CH-2	**DE/Best/1**	0.020309	0.019821	0.058919
	DE/Best/2	0.020620	0.021433	0.059817
	DE/Rand/1	0.019709	0.019543	0.059556
	DE/Rand/2	0.018898	0.019072	0.058644

A comparative performance analysis is summarized for four schemes of DE corresponding to channel 1 and channel 2 in Table 3. In terms of convergence rate, the DE/best/2 converges faster compared to others whereas in terms of MMSE it performs worst compared to other variants. The DE/rand/1 yields the least BER. There is not much difference in terms of BER performance using different schemes of DE. From the BER plots is seen that as the SNR increases the probability of error decreases. The DE/rand/2 performs well in terms of MMSE compared to other schemes because the trial vector is obtained using two difference vectors multiplied with the scaling factor compared to only one difference vector in DE/rand/1 scheme. Whereas, the convergence rate is fastest for DE/best/2 as trial vector is obtained by adding the scaled difference vectors to the vector having best fitness value in that generation.

Further, the proposed DE-based channel equalizer (DE/rand/1) is compared with that of existing BFO based equalizer model [24-26]. Figures 16 and 17 show the bit-error-rate plots taking the above channels (nonlinearity NLF1) into consideration, which shows that DE-based channel equalizer performs better as compared to that of BFO.

Table 3. Comparison analysis for CH-1.

Performance criteria	Fastest convergence	Least MSE	Least BER
NL 1	DE/best/2	DE/rand/2	DE/rand/1
NL 2	DE/best/2	DE/rand/2	DE/rand/1
NL 3	DE/best/2	DE/rand/2	Same for all 4 variants

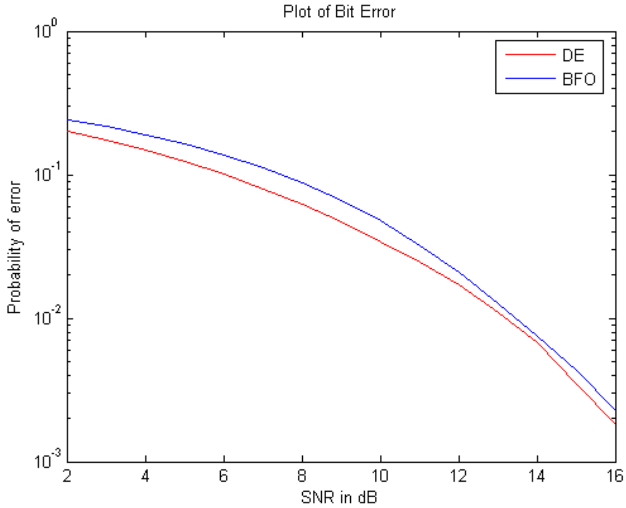

Fig. 16. BER vs SNR CH1 for DE and BFO.

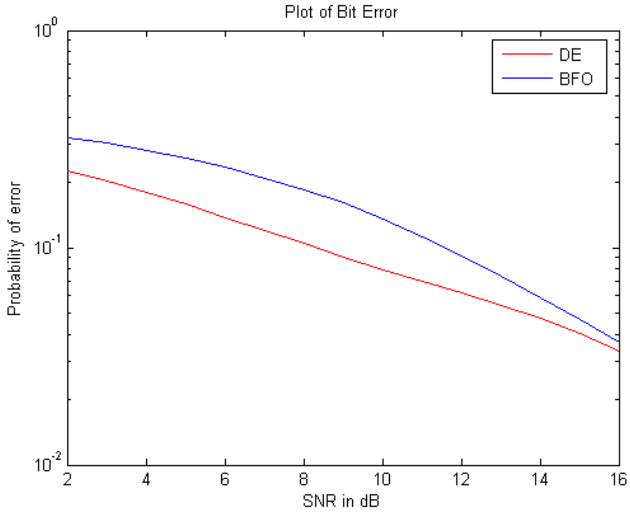

Fig. 17. BER vs SNR CH2 for DE and BFO.

5. Conclusions

The data transmitted through a band limited communication channel suffers from linear, nonlinear and additive distortions. Equalization compensates for this ISI caused by multipath within time-dispersive channels. The DE-based adaptive non-linear channel equalization is modelled as an iterative optimization problem where the weights of the equalizer are adaptively tuned by different DEs to recover the source signal transmitted through the channel. The results of variants of DE are evaluated in terms of convergence speed, optimality of the solution and BER plots. The DE algorithm, in general, performs well for the recovery of the transmitted signals during training. The convergence rate is faster and this algorithm updates the equalizer weights to best possible values and gives satisfactory MSE during training. Thus, the learning capability of different variants of DE is studied and compared for different channel conditions and nonlinearities, which shows that the DE algorithm performs efficiently for nonlinear adaptive channel equalization tasks. This work can further be extended by applying newer and hybrid optimization algorithms for training equalizer parameters such as self-adaptive DE [30, 31], etc. This optimization principle can also be applied to fading and recursive channels.

Nomenclatures

a	Channel coefficients
d	Delayed signal
F	Scaling factor
e	Error signal
f	Cost function
g	Generation
h	Equalizer filter coefficients
k	k^{th} sample
M	Delay taps
m	No. of delays
$m_{i,j}$	i^{th} mutant vector for j^{th} parameter
N	No. of taps
NP	No. of initial parameter vector solutions
n	AWGN noise
p	No. of parameters
p_L	Lower bound
p_U	Upper bound
r_e	Equalizer input
S	Total no. of samples
$u_{i,j}$	i^{th} Trial vector for j^{th} parameter
$v_{i,j}$	i^{th} initial vector for j^{th} parameter
x	Transmitted symbol sequence
y	Channel equalizer output
z^{-1}	Delay element

Abbreviations

AWGN	Additive White Gaussian Noise
BER	Bit Error Rate

BFO	Bacteria Foraging Optimization
CR	Crossover Ratio
CSO	Cat Swarm Optimization
DE	Differential Evolution
FA	Firefly Algorithm
GA	Genetic algorithm
ISI	Inter-Symbol Interference
LMF	Least Mean Fourth
LMS	Least Mean Square
MLP	Multi-layer Perceptron
MMSE	Minimum MSE
MSE	Mean Square Error
NL	Non-linearity
PPN	Polynomial Perceptron Network
PSO	Particle Swarm Optimization
RLS	Recursive Least Squares
SA-BFO	Self-adaptation BFO
SNR	Signal to Noise Ratio

References

1. Widrow, B.; and Stearns, S.D. (1985). *Adaptive signal processing*. Pearson Education Incorporation.
2. Proakis, J.G.; and Salehi, M. (2008). *Digital communications (5th ed.)*. New York: McGraw Hill.
3. Goldsmith, Andrea. (2005). *Wireless communications*. Camridge: Cambridge University Press.
4. Sun, H.; Mathew, G.; and Farhang-Boroujeny, B. (2005). Detection techniques for high-density magnetic recording. *IEEE Transactions on Magnetics*, 41(3), 1193-1199.
5. Gibson, G.J.; Siu, S.; and Cowan, C.F.N. (1991). The application of nonlinear structures to reconstruction of binary signals. *IEEE Transactions on Signal Processing*, 39(8), 1877-1884.
6. Lucky, R.W. (1965). Automatic equalization for digital communication. *The Bell System Technical Journal*, 44(4), 547-588.
7. Reuter, M.; and Zeidler, J.R. (1999). Nonlinear effects in LMS adaptive equalizers. *IEEE Transactions on Signal Processing*, 47(16), 1570-1579.
8. Kumar, S.J.R.; Vaddadi, M.S.B.S.; and Penumala, S.K. (2008). Biologically inspired evolutionary computing tools for channel equalization. *Proceedings of the International Conference on Electronic Design.* Penang, Malaysia, 1-6.
9. Faiz, M.M.U; and Zerguine, A. (2011). Adaptive channel equalization using the sign regressor least mean fourth algorithm. *Proceedings of the Saudi International Electronics, Communications and Photonics Conference*. Riyadh, Saudi Arabia, 1-4.
10. Patra, J.C.; Pal, R.N.; Baliarsingh, R.; and Panda, G. (1999). Nonlinear channel equalization for QAM signal constellation using artificial neural networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 29(2), 262-271.

11. Chang, C-H.; Siu, S.; and Wei, C-H. (1995). A polynomial-perceptron based decision feedback equalizer with a robust learning algorithm. *Signal Processing*, 47(2), 145-158.
12. Chen, S.; Mulgrew, B.; and Grant, P.M. (1993). A clustering technique for digital communications channel equalization using radial basis function networks. *IEEE Transactions on Neural Networks*, 4(4), 570-590.
13. Chen, S.; Gibson, G.J.; Cowan, C.F.; and Grant, P.M. (1990). Adaptive equalization of finite non-linear channels using multiplayer perceptrons. *Signal Processing*, 20(2), 107-119.
14. Chen, S.; and Mulgrew, B. (1992). Overcoming co-channel interference using an adaptive radial basis function equalizer. *Signal Processing*, 28(1), 91-107.
15. Zerdoumi, Z.; Chikouche, D.; and Benatia, D. (2013). Adaptive decision feedback equalizer based neural network for nonlinear channels. *Proceedings of the 3rd International Conference on Systems and Control*. Algiers, Algeria, 850-855.
16. Guha, D.R.; and Patra. S.K. (2009). Channel equalization for ISI channels using wilcoxon generalized RBF. *Proceedings of the Fourth International Conference on Industrial and Information Systems*. Sri Lanka, 133-136.
17. Mota, T.A.; Leal, J.F.; and de Casto Lima, A.C. (2015). Neural equalizer performance evaluation using genetic algorithm. *IEEE Latin America Transactions*, 13(10), 3439-3446.
18. Merabti, H.; and Massicotte, D. (2014). Nonlinear adaptive channel equalization using genetic algorithms. *Proceedings of the IEEE 12th International New Circuits and Systems Conference (NEWCAS)*. Trois-Rivieres, QC, Canada, 209-212.
19. Sarangi, A.; Priyadarshini, S.; and Sarangi, S.K. (2016). A MLP equalizer trained by variable step size firefly algorithm for channel equalization. *Proceedings of the IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)*. Delhi, India, 1-5.
20. Yogi, S.; Subhashini, K.R.; and Satapathy, J.K. (2010). A PSO functional link artificial neural network training algorithm for equalization of digital communication channels. *Proceedings of the 5th International Conference on Industrial and Information Systems*. Mangalore, India, 107-112.
21. Wu, Z.; Huang, H.; Zhang, X.; Yang, B.; and Dong, H. (2008). Adaptive equalization using differential evolution. *Proceedings of the IEEE Congress on Evolutionary Computation.* Hong Kong, China, 1962-1967.
22. Thethi, H.P.; and Swayamsiddha, S. (2013). Performance analysis of adaptive channel equalizer using population based update algorithms. *International Journal of Computer Applications*, 74(12), 41-46.
23. Khuntia, P.K.; Sahu, B.; and Mohanty, C.S. (2012). Development of adaptive channel equalization using DE. *Proceedings of the World Congress on Information and Communication Technologies*. Trivandrum, India, 419-424.
24. Majhi, B.; and Panda, G. (2006). Recovery of digital information using bacterial foraging optimization based nonlinear channel equalizers. *Proceedings of the International Conference on Digital Information Management*. Bangalore, India, 367-372.

25. Majhi, B.; and Panda, G. (2006). On the development of a new adaptive channel equalizer using bacterial foraging optimization technique. *Proceedings of the IEEE India Conference*. New Delhi, India, 1-6.
26. Su, T-J.; Cheng, J-C.; and Yu, C.-J. (2010). An adaptive channel equalizer using self-adaptation bacterial foraging optimization. *Optics Communications*, 283(20), 3911-3916.
27. Diana, D.C.; and Joy Vasantha Rani, S.P. (2017). Novel cat swarm optimization algorithm to enhance channel equalization. *The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, 36(1), 350-363.
28. Liang, Q.; and Mendel, J.M. (2000). Equalization of nonlinear time-varying channels using type-2 fuzzy adaptive filters. *IEEE Transactions on Fuzzy Systems*, 8(5), 551-563.
29. Storn, R.; and Price, K. (1997). Differential Evolution - A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization,* 11(4), 341-359.
30. Qin, A.K.; Huang, V.L.; and Suganthan, P.N. (2009). Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Transactions on Evolutionary Computation*, 13(2), 398-417.
31. Wang, Y.; Cai, Z.; and Zhang, Q. (2011). Differential evolution with composite trial vector generation strategies and control parameters. *IEEE Transactions on Evolutionary Computation*, 15(1), 55-66.