

Chapter 12

Modified PSO-Based Equalizers for Channel Equalization

D.C. Diana and S.P. Joy Vasantha Rani

12.1 Introduction

Adaptive equalization [1] plays an important role in the high-speed digital transmission to remove and recover the problem of inter-symbol interference (ISI). The adaptive algorithms [2] such as steepest descent, least mean square (LMS), recursive least square (RLS), affine projection algorithm (APA), and their variants [3] reported in literature have the chance of getting trapped in local minima [4–6] while optimizing the equalizer weights. The performance of these algorithms further degraded in nonlinear channel conditions [6]. To overcome these problems, different derivative-free optimization algorithms are proposed, whereas PSO is one among them. For solving optimization algorithms, PSO is proven as an efficient method and was applied successfully in the area of adaptive equalization [6]. PSO stays as one of the best algorithms for channel equalization in the recent years [5–7]. And also it provides minimum mean square error (MSE) compared to genetic algorithms used in the channel equalization [6].

From the first introduction of PSO [8], several variants [9–16] are proposed. The inertia weight parameter ‘ w ’ is the first modification found in literature which plays a major role in convergence and improves the simulation time. Initially, Shi and Eberhart introduced inertia weight [10]. In their work, a range of constant w values are used and found that PSO shows a weak exploration for large w values, i.e., $w > 1.2$, and it tends to trap in local optima with $w < 0.8$. When w is in the range [0.8, 1.2], PSO shows the global optimum in least average

D.C. Diana (✉) · S.P. Joy Vasantha Rani
Department of Electronics Engineering, Madras Institute of Technology,
Anna University, Chennai, India
e-mail: diana_ece@yahoo.co.in

S.P. Joy Vasantha Rani
e-mail: joy_mit@annauniv.edu

number of iterations. A random value is selected to track the optima in [11], as given in Eq. (1.1).

$$w = 0.5 + \frac{\text{rand}()}{2} \quad (12.1)$$

where $\text{rand}()$ is a random value between $[0, 1]$.

Time-varying inertia weight is a common approach used in PSO, which determines the inertia weight based on the current and total iteration. Time-decreasing inertia weight methods are used in literature to improve the convergence rate. A linearly decreasing inertia weight has been used for adaptive equalization in [6], based on the update law:

$$w_n = [(w_i - w_f)(m - n)/(m - 1)] + w_f \quad (12.2)$$

where w_i is the initial weight, w_f the is maximum weight, 'm' is the maximum iteration value, and 'n' is the current iteration index. A nonlinear decreasing inertia weight is proposed by Chatterjee and Siarry [12] based on the equation:

$$w_n = [(w_i - w_f)(m - n)^{np}/(m - 1)^{np}] + w_f \quad (12.3)$$

where np is the nonlinear modulation index. With different values of np , inertia weight gives different variations from w_i to w_f .

Feng et al. [13, 14] proposed another modification as given in Eq. (1.4).

$$w_n = [(w_i - w_f)(m - n)/m] + w_f * z \quad (12.4)$$

where $z = 4z(1 - z)$. Initially, z is the random value between $(0, 1)$.

Lei et al. [15] choose a fuzzy complement function as inertia weight defined by Eq. (1.5).

$$w = \frac{1 - \beta}{1 - s\beta} \quad (12.5)$$

where β is n/m and s is greater than -1 .

The other nonlinear approach [16] is given as:

$$w = \left(\frac{2}{n}\right)^{0.3} \quad (12.6)$$

Zheng et al. [17] proposed an increasing inertia weight. Similarly, Jiao et al. [18] proposed another nonlinear increasing inertia weight as in Eq. (1.7),

$$w = w_{\text{initial}} \times u^n \quad (12.7)$$

Here w_{initial} is usually between $[0, 1]$ and u is $[1.0001, 1.005]$. As per [18], u is set as 1.0002.

This paper analyzes the above existing modifications and proposes novel time-varying inertia weight methods for adaptive equalization with minimum mean square error. The following section describes the methodology used in adaptive equalization. Section 12.3 explains the structure and training of the equalizer filter using PSO. Section 12.4 discusses the results.

12.2 Methodology

Figure 12.1 depicts a basic block diagram used in adaptive equalization [1].

The equalizer input is the convolution sum of the random input Bernoulli sequence $\{x(n)\} = \pm 1$ (with zero mean and variance 1) and channel model. The channel output is added with the random additive white Gaussian noise (AWGN). The noise sequence has zero mean and variance 0.001.

Initially, the channel model is assumed as a linear channel [2] with three paths as given in Eq. (12.8).

$$\begin{aligned} h_n &= \frac{1}{2} \left[1 + \cos\left(\frac{2\pi}{W}(n-2)\right) \right], & n = 1, 2, 3 \\ h_n &= 0 & \text{otherwise} \end{aligned} \tag{12.8}$$

The factor W controls the amount of distortion. The effect of amplitude distortion is analyzed and given in Table 12.3. The effect of nonlinearities generated in the transmitter is modeled as three different nonlinear equations in (12.9), (12.10), and (12.11). The nonlinearity is introduced by relating output $y(n)$ and input $x(n)$ as in [19],

$$\text{Channel 1: } y(n) = \tanh(x(n)) \tag{12.9}$$

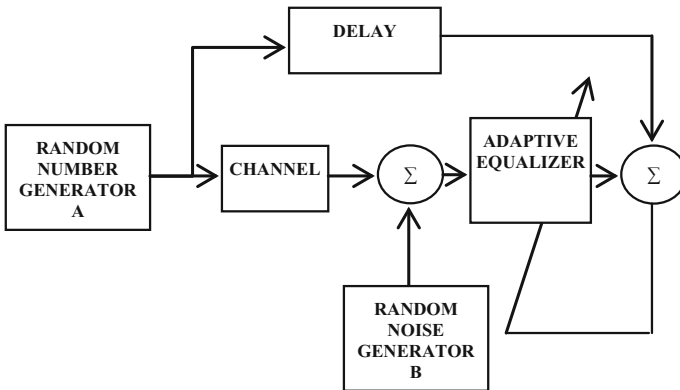


Fig. 12.1 Block diagram of digital communication system

$$\text{Channel 2: } y(n) = x(n) + 0.2x^2(n) - 0.1x^3(n) \quad (12.10)$$

$$\text{Channel 3: } y(n) = x(n) + 0.2x^2(n) - 0.1x^3(n) + 0.5 \cos(\pi x(n)) \quad (12.11)$$

The nonlinear effect is introduced in the input $x(n)$ and it is denoted as $y(n)$. The channel output is represented as

$$r(n) = d_{\text{distorted}}(n) + v_n$$

Here $d_{\text{distorted}}(n)$ is the distorted version of the desired signal. The distortion is introduced by applying the input signal through any one of the nonlinear equations (Eqs. (12.9), (12.10), and (12.11)) and then convolved with the linear channel in Eq. (12.8).

$$d_{\text{distorted}}(n) = y(n) * h_n$$

The signal can be distorted by bandwidth limitation, multipath effect, and the nonlinearities introduced in the transmitter. v_n is the noise component modeled as white Gaussian noise with variance σ_n^2 .

The error $e(n)$ is calculated as

$$e(n) = d(n) - r(n)$$

The adaptive algorithm updates the equalizer weights iteratively to minimize $e^2(n)$. Since $e^2(n)$ is positive and gives the instantaneous power, selected as cost (fitness) function.

12.3 System model

The system model [1] used for equalization is the simple linear transversal equalizer as in Fig. 12.2 and decision feedback equalizer shown in Fig. 12.3.

12.3.1 Linear Transversal Equalizer (LTE)

In this type of structure, the present and old received samples $r(t-kT)$ are appropriately weighted by the coefficients c_q and added to generate the output. The weights are trained to optimum value using adaptive algorithms. The output Z_k becomes

$$Z_k = \sum_{q=0}^{N-1} c_q r(t - qT) \quad (12.12)$$

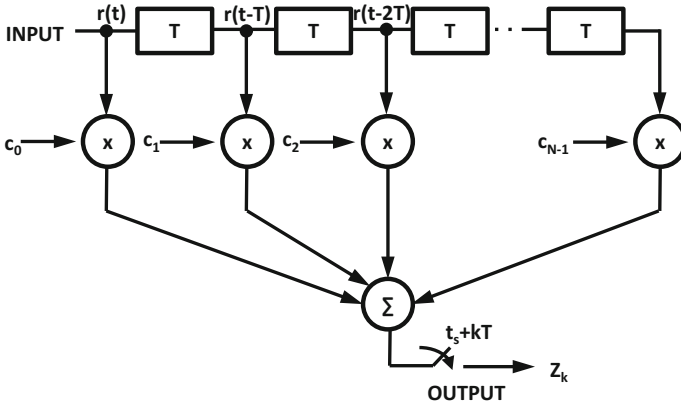


Fig. 12.2 Linear transversal equalizer structure

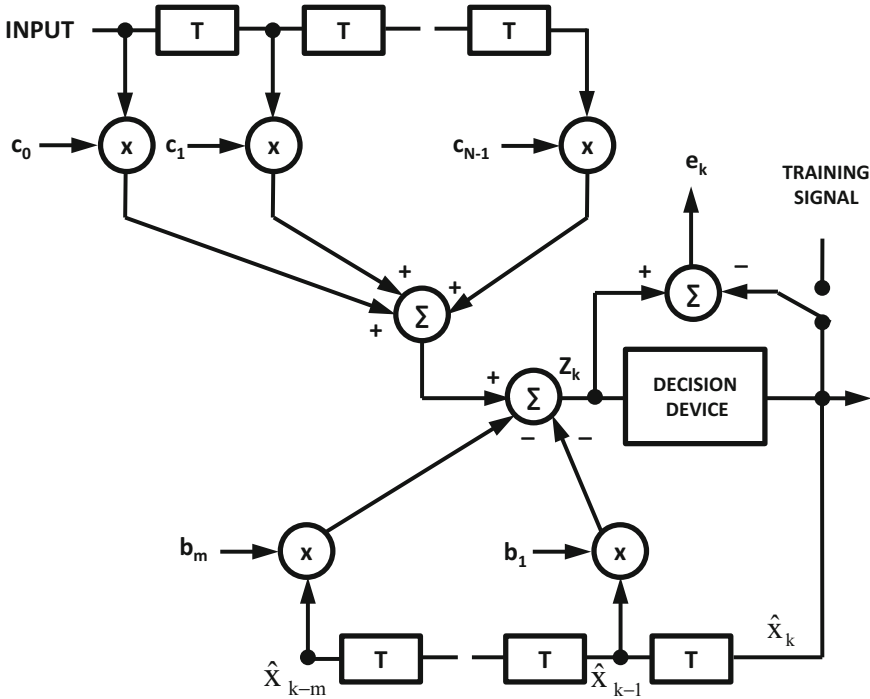


Fig. 12.3 Decision feedback equalizer structure

12.3.2 Decision Feedback Equalizer (DFE)

This type of nonlinear equalizer uses a forward and feedback filter. The forward and feedback filter outputs are summed to find the output of the equalizer. The forward filter output is sent back via the feedback filter. The ISI is canceled by deducting past symbol values from the equalizer output. The output of DFE is calculated as in Eq. (12.13)

$$Z_k = \sum_{q=0}^{N-1} c_q r(t - qT) + \sum_{i=1}^m b_i X_{k-m} \quad (12.13)$$

12.3.3 Basic PSO

PSO [8] starts by initializing particles in the random search space and then considers the social and cognitive behavior of the particles. The candidate solutions called ‘particle’ move around the n -dimensional search space with a velocity, which is adjusted based on its own experience and its neighbors’ experiences. The own experience of a particle is denoted as P_{best} ; all the particles’ experience is denoted as G_{best} . A new G_{best} is generated in each update process, making other particles to fly toward G_{best} . The new velocity and position are calculated as:

$$V_i(t+1) = w * V_i(t) + ac_1 * \text{rand}_1 * (P_{\text{best}} - X_i) + ac_2 * \text{rand}_2 * (G_{\text{best}} - X_i) \quad (12.14)$$

$$c_i(t+1) = c_i(t) + V_i(t+1) \quad (12.15)$$

where ac_1 and ac_2 are positive constants, called as cognitive and social acceleration coefficients, respectively. The two random functions rand_1 and rand_2 are in the interval $[0, 1]$. $V_i(t)$ and $c_i(t)$ are velocity and position of particle i , respectively, in t th iteration. Location of the best solution (best tap weights) for particle i is P_{best} and G_{best} represents the best solution among all particles. In Eq. (12.14), w is the inertia weight, which controls the local search and global search.

12.3.4 Training by PSO

The PSO-based equalizer [6] is optimized the tap weights based on the following steps:

For LTE:

- T number of tap weights is assigned for equalizer.
- ‘ ws ’ samples of data are passed from channel output (distorted signal) to equalizer which generates ws numbers of estimated samples.
- Error is estimated by comparing delayed version of each input sample with equalizer output
- The mean square error function of each particle P is

$$\text{MSE}(P) = \frac{\sum_{i=1}^K e_i^2}{ws}$$

- Fitness value $\text{MSE}(P)$ is minimized using PSO-based optimization.
- If the MSE of a particle is less than its previous value, then term it as current local best value and its corresponding weight values as P_{best}
- The minimum of MSE of all particles in every iteration is taken as global best value.
- If the global best value is better than the previous one, select the corresponding tap weights to G_{best} .
- Calculated the change in position (Tap weights) of each particle using Eq. (12.14)
- Moved each particle (Tap weights) c_k in Eq. (12.12) to new position by Eq. (12.14).
- Repeated the above steps for the number of iterations specified or stopped when the algorithm converges to an optimum value with least MSE value.

For DFE:

- The coefficients are initialized randomly for forward and feedback filter.
- In the first iteration, only forward filter is active and after calculating the error, the output of the forward filter is fed back through feedback filter.
- The mean square error is calculated by subtracting the output of forward and feedback filters.
- The forward and feedback filter coefficients c_k and b_i in Eq. (12.13) are updated based on Eqs. (12.14) and (12.15).

12.3.5 Proposed Strategies

In PSO, higher values of inertia weight enhance global search while smaller values improve local search. Generally, the inertia weight decreases linearly from 1 to 0 over the entire iterations. The inertia value is high at first, which allows all particles to move freely in the search space during the initial steps, while it decreases over time in the later steps. The later search requires a low inertia value to gradually shift the search from global search to local search. The speed of convergence and

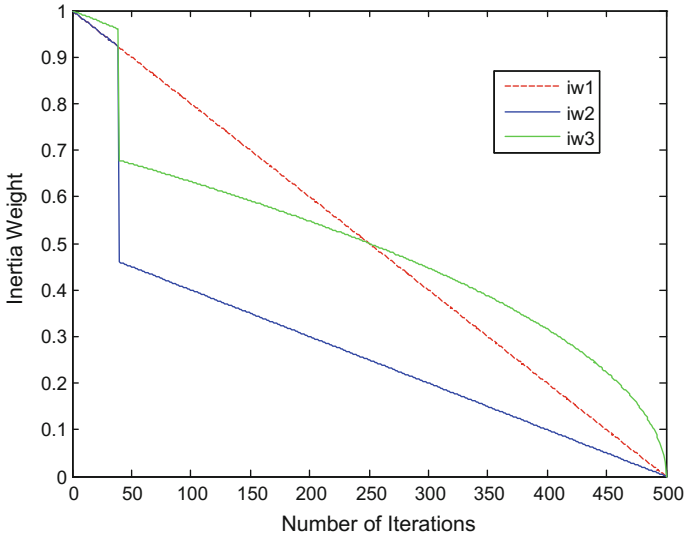


Fig. 12.4 Proposed inertia weight strategies

minimum MSE is improved when the inertia weight is suddenly changed from high to low after a particular iteration. Hence, the inertia weight is updated in three different ways as given in Eqs. (12.16), (12.17), and (12.18).

$$iw_1 = \eta \quad (12.16)$$

$$iw_2 = \begin{cases} \eta & \text{if } n \leq N \\ \eta/2 & \text{if } n > N \end{cases} \quad (12.17)$$

$$iw_3 = \begin{cases} \sqrt{\eta} & \text{if } n \leq N \\ \sqrt{\eta}/2 & \text{if } n > N \end{cases} \quad (12.18)$$

Figure 12.4 shows that, the inertia weight iw_1 decreases linearly from 1 to 0, whereas in the case of iw_2 and iw_3 the inertia weight decreases linearly until N iteration after that there is a sudden reduction in inertia weight. The common factor used in all time-varying inertia weight algorithms is $\eta = (m-n)/m$, where m denotes maximum iteration and n denotes current iteration. This factor η changes linearly from 1 to 0. If Eq. (12.14) is modified with a decreasing control function, it gives an effective time-varying inertia weight strategy as shown in Eqs. (12.17) and (12.18). The term N in Eqs. (12.17) and (12.18) is the intermediate iteration value used to reduce the inertia weight suddenly after N th iteration. This reduction produces optimum performance compared to existing inertia weight-modified methods in terms of convergence speed and MSE.

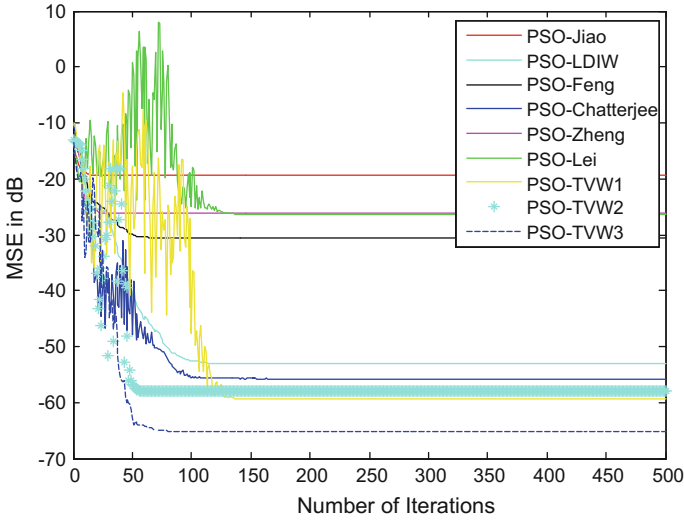


Fig. 12.5 Performance of proposed and other time-varying strategies in linear channel for LTE

12.4 Simulation Results

The simulations are performed in MATLAB 7.7 version. The parameters are set as amplitude distortion $W = 2.9$ which gives an eigenvalue spread 6, particle size $P = 40$, input window size $ws = 20$, the acceleration coefficients $ac_1 = 1$ and $ac_2 = 1$, tap weights $T = 7$, and the initial and final inertia weights $w_i = 0.9$ and $w_f = 0.3$, respectively. The term np used in Eq. (12.3) is taken as 0.7 and w_{initial} in Eq. (12.7) is selected as 0.5. The parameters considered above are minimum values to minimize the computational complexity in the algorithm. If the parameter values such as P , ws , ac_1 and ac_2 are increased, it can improve the performance and the same is analyzed in Sect. 4.1.

The PSO variants considered are PSO-LDIW [9], PSO-Chatterjee [17], PSO-Feng [18, 19], PSO-Lei, PSO-Zheng and PSO-Jiao. The proposed techniques, PSO with time-varying inertia weights, PSO-TVW, PSO-TVW1, PSO-TVW2, and PSO-TVW3 based on Eqs. (12.14), (12.15), and (12.16), are compared with the above variants in Fig. 12.5. Among all it is seen that the PSO variant PSO-LDIW and PSO-Chatterjee gives the best MSE value with convergence in 80th iteration. PSO variant PSO-LDIW is compared with proposed techniques in Table 12.1. Figure 12.6 depicts the convergence and MSE analysis of the proposed algorithms for LTE structure. The proposed PSO-TVW3-based algorithm converges fast with minimum MSE value.

Table 12.1 gives the comparison and effect of different channels using least mean square (LMS) algorithm, PSO-LDIW [9], PSO-TVW2, and PSO-TVW3 for LTE and DFE structures. From Table 12.1 and Figs. 12.5 to 12.6, it is shown that the proposed modifications outperform the other existing modifications based on

Table 12.1 Comparison of convergence rate and MSE with different channel models

Channel used	LMS			PSO-LDIW			PSO-TVW2			PSO-TVW3		
	MSE in dB	Convergence rate		MSE in dB	Convergence rate		MSE in dB	Convergence rate		MSE in dB	Convergence rate	
Linear channel LTE	-53	200		-50	80		-58	50		-67	50	
Linear channel DFE	-50	250		-48	90		-57	90		-67	100	
Nonlinear channel 1-LTE	-15	200		-40	45		-37	50		-46	50	
Nonlinear channel 1-DFE	-14	250		-40	60		-35	80		-46	80	
Nonlinear channel 2-LTE	-40	200		-57	70		-55	50		-78	50	
Nonlinear channel 2-DFE	-37	250		-55	80		-55	90		-62	100	
Nonlinear channel 3-LTE	-15	600		-15	30		-18	50		-28	50	
Nonlinear channel 3-DFE	-14	700		-15	50		-18	70		-23	50	

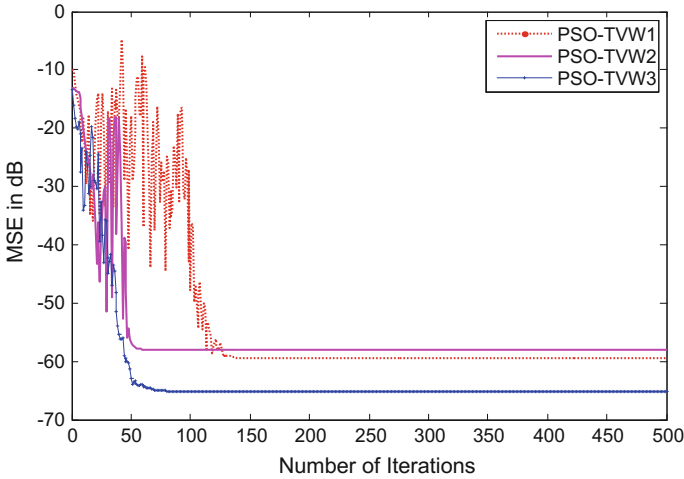


Fig. 12.6 Proposed PSO enhancements in linear channel for LTE

convergence and MSE. The PSO-TVW3 algorithm shows the best performance in all channel conditions. The LTE and DFE structures give approximately same MSE value, but differ in convergence rate which is shown in Table 12.1.

12.4.1 Sensitivity Analysis

The simulation-based sensitivity analysis is carried out to select optimum parameters in the algorithm. The parameter’s values and choices have high impact on the efficiency of the method, and few others have less or no effect. The analysis is done with respect to six key parameters, namely the intermediate iteration value N , data window size ws , acceleration constants ac_1 and ac_2 , the population size P , number of tap weights T , and distortion factor W . The effect of the basic PSO parameters swarm size or number of particles, window size, number of tap weights and acceleration coefficients are analyzed in [6]. The same is analyzed for PSO-TVW3 and is given in Table 12.2.

Table 12.2 Effect of PSO parameters on PSO-TVW3

Population P		Tap weights T		Window size WS		Acceleration coefficients c_1 and c_2		
P	MSE in dB	T	MSE in dB	WS	MSE in dB	c_1	c_2	MSE in dB
10	-52	5	-53	20	-65	1	1	-65
20	-64	7	-65	200	-68	1	2	-58
40	-70	9	-70	500	-69	2	1	-56
60	-72	13	-74	100	-70	2	2	-63

On average, an increase in the number of particles will always provide faster convergence. In contrast, the computational complexity can increase linearly with increase in population size. In Table 12.2, population size of 40 gives better convergence. So a problem-dependent minimum population size is good for better performance.

Setting the acceleration coefficients to a minimum value slows down the convergence speed. The local search and global search are best when the summation of acceleration coefficients becomes $ac_1 + ac_2 < 4$ in adaptive equalization. The acceleration coefficients greater than 1 also seem to give the best performance. For equal value of acceleration constants, the algorithm reaches its minimum in least number of iterations. The MSE calculated on each iteration is the average of the MSE over the window; a large window size increases the complexity per iteration and time consumption. In Table 12.2, window size does not make any greater changes in the MSE value. If the window size is small, the complexity can be reduced.

The tap weights are problem dependent. As given in Table 12.2, the increase in tap weights above a certain limit does not make much difference in MSE value, but it may increase the complexity. Figure 12.7 shows the analysis for different intermediate iteration N for PSO-TVW2. Table 12.3 compares the convergence rate and MSE for PSO-TVW2 and PSO-TVW3 with reference to N . An increase in the value of N increases the number of iterations required for convergence. Decreasing N value degrades the MSE performance. N value between 30 and 40 exhibits minimum MSE with faster convergence.

Table 12.4 explains the effect of amplitude distortion parameter W in linear channel. The MSE is computed with different amplitude distortion that leads to

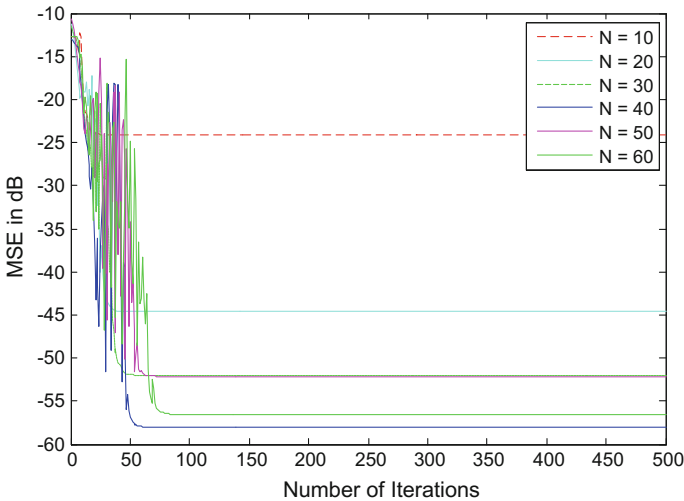


Fig. 12.7 Effect of different intermediate iteration value N on PSO-TVW2

Table 12.3 Comparison of convergence rate with different intermediate iteration value N

Intermediate iteration N	Convergence rate (iterations)	MSE in dB for PSO-TVW2	MSE in dB for PSO-TVW3
10	30	-24	-28
20	40	-45	-63
30	45	-52	-65
40	50	-58	-67
50	60	-52	-65
60	70	-56	-64

Table 12.4 Effect of amplitude distortion W on PSO-TVW2 for LTE and DFE

SNR in dB	LTE			DFE		
	For $W = 3.7$ MSE in dB	For $W = 3.1$ MSE in dB	For $W = 2.9$ MSE in dB	For $W = 3.7$ MSE in dB	For $W = 3.1$ MSE in dB	For $W = 2.9$ MSE in dB
5	-17	-18	-20	-17	-18	-19
10	-24	-26	-27	-29	-27	-25
15	-25	-27	-33	-32	-30	-32
20	-27	-35	-40	-35	-37	-37
25	-33	-44	-53	-38	-45	-53
30	-37	-61	-67	-46	-62	-67

different eigenvalue spread. Increase in amplitude distortion degrades the MSE performance. The performance degradation is not severe in proposed PSO-based algorithms compared to existing algorithms.

12.5 Conclusion

In this work, two new strategies are proposed to update in PSO algorithm for adaptive equalization. The results are discussed in linear and nonlinear channels for LTE and DFE structures. The proposed time-varying PSO algorithms, PSO-TVW2 and PSO-TVW3, show better performance than the existing algorithms in linear and nonlinear channels. Also it shows performance better for LTE and DFE structures. The PSO-TVW3 algorithm outperforms other modifications based on convergence and MSE. The sensitivity analysis is done to find the optimum parameter values. The performance is degraded for severe amplitude distortions in linear channel. The DFE structure delays the convergence but it performs well in severe amplitude distortion conditions.

References

1. Qureshi S (1985) Adaptive equalization. *Proc IEEE* 73:1349–1387
2. Haykin S (1996) Adaptive filter theory, 3rd edn. Prentice-Hall, Inc
3. Shin HC, Saved AH (2004) Mean-square performance of a family of affine projection algorithms. *IEEE Trans Signal Process* 52:90–101
4. Karaboga N, Cetinkaya B (2011) A novel and efficient algorithm for adaptive filtering: artificial bee colony algorithm. *Turk J Electr Eng Comput Sci* 19(1):175–190
5. Krusienski DJ (2004) Enhanced structured stochastic global optimization algorithms for IIR and nonlinear adaptive filtering. Ph.D. thesis, Department of Electrical Engineering, The Pennsylvania State University, University Park, PA
6. Al-Awami AT, Zerguine A, Cheded L, Zidouri A, Saif W (2011) A new modified particle swarm optimization algorithm for adaptive equalization. *J Digital Signal Process* 21(2): 195–207
7. Das G, Pattnaik PK, Padhy SK (2014) Artificial neural network trained by particle swarm optimization for non-linear channel equalization. *J Exp Syst Appl* 41(7):3491–3496
8. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of the IEEE international conference on neural networks, vol 4, pp 1942–1948
9. Nickabadi A, Ebadzadeh MM, Safabakhsh R (2011) A novel particle swarm optimization algorithm with adaptive inertia weight. *J Appl Soft Comput* 11(4):3658–3670
10. Shi Y, Eberhart R (1998) A modified particle swarm optimizer. In: Proceedings of the international conference on evolutionary computation, IEEE world congress on computational intelligence, pp 69–73
11. Eberhart RC, Shi Y (2000) Comparing inertia weights and constriction factors in particle swarm optimization. In: Proceedings of the 2000 congress on evolutionary computation, vol 1, pp 84–88
12. Chatterjee A, Siarry P (2006) Nonlinear inertia weight variation for dynamic adaption in particle swarm optimization. *Comput Oper Res* 33(3):859–871
13. Feng Y, Teng G, Wang A, Yao YM (2007) Chaotic inertia weight in particle swarm optimization. In: Second international conference on innovative computing, information and control (ICICIC 07), pp 475–475
14. Feng Y, Yao YM, Wang A (2007) Comparing with chaotic inertia weights in particle swarm optimization. In: International conference on machine learning and cybernetics, pp 329–333
15. Lei K, Qiu Y, He Y (2006) A new adaptive well-chosen inertia weight strategy to automatically harmonize global and local search ability in particle swarm optimization. In: ISSCAA
16. Fan S, Chiu Y (2007) A decreasing inertia weight particle swarm optimizer. *Eng Optim* 39 (2):203–228
17. Zheng Y, Ma L, Zhang L, Qian J (2003) Empirical study of particle swarm optimizer with an increasing inertia weight. In: IEEE congress on evolutionary computation
18. Jiao B, Lian Z, Gu X (2008) A dynamic inertia weight particle swarm optimization algorithm. *Chaos Solitons Fractals* 37(3):698–705
19. Patra JC, Pal RN, Baliarsingh R, Panda G (1999) Nonlinear channel equalization for QAM signal constellation using artificial neural networks. *IEEE Trans Syst Man Cybern* 29(2):262–271