International Journal of Electrical and Computer Engineering (IJECE)
Vol. 10, No. 3, June 2020, pp. 2997~3006
ISSN: 2088-8708, DOI: 10.11591/ijece.v10i3.pp2997-3006

Optimized BER for channel equalizer using cuckoo search and neural network

Swati Katwal[1], Vinay Bhatia[2]
[1]Department of Electronics and Communication Engineering,
Baddi University of Emerging Sciences and Technology, India
[2]Department of Electronics and Communication Engineering, Chandigarh Engineering College, India

Article Info

Article history:

Received Apr 16, 2019
Revised Dec 5, 2019
Accepted Dec 10, 2019

Keywords:

ANN
Channel estimation
CS
Equalization

ABSTRACT

The digital data transfer faces issues regarding Inter-Symbol Interference (ISI); therefore, the error rate becomes dependent upon channel estimation and its equalization. This paper focuses on the development of a method for optimizing the channel data to improve ISI by utilizing a swarm intelligence series algorithm termed as Cuckoo Search (CS). The adjusted data through CS is cross-validated using Artificial Neural Network (ANN). The data acceptance rate is considered with 0-10% marginal error which varies in the given range with different bit streams. The performance evaluation of the proposed algorithm using the Average Bit Error Rate (A-BER) and Logarithmic Bit Error Rate (L-BER) had shown an overall improvement of 30-50% when compared with the Kalman filter based algorithm.

Corresponding Author:

Swati Katwal,
Department of Electronics and Communication Engineering,
Baddi University of Emerging Sciences and Technology,
Solan, India.
Email: engg.swati@yahoo.co.in

1. INTRODUCTION

Digital Communication (DC) provides digital data transfer with high transfer rate [1-2]. The average power of the transmitted signal reduces as the distance between transmitter and receiver surges. The noise comes in the form of disturbance and Inter-Symbol Interference (ISI) in the received signal. Figure 1 shows the occurrence of ISI in the received signal. It is seen that due to multipath propagation of the transmitted signal, the strength of the received signal reduces due to the presence of buildings and another obstacle like airplane and trees etc.

In this paper, the authors have addressed the ISI distortions and propose an approach to reduce this unwanted phenomenon to improve the reliability of the communication.Basically, there are two main causes of occurring ISI. These are non-linear frequency between the channels and multipath propagation. Many attempts have been made by the researchers to reduce the interference effect. There are various algorithms and filter proposed by scholars to combat the interference effect. For instance, Kalman Filter (KF) is widely used to reduce the ISI and channel interference effects. This filter is used to linearize the nonlinear models [3]. If the data is well structured, it results in a balanced channel estimation whichresults in a minimum error rate. Although KF and Extended Kalman Filter (EKF) are used worldwide to rectify the channel data, it lacks in adaptive filtering [4]. Equalizers are used in the adaptive filters. Figure 2 shows the use of equalizer between the transmitter and the receiver. The data is transferred from transmitter to receiver. But, the transfer data stream cannot be kept homogenous throughout [5-6]. For estimating the time-varying channel coefficients different algorithms are available. Due to the simplicity of the adaptive algorithm, the coefficients of the channel are estimated using Least Mean Square (LMS) algorithm.

LMS adaptive algorithm aims to minimize the error relying on the network. The filter has a one-step predictor-corrector structure and minimizes the mean square estimation error in each step [7-9].

This paper compares the KF based equalizer with EKF based for time-varying channels by updating its coefficients.By utilizing KF only it may not be the best solution for the future complex demands of channels interference, noise, and redundancies. The paper proposes the establishment of a hybrid method of Swarm Intelligence (CS) and machine learning based mechanism. CS is a hard threshold optimization technique and hence to rectify the bit stream, CS is utilized here [10]. A new fitness function is designed, implemented, and cross-validated using feed forward back propagation Neural Network (NN). Artificial Neural Network (ANN) is being used all over in digital communication [11-12]. Further multi-model perceptron model which is an extension of uni-model perceptron model is also used to equalize the channel response [13]. The proposed work took the inspiration from the utilization of multi-model perceptron model and aimed to design a better multi-model network to optimize the channel response. ANN is combined with swarm intelligence to equalize the channel [14 -15]. The literature review is presented in section 2. The proposed model and its validation using ANN are described in Section 3. The evaluation of the proposed work model and a comparison with KF along with resultsis also presented in Section 4. The conclusion of the paper is presented in Section 5.

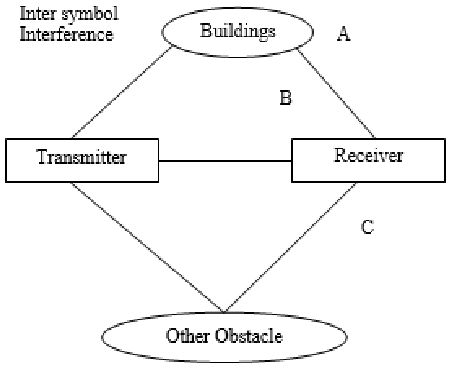


Figure 1. Effect of ISI on the received signal

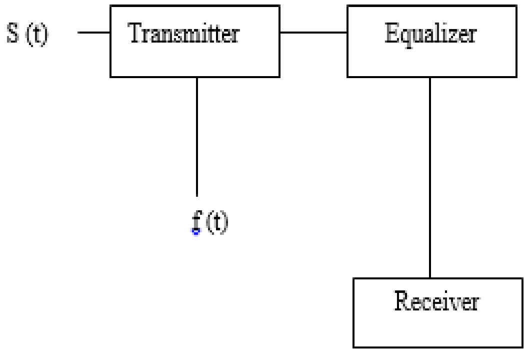


Figure 2. Use of equalizer to remove the noise level

2. BACKGROUND REVIEW

In a digital communication system, bandwidth is a scarce resource in the modern world. The efficient utilization of this resource is vital in these days. The data transfer rate depends upon the utilization of bandwidth. The strength of the signal transmitted over the channelreduces due to the presence of various obstacles during transmission.Moreover, the signal quality also dampens to a large extent due to the presence of many other frequencies in the channel. Due to the presence of objects, ISI and noise occur with the input signal at the receiver end.Many filters are used to reduce these disturbances. But nonlinear filters are insufficient to reduce the level of noise. Therefore, linear filters such as Adaptive filters are in trend, used to reduce the level of noise intensity. Equalizersare used in these filters for structural development. Equalizersare used basically for channel equalization. It is a process which reduces the interference level to the desired amount.The process of channel equalization is carried out at the destination end which modifies the signal quality and reduces the interference effect. Deng equalizes the noise level using the Complex-Valued Minimal Radial Basis Function Neural Networks [16]. Besides, researchers and practitioners also develop filters and algorithms to equalize the noise level. The developed filter reduces the bit error rate by suppressing the major calamities like ISI and white noise. In addition, Candy uses the Kalman filter to achieve optimum results. This filter is used with Recursive least squares and least mean a square which reduces the performance of bit error rate [17]. The output signal having disturbance factors such as noise and error can be overcome by using these filters. The channel equalization is achieved using these filters but there is one drawback. These adaptive filters in non-linear reduce the channel performance which is a major drawback. Moreover, adaptive filters generally diminish in non-linear filtering [18].

Researchers attempted to modify the filter structures to improve the performance of the adaptive filters in non-linear filtering using ANN [19]. Therefore, different algorithms are developed in the literature which revamps the performance of the channel [20].But these models do not reduce the interference level.Therefore traditional methods are compared with the present model. It is noted that ANN provides better results than traditional methods. ANN is popular to combat complex problems. Artificial neural networks are used to equalize the level of noise. If the channel suffers the no-n-equalizing problem then equalization is basically used to inverse the filter results. The traditional methods use the inverse filtering technique to combat the disturbances. The process of inverse filtering is used at the front end of the receiver in the digital communication system. The traditional methods used to combat the time-varying distortions and noise. In addition, the characteristics of the filter are adjusted in such a way that the signal level is maintained. The noise level and ISI reduces using adaptive linear filters [21]. These linear filters play an important role in determining the characteristic of the receiver. Due to advancement in technology, advanced linear filters were developed by many scholars and practitioners. In order to achieve, fast convergence rate, ANNare used to equalize the noise level. ANN improves the performance of the channel by solving complex problems. A neural architecture is developed using linear filtering techniques.

In the case of neural networking, the problem of inverse filtering reduces to a great extent. This filtering technique can be used at all the channels. A multilevel channel equalizer reduces the level of the noise by reducing the mean square error andnon- linear communication channels can be handled easily using the neural networks. The network outcomes result in a fast convergence rate and fast speed of the network. In addition, multilevel perceptron combats non-linear optimization problems. Moreover, the properties of ANN are organized in such a way that optimizing problems can be solved using linear equalizing [22]. The input and output mapping are sorted by using equalizer at the front end of the main receiver. The developed equalizer determines the time-varying coefficients of the signal. Furthermore, the coefficients are estimated using the training sequences. In the networks, the blind equalizer is trained using the training sequences.These training sequences are also used in adaptive filtering. A direct search algorithm is implemented using the neural network to converge the outcomes [23]. Baloch uses the perceptron ANN in combination with the simulated network to combat the ISI. An error back propagation algorithm is used for performing blind optimization.The optimization technique improves the performance of the network. In addition, the developed algorithm achieves the desired result by reducing the level of the noise.

Some scholars also use fitness functions for consistent results. The fitness function using neural networks develop the filters using time-varying components. The new structured filters have high computational complexity and high data speed rate. The data symbols are extracted with a fast convergence rate. The addition of white Gaussian noise and ISI doesn’t affect the signal strength using fitness function. Moreover, the signal is easily detected in the presence of these interferences. The ISI component almost approaches to zero with the newly developed structures of the filter. Moreover, Katwal also uses the Artificial Neural Network (ANN) for adaptive channel equalization using 4-Quadrature Amplitude Modulation (QAM) signal in the digital communication system [24].The well-structured filters using the latest models like ANN and simulated network equalize the nonlinear networks [25]. However, casting such a network requires high bandwidth and network must be a single layer.The non-linear models are less converged to the required output [26]. Therefore, an attempt has been made to converge the non-linear models into linear models using minimal radial neural networks [27]. Tsuda develops the improved NLMS algorithm for channel equalization.The non-linear models are converged into linear optimized models which provide the consistent outcomes [28]. The linear models used in the network equalize the noise level. The developed filters reduce the interference level in both non-stationary and stationary networks. In this paper, an efficient technique is developed. An attempt has been made to propose the multi-model network to achieve the desired result. The channel response is optimized using linear filtering. The neural network methods are combined with soft computing based optimization technique. The swarm intelligence model is combined with ANN to reduce the ISI in the signal. Moreover, effective and desired results are achieved by implementing the novel technique. A novel strategy is introduced in this paper using optimization techniques. The channels are equalized using the developed models and algorithms. An efficient model is developed in this paper to improve the performance of BER. An optimal channel BER is achieved using the fitness function. The developed algorithm equalizes the noise level using linear optimization techniques. The equalization models are developed for stationary and non-stationary channels. In addition, the fitness function is designed using swarm optimization techniques to reduce the ISI.The simulation results arepresented in this paper which proves the conformity of the developed approach.

3. PROPOSED WORK MODEL

The proposed structure essentially aims to reduce the Bit Error Rate (BER) of the data and it also focuses on algorithms to equalize the channel. This section is divided into two parts as follows:

3.1. CS algorithm for channel equalization

CS is to be applied over the data which is to be transferred. The data is transferred through a channel, which model for both stationary and nonstationary has been presented in the sequel.

3.2. Equalization for stationary channel

Discrete-time filter with additive white Gaussian noise (AWGN) and Raleigh Fading (RF) is utilized one by one, and the output of the channel $\mathrm{z(n)}$ with measurement noise $\mathrm{v(n)}$is

$$\mathrm{z(n) = C^T U(n) + v(n)} \qquad (1)$$

As shown in (1), $\mathrm{c = [c_0, c_1, c_2, .., c_{M-1}]^T}$ is the channel co-efficient, subscript T denotes the transpose and input signal vector is given by

$$\mathrm{U(n) = [u(n), u(n-1), .., u(n-M+1)]^T} \qquad (2)$$

If we want to represent this transversal filter into the state space equation, then we can consider the time delayed input as states. By considering the time delayed input as state variables, the state space model of the channel can be obtained by [6, 7].

$$\mathrm{X(n+1) = AX(n) + Bu(n) + w(n)} \qquad (3)$$

$$\mathrm{z(n) = C^T X(n) + v(n)} \qquad (4)$$

$\mathrm{X(n) = [x(n), x(n-1), \ldots x(n-M+1)]^T}$ are the state variables of the state model, w(n) is the process noise and v(n)is the measurement noise. The state transfer, input and output matrices are

$$A = \begin{bmatrix} 0 & \cdots & \cdots & \cdots & 0 \\ 1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & 1 & 0 \end{bmatrix} B = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix} \qquad (5)$$

$$C = [c_0, c_1, c_2, \ldots., c_{m-1}]^T \qquad (6)$$

Since at the receiving end the actual state is unknown and actual input so we will model a filter where input, as well as original states, are estimated. The estimated input and estimated state equations are represented as:

$$\mathrm{X^{n+1} = AX^n + Bu^n} \qquad (7)$$

$$\mathrm{\hat{z}(n) = C^T \hat{X}(n)} \qquad (8)$$

As shown in (7) $\mathrm{\hat{X}(n)}$ is estimated state and in (6) $\mathrm{\hat{u}(n)}$ is estimated input.

3.3. Equalizer a for non-stationary channel

The system for a time-invariant channel whose coefficients C are known is considered first. But in practical cases, most of the channel is time-varying. So to implement such channels in the above scheme it is required to estimate the channel coefficients C = $\mathrm{\hat{C}(n)}$ using the LMS algorithm, thus (5) is as,

$$\mathrm{\hat{z}(n) = \hat{C}^T \hat{X}(n)} \qquad (9)$$

This estimated output$\mathrm{\hat{z}(n)}$) is compared with actual output $\mathrm{z(n)}$ and the estimated states are updated in the equation:

$$\mathrm{X^{n+1} = X^n + K[z(n - z^n)]} \qquad (10)$$

This is the time when the data is going to be received by the receiver. The equalizer plays its part and CS is introduced in the process. Figure 3 represents the architecture of the equalizer where CS is to be applied. w_k and v_k represent the process noise and measurement noise. In the difference equation, the non-linear function f relates the state at the previous time step $k-1$ to the state at the current time stepk. In the measurement equations, the non-linear function relates the state to the measurementz_k.

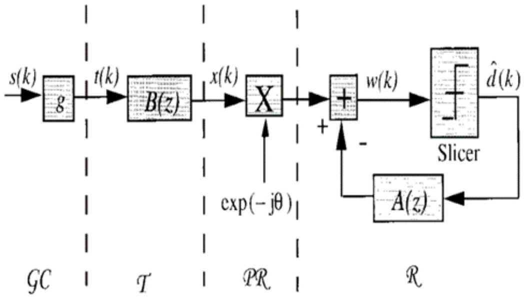


Figure 3. Structure of the equalizer

If the value of two random variables is not known, then we can approximate the state and measurement vector as:

$$x_k = f_{k-1}, u_k \tag{11}$$

UtilizingEKF

$$z_k = h(x_k, v_{k)} \tag{12}$$

As shown in (12), x_k is a posteriori estimate of the state. The new governing equations that linearize an estimate about (10) and (11) are as following:

$$x_{\tilde{k}} = \ f(x_{k-1}, u_k, 0) \tag{13}$$

$$z_{\tilde{k}} = h(x_k, 0) \tag{14}$$

where x_k and z_k the actual state and measurement vectors,$x_{\tilde{k}}$and $z_{\tilde{k}}$are the approximate state and measurement vectors, $x_{\tilde{k}}$is an aposteriori estimate of the state at stepK, A,W,H, V are the matrix of partial derivatives of x,w,x,v. Now the prediction and measurement errors are:

$$x_k \sim x_{\tilde{k}} + A(x_{k-1} - x_{\tilde{k}} - 1) + WW_{k-1} \tag{15}$$

$$z_k \sim z_{\tilde{k}} + H(x_{k-1} - x_{\tilde{k}}) + VV_k \tag{16}$$

where $\in_k$ and n_k are independent random variables having zero mean. The data passes through the equalizer and gets into the channel. The system mainly operates in decision directed mode when actual transmission and receiving information takes place. In the decision directed mode the output of CS is used as input for and the aptive filter. We are already using $u(n)$ and $\hat{u}(n)$as input for training and decision directed mode to adaptive filter at receiving end.The similar input can be provided to KF i.e. in decision directed mode whereas in training mode [20].The data must satisfy the demands of CS to get through. The structure is as follows.

Algorithm 1: Application of CS

```
Input : Bit Stream(BS)    Output: Optimal Bit Stream
Total_Egg_Slots=4;
Divide the BS into Egg_slot Time Segments. // Dividing the bit stream into frames to process // the data smoothly
eggsperslot=(duration)/Total_Egg_Slots;
initialcounter=1; // Taking a counter to keep optimal
// eggs only
```

```
finalcounter=eggsperslot
foreach egg in Total_Egg_Slots
Eggcounter=initialcounter:finalcounter // The
The//encounter will start from 1 and will go to 4
Eggsinslot=BS(Eggcounter);// Extracting data from Bit
//Stream
TotalEggs=BS;
Fitvalue=CuckooFitness(Eggsinslot);// Passing the data to the fitness function of Cuckoo Search
End For
End Algorithm
```

CS is a hard threshold algorithm. The entire bitstream received is divided into four egg slots. The egg value is passed to the fitness function of the CS when the cuckoo is out. The fitness function of the CS is defined as follows.

Let Nc be a natural change in the structure of the eggs when the cuckoo bird is out. The effectiveness of the data is evaluated in contrast to the other data 'De' which is present in that environment. Hence the Cuckoo Search Algorithm utilizes both the Nc and De in its fitness function:

$$\mathrm{De} = \sum_{k=0}^{n}(\mathrm{d})/\mathrm{n} \quad (17)$$

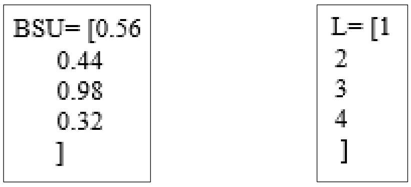
where D is the data in the bitstream, n is the total number of bits in the time stream

$$\text{if } \mathrm{D} * \mathrm{N_C} < \mathrm{D_e} * \mathrm{N_c} \quad (18)$$

Otherwise: If the current data element D steps in with the Nc and still is smaller than the overall other data present in the system, then the cuckoo will keep this egg else the egg will be dumped that means the data bit will be forwarded else the data bit will be dumped. This architecture frame is further cross-validated by ANN, to prove that the fitness function of the CS is sufficient in this contrast. ANN is a three-layer structure namely input, hidden and an output layer.

3.4. Cross-validation using ANN

The input layer of the Neural network intakes the raw data at the front. ANN does not understand what it is provided to it if it is not converted into a subtype which is understood by its architecture and hence it utilizes the sigmoid function and weight converted to generate intermediate layer data of its processing. This paper considers supervised machine learning for its processing. The architecture takes the entire optimized bit pattern as input with additive label series starting from 1 to n where n is a total number of the bit sequence. Hence if the bitstream is Figure 4(a) then its label would be Figure 4(b). BSU is optimized bit stream and L is target label.Neural Network is utilized in such a manner that every bitstream is trained using Levenberg architecture of ANN. Table 1 represents the simulation architecture of ANN. A total of 500 epochs is provided for every simulation of ANN. Algorithm 2 explains the working of the ANN in the proposed structure.



BSU= [0.56
0.44
0.98
0.32
]

L= [1
2
3
4
]

(a) (b)

Figure 4 (a). Input layer data, (b). Input layer label

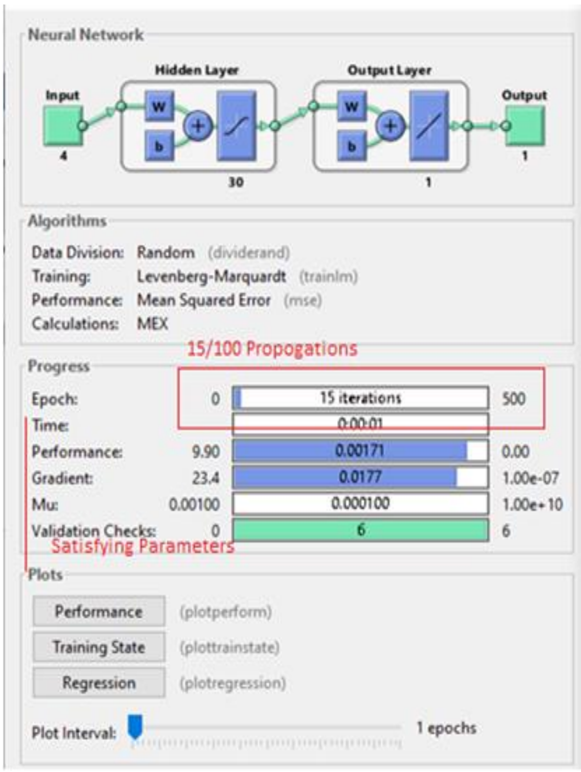
Table 1. Neural structure

Total Number of Neurons	10-30
Architecture Type	Feed Forward Back Propagation
Performance Measure	Mean Square Error
Feeding Parameters	Gradient, Time, Iterations

Algorithm 2: WorkFlow of ANN

```
1. Net=Train_Neural(BSU,L ,10-30); // Initiated Training Mechanism for optimized Bit //
   // Stream(BSU) with its corresponding label L.
   // The trained structure is saved in a structure name
   //"net"
2. Net.trainparam.epochs=500// 500 training iterations
3. If satisfied(Gradient)  // If the Gradient is satisfied, then stop feeding in the forward direction
4. BackTrack(); // BackTrack and Find most suitable MSE
5. End If
6. Find Most Suitable Iteration based on MSE;
7. Simulate(BSU,Net); // Simulate BSU with stored structure
End Algorithm
```

Algorithm 2 is a combination of FeedForward and Back Propagation Algorithm in which the Feeding in forwarding direction takes place until either the propagation iteration becomes equal to a total number of provided iteration or the gradient of the ANN is not satisfied. This process is followed by the Back Propagation method whose satisfying parameter is MSE. The simulator rolls back to find the least MSE based on the understanding of the gradient. Once the ANN is trained, each optimized BSU is classified with the trained structure. If the result label of that BSU is not equal to the Label set L then the main bit value of that optimized bit value is optimized again. Figure 5 and Figure 6 represents the training architecture of ANN and Backpropagation.



Algorithms

Data Division: Random (dividerand)
Training: Levenberg-Marquardt (trainlm)
Performance: Mean Squared Error (mse)
Calculations: MEX

15/100 Propogations

Progress

Epoch:	0	15 iterations	500
Time:		0:00:01	
Performance:	9.90	0.00171	0.00
Gradient:	23.4	0.0177	1.00e-07
Mu:	0.00100	0.000100	1.00e+10
Validation Checks:	0	6	6

Satisfying Parameters

Plot Interval: 1 epochs

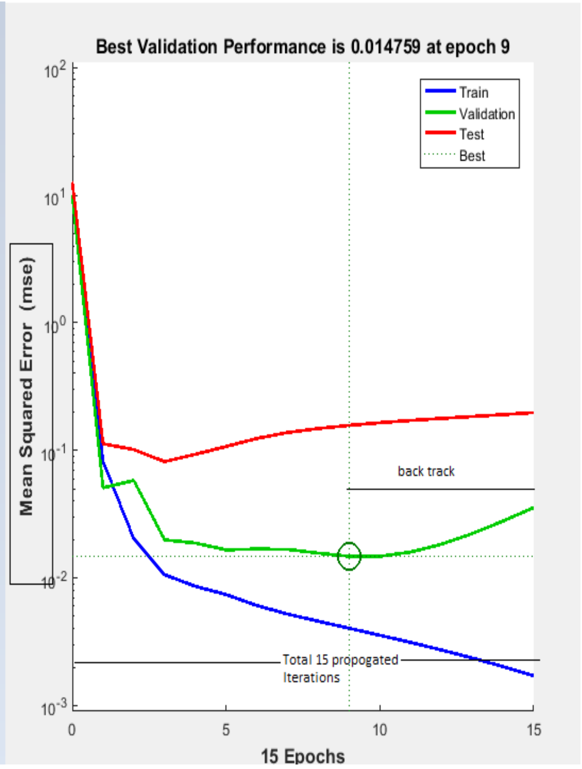


Figure 5. Training model of ANN

Figure 6. Back propagation

As shown in Figure 6, the network back propagates in the network. ANN considers some data as test data out of the provided training data even at the training phase. Tansig is the training function which generates the weight W of the training data D considering a and b as arbitrary constants. In a similar fashion, purelin function is utilized at the simulation function. The ANN utilizes sigmoid training function to train the data and also to create a cross-validation data for testing. Neural architecture is shown in Figure 7.

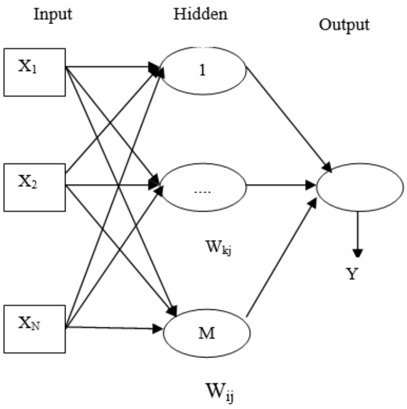


Figure 7. Neural architecture

4. RESULTS AND DISCUSSION

Following parameters are analyzed based on the proposed work.

4.1. A-BER

It is the average bit error rate of the network which is generated after 10000 simulation rounds. Each simulation round generates a bit error rate value which is stored in the simulation array and a comparison of proposed work and Kalman filter results are hereby compared. A-BER is noted after every 10000 simulation round. Average Bit Error Ratecan be calculated as follows:

$$\mathrm{A-BER} = \frac{\sum_{k=0}^{n-10000} \mathrm{BER}}{n} \tag{19}$$

Figure 8 presents the A-BER for Proposed work model evaluated against Kalman Filter Model. It is observed that our proposed model demonstrated lesser average bit error rate as compared to the Kalman Filter Model when simulated over 10000 simulations.

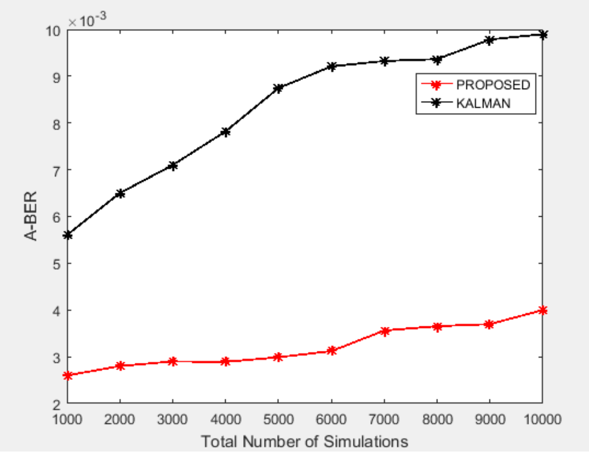


Figure 8. A-BER vs total number of simulations

4.2. L-BER

It is the logarithm scale of the A-BER but it is taken again the SNR value of the communication channel. Figure 9 shows the comparison of L-BER and SNR. The simulation graph shows that L-BER plotted against the signal to noise ratio also showed the effectiveness of our proposed design. The results show that proposed hybrid is more effective in improving the ISI as compared to the Kalman Filter Model. In both cases, the proposed solution has shown improvement by a margin of 30-50%. The maximum A-BER for the proposed solution is noted to be 0.003999 whereas it is 0.009899 for Kalman Filter. The proposed model also evaluated MSE at each training interval and the results are depicted in Figure 10. The graph shows that root mean square changes with each simulation and the graph compares the observed error for the 10 simulations.

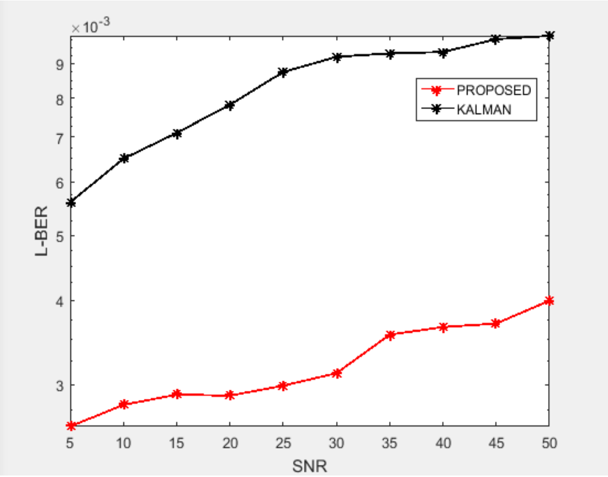


Figure 9. L-BER vs SNR

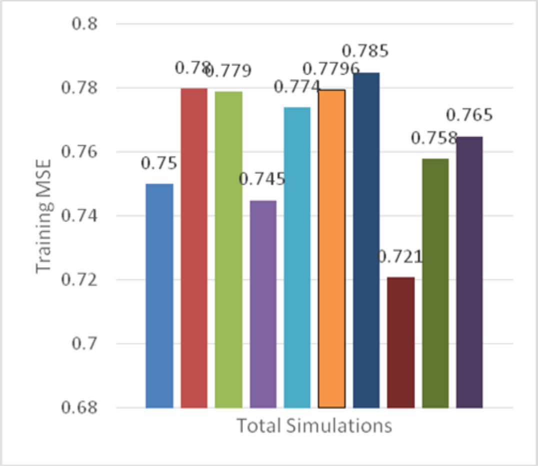


Figure 10. Training MSE

5. CONCLUSION

This paper presents an enhancement of the filtering technique utilizing the CS and ANN. A new fitness function is designed in the CS and ANN has acted like a cross validator to the CS. In order to validate the proposed algorithm, both the training techniques like Feed Forward and Back Propagation have been implemented. MSE is considered as the parameter of cross-validation whereas the proposed work is compared with the Kalman Filter and is evaluated on the base of A-BER and L-BER. An enhance improvement in ISI ranging from 30% to 50% has been achieved when the data have been simulation on 10000 simulations.

REFERENCES

[1] P. S. Henry, *et al.*, “Transmission device with channel equalization and control and methods for use therewith,” *United States Patent Application*, no. 9, pp. 670-680, 2017.

[2] R. Martinek, *et al.*, “Adaptive optimization of control parameters for feed-forward software defined equalization,” *Wireless Personal Communications*, vol. 95, no. 4, pp. 4001-4011, Aug. 2017.

[3] G. Das, *et al.*, “Artificial neural network trained by particle swarm optimization for non-linear channel equalization,” *Expert Systems with Applications*, vol. 41, no. 7, pp. 3491-3496, Jun. 2014.

[4] S. Baloch, *et al.*, “Channel Equalization Using Multilayer Perceptron Networks,” *Mehran University Research Journal of Engineering and Technology*, vol. 31, no. 3, Jul. 2012

[5] X. Yang and Suash Deb, "Cuckoo Search via Lévy flights," *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, Coimbatore, pp. 210-214, 2009.

[6] K. M. S. Huq, *et al.*, “Channel Equalization in Digital Transmission,” *International Journal of Computer Science and Information Security (IJCSIS)*, USA, vol. 4, no. 1&2, Aug. 2009.

[7] M. Enescu, *et al.*, “Adaptive Equalization of Time-Varying MIMO channels,” *Signal Processing*, vol. 85, no. 1, pp. 81-93, Jan. 2005.

[8] R.A. Boujemaa and S. Marcos, “Parallel Kalman Filtering for Optimal Symbol-By-Symbol Estimation in an Equalization Context,” *Signal Processing*, Elsevier, vol. 85, no. 6, pp. 1125-1138, Jun. 2005.

[9] F.Orderud, “Comparison of Kalman Filter Estimation Approaches for State Space Models with Nonlinear Measurements,” *In Proceeding of Scandinavian Conference on Simulation and Modeling*, pp. 1-8, 2005.

[10] J.W. Mark, “A Note on the Modified KalmanFilter for channel Equalization,” in *Proceedings of the IEEE*, vol. 61, no. 4, pp. 481-482, Apr. 1973.

[11] H. N. Kim and W. J. Song, “An Adaptive IIR Equalizer for Non-Minimum-Phase channel,” *ICSP '98. 1998 Fourth International Conference on Signal Processing (Cat. No.98TH8344)*, Beijing, China, vol. 1, pp. 441-444, 1998.

[12] J.C. Patra and R.N. Pal, “A functional link artificial neural network for adaptive channel equalization,” *Signal Processing*, vol. 43, no. 2, pp. 181-195, May 1995.

[13] T. Shimamura, *et al.*, “Equalization of Time-Variant Communication Channel via Channel Estimation Based Approaches,” *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings,* Atlanta, GA, USA, vol. 3, pp. 1703-1706, 1996.

[14] P. Grohan and S. Marcos, “Structures and Performances of Several Adaptive Kalman Equalizers,” *1996 IEEE Digital Signal Processing Workshop Proceedings,* Loen, Norway, pp. 454-457,1996.

[15] S. Mclaughlin, “Adaptive Equalization via Kalman Filtering Techniques,” in *IEEE Proceedings F - Radar and Signal Processing*, vol. 138, no. 4, pp. 388-396, Aug. 1991.

[16] Deng J., *et al.*, “Communication Channel Equalization Using Complex-Valued Minimal Radial Basis Function Neural Networks,” in *IEEE Transactions on Neural Networks*, vol. 13, no. 3, pp. 687-696, May 2002.

[17] J.V. Candy, “Model-based signal processing,” *John Wiley & Sons*, vol. 36, Oct. 2005.

[18] S. O. Haykin, “Adaptive Filter Theory,” Prentice Hall, Eagle wood Cliffs, NJ, 2nd Ed., 1991.

[19] G.R. Patra, *et al.*, “Non linear channel equalization for digital communication using DE trained functional link artificial neural networks,” *International Conf. on Contemporary Computing, Heidelberg,* vol. 2, pp. 403-14, 2011.

[20] P. S. R. Deniz, “Adaptive Filtering and Practical Implementation,” *Springer*, Jan. 2008.

[21] S. McLaughlin, *et al*., “A Performance Study of Three Adaptiv Equalizer environment,” *IEEE International Conf. on Communications, World Prosperity Through Communications*, Boston, MA, USA, vol. 1, pp. 193-197, 1989.

[22] K. Burse, R.N. Yadav, S.C. Shrivastava, “Channel Equalization using neural network,”in *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews),* vol. 40, no. 3, pp. 352-357, May 2010.

[23] S. Panda, P. K. Mohapatra, “A new training scheme for neural networks and application in Non-Linear Channel Equalization,” *Applied Softcomputing*, vol. 27, pp. 47–52, Feb. 2015.

[24] S. Katwal *et al*., “Non Linear Channel Equalization Using Artificial Neural Network,” *RIEECE*, vol. 3, no. 1, pp. 333-334, 2016.

[25] J.C. Patra, *et al.*, “Nonlinear Channel Equalization with QAM Signal Using Chebyshev Artificial Neural Network,” *Proc. 2005 IEEE International Joint Conference on Neural Networks*, Montreal, Que., vol. 5, pp. 3214-3219, 2005.

[26] W.D. Weng, *et al*, “The Design of an SCFNN Based Nonlinear Channel Equalizer,” *Journal of Information Science and Engineering*, vol. 21, no. 4, pp. 695-709, Jul. 2005.

[27] C. P. Kumar, *et al.*, “Nonlinear channel equalization using minimal radial basis function neural networks,” *Proc. Inst. Elect. Eng. Vision Image Signal Processing,* vol. 147, pp. 428-435, 2000.

[28] Y. Tsuda, *et al.*, "An improved NLMS algorithm for channel equalization," *2002 IEEE International Symposium on Circuits and Systems. Proceedings (Cat. No.02CH37353)*, Phoenix-Scottsdale, AZ, USA, pp. V-V, 2002.