

Artificial Neural Network Trained by Particle Swarm Optimization for Non-Linear Channel Equalization

Dr. Ravi Narayan Panda¹, Kaustav Das² and Subrat Kumar Panda³

¹Professor, Department of Electronics and Communication Engineering, Gandhi Institute for Technology (GIFT), Bhubaneswar

²Assistant Professor, Department of Electronics and Communication Engineering, Gandhi Engineering College, Bhubaneswar

³Assistant Professor, Department of Electronics and Communication Engineering, Gandhi Institute for Technology (GIFT), Bhubaneswar

Publishing Date: March 30, 2016

Article info

Keywords:

Artificial Neural Network
Particle Swarm Optimization
Channel equalization

abstract

In this paper, we apply Artificial Neural Network (ANN) trained with Particle Swarm Optimization (PSO) for the problem of channel equalization. Existing applications of PSO to Artificial Neural Networks (ANN) training have only been used to find optimal weights of the network. Novelty in this paper is that it also takes care of appropriate network topology and transfer functions of the neuron. The PSO algorithm optimizes all the variables, and hence network weights and network parameters. Hence, this paper makes use of PSO to optimize the number of layers, input and hidden neurons, the type of transfer function set. This paper focuses on optimizing the weights, transfer function, and topology of an ANN constructed for channel equalization. Extensive simulations presented in this paper show that, as compared to other ANN based equalizers as well as Neuro-fuzzy equalizers, the proposed equalizer performs better in all noise conditions.

1. Introduction

Adaptive channel equalizers play an important role in recovering digital information from digital communication channels. In Voulgaris and Hadjicostis (2004), the authors have proposed optimal preprocessing strategies for perfect reconstruction of binary signals from a dispersive communication channels. Touri et al. have developed (Touri, Voulgaris, & Hadjicostis, 2006) deterministic worst case framework for perfect reconstruction of discrete data transmission through a dispersive communication channel. Few adaptive equalizers have been suggested using soft computing tools such as ANN, PPN and the FLANN in late 90s. It has been reported that these methods are best suited for nonlinear and complex channels. Chebyshev Artificial Neural Network has also been proposed for nonlinear channel equalization (Patra, Poh, Chaudhari, & Das, 2005). The drawback of these methods is that the estimated weights may likely fall to local minima during training. Joint-processing adaptive nonlinear equalizer based on a pipelined recurrent neural network (JPRNN) using a modified real-time recurrent learning (RTRL) algorithm is proposed in Zhao et al. (2011a). Adaptive decision feedback equalizer (DFE) with the combination of finite impulse response (FIR) filter and functional link neural network (CFFLNDFE) is introduced in Zhao et al. (2011b). In both of these papers it is shown that improvement in

performance is in expense of increased complexity. To overcome this complexity, Zhao et al. (2011c), Zhao and Zhang (2009), Zhao, Zeng, He, Jin and Li (2012), Zhao, Zeng and Zhang (2010a), Zhao, Zeng, Zhang and Li (2010b) proposed some of typical forms of ANN based equalizers.

Despite of complexity problems, ANN remains as one of best tools for the problem of equalization (Abiyev, Okyay, Tayseer, & Fakhreddin, 2011; Panigrahi, Santanu, & Sasmita, 2008a; Zhao & Zhang, 2009; Zhao et al., 2010a; Zhao et al., 2010b; Zhao et al., 2012; Zhao et al., 2011a, 2011b, 2011c). However, because traditional training algorithms fail in many cases, there is a recent trend to train ANN with bio-inspired optimization algorithms for different applications (Chau, 2006; Yogi, Subhashini & Satapathy, 2010). PSO has been an increasingly applied in the area of computational intelligence. Also, it lends itself as being applicable to a variety of optimization problems. Evolutionary algorithms are also successful in the training of Artificial Neural Networks (ANN) Yogi et al. (2010), Lee and Lee (2012), Lin and Liu (2009), Lin and Chen (2011), Hong (2008), Potter, Venayagamoorthy, and Kosbar (2010). Interestingly, applications of PSO to ANN training in these works have only been used to find optimal weights of a given network. But, there is also a need for the appropriate topology and transfer functions of the neuron. The PSO algorithm optimizes all the variables, and hence capable of optimizing network weights and network parameters (may be variables). As a result, we can make use of PSO to optimize all parameters of a network, i.e. the number of layers, input and hidden neurons, the type of transfer

↑ Corresponding author. Mobile: +91 9437165431.

E-mail address: dr.ravinarayan@gift.edu.in (Dr. R N Panda).

functions etc. Hence, this paper focuses on optimizing the weights, transfer function, and topology of an ANN constructed for channel equalization.

Recurrent Neural Networks (RNN) used in Potter et al. (2010) for channel prediction using hybrid and variants of PSO uses only for weight optimization. Moreover, here complexity becomes very high because of use of so many optimization algorithms and becomes dead slow because of use of Differential Evolution (DE). However, the advancement made in this paper over existing work can be seen as optimizing the weights, transfer function, and topology of an ANN based channequalizer.

Though grammatical swarm optimization (de Mingo López, Blas, & Arteta, 2012) can be used to obtain a neural network topology, topology of ANN model used in this paper is decided using training algorithm discussed in later sections in this paper. The real essence of the proposed equalizer is its performance that outperforms contemporary ANN (Zhao et al., 2011a, 2011b, 2011c) based and Neuro-fuzzy (Abiyev et al., 2011; Panigrahi et al., 2008a) equalizers available in the literature.

The organization of the paper is as follows: Section 2 discusses the problem statement followed by the proposed system model in Section 3. For performance evaluation, the simulation study is carried out which is dealt in Section 4. Finally conclusion of the paper is outlined in Section 5.

2. Problemstatement

A model of digital transmission system is depicted in Fig. 1.

Impulse response of channel & co-channel can be represented as (Panigrahi, Santanu, & Sasmita, 2008b)

$$H_i(z) = \sum_{j=0}^{p_i-1} a_{ij} z^{-j} \tag{1}$$

Here p_i and a_{ij} are length and tap weights of i th channel impulse response. We assume a binary communication system, which would make the analysis simple, though it can be extended to any communication system in general. The transmitted symbols $x_i(n)$, $0 \leq i \leq n$ for channel and co-channel are drawn from a set of independent, identically distributed (i.i.d) dataset comprising of $\{\pm 1\}$ and these are mutually independent. This satisfies the condition

$$E\{x_i(n)\} = 0 \tag{2}$$

$$E\{x_i(n) x_j(n-k)\} = \delta_{ij} \delta_{n-k} \tag{3}$$

where $E\{ \cdot \}$ represents the expectation operator and $\delta_{ij} = 1$ if $i=j$ and 0 otherwise.

$$d_{co}(n) = \sum_{i=1}^M a_{ij} x_i(n-k) \tag{4}$$

The channel output scalars can be represented as

$$y(n) = \sum_{i=1}^M d_{co}(n) + g(n) \tag{5}$$

Here $d(n)$ desired received signal, $d_{co}(n)$ is interfering signal and $g(n)$ is noise component assumed to be Gaussian with variance

$E\{g^2(n)\} = \sigma_g^2$ and uncorrelated with data. The desired and interfering signal can be represented as

$$d(n) = \sum_{j=0}^{p-1} a_{0j} x_0(n-j) \tag{6}$$

$$d_{co}(n) = \sum_{i=1}^M \sum_{j=0}^{p_i-1} a_{ij} x_i(n-j) \tag{7}$$

The task of the equalizer is to estimate the transmitted sequence $x_0(n-k)$ based on channel observation vector, $y(n) = [y(n), y(n-1), \dots, y(n-m+1)]^T$, where m is order of equalizer and k is decision delay.

The error $e(n)$ can be expressed as:

$$e(n) = d(n) - y(n) \tag{8}$$

Since $e^2(n)$ is always positive and represents the instantaneous power of the difference signal, chosen as cost function instead of $e(n)$. The objective of an adaptive algorithm is to change the equalizer weights iteratively so that $e^2(n)$ is minimized iteratively and subsequently reduced to zero.

3. System model

System model for the proposed equalizer in this paper is a multi layer ANN, where the network is trained for optimized value with use of PSO. System model for the proposed equalizer consists of an ANN where all its neurons trained with PSO. Technical advantages of the paper are evidenced from its novelty and performance results.

PSO has been an increasingly applied in the area of computational intelligence. Also, it lends itself as being applicable to a variety of optimization problems. PSO is also successful in the training of Artificial Neural Networks (ANN).

This section first provides a brief idea on PSO and ANN for the ease of the reader in following two subsections. Then, the PSO model of Ribeiro and Schlansker (2004) as proposed for channel equalization discussed in next subsection.

Artificial Neural Networks

Artificial Neural Networks (ANN), are artificial models of the human brain. Human brain is capable of adapting to changing situations and learns quickly in the correct context. ANN works on simulation of the human brain. At their basic level ANN consist of an interconnected network of neurons and synapses.

Neurons are the fundamental element of an ANN. Neurons accepts inputs from other neurons and produce an output by firing their synapse. Neurons perform a weighted sum on all of their inputs and then the result goes through a transfer function to produce an output. ANNs are organized into layers. There is an input layer, an output layer and sometimes one or more hidden layers. The hidden layers are the root of the ANN that performs the actual computations of the network.

A network is comes into force when it is given with a set of inputs and the output layer produces the desired result. Weights of neurons may be different. Similarly, transfer function of different neurons may be different (usually they are the same). Training of ANN is required to facilitate the proper arrangement of a network.

Training neural networks

Out of several methods of training the ANN, back-propagation is the most common one. An ANN is trained by a set of data that consists of inputs and a desired output. The training steps are:

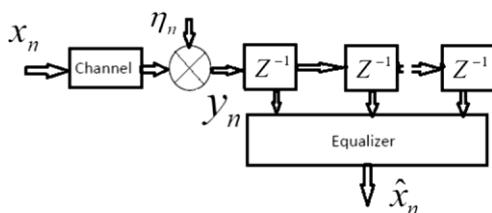


Fig. 1. Model of digital transmission system.

1. Read in the inputs and expected outputs
2. Compute the result by weighted sum of inputs and passing through the transfer functions
3. Compare the result with desired result
4. Compute and update fitness value based on comparison.
5. Repeat steps 2 and 3 until all training points are finished
6. Adjust weights in the appropriate direction to optimize fitness.
7. Repeat 1-6 until acceptable fitness value is found

The Back-propagation method Rumelhart, Geoffrey, & Ronald, 1986 is a gradient type adjustment for weight modification and may take an extremely long time to train a network. In this paper, we suggest the use of PSO as a training algorithm.

Transfer functions

Each of the neurons associated with a transfer function that operates on the input. The input of a neuron is the weighted sum of its inputs. A good transfer function is the sigmoid function.

The sigmoid function (9) maps the input to the range [0, 1], and given as:

$$Sigmoid(x) = \frac{1}{1 + e^{-ax}} \tag{9}$$

Particle Swarm Optimization

PSO Kennedy & Eberhart, 1995, del Valle, Venayagamoorthy, Mohagheghi, Hernandez, & Harley, 2008 is a population based search algorithm and is inspired by natural habits of bird flocking and fish schooling. In this paper, PSO is used to train ANN to be used as a channel equalizer.

PSO exploits the cooperation aspect and applies it to engineering optimization problems. The particles simply follow a predefined set of rules. PSO computes the particles based on a fitness function and finds a particle with a good solution. The particle with the best fitness is chosen as teacher. All other particles then learn from this best particle. No two particles are same and still each learns the attributes of other that will help to improve their fitness.

PSO model

A Particle Swarm is a population of individuals, where, each one influences the neighbors, have contribution in some features for the problem. Any single particle can be a possible solution, defined by its position, of the problem. Particles move through the problem space and adjust their path based on influences from other particles.

Each particle is randomly initialized to a certain position in the problem space. The number of dimensions in the problem space is equal to the number of components there are to optimize. If, p and v represent position and velocity vector, a particle updates its position according to the Euler integration equation for physical movement given as:

$$p_{i,t+1} = p_{i,t} + v_{i,t+1} \tag{10}$$

Velocity of the particle is computed from its current velocity (randomly initialized) and the velocity of the best particle in its neighborhood, including two stochastic variables. One of the two stochastic variables takes care of the portion of the velocity vector corresponding to its previous velocity, while the other one takes care of the portion corresponding to the velocity of the best particle. The resultant, generally, is a constant Kennedy & Eberhart, 1995. Update rule of the velocity vector is:

Here, q_1 and q_2 are two random constants corresponding to social and cognitive behavior of particle. For the problem given, a population of particles (around 20-50) are randomly initialized. Then they allowed moving in the problem space in search of an optimal or near optimal solution. The particles continue to move till they reach the desired position, i.e., global best solution.

PSO seems to be a better method of training ANN as compared to the traditional methods. PSO does not just train one network, rather, trains a network of networks. PSO forms a number of ANN and initializes all weights to arbitrary random values and trains each one. PSO compares fitness each network's. The network with the best fitness is chosen to be the teacher network (the global best). This network trains the others to update themselves forgetting their personal error or fitness.

Each neuron contains a position and velocity. The position corresponds to the weight of a neuron. The velocity is used to update and control the position (weight). If a particular neuron is far away from the global best position, then it will learn adjustment of its weight from a neuron which is closer to the global best.

Here, the particles are the individual networks not the neurons.

The number of neurons in the network defines the dimension of the hyperspace. Hence, location of the network in the problem hyperspace is effectively defined by the positions of each neuron in a network.

There may be a number of maxima and minima in the problem hyperspace. Particle swarm around in the hyperspace, updating

their position as seen from best position found by their neighbor particles. Finally, a particle will reach the optimal position. When they reach that point, it will continue to move towards the global optima. Other particles will quickly learn this and adjust their positions accordingly towards this optima. This ensures that a team of particles covered the optima area. Training of the network stops, if this optimum fitness is acceptable,

Failing to reach the global optimal positions, the position of the neurons are once again randomized and the swarm restarts. There are number of solutions for a real-number ANN. In PSO, it is assured that the network will never converge to false maxima. For solution hunting, PSO takes on two major methods, namely, exploration and exploitation. Exploration is the generalized search for maxima and minima. This occurs with a larger population moving over the entire problem space.

Exploitation is the convergence on a particular maxima or minima. Then, exploration starts to examine it for a second time. Update during exploitation is with lesser speed than in exploration. This is done by a smaller step size and also taken care that there won't be overstep to a possible optima. This part is an addition to original PSO and monitored by an annealing factor. As used in this paper, this annealing factor starts with 1 and decreases while moving towards optimal position, i.e. taking smaller steps.

Constructing a network swarm

This research focuses around training ANN with PSO. To do this, a population of networks must be constructed. Here, each ANN, as per the weights of the network ingredients, is treated as a particle in problem space. Usually, a 20-networks population works well. PSO neighborhoods is formed by this population and initialized. This construction is called the topology of the swarm system.

Training the swarm of neural networks

The training process is as follows:

- Examine over the training data and record the sum of the network errors, for each network.

- To get the best network in the problem space, compare all of the errors.
- If any network has reached the minimum error desired, record its weights and exit the program.
- Else, run PSO to update position and velocity vectors of each network.
- Repeat from step1.

If the required fitness achieved by a particle, indicating that a solution has obtained, then this particle changes from being a student searching solution to a teacher in production of ANN. Channel Equalization is a complex problem with a large number of control variables. ANN has proven to be excellent tools for the problem. The proposed solution is to construct and train ANN to predict the channel state.

The training data consisted of $[\pm 1]$. A network was constructed which used these values as inputs to the network. Network fitness, given in (12) was determined to be the mean square of errors for the entire training set, where error is defined to be the recorded symbol minus the network predicted symbol.

$$Fitness = \frac{1}{N} \sum_{i=1}^N (RecordedValue - NetworkpredictedValue)^2 \quad (12)$$

A network is deemed usable once it has met some minimal requirements for performance. The requirement used was the statistical calculation known as the multiple correlation coefficients and is given as:

$$R^2 = \frac{1}{N} \frac{Fitness}{\delta RecordedValue - MeanRecordedValue} \quad (13)$$

This measurement subtracts, from unity, the network fitness divided by the square of the mean subtracted by each output. As the network becomes more accurate, the resulting value approaches one.

As mentioned earlier, a multi layer ANN is used for channel equalizer. PSO is used to train the parameters of the network. Each of the parameters, i.e., weight, topology, transfer function etc. were trained with PSO as mentioned earlier that PSO trains the network as a whole. While going for a pseudo-code, it is assumed that ANN first acts as an administrator to provide resources (i.e., which parameter is to be optimized) to PSO which acts as a teacher for student ANN learning the equalization problem. Pseudo-code for the problem is provided in Fig. 2. In the pseudo-code, N and M represent number of particles and number of hidden nodes respectively. There is a 5:1 ratio of ants to candidate network topologies.

4. Simulations

For the simulations, value of N is set at 25 and value of M initialized with 5. The velocity factors are 0.8 for the inertial constant, 2 for the cognitive constant, and 2 for the social constant. Number of iterations and allowable error are set at 1000 and 10^{-3} , respectively.

For Evaluation of the performance of the proposed equalizer, simulation examples are presented in this section. Two different comparisons are used in these examples.

The widely used channel Liang & Zhi, 2004 is used for simulations. The system transfer function of this 3rd order channel model is:

$$H(z) = \frac{1 - 0.9z^{-1} + 0.385z^{-2} + 0.771z^{-3}}{\delta} \quad (14)$$

This channel have system zeros at 0.6 and $0.75 \pm j0.85$. To illustrate the effect of nonlinearity on the equalizer performance, nonlinear channel models with the following nonlinearity are introduced.

$$y(n) = \tanh(x(n)) \quad (15)$$

In the examples, the channel input signals are selected as i.i.d. sequences having zero mean. The additive channel noise is modeled

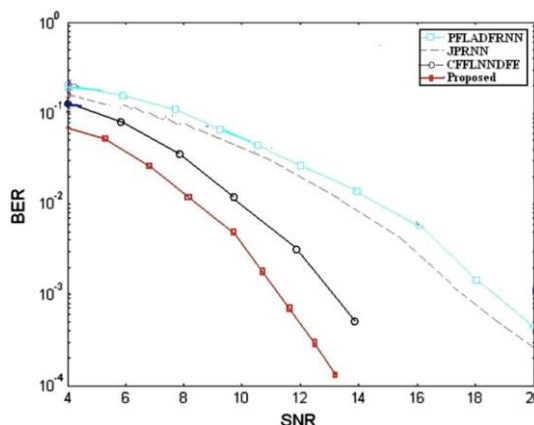


Fig. 3. $\log_{10}(\text{BER})$ vs. SNR (in dB) performance for ANN based equalizers.

as complex white Gaussian processes with zero-mean and independent to the channel input. It is noted that the symbol error rate (or, Bit Error Rate (BER)) is the final test for communication performance evaluation, and that the equalizer output noise variance is directly related to the symbol error rate via the complementary error function (The noise is assumed to have a Gaussian probability density function.). More specifically, the probability of BER is related to SNR at the equalizer output.

Comparison with other neural network based equalizers

For comparisons with other neural network based equalizers available in the literature, equalizers proposed in Zhao et al.

confirmed that the proposed equalizer converges better than other ANN based equalizers.

Comparison with Neuro-fuzzy equalizer

(2011a, 2011b, 2011c), i.e., PFLADFRNN Zhao et al. (2011a), JPRNN Zhao et al. (2011b) and CFLNDFE Zhao et al. (2011c) were simulated along with proposed equalizer to evaluate BER under similar conditions discussed above and resulting plot is shown in Fig. 3. Simulation runs up to the point when we receive 100 symbol errors at the output of the equalizer. The BER has been evaluated as 100 divided by total symbols sent during the simulation. Convergence characteristics of ANN based equalizers are shown in Fig. 4.

It is seen from Fig. 3 that the proposed equalizer outperforms equalizers proposed in literature, PFLADFRNN, JPRNN and CFLNDFE at all noise conditions. From Fig. 4, it is further

For comparisons with Neuro-fuzzy equalizers available in the literature, type-2TSKFNS as discussed in Abiyev et al. (2011) and Neuro-fuzzy equalizer trained with GA of Panigrahi et al. (2008a) simulated along with proposed equalizer to evaluate BER under similar conditions discussed above and resulting plot is shown in Fig. 5. Simulation runs up to the point when we receive 100 symbol errors at the output of the equalizer. The BER has been evaluated as 100 divided by total symbols sent during the simulation. Convergence characteristics of ANN based equalizers are shown in Fig. 6. It is seen from Fig. 5 that the proposed equalizer outperforms both type-2TSKFNS and GA trained NFN at all noise conditions. From Fig. 6, it is further confirmed that the proposed equalizer converges better than other Neuro-fuzzy equalizers.

5. Conclusion

This paper proposed an ANN based equalizer trained with PSO. As compared to other ANN based equalizers as well as Neuro-fuzzy equalizers the proposed equalizer performs better in all noise conditions.

Contributions of the paper can be outlined as:

- Development of a learning method for optimization of network topology of ANN
- Use of PSO trained ANN in channel equalization
- This article paves a way for future works utilizing other nature inspired algorithms for training of ANN based equalizers.
- This article paves a way for future works utilizing grammatical swam, that also can be used for optimization of ANN topology, for training of ANN based equalizers.
- This article also paves a way for future works utilizing PSO and other nature inspired algorithms for training of ANN for other practical applications those can be formulated either as an optimization or as a classification problem.

References

- Abiyev, R. H., Okyay, K., Tayseer, A., & Fakhreddin, M. (2011). A type-2 neuro-fuzzy system based on clustering and gradient techniques applied to system identification and channel equalization. *Applied Soft Computing*, 11, 1396–1406.
- Chau, K. W. (2006). Particle swarm optimization training algorithm for ANNs in stage prediction of Shing Mun River. *Journal of Hydrology*, 329, 363–367.
- de Mingo López, L. F., Blas, N. G., & Arteta, A. (2012). The optimal combination: Grammatical swarm, particles warm optimization and neural networks. *Journal of Computational Science*, 3(1-2), 46–55.
- del Valle, Y., Venayagamoorthy, G. K., Mohagheghi, S., Hernandez, J.-C., & Harley, R. G. (2008). Particle swarm optimization: basic concepts, variants and applications in power systems. *IEEE Transactions on Evolutionary Computation*, 12, 171–195.
- Hong, W.-C. (2008). Rainfall forecasting by technological machine learning models. *Applied Mathematics and Computation*, 200(1), 41–57.
- Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proc. IEEE int. conf. neural networks* (pp. 39–43).
- Lee, C.-H., & Lee, Y.-C. (2012). Nonlinear systems design by a novel fuzzy neural system via hybridization of electromagnetism-like mechanism and particles warm optimization algorithms. *Information Sciences*, 186(1), 59–72.
- Liang, J., & Zhi, D. (2004). FIR channel estimation through generalized cumulants like weighting. *IEEE Transactions on Signal Processing*, 52(3), 657–667.
- Lin, C.-J., & Chen, C.-H. (2011). Nonlinear system control using self-evolving neural fuzzy inference networks with reinforcement evolutionary learning. *Applied Soft Computing*, 11(8), 5463–5476.
- Lin, C.-J., & Liu, Y.-C. (2009). Image backlight compensation using neuro-fuzzy networks with immune particle swarm optimization. *Expert Systems with Applications*, 36(3, Part 1), 5212–5220.
- Panigrahi, S. P., Santanu, K. N., & Sasmita, K. P. (2008a). A genetic-based neuro-fuzzy controller for blind equalization of time-varying channels. *Int. J. Adapt. Control Signal Process.*, 22, 705–716. Wiley Interscience.

- Panigrahi, S. P., Santanu, K. N., & Sasmita, K. P. (2008b). Hybrid ANN reducing training time requirements and decision delay for equalization in presence of co-channel interference. *Applied Soft Computing*, 8, 1536–1538.
- Patra, J. C., Poh, W. B., Chaudhari, N. S., & Das, A. (2005). Nonlinear channel equalization with QAM signal using Chebyshev artificial neural network. In *Proc. of international joint conference on neural networks* (pp. 3214–3219), Montreal, Canada.
- Potter, C., Venayagamoorthy, G. K., & Kosbar, K. (2010). RNN based MIMO channel prediction. *Signal Processing*, 90(2), 440–450.
- Ribeiro, P. F., & Schlansker, W. K. (2004). A hybrid particle swarm and neural network approach for reactive power control. <<http://webapps.calvin.edu/~pribeiro/courses/engr302/Samples/ReactivePower-PSO-wks.pdf>>.
- Rumelhart, D. E., Geoffrey, E. H., & Ronald, J. W. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088), 533–536.
- Touri, R., Voulgaris, P. G., & Hadjicostis, C. N. (2006). Time varying power limited preprocessing for perfect reconstruction of binary signals. In *Proc. of the 2006 American control conference* (pp. 5722–5727). Minneapolis, USA.
- Voulgaris, P. G., & Hadjicostis, C. N. (2004). Optimal processing strategies for perfect reconstruction of binary signals under power-constrained transmission. In *Proc. IEEE conference on decision and control*, (Vol. 4, pp. 4040–4045). Atlantis, Bahamas.
- Yogi, S., Subhashini, K. R., & Satapathy, J. K. (2010). A PSO based functional link artificial neural network training algorithm for equalization of digital communication channels. In *Int. conf. on industrial and information systems* (pp. 107–112).
- Zhao, H., & Zhang, J. (2009). Adaptively combined FIR and functional link neural network equalizer for nonlinear communication channel. *IEEE Transactions on Neural Networks*, 20(4), 665–674.
- Zhao, H., Zeng, X., & Zhang, J. (2010a). Adaptive reduced feedback FLNN nonlinear filter for active control of nonlinear noise processes. *Signal Processing*, 90(3), 834–847.
- Zhao, H., Zhang, X., Zhang, J., & Li, T. (2010b). Nonlinear adaptive equalizer using a pipelined decision feedback recurrent neural network in communications systems. *IEEE Transactions on Communications*, 58(8), 2193–2198.
- Zhao, H., Zeng, X. P., He, Z. Y., Jin, W. D., & Li, T. R. (2012). Complex-valued pipelined decision feedback recurrent neural network for nonlinear channel equalization. *IET Communications*, 6(9), 1082–1096.
- Zhao, H., Zeng, X., Zhang, J., Li, T., Liu, Y., & Ruan, D. (2011c). Pipelined functional link artificial recurrent neural network with the decision feedback structure for nonlinear channel equalization. *Information Sciences*, 181, 3677–3692.
- Zhao, H., Zeng, X., Zhang, J., Li, Y., Wang, X., & Li, T. (2011a). A novel joint-processing adaptive nonlinear equalizer using a modular recurrent neural network for chaotic communications systems. *Neural Networks*, 24, 12–18.
- Zhao, H., Zeng, X., Zhang, X., Zhang, J., Liu, Y., & Wei, T. (2011b). An adaptive decision feedback equalizer based on the combination of the FIR and FLNN. *Digital Signal Processing*, 21, 679–689.