JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 22, 1163-1176 (2006)

# A New ISF-FLANN Channel Equalizer for 4QAM Digital Communication Systems and Its FPGA Verification

WAN-DE WENG, CHIN-TSU YEN AND RUI-CHANG LIN
*Graduate School of Engineering Science and Technology*
*National Yunlin University of Science and Technology*
*Touliu, Yunlin, 640 Taiwan*

A new improved soft-feedback functional link artificial neural-network (ISF-FLANN) based nonlinear channel equalizer is proposed in this paper. By using the functional expansion utilities, the ISF-FLANN does not need the hidden layers, which are existed in most of the multilayer perceptron network (MLP)-based equalizers. So the ISF-FLANN exhibits much simpler structure and thus requires less amount of computation during the training mode. We find that the use of soft feedback can greatly improve the performance of our previous work on FLANN structure [13]. The comparison of the average transmission symbol error rates (SER) of the ISF-FLANN with the linear transversal filters (LTF) and the traditional FLANN based on FPGA verification are presented. Simulation results demonstrate that ISF-FLANN outperforms FLANN by about 2 to 3 dB, and is about 6 to 7 dB better than LTF. The learning curves (LC) show that our design well fits the real-time processing requirement for 4QAM modern digital communication systems.

***Keywords:*** nonlinear channel equalizers, neural network, LTF, MLP, FLANN, ISF-FLANN

## 1. INTRODUCTION

Equalizers are usually used to compensate the received signals which are corrupted by the inevitable noise, interference and signal power attenuation introduced by communication channels during transmission [1]. Traditionally linear transversal filters (LTF) [2] are commonly used in the design of channel equalizers. The linear equalizers, however, fail to work well when transmitted signals have encountered severe nonlinear distortion. A neural network (NN) has the capability of complicatedly mapping the input to the output signals, which makes the NN-based equalizers a potentially suitable solution to deal with nonlinear channel distortion. Neural networks have been successfully used in various applications, such as pattern classification, speech recognition, cryptography and equalizers. In this paper we mainly focus on the design of nonlinear channel equalizer for 4QAM modern digital communication systems.

Several NN structures and training algorithms have been used in the design of channel equalizers. The neural network was first applied to channel equalizers in digital communication systems by Siu [3]. It is mentioned in [3] that the bit error rate (BER) and mean-square-error (MSE) performance of MLP is much better than LTF. And the performance of MLP is less sensitive to the variation of the learning gain. Theoretically, as

Received May 24, 2004; revised September 8, 2004 & March 7, 2005; accepted April 13, 2005.
Communicated by Liang-Gee Chen.

the number of nodes inside the MLP structure is increasing, the network will achieve better ability in approaching arbitrary functional mappings. The increasing nodes, however, tend to force the equalizer output to trap in local minima instead of the global one [4]. Zerguine has proposed an MLP decision feedback equalizer with lattice filter to overcome the local minimum problem [5]. Although this structure possesses superior performance than MLP, the network is much more complicated and demands for more computational resources. Meanwhile, some published articles (e.g., [6]) have mentioned the application of genetic algorithms (GA) in solving the local minimum problem. Theoretically, GA uses reproduction, crossover and mutation techniques based on biological genetics, and is indeed able to determine the global optimum. Unfortunately, the determination of the global minimum would require the network to widely search and test over the signal space. This generally not only lowers the learning speed of the network, but also makes the circuit difficult to realize. So the algorithm may not meet the real-time processing requirement of a modern communication system. Cha has used adaptive complex radial basis function networks (RBF) to deal with the channel equalization [7]. But as the RBF needs a large number of hidden nodes to achieve acceptable system performance, it is not quite suitable for parallel processing. The problem of huge number of hidden nodes encountered by RBF seems to be solved by using the minimum radial basis function (MRBF) neural networks proposed by Jianping [8]. But in the equalization procedure of a system applying MRBF, the network has to first increase the number of hidden nodes, and then omits the unnecessary nodes according to rules defined in the algorithm. Since the chip size of circuit depends on the maximum number of nodes along the equalization process, the MRBF technique cannot help in simplifying the hardware design. It is claimed in [9] that a support vector machine (SVM)-based equalizer can achieve optimal decision boundaries for a given training sequence. But as the channel noise is generally random and time-varying, even a well-trained neural network will still incorrectly classify the received signals. So its demand for the huge computational amount during the calculation of the optimal results does not seem quite worthy. In 1999, Patra utilized functional link artificial neural networks (FLANN) to build the nonlinear channel equalizer [10] and the identification of nonlinear dynamic systems [11]. The basic principle of an FLANN is to expand the dimensionality of the input signal space by using a set of linearly independent functions. The expansion can produce complicated decision boundaries at the output space, so the FLANN is capable of dealing with linear inseparable problems. As FLANN has a two-layer structure, its circuit is generally simpler than MLP, and thus achieves faster processing speed. It is pointed out in [10] that the FLANN outperforms MLP and the polynomial-perceptron structure (PPN) based on Voltera filters [12].

In our previous work [13], we have realized an FLANN channel equalizer using FPGA. We have found that, although FLANN exhibits better performance than MLP, it still has some potential drawbacks. Specifically, if we want to further improve the BER performance, we will need to enlarge the dimensionality of its input signal space. This will significantly increase the number of nodes in the input layer, and thus the circuit may become too complicated to be practical. This motivates us to design an NN-based nonlinear channel equalizer under the consideration of tradeoffs among the hardware chip size, the processing speed and the cost. In this paper we present a modified structure called improved soft feedback FLANN (ISF-FLANN). The ISF-FLANN structure feeds

the soft feedback signals directly into the input layer of the FLANN-based neural network. The modification not only reduces the error-propagation-delay caused by the hard-decision feedback in the decision-feedback FLANN (DF-FLANN) [14], but also greatly simplifies the hardware structure. We have shown that the computational complexity of ISF-FLANN is only about 10% more than that of FLANN, but it can save as much as 55% that of DF-FLANN. With the slightly increasing complexity, the ISF-FLANN achieves a much better error performance than FLANN. Simulation results show that the error performance of ISF-FLANN can be up to 2 to 3 dB better than FLANN, and is about 6 to 7 dB better than LTF [13]. This demonstrates that the performance of ISF-FLANN is quite close to the DF-FLANN. In this paper we will present the new ISF-FLANN channel equalizer and its FPGA verification.

## 2. ARCHITECTURE

### 2.1 System Architecture

The block diagram of the digital communication system for 4-QAM modulated signals is shown in Fig. 1, where the block labeled 'NL' represents the nonlinearity introduced by the channel. At time instant $kT$, the transmitted complex symbol is $t(k) = t_{k,i} + jt_{k,q}$, where $T$ denotes the symbol duration. The subscripts $i$ and $q$ represent the in-phase and quadrature components of the complex symbol, respectively. The transmitted data bits are assumed to have statistically independent and equally probable values from {1, –1}. The nonlinearly distorted output $b(k)$ associated with the input $a(k)$ can be written as $b(k) = \phi(a(k))$, where $\phi(\cdot)$ is the nonlinear function associated with the block labeled as 'NL'. We also assume the channel is affected by additive white Gaussian noise (AWGN) with variance $\sigma^2$. So the received signal is $r(k) = b(k) + q(k)$, where $q(k)$ is the AWGN sample at time instant $kT$. The compensated output $\hat{y}(k)$ from the equalizer is compared with the desired signal. The error signal is defined as $e(k) = y(k) - \hat{y}(k)$, where the desired signal $y(k) = t(k - D)$ represents the delayed version of the received signal, and $D$ is the time delay of the signal transmitted through the physical channel. The error signal $e(k)$ is used to modify the internal parameters of the network according to the training algorithm.

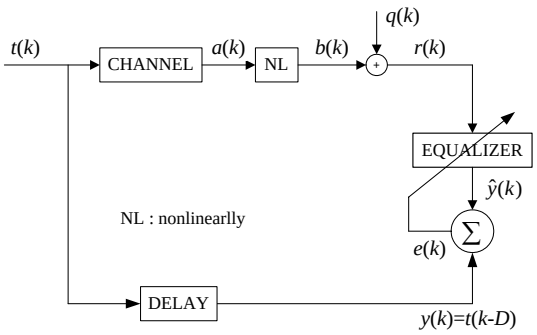



Fig. 1. A digital communication system with equalizer.

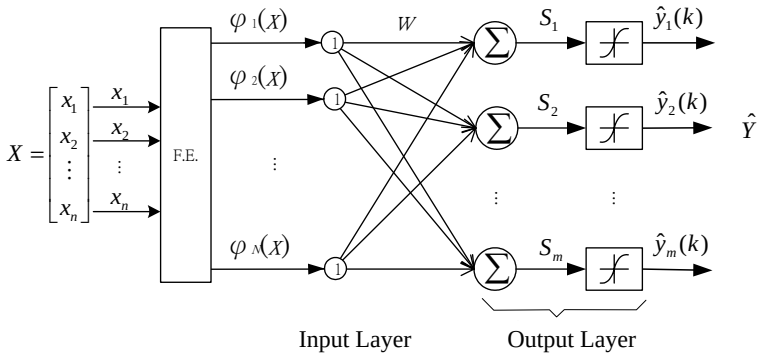



Fig. 2. FLANN structure.

### 2.2 FLANN Structure

The block diagram of a system with FLANN is shown in Fig. 2, where the block labeled F.E. denotes a functional expansion. The functional expansion maps the input signal vector $X = [x_1\ x_2\ \ldots\ x_n]^T$ through $N$ linearly independent functions $\Phi = [\varphi_1(X)\ \varphi_2(X)\ \ldots\ \varphi_N(X)]^T$. The linear combination of these function values can be written in its matrix form, that is, $S = W\Phi$, where $S = [s_1\ s_2\ \ldots\ s_m]^T$, and $W$ is the $m \times N$ dimensional weighting matrix. The matrix $S$ is fed into a bank of identical nonlinear functions to generate the equalized output $\hat{Y} = [\hat{y}_1\ \hat{y}_2\ \ldots\ \hat{y}_m]^T$, where $\hat{y}_j = \rho(s_j)$, $j = 1, 2, \ldots, m$. Here the nonlinear function is defined as $\rho(\cdot) = \tanh(\cdot)$. The major difference between the hardware structures of MLP [13] and FLANN is that FLANN has only input and output layers, and the hidden layers are completely replaced by the nonlinear mappings. In fact, the task performed by the hidden layers in MLP is carried out by functional expansions in FLANN. Since the input signals are nonlinearly mapped into the output signal space, FLANN has also the ability to resolve the equalization problems for nonlinear channels. The FLANN uses BP algorithm [15] to train the neural networks. The advantage of the BP algorithm is its hardware circuit can be easily realized. By the way, when a received signal $x$ has a complex value, the phase distortion caused by the sigmoid transfer function $f(x) = 1/\{1 + \exp(-x)\}$ can be avoided. Since the FLANN has much simpler structure than MLP, its speed of convergence for training process is a lot faster than MLP. The whole learning algorithm for the FLANN is summarized as follows [10, 15]:

$$\hat{y}_j(k) = \rho\left(\sum_{i=1}^{N} w_{ji}(k)\varphi(X_k)\right) = \rho(W_j(k)\Phi(X_k)),\ j = 1,\ 2,\ \ldots,\ m, \tag{1}$$

where $X_k = [x_1(k)\ x_2(k)\ \ldots\ x_n(k)]^T$, and $W_j(k) = [w_{j1}(k)\ w_{j2}(k)\ \ldots\ w_{jN}(k)]$, $\Phi(X_k) = [\varphi_1(X_k)\ \varphi_2(X_k)\ \ldots\ \varphi_N(X_k)]^T$

$$\Delta(k) = \delta(k)(\Phi(X_k))^T \tag{2}$$

$$\Delta W(k) = \mu\Delta(k) + \gamma\Delta(k-1) \tag{3}$$

$$W(k+1) = W(k) + \Delta W(k), \tag{4}$$

where $\delta(k) = [\delta_1(k)\ \delta_2(k)\ \ldots\ \delta_m(k)]^T$, and $\delta_j(k) = (1-\hat{y}_j^2(k))e_j(k)$, $j = 1, 2, \ldots, m$, and $e_j(k) = y_j(k) - \hat{y}_j(k)$, and $W(k) = [W_1(k)\ W_2(k)\ \ldots\ W_m(k)]^T$.

In the above equations, $\mu$ is the learning factor and $\gamma$ is the momentum factor that helps to accelerate the speed of convergence of algorithms.

### 2.3 ISF-FLANN Structure

Here we propose an improved soft feedback functional link artificial neural-networks (ISF-FLANN) structure to lower down the hardware cost without sacrificing the system performance. In the ISF-FLANN structure the feedback output signals are directly fed into the input layer of the neural network instead of being taken as the input signals, which is its major difference with the DF-FLANN [14] as given in Fig. 3. In other words, the input layer signals of the ISF-FLANN are composed of the functional expansions $\Phi = [\varphi_1(X)\ \varphi_2(X)\ \ldots\ \varphi_N(X)]^T$ from the received signals $X = [x_1\ x_2\ \ldots\ x_n]^T$ as well as the feedback signal $Z_l = [\hat{Y}_{(k-1)}\ \hat{Y}_{(k-2)}\ \ldots\ \hat{Y}_{(k-l)}]^T$ directly obtained from the output, where $l$ represents the number of stages in the feedback delay path. The linear combinations of these signals are nonlinearly transformed by $\rho(\cdot) = \tanh(\cdot)$ to attain the estimate of the current received data. The structure of an ISF-FLANN is shown in Fig. 4.

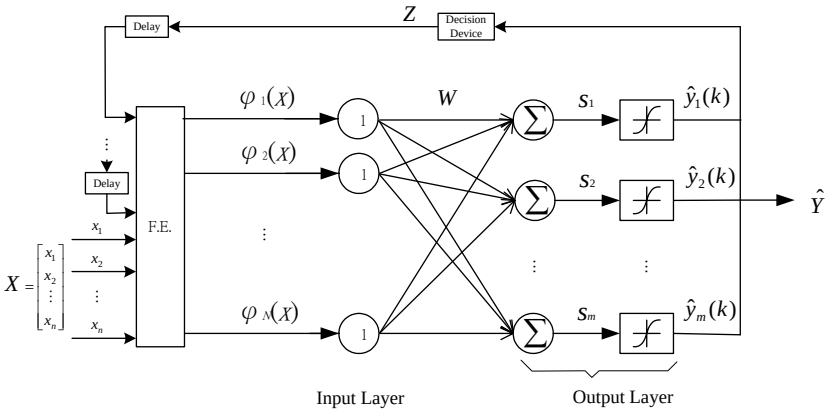
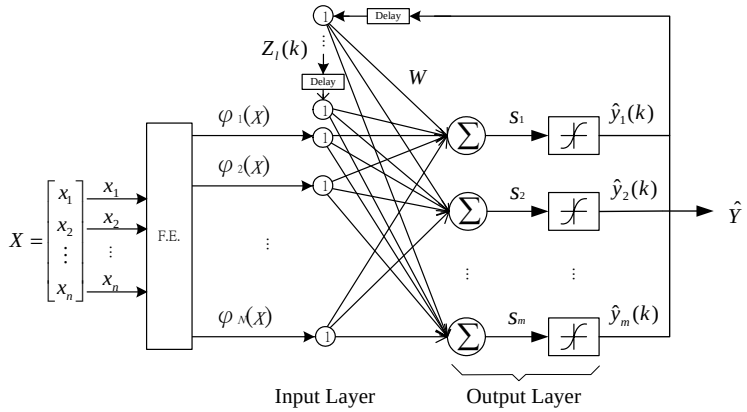



Fig. 3. DF-FLANN structure.

We can see that the ISF-FLANN has a two-layer structure, so only the weighting coefficients between these two layers have to be updated with the algorithm in the training process. This compactness of layers can help to decrease the learning time and as a result it generally makes the overall speed of convergence faster. The learning algorithm for ISF-FLANN is derived as follows. From Fig. 4 we know that

$$\hat{y}_j(k) = \rho\left(\sum_{i=1}^{M} w_{ji}(k)\psi_i(k)\right) = \rho(W_j(k)\Psi(k)), j = 1, 2, \ldots, m, \text{ and } M = N + l \tag{5}$$




F.E.=Functional Expansion

Fig. 4. ISF-FLANN structure.

where $\Psi(k) = [\hat{Y}_{(k-1)}\ \hat{Y}_{(k-2)}\ \ldots\ \hat{Y}_{(k-l)}\ \varphi_1(X_k)\ \varphi_2(X_k)\ \ldots\ \varphi_N(X_k)]^T = [\psi_1(k)\ \psi_2(k)\ \ldots\ \psi_M(k)]^T$, and $W_j(k) = [w_{j1}(k)\ w_{j2}(k)\ \ldots\ w_{jM}(k)]$. Now we define the error function associated the $k$-th training vector as

$$E(k) = \frac{1}{2}\sum_{j=1}^{m} e_j^2(k), \tag{6}$$

where $e_j(k)$ denotes the difference between $y_j(k)$ and $\hat{y}_j(k)$. We use the steepest-descend algorithm to minimize $E(k)$, that is,

$$w_{ji}(k+1) = w_{ji}(k) - \mu\frac{\partial E(k)}{\partial w_{ji}(k)},\ j = 1, 2, \ldots, m,\ i = 1, 2, M, \tag{7}$$

where $\mu$ is the learning rate of the network. According to the chain rule, we have

$$\frac{\partial E(k)}{\partial w_{ji}(k)} = \frac{\partial E(k)}{\partial \hat{y}_j(k)}\frac{\partial \hat{y}_j(k)}{\partial s_j(k)}\frac{\partial s_j(k)}{\partial w_{ji}(k)} = -e_j(k)(1-\hat{y}_j^2(k))\psi_i(k) = -\delta_j(k)\psi_i(k), \tag{8}$$

where $\delta_j(k) = e_j(k)\ (1-\hat{y}_j^2(k))$. Plugging (8) into (7), we get

$$w_{ji}(k+1) = w_{ji}(k) + \mu\delta_j(k)\psi_i(X_k). \tag{9}$$

The result can be written in matrix form as $W(k+1) = W(k) + \mu\delta(k)(\Psi(k))^T$, where $\delta(k) = [\delta_1(k)\ \delta_2(k)\ \ldots\ \delta_m(k)]^T$ and $W(k) = [W_1(k)\ W_2(k)\ \ldots\ W_m(k)]^T$. To further improve the speed of the convergence, we add a momentum term when updating $W$, that is, $W(k+1) = W(k) + \mu\Delta(k) + \gamma\Delta(k-1)$, where $\Delta(k) = \delta(k)(\Psi(k))^T$, and $\gamma$ is the momentum factor.

The major purpose of feeding the feedback signals directly into the input layer of the neural network instead of the functional expansion blocks is to reduce the number of nodes in the input layer. This property can greatly help in reducing the complexity of the circuit.

## 3. DESIGN PROCEDURE

The procedure of the equalizer during one iteration can be divided into the following four steps:

(i) Compute the estimated output $\hat{y}$ of the network in the forward direction.
(ii) Evaluate the errors $e$ between the signals from the output layer and the input layer.
(iii) Calculate the amount of modification $\Delta W$ for the weightings between layers.
(iv) Update the weighting vector $W$ for each layer.

The computational complexities of LTF, PPN, DF-FLANN, ISF-FLANN, FLANN, and MLP for training the neural network during each iteration are summarized in Table 1, where $n_0^+$ represents the number of input signals of DF-FLANN, ISF-FLANN, FLANN and PPN structure, $n_0$ is the number of nodes in LTF and the MLP input layer, $n_i$, $i$ = 1, 2, …, $L$ − 1, denotes the number of nodes in the $i$-th hidden layer, $n_L$ stands for the number of nodes in the output layer, and ${}_nC_m = n!/m!(n - m)!$ is the combinatorial operator. For the particular case specified in the previous paragraph, the computational complexities of these structures are given in Table 2. We see that the computational complexity of ISF-FLANN is only about 10% more than that of FLANN, and is about 50% and 45% of the amount required by MLP and DF-FLANN, respectively. We may note that the

**Table 1. Computation complexity of various NN structures.**

| Number of operations | addition | multiplication | tanh(·) | cos(·) and sin(·) |
|---|---|---|---|---|
| LTF | $2n_0n_L + n_L$ | $3n_0n_L$ | 0 | 0 |
| PPN | $3(3n_0^+ + {}_{n_0^+}C_2)n_L + 4n_L$ | $4(3n_0^+ + {}_{n_0^+}C_2)n_L + 2n_L + {}_{n_0^+}C_2$ | $n_L$ | 0 |
| DF-FLANN | $3(5n_0^+ + {}_{n_0^+}C_2)n_L + 4n_L$ | $4(5n_0^+ + {}_{n_0^+}C_2)n_L + 2n_L + {}_{n_0^+}C_2$ | $n_L$ | $2n_0^+$ |
| MLP | $3\sum_{i=0}^{L-1} n_i n_{i+1} + 2\sum_{i=1}^{L} n_i - n_0 n_1 + 2n_L$ | $6\sum_{i=0}^{L-1} n_i n_{i+1} + 3\sum_{i=1}^{L} n_i - n_0 n_1 + 2n_L$ | $\sum_{i=1}^{L} n_i$ | 0 |
| ISF-FLANN | $3(3n_0^+ + {}_{n_0^+}C_2 + l \cdot n_L)n_L + 4n_L$ | $4(3n_0^+ + {}_{n_0^+}C_2 + l \cdot n_L)n_L + 2n_L + {}_{n_0^+}C_2$ | $n_L$ | $2n_0^+$ |
| FLANN | $3(3n_0^+ + {}_{n_0^+}C_2)n_L + 4n_L$ | $4(3n_0^+ + {}_{n_0^+}C_2)n_L + 2n_L + {}_{n_0^+}C_2$ | $n_L$ | $2n_0^+$ |

**Table 2. Computation complexity for a special case.**

| Number of operations | addition | multiplication | tanh(·) | cos(·) and sin(·) |
|---|---|---|---|---|
| LTF | 34 | 48 | 0 | 0 |
| PPN | 116 | 154 | 2 | 0 |
| DF-FLANN | 278 | 379 | 2 | 12 |
| MLP | 136 | 290 | 10 | 0 |
| ISF-FLANN | 128 | 170 | 2 | 8 |
| FLANN | 116 | 154 | 2 | 8 |

complexity of LTF is comparatively much lower than the other structures, but it possesses a major drawback of being unable to deal with linear non-separable problems.

The normalized responses of the channel models considered in this paper are expressed in the form of their $Z$-transformation as follows:

CH = 1: 1.0,
CH = 2: $0.447 + 0.894z^{-1}$,
CH = 3: $0.341 + 0.876\, z^{-1} + 0.341z^{-2}$.

Where CH = 1 corresponds to an ideal channel with unit impulse response that has negligible intersymbol interference (ISI). The model CH = 2 is a channel with non-minimum phase. And CH = 3 stands for a finite-impulse-response (FIR) channel with channel length 3. Following the ISI models we consider three kinds of nonlinearities:

NL = 0: $b(k) = a(k)$,
NL = 1: $b(k) = \tanh(a(k))$,
NL = 2: $b(k) = a(k) + 0.2a^2(k) - 0.1a^3(k)$.

The nonlinear model NL = 0 stands for a purely linear module, that is, there is no nonlinearity in the model. The model NL = 1 corresponds to systems suffering from nonlinear distortion which is possibly caused by the saturation of amplifiers used in the transceivers. In models NL = 2 we assume the signals have suffered from some second-order and third-order nonlinear distortions during transmission.

The original input signal for FLANN and PPN $X(k) = [r_{k,i}\ r_{k,q}\ r_{k-1,i}\ r_{k-1,q}]^T$ is expanded into 18 dimensional patterns by functional expansion [10]. These patterns include 4 original signals, the sin(·) and cos(·) transformations of these four signals, and 6 outer product terms generated by the four original signals. By adding a unit biasing signal, the numbers of nodes in the input and output layers become 19 and 2, respectively. In ISF-FLANN the original input signals are also expanded into 18 terms. So the total number of input layer nodes is 21, including the feedback signals $[\hat{Y}_{k-1,i}\ \hat{Y}_{k-1,q}]^T$ and the unit biasing value. The input to the DF-FLANN is composed of the received signals from the channel $X(k) = [r_{k,i}\ r_{k,q}\ r_{k-1,i}\ r_{k-1,q}]^T$ and the feedback from the decision device output $[Z_{k-1,i}\ Z_{k-1,q}]^T$. These signals are expanded into 45 terms, which include the 6 original signals, the trigonometric function values sin($x$), sin(2$x$), cos($x$) and cos(2$x$) of these signals, and their outer product terms. Besides, a unit biasing signal is included in the input layer signal. Therefore the total numbers of nodes in the input and output layers are 46 and 2, respectively. The number of nodes in the input, hidden, and output layers for MLP is chosen to be 4-8-2. The choice of the number of nodes in the hidden layer is based on our observation on its effects on the system performance. We have found that increasing the number of nodes in the hidden layer can improve the performance. But when we choose more than 8 nodes, the system is only slightly improved. The nonlinear function at each node in the hidden and output layers is chosen to be tanh(·). Two sets of four-input tapped delay lines are applied in the input layer of LTF, of which the corresponding input signal is $X(k) = [r_{k,i}\ r_{k,q}\ r_{k-1,i}\ r_{k-1,q}\ \ldots\ r_{k-3,i}\ r_{k-3,q}]$. In the simulation, 500 source signals are randomly generated and passed through the nonlinear channel model. The channel output is fed into the neural network to train the neural network. After 100 iterations of training,

we send 10000 source symbols through the channel for testing. The above procedure is independently repeated 100 times to calculate the SER of the whole system. In Fig. 5 we present the SER curves under CH = 3 and NL = 2 for various equalizers ever mentioned in this paper. We can see that ISF-FLANN and DF-FLANN have almost the same performance and are about 2 to 3 dB better than FLANN, PPN and MLP. The LTF-based equalizer is much worse than all the others and is obviously not a proper choice for nonlinear channel equalization. Summarizing the consideration about the cost and performance analysis, we can suggest using the ISF-FLANN in the equalizer. Fig. 6 presents the learning curves of ISF-FLANN, DF-FLANN and FLANN when they are operated in the condition CH = 3, NL = 0 and SNR = 12 dB. We can see that the speed of convergence for ISF-FLANN and DF-FLANN are almost the same, and ISF-FLANN exhibits lower MSE than DF-FLANN.

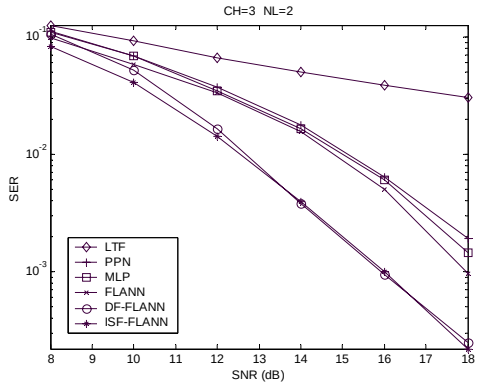



Fig. 5. SER vs. SNR of various channel equalizers, for CH = 3, NL = 2.

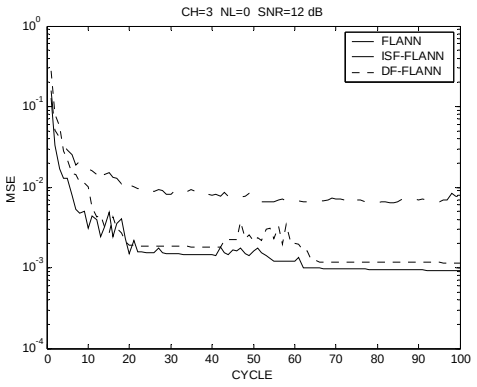



Fig. 6. Learning curves for CH = 3, NL = 0, SNR = 12 dB.

The only limitation on the choice of $\mu$ and $\gamma$ in the BP algorithm is that these parameters have to be in some prescribed range. Their values will only affect the speed of convergence of the algorithms. But a careful selection of the parameters will help to simplify the hardware structure and can largely reduce the number of multiplications. For ISF-FLANN, we have found from software simulation that increasing the length of the decimal fraction of the weightings does not significantly improve the performance of the equalizer when it is over 12 bits. And the symbol error rate of the equalizer will dramatically increase when the length is below 12 bits. The relationship between the error performance and the number of bits in the fraction part is given in Fig. 7.

## 4. FPGA VERIFICATION

The generalized module of an equalizer is depicted in Fig. 8. The Source generator (SG) is responsible for generating the input layer signals in a neural network. An SG is generally composed of a series-in-parallel-out (SIPO) shift register and a Functional Expansionor. The trigonometric operations sin($x$) and cos($x$) are built in a read-only-memory (ROM) look-up table (LUT). Besides, multiplier is used to generate outer product term.

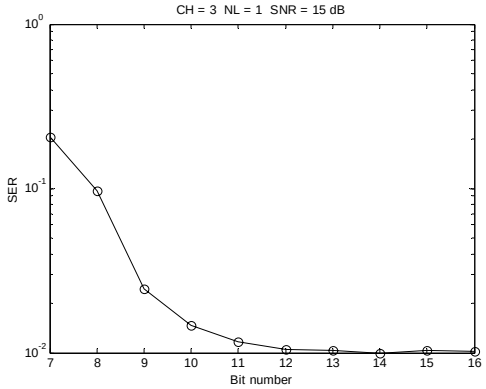



Fig. 7. Finite precision vs. SER of ISF-FLANN for CH = 3, NL = 1, SNR = 15 dB.

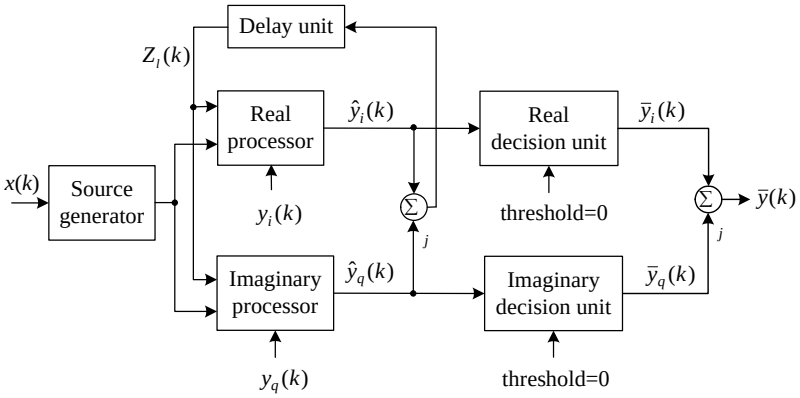



Fig. 8. Generalized module of an equalizer.

The real and the imaginary processors respectively deal with the estimation of $y_i(k)$ and $y_q(k)$. And then the real and the imaginary decision units transfer the values of the estimates $\hat{y}_i(k)$ and $\hat{y}_q(k)$ into $\bar{y}_i(k)$ and $\bar{y}_q(k)$, whose values take on 1 or − 1, depending on whether the real and imaginary processor outputs are positive or negative. The outputs of the real and the imaginary decision units are then combined to form $\bar{y}(k) = \bar{y}_i(k) + j\bar{y}_q(k)$. The detailed structure of the Processor Unit for ISF-FLANN is shown in Fig. 9. The major task of the Matrix operator is to compute $S(k) = W(k)[\Phi(X(k)Z_l)]^T$ and $\delta(k)[\Phi(X(k)Z_l)]^T$. The nonlinear function $\rho(\cdot)$ is carried out in the output estimator, where the nonlinear transformation is replaced by a ROM LUT. The Weight refresher for ISF-FLANN structure is used to update $W$ and $\Delta W$. To prevent overflow, a limiter is included in the Weight refresher to strictly restrict the weights to be always in the range (– 32, + 32). Another point worth mentioning is that the values of $\mu$ and $\gamma$ are both chosen to be 1/4 in our design. By doing this, the operations $\mu \times \Delta(k)$ and $\gamma \times \Delta(k-1)$ can be performed by simple shifting instead of direct multiplying the two operands. So the chip area as well as the processing time can be largely reduced. The FPGA chip we choose for our circuit implementation is EP20K600E produced by Altera Company. It is composed of 24000 logic elements and its clock rate is set to be 50 MHz. The length of the weight

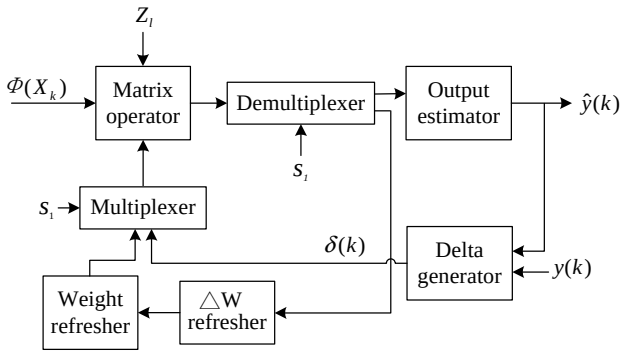



Fig. 9. Detailed block diagram of ISF-FLANN processor unit.

**Table 3. hardware simulation results for the ISF-FLANN and the FLANN.**

| | number of LEs | iteration time | recalling time | FPGA |
|---|---|---|---|---|
| LTF | 7238 | 500 ns | 220 ns | EPF10K200SRC240-1 |
| FLANN | 18804 | 500 ns | 220 ns | EP20K600E |
| ISF-FLANN | 20329 | 500 ns | 220 ns | EP20K600E |

is chosen to be 18 bits, which is composed of 1 sign bit, a 5-bit integer part, and 12 bits of fractional part. Table 3 summarizes the number of logic elements, the iteration time, and the recalling time required by the ISF-FLANN, FLANN and LTF equalizers [13]. We see that the hardware cost of ISF-FLANN is only about 8% more than FLANN. Although the number of logic elements required by LTF appears much smaller, the poor ability of equalizing nonlinear channels will greatly limit the applications of LTF. The ISF-FLANN can be easily fitted into a single Altera Apex 20k series FPGA chip. The clock rate is set to be 50 MHz, that is, the duration of each clock is 20 ns. The iteration and recalling time for each data symbol are 25 and 11 clock durations, respectively. The overall speeds of training and recalling can thus be up to 2 Mbps and 4.5 Mbps, respectively.

In the simulation we first randomly generate 500 complex signals from the discrete ensemble {– 1, 1} as the input data sequence. After passing through the nonlinear channels, the received signals are used for training the neural network. To see the equalizer's capability of real-time processing, the network is trained for 10 cycles. Fig. 10 presents the learning curve of decoding error rate (DER) for the ISF-FLANN structures under various channel nonlinearity conditions. In the simulation we arbitrarily choose CH = 2 and SNR = 12 dB. It can be easily seen that ISF-FLANN can achieve convergence within 5 training cycles.

In our simulation the neural network is trained for ten cycles, and 500 symbols are used for verification during each cycle. At the end of the training process, 2000 testing symbols are input for calculating the symbol error rate. The system performance for CH = 3 and NL = 2, that is, the transmission environment assuming some second-order, third- order nonlinear distortions and severe nonlinear ISI, is shown in Fig. 11. We can see that ISF-FLANN outperforms FLANN by about 2 to 3 dB, and is about 5 to 6 dB better than LTF [13].

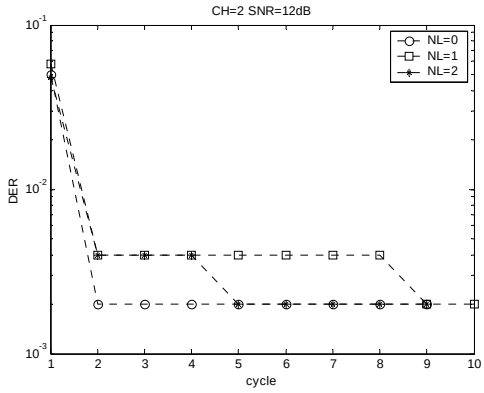



Fig. 10. Learning curves of DER for CH = 2, SNR = 12 dB.

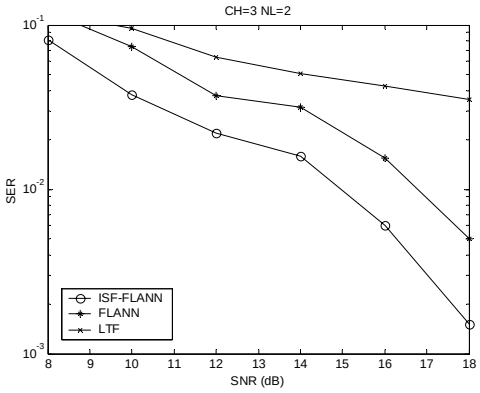



Fig. 11. SER vs. SNR, for CH = 3, NL = 2.

## 5. CONCLUSIONS

In this paper, we have proposed a new ISF-FLANN based nonlinear channel equalizer. The ISF-FLANN inherits the advantages of the FLANN so that it has simpler structure than MLP and DF-FLANN. Moreover, the feedback mechanism utilized in the ISF-FLANN structure can greatly speed up the convergence of network settings during the training process. We have verified that an equalizer exploiting ISF-FLANN structure achieves much smaller steady-state error at the output. The simulation results about the SER behavior have shown that the ISF-FLANN outperforms the FLANN by about 2 to 3 dB. And the performance of LTF is so poor that it is rarely used in nonlinear channel equalization. These statements purport that the ISF-FLANN possesses an improved ability of manipulating linearly inseparable problems than FLANN. So we conclude that the ISF-FLANN structure is very suitable for applications exploiting real-time processing and is potentially a better choice for implementing nonlinear channel equalizers.

## REFERENCES

1. J. G. Proakis, *Digital Communications*, 4th ed., McGraw-Hill, New York, 2001.
2. S. U. H. Qureshi, "Adaptive equalization," in *Proceedings of the IEEE*, Vol. 73, 1985, pp. 1349-1387.
3. S. Siu, G. J. Gibson, and C. F. N. Cowan, "Decision feedback equalization using neural network structures and performance comparison with standard architecture," in *Proceedings of the IEEE*, Vol. 137, 1990, pp. 221-225.
4. S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed., Englewood Cliffs, Prentice Hall, New Jersey, 1999.
5. A. Zerguine, A. Shafi, and M. Bettayeb, "Multilayer perceptron-based DFE with lattice structure," *IEEE Transactions on Neural Networks*, Vol. 12, 2001, pp. 532-545.
6. C. Lu, B. Shi, and L. Chen, "BG: a hybrid algorithm for MLP," *The 2000 IEEE Asia-Pacific Conference on Circuits and Systems*, 2000, pp. 360-363.

7. I. Cha and S. Kassam, "Channel equalization using adaptive complex radial basis function networks," *IEEE Journal on Selected Area Communications*, Vol. 13, 1995, pp. 122-131.
8. D. Jianping, N. Sundararajan, and P. Saratchandran, "Communication channel equalization using complex–valued minimal radial basis function neural networks," *IEEE Transactions on Neural Networks*, Vol. 13, 2002, pp. 687-696.
9. D. J. Sebald and J. A. Bucklew, "Support vector machine techniques for nonlinear equalization," *IEEE Transactions on Signal Processing*, Vol. 48, 2000, pp. 3217-3226.
10. J. C. Patra, R. N. Pal, R. Baliarsingh, and G. Panda, "Nonlinear channel equalization for QAM signal constellation using artificial neural networks," *IEEE Transactions on Systems, Man and Cybernetic – Part B*, Vol. 29, 1999, pp. 262-271.
11. J. C. Patra, R. N. Pal, B. N. Chatterji, and G. Panda, "Identification of nonlinear dynamic systems using functional link artificial neural networks," *IEEE Transactions on Systems, Man and Cybernetic – Part B*, Vol. 29, 1999, pp. 254-262.
12. S. Chen, G. J. Gibson, and C. F. N. Cowan, "Adaptive channel equalization using polynomial-perceptron structure," in *Proceedings of the IEE*, Vol. 137, 1990, pp. 257-264.
13. C. T. Yen, W. D. Weng, and Y. T. Lin, "FPGA realization of a neural network based nonlinear channel equalizer," *IEEE Transactions on Industrial Electronics*, Vol. 51, 2004, pp. 472-479.
14. A. Hussain, J. J. Soraghan, and T. S. Durrani, "A new adaptive functional-link neural-network-based DFE for overcoming co-channel interference," *IEEE Transactions on Communications*, Vol. 45, 1997, pp. 1358-1362.
15. C. You and D. Hong, "Adaptive equalization using the complex back propagation algorithm," *IEEE International Conference on Neural Networks*, Vol. 4, 1996, pp. 2136-2141.

**Wan-De Weng (翁萬德)** was born in Taipei, Taiwan, R.O.C., in 1963. He received the B.Eng. degree in Electrical Engineering from National Taiwan University, Taipei, Taiwan, R.O.C., the M.E. degree from the University of Utah, Salt Lake City, Utah, U.S.A., and the Ph.D. degree from Purdue University, West Lafayette, IN, U.S.A., in 1985, 1990, and 1993, respectively. In 1993, he joined the Department of Electrical Engineering, National Yunlin University of Science and Technology, Yunlin, Taiwan, R.O.C., as an Associate Professor. His research interests are signal processing in digital communication systems, VLSI design, and error control coding.

**Chin-Tsu Yen (顏景池)** was born in Yunlin, Taiwan, R.O.C, on October 2, 1963. He received the B.S. degree in Industrial Education from National Taiwan Normal University, Taipei, in 1986, and the M.S. degree in Electronic Engineering from National Taiwan Institute of Technology, Taipei, in 1992, and the Ph.D. degree in the Graduate School of Engineering and Technology from National Yunlin University of Science and Technology, Yunlin, Taiwan, in 2004. In 1993, he joined the Department of Electronic Engineering, National Siluo Agricultural Industrial High School, Yunlin, Taiwan, R.O.C., as an In structor. His research interests include signal processing, VLSI design, neural networks, and digital communications.

**Rui-Chang Lin (林瑞昌)** was born in Nantou, Taiwan, R.O.C., in 1955. He received the bachelor degree in Electrical Engineering from National Taiwan University of Science and Technology, Taipei, Taiwan, R.O.C. and the master degree from the Taiwan Normal University, Taipei, in 1983, and 1990, respectively. Mr. Lin is currently a doctorate candidate in the Graduate School of Engineering and Technology, National Yunlin University of Science and Technology. In 1990, he joined the Department of Electronic Engineering, Nankai Institute of Technology, Nantou, Taiwan, R.O.C., as an instructor. His research interests are neural network design and application in digital communication systems.