Adaptive Packet Equalization for Indoor Radio Channel Using Multilayer Neural Networks

Po-Rong Chang, *Member, IEEE*, Bao-Fuh Yeh, and Chih-Chiang Chang

Abstract—**This paper investigates the application of the multilayer perceptron structure to the packet-wise adaptive decision feedback equalization of a *M*-ary QAM signal through a TDMA indoor radio channel in the presence of intersymbol interference (ISI) and additive Gaussian noise. Since the multilayer neural networks are capable of producing complex decision regions with arbitrarily nonlinear boundaries, this would greatly improve the performance of conventional decision feedback equalizer (DFE) where the decision boundaries of conventional DFE are linear. However, the applications of the traditional multilayer neural networks have been limited to real-valued signals. To tackle this difficulty, a neural-based DFE is proposed to deal with the complex QAM signal over the complex-valued fading multipath radio channel without performing time-consuming complex-valued back-propagation training algorithms, while maintaining almost the same computational complexity as the original real-valued training algorithm. Moreover, this neural-based DFE trained by packet-wise back-propagation algorithm would approach an ideal equalizer after receiving a sufficient number of packets. In this paper, another fast packet-wise training algorithm with better convergence properties is derived on the basis of a recursive least-squares (RLS) routine. Results show that the neural-based DFE trained by both algorithms provides a superior bit-error-rate performance relative to the conventional least mean square (LMS) DFE, especially in poor signal to noise ratio conditions.**

I. Introduction

In recent years, the possibility of using radio for indoor data and voice communications within offices, manufacturing floors, warehouses, hospitals, and convention centers has become an attractive issue. It would drastically reduce wiring in a new building and provide the flexibility of changing or creating various communication services in existing buildings without the need for expensive, time-consuming rewiring. The relationship between behavior of radio-wave propagation and building architecture in a small indoor environment is usually characterized by a time-varying multipath fading cluster channel model [1]. In this model. the arriving paths are divided into clusters, formed by the building structure, and individual rays within the clusters, formed by objects in the vicinity of the transmitter and receiver. The arrivals of clusters and rays form Poisson processes with different rates. Data rates on the order of 10 Mbps are desirable for these indoor radio channels to make them compatible with the existing wired or cabled local area networks. Meanwhile, the severe multipath fading, which is characteristic of the indoor radio channels, limits the data rate. Moreover, the multipath fading often introduces an inter-symbol interference (ISI) which deteriorates the system performance.

Various techniques such as decision-feedback equalization (DFE) and adaptive equalization [2], [3] were developed to eliminate the channel impairments. The performance, however, is limited by the linearity of decision boundaries produced by such equalizers. Recently, Siu *et al.* [4] have effectively utilized neural network as adaptive equalizers for a simplistic finite impulse response (FIR) channel model. They demonstrated that the neural-based equalizer trained by the back propagation algorithms showed superior performance over conventional decision feedback equalizer because of its capability to form complex decision regions with nonlinear boundaries. Nevertheless, their applications have been limited to real-valued baseband channel models and binary signals. However, for indoor radio communication, the channel models and the information bearing signals are complex-valued. So, there is a great need to develop a neural network equalizer that can deal with higher level signal constellations, such as *M*-ary quadrature amplitude modulation (QAM), as well as with complex-valued channel models. QAM is a very effective technique to achieve a high bit-rate transmission without increasing the bandwidth. In Section II, we proposed a new neural-based decision-feedback equalizer to QAM systems over complex-valued channel without performing complex-valued back propagation algorithms. In Section III, a packet-wise neural equalization is introduced to track channel time variations and improve the system performance. Furthermore, it will be proven that the neural equalizer trained by packet-wise back propagation algorithm approaches an ideal equalizer after receiving a sufficient number of packets. In Section IV, an algorithm based on a recursive least-squares (RLS) routine is proposed to improve the computational efficiency of the back propagation and provide faster network training of the neural equalizers. Computer simulations are presented in Section V, and conclusions are outlined in Section VI.

Manuscript received November 4, 1993; revised March 8, 1994. This work was supported by the National Science Council Contract NSC 83-0404-E009-048.

The authors are with the Department of Communication Engineering, National Chiao-Tung University, Hsin-Chu, Taiwan.

IEEE Log Number 9403219.

II. Neural-Based Decision Feedback Equalization for Complex-Valued Channel

A feedforward neural network is a layered network consisting of an input layer, an output layer, and at least one layer of nonlinear processing elements. The nonlinear processing elements, which sum incoming signals and generate output signals according to some predefined function, are called neurons. In this paper, the function used by nonlinear neurons is called the sigmoidal hyperbolic tangent function G, which is similar to a smoothed step function

$$G(x) = \tanh(x). \quad (1)$$

The neurons are connected by terms with variable weights. The output of one neuron multiplied by a weight becomes the input of an adjacent neuron of the next layer.

A. The Architecture of Neural-Based Decision Feedback Equalizer

Before applying the multilayer feedforward neural networks to adaptive equalization problem, it is important to establish their approximation capabilities to some arbitrary nonlinear real-vector-valued continuous mapping $y = f(x): D \subseteq R^{n_i} \rightarrow R^{n_o}$ from input/output data pairs $\{x, y\}$ where D is a compact set on R. Consider a feedforward network $\mathrm{NN}(x, w)$ with x as a vector representing inputs and w as a parameter weight vector that is updated by some learning rules. It is desired to train $\mathrm{NN}(x, w)$ to approximate the mapping $f(x)$ as close as possible. The Stone–Weierstrass theorem [5] showed that for any continuous function $f \in C^1(D)$ with respect to x, a compact metric space, an $\mathrm{NN}(x, w)$ with appropriate weight vector w can be found such that $\|\mathrm{NN}(x, w) - f(x)\|_x < \epsilon$ for an arbitrary $\epsilon > 0$, where $\|e\|_x$ is the mean squared error defined by

$$\|e\|_x = \sum_{x \in D} \|e(x)\|^2, \text{ where } \|\cdot\| \text{ is the vector norm.} \quad (2)$$

For network approximators, key questions are how many layers of hidden units should be used, and how many units are required in each layer? Cybenco [6] has shown that the feedforward network with a single hidden layer can uniformly approximate any continuous function to an arbitrary degree of exactness provided that the hidden layer contains a sufficient number of units. However, it is not cost-effective for the practical implementation. Nevertheless, Chester [7] gave a theoretical support to the empirical observation that networks with two hidden layers appear to provide high accuracy and better generalization than a single hidden layer network, and at a lower cost (i.e., fewer total processing units). Since, in general, there is no prior knowledge about the number of hidden units needed, a common practice is to start with a large number of hidden units and then prune the network whenever possible. Additionally, Huang and Huang [8] gave the lower bounds on the number of hidden units which can be used to estimate its order.

As discussed above, the feedforward neural network results in a static network which maps static input patterns to static output patterns. However, the radio channel exhibits the temporal behavior where the output has a finite temporal dependence on the input. These temporal patterns in the input data are not recognizable by such a network. If the input signal is passed through a set of delay elements, the outputs of the delay elements can be used as the network inputs and temporal patterns can be trained with the standard learning algorithms of feedforward neural network. An architecture like this is often referred to as time delay neural network (TDNN). It is capable of modeling dynamical systems where their input–output structure has finite temporal dependence.

Generally, the received signal transmitted over the multipath channel can be governed by the following discrete-time difference dynamic equation:

$$r(t) = C(s(t), \cdots, s(t - n_D + 1)) \quad (3)$$

where $r(t)$, $s(t)$'s are the complex-valued received signal and the transmitted symbols, respectively; and n_D is the maximum lag involved in the multipath channel. The symbol $s(t)$ equals either 0 or 1 when the transmission is binary signaling. However, here $s(t)$'s are suggested to be in bipolar form $\{-1, 1\}$. In a general M-ary signaling system, the waveforms used to transmit the information are denoted by $\{q_m(t), m = 1, 2, \cdots, M\}$. It is possible to represent each symbol of the M-ary system by a $\log_2 M \times 1$ binary-state or bipolar-state vector, $s(t)$. Here, we are interested in QAM systems. The constellations have their signal points on a rectangular grid, at $(\{\pm 1\}, \{\pm 1\})$ for 4-QAM and $(\{\pm 1, \pm 1/3\}, \{\pm 1, \pm 1/3\})$ for 16-QAM, . . . etc. The location of any signal point may be assigned to a particular bipolar-state vector, $s(t)$. The correspondence between signal location and the values of components in the bipolar-state vector is not unique. However, this correspondence is usually an one-to-one mapping. For example, the 16-QAM signal locations $(-1, -1)$, $(1/3, -1)$ and $(1/3, 1/3)$ may be assigned to $[-1, -1, 1, -1]^T$ and $[1, -1, 1, -1]^T$ which correspond to the decimal representations, "2," "6," and "10," respectively. The above assignment is similar to the labeling of 16-QAM in the coded modulation techniques. Generally, the bipolar representation can be extended to any other M-ary systems, i.e., PSK, GMSK, and CPM. Equation (2) becomes a weighted linear sum of transmitted symbols, $s(t)$'s, when the radio channel does not include the nonlinear transmission medium. Thus, the transfer function of the multipath channel between the transmitter and receiver is denoted as $H(z)$, which is a FIR system with time-varying coefficients resulted from the moving objects within the indoor communication environment. $H(z)$ represents the reflected radio wave as well as the direct wave between the transmitter and receiver.

Widrow [9] showed that a causal infinite impulse response (IIR) filter can achieve a delayed version of the system inverse to $H(z)$. The inverse or the equalizer filter for the general channel model can be governed by the following IIR-type dynamic equation

$$\hat{s}(t) = \mathrm{EQ}\,(r(t), \cdots, r(t - n_f + 1), \hat{s}(t-1), \cdots, \hat{s}(t - n_b)) \quad (4)$$

where $\hat{s}(t)$ represents the equalized output signal or vector; n_f and n_b are maximum lags in the input and output respectively. It should be noted that the responses $\hat{s}(t)$ are identical to the transmitted symbols $s(t)$ when the equalizer is a perfect and ideal channel inverse which can compensate the undesired multipath effect completely. Moreover, to model the dynamics represented by (4), it is possible to convert the temporal sequence of radio frequency signal into a static pattern by unfolding the sequence over time and then use this pattern to train a static network. From a practical point of view, it is suggested to unfold the sequence over a finite period of time. This can be accomplished by feeding the input sequence into a tapped delay line of finite extent, then feeding the taps from the delay line into a static feedforward network. Thus, the channel inverse is achieved by training the static feedforward network. This can be referred to as inverse system identification. The basic configuration for achieving this is shown schematically in Fig. 1. The feedforward neural network based decision feedback equalizer is placed behind the channel and receives both the channel outputs and detected symbols as its inputs. The network inputs at time t can be represented by $\boldsymbol{x}_t = [\boldsymbol{r}_t^T, \bar{\boldsymbol{s}}_t^T]^T$, where $\boldsymbol{r}_t = [\boldsymbol{r}^T(t), \cdots, \boldsymbol{r}^T(t - n_f + 1)]^T$ and $\bar{\boldsymbol{s}}_t = [\bar{\boldsymbol{s}}^T(t - 1), \cdots, \bar{\boldsymbol{s}}^T(t - n_b)]^T$. Notice that the received complex-valued signals $\boldsymbol{r}(t)$ should be represented by a 2×1 vector, i.e., $[r_R, r_I]^T$, because the error back-propagation algorithms cannot be applied to the complex-valued inputs directly where r_R and r_I represent the real and imaginary parts of $r(t)$. The detected symbols $\bar{s}(t)$ are generated by feeding the static neural network outputs $\tilde{s}(t)$ through a hardlimiter at time instant t and given by $\bar{s} = \mathbf{sign}(\tilde{s}(t))$ where $\tilde{s}(t) = \text{NN}(\boldsymbol{x}_t; \boldsymbol{w})$. According to Fig. 1, the input-output relationship of the neural equalizer can be characterized by the function

$$\bar{s}(t) = \text{NNDFE}\ (\boldsymbol{x}_t; \boldsymbol{w}) = \mathbf{sign}(\tilde{s}(t)) = \mathbf{sign}(\text{NN}(\boldsymbol{x}_t; \boldsymbol{w})) \tag{5}$$

where $\boldsymbol{w}$ is the weight vector of the feedforward network, and $\bar{s}(t)$ is the estimate of $s(t)$.

The training data involved in transmitted symbols provide the desired response of the static feedforward network, $\boldsymbol{d}_t (= s(t))$ to train the network to approximate the perfect channel inverse or ideal equalizer $\text{EQ}_{\text{ideal}}(\cdot)$. Notice that a replica of the desired response is stored in the receiver. By the Stone–Weierstrass theorem, it is possible to find the appropriate weight vector $\boldsymbol{w}^*$ of the static feedforward network of the neural-based equalizer, such that

$$\|\text{NN}\ (\boldsymbol{x}_t; \boldsymbol{w}^*) - \text{EQ}_{\text{ideal}}(s_t)\|_{x_t} < \epsilon \tag{6}$$

for an arbitrary $\epsilon > 0$ and all the $\boldsymbol{x}_t$ in the region of interest.

Since $\boldsymbol{d}_t$ is represented by a bipolar-state vector, each component of $\text{NN}(\boldsymbol{x}_t; \boldsymbol{w}^*) (= \tilde{s}(t))$ becomes either -1 or 1 after a sufficiently long training period. This would imply that $\bar{s}(t) = \mathbf{sign}(\tilde{s}(t)) = \tilde{s}(t)$ or $\text{NNDFE}\ (\boldsymbol{x}_t; \boldsymbol{w}^*) = \text{NN}(\boldsymbol{x}_t; \boldsymbol{w}^*)$. From (6), we have

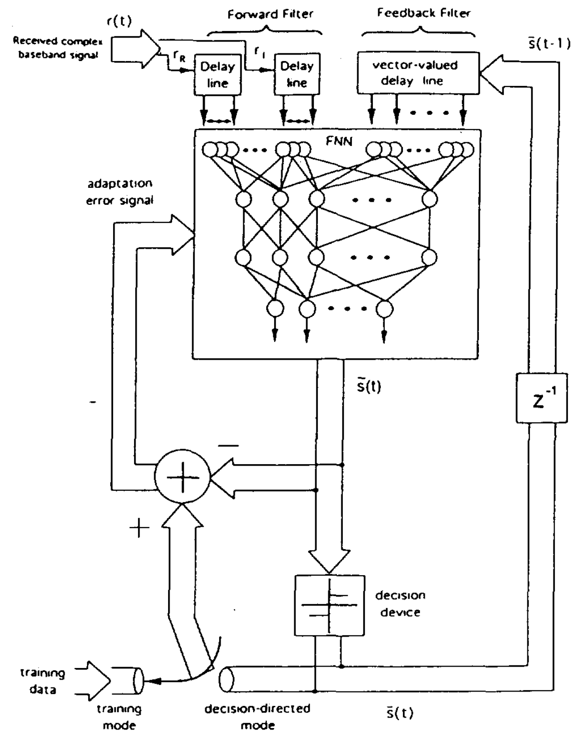

Fig. 1. Architecture of neural-based decision feedback equalizer for QAM system.

$$\|\text{NNDFE}\ (\boldsymbol{x}_t; \boldsymbol{w}^*) - \text{EQ}_{\text{ideal}}(\boldsymbol{x}_t)\|_{xt} < \epsilon \tag{7}$$

For a M-ary QAM signaling communication system, $s(t)$, $\tilde{s}(t)$, and $\bar{s}(t)$ should be represented by $\log_2 M \times 1$ vectors. Moreover, the received signal $r(t)$ is complex-valued and then can be expressed as a two-dimensional vector. Thus, the input layer of the network consists of a n_f-tap two-dimensional vector-valued forward filter, and a n_b-tap $\log_2 M$-dimensional vector-valued feedback filter. As a result, the number of neurons in the input layer is $n_i = 2 \times n_f + (\log_2 M) \times n_b$. In the output layer, the number of neurons is $n_o = \log_2 M$.

B. Feedforward Neural Networks and Their Learning Rules

In 1986, Rumelhart *et al.* [10] proposed a generalized delta rule known as back-propagation for training layered neural networks in a pattern-wise manner. In mathematical sense, the back-propagation learning rule is used to train the feedforward network NN (x, w) to approximate a function $f(x)$ from compact subset D of n_i-dimensional Euclidean space to a bounded subset $f(D)$ of n_o-dimensional Euclidean space. Let $\boldsymbol{x}_t$ which belongs to D be the tth pattern or sample and selected randomly as the input of the neural network at time instant t, let $\text{NN}(\boldsymbol{x}_t, \boldsymbol{w}) (= \boldsymbol{o}_t)$ be the output of the neural network, and let $\boldsymbol{f}(\boldsymbol{x}_t) (= \boldsymbol{d}_t)$ which also belongs to $\boldsymbol{f}(D)$ be the desired output.

This task is to adjust all the variable weights of the neural network such that the pattern-wise quadratic error Ξ_t can be reduced, where Ξ_t is defined as

$$\Xi_t = \frac{1}{2}\|\mathrm{NN}(\boldsymbol{x}_t, \boldsymbol{w}) - \boldsymbol{f}(\boldsymbol{x}_t)\|^2 = \frac{1}{2}\sum_{j=1}^{n_0} (o_{tj} - d_{tj})^2 \quad (8)$$

where n_0 is the number of output nodes, o_{tj} and d_{tj} are the jth components of $\boldsymbol{o}_t$ and $\boldsymbol{d}_t$, respectively.

Here, we define the weighted sum of the outputs of the previous layer by the presentation of input pattern $\boldsymbol{x}_t$:

$$\mathrm{net}_{tj} = \sum_i w_{ji} o_{ti} \quad (9)$$

where w_{ji} is the weight which connects the output of the ith neuron in the previous layer with respect to the jth neuron, and o_{ti} is the output of the ith neuron.

It should be noted that o_{ti} is equal to x_{ti} when the ith neuron is located in the input layer, where x_{ti} is the ith component of pattern $\boldsymbol{x}_t$. Using (1), the output of neuron j is

$$o_{tj} = \begin{cases} x_{tj} & \text{if the neuron } j \text{ belongs to the input layer} \\ G(\mathrm{net}_{tj}) & \text{otherwise.} \end{cases} \quad (10)$$

The pattern-wise or on-line back-propagation algorithm [10] minimizes the quadratic error Ξ_t by recursively altering the connection weight vector at each pattern according to the expression

$$w_{ji}(t+1) = w_{ji}(t) - \eta \left.\frac{\partial \Xi_t}{\partial w_{ji}}\right|_{w_{ji} = w_{ji}(t)} \quad (11)$$

where the learning rate η is usually set to be equal to a positive constant less than utility. An error signal term δ called delta produced by the jth neuron is defined as follows:

$$\delta_{tj} \triangleq -\frac{\partial \Xi_t}{\partial(\mathrm{net}_{tj})}. \quad (12)$$

Rumelhart [10] showed that the error signals δ's for all neurons in the network can be computed according to the following recursive procedure:

$$\delta_{tj} = \begin{cases} (d_{tj} - o_{tj})G'(\mathrm{net}_{tj}) & \text{if neuron } j \text{ belongs to the output layer} \\ G'(\mathrm{net}_{tj}) \sum_l \delta_{tl} w_{lj} & \text{otherwise.} \end{cases} \quad (13)$$

It should be mentioned that o_{ti} is equal to x_{ti} when neuron i belongs to the input layer. The expression of (13) is also called the generalized delta learning rule. Once those error signal terms have been determined, the partial derivatives for the quadratic error of the tth pattern can be computed directly by

$$\frac{\partial \Xi_t}{\partial w_{ji}} = \frac{\partial \Xi_t}{\partial \mathrm{net}_{tj}} \frac{\partial \mathrm{net}_{tj}}{\partial w_{ji}} = -\delta_{tj} o_{ti}. \quad (14)$$

Thus the update rule of the on-line back-propagation algorithm is

$$w_{ji}(t+1) = w_{ji}(t) + \eta \delta_{tj} o_{ti}. \quad (15)$$

In the traditional equalizer, a replica of the desired response is stored in the receiver. Naturally, the generator of this stored reference has to be electronically synchronized with the known transmitted sequence. A widely used training signal consists of a pseudonoise (PN) sequence of length N_B. Moreover, the training signal can also be expressed as a collection of input-output data pairs, $\{\boldsymbol{x}_t, \boldsymbol{o}_t\}_{t=1}^{N_B}$. The weights of neural-based equalizer are updated by using the batch of these data pairs. Thus, the objective function should be modified in an expression of summation, $\Xi = \Sigma_{t=1}^{N_B} \Xi_t$, during the initial training. Thus, the update rule becomes

$$w_{ji}^{\mathrm{new}} = w_{ji}^{\mathrm{old}} - \eta \frac{\partial \Xi}{\partial w_{ji}} = w_{ji}^{\mathrm{old}} + \eta \sum_{t=1}^{N_B} \delta_{tj} o_{ti}. \quad (16)$$

Notice that the quantity $\boldsymbol{w}(t+1)$ is the updated weight vector after one pattern of learning; $\boldsymbol{w}^{\mathrm{new}}$ is the updated weight vector after one batch of learning.

It is shown that the batch back-propagation learning algorithm is used to initialize the weight coefficients of the neural-based equalizer when the channel is unknown. Nevertheless, the on-line learning algorithm is used to adjust the weights to track channel time variations and said to be decision directed. However, from [11], the initial training can be executed by the on-line learning algorithm instead of batch learning since on-line learning is shown to approach batch learning provided that η is small. Initialization may be aided by the transmission of N_B known training symbols. The trained neural-based equalizer converges to the channel inverse when N_B is sufficiently large. It is known that the decision errors in equalizer tracking can lead directly to crashing of the equalizer, especially when the adaptation gain is high. Decision errors become more prevalent when the received signal-to-noise ratio is low, a condition that occurs unpredictably in fading channels. The susceptibility of adaptive equalizers or neural-based equalizers to crashes caused by propagation of decision errors implies that retraining procedures must be specified. For fading channel, periodic retraining is often used to improve reliability at some cost in throughput efficiency, via periodic insertion of training symbols into the data stream. More details about the packet adaptive equalization will be discussed in the next section.

III. Neural-Based Packet Adaptive Equalization for TDMA Wireless Channels

Packet equalization is a problem that arises in TDMA communication systems, in which data is transmitted in fixed-length packets, rather than continuously. It is usually assumed that the packet is largely self-contained for error detection, i.e., in terms of equalizer initialization and at least fine synchronization. This overhead can achieve good performance with reasonable complexity. Packet equalization has some similarities with block-oriented methods for periodic training on continuous channels, although it is assumed that the channel state is independent from packet to packet. Frequency and packet synchronization are assumed to be maintained once initialized, but symbol timing and phase synchronization typically need to be restored for each packet. It is known that the optimum approach to equalization is an off-line, noncausal batch processing of the received signal with a large amount of training data. Such a formulation would be infeasible and too complex to implement, but iterative approaches based on periodic training are possible. We will show that there is an equivalence of the off-line equalization and the packet-wise neural-based equalizer when the number of packets approaches infinity.

Considering the packet transmission, it is assumed that the length of the training data contained in the packet header and the total packet length are n_u and n_p, respectively. The main idea of the packet training scheme is used to train the neural-based equalizer with n_u training data for each packet. The neural-based equalizer can be retrained for every packet and thus track the time variations in the channel. This is quite similar to the on-line training for a sequence of packets.

The packet version of the back-propagation-based algorithm can be obtained by modifying (16), given by

$$w_{ji}^{\text{new}} = w_{ji}^{\text{old}} + \eta \sum_{t=1}^{n_u} \delta_{tj} o_{ti}. \tag{17}$$

Furthermore, during data transmission after packet training period, the decision-directed adaptation is executed by (15).

A. Global Convergence and Approximation Capability of Packet-Wise Back-propagation Algorithm

The most important feature of the packet-wise back-propagation algorithm is its global convergence property. It ensures that the algorithm can always find the optimal weight values of the neural equalizer from any arbitrary starting point in the training phase during the packet transmission. It has been shown in [11] that the batch back-propagation learning algorithm can exactly implement the gradient descent algorithm. Since the gradient descent algorithm is globally convergent, this implies that the batch back-propagation algorithm is also globally convergent.

Lemma 1: Equivalence of Batch and Packet Back-Propagation Algorithms Reference [11] also showed there is an equivalence of batch and on-line back-propagation algorithms. As discussed above, for packet back-propagation algorithms, their training procedures are executed in an on-line manner by inputting n_u training data for each packet to the algorithm. Similarly, from [11], it can be shown that the packet-wise on-line algorithm will approximately implement the batch algorithm after multiple sweeps through the training data if the learning rate η is small enough. In other words, the resulting connection weights obtained by the batch learning are approximately the same as packet learning results after packet learning goes through the whole batch of training data. In a mathematical sense, let, $A_{\text{batch}}^{(N_B)} \triangleq$ batch back-propagation algorithm with N_B training data, and, $A_{\text{packet}}^{(N, n_u)} \triangleq$ packet back-propagation algorithm with N packets, and n_u training data in each packet. From [11], $A_{\text{packet}}^{(N, n_u)}$ approaches $A_{\text{batch}}^{(N_B)}$ when the learning rate η is sufficiently small and $N_B = N \cdot n_u$. Thus by above discussion, $A_{\text{packet}}^{(N, n_u)}$ is globally convergent. Moreover, training data for each packet can be performed in either sequential adaptation of (15) or block adaptation of (17) since both adaptations yield the almost same result.

Furthermore, by Stone–Weierstrass theorem, the exact channel inverse can be obtained by the batch learning when the length of training data is large enough and the maximum lags in both input and output of channel inverse are known. Especially, for time-varying channels, N_B would approach infinity. But, it is infeasible for any channel equalizer transmitting a large amount of training data continuously. Fortunately, Lemma 1 shows that the problem is solved by inserting a finite number of training data into the data stream periodically.

Theorem 1: The neural equalizer is guaranteed to be capable of converging to the channel inverse globally by performing the packet back propagation algorithms with a sufficient number of hidden nodes when the maximum lags in the input and output of channel inverse are known.

From Lemma 1 and Stone–Weierstrass theorem, a solution w^* can be generated by $A_{\text{packet}}^{(N, n_u)}$ such that

$$\|\text{NNDFE}\,(x_t;\, w^*) - \text{EQ}_{\text{ideal}}(x_t)\|_{x_t} < \epsilon, \tag{18}$$

for an arbitrary $\epsilon > 0$, as $N \to \infty$.

IV. Efficient Learning Algorithms for Neural Networks

The back propagation algorithm is an useful method to find an optimal solution of a set of weight values that enables a network to perform a certain input–output mapping function. However, it is widely recognized that the back propagation suffers from the drawback of slow convergence. More recent work has produced the improved learning strategies based on a recursive least-squares algorithm (RLS) [12] and an extended Kalman routine [13]. Although these two algorithms were each derived independently based on a different approach, they are actually equivalent. They both use the same search direction called the Gauss–Newton direction for which the negative gradient is multiplied by the inverse of an approximate hessian matrix of the given criterion. This is a more efficient search direction than the steepest-descent approach of back propagation and it significantly improves the convergence performance. Reference [12] showed that their RLS algorithm called ELEANNE7 involves the tradeoff between computational complexity and convergence. The

TABLE I.
NUMBER OF ARITHMETIC OPERATIONS PER PATTERN ADAPTATION CYCLE

Algorithm	Multiplications	Additions
LMS	$2n_i + 1$	0
Back Propagation	$4n_h n_i + 4n_h n_o + 4n_o + 2n_h + 1$	$3n_h n_i + 4n_h n_o + 2n_o - n_h + 1$
ELEANNE7	$3n_h^2 + 4n_h n_i + 4n_h n_o + 4n_o + 3n_h + 7$	$2n_h^2 + 3n_h n_i + 4n_h n_o + 2n_o - n_h + 2$

ELEANNE7 algorithm is used to provide faster network training of our equalizer. Moreover, [14] showed that the packet version of ELEANNE7 can be obtained by modifying the RLS algorithm into a block-type RLS formulation.

Usually, the efficiency of a learning algorithm depends on the number of arithmetic operations required for each pattern adaptation cycle. Table I shows the number of arithmetic operations per adaptation cycle required by LMS, back propagation, and ELEANNE7 algorithms for a network with single hidden layer, where n_h is the number of hidden units. Similarly, the comparison of these algorithms for a network with more than one hidden layers in terms of their computational requirements can be found in [12]. For simplicity, we are interested in the analysis of three-layered neural networks.

Assuming that $n_h n_i >> n_0$ and $n_h^2 >> 1$, [12] showed that

$$\frac{M_{E7}}{M_{\text{BP}}} \approx 1 + \frac{3n_h + 1}{2(2n_i + 2n_0 + 1)} \tag{19}$$

where M_{E7} and M_{BP} are the number of multiplications required by ELEANNE7 and back propagation algorithms respectively. For M-ary QAM equalizer, $n_i = 2 \times n_f + (\log_2 M) \times n_b$ and $n_0 = \log_2 M$. Thus, (19) becomes

$$\frac{M_{E7}}{M_{\text{BP}}} \approx 1 + \frac{3n_h + 1}{2(4n_f + 2\log_2 M(n_b + 1) + 1)}. \tag{20}$$

In practical situations, $10 \leq n_h < 30$, $M \geq 4$ and $n_f = n_b = 3$ for the use of indoor radio channel [2], it can be shown that the ratio M_{E7}/M_{BP} is upper bounded by a value of 2.57. A similar analysis can be carried out for the number of additions per adaptation cycle required by back propagation and ELEANNE7 algorithms. Furthermore, one may use the fast RLS or RLS lattice algorithm to improve the computational complexity of ELEANNE7. In fact, the overall efficiency of a learning algorithm is evaluated on the basis of a criterion comprising the convergence rate achieved by the algorithm during the training and the number of arithmetic operations per adaptation cycle required by the algorithm. In next section, the performance of algorithms, in conjunction with their computational requirements, indicates that ELEANNE7 performs the network training faster than back propagation algorithm to reach a predetermined small value of the total error.

V. SIMULATION RESULTS

The indoor radio channel model used in performance evaluation is on the basis of a cluster channel model developed as a result of measurement [1]. Using the parameters in [1], i.e., 200 and 5 ns cluster and ray mean interarrival times and 60 and 20 ns cluster and ray power delay time constants, a mean rms delay spread of 44 ns was found over 10000 sets of independent portable-to-base station antenna impulse response realizations. Moreover, the channel output is corrupted by zero-mean additive white Gaussian noise (AWGN).

The complex-valued data are transmitted at a bit rate of 20 Mb/s over a 1 G Hz radio channel. The modulation scheme is four-level QAM with a symbol rate of 10 M symbols/s and a symbol interval of 100 ns. The packet length is set to 400 symbols (800 bits). A 10% overhead would allow maximum of 40 symbols for training, i.e., $n_u = 40$. The neural-based decision feedback equalizer includes a four-layer feedforward neural network. For simplicity, the neural equalizer is denoted by a short hand notation NNDFE $((n_f, n_b), n_1, n_2, n_0)$ where n_f is the number of forward taps, n_b is the number of feedback taps, n_1 is the number of neurons in hidden layer 1, n_2 is the number of neurons in hidden layer 2, n_0 is the number of neurons in output layer. It should be mentioned that NNDFE1 and NNDFE2 represent the neural networks trained by the back propagation and ELEANNE7 algorithms, respectively. Similarly, traditional LMS decision feedback equalizer is denoted by LMSDFE (n_f, n_b). [2] showed that n_f is set to 3 which is found to be nearly optimal. In addition, the minimum number of feedback taps necessary to eliminate the ISI for data rates of interest is 3 [2]. As discussed in Section II-A, n_i and n_0 can be found as 12 and 2 respectively for 4-QAM system. According to Huang and Huang's [8] suggestions, it is possible to estimate the lower bounds on the numbers of neurons in both hidden layers. Thus, n_1 and n_2 can be chosen as 20 and 10, respectively. Fig. 2 illustrates MSE (mean square error) convergence of the packet-wise neural equalizers, NNDFE1((3,3),20,10,2), NNDFE2((3,3),20,10,2), and LMSDEF(3,3) with the same learning rate (η) 0.03 for training mode and 0.005 for decision-directed mode. The NNDFE1 requires at least 250 packets to converge while the LMSDFE converges in about two packets. The results also show that the steady-state value of averaged square error produced by the NNDFE1 converges to a value (≤ -100 dB) which is greatly lower than the additive noise (-10 dB). This is a result of the approximation capability of packet back propagation. Theorem 1 indicates that the approximation error approaches zero when the number of packets approaches infinity. Similarly, the NNDFE2 converges to a value (≤ -160 dB) which is also considerably lower than the noise level in about 150 packets. The LMSDFE gives a steady value of averaged squared error at about -20 dB which is around the noise floor. Fig. 3 illustrates MSE convergence of the symbol-wise (pattern-wise) training of NNDFE1, NNDFE2, and LMSDFE within the first packet. A comparison of the learning curves in Fig. 3 indicates that the NNDFE2 (ELEANNE7) converges to the target error value (≤ -20 dB) much faster than NNDFE1 (back propagation).

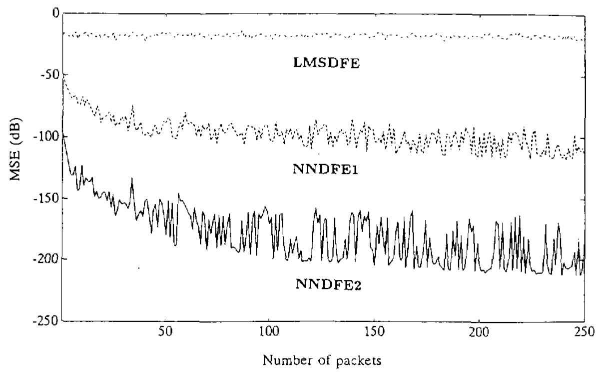

Fig. 2. Comparison of mean square errors achieved by LMSDFE, NNDFE1, and NNDFE2 on a packet-by-packet basis when $n_u = 40$ and SNR = 10 dB.

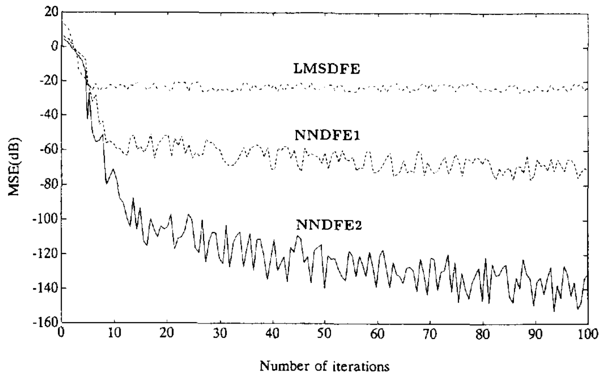

Fig. 3. Comparison of mean square errors achieved by LSMDFE, NNDFE1, and NNDFE2 on a symbol-by-symbol basis when $n_u = 40$ and SNR = 10 dB.

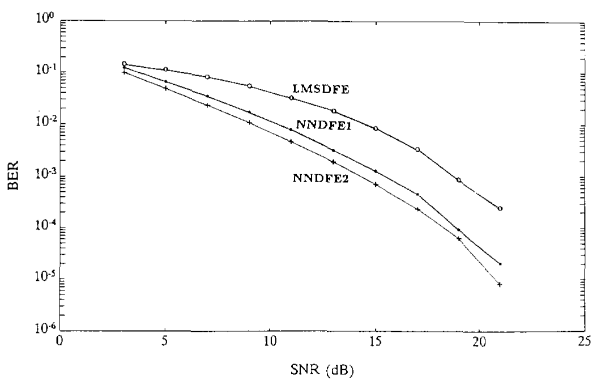

Fig. 4. Comparison of bit error rated achieved by LMSDFE, NNDFE1, and NNDFE2 when $n_u = 40$.

Among the algorithms proposed for training decision feedback equalizer, LMS algorithm is computationally less demanding. However, it can not achieve the better MSE with a value less than −20 dB. Fig. 4 compares the respective bit error rates (BER's) achieved by NNDFE1, NNDFE2, and LMSDFE. It may be observed from Fig. 4 that both the NNDFE1 and NNDFE2 attain about the same 3 dB improvement at BER = 10^{-4} relative to the LMSDFE having the same number of input samples.

VI. Conclusion

This paper has introduced an adaptive decision feedback equalizer based on a four-layer perceptron structure which is capable of dealing with the M-ary QAM signals over the complex fading multipath radio channel by using cost-effective real-valued training algorithms, where each symbol of the M-ary system and received complex signal are represented by $\log_2 M \times 1$ and 2×1 vectors, respectively. The neural-based DFE offers a superior performance as a channel equalizer to that of the conventional LMS DFE because of its ability to approximate arbitrarily nonlinear mapping. For comparison of simulation results, it can be seen that the neural-based DFE provides better BER performance, especially in high noise conditions, also that the MSE of neural-based DFE converges to a value which is greatly lower than that of LMS DFE after receiving a sufficient number of packets. These results would be conducted to verify the performance and approximation capability of packet-wise back-propagation and ELEANNE7 algorithms.

References

[1] A. M. Saleh and R. A. Valenzuala, "A statistical model for indoor multipath propagation," *IEEE J. Select. Areas Commun.*, vol. SAC-5, pp. 128–137, Feb. 1987.

[2] T. A. Sexton and K. Pahlavan, "Channel modeling and adaptive equalization of indoor radio channels," *IEEE J. Select. Areas Commun.*, vol. SAC-7, pp. 114–121, Jan. 1989.

[3] K. Pahlavan, S. J. Howard, and T. A. Sexton, "Decision feedback equalization of the indoor radio channel," *IEEE Tran. Commun.*, vol. 41, pp. 164–170, Jan. 1993.

[4] S. Siu, G. J. Gibson, and C. F. N. Cowan, "Decision feedback equalization using neural network structures and performance comparison with standard architecture," *IEE Proc.*, vol. 137, part. I, no. 4, pp. 221–225, Aug. 1990.

[5] M. E. Cotter, "The Stone–Weierstrass theorem and its application to neural nets," *IEEE Trans. Neural Networks*, vol. 1, pp. 290–295, 1990.

[6] G. Cybenko, "Approximations by superposition of a sigmoidal function," *Math Contr. Syst. Signals*, vol. 2, no. 4, 1989.

[7] D. Chester, "Why two hidden layers are better than one," in *Proc. Int. Joint Conf. Neural Networks*, Washington, DC, June 1989, pp. I-613–I-618.

[8] S. C. Huang and Y. F. Huang, "Bounds on the number of hidden neurons in multilayer perceptrons," *IEEE Trans. Neural Networks*, vol. 2, pp. 47–55, Jan. 1991.

[9] B. Widrow and R. Winter, "Neural nets for adaptive filtering and adaptive pattern recognition," *IEEE Comput. Mag.*, vol. 21, pp. 25–39, Mar. 1988.

[10] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representation by error propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Congnition*, Cambridge, MA: MIT Press, 1986, vol. 1, ch. 8.

[11] S. Z. Qin, H. T. Su and T. J. McAvoy, "Comparison of four neural net learning methods for dynamic system identification," *IEEE Trans. Neural Networks*, vol. 3, pp. 122–130, Jan. 1992.

[12] N. B. Karayiannis and A. N. Venetsanopoulos, "Efficient learning algorithms for neural networks (ELEANNE)," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, pp. 1372–1383, Sep/Oct 1993.

[13] Y. Iiguni, H. Sakai, and H. Tokumaru, "A real-time learning algorithm for a multilayered neural network based on the extended Kalman filter," *IEEE Tran. Signal Process.*, vol. 40, pp. 959–966, Apr. 1992.

[14] C. M. Tseng, "Neural-based packet equalization for indoor radio channel by efficient learning algorithms," MS thesis (supervised by P. R. Chang), Institute of Communication Arts and Science, National Chiao-Tung Univ., Hsinchu, Taiwan, July, 1993.

Po-Rong Chang (M'87) received the B.S. degree in electrical engineering from the National Tsing-Hua University, Taiwan, in 1980, the M.S. degree in telecommunication engineering from National Chiao-Tung University, Hsinchu, Taiwan, in 1982, and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, 1988.

From 1982 to 1984, he was a lecturer in the Chinese Air Force Telecommunication and Electronics School for a two-year military service. From 1984 to 1985 he was an instructor of electrical engineering at National Taiwan Institute of Technology, Taipei, Taiwan. From 1989 to 1990, he was a project leader in charge of SPARC chip design team at ERSO of Industrial Technology and Research Institute, Chu-Tung Taiwan. Currently he is an Associate Professor of Communication Engineering at National Chiao-Tung University, His current interests include fuzzy neural network, wireless multi-media systems, and virtual reality.

Bao-Fuh Yeh was born in Taipei, Taiwan, on July 17, 1965. He received the B.S. degree in electronic engineering from the Chung-Yung University in 1989 and the M.S. degree in communication engineering from the National Chiao-Tung University in 1991. Currently he is working towards the Ph.D. degree in electrical engineering at the National Chiao-Tung University, Hsinchu, Taiwan.

His research interests are in radio propagation model, adaptive equalization, PCS system, and neural networks.

Chih-Chiang Chang was born on December 28, 1967 in Hsinchu, Taiwan. He received the B.S. and the M.S. degrees in electrical engineering from the University of Chen-Kung, Tainan, Taiwan, in 1990 and 1992, respectively. He is presently working towards the Ph.D. degree in communication engineering at the University of Chiao-Tung, Hsinchu, Taiwan.

His current research interests are in multimedia wireless system, CDMA cellular system, color image processing, and fuzzy neural network.