

Adaptive Asymptotic Bayesian Equalization Using a Signal Space Partitioning Technique

Ren-Jr Chen and Wen-Rong Wu, *Member, IEEE*

Abstract—The Bayesian solution is known to be optimal for symbol-by-symbol equalizers; however, its computational complexity is usually very high. The signal space partitioning technique has been proposed to reduce complexity. It was shown that the decision boundary of the equalizer consists of a set of hyperplanes. The disadvantage of existing approaches is that the number of hyperplanes cannot be controlled. In addition, a state-search process, that is not efficient for time-varying channels, is required to find these hyperplanes. In this paper, we propose a new algorithm to remedy these problems. We propose an approximate Bayesian criterion that allows the number of hyperplanes to be arbitrarily set. As a consequence, a tradeoff can be made between performance and computational complexity. In many cases, the resulting performance loss is small, whereas the computational complexity reduction can be large. The proposed equalizer consists of a set of parallel linear discriminant functions and a maximum operation. An adaptive method using stochastic gradient descent has been developed to identify the functions. The proposed algorithm is thus inherently applicable to time-varying channels. The computational complexity of this adaptive algorithm is low and suitable for real-world implementation.

Index Terms—Adaptive, Bayesian, nonlinear equalizer.

I. INTRODUCTION

EQUALIZERS are commonly used in digital communication systems to compensate for the channel effect. These equalizers can be classified as linear or nonlinear. In general, nonlinear equalizers may perform better, but their computational complexity is higher. One exception is the decision feedback equalizer (DFE). The DFE is a nonlinear equalizer, but its computational complexity is low. In many applications, the linear equalizer or the DFE cannot give satisfactory results, and a more sophisticated nonlinear equalizer is required [22], [23]. Two kinds of nonlinear equalizers are well known: sequence estimation and symbol-by-symbol equalizers. The optimal sequence estimation equalizer is known to be the maximum likelihood sequence estimation (MLSE) [1], whereas the optimal symbol-by-symbol equalizer is the Bayesian equalizer [2]. Although the MLSE yields the lowest bit error rate, its processing delay is larger, and its channel tracking ability is lower making it less effective for time-varying channels [24]. In this paper, we concentrate on the symbol-by-symbol equalizer.

Although the Bayesian equalizer is optimal, its computational complexity is significantly higher than that of linear equalizers.

Manuscript received June 25, 2002; revised May 19, 2003. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Naofal Al-Dhahir.

The authors are with the Department of Communication Engineering, National Chiao Tung University, Hsinchu 300, Taiwan, R.O.C. (E-mail: rzchen.cm88g@nctu.edu.tw; wrwu@cc.nctu.edu.tw).

Digital Object Identifier 10.1109/TSP.2004.826162

Thus, many nonlinear algorithms have been employed to approximate or realize the Bayesian solution. In general, the structures of these algorithms allow a tradeoff between performance and computational complexity. These algorithms include the polynomial-based nonlinear equalizer [3], [4] and the artificial neural network [5]–[12]. The disadvantages of these approaches are long training times and lack of methodologies for architecture selection. In addition, their computational complexities may still be too high for many applications. Efficient algorithms for solving the Bayesian equalization problem were then developed. The support vector machine (SVM) approach [13], [14], which is a nonlinear modeling tool, was applied to nonlinear equalization. It was shown that the computational complexity of the SVM can be much lower than polynomial and neural network-based equalizers. However, the learning algorithm in the SVM needs to solve a quadratic programming problem. The optimization method is somewhat computationally intensive.

The other type of efficient algorithm employs the signal space partitioning technique [15]–[18]. This type of approach treats equalization as a classification problem. It has been shown that the Bayesian decision boundary consists of a set of hyperplanes when the signal-to-noise ratio (SNR) is infinite [17]. In [15] and [16], the authors used a combinatorial search and optimization process to find these planes. Despite its high computational complexity, this method is not guaranteed to obtain the asymptotic Bayesian solution. It has been shown [17] that the hyperplanes can be formed by so-called dominant signal state pairs. A simpler method was proposed in [17] to search for these dominant pairs. This design is guaranteed to asymptotically achieve the Bayesian solution. The signal space partitioning techniques mentioned above all require channel information for signal state calculation. The number of hyperplanes and dominant states also depends on channel characteristics. If the channel response changes, the hyperplanes must be recalculated. Thus, these approaches are inefficient for time-varying channels. An alternate method proposed in [25] partitions the signal space into several subspaces using the value of a received signal symbol. A hyperplane is then used for the decision boundary in each subspace. The problem with this approach is that the decision boundary in each subspace may not be well approximated using a single hyperplane.

In this paper, we propose a new method that provides efficient approximation of the Bayesian solution. As in [15]–[18], we use a set of hyperplanes to form the decision boundary; however, the implication of these planes and the method for finding them are quite different. In existing methods, the number of hyperplanes is determined by channel responses. In our approach, the number of hyperplanes is arbitrarily set. The hyperplanes found

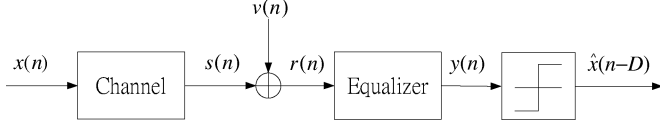


Fig. 1. Typical digital communication system.

by the proposed algorithm are generally different from those found by the method in [15]–[18]. Our method allows an easy tradeoff between complexity and performance. In many cases, we can make the performance loss small, whereas the computational complexity reduction is large. Another feature is that the parameters of these hyperplanes can be adaptively identified using a stochastic gradient descent method. As a result, the proposed equalizer can be effectively applied to time-varying channels. Signal detection is performed using a set of parallel linear discriminant functions followed by a maximum operation. The computational complexity of the proposed equalizer is low and suitable for real-world implementation.

This paper is organized as follows. Section II describes the symbol-by-symbol and optimum Bayesian equalizers. Section III discusses the concept and algorithm of the proposed equalizer. Section IV shows simulation results that demonstrate the effectiveness of the proposed equalizer. Section V gives some concluding remarks.

II. PROBLEM STATEMENT

A typical digital communications system is shown in Fig. 1, where $x(n)$ denotes the transmitted symbol, $s(n)$ the channel output, $v(n)$ the channel noise, $r(n)$ the received signal, $y(n)$ the equalizer output, and $\hat{x}(n-D)$ the equalizer output decision where D is the desired output delay. Let L_c be the memory length of the channel, $\mathbf{x}(n) = [x(n), x(n-1), \dots, x(n-L_c+1)]^T$ be the channel input vector, and the channel input and output be characterized by a mapping function $g(\cdot)$. We then have the following relationship:

$$\begin{aligned} r(n) &= s(n) + v(n) \\ &= g(\mathbf{x}(n)) + v(n). \end{aligned} \quad (1)$$

To simplify the problem, we assume that $x(n)$ is a binary phase shift keying (BPSK) signal, i.e., $\{x(n) = \pm 1\}$, and $v(n)$ is an additive white Gaussian noise (AWGN) with variance σ_v^2 . Conventionally, equalization is treated as an estimation problem, and the output of the equalizer is viewed as an estimate of $x(n-D)$. Let the memory length of the equalizer be L_e , $\mathbf{r}(n) = [r(n), r(n-1), \dots, r(n-L_e+1)]^T$ be the equalizer input vector, and the relationship between the equalizer input and output be described by a mapping function $f(\cdot)$. A criterion is then optimized to find the unknown function. The minimum mean-square error (MMSE) criterion expressed as $E\{[x(n-D) - f(\mathbf{r}(n))]^2\}$ is a commonly used one. Once the optimal function has been found, say, $f_{\text{opt}}(\cdot)$, the equalizer output may be $y(n) = f_{\text{opt}}(\mathbf{r}(n))$, and the decision $\hat{x}(n-D) = +1$ if $y(n) > 0$ and $\hat{x}(n-D) = -1$ if $y(n) \leq 0$. Finding the optimal function may not be always desirable. A simpler approach is to use a predetermined function with unknown parameters and minimize the same criterion to find the parameters.

There is yet another way to solve the equalization problem. Let $\mathbf{s}(n) = [s(n), s(n-1), \dots, s(n-L_e+1)]^T$ and $\mathbf{v}(n) = [v(n), v(n-1), \dots, v(n-L_e+1)]^T$. From (1), we have

$$\mathbf{r}(n) = \mathbf{s}(n) + \mathbf{v}(n). \quad (2)$$

If noise is absent, the equalizer output is then $y(n) = f(\mathbf{s}(n))$. From Fig. 1, we know that the memory size for the whole system is $L_c + L_e - 1$. Thus, we have

$$\mathbf{s}(n) = \mathbf{h}(\mathbf{x}_c(n)) \quad (3)$$

where

$$\begin{aligned} \mathbf{x}_c(n) &= [x(n), x(n-1), \dots, x(n-D+1), x(n-D) \\ &\quad x(n-D-1), \dots, x(n-L_c-L_e+2)]^T \end{aligned}$$

and $\mathbf{h}(\cdot)$ denotes a vector mapping function. Since $x(n)$ has two possible values $+1$ or -1 , $\mathbf{x}_c(n)$ has N_a , which is $2^{L_c+L_e-1}$ possible combinations. From (3), we see that $\mathbf{s}(n)$ also has N_a possible combinations. We call $\mathbf{s}(n)$ the signal vector and its possible values signal states. In practice, noise is always present. From (2), we see that $\mathbf{r}(n)$ will form N_a clusters in a L_e -dimension Euclidean space. Thus, if we can classify the space into two subspaces corresponding to $x(n-D) = +1$ or -1 and determine to which subspace a given $\mathbf{r}(n)$ belongs, we solve the equalization problem. Thus, channel equalization can be viewed as a classification problem [7]. Denote \mathbf{s}_i as the signal state, $i = 1, 2, \dots, N_a$, \mathcal{R}^{L_e} as the L_e -dimension Euclidean space, and the set of all possible signal states in the space as

$$\mathcal{S} \triangleq \{\mathbf{s}_i \in \mathcal{R}^{L_e}, 1 \leq i \leq N_a\}. \quad (4)$$

We call \mathcal{S} the signal state set. Two subsets of \mathcal{S} are defined as

$$\begin{aligned} \mathcal{S}^\pm &\triangleq \{\mathbf{s}_i \in \mathcal{S} : x(n-D) = \pm 1\} \\ \mathcal{S} &= \mathcal{S}^+ \cup \mathcal{S}^-. \end{aligned} \quad (5)$$

The equalizer is then used to classify the received signal space (in \mathcal{R}^{L_e}) into two regions and assign each region a decision value. This can be formulated as the hypothesis testing problem described as follows:

$$H_0 : \{x(n-D) = -1\} : \mathbf{r}(n) = \mathbf{s}^- + \mathbf{v}(n) \quad (6)$$

$$H_1 : \{x(n-D) = +1\} : \mathbf{r}(n) = \mathbf{s}^+ + \mathbf{v}(n) \quad (7)$$

where \mathbf{s}^\pm is a discrete random vector, and its possible values are in \mathcal{S}^\pm . The optimum decision rule for this hypothesis testing problem is the Bayesian decision rule minimizing the probability of error, i.e.,

$$\hat{x}(n-D) = \begin{cases} +1, & P(x(n-D) = +1|\mathbf{r}(n)) \\ & > P(x(n-D) = -1|\mathbf{r}(n)) \\ -1, & P(x(n-D) = +1|\mathbf{r}(n)) \\ & \leq P(x(n-D) = -1|\mathbf{r}(n)). \end{cases} \quad (8)$$

Under the AWGN assumption, it was shown in [7] that the Bayesian decision rule is

$$\hat{x}(n-D) = \begin{cases} +1, & f_B(\mathbf{r}(n)) > 0 \\ -1, & f_B(\mathbf{r}(n)) \leq 0 \end{cases} \quad (9)$$

where

$$f_B(\mathbf{r}(n)) = \sum_{\mathbf{s}_k \in \mathcal{S}^+} \exp\left(-\frac{\|\mathbf{r}(n) - \mathbf{s}_k\|^2}{2\sigma_v^2}\right) - \sum_{\mathbf{s}_k \in \mathcal{S}^-} \exp\left(-\frac{\|\mathbf{r}(n) - \mathbf{s}_k\|^2}{2\sigma_v^2}\right). \quad (10)$$

III. PROPOSED EQUALIZER

A. Approximate Bayesian Criterion

Although the Bayesian solution is optimal (in the minimum error probability sense), the computational complexity is usually very high because in (10), there are N_a exponential terms to be evaluated, and N_a grows exponentially with $L_c + L_e$. In this section, we propose a new method for solving this problem. Our idea is to reduce the number of terms involved in (10). First, we approximate (9) and (10) using the following decision rule:

$$\hat{x}(n-D) = \begin{cases} +1, & \mathbf{s}_{\max} \in \mathcal{S}^+ \\ -1, & \mathbf{s}_{\max} \in \mathcal{S}^- \end{cases} \quad (11)$$

$$\begin{aligned} \mathbf{s}_{\max} &= \arg \max_{\mathbf{s}_k \in \mathcal{S}^+ \cup \mathcal{S}^-} \left\{ \exp\left(-\frac{\|\mathbf{r}(n) - \mathbf{s}_k\|^2}{2\sigma_v^2}\right) \right\} \\ &= \arg \max_{\mathbf{s}_k \in \mathcal{S}^+ \cup \mathcal{S}^-} \left(\mathbf{s}_k^T \mathbf{r}(n) - \frac{\|\mathbf{s}_k\|^2}{2} \right). \end{aligned} \quad (12)$$

Because the exponential operation is a monotonic function, the second equality in (12) holds. Due to the rapid-decay property of the Gaussian function, the decision using (11) and (12) usually provides a good approximation to that using (9) and (10). It is straightforward to see that the smaller the noise level, the smaller the approximation error. The decision using (11) and (12) is asymptotically identical to that using (9) and (10). It is simple to show that the decision boundary of (11) always consists of hyperplanes regardless of noise variance. The advantage of using (11) and (12) is that we do not have to evaluate exponential functions. However, we still have N_a inner product terms to evaluate. To reduce computation, we divide \mathcal{S}^+ (as well as \mathcal{S}^-) into some subsets and merge states in each subset into a new state. If we let the number of all subsets be N_b , we then have N_b new states.

How to determine these N_b new states optimally becomes the key problem. As we show below, the optimal new states are usually not in the \mathcal{S} signal space. For this reason, we call them pseudo states. Denote the set consisting of the pseudo states as \mathcal{S}_p^\pm . We can then further approximate (12) as

$$\mathbf{s}_{\max} = \arg \max_{\mathbf{s}_k \in \mathcal{S}_p^+ \cup \mathcal{S}_p^-} \left(\mathbf{s}_k^T \mathbf{r}(n) - \frac{\|\mathbf{s}_k\|^2}{2} \right). \quad (13)$$

We now have N_b inner product terms to evaluate instead of N_a . If N_b is significantly smaller than N_a and equalizer performance is not affected, then the goal of complexity reduction has been achieved. As [17] showed, the Bayesian decision boundary consists of a set of hyperplanes if noise is absent. Each hyperplane is determined by a dominant state pair (a state in \mathcal{S}^+ and another state in \mathcal{S}^-). Only those terms associated with the dominant states need be evaluated. The number of dominant pairs, however, is determined by the channel response. Although our

approach also leads to a decision boundary consisting of hyperplanes, there are some fundamental differences to the method in [17]. First, the number of pseudo states can be set arbitrarily in our approach and is not dependent on channels. We can then easily trade performance for complexity. Second, the pseudo states are found through minimization of some criterion and not by searching in \mathcal{S} space. Thus, the stochastic gradient method can be applied, and this results in an adaptive equalization algorithm. Note that hyperplanes found by the method in (13) are generally not identical to those by the method in [17]. The key problem is, as mentioned above, how to find those pseudo states. As we will show, our method is simple and effective.

B. Equalization Using a Single Hyperplane

We start with a simple case in which the number of pseudo states is two (i.e., one for \mathcal{S}^+ and the other for \mathcal{S}^-). From (13), we can see that the decision boundary is just a hyperplane. We propose using the following MMSE criterion and nonlinear function to estimate the pseudo states:

$$\min J(n) = E \left\{ [x(n-D) - y(n)]^2 \right\} \quad (14)$$

where

$$y(n) = \frac{\exp\left(-\frac{\|\mathbf{r}(n) - \mathbf{m}_1\|^2}{2\sigma_v^2}\right) - \exp\left(-\frac{\|\mathbf{r}(n) - \mathbf{m}_2\|^2}{2\sigma_v^2}\right)}{\exp\left(-\frac{\|\mathbf{r}(n) - \mathbf{m}_1\|^2}{2\sigma_v^2}\right) + \exp\left(-\frac{\|\mathbf{r}(n) - \mathbf{m}_2\|^2}{2\sigma_v^2}\right)} \quad (15)$$

where \mathbf{m}_1 and \mathbf{m}_2 denote the pseudo states. Derivation of the nonlinear function in (15) is shown in the Appendix. Equation (15) corresponds to the MMSE estimate of $x(n-D)$ when the number of signal states are two and their values are known. Thus, we can say that if the number of signal states is actually two, the solution of (14) will give the true signal states. In general cases, the number of states in \mathcal{S}^\pm is greater than two. How do we interpret the equalization results using the pseudo state found in (14) and (15) and the decision rule in (13)? To answer this question, we first rewrite (14) as follows. If $x(n-D)$ is +1, the resultant signal state is in \mathcal{S}^+ , and the corresponding pseudo state is \mathbf{m}_1 . On the contrary, if $x(n-D)$ is -1, the resultant signal state is in \mathcal{S}^- , and the corresponding pseudo state is \mathbf{m}_2 . Then, (14) can be rewritten as

$$\min J(n) = E \left\{ [1 - y_{i,k}(n)]^2 \right\} \quad (16)$$

where

$$y_{i,k}(n) = \frac{\exp\left(-\frac{\|\mathbf{r}(n) - \mathbf{m}_i\|^2}{2\sigma_v^2}\right) - \exp\left(-\frac{\|\mathbf{r}(n) - \mathbf{m}_k\|^2}{2\sigma_v^2}\right)}{\exp\left(-\frac{\|\mathbf{r}(n) - \mathbf{m}_i\|^2}{2\sigma_v^2}\right) + \exp\left(-\frac{\|\mathbf{r}(n) - \mathbf{m}_k\|^2}{2\sigma_v^2}\right)} \quad (17)$$

$i = 1, k = 2$ when $x(n-D)$ is +1, and $i = 2, k = 1$ when $x(n-D)$ is -1. Note that $y_{1,2}(n)$ is the same as $y(n)$ in (15). We introduce this notation for later development convenience. From (16) and (17), we can observe that the form of the cost function is similar for both transmitted symbols. Only the output definition is different. If the signal state for a transmitted symbol is in \mathcal{S}^+ and the received signal $\mathbf{r}(n)$ is closer to \mathbf{m}_1 than \mathbf{m}_2 , using the decision rule (13), we find that the decision is +1 and that it is correct. However, if $\mathbf{r}(n)$ is closer to \mathbf{m}_2 than \mathbf{m}_1 , the decision

is -1 , and it is wrong. Note that if σ_v^2 is small, the nonlinear function in (17) approaches a step function. When the decision is correct, the cost function $J(n)$ tends to be 0. When the decision is wrong, the cost function $J(n)$ tends to be a constant 4. The result is similar for the case in which the transmitted symbol signal state is in \mathcal{S}^- . This property has a significant implication, as described below.

A standard technique for classification employs a set of discriminant functions. A discriminant function is designed to have a maximum value for a certain class of objects. As [26] reveals, a classifier using a discriminant function will yield a minimum classification error probability if the parameters of the function is obtained by minimizing a cost function that gives 0 value when the decision is right and a constant when the decision is wrong. As described above, we treat equalization as a classification problem. Two functions in (13) actually correspond to two discriminant functions. Thus, we conclude that equalization results using the decision rule in (13), and the associated pseudo state estimates found in (14) and (15) will achieve a minimum error probability.

Since (14) is highly nonlinear, it is difficult to obtain the solution directly. Here, we employ the adaptive method to solve the problem. There are at least two advantages to this approach. First, the optimal solution can be found using a process called training, and the computational complexity of the training algorithm is usually very low. Second, the equalizer can be continuously trained using former decisions such that it can operate in a time-varying environment. The specific method we use is called the steepest descent method [27]. Let $i, k \in \{1, 2\}$ and $i \neq k$. For equations shown below, if the signal state for a training symbol is in \mathcal{S}^+ , then $i = 1$. Otherwise, $i = 2$. Using the chain rule, we can have the gradient vectors from (16). Then, the update equations are given as

$$\mathbf{m}_i(n+1) = \mathbf{m}_i(n) + \mu(1 - y_{i,k}(n))(1 - y_{i,k}^2(n)) \times (\mathbf{r}(n) - \mathbf{m}_i(n)) \quad (18)$$

$$\mathbf{m}_k(n+1) = \mathbf{m}_k(n) - \mu(1 - y_{i,k}(n))(1 - y_{i,k}^2(n)) \times (\mathbf{r}(n) - \mathbf{m}_k(n)) \quad (19)$$

where μ is the step size. As we can see, the computational complexity requirement for the adaptive algorithm is quite low. Once the pseudo states are obtained, we can have \mathbf{s}_{\max} as

$$\mathbf{s}_{\max} = \arg \max_{i \in \{1, 2\}} \left(\mathbf{m}_i^T \mathbf{r}(n) - \frac{\|\mathbf{m}_i\|^2}{2} \right). \quad (20)$$

The decision is then

$$\hat{x}(n-D) = \begin{cases} +1, & \mathbf{s}_{\max} = \mathbf{m}_1 \\ -1, & \mathbf{s}_{\max} = \mathbf{m}_2. \end{cases} \quad (21)$$

From the stand point of classification, (20) and (21) divides received signal space \mathcal{R}^{L_e} into two regions \mathcal{R}_d^+ and \mathcal{R}_d^- and assigns a decision value to each region. We call \mathcal{R}_d^\pm the decision region for $x(n-D) = \pm 1$. From (20), we can have the decision region as

$$\mathcal{R}_d^+ \triangleq \{\mathbf{r}(n) \in \mathcal{R}^{L_e} : f^1(\mathbf{r}(n)) > f^2(\mathbf{r}(n))\} \quad (22)$$

$$\mathcal{R}_d^- \triangleq \{\mathbf{r}(n) \in \mathcal{R}^{L_e} : f^1(\mathbf{r}(n)) \leq f^2(\mathbf{r}(n))\} \quad (23)$$

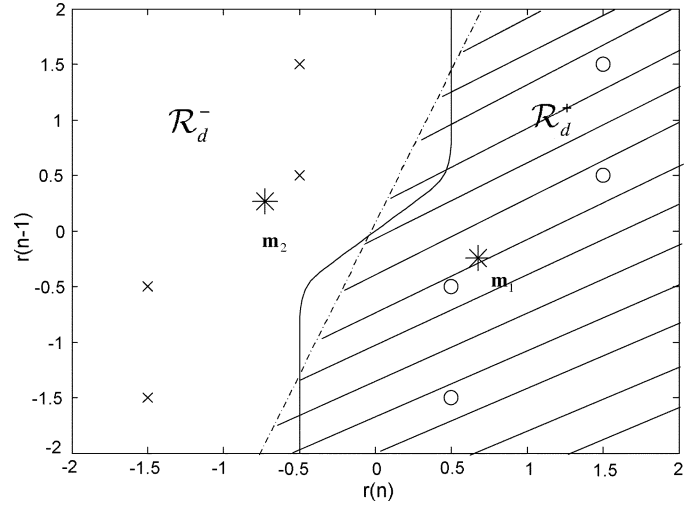


Fig. 2. Decision boundaries for the proposed (dashed line) and the Bayesian (solid line) equalizer with two subsets.

where

$$f^1(\mathbf{r}(n)) = \mathbf{m}_1^T \mathbf{r}(n) - \frac{\|\mathbf{m}_1\|^2}{2} \quad (24)$$

$$f^2(\mathbf{r}(n)) = \mathbf{m}_2^T \mathbf{r}(n) - \frac{\|\mathbf{m}_2\|^2}{2}. \quad (25)$$

We now use an example to describe the algorithm proposed above.

Example 1: Consider a linear channel with memory length $L_c = 2$ and impulse response $\{1, 0.5\}$. Since the channel is linear, the output is obtained using a convolution of $x(n)$ and the channel response. Let $L_e = 2$ and $D = 0$. Thus, $\mathbf{x}_c(n) = [x(n), x(n-1), x(n-2)]^T$, and eight signal states result. The signal state set is then divided into two subsets corresponding to $x(n) = 1$ and $x(n) = -1$.

$$\begin{aligned} \mathcal{S}^+ &= \{\mathbf{s}_1 = [1.5, 1.5]^T, \mathbf{s}_2 = [1.5, 0.5]^T \\ &\quad \mathbf{s}_3 = [0.5, -0.5]^T, \mathbf{s}_4 = [0.5, -1.5]^T\} \\ \mathcal{S}^- &= \{\mathbf{s}_5 = [-0.5, 1.5]^T, \mathbf{s}_6 = [-0.5, 0.5]^T \\ &\quad \mathbf{s}_7 = [-1.5, -0.5]^T, \mathbf{s}_8 = [-1.5, -1.5]^T\}. \end{aligned} \quad (26)$$

The noise variance σ_v^2 is set to 0.05. The proposed algorithm is applied to carry out equalization, and the result is shown in Fig. 2 in which the symbol "o" denotes the signal state corresponding to $x(n) = +1$, and "x" denotes the signal state corresponding to $x(n) = -1$. Since $L_e = 2$, the received signal space is a plane, and the decision boundary is a one-dimensional curve. As we can see from the figure, the decision boundary of the proposed algorithm is linear, which was expected. It is also interesting to note that the determined pseudo state in \mathcal{S}^+ (or \mathcal{S}^-) is quite close to one of the states in \mathcal{S}^+ (or \mathcal{S}^-). From the figure, it is apparent that there is much room for performance enhancement. If we can break the decision boundary into smaller pieces and approximate each piece using a linear boundary, we can better approximate the optimal decision boundary. To implement this idea, we must then subdivide \mathcal{S}^+ (or \mathcal{S}^-) into smaller subsets. This is elaborated upon in the next subsection.

C. Equalization Using Multiple Hyperplanes

In the previous subsection, we developed an equalization algorithm using a single hyperplane. The signal state set \mathcal{S} was divided into two subsets \mathcal{S}^\pm corresponding to $x(n-D) = \pm 1$, and two pseudo states were used. In this subsection, we extend this method to accommodate the general equalization problem. The idea is to subdivide each signal state set \mathcal{S}^\pm into more subsets. Each subset is represented by a pseudo state. A hyperplane decision boundary is then determined using a pair of pseudo states. Our subdividing approach is simple and straightforward. For example, the signal state set \mathcal{S} can be divided into four subsets corresponding to $x(n-D) = \pm 1$ and $x(n-D-1) = \pm 1$ or one corresponding to $x(n-D+1) = \pm 1$ and $x(n-D) = \pm 1$. To have a general formulation, we first rewrite the input vector $\mathbf{x}_c(n)$ as a combination of three vectors.

$$\mathbf{x}_c(n) = [\mathbf{x}_{c,1}^T(n), \mathbf{x}_{c,2}^T(n), \mathbf{x}_{c,3}^T(n)]^T \quad (27)$$

where

$$\begin{aligned} \mathbf{x}_{c,1}(n) &= [x(n), x(n-1), \dots, x(n-D+P+1)]^T \\ \mathbf{x}_{c,2}(n) &= [x(n-D+P), \dots, x(n-D+1), x(n-D) \\ &\quad x(n-D-1), \dots, x(n-D-Q)]^T \\ \mathbf{x}_{c,3}(n) &= [x(n-D-Q-1), x(n-Q-2), \dots \\ &\quad x(n-L_c-L_e+2)]^T \end{aligned} \quad (28)$$

where $D \geq P \geq 0$, and $L_c + L_e - D - 2 \geq Q \geq 0$. Note that $\mathbf{x}_{c,2}(n)$ includes $x(n-D)$, and its length is $P+Q+1$. Depending on the value of D , $\mathbf{x}_{c,1}(n)$ or $\mathbf{x}_{c,3}(n)$ may or may not exist. For example, if $D=0$, we do not have $\mathbf{x}_{c,1}(n)$. Let $M = 2^{P+Q+1}$. We then have M possible vector values for $\mathbf{x}_{c,2}(n)$. Denote these vectors as \mathbf{x}_i , $i = 1, 2, \dots, M$. Now, we can divide the signal state set according to the value of $\mathbf{x}_{c,2}(n)$:

$$\mathcal{S}^i \triangleq \{\mathbf{s}_i \in \mathcal{S} : \mathbf{x}_{c,2}(n) = \mathbf{x}_i\}, \quad 1 \leq i \leq M \quad (29)$$

and

$$\mathcal{S} = \bigcup_{1 \leq i \leq M} \mathcal{S}^i. \quad (30)$$

For example, if we let $D=0$, $P=0$, and $Q=1$, then $\mathbf{x}_{c,2}(n) = [x(n), x(n-1)]^T$, and the corresponding subsets are

$$\mathcal{S}^1 \triangleq \{\mathbf{s}_i \in \mathcal{S} : \mathbf{x}_{c,2}(n) = [+1, +1]^T\} \quad (31)$$

$$\mathcal{S}^2 \triangleq \{\mathbf{s}_i \in \mathcal{S} : \mathbf{x}_{c,2}(n) = [+1, -1]^T\} \quad (32)$$

$$\mathcal{S}^3 \triangleq \{\mathbf{s}_i \in \mathcal{S} : \mathbf{x}_{c,2}(n) = [-1, +1]^T\} \quad (33)$$

$$\mathcal{S}^4 \triangleq \{\mathbf{s}_i \in \mathcal{S} : \mathbf{x}_{c,2}(n) = [-1, -1]^T\}. \quad (34)$$

If $P=Q=0$, we then have two signal subsets, and this degenerates to the case discussed above. We then define M pseudo states $\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_M$ to represent the corresponding M subsets. Thus, we have M decision regions associated with these subsets. For the time being, we assume that these pseudo states are known. The decision rule using (13) suggests that we can assign a received signal vector $\mathbf{r}(n)$ to subset i if the distance to \mathbf{m}_i is minimal. Once the signal subset has been determined, the

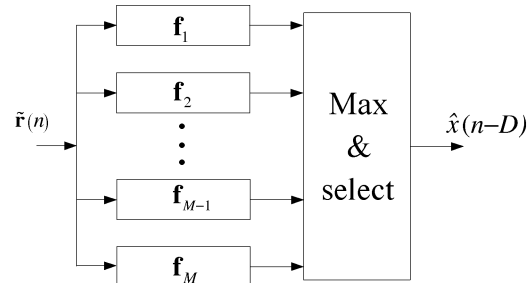


Fig. 3. Structure of the proposed nonlinear equalizer.

$x(n-D)$ value corresponding to the subset gives the decision. Define \mathcal{I}^\pm as a set with indexes such that

$$\mathcal{I}^\pm \triangleq \{i \in \mathcal{N} : \mathbf{J}_D^T \mathbf{x}_i = \pm 1\} \quad (35)$$

where \mathbf{J}_D is a $(P+Q+1) \times 1$ vector, and

$$\mathbf{J}_D = [0 \quad \underbrace{\dots \quad 0}_{P+1} \quad 1 \quad 0 \quad \dots \quad 0]^T. \quad (36)$$

In addition, define

$$f^i(\mathbf{r}(n)) = \mathbf{m}_i^T \mathbf{r}(n) - \frac{\|\mathbf{m}_i\|^2}{2}. \quad (37)$$

Thus, the output decision can be obtained as

$$i = \arg \max_l f^l(\mathbf{r}(n)) \quad (38)$$

$$\hat{x}(n-D) = \begin{cases} +1, & i \in \mathcal{I}^+ \\ -1, & i \in \mathcal{I}^- \end{cases} \quad (39)$$

The overall structure of the proposed nonlinear equalizer is shown in Fig. 3. The response of each linear discriminant function in Fig. 3 corresponds to $\mathbf{f}_i = [\mathbf{m}_i^T (\|\mathbf{m}_i\|^2/2)]^T$, and the input of each function is $\tilde{\mathbf{r}}(n) = [\mathbf{r}(n)^T \quad 1]^T$. If $\mathbf{r}(n)$ is in the decision region of subset i , then

$$f^i(\mathbf{r}(n)) > f^j(\mathbf{r}(n)), \quad j = 1, 2, \dots, M \quad \text{and} \quad i \neq j. \quad (40)$$

Note that each inequality in (40) forms a hyperplane between Subset i and j . If we pretend that only these two subsets exist, this gives a decision region for subset i . For the same subset i , there are $M-1$ such regions ($j = 1, 2, \dots, M, j \neq i$), and the true decision region for subset i is the intersection of these regions. Let $\mathcal{R}_d^{i,j}$ be the decision region formed by subset i and j (with respect to subset i), and \mathcal{R}_d^i is the decision region for subset i . Then

$$\mathcal{R}_d^{i,j} \triangleq \{\mathbf{r}(n) \in \mathcal{R}^{L_e} : f^i(\mathbf{r}(n)) > f^j(\mathbf{r}(n))\} \quad (41)$$

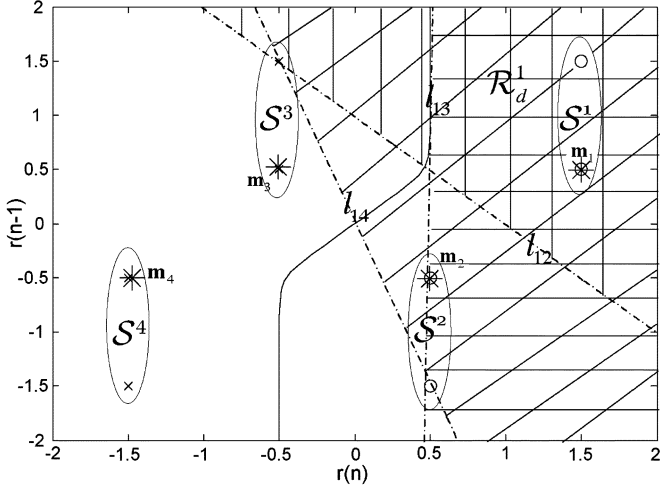
and

$$\mathcal{R}_d^i \triangleq \bigcap_{j \neq i} \mathcal{R}_d^{i,j}. \quad (42)$$

Then, the final decision region \mathcal{R}_d^\pm corresponding to $x(n-D) = \pm 1$ is

$$\mathcal{R}_d^\pm \triangleq \bigcup_{i \in \mathcal{I}^\pm} \mathcal{R}_d^i. \quad (43)$$

To have a better understanding of this idea, we give an example here.

Fig. 4. Decision region of subset 1, \mathcal{R}_d^1 .

Example 2: Consider the same scenario given in Example 1. We divide signal state set \mathcal{S} into four subsets ($M = 4$) using the method described in (31)–(34). This means that $\mathbf{x}_c(n) = [\mathbf{x}_{c,2}(n), \mathbf{x}_{c,3}(n)]^T$, where $\mathbf{x}_{c,2}(n) = [x(n), x(n-1)]^T$ and $\mathbf{x}_{c,3}(n) = x(n-2)$. Note that $\mathbf{x}_{c,1}(n)$ is absent in this case. We then have the corresponding subsets

$$\begin{aligned} \mathcal{S}^1 &\triangleq \{\mathbf{s}_1 = [1.5, 1.5]^T \\ &\quad \mathbf{s}_2 = [1.5, 0.5]^T : \mathbf{x}_{c,2}(n) = [+1, +1]^T\} \end{aligned} \quad (44)$$

$$\begin{aligned} \mathcal{S}^2 &\triangleq \{\mathbf{s}_3 = [0.5, -0.5]^T \\ &\quad \mathbf{s}_4 = [0.5, -1.5]^T : \mathbf{x}_{c,2}(n) = [+1, -1]^T\} \end{aligned} \quad (45)$$

$$\begin{aligned} \mathcal{S}^3 &\triangleq \{\mathbf{s}_5 = [-0.5, 1.5]^T \\ &\quad \mathbf{s}_6 = [-0.5, 0.5]^T : \mathbf{x}_{c,2}(n) = [-1, +1]^T\} \end{aligned} \quad (46)$$

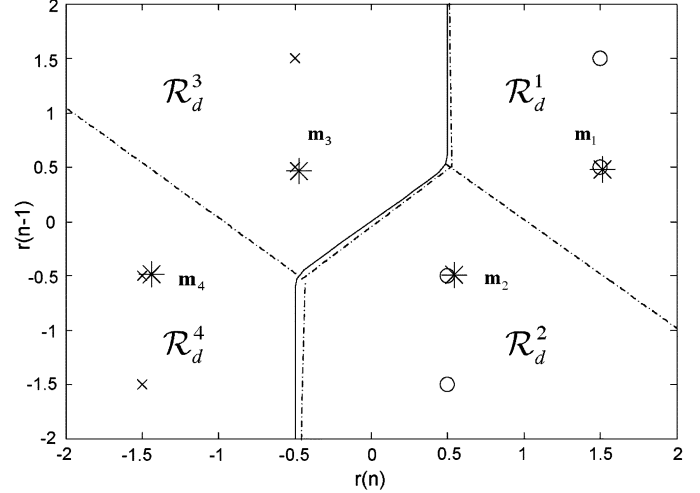
$$\begin{aligned} \mathcal{S}^4 &\triangleq \{\mathbf{s}_7 = [-1.5, -0.5]^T \\ &\quad \mathbf{s}_8 = [-1.5, -1.5]^T : \mathbf{x}_{c,2}(n) = [-1, -1]^T\}. \end{aligned} \quad (47)$$

The proposed algorithm is then applied to perform equalization. Fig. 4 shows how a decision region is formed in detail. Here, we use \mathcal{R}_d^1 for detailed description. Since $M = 4$, the region is formed by three dashed lines l_{12} , l_{13} , and l_{14} , where l_{1j} is the decision boundary between \mathcal{R}_d^1 and \mathcal{R}_d^j . It is obtained by solving $f^1(\mathbf{r}(n)) = f^j(\mathbf{r}(n))$, $j = 2, 3$, and 4 . Symbol * in the figure represents pseudo states. The three shadow regions on the right side of l_{1j} show the region of $f^1(\mathbf{r}(n)) > f^j(\mathbf{r}(n))$. We can see that shadow region \mathcal{R}_d^1 is the intersection of all regions for $f^1(\mathbf{r}(n)) > f^j(\mathbf{r}(n))$, $j = 2, 3$, and 4 . Fig. 5 shows all of the decision regions \mathcal{R}_d^i , $j = 1, 2, 3$, and 4 . By definition in (35), $\mathcal{I}^+ = \{1, 2\}$, and $\mathcal{I}^- = \{3, 4\}$. The decision \mathcal{R}_d^\pm corresponding to $x(n) = \pm 1$ is then

$$\mathcal{R}_d^+ = \mathcal{R}_d^1 \cup \mathcal{R}_d^2 \quad (48)$$

$$\mathcal{R}_d^- = \mathcal{R}_d^3 \cup \mathcal{R}_d^4. \quad (49)$$

The final decision boundary can also be seen in Fig. 5. It shows that the decision boundary of the proposed equalizer is very close to that of the Bayesian equalizer. However, as we show below, the computational complexity of the proposed algorithm is significantly lower.

Fig. 5. Decision regions for all subsets \mathcal{R}_d^i , $i = 1, 2, 3, 4$ and decision boundaries of the proposed (dashed line) and the Bayesian (solid line) equalizer.

We next describe how to find the pseudo states $\mathbf{m}_1 \mathbf{m}_2 \cdots \mathbf{m}_M$. We extend the method in (16). If the signal state for a transmitted symbol is in \mathcal{S}^i , then the cost function to be minimized is defined as

$$\min J(n) = \sum_{k \neq i} E \left\{ [1 - y_{i,k}(n)]^2 \right\} \quad (50)$$

where

$$y_{i,k}(n) = \frac{\exp\left(-\frac{\|\mathbf{r}(n) - \mathbf{m}_i\|^2}{2\sigma_v^2}\right) - \exp\left(-\frac{\|\mathbf{r}(n) - \mathbf{m}_k\|^2}{2\sigma_v^2}\right)}{\exp\left(-\frac{\|\mathbf{r}(n) - \mathbf{m}_i\|^2}{2\sigma_v^2}\right) + \exp\left(-\frac{\|\mathbf{r}(n) - \mathbf{m}_k\|^2}{2\sigma_v^2}\right)}. \quad (51)$$

Note that $J(n)$ tends to be 0 if the decision is right and a value less than $4(M-1)$ if the decision is wrong. Thus, as mentioned above, the decision rule associated with the estimates in (51) will yield a minimum error probability among all detectors using the same number of hyperplanes. To find the pseudo states, we still apply the adaptive method. For each received signal vector $\mathbf{r}(n)$, we can obtain its corresponding signal state $\mathbf{x}_c(n)$. According to $\mathbf{x}_{c,2}(n)$, we know to which subset the signal state belongs (in training mode). The stochastic gradient descent method can still be used here. We then perform pseudo state adaptation similar to (18) and (19). If a received signal vector belongs to subset i , we then have the following adaptive algorithm ($i \neq k$):

$$\begin{aligned} \mathbf{m}_i(n+1) &= \mathbf{m}_i(n) + \mu \sum_{k \neq i} (1 - y_{i,k}(n)) \\ &\quad \times (1 - y_{i,k}^2(n)) (\mathbf{r}(n) - \mathbf{m}_i(n)) \end{aligned} \quad (52)$$

$$\begin{aligned} \mathbf{m}_k(n+1) &= \mathbf{m}_k(n) - \mu (1 - y_{i,k}(n)) \\ &\quad \times (1 - y_{i,k}^2(n)) (\mathbf{r}(n) - \mathbf{m}_k(n)) \end{aligned} \quad (53)$$

$$y_{i,k}(n) = \frac{\exp\left(-\frac{\|\mathbf{r}(n) - \mathbf{m}_i(n)\|^2}{2\sigma_v^2}\right) - \exp\left(-\frac{\|\mathbf{r}(n) - \mathbf{m}_k(n)\|^2}{2\sigma_v^2}\right)}{\exp\left(-\frac{\|\mathbf{r}(n) - \mathbf{m}_i(n)\|^2}{2\sigma_v^2}\right) + \exp\left(-\frac{\|\mathbf{r}(n) - \mathbf{m}_k(n)\|^2}{2\sigma_v^2}\right)}. \quad (54)$$

Note that the physical interpretation of the value 1 in (52) and (53) is different from that in (18) and (19). In (18) and (19), the value of 1 corresponds to the desired signal, which is $x(n-D)$, whereas in (52) and (53), it is just a desired constant for a right decision.

Since the cost function is a nonlinear function, the adaptive algorithm may converge to a local minimum. Thus, the proper choice of initial value is important. We suggest using the clustering method [11] to obtain a reliable initial value. The computational complexity requirement for this algorithm is very low. The clustering method is given as

$$\begin{aligned}
 &\text{if } \mathbf{x}_{c,2}(n) == \mathbf{x}_i \\
 &\quad \mathbf{m}_i(n+1) = \text{counter}_i \times \mathbf{m}_i(n) + \mathbf{r}(n) \\
 &\quad \text{counter}_i = \text{counter}_i + 1 \\
 &\quad \mathbf{m}_i(n+1) = \frac{\mathbf{m}_i(n+1)}{\text{counter}_i} \\
 &\text{end.} \tag{55}
 \end{aligned}$$

We have thus assumed that the noise variance is known *a priori*. As a matter of fact, this assumption is not required. The smaller the noise variance, the more closely the function in (54) approaches a step function, and the closer the equalizer is to achieve the optimal performance (minimum classification rate). However, the adaptive algorithm also converges more slowly, which affects the final pseudo states positions. Thus, we can treat σ_v^2 as a design parameter. As we show in the next section, the equalization results are not sensitive to the choice of this parameter. Denote the parameter as α . According to the adaptive algorithm given above, we may summarize the overall adaptive algorithm to adjust $\mathbf{m}_1 \mathbf{m}_2 \cdots \mathbf{m}_M$ as follows:

- 1) Initially set $\mathbf{m}_1 \mathbf{m}_2 \cdots \mathbf{m}_M$ by the clustering method
- 2) For each instant of time, $n = 1, 2, \dots$,
 - if $\mathbf{x}_{c,2}(n) == \mathbf{x}_i$
 - for $k = 1 : M$ and $k \neq i$
$$y_{i,k}(n) = \frac{\exp\left(-\frac{\|\mathbf{r}(n) - \mathbf{m}_i(n)\|^2}{2\alpha}\right) - \exp\left(-\frac{\|\mathbf{r}(n) - \mathbf{m}_k(n)\|^2}{2\alpha}\right)}{\exp\left(-\frac{\|\mathbf{r}(n) - \mathbf{m}_i(n)\|^2}{2\alpha}\right) + \exp\left(-\frac{\|\mathbf{r}(n) - \mathbf{m}_k(n)\|^2}{2\alpha}\right)};$$

$$\begin{aligned}
 \mathbf{m}_i(n+1) &= \mathbf{m}_i(n) + \mu(1 - y_{i,k}(n))(1 - y_{i,k}^2(n)) \\
 &\quad \times (\mathbf{r}(n) - \mathbf{m}_i(n)); \\
 \mathbf{m}_k(n+1) &= \mathbf{m}_k(n) - \mu(1 - y_{i,k}(n))(1 - y_{i,k}^2(n)) \\
 &\quad \times (\mathbf{r}(n) - \mathbf{m}_k(n));
 \end{aligned}$$
 - end
 - end

The proposed method described above partitions the signal space into 2^I subsets, where I is an integer. It is straightforward to relax this constraint and partition the signal space into an arbitrary number of subsets. The details, however, are omitted here.

With a minor modification, the clustering algorithm described above can find the mean of each cluster as well as the variance. Assuming that each cluster has a Gaussian distribution and treating its mean as a state, we can then apply the Bayesian decision rule. We call this a reduced-state (RS) Bayesian equalizer.

TABLE I
COMPUTATIONAL COMPLEXITY COMPARISON FOR THE LINEAR AND PROPOSED EQUALIZERS IN THE TRAINING PHASE

	Proposed		Linear
	Initial	Adaptive	
Multiplications	L_e	$(M-1)(3L_e+5)+(L_e+1)$	$2L_e + 1$
Additions	L_e	$(M-1)(4L_e+3) + (2L_e-1)$	$2L_e$
Others		$2(M-1) \times \exp(\cdot)$	

TABLE II
COMPUTATIONAL COMPLEXITY COMPARISON FOR THE LINEAR, PROPOSED, AND BAYESIAN EQUALIZERS IN THE DECISION PHASE

	Bayesian	Proposed	Linear
Multiplications	$N_a(L_e+1)$	ML_e	L_e
Additions	$2L_eN_a-1$	ML_e	L_e-1
Others	$N_a \times \exp(\cdot)$	Compare logs	

Depending on the number of clusters, the computational complexity of the RS Bayesian equalizer can be much lower than that of the original Bayesian equalizer. However, as we show in the next section, the performance of the RS Bayesian equalizer is poor unless the number of the clusters is large.

D. Computational Complexity

There are two phases for the proposed algorithm: the training and decision phases. For simplicity, we treat the computational requirement for division the same as that for multiplication. We summarize the overall computational requirement for both phases in Tables I and II. Roughly speaking, in the decision phase, the complexity of the proposed equalizer is M times higher than the linear equalizer. The actual choice of M is dependent on the desired performance. In our design, the decision region for each subset is automatically and adaptively formed, and these decision regions are approximations of Bayesian decision regions. In principle, if the Bayesian decision boundary has a complex shape, we need more subsets. In many cases, however, only a small number of subsets is sufficient. The computational complexity of the proposed equalizer can be much lower than that of the Bayesian equalizer because M can be much smaller than N_a without significantly sacrificing performance. Note that if a time-varying channel is considered, decisions can be used to train the proposed equalizer continuously such that channel variations can be properly tracked. In this case, the computational complexity for the proposed equalizer is the summation of that listed in Table I (the adaptive part) and that in Table II. Since the value of M used is usually small, the overall complexity will not be increased significantly.

IV. SIMULATION RESULTS

In this section, we report simulation results demonstrating the effectiveness of the proposed equalizer. Two linear channels and one nonlinear channel were used. We compared the proposed equalizer with an optimum Bayesian equalizer, a conventional MMSE linear equalizer, and a minimum bit error rate (MBER) linear equalizer [20]. To reflect the relative noise level, we defined the SNR as the ratio of σ_s^2 and σ_v^2 , where σ_s^2 is the variance

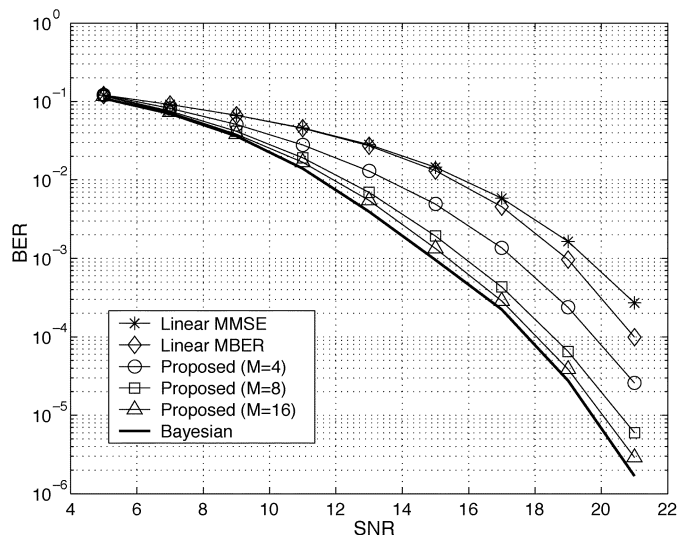


Fig. 6. BER comparison for the MMSE linear, MBER linear, proposed, and Bayesian equalizers.

of $s(n)$, and σ_v^2 is that of $v(n)$. The bit error rate (BER) was used as the performance measure. We first considered a linear channel [11]. The channel is described using a difference equation as

$$r(n) = 0.3482x(n) + 0.8764x(n-1) + 0.3482x(n-2) + v(n). \quad (56)$$

As we can see, the channel length is 3 ($L_c = 3$). For all equalizers compared, we let $L_e = 4$ and $D = 1$. Thus, $N_a = 2^{L_c+L_e-1} = 64$. For the proposed equalizer, we let $M = 4$ and $M = 8$, for which $\mathbf{x}_{c,2}(n) = [x(n), x(n-1)]^T$ and $\mathbf{x}_{c,2}(n) = [x(n), x(n-1), x(n-2)]^T$, respectively. The simulation results for various SNR conditions are shown in Fig. 6. In the figure, we see that the performance of the MMSE linear equalizer was the worst, and the Bayesian one was the best. There is a performance gap between them. The performance of the linear MBER equalizer [20], [21] was only slightly better than that of the linear MMSE equalizer. The performance of the proposed equalizer was close to that of the Bayesian equalizer when $M = 8$. However, the computational complexity was significantly lower. While the performance of the proposed equalizer when $M = 4$ was worse than when $M = 8$, it was much better than that of the MBER linear equalizer. We observed the learning curve for the proposed equalizer ($M = 4$ and SNR = 15 dB) and found that the adaptive algorithm converged around 5000 iterations. Fig. 7 gives a performance comparison for the proposed and the RS Bayesian equalizers. The RS Bayesian equalizer performed poorly when $M = 4$ and $M = 8$. Although the performance of the RS Bayesian equalizer was good when $M = 16$, it was worse than that of the proposed equalizer when $M = 8$. We have also tried various α values for the proposed algorithm ($M = 4$). We varied α from 0.05 to 0.8 and found that the performance was almost unaffected. The proposed algorithm performance is not sensitive to the choice of α .

The second channel we considered was also a linear channel [7]:

$$r(n) = 0.5x(n) + x(n-1) + v(n). \quad (57)$$

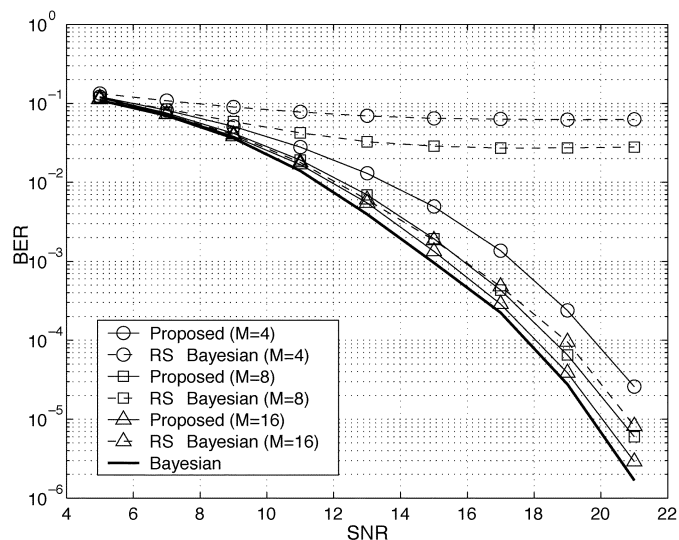


Fig. 7. BER comparison for the proposed and RS Bayesian equalizers.

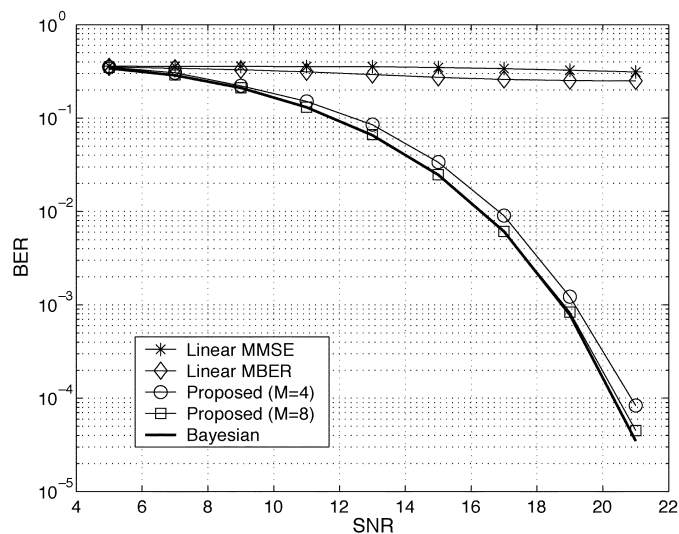


Fig. 8. BER comparison for the MMSE linear, MBER linear, proposed and Bayesian equalizers.

We let $L_c = 2$ and $D = 0$ for all equalizers. Note that for this scenario, the signal state spaces \mathcal{S}^+ and \mathcal{S}^- were not linearly separable. Since $L_c = 2$, $N_a = 8$. We also used $M = 4$ and $M = 8$ for the proposed equalizer. The simulation results are shown in Fig. 8. In the figure, we can see that the performance of the linear equalizers was very poor. This is not surprising since the signal space was not linearly separable. Even for high SNRs, the linear equalizers could not give satisfactory results. Nonlinear equalizers are not affected by this problem: The higher the SNR, the lower the BER. The performance of the proposed equalizer was very close to that of the Bayesian equalizer. When $M = 8$, there was almost no difference. To explain this, we show decision boundaries for the Bayesian equalizer and the proposed equalizer when $M = 4$ in Fig. 9. In the figure, we can clearly see that the decision boundary of the proposed equalizer closely approximated that of the Bayesian equalizer. Only in the central region in Fig. 9 is there some discrepancy. However, this does not contribute significant bit errors. The decision boundary for the proposed equalizer when $M = 8$ is further shown in Fig. 10. Note that the decision boundary of the

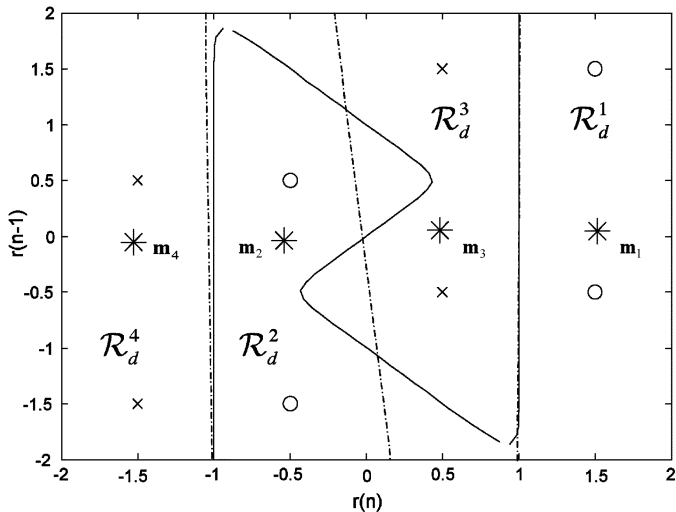


Fig. 9. Decision regions for all subsets \mathcal{R}_d^i , $i = 1, 2, 3, 4$ and decision boundaries of the proposed equalizer when $M = 4$ (dashed line) and the Bayesian (solid line) equalizer.

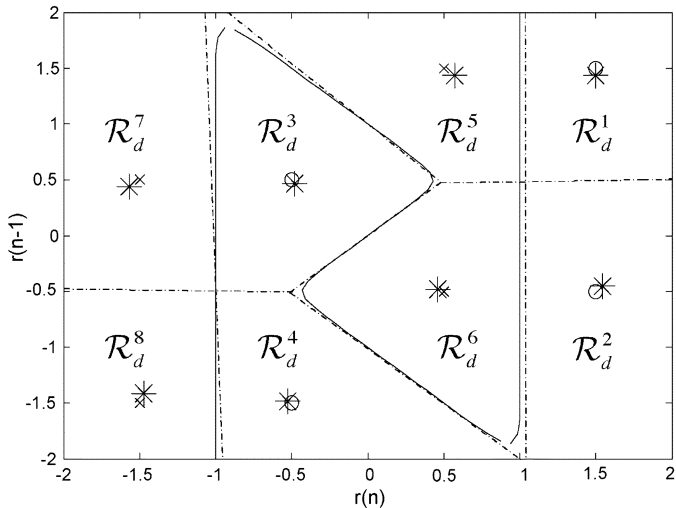


Fig. 10. Decision regions for all subsets \mathcal{R}_d^i , $i = 1, 2, \dots, 8$ and the decision boundaries of the proposed equalizer when $M = 8$ (dashed line) and the Bayesian (solid line) equalizer.

proposed equalizer almost exactly matched that of the Bayesian equalizer. Note also that the pseudo states identified were very close to the true signal states. This was expected since there was only one state in each subset. It is important to realize that even in this case, the computational complexity of the proposed algorithm was still lower than that of the Bayesian equalizer.

We next considered a nonlinear channel, which is a linear channel followed by a memoryless nonlinearity. The input-output relationship of the linear channel is given by

$$q(n) = 0.08x(n) + 0.42x(n-1) + x(n-2) + 0.38x(n-3) - 0.1x(n-4) + 0.09x(n-5). \quad (58)$$

The output of the memoryless nonlinearity is given by

$$r(n) = q(n) + 0.036q^2(n) - 0.01q^3(n) + v(n). \quad (59)$$

For this case, $L_c = 6$, and set $L_e = 4$ and $D = 2$ for the equalizers. It turned out that $N_a = 512$. The number of states was

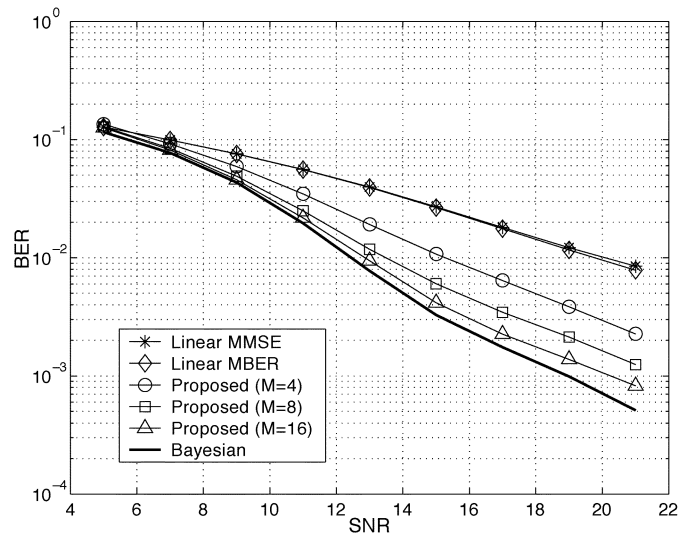


Fig. 11. BER comparison for the MMSE linear, MBER linear, proposed, and Bayesian equalizers.

TABLE III
COMPUTATIONAL COMPLEXITY COMPARISON FOR THE LINEAR, PROPOSED ($M = 8$), AND BAYESIAN EQUALIZERS (FOR THE NONLINEAR CHANNEL IN SIMULATIONS)

	Bayesian	Proposed	Linear
Multiplications	2560	32	4
Additions	4095	32	3
Others	$512 \times \exp(\cdot)$	Compare logics	

then large, and the computational complexity of the Bayesian equalizer became huge. Fig. 11 shows the performance for these equalizers. Each BER result was obtained using 10^8 runs. As the figure shows, the nonlinear equalizers significantly outperformed the linear ones. The proposed equalizer efficiently approximated the optimal Bayesian equalizer. While the performance loss was moderate, the computational complexity reduction was significant. Table III shows a computational complexity comparison for all equalizers (in the decision phase).

V. CONCLUSION

An adaptive nonlinear equalizer is proposed. This equalizer was derived using an approximate Bayesian criterion. It consists of a set of parallel linear discriminant functions followed by a maximum function. The number of functions can be arbitrarily chosen, and the corresponding coefficients can be adaptively trained. Thus, we can have the flexibility to trade performance for reduced computational complexity. We demonstrated that the proposed equalizer can closely approximate the performance of an optimal Bayesian equalizer, whereas its computational complexity remains significantly lower. Due to its adaptive nature, this equalizer is applicable to time-varying channels. The adaptive algorithm with the stochastic gradient descent is simple and robust, which will be a great advantage for real-world applications.

The idea of subset partition can be further extended. For example, if we want to have eight subsets, we may choose $x(n-D)$ and any two other elements from $\mathbf{x}_c(n)$. Using these three el-

ements, we can partition the signal space into eight subsets. This is different from the partitioning method described in this paper, where all three elements of $\mathbf{x}_{c,2}(n)$ must be consecutive. The specific elements to choose may depend on the channel mapping function. An intuitive thought is to choose the ones contributing most energy in the received signal vector. To do that, we may need a channel identification filter. This extended method may make the signal subsets more linearly separable and facilitate decisionmaking.

APPENDIX

Consider an equalizer whose output is determined using the MMSE criterion described below.

$$\min J(n) = E \left\{ (x(n-D) - y(n))^2 \right\}. \quad (60)$$

The optimal solution for this cost function is known to be the conditional mean [19], which is

$$y(n) = E \{ x(n-D) | \mathbf{r}(n) \}. \quad (61)$$

For the equalization problem mentioned above, we have

$$\begin{aligned} E \{ x(n-D) | \mathbf{r}(n) \} &= +1 \cdot P(x(n-D) = +1 | \mathbf{r}(n)) \\ &+ (-1) \cdot P(x(n-D) = -1 | \mathbf{r}(n)). \end{aligned} \quad (62)$$

Let the number of elements in \mathcal{S}^\pm be one. Using the Bayes rule and the fact that $P(x(n-D) = +1) = P(x(n-D) = -1) = 1/2$, we have

$$\begin{aligned} P(x(n-D) = \pm 1 | \mathbf{r}(n)) \\ = \frac{P(x(n-D) = \pm 1) p(\mathbf{r}(n) | x(n-D) = \pm 1)}{p(\mathbf{r}(n))} \end{aligned} \quad (63)$$

where $p(\mathbf{r}(n) | x(n-D) = \pm 1)$ is the *a priori* probability density function (PDF) conditioned on $x(n-D) = \pm 1$, and $p(\mathbf{r}(n))$ is the PDF of $\mathbf{r}(n)$. They are given as follows:

$$\begin{aligned} p(\mathbf{r}(n) | x(n-D) = +1) \\ = \frac{1}{\sqrt{2\pi\sigma_v^2}} \exp \left(-\frac{\|\mathbf{r}(n) - \mathbf{s}_1\|^2}{2\sigma_v^2} \right) \end{aligned} \quad (64)$$

$$\begin{aligned} p(\mathbf{r}(n) | x(n-D) = -1) \\ = \frac{1}{\sqrt{2\pi\sigma_v^2}} \exp \left(-\frac{\|\mathbf{r}(n) - \mathbf{s}_2\|^2}{2\sigma_v^2} \right) \end{aligned} \quad (65)$$

where \mathbf{s}_1 is the only state in \mathcal{S}^+ , and \mathbf{s}_2 is the only state in \mathcal{S}^- . Note that

$$\begin{aligned} p(\mathbf{r}(n)) &= P(x(n-D) = +1) p(\mathbf{r}(n) | x(n-D) = +1) \\ &+ P(x(n-D) = -1) p(\mathbf{r}(n) | x(n-D) = -1). \end{aligned} \quad (66)$$

Substituting (63)–(66) into (61) and (62) and simplifying the result, we obtain

$$y(n) = \frac{\exp \left(-\frac{\|\mathbf{r}(n) - \mathbf{s}_1\|^2}{2\sigma_v^2} \right) - \exp \left(-\frac{\|\mathbf{r}(n) - \mathbf{s}_2\|^2}{2\sigma_v^2} \right)}{\exp \left(-\frac{\|\mathbf{r}(n) - \mathbf{s}_1\|^2}{2\sigma_v^2} \right) + \exp \left(-\frac{\|\mathbf{r}(n) - \mathbf{s}_2\|^2}{2\sigma_v^2} \right)}. \quad (67)$$

REFERENCES

- [1] G. D. Forney, "Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 363–378, Mar. 1972.
- [2] S. Chen, S. McLaughlin, and B. Mulgrew, "Complex valued radial basis function network, Part II: Application to digital communications channel equalization," *Signal Process.*, vol. 35, no. 1, pp. 175–188, Jan. 1994.
- [3] S. Chen, G. J. Gibson, and C. F. N. Cowan, "Adaptive channel equalization using a polynomial-perceptron structure," *Proc. IEE*, pt. 1, vol. 137, no. 5, pp. 257–264, 1990.
- [4] Z. Xiang, G. Bi, and T. Le-Ngoc, "Polynomial perceptrons and their applications to fading channel equalization and co-channel interference suppression," *IEEE Trans. Signal Processing*, vol. 42, pp. 2470–2480, Sept. 1994.
- [5] G. J. Gibson, S. Siu, and C. F. N. Cowan, "Application of multilayer perceptrons as adaptive channel equalisers," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Glasgow, U.K., 1989, pp. 1183–1186.
- [6] —, "The application of nonlinear structures to the reconstruction of binary signals," *IEEE Trans. Signal Processing*, vol. 39, pp. 1877–1884, Aug. 1991.
- [7] S. Chen, G. J. Gibson, C. F. N. Cowan, and P. M. Grant, "Reconstruction of binary signals using an adaptive radial-basis-function equalizer," *Signal Process.*, vol. 22, pp. 77–93, 1991.
- [8] I. Cha and S. A. Kassam, "Channel equalization using adaptive complex radial basis function networks," *IEEE J. Select. Areas Commun.*, vol. 13, pp. 122–131, Jan. 1995.
- [9] J. Cid-Sueira, A. Artes-Rodriguez, and A. R. Figueiras-Vidal, "Recurrent radial basis function networks for optimal symbol-by-symbol equalization," *Signal Process.*, vol. 40, pp. 53–63, 1994.
- [10] S. Chen, S. McLaughlin, B. Mulgrew, and P. M. Grant, "Adaptive Bayesian decision feedback equaliser for dispersive mobile radio channels," *IEEE Trans. Commun.*, vol. 43, pp. 1937–1946, May 1995.
- [11] S. Chen, B. Mulgrew, and P. M. Grant, "A clustering techniques for digital communications channel equalization using radial basis function networks," *IEEE Trans. Neural Networks*, vol. 4, pp. 570–579, July 1993.
- [12] T. Adali, X. Liu, and M. K. Sönmez, "Conditional distribution learning with neural networks and its application to channel equalization," *IEEE Trans. Signal Processing*, vol. 45, pp. 1051–1064, Apr. 1997.
- [13] D. J. Sebald and J. A. Bucklew, "Support vector machine techniques for nonlinear equalization," *IEEE Trans. Signal Processing*, vol. 48, pp. 3217–3226, Nov. 2000.
- [14] S. Chen, S. Gunn, and C. J. Harris, "Decision feedback equaliser design using support vector machines," *Proc. Inst. Elect. Eng. Vis. Image Signal Process*, vol. 147, no. 3, June 2000.
- [15] Y. Kim and J. Moon, "Delay-constrained asymptotically optimal detection using signal-space partitioning," in *Proc. IEEE ICC*, Atlanta, GA, 1998.
- [16] —, "Multidimensional signal space partitioning using a minimal set of hyperplanes for detecting ISI-corrupted symbols," *IEEE Trans. Commun.*, vol. 48, pp. 637–647, Apr. 2000.
- [17] S. Chen, B. Mulgrew, and L. Hanzo, "Asymptotic Bayesian decision feedback equalizer using a set of hyperplanes," *IEEE Trans. Signal Processing*, vol. 48, pp. 3493–3500, Dec. 2000.
- [18] S. Chen, L. Hanzo, and B. Mulgrew, "Decision feedback equalization using multiple-hyperplane partitioning for detecting ISI-corrupted M -ary PAM signals," *IEEE Trans. Commun.*, vol. 49, pp. 760–764, May 2001.
- [19] H. V. Poor, *An Introduction to Signal Detection and Estimation*, 2nd ed. New York: Springer.
- [20] C. C. Yeh and J. R. Barry, "Approximate minimum bit-error rate equalization for binary signaling," in *Proc. Int. Conf. Commun.*, vol. 2, Montreal, QC, Canada, June 1997, pp. 1095–1099.

- [21] S. Chen, E. Chng, B. Mulgrew, and G. Gibson, "Minimum-BER linear-combiner DFE," in *Proc. IEEE Int. Contr. Conf.*, 1996, pp. 1173–1177.
- [22] S. Benedetto and E. Biglieri, "Nonlinear equalization of digital satellite channels," *IEEE J. Select. Areas Commun.*, vol. SAC-1, pp. 57–62, Jan. 1983.
- [23] G. Karam and H. Sari, "Analysis of predistortion, equalization, and ISI cancellation techniques in digital radio systems with nonlinear transmit amplifiers," *IEEE Trans. Commun.*, vol. 37, pp. 1245–1253, Dec. 1989.
- [24] S. U. H. Qureshi, "Adaptive equalization," *Proc. IEEE*, vol. 73, pp. 1349–1387, Sept. 1985.
- [25] C. P. Callender, C. F. N. Cowan, and S. Theodoridis, "Non-linear adaptive equalization using a switched coefficient adaptive filter," in *Proc. IEE Colloq. Adaptive Filtering, Nonlinear Dynamics Neural Networks*, 1991, pp. 3/1–3/4.
- [26] B. H. Juang and S. Katagiri, "Discriminative learning for minimum error classification," *IEEE Trans. Signal Processing*, vol. 40, pp. 3043–3054, Dec. 1992.
- [27] S. Haykin, *Adaptive Filter Theory*, 3rd ed. New York: Prentice-Hall, 1996.



Ren-Jr Chen was born in Taipei, Taiwan, R.O.C., in May 1975. He received the B.E. degree in electrical engineering from Feng Chia University, Taichung, Taiwan, in 1997 and the M.S. degree in electrical engineering from National Tsing Hua University, Hsinchu, Taiwan, in 1999. Since 1999, he has been with the Department of Communication Engineering, National Chiao Tung University, Hsinchu, where he is currently pursuing the Ph.D. degree.

His research interests include communication signal processing, mainly adaptive nonlinear

equalization.



Wen-Rong Wu (M'89) received the M.S. and Ph.D. degrees in electrical engineering from State University of New York at Buffalo in 1986 and 1989, respectively.

Since August 1989, he has been a faculty member with the Department of Communication Engineering, National Chiao Tung University, Hsinchu, Taiwan, R.O.C. His research interests include statistical signal processing and digital communications.