A Binary Adaptive Decision-Selection Equalizer for Channels With Nonlinear Intersymbol Interference

Daniel J. Sebald, *Member, IEEE*, and James A. Bucklew

***Abstract*—An enhanced adaptive decision feedback equalizer (ADFE) is presented for binary data transmission applications where the communication channel exhibits nonlinear intersymbol interference (ISI). The nonlinearity in the channel manifests itself as a distorted constellation space constructed from the equalizer input state variables. Since a conventional ADFE can construct a hyperplane decision boundary of only one orientation with symmetrically spaced distance from the origin as a function of the detected feedback symbols and feedback filter coefficient values, there is room for improvement since the distorted constellation of the nonlinear system is better served by hyperplane boundaries of varying orientation.**

The method proposed here is not to feed back the decision variables but, instead, to use these binary variables to choose and adapt different sets of coefficients, i.e., different hyperplane boundaries. Hence, the name given to this new method is the adaptive decision-selection equalizer (ADSE). Although the hyperplane may not be the optimum boundary for the conditional constellations, in many cases, it is an adequate approximation. Nonetheless, for nonlinear channels, the ADSE is generally an improvement over the conventional ADFE in high signal-to-noise ratio (SNR) regimes, where the bit error rate (BER) is within the desired operating range.

The major advantage of the new method is improved performance on the studied channel while retaining simplicity when implemented as a variation of the least-mean-squared (LMS) algorithm. Some drawbacks are decreased convergence rate and limitations of the minimum mean-squared-error (MMSE) strategy of optimization, as implemented by the LMS algorithm, for a system where error probability, not MMSE, is important.

***Index Terms*—Adaptive equalizers, communication channels, communication system nonlinearities, decision feedback equalizers, nonlinear detection, nonlinear systems.**

I. Introduction

ADAPTIVE equalizers were first proposed by Lucky [1], who used a sign-based update algorithm to minimize a distortion measure of a tapped-delay line filter. Application of the LMS algorithm to the adaptive transversal equalizer is studied in Niessen and Drouilhet [2]. (See also Proakis and Miller [3] for greater detail.) Austin [4] introduces a nonlinear slicer and linear feedback loop to the nonadaptive transversal equalizer and creates the decision feedback equalizer (DFE) to greatly improve performance over the transversal equalizer. The LMS-based adaptive version of the DFE (the ADFE), is proposed and studied in George *et al.* [5]. (See also Proakis [6] for further studies.)

Fig. 1 illustrates the ADFE in a binary pulse amplitude modulation (PAM) scenario at baseband. An independent, identically distributed (i.i.d.) information signal $u(n) \in \{+1, -1\}$ is passed through a nonlinear deterministic channel encompassing transmission pathway effects, the receiver filter, and decorrelation signal processing. In our model, a zero mean, additive white Gaussian noise (AWGN) $\eta(n) \in \mathbb{R}$ is added to the channel output $\hat{x}(n) \in \mathbb{R}$ to form the equalizer input $x(n)$. The detector consists of two linear, discrete-time subnetworks. The input is passed through an adaptive, delay network with coefficients $w_k(n) \in \mathbb{R}$, $k = 0, \ldots, N_w - 1$, whereas the past decisions $\hat{u}(n - D) \in \{+1, -1\}$ are fed back through another adaptive, delay network with coefficients $b_k(n) \in \mathbb{R}$, $k = 1, \ldots, N_b$. A training sequence (position A in Fig. 1) $\psi(n) = u(n - D)$ is used to adapt the ADFE starting from some initial set of coefficients. After training, the ADFE is switched over to decision-directed mode (position B in Fig. 1). A processing delay D must be introduced to compensate for the delay of the channel and length of the equalizer feedforward filter. If N_c is the memory length of the channel, then a general rule is to select N_b, which is the number of feedback variables, to be approximately $N_c - 1$. The reason for this is that the feedback filter cancels ISI, and there are $N_c - 1$ past symbols contributing to the ISI. [Note that "past" is referenced to delay D, e.g., $\hat{u}(\ell - D - 1)$ is the most recent past decision with regard to input $u(n)$ when $n = \ell$.]

Define vectors

$$\begin{aligned}\mathbf{x}(n) &= [x(n) \ \cdots \ x(n - N_w + 1)]^T \\ \mathbf{w}(n) &= [w_0(n) \ \cdots \ w_{N_w-1}(n)]^T \\ \hat{\mathbf{u}}(n) &= [\hat{u}(n - D - 1) \ \cdots \ \hat{u}(n - D - N_b)]^T \\ \mathbf{b}(n) &= [b_1(n) \ \cdots \ b_{N_b}(n)]^T.\end{aligned}$$

The LMS algorithm [7] for adjusting filter tap coefficients takes the form

$$\begin{aligned}\mathbf{w}(n+1) &= \mathbf{w}(n) + \mu\, e(n)\, \mathbf{x}(n) \\ \mathbf{b}(n+1) &= \mathbf{b}(n) + \mu\, e(n) \hat{\mathbf{u}}(n)\end{aligned}$$

where

$$e(n) = \text{sign}(y(n)) - y(n)$$

and

$$y(n) = \mathbf{w}^T(n)\, \mathbf{x}(n) + \mathbf{b}^T(n)\, \hat{\mathbf{u}}(n)$$

Manuscript received December 14, 2000; revised May 29, 2002. The associate editor coordinating the review of this paper and approving it for publication was Prof. Colin F. N. Cowan.

D. J. Sebald is a freelance engineer in Madison, WI 53711 USA (e-mail: daniel.sebald@ieee.org).

J. A. Bucklew is with the Department of Electrical and Computer Engineering, University of Wisconsin, Madison, WI 53766 USA.

Publisher Item Identifier 10.1109/TSP.2002.801913.

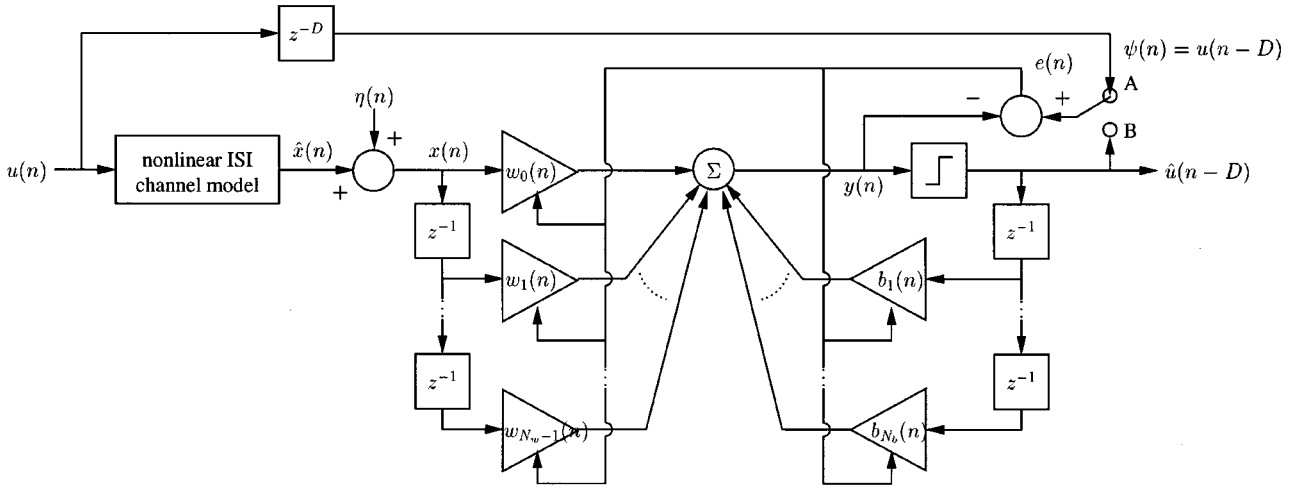


Fig. 1. Block diagram of an ADFE. When the switch is in position A, the ADFE is in training mode. Position B corresponds to decision-directed mode.

except during training, in which case

$$e(n) = \psi(n) - y(n).$$

The goal is to drive the energy of the error signal $e(n)$ to a minimum, i.e., adapt $\mathbf{w}(n)$ and $\mathbf{b}(n)$ so that the preslicer filter output is as close as possible in the MMSE sense to a valid symbol. Update constant μ is chosen large enough for sufficient tracking yet small enough to ensure stability and small residual misadjustment for increased accuracy [7].

The ADFE has remained a useful equalizing technique because of its adaptive nature for unknown channels as well as its simplicity of implementation. However, there has been interest in equalizing data communication channels exhibiting nonlinear behavior [8]–[12], an application for which an ADFE often does not work well since it contains only linear filtering components, as shown in Fig. 1.

One of the original approaches to equalizing a nonlinear ISI channel is [13], where a Volterra series replaces the linear feedback and feedforward portions of the ADFE. Research on a fast transform-domain version of Volterra filtering is discussed in [14]. A more recent detection strategy is the adaptive Bayesian neural network of [10]. There, it is shown how conditioning on the decision state can be viewed as a reduction in the pattern space constellation formed by the equalizer input variables, which is a phenomenon that is central to our method proposed in this paper. In [15], an adaptive Kalman filter is proposed for improved equalization performance on linear ISI channels, and it is suggested therein that the method may be extended to nonlinear equalization.

In [12], the authors study a method of nonlinear equalization based on the support vector machine (SVM) [16], [17]. The decision feedback strategy is used in [12] by simply feeding past decisions into the SVM input. Although this improves performance noticeably, we argue that the binary nature of the decision is not suited for the SVM input, which typically is a real-valued variable, and propose a modification of the feedback idea whereby the past decisions are used to select from a set of SVMs. The various SVMs are trained on subsets of data, conditioned on the past decisions $\hat{\mathbf{u}}(n)$. This decision-selection method shows even further improvement over the decision feedback approach. The SVM, in its current state of research, is a block adaptive algorithm as opposed to a symbol-by-symbol adaptive algorithm.

In this paper, we investigate the same approach of selecting a detector model based on previous decisions but with conventional linear elements of the ADFE as opposed to the nonlinear mappings of the SVM. It is true that given the appropriate kernel for the SVM, the linear model is simply a subset of that which the SVM can provide, and hence, the investigation to follow may be encompassed by that of [12]. However, the merits of linear elements warrant more specific consideration. The training algorithm for the ADSE is based on the LMS algorithm, and therefore, the proposed method is adaptive on a symbol-by-symbol basis. It is useful for slowly varying channels, just as is the ADFE. The ADSE retains the simplicity of the ADFE and has the potential to significantly improve performance, depending on the nature of the channel nonlinearity. The only added overhead is memory. We are interested in these simple structures, although they may be suboptimum, because in many applications, speed and simplicity are of overriding importance.

In Section II, we present the baseband model for nonlinear channels often seen in voiceband digital communication links. Section III discusses the concept of the ADSE. First, an example input space is used to illustrate conditional constellations. From that, the motivation for the ADSE becomes evident. Section IV gives results of some simulations on nonlinear ISI channels. We conclude in Section V with comments about ADSE characteristics and limitations of the LMS-based approach.

II. Channel Model

The linear channel model is described by[1]

$$\hat{x}(n) = \sum_{k=0}^{N_c-1} h_k u(n-k)$$

[1]Technically, the coefficients should be $h_k(n)$ for time-varying channels. However, we leave out the time-varying index to avoid confusing notation. The same holds true for $c_p(n)$.

where h_k is the causal finite impulse response (FIR) model of the channel, normalized to have unit energy, i.e., $\sum_{k=0}^{N_c-1} h_k^2 = 1$. Let σ_η^2 be the variance of the zero mean AWGN. Then, for the linear channel, the SNR is

$$\mathrm{SNR}_{\mathrm{linear}} = 1/\sigma_\eta^2$$

because of the unit energy property of the channel and channel input symbols.

Typically, the nonlinear channels of interest have a memoryless nonlinearity in combination with a linearly dispersive element [8]. Therefore, we will examine two basic nonlinear models and assume that the adaptive equalizer is to be used on channels having mild or, at least, less than severe nonlinearities. The two types of nonlinearity are the simple Wiener model and the simple Hammerstein model [18]. Let $\tilde{x}(n)$ be an intermediate variable. The Wiener model is a linear FIR filter followed by a polynomial nonlinearity, i.e.,

$$\tilde{x}(n) = \sum_{k=0}^{N_c-1} h_k u(n-k), \qquad \hat{x}(n) = \sum_{p=1}^{P} c_p \tilde{x}^p(n)$$

where $\{h_k\}$ are FIR filter coefficients, and $\{c_p\}$ are polynomial coefficients. Because the AWGN is added after the dispersion and nonlinearity, it is straightforward to adjust the SNR definition. Let $\mathbf{v}_j$ for $j = 1, \ldots, 2^{N_c}$ be all permutations of the channel input space $\{+1, -1\}^{N_c}$, and arrange the FIR coefficients in vector form as $\mathbf{h} = [h_0\, h_1 \cdots h_{N_c-1}]^{\mathbf{T}}$. Then, the average SNR is

$$\overline{\mathrm{SNR}}_{\mathrm{Wiener}} = \frac{1}{2^{N_c}\sigma_\eta^2} \sum_{j=1}^{2^{N_c}} \left(\sum_{p=1}^{P} c_p (\mathbf{h}^T \mathbf{v}_j)^p \right)^2.$$

The Hammerstein model interchanges the dispersion and polynomial operations

$$\tilde{x}(n) = \sum_{p=1}^{P} c_p u^p(n), \qquad \hat{x}(n) = \sum_{k=0}^{N_c-1} h_k \tilde{x}(n-k).$$

Again, adjustment of the definition for average SNR is

$$\overline{\mathrm{SNR}}_{\mathrm{Hammerstein}} = \frac{1}{2\sigma_\eta^2} \left(\left(\sum_{p=1}^{P} c_p \right)^2 + \left(\sum_{p=1}^{P} c_p (-1)^p \right)^2 \right)$$

which is independent of the channel dispersion because of the unity power assumption of the impulse response.

III. Adaptive Decision-Selection Equalization

A. Pattern Spaces and the ADFE Limitation for Nonlinear Channels

The detection problem may be viewed as a pattern recognition problem where the states of the equalizer feedforward filter are input variables. A good presentation of this viewpoint is included in Chen *et al.* [10]. Since $x(n)$ is a function of $u(n), \ldots, u(n - N_c + 1)$, the input space $\mathbf{x}(n)$ is a function of $u(n), \ldots, u(n - N_w - N_c + 1)$. Thus, the noise-free system has $2^{N_w+N_c-1}$ (not necessarily unique) constellation points $\hat{\mathbf{x}}_k(n) \in \mathbb{R}^{N_w}$, $k = 1, \ldots, 2^{N_w+N_c-1}$ grouped into two sets

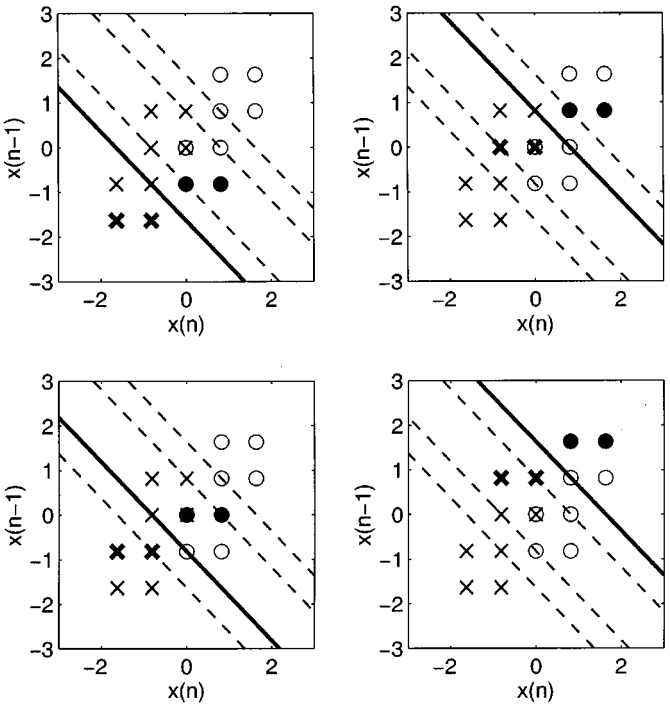


Fig. 2. Conditional constellations for a two-input ($N_w = 2$), two decision state ($N_b = 2$), one delay ($D = 1$) equalizer where the channel is linear ISI with $\hat{x}(n) = 0.4084\,u(n) + 0.8164\,u(n-1) + 0.4084\,u(n-2)$ conditioned on the correct past decision variables.

$$C_{\pm 1,\,D} = \{\hat{\mathbf{x}}_k(n) | u(n-D) = \pm 1\}$$

one set for each desired classification. We can further partition the constellation sets based on the 2^{N_b} possible past states. Similar to the definition of the equalizer decision state $\hat{\mathbf{u}}(n)$, define the correct past symbols as

$$\mathbf{u}(n) = [u(n-D-1) \cdots u(n-D-N_b)]^T$$

and let $\mathbf{u}_j$, $j = 1, \ldots, 2^{N_b}$ be an enumeration of all the elements of $\{+1, -1\}^{N_b}$. Then, the conditional constellations take the form

$$C_{\pm 1,\,D,\,j} = \{\hat{\mathbf{x}}_k(n) | u(n-D) = \pm 1,\, \mathbf{u}(n) = \mathbf{u}_j\}.$$

As an example, Fig. 2 illustrates the constellation for a channel with linear ISI $\hat{x}(n) = 0.4084\,u(n) + 0.8164\,u(n-1) + 0.4084\,u(n-2)$ and equalizer $N_w = 2$, $N_b = 2$ and $D = 1$. Concentrating on just one of the subfigures and ignoring the lines for the moment, points belonging to $C_{+1,1}$ are marked $\circ$ or $\bullet$, and points belonging to $C_{-1,1}$ are marked $\times$ or $\mathbf{\times}$. Since the channel has memory $N_c = 3$ and feedforward filter length $N_w = 2$ in this example, the cardinality of each set is $2^{N_w+N_c-1}/2 = 8$ since members within sets are unique. However, note that this channel has severe ISI because a member of one set is very near a member of the other set. For this reason, symbol-based equalizers without decision feedback perform poorly on this channel. The four separate subfigures of Fig. 2 show the conditional constellations where points belonging to $C_{+1,1,j}$ are marked $\bullet$ and where points belonging to $C_{-1,1,j}$ are marked $\times$.

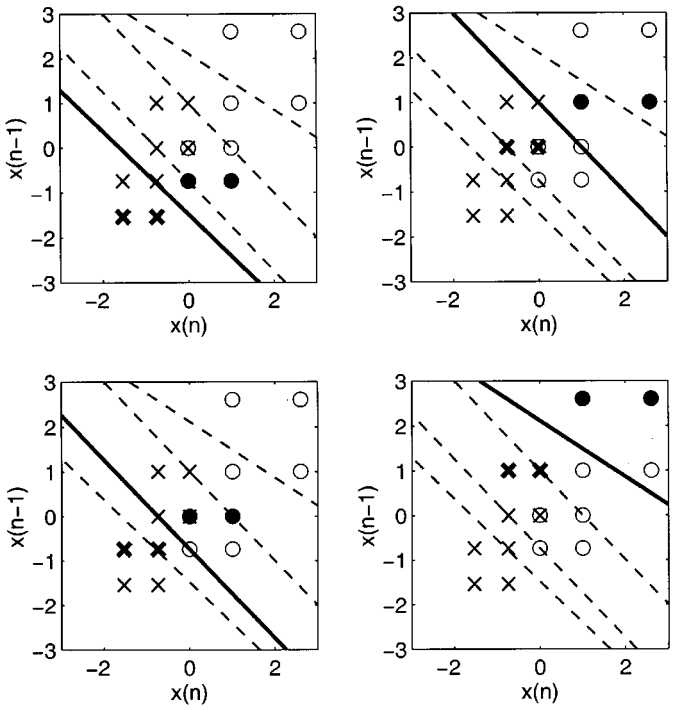


Fig. 3. Conditional constellations for a two-input, two-decision, one-delay equalizer where the channel is nonlinear ISI with $\bar{x}(n) = 0.4084\,u(n) + 0.8164\,u(n-1) + 0.4084\,u(n-2)$ and $\hat{x}(n) = \tilde{x}(n) + 0.2\,\tilde{x}^2(n) + 0.1\tilde{x}^3(n)$.

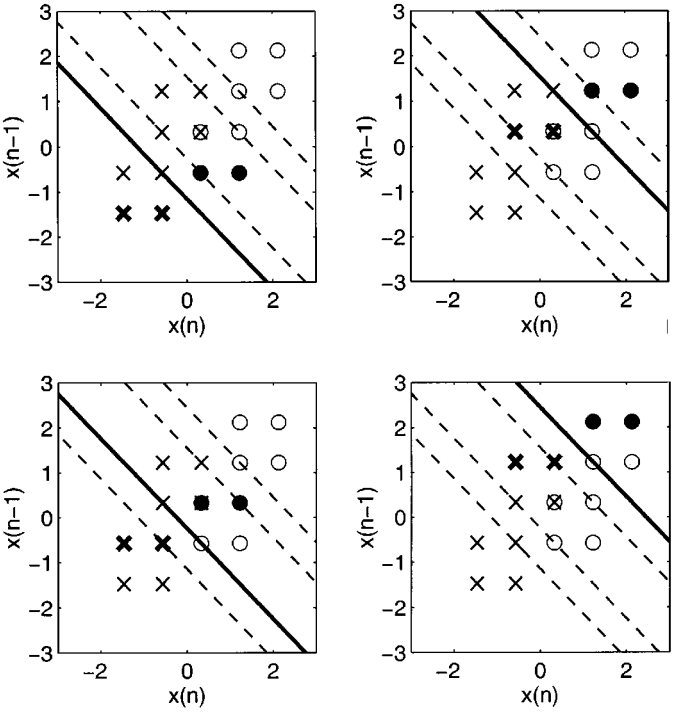


Fig. 4. Conditional constellations for a two-input, two-decision, one-delay equalizer where the channel is nonlinear ISI with $\bar{x}(n) = u(n) + 0.2\,u^2(n) + 0.1u^3(n)$ and $\hat{x}(n) = 0.4084\,\tilde{x}(n) + 0.8164\,\tilde{x}(n-1) + 0.4084\,\tilde{x}(n-2)$.

If we consider the space of input variables $\mathbf{x} \in \mathbb{R}^{N_w}$ and fix $\hat{\mathbf{u}}(n) = \mathbf{u}_j$, $j \in \{1, 2, \ldots, 2^{N_b}\}$, setting the preslicer output $y(n)$ to zero defines a hyperplane in the input space, i.e.,

$$\mathcal{H}_j = \{\mathbf{x}\colon \mathbf{w}^T(n)\,\mathbf{x} + \mathbf{b}^T(n)\,\mathbf{u}_j = 0\}$$

which serves as the decision boundary for the detector. We will call the set of such hyperplanes the *conditional* hyperplanes. Assuming the correct decisions are in the decision state vector—a reasonable assumption when operating in the low probability of error regime—and restricting the class of detectors to be linear hyperplanes, decision regions are superimposed over the pattern spaces of Fig. 2 using a Voronoi partition [19] of the two closest points among $C_{+1,1,j}$ and $C_{-1,1,j}$ as a rough approximation to the equalizer providing the smallest probability of error among all suboptimum linear detectors.

An interesting property of the constellation sets $C_{\pm 1, D}$ in the linear ISI case (see Fig. 2) is that they are symmetric about the origin since they are formed from a linear combination of input symbols belonging to $\{+1, -1\}$. A similar property exists for the hyperplanes that can be constructed by an ADFE. As illustrated in Fig. 2, the feedback portion of an ADFE ostensibly selects a different ISI removal constant based on the decision state, effectively changing the conditional hyperplane boundary distance from the origin. Since the input to the feedback filter belongs to $\{+1, -1\}$, the hyperplanes must be distanced symmetrically about the origin.

If we now examine the constellation in the case of nonlinear ISI, we find that symmetry about the origin no longer exists. Fig. 3 shows how the pattern space is distorted in the case of a cubic Wiener nonlinear channel model with $\tilde{x}(n) = 0.4084\,u(n) + 0.8164\,u(n-1) + 0.4084\,u(n-2)$ and $\hat{x}(n) = \tilde{x}(n) + 0.2\,\tilde{x}^2(n) + 0.1\tilde{x}^3(n)$. Again, the Voronoi partition of the two closest points among the constellation sets serves as an approximation to the best linear suboptimum detector. Clearly, the best conditional hyperplanes are no longer parallel, nor are they positioned symmetrically about the origin. Since the ADFE can only construct parallel hyperplanes with symmetric distances about the origin, it is no surprise that ADFE performance in the nonlinear scenario degrades.

As a second example, consider the cubic Hammerstein channel model where the filter and nonlinear operation of the Wiener channel are interchanged, the pattern space for which is shown in Fig. 4. The optimum conditional hyperplanes are now parallel, which is good for the ADFE solution. However, the whole constellation is shifted toward the first quadrant, and once again, the ADFE will perform poorly on such a channel. The experience of Biglieri *et al.* [8] is that when the linear dispersion follows the nonlinearity (i.e., Hammerstein), distortion is not as severe as when the nonlinearity follows the linear dispersion (i.e., Wiener). The illustrations of Figs. 3 and 4 attest to this, but certainly, this is no argument for generalization.

B. Benchmark

As a benchmark detector, we use a Bayesian classifier for equalizers with input parameters defined similar to the ADFE of Section I. A probability density function is constructed for the input space under the two classes, and a hypothesis test decides the output symbol, i.e.,

$$\lambda_{\mathbf{x}(n)}(\mathbf{x}|u(n-D) = +1, \mathbf{u}(n) = \mathbf{u}_j) \underset{H_0}{\overset{H_1}{\gtrless}} \lambda_{\mathbf{x}(n)}(\mathbf{x}|u(n-D) = -1, \mathbf{u}(n) = \mathbf{u}_j) \qquad (1)$$

where hypothesis H_0 declares $\hat{u}(n-D)$ to be -1, hypothesis H_1 declares $\hat{u}(n-D)$ to be $+1$, and

$$\lambda_{\mathbf{x}(n)}(\mathbf{x}|u(n-D)=\pm 1, \mathbf{u}(n)=\mathbf{u}_j) = \sum_{\{k:\hat{\mathbf{x}}_k(n)\in C_{\pm 1,D,j}\}} 2^{-(N_w+N_c-N_b-2)} \lambda_{\mathbf{x}_k(n)}(\mathbf{x})$$

where $\lambda_{\mathbf{x}_k(n)}(\mathbf{x})$ is the probability density associated with random vector $\mathbf{x}_k(n) = \hat{\mathbf{x}}_k(n) + \boldsymbol{\eta}(n)$, and $\boldsymbol{\eta}(n)$ is an N_w-length random vector with independent components distributed similar to $\eta(n)$. In our channel model with zero mean, AWGN

$$\lambda_{\mathbf{x}_k(n)}(\mathbf{x}) = \frac{1}{(2\pi)^{N_w/2}\sigma_\eta^{N_w}} \exp\left(\frac{-\|\mathbf{x}-\hat{\mathbf{x}}_k(n)\|^2}{2\sigma_\eta^2}\right).$$

C. ADSE Concept

There are several reasons for staying with the class of hyperplanes for decision boundaries, although it was clearly shown in [10] that methods such as neural networks with Gaussian kernels can be used to achieve performance near that of the Bayesian classifier. First and foremost, the hyperplane is the easiest to work with in terms of processing and adapting. It classifies efficiently, and the LMS algorithm may be used for time varying channels. (However, there is a caveat with the LMS algorithm, as discussed later.) Second, if the added noise does not match the model used in the neural network method (or any method assuming a distribution for the noise), performance may suffer. Third, in the case of a memoryless nonlinearity, the decision conditioning method may reduce the pattern space to something where a hyperplane is a good approximation to the optimum boundary. This phenomenon was observed and utilized in [10].

Given what we have seen, a logical progression—if we want to restrict the detector to the class of linear hyperplanes—is to use the decision state to pick a different hyperplane as opposed to filtering and feeding back the decision state. That is, there are 2^{N_b} possible states, and for each state, we assign a different set of feedforward filter coefficients $\mathbf{w}_j(n)$ and ISI removal constant $b_j(n)$ for $j = 1, \ldots, 2^{N_b}$. In this way, the conditional hyperplanes can achieve nonsymmetric distances about the origin as well as different orientations. Let γ: $\{+1, -1\}^{N_b} \mapsto \{1, \ldots, 2^{N_b}\}$ be a 1-to-1 function for choosing which classifier to use based on the decision state.[2] Then, the general LMS-based ADSE algorithm is described in Table I and illustrated in Fig. 5.

As a contrast to the selection idea, consider replacing the linear filter of the ADFE feedback loop in Fig. 1 with a Volterra series, which is the method studied in [8] and [13]. As noted in [8], such an approach must deal with a very large number of parameters. In [13], the number of parameters is reduced by manually selecting those of significance for the typical voiceband telephone channel, which is a somewhat undesirable practice. The parameters of the model adapt using a MMSE gradient descent algorithm. Conveniently, the cost is a linear function of the adapted parameters, yet the update algorithm is rather complex. The important point is that a Volterra filter (or any nonlinear model for that matter) in the feedback loop is somewhat extraneous since there are only 2^{N_b} values input to the filter. Rather than use a sophisticated nonlinear model to map 2^{N_b} points for removing ISI, the current proposed approach is much easier to implement. Furthermore, the ADSE has full degrees of freedom with regard to ISI removal.

[2]The mapping can be anything since adaptive equalizer coefficients are initialized to the same values for the various conditional hyperplanes.

TABLE I
LMS-BASED ADSE ALGORITHM WHERE HYPERPLANE ORIENTATION AND DISTANCE FROM ORIGIN ARE ALLOWED TO VARY AS A FUNCTION OF THE DECISION STATE

0) Initialize variables: $n = 0$, $b_j(0) = 0$ and

$$w_{k,j}(0) = \begin{cases} 1, & k = D \\ 0, & \text{otherwise} \end{cases}$$

for $j = 1, 2, \ldots, 2^{N_b}$.

1) Filter data:

$$y(n) = \mathbf{w}^T_{\gamma(\hat{\mathbf{u}}(n))}(n)\,\mathbf{x}(n) + b_{\gamma(\hat{\mathbf{u}}(n))}(n)$$

2) Compute decision error:

$$e(n) = \text{sign}(y(n)) - y(n)$$

3) Update coefficients:

$$\mathbf{w}_{\gamma(\hat{\mathbf{u}}(n))}(n+1) = \mathbf{w}_{\gamma(\hat{\mathbf{u}}(n))}(n) + \mu\, e(n)\,\mathbf{x}(n)$$
$$b_{\gamma(\hat{\mathbf{u}}(n))}(n+1) = b_{\gamma(\hat{\mathbf{u}}(n))}(n) + \mu\, e(n)$$

and

$$\mathbf{w}_j(n+1) = \mathbf{w}_j(n)$$
$$b_j(n+1) = b_j(n)$$

for $j = 1, 2, \ldots, 2^{N_b}$, $j \neq \gamma(\hat{\mathbf{u}}(n))$.

4) Repeat: increment n, goto step 1.

It is true that when N_b is chosen too large, the ADSE will exhibit problems of excessive parameterization similar to the Volterra filter. Excessive parameterization is a generic problem in the implementation of any nonlinear scheme. Excess parameters essentially act as nuisance parameters and adversely effect system performance. Furthermore, large N_b means that large amounts of training data are necessary. However, we have the option of not making all the decision states be part of the selection process. It is typically the case that the most recent past decisions are the most influential on ISI. Therefore, it is our opinion that the most recent decisions can be used for selection, whereas the older decisions are used in the ADFE fashion if computational complexity is an issue.

D. Discussion About LMS Algorithm

Although the idea of selecting different hyperplanes seems logical, there is a minor problem with this approach—one not so much rooted in the conditioning process but more having to do with the behavior of MMSE-based algorithms. The MMSE solution, which the LMS algorithm stochastically approaches, does not necessarily achieve the minimum error probability. To illustrate this, a conditional pattern space ($j = 2$) for the system used to generate Fig. 3 is isolated in Fig. 6. The Bayesian solution restricted to hyperplanes (the solid line $\mathcal{H}_2^{\text{Bayes}}$) and the empirical MMSE solution restricted to hyperplanes (the dashed

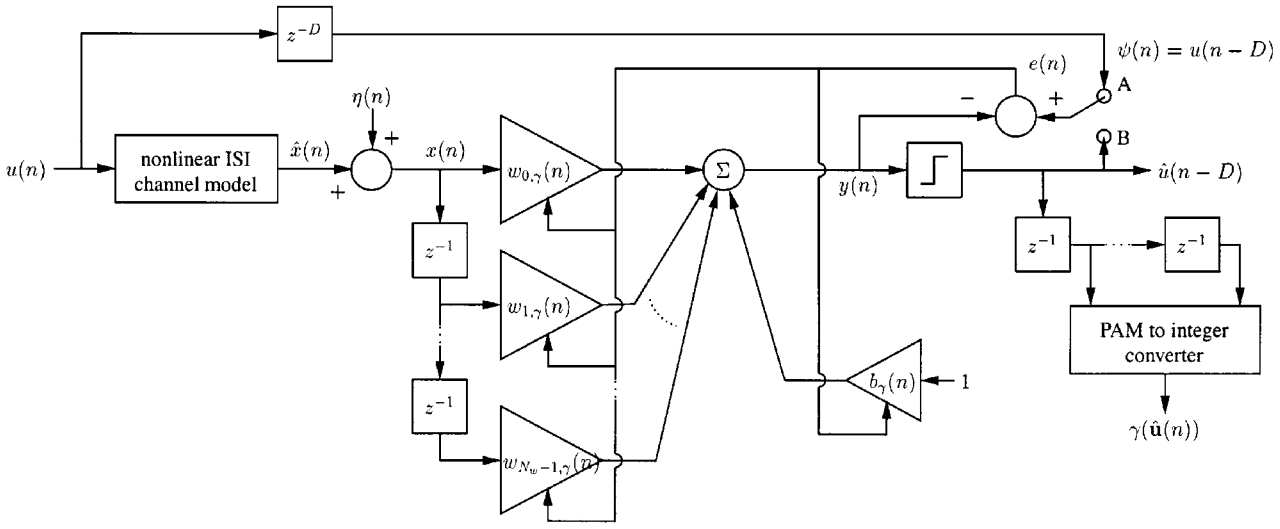


Fig. 5. Block diagram of the ADSE. Rather than feeding back past decisions as in Fig. 1, the PAM to integer converter selects separating hyperplane constant b_γ and orientation $\mathbf{w}_\gamma$.

line $\mathcal{H}_2^{\mathrm{MMSE}}$) are also shown. The Bayes solution restricted to hyperplanes is approximated by the Voronoi partition for the two nearest points between $C_{+1,1,2}$ and $C_{-1,1,2}$. The empirical MMSE solution was constructed by simulating the ADSE for the Wiener nonlinear channel. It is shown in [20] that the asymptotic linear MMSE boundary for an example of this nature is a line perpendicular to the y-axis, bisecting the two nearest points of the projections of $C_{+1,1,2}$ and $C_{-1,1,2}$ onto the y-axis. This is, in fact, the result for a noise-free ADSE simulation. What is shown in Fig. 6 is the MMSE boundary that occurs for a simulation with SNR of 17 dB, and we see that the boundary deviates somewhat from the asymptotic solution.

Because of the ISI, the orientation of the MMSE solution will not necessarily be near the desired Bayesian solution. Conditioning on past symbol decisions reduces the constellation, which is certainly desirable, but it also creates a twisted pattern space in this example. If one imagines Gaussian distributions centered about the points of the constellation, it should be clear how performance degrades using an MMSE-based algorithm. Performance may not degrade too significantly in this two-dimensional scenario, but when extending to higher dimensions, the result is precarious.

IV. Monte Carlo Simulations

Chen *et al.* [10] show that under the assumption that the decision state is correct—which is a mild assumption when operating in the low BER regime—choosing the number of feedforward input variables one greater than the decision delay, i.e., $N_w = D + 1$, yields an equalizer performance as good as when the number of input variables is more than one greater than the decision delay, i.e., $N_w > D + 1$. We argue the same is approximately true in the case of the ADSE. The last element of the input vector, i.e., $x(n - N_w + 1)$, is a function of $u(n - N_w + 1), \ldots, u(n - N_w - N_c + 2)$. Hence, any $x(n - N_w + 1)$ with $N_w > D + 1$ is independent of $u(n - D)$ since the transmitted binary signal is assumed i.i.d. These additional input variables act as nuisance parameters if included as

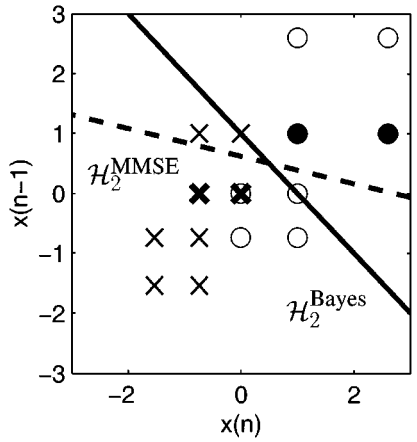
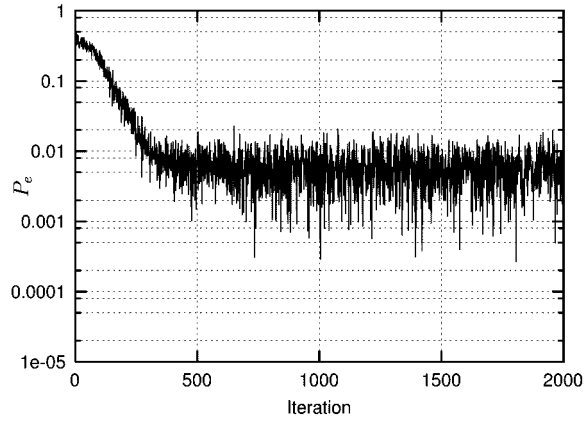


Fig. 6. Conditional constellation ($j = 2$) subject to 17 dB AWGN for the system used to generate Fig. 3. By imagining $\mathbb{R}^2$ Gaussian distributions about the constellation points, one can see that the MMSE hyperplane $\mathcal{H}_2^{\mathrm{MMSE}}$ will lead to a higher error probability decision than does the Bayesian hyperplane $\mathcal{H}_2^{\mathrm{Bayes}}$.

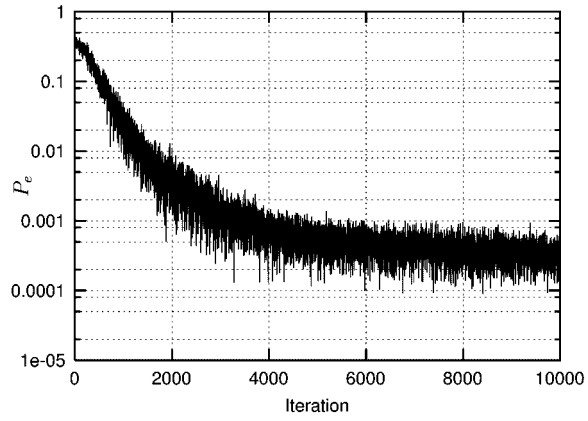
part of the feedforward filter input.[3] A general principle is to choose D long enough to include most of the significant energy of the channel impulse response. When $N_w = D+1$, $\mathbf{x}(n)$ is a function of $u(n), \ldots, u(n-D-N_c+1)$. Therefore, the useful past decision states are $u(n-D-1), \ldots, u(n-D-N_c+1)$, and $N_b = N_c - 1$ is adequate. Naturally, with the channel unknown, the designer must make reasonable approximations for N_w, D, and N_b.

We now give some Monte Carlo simulation results that elucidate the properties of the ADSE. The first set of simulations (see Fig. 7) are ADFE and ADSE probability of error convergence during training for the Wiener nonlinear system presented

[3]This is not precisely accurate. After all, a term that is dependent on previous states $u(n - D - 1), \ldots, u(n - D - N_b)$ selects the ISI removal constant, and $x(n - N_w - 1)$ for $N_w > D + 1$ is certainly correlated with $u(n - D - 1), \ldots, u(n - D - N_b)$, but where to draw the line and truncate this tail of weakly correlated parameters is a difficult question—the same question the Viterbi algorithm must address. Our empirical observation is that if there is some benefit to choosing N_w slightly greater than $D + 1$, it is not statistically significant in the channel we have studied.



(a)

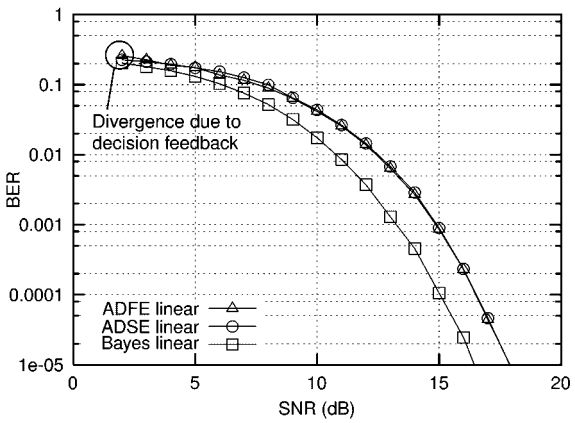


(b)

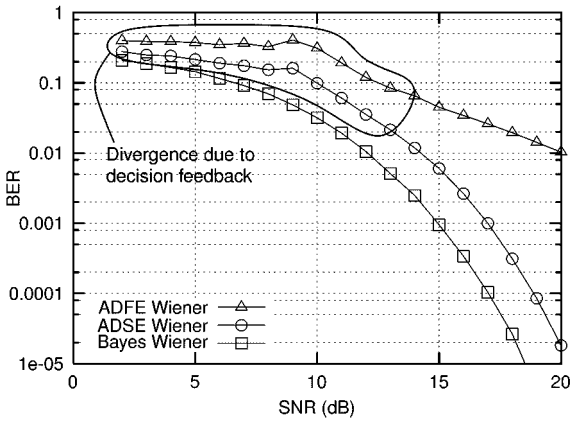
Fig. 7. Convergence results for the (a) ADFE and (b) ADSE on the Wiener nonlinear channel of Section III, with $N_b = 2, N_w = 6, D = 5$ and $\mu = 0.005$ at SNR = 17 dB. The curves are 100 ensemble averaged trials.

in Section III with 17 dB SNR. Since $N_c = 3$, the number of feedback or decision states is chosen as $N_b = 2$. In addition, $N_w = 6$, $D = 5$, and $\mu = 0.005$. The probability of error is computed as described in the Appendix, given the random orientation of the conditional hyperplanes while they converge. Note that the error probability in training mode is essentially the probability of error under the condition that the feedback or selection state is correct. It does not account for propagation of errors, and therefore, the probability of error limits in Fig. 7 are biased toward better performance than the estimates we will find in later results [i.e., the 17 dB result of Fig. 8(b)].

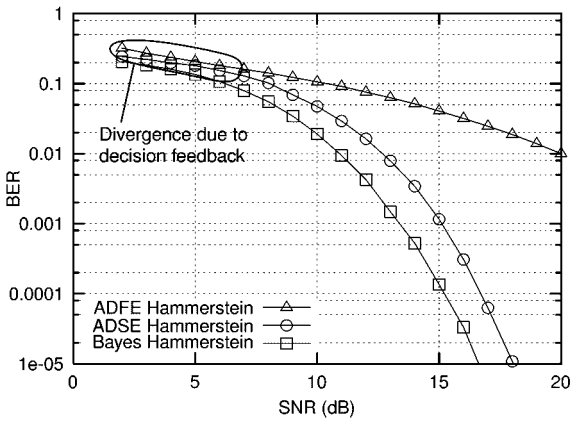
The convergence of the probability of error is not of the same nature as the convergence of the MSE. The ADSE result of Fig. 7(b) does exhibit an exponential decay, but it is on a logarithmic scale. The ADFE result of Fig. 7(a) shows a similar behavior, but the exponential nature is not evident because the ADFE converges quickly to a steady state with more misadjustment error. The ADSE takes approximately 3000 iterations to converge, whereas the ADFE takes approximately 400 iterations to converge. Since $N_b = 2$, meaning that the ADSE has four hyperplanes, this difference in convergence rate is as expected, allowing for the fact that ADFE convergence is reached sooner. One approach to increasing ADSE convergence would be to operate in an ADFE mode for a short period of time and then use the resulting hyperplane as first estimates for the various ADSE conditional hyperplanes.



(a)



(b)



(c)

Fig. 8. Simulation results for ADSE with $N_b = 2$, $N_w = 6$, $D = 5$, and $\mu = 0.005$ on the channels of Section III.

Given the above results for convergence, simulations for steady-state performance as a function of SNR were conducted by training the equalizers with 3000 samples of data. This may seem like an excessive training interval, but it is meant to ensure that the ADSE is near steady state before collecting statistics. Fig. 8 shows the performance of the ADFE (△), ADSE (◯), and nonadaptive Bayesian network (□) on the (a) linear channel, (b) Wiener nonlinear channel, and (c) Hammerstein nonlinear channel described in Section III. The first thing we notice in these results is that there are regions of divergence of both the ADFE

and ADSE, where the performance degrades considerably. There are several important details regarding this divergence.

The most important factor is that stability of an ADSE (or ADFE) in decision-directed mode is very different from stability when it is in training mode. When in training mode, the ADSE is essentially an adaptive FIR filter, the stability of which is analyzed in [7]. Conditioned on the feedback state, choosing the *conditional* update constant as

$$0 < \mu_j < \frac{2}{\text{trace}(\mathbf{R}_j)} \tag{2}$$

where

$$\mathbf{R}_j = E\left\{[\mathbf{x}^T(n)\ 1]^T\,[\mathbf{x}^T(n)\ 1] | \gamma(\hat{\mathbf{u}}(n)) = j\right\}$$

will ensure convergence in training mode. This is only loosely true since many of the conditions on the filter input data that are necessary for the stability analysis, e.g., uncorrelated and Gaussian, are not precisely met. In any case, for the Wiener nonlinearity of Section III with 10 dB SNR, the right side inequality limits of (2) range from 0.14 to 0.22. Thus, given the relatively small range, we have opted to fix μ_j to be simply μ. The results of Fig. 8(b) for 10 dB SNR indicate that even for μ approximately 28 times smaller than these upper limits, the ADSE diverges when in decision-directed mode.

In decision-directed mode, an error will lead to a run of errors. In such a situation, coefficients clearly are not adapting to the desired solution until by random $\hat{\mathbf{u}}(n) = \mathbf{u}(n)$. However, during such an error run interval, the system could adapt to a nonrecoverable model. Choosing μ smaller will shift the regions of divergence in Fig. 8 toward lower SNR. Therefore, there is a tradeoff between tracking ability and decision-directed mode stability. We have chosen a μ with the goal of fastest convergence yet stability in the usable SNR range. Naturally, we have the option of two different choices for μ dependent on whether the system is training or tracking.

The divergence problem is compounded by the situation described in Section III regarding the MMSE strategy. Upon convergence, the ADSE does not attain the optimal hyperplane in the probability sense. Thus, the likelihood of errors is greater than it could be, which means that the likelihood that the system diverges over some given interval is also greater than it could be with an appropriately chosen hyperplane.

As for overall performance for our example, the ADSE matches the ADFE on the linear ISI channel Fig. 8(a). The ADSE shows significant performance improvement over the ADFE for the Wiener and Hammerstein channel nonlinearities Fig. 8(b) and (c), respectively. When comparing results among the channels, performance should be considered relative to the Bayesian performance. For the Hammerstein channel equalizer, its ADSE performance compares with its Bayesian performance roughly the same as the linear ADSE performance compares with the linear Bayesian performance. This suggests that with the ADSE, little performance is lost due to the Hammerstein nonlinearity of this example. This comes as no surprise in light of the discussion in Section III-A. For the Wiener channel equalizer, its ADSE performance compared with its Bayesian performance is not quite as good as the linear ADSE performance compared with the linear Bayesian performance, suggesting that some performance degradation occurs for the ADSE due to the Wiener nonlinearity of this example.

V. Conclusions

A method of equalizing nonlinear ISI that is a simple extension of the ADFE is proposed. The innovation is to select and adapt different linear decision boundary models based on past decisions, as opposed to feeding back the past decisions through a sophisticated nonlinear function. This ADSE solution shows marked improvement over the ADFE on the nonlinear channels we have studied. Its key advantage is simplicity.

There is room for improvement over current implementation of the ADSE because of the nature of the MMSE cost minimization driving its LMS-based algorithms. As such, we suggest that further research focus on an adaptive algorithm that attempts to minimize probability of error.

In light of the problems with the MMSE strategy, the concept of decision-selection still remains a valid one. The Bayesian DFE of [10] is actually the generalization of what we have proposed because it selects different Bayesian detectors based on the state of past decisions. We suggest that their method be called the Bayesian ADSE for this reason. However, adaptively determining the constellation necessary for the Bayesian ADSE likely becomes more difficult with nonlinear channels, large numbers of feedforward inputs, and non-Gaussian noise. These issues were not included in the research of [10], but further study may be warranted. A very recent paper [21] studies a novel piece-wise linear, Boolean mapping-based method for asymptotically realizing the Bayesian ADSE for a linear ISI channel.

Appendix
Gaussian Tail in $\mathbb{R}^N$

Let $\boldsymbol{\mu}$ and $\mathbf{K}$ be the mean vector and covariance matrix of a multivariate Gaussian distribution in $\mathbb{R}^N$. A Gaussian tail in $\mathbb{R}^N$ is that portion of the density function

$$f_{\mathbf{X}}(\mathbf{x}) = \frac{1}{(2\pi)^{n/2}[\det(\mathbf{K})]^{1/2}}\ \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\mathbf{K}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right]$$

on the side of a hyperplane opposite the density center $\boldsymbol{\mu}$, i.e., the space associated with probability less than 0.5. It is a generalization of a tail for the univariate normal distribution. Define the hyperplane as $\{\mathbf{x}\colon \mathbf{w}^T\mathbf{x} + b = 0\}$.

To compute the probability associated with the Gaussian tail, a translation may be done in a fashion similar to the univariate case. First, the mean is removed by the substitution $\mathbf{z} = \mathbf{x} - \boldsymbol{\mu}$. Next, since $\mathbf{K}$ is positive definite, there is a matrix $\mathbf{C}$ for which $\mathbf{K} = \mathbf{C}\mathbf{C}^T$ and $\mathbf{C}^T\mathbf{K}^{-1}\mathbf{C} = \mathbf{I}$, where -1 is matrix inverse, and $\mathbf{I}$ is an identity matrix [22]. The linear transformation $\mathbf{z} = \mathbf{C}\mathbf{y}$ makes $\mathbf{y}$ a zero mean, multivariate normal distribution with covariance matrix $\mathbf{I}$. The original hyperplane transforms to the resulting space as $\{\mathbf{y}\colon \mathbf{w}^T\mathbf{C}\mathbf{y} + \mathbf{w}^T\boldsymbol{\mu} + b = 0\}$.

Because of the nature of the zero mean, multivariate normal distribution with covariance matrix $\mathbf{I}$, the only parameter necessary for computing the probability of error is the minimum distance from the origin to the hyperplane, say, d. This distance is the magnitude of the constant associated with a normalized hyperplane representation, i.e., $d = |\mathbf{w}^T\boldsymbol{\mu} + b|/\sqrt{\mathbf{w}^T\mathbf{K}\mathbf{w}}$.

Without loss of generality, assume $\mathbf{C}$ is a translation that puts the hyperplane parallel to $N-1$ axes. Then

$$P_{\text{tail}} = \int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}\int_{d}^{\infty} \frac{1}{(2\pi)^{n/2}} \exp\left(-\frac{1}{2}\sum_{i=1}^{N} y_i^2\right) \cdot \mathrm{d}y_1\,\mathrm{d}y_2\cdots\mathrm{d}y_N = \mathrm{Q}\left(\frac{|\mathbf{w}^T\boldsymbol{\mu}+b|}{\sqrt{\mathbf{w}^T\mathbf{K}\mathbf{w}}}\right)$$

where

$$\mathrm{Q}(s) = \frac{1}{\sqrt{2\pi}}\int_{s}^{\infty} e^{-t^2/2}\mathrm{d}t.$$

ACKNOWLEDGMENT

The authors would like to thank the reviewers for their thoughtful comments and for pointing out a flaw in the original manuscript regarding the convergence of the ADSE. This paper is much improved as a result of their reviews.

REFERENCES

[1] R. W. Lucky, "Techniques for adaptive equalization of digital communication systems," *Bell Syst. Tech. J.*, vol. 45, no. 2, pp. 255–286, Feb. 1966.
[2] C. W. Niessen and P. R. Drouilhet, Jr., "Adaptive equalizer for pulse transmission," in *IEEE Int. Conf. Commun. Dig. Techn. Papers*, Minneapolis, MN, June 12–14, 1967, p. 117.
[3] J. G. Proakis and J. H. Miller, "An adaptive receiver for digital signaling through channels with intersymbol interference," *IEEE Trans. Inform. Theory*, vol. IT-15, pp. 484–497, July 1969.
[4] M. E. Austin, "Decision feedback equalization for digital communication over dispersive channels," M.I.T./R.L.E., Tech. Rep. 461, Aug. 1967.
[5] D. A. George, R. R. Bowen, and J. R. Storey, "An adaptive decision feedback equalizer," *IEEE Trans. Commun. Technol.*, vol. COM-19, pp. 281–293, June 1971.
[6] J. G. Proakis, "Advances in equalization for intersymbol interference," in *Advances in Communication Systems*, A. J. Viterbi, Ed. New York: Academic, 1975, vol. 4.
[7] S. Haykin, *Adaptive Filter Theory*, 4th ed. Upper Saddle River, NJ: Prentice-Hall, 2001.
[8] E. Biglieri, A. Gersho, R. D. Gitlin, and T. L. Lim, "Adaptive cancellation of nonlinear intersymbol interference for voiceband data transmission," *IEEE J. Select. Areas Commun.*, vol. SAC-2, pp. 765–777, Sept. 1984.
[9] S. Chen, G. J. Gibson, C. F. N. Cowan, and P. M. Grant, "Adaptive equalization of finite nonlinear channels using multilayer perceptrons," *Signal Process.*, vol. 20, no. 2, pp. 107–119, June 1990.
[10] S. Chen, B. Mulgrew, and S. McLaughlin, "Adaptive Bayesian equalizer with decision feedback," *IEEE Trans. Signal Processing*, vol. 41, pp. 2918–2927, Sept. 1993.
[11] A. Uncini, L. Vecci, P. Campolucci, and F. Piazza, "Complex-valued neural networks with adaptive spline activation function for digital radio links nonlinear equalization," *IEEE Trans. Signal Processing*, vol. 47, pp. 505–514, Feb. 1999.
[12] 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.
[13] D. D. Falconer, "Adaptive equalization of channel nonlinearities in QAM data transmission systems," *Bell Syst. Tech. J.*, vol. 57, no. 7, pp. 2589–2611, Sept. 1978.
[14] R. Bernardini,, "A fast algorithm for general Volterra filtering," *IEEE Trans. Commun.*, vol. 48, pp. 1853–1864, Nov. 2000.
[15] S. Marcos, "A network of adaptive Kalman filters for data channel equalization," *IEEE Trans. Signal Processing*, vol. 48, pp. 2620–2627, Sept. 2000.
[16] V. N. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.
[17] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining Knowledge Discovery*, vol. 2, no. 2, pp. 1–47, 1998.
[18] R. Haber and L. Keviczky, *Nonlinear System Identification: Input–Output Modeling Approach*. Boston, MA: Kluwer, 1999, vol. 2.
[19] L. Devroye, L. Györfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*. New York: Springer, 1996.
[20] S. Chen, B. Mulgrew, E. S. Chng, and G. Gibson, "Space translation properties and the minimum-BER linear-combiner DFE," *Proc. Inst. Elect. Eng., Commun.*, vol. 145, no. 5, pp. 316–322, 1998.
[21] 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.
[22] H. Stark and J. W. Woods, *Probability, Random Processes, and Estimation Theory for Engineers*. Englewood Cliffs, NJ: Prentice-Hall, 1986.

Daniel J. Sebald (S'89–M'01) received the B.S. degree from the Milwaukee School of Engineering, Milwaukee, WI, in 1987, the M.S. degree from Marquette University, Milwaukee, in 1992, and the Ph.D. degree from the University of Wisconsin, Madison, in 2000, all in electrical engineering.

He is a registered P.E. in the State of Wisconsin and has worked for Camtronics Medical Systems, Hartland, WI; Nicolet Instrument Technologies, Madison; Xyte, Madison; and OB Scientific, Germantown, WI. His research interests include signal processing, image processing, communications, real-time DSP, and medical technology.

James A. Bucklew received the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, in 1979.

He is currently a Professor with the Department of Electrical and Computer Engineering and the Department of Mathematics, University of Wisconsin, Madison. He is interested in the general area of statistical signal processing and applied probability and has published well over 100 papers in these areas. He is the author of *Large Deviation Techniques in Decision, Simulation, and Estimation*.

Dr. Bucklew has served in the past as Associate Editor for the IEEE TRANSACTIONS ON INFORMATION THEORY and for the IEEE TRANSACTIONS ON SIGNAL PROCESSING.