Bayesian ML Sequence Detection for ISI Channels

Jill K. Nelson
Department of Electrical and Computer Engineering
George Mason University
Fairfax, VA 22030
Email: jnelson@gmu.edu

Andrew C. Singer
Department of Electrical and Computer Engineering
University of Illinois
Urbana, IL 61801
Email: acsinger@uiuc.edu

***Abstract*— We propose a Bayesian technique for blind detection of coded data transmitted over a dispersive channel. The Bayesian maximum likelihood sequence detector views the channel taps as stochastic quantities drawn from a known distribution and computes the probability of any transmitted sequence by averaging over the tap values. The resulting path metric requires memory of all previous symbols, and hence a tree-based algorithm is employed to find the most likely transmitted sequence. Simulation results show that the Bayesian detector can achieve bit error rates within 1/4 dB of the conventional known-channel maximum likelihood (ML) sequence detector.**

I. Introduction

A variety of communication channels suffer from intersymbol interference, or ISI. Interference among transmitted symbols may be introduced by transmit or receive filtering, bandwidth limitations at the transmitter or receiver, or through multi-path propagation. Significant research has been devoted to developing effective equalization methods for dispersive channels. Common techniques range from the bit-error rate (BER) optimal maximum a posteriori probability (MAP) detector to the low-complexity minimum mean-square error (MMSE) linear equalizer.

In many communication systems, the parameters of the dispersive channel may be affected by a variety of physical and environmental conditions that are not known *a priori*. As a result, receivers are often faced with the challenge of detecting data transmitted over channels that are partially, or perhaps entirely, unknown. The behavior of unknown channels may also vary with time, thereby presenting an even more difficult problem to the receiver. One approach to equalization of an unknown channel is to first generate an estimate of the channel, often by transmitting training symbols, and use the channel estimate to detect transmitted information [1]. Such an approach is less desirable for time-varying channels, since the estimate must be updated or regenerated as the channel evolves.

Alternatively, the channel and data may be estimated concurrently to identify the most likely combination of channel and associated data [2]–[5]. Since joint estimation of the channel and data can be prohibitively complex, approximate solutions to this problem are often implemented in practice. Seshadri [4] proposes such a technique based on the Viterbi algorithm. Rather than expanding the trellis on which the Viterbi algorithm is performed, Seshadri's scheme allows an increase in the number of surviving sequences retained in the Viterbi trellis. Because the unknown channel parameters used in the likelihood computation are estimated based on the data along the path in question, Seshadri's algorithm falls in a class of algorithms known as per-survivor processing (PSP) techniques [6].

Researchers have also considered blind detection approaches, i.e., recovering the transmitted data without generating an explicit estimate of the channel. Blind detection schemes, such as the well-known constant modulus (CM) algorithm [7], typically employ estimates of higher order statistics (HOS) of the channel to generate data estimates [8].

We propose an alternative blind detection scheme in which, rather than assuming a deterministic channel response, we view the channel taps as stochastic quantities drawn from a known probability distribution. Using Bayesian techniques, we average over the unknown quantities to compute the probability of a sequence of transmitted symbols. A stack-like tree search algorithm is used to identify the most likely sequence based on the Bayesian probability metric. While Bayesian approaches to blind detection have been proposed in the past, they have been used within different detection structures, such as particle filtering and iterative Viterbi-based schemes [9], [10]. Reader and Cowley considered an approach similar to ours, employing Bayesian techniques to generate sequence likelihoods [11]. Rather than using a tree-based algorithm, however, the authors used the Viterbi algorithm, which requires either a continually expanding trellis or the sacrifice of information in prior data [12].

In the following section, we describe the system model under consideration. The tree-based algorithm used for Bayesian ML sequence detection is described in Section 3. We then derive the Bayesian path metric in Section 4 and explore its empirical performance in Section 5. Conclusions are drawn in Section 6.

II. System Model

We consider a system model, pictured in Figure 1, in which the information bits are encoded using a rate $R = \frac{1}{r}$ error-control code prior to transmission over the channel. We assume the use of a random binary tree code in which the bits on each branch are independent of each other and of those on other branches. In practice, a convolutional code with a large constraint length provides a reasonable approximation. Information bits are transmitted in blocks of length N, denoted

$\mathbf{b}_1^N$, yielding blocks of coded bits of length rN, denoted $\mathbf{x}_1^{rN}$, at the output of the encoder. The encoded bits are transmitted over a length-L ISI channel $h[n]$ with additive white Gaussian noise (AWGN) $w[n]$ of variance $\sigma^2 = N_0/2$. The effect of the channel can be modeled as

$$z_n = \sum_{k=0}^{L-1} h_k x_{n-k} + w_n, \tag{1}$$

where z_n denotes the channel output at time n, and $\mathbf{h} = [h_0 \ \ldots \ h_{L-1}]$ denote the taps of the ISI channel. For simplicity, we consider transmission of only binary phase shift keying (BPSK) encoded data. Transmission of data from larger symbol constellations is a straightforward extension of the work presented here. Each block of received samples $\mathbf{z}_1^{rN}$ serves as input to the detection and decoding block, which uses an algorithm based on the stack decoder to estimate the data sequence $\mathbf{b}_1^N$ by navigating the tree generated by the combined code and channel. We assume that the receiver has knowledge of the error-control code used at the transmitter, the length of the channel, the variance of the AWGN, and the initial state of the encoder.

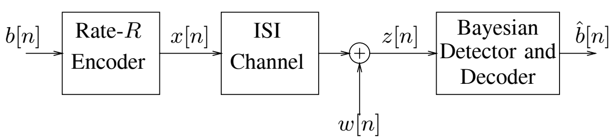

Fig. 1. System model for Bayesian ML sequence detection. BPSK-encoded data is passed through a rate $R = \frac{1}{r}$ error-control encoder, transmitted over a channel with unknown taps $\mathbf{h}$, and processed by a detection and decoding block that employs a stack-like tree search algorithm.

III. ML Sequence Detection via Tree Search

When the channel (or an estimate of the channel) is known, ML sequence detection can be performed efficiently via the Viterbi algorithm, which navigates a trellis and stores only the most likely path to each state [13]. When the channel is unknown, however, the conditional likelihood of any new branch in a path, i.e., $p\left(b_n|\mathbf{b}_1^{n-1}\right)$, is a function of all previous bits in the path. Computation of the most likely sequence, then, must be represented as navigation through a tree rather than through a trellis and typically requires significantly higher computation per bit estimated. We use a sequential algorithm, adopted from the decoding of convolutional codes, to determine an estimate of the most likely sequence.

The stack algorithm, which was originally developed as a method for decoding tree codes and convolutional codes [14], navigates a tree (defined by the code) in search of the path with the largest likelihood, or metric. A set of possible paths and their associated metrics are stored in a list (or stack), and at each iteration, the path with the largest metric is extended.

The stack algorithm terminates when the most likely path in the stack reaches a leaf of the tree, i.e., when a complete path containing N information bits has the largest metric of any path in the stack. Note that the various paths contained in the stack at a particular time are not all of the same length. The algorithm may extend a path of length k in one iteration and in the next iteration find the most likely path to be one of length j for some $j \ll k$. As a result, the path metric must include a bias term to compensate for the differences in likelihood that accompany a difference in path length. As will be discussed, the bias term has a significant impact on the performance of the proposed detector.

The stack-like algorithm employed in the Bayesian sequential detection scheme searches through a tree in a manner similar to that used by the conventional stack decoder. The tree navigated, however, is generated by the combination of the code and the response of the channel. Additionally, the metric used to order the paths is derived from the proposed Bayesian structure and assumes a lack of channel knowledge.

IV. Derivation of the Bayesian Path Metric

To implement a stack-like algorithm we must derive an expression for the probability of a length-n sequence of bits, i.e., we must compute $p(\mathbf{b}_1^n|\mathbf{z}_1^{rN}, C^{(k)})$. Here, $C^{(k)}$ denotes the code known at the current iteration k, i.e., the bits along all branches of the code tree that have been explored by the detector thus far. Since we assume that each block of data bits has been encoded using a random binary tree code, knowledge of the code bits on explored paths give no information about unknown code bits in unexplored sections of the tree and hence simplifies derivation of the Bayesian path metric.

Note that the channel taps $\mathbf{h}$ do not explicitly appear in the probability expression $p(\mathbf{b}_1^n|\mathbf{z}_1^{rN}, C^{(k)})$, as they are viewed as stochastic quantities. We assume a known distribution for the parameters and average over the distribution to generate the probability of any sequence. A general form for the probability of a length-n sequence $\mathbf{b}_1^n$ is derived as

$$\begin{aligned} p(\mathbf{b}_1^n|\mathbf{z}_1^{rN}, C^{(k)}) &= \int_{\mathbf{h}} p(\mathbf{b}_1^n, \mathbf{h}|\mathbf{z}_1^{rN}, C^{(k)}) d\mathbf{h} \\ &= \int_{\mathbf{h}} \frac{p(\mathbf{z}_1^{rN}|\mathbf{b}_1^n, \mathbf{h}, C^{(k)}) p(\mathbf{b}_1^n, \mathbf{h}, C^{(k)})}{p(\mathbf{z}_1^{rN}, C^{(k)})} d\mathbf{h} \\ &= \frac{P(\mathbf{b}_1^n)}{p(\mathbf{z}_1^{rN}|C^{(k)})} \int_{\mathbf{h}} p(\mathbf{z}_1^{rN}|\mathbf{b}_1^n, \mathbf{h}, C^{(k)}) p(\mathbf{h}) d\mathbf{h}, \end{aligned} \tag{2}$$

where the third equality follows from the joint independence of the information bits $\mathbf{b}_1^n$, the channel taps $\mathbf{h}$, and the code known so far $C^{(k)}$. Eliminating $p(\mathbf{z}_1^{rN}|C^{(k)})$ since it is equal for all paths, we can simplify the Bayesian metric, denoted by $m_B(\mathbf{b}_1^n)$, to the form

$$m_B(\mathbf{b}_1^n) = P(\mathbf{b}_1^n) \int_{\mathbf{h}} p(\mathbf{z}_1^{rN}|\mathbf{b}_1^n, \mathbf{h}, C^{(k)}) p(\mathbf{h}) d\mathbf{h}. \tag{3}$$

We would like to find an integrable form for $p(\mathbf{z}_1^{rN}|\mathbf{b}_1^n, \mathbf{h}, C^{(k)}) p(\mathbf{h})$, the integrand in (3). Because the channel is unknown, and because the ISI channel introduces memory in the received data, received samples beyond the path of interest, i.e. $\mathbf{z}_{rn+1}^{rN}$, are dependent upon those before. However, in order to develop a closed-form metric of practical complexity, we assume the two subsets of the received sequence are nearly independent and approximate the overall likelihood by

$$\begin{aligned} p(\mathbf{z}_1^{rN}|\mathbf{b}_1^n,\mathbf{h},C^{(k)}) & \\ \approx \quad & p(\mathbf{z}_1^{rn}|\mathbf{b}_1^n,\mathbf{h},C^{(k)})p(\mathbf{z}_{rn+1}^{rN}|\mathbf{b}_1^n,\mathbf{h},C^{(k)}) \\ = \quad & p(\mathbf{z}_1^{rn}|\mathbf{b}_1^n,\mathbf{h},C^{(k)})p(\mathbf{z}_{rn+1}^{rN}|\mathbf{h}). \end{aligned} \tag{4}$$

A similar approximation is used in the derivation of the path metric for both stack decoding of convolutional codes [15] and stack-based equalization of known ISI channels [16].

The form of $p(\mathbf{z}_1^{rn}|\mathbf{b}_1^n,\mathbf{h},C^{(k)})$ is simply the path metric used for the conventional Viterbi algorithm and is given by

$$\begin{aligned} p(\mathbf{z}_1^{rn}|\mathbf{b}_1^n,\mathbf{h},C^{(k)}) &= \prod_{i=1}^{rn} p(z_i|\mathbf{b}_1^n,\mathbf{h},C^{(k)}) \\ &= \frac{\exp\left\{\frac{-1}{2\sigma^2}(R_{zz}^{rn}[0]+2\mathbf{h}^T\mathbf{r}_{zx}^{rn}-\mathbf{h}^T R_{xx}^{rn}\mathbf{h})\right\}}{(2\pi\sigma^2)^{rn/2}}, \end{aligned} \tag{5}$$

where

$$R_{zz}^k[0] = \sum_{i=1}^k z_i^2, \tag{6}$$

$$\mathbf{r}_{zx}^k = \sum_{i=1}^k z_i \mathbf{x}_{i-L+1}^i, \tag{7}$$

and

$$R_{xx}^k = \sum_{i=1}^k \left(\mathbf{x}_{i-L+1}^i\right)\left(\mathbf{x}_{i-L+1}^i\right)^T. \tag{8}$$

To compute $p(\mathbf{z}_{rn+1}^{rN}|\mathbf{h})$, we average over all possible length-$r(N-n)$ sequences of code bits $\mathbf{x}_{rn+1}^{rN}$, yielding

$$p(\mathbf{z}_{rn+1}^{rN}|\mathbf{h}) = \left(\frac{1}{2}\right)^{r(N-n)} \sum_{\mathbf{x}\in\{-1,1\}^{r(N-n)}} p(\mathbf{z}_{rn+1}^{rN}|\mathbf{h},\mathbf{x}). \tag{9}$$

Such a sum over $2^{r(N-n)}$ terms is impractically complex to include in the path metric. To simplify the expression, we assume independence among the future received samples and approximate the joint likelihood by

$$p(\mathbf{z}_{rn+1}^{rN}|\mathbf{h}) \approx \prod_{i=rn+1}^{rN} p(z_i|\mathbf{h}). \tag{10}$$

Such an approximation can be justified using Massey's approach of appending a random tail to the length-n codeword [17]. Given the channel tap values $\mathbf{h}$ and the L code bits $\mathbf{x}_{i-L+1}^i$, received sample z_i has a Gaussian distribution with mean

$$E[z_i|\mathbf{h},\mathbf{x}_{i-L+1}^i] = \mathbf{h}^T\mathbf{x}_{i-L+1}^i \tag{11}$$

and variance σ^2. Averaging over all length-L binary sequences, we obtain

$$\prod_{i=rn+1}^{rN} p(z_i|\mathbf{h}) = \prod_{i=rn+1}^{rN}\left(\frac{2^{-L}}{\sqrt{2\pi\sigma^2}}\sum_{\mathbf{x}\in\{-1,1\}^L}\exp\left\{-\frac{1}{2\sigma^2}(z_i-\mathbf{h}^T\mathbf{x})^2\right\}\right), \tag{12}$$

which appears in the Bayesian metric for any path segment of length-n information bits. To generate an expression of reasonable complexity for practical implementation, we approximate the Gaussian mixture in (12) by a single Gaussian, i.e.

$$p(z_i|\mathbf{h}) \approx \frac{1}{\sqrt{2\pi\sigma_z^2}}\exp\left\{-\frac{1}{2\sigma_z^2}(z_i-\mu_z)^2\right\}, \tag{13}$$

where

$$\mu_z = E[z_i|\mathbf{h}] = 0, \tag{14}$$

and

$$\sigma_z^2 = E[z_i^2|\mathbf{h}] = \sigma^2 + \mathbf{h}^T\mathbf{h}. \tag{15}$$

We assume the channel energy $\mathcal{E} = \mathbf{h}^T\mathbf{h}$, or an estimate thereof, is known at the receiver.

Substituting the approximations in (4), (10), and (13) into the general Bayesian metric expression in (3) yields the result

$$\begin{aligned} m_B(\mathbf{b}_1^n) &= \left(\frac{1}{2}\right)^n\left(\frac{1}{2\pi\sigma^2}\right)^{rn/2}\left(\frac{1}{2\pi(\sigma^2+\mathcal{E})}\right)^{r(N-n)/2} \\ &\times \int_{\mathbf{h}}\exp\left\{\frac{-1}{2\sigma^2}(R_{zz}^{rn}[0]+2\mathbf{h}^T\mathbf{r}_{zx}^{rn}-\mathbf{h}^T R_{xx}^{rn}\mathbf{h}) -\frac{1}{2(\sigma^2+\mathcal{E})}\sum_{i=rn+1}^{rN} z_i^2\right\}p(\mathbf{h})d\mathbf{h}. \end{aligned} \tag{16}$$

Given the quadratic exponential form of (16), we choose the conjugate Gaussian prior over $\mathbf{h}$ to allow for closed-form integration. To this end, we assume the channel taps are drawn from a pdf of the form

$$f(\mathbf{h}) = \frac{(2\pi)^{-\frac{L}{2}}}{|K_h|^{\frac{1}{2}}}\exp\left\{-\frac{1}{2}\left((\mathbf{h}-\mu_{\mathbf{h}})^{\mathbf{T}}\mathbf{K}_{\mathbf{h}}^{-\mathbf{1}}(\mathbf{h}-\mu_{\mathbf{h}})\right)\right\}, \tag{17}$$

where $\mu_{\mathbf{h}}$ denotes the vector mean of the channel taps, K_h denotes the covariance matrix of the channel taps, and $|K_h|$ denotes the determinant of K_h. Substituting (17) into (16) and integrating yields

$$\begin{aligned} m_B(\mathbf{b}_1^n) &= \left(\frac{1}{2}\right)^n \frac{(\sigma^2+\mathcal{E})^{\frac{r(n-N)}{2}}}{(\sigma^2)^{\frac{rn}{2}}}\left(\frac{|M|}{|K_h|}\right)^{1/2}\times \\ &\exp\left\{-\frac{R_{zz}^{rn}[0]}{2\sigma^2}-\frac{\mu_{\mathbf{h}}^{\mathbf{T}}\mathbf{K}_{\mathbf{h}}^{-\mathbf{1}}\mu_{\mathbf{h}}}{2}-\frac{\sum_{i=rn+1}^{rN} z_i^2}{2(\sigma^2+\mathcal{E})}\right\}\times \\ &\exp\left\{\frac{1}{2}\left(\frac{\mathbf{r}_{zx}^{rn}}{\sigma^2}+K_h^{-1}\mu_{\mathbf{h}}\right)^T M\left(\frac{\mathbf{r}_{zx}^{rn}}{\sigma^2}+K_h^{-1}\mu_{\mathbf{h}}\right)\right\}, \end{aligned} \tag{18}$$

where

$$M = \left(\frac{R_{xx}^{rn}}{\sigma^2}+K_h^{-1}\right)^{-1}. \tag{19}$$

For the simulations and discussion presented here, we consider scenarios in which the channel taps have mean zero, equal variance σ_h^2, and are jointly independent. Under these assumptions, the distribution over the channel taps simplifies to

$$p(\mathbf{h}) = \frac{1}{(2\pi\sigma_h^2)^{L/2}}\exp\left\{-\frac{\mathbf{h}^T\mathbf{h}}{2\sigma_h^2}\right\}. \tag{20}$$

These assumptions on the prior over the channel taps can be interpreted as a "minimum channel knowledge" scenario, since no prior information about (nonzero) tap values nor knowledge of channel tap correlations is used. In cases in which the system designer has some prior knowledge of the channel tap values and/or correlation between the taps, this information could be incorporated into the pdf over $\mathbf{h}$ via a nonzero mean vector and non-diagonal correlation matrix. Using the prior over $\mathbf{h}$ given in (20), the Bayesian path metric for an ISI channel simplifies to

$$m_B(\mathbf{b}_1^n) = \left(\frac{\sigma_h^L}{2^n}\right)\left(\frac{(\sigma^2+\mathcal{E})^{\frac{r(n-N)}{2}}}{\sigma^{rn}}\right)\left|\frac{R_{xx}^{rn}}{\sigma^2}+\frac{I}{\sigma_h^2}\right|^{-1/2}\times \exp\left\{-\frac{R_{zz}^{rn}[0]}{2\sigma^2}+\frac{1}{2\sigma^4}\mathbf{r}_{zx}^{rnT}\left(\frac{R_{xx}^{rn}}{\sigma^2}+\frac{I}{\sigma_h^2}\right)^{-1}\mathbf{r}_{zx}^{rn}\right\}\times \exp\left\{-\frac{1}{2(\sigma^2+\mathcal{E})}\sum_{i=rn+1}^{rN} z_i^2\right\}. \tag{21}$$

The simplified path metric takes a quadratic exponential form. Note that, while the metric is a function of all received samples, future samples (those associated with information bits beyond the path segment of interest) are normalized by the larger variance $\sigma^2+\mathcal{E}$, indicating that less information about these samples is available since the associated transmitted bits are unknown. As simulation results will show, the approximation yielding this bias term underestimates the likelihood of future symbols, thereby favoring longer paths.

Perhaps the most interesting term in the simplified metric is the $\mathbf{r}_{zx}^{rnT}\left(\frac{R_{xx}^{rn}}{\sigma^2}+\frac{I}{\sigma_h^2}\right)^{-1}\mathbf{r}_{zx}^{rn}$ term in the exponential. This term takes a form very similar to least squares estimation of the channel taps and is the element of the metric in which we see implicit learning of the channel. The $\mathbf{r}_{zx}^{rnT}\left(\frac{R_{xx}^{rn}}{\sigma^2}+\frac{I}{\sigma_h^2}\right)^{-1}\mathbf{r}_{zx}^{rn}$ term is also the element of the metric that prohibits implementation of the algorithm via trellis search. Because the algorithm does not generate explicit channel estimates but instead uses information about each path and the received data to generate a metric, the number of states increases by the size of the symbol alphabet (a factor of 2 for BPSK) for each increase in path length. (If the channel were known, the number of states would remain constant at 2^{L-1} after L stages.) The stack-like algorithm avoids an exhaustive search of the exponentially increasing number of states by extending only the path that appears most likely rather than every possible path.

V. Simulated Performance of the Bayesian Detector

To evaluate the performance of the proposed Bayesian ML detection scheme, we simulate the detector for a three-tap channel with impulse response

$$h[n] = 0.407\delta[n+1] + 0.815\delta[n] + 0.407\delta[n-1]. \tag{22}$$

The performance of the Bayesian detector is simulated for block lengths of $N = 10, 100$, and 1000 data bits. For comparison, we also simulate the performance of conventional ML sequence detection (via the Viterbi algorithm) over the channel $h[n]$, using blocks of length 1000. For simulation of both the conventional ML detector and the Bayesian detection scheme, the information bits are encoded using a rate $R = 1/2$ ($r = 2$) convolutional code with generator matrix

$$G(x) = [x+1 \quad x^2+x+1]. \tag{23}$$

Figure 2 shows the BER of the Bayesian and standard ML detectors over a range of SNR values. The results reveal that, for SNR between 4 and 9 dB, the Bayesian ML detection scheme with $N = 1000$ achieves performance within approximately 1/2 dB of that of the conventional known-channel ML detector. It is also clear from Figure 2 that the BER performance of Bayesian ML sequence detection improves significantly as block size increases. This behavior can be explained by noting that the Bayesian algorithm is using the received data, as well as the assumed transmitted data along each path, to learn the values of the channel taps. Larger block lengths allow the detector more information from which to learn the response of the channel. This behavior is akin to that of least-squares estimation, which generates improved parameter estimates as the amount of data available increases. Note, however, that the performance for $N = 100$ is nearly equivalent to that for $N = 1000$, indicating that the Bayesian detector can achieve low BER with relatively small block length.

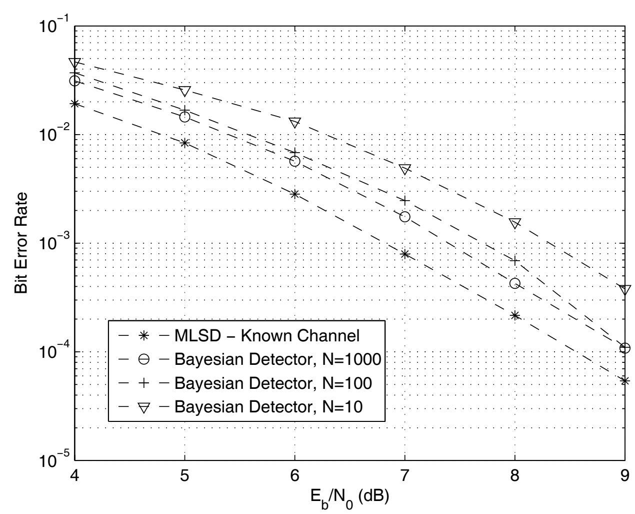

Fig. 2. Simulated performance of Bayesian ML sequence detector and conventional (Viterbi) ML sequence detector. The channel response is $h[n] = 0.407\delta[n+1] + 0.815\delta[n] + 0.407\delta[n-1]$, and the convolutional code has generator matrix $G(x) = [x+1 \quad x^2+x+1]$.

Additional simulations have been conducted to explore the performance of the Bayesian sequential detector when the channel is essentially known. To generate such a scenario, we let $\mu_{\mathbf{h}} = \mathbf{h}$ and $\sigma_h^2 \ll 1$. Results reveal that, even when the channel is known, the Bayesian detector suffers a performance loss of nearly 1/2 dB with respect to the Viterbi algorithm, indicating that the loss is likely due to a suboptimal bias term rather than a lack of channel knowledge. To address this problem, we consider an alternative path metric derivation that avoids the assumption of independence among future received elements and the Gaussian approximation of $p(z_i|\mathbf{h})$.

Rather than computing the path likelihood conditioned on the entire received sequence, we consider the probability of the current path based only on the received sequence thus far, i.e.,

$$
\begin{aligned}
&p(\mathbf{b}_1^n|\mathbf{z}_1^{rn}, C^{(k)}) \qquad (24)\\
&\quad = \frac{P(\mathbf{b}_1^n)}{p(\mathbf{z}_1^{rn}|C^{(k)})}\int_{\mathbf{h}} p(\mathbf{z}_1^{rn}|\mathbf{b}_1^n, \mathbf{h}, C^{(k)})p(\mathbf{h})d\mathbf{h}\\
&\quad = \frac{(1/2)^n \sigma_h^L \left|\frac{R_{xx}^{rn}}{\sigma^2} + \frac{I}{\sigma_h^2}\right|^{-1/2}}{p(\mathbf{z}_1^{rn}|C^{(k)})(2\pi\sigma^2)^{rn/2}} \times\\
&\quad \exp\left\{-\frac{R_{zz}^{rn}[0]}{2\sigma^2} + \frac{1}{2\sigma^4}\mathbf{r}_{zx}^{rnT}\left(\frac{R_{xx}^{rn}}{\sigma^2} + \frac{I}{\sigma_h^2}\right)^{-1}\mathbf{r}_{zx}^{rn}\right\},
\end{aligned}
$$

where the prior distribution $p(\mathbf{h})$ is given by (20). The quantity $p(\mathbf{z}_1^{rn}|C^{(k)})$ can be computed using an external stack which, rather than extending only one path at each stage, extends all paths in the stack. We can then iteratively generate the conditional likelihoods for all n according to

$$
\begin{aligned}
&p(\mathbf{z}_1^{rn}|C^{(k)}) \qquad (25)\\
&\quad = (1/2)^n \sum_{\mathbf{x}_1^{rn}\in C^{(k)}} \int_{\mathbf{h}} p(\mathbf{z}_1^{rn}|\mathbf{x}_1^{rn}, \mathbf{h})p(\mathbf{h})d\mathbf{h}.
\end{aligned}
$$

Note that the integral in (25) takes the same form as that in (24). Akin to the work of Kuhn and Hagenauer [18], we can limit the size of the external stack to 2^S entries ($S \ll N$) to maintain a resonable complexity level. At the end of each iteration, only the 2^{S-1} largest stack entries are retained.

Simulation results using the alternative metric given in (24) are presented in Figure 3. In addition to the conventional ML and Bayesian detectors, we also consider the performance of the PSP technique presented in [4], which navigates a trellis rather than a tree and uses LMS to update a channel estimate for each surviving path. For all detectors, information blocks of length $N = 500$ were encoded using a convolutional code with generator matrix $G(x) = [1 \quad 1 + x + x^2]$. The external stack for the Bayesian detector was limited to 2^6 entries, and one path to each state was retained in the LMS-Viterbi scheme.

At low SNR, all three detectors show similar bit error rates. As SNR increases, however, the Viterbi-based scheme reaches an error floor, while the Bayesian detector closely tracks the performance of the conventional ML detector. The error floor results from path eliminations made early in the data block when the LMS channel estimate has not converged. Because the Bayesian algorithm navigates a tree rather than a trellis, it is able to postpone eliminating path segments until it has more channel information, thereby significantly improving bit error rate for the early bits in each block.

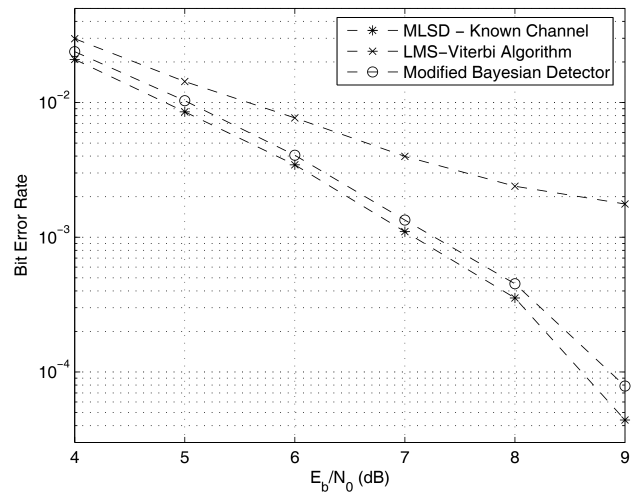

Fig. 3. Simulated performance of the conventional ML sequence detector, the Bayesian detector with modified metric, and a Viterbi-based PSP algorithm employing LMS. The channel has impulse response $h[n] = 0.407\delta[n+1] + 0.815\delta[n] + 0.407\delta[n-1]$, and the convolutional code has generator matrix $G(x) = [1 \quad 1 + x + x^2]$.

VI. Conclusion

We have proposed a novel Bayesian technique for detecting data transmitted over an unknown ISI channel. To allow implicit learning of the channel as the detector progresses, the most likely transmitted sequence is identified via a tree search algorithm. Without requiring training data, the Bayesian detector can achieve BER within 1/4 dB of ML sequence detection for a known channel. These promising results warrant further study of the Bayesian detector, perhaps characterizing its computational complexity (a Pareto random variable for standard stack decoders) and exploring extensions to equalization of time-varying channels.

References

[1] S. Haykin, *Adaptive Filter Theory*. Upper Saddle River, NJ: Prentice-Hall, 1996.

[2] R. Amara and S. Marcos, "A blind network of extended Kalman filters for nonstationary channel equalization," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, May 2001, pp. 2117–2120.

[3] L. M. Davis, I. B. Collings, and P. Hoeher, "Joint MAP equalization and channel estimation for frequency-selective and frequency-flat fast-fading channels," *IEEE Transactions on Communications*, vol. 49, no. 12, pp. 2106–2114, December 2001.

[4] N. Seshadri, "Joint data and channel estimation using blind trellis search techniques," *IEEE Transactions on Communications*, vol. 42, no. 2-4, pp. 1000–1011, February-April 1994.

[5] R. Iltis, J. Shynk, and K. Giridhar, "Bayesian algorithms for blind equalization using parallel adaptive filtering," *IEEE Transactions on Communications*, vol. 42, no. 2-4, pp. 1017–1032, February-April 1994.

[6] R. Raheli, A. Polydoros, and T. Ching-Kae, "Per-survivor processing: A general approach to MLSE in uncertain environments," *IEEE Transactions on Communications*, vol. 43, pp. 354–364, February-April 1995.

[7] C. R. Johnson, P. Schniter, T. J. Endres, J. D. Behm, D. R. Brown, and R. A. Casas, "Blind equalization using the constant modulus criterion: A review," *Proceedings of the IEEE*, vol. 86, no. 10, pp. 1927–1950, October 1998.

[8] Z. Ding and Y. Li, *Blind Equalization and Identification*. New York, NY: Marcel Dekker, 2001.

[9] J. Miguez and P. M. Djuric, "Blind equalization by sequential importance sampling," in *Proc. IEEE Int. Symposium on Circuits and Systems*, May 2002, pp. I–845–I–848.
[10] X. Wang and R. Chen, "Blind turbo equalization in Gaussian and impulsive noise," *IEEE Transactions on Vehicular Technology*, vol. 50, no. 4, pp. 1092–1105, July 2001.
[11] D. Reader and W. Cowley, "Blind maximum likelihood sequence detection," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, 1996, pp. 2694–2697.
[12] D. Reader, "Blind maximum likelihood sequence detection over fast fading communication channels," Ph.D. dissertation, University of South Australia, August 1996.
[13] G. D. Forney, "Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference," *IEEE Transactions on Information Theory*, vol. 18, pp. 363–378, May 1972.
[14] R. E. Blahut, *Algebraic Codes for Data Transmission*. New York, NY: Cambridge University Press, 2003.
[15] A. J. Viterbi and J. K. Omura, *Principles of Digital Communication and Coding*. New York, NY: McGraw-Hill, 1979.
[16] F. Xiong, A. Zerik, and E. Shwedyk, "Sequential sequence estimation for channels with intersymbol interference of finite or infinite length," *IEEE Transactions on Communications*, vol. 38, no. 6, pp. 795–804, June 1990.
[17] J. L. Massey, "Variable-length codes and the fano metric," *IEEE Transactions on Information Theory*, vol. 18, no. 1, pp. 196–198, January 1972.
[18] C. Kuhn and J. Hagenauer, "8-PSK turbo equalization with the list-sequential (LISS) algorithm," in *International Symposium on Information Theory*, June 2004, p. 555.