Prefiltered Space–Time M-BCJR Equalizer for Frequency-Selective Channels

Christina Fragouli, *Member, IEEE*, Naofal Al-Dhahir, *Senior Member, IEEE*, Suhas N. Diggavi, *Member, IEEE*, and William Turin, *Senior Member, IEEE*

***Abstract*—This paper addresses the problem of soft equalization for space–time-coded transmissions over frequency-selective fading channels. The structure of the space–time code is embedded in the channel impulse response for efficient joint equalization and decoding. The proposed equalization/decoding approach uses a prefilter to concentrate the effective channel power in a small number of taps followed by a reduced-complexity maximum *a posteriori* probability (MAP) equalizer/decoder to produce soft decisions. The prefilter introduces residual intersymbol interference which degrades the performance of MAP when applied to the trellis of the shortened channel. However, the shape of the overall shortened channel impulse response allows the M-algorithm to approximate the prefiltered MAP performance with a small number of states. Based on this general framework, we investigate several enhancements such as using different prefilters for the forward and backward recursions, concatenating two trellis steps during decoding, and temporal oversampling. The performance is evaluated through simulations over the EDGE typical urban channel.**

***Index Terms*—BCJR algorithm, equalization, FIR prefilters, space–time coding.**

I. Introduction

THIS paper considers the problem of soft equalization for space–time-coded signals over frequency-selective channels such as mobile radio channels. The receiver has to compensate for intersymbol interference (ISI) by using some form of equalization. Moreover, soft equalization is desirable because it allows graceful combination with powerful coding schemes.

Decision feedback equalization (DFE) is a popular, low-complexity equalization method but is inferior in performance to trellis-based equalization due to error propagation and its use of symbol-by-symbol hard decisions. Combining the DFE with coding for wireless channels is also challenging. These considerations motivated us to focus on trellis-based equalization techniques.

A frequency-selective channel with L taps can be modeled as a discrete-time finite-state machine, whose state X_n at time n is defined by the $L-1$ most recent channel inputs $X_n = \{s(n-L+2)\cdots s(n)\} = s_{n-L+2}^{n}$,[1] where $s(n)$ is the channel input at time n. Thus, the channel can be described by a trellis with C^{L-1} states, where C is the constellation size. The trellis allows us to perform sequence equalization through maximum likelihood sequence estimation (MLSE), or symbol by symbol maximum *a posteriori* probability (MAP) estimation. Trellis equalization provides improved performance as compared to linear or decision feedback equalization but its complexity grows exponentially with the number of symbols that are affected by the ISI and the transmission spectral efficiency (in b/s/Hz). Therefore, it is important to develop suboptimal algorithms that require less computing resources without significant degradation in performance.

Several papers such as [1]–[3] proposed reduced-state sequence equalization methods. The estimator uses a Viterbi algorithm with decision feedback to search a reduced-state trellis. The number of states is reduced by using hard estimates either for some state components $s(i)$ (i.e., reduce the number of states by reducing L in C^{L-1}) or for the group where the state components belong (i.e., reduce C for some components). In the first category, the most prevalent method is delayed-decision-feedback sequence estimation (DDFSE) [1] while in the second category belong schemes like those in [2] and [3]. The sequence equalization methods produce hard decisions at the equalizer output and again may suffer from error propagation effects.

MAP equalizers can produce soft decisions on the transmitted symbols. The soft decision is achieved by computing a transmitted symbol *a posteriori* probability (APP) on the basis of the received sequence. For an ISI channel with independent noise, the APP can be computed using the forward–backward algorithm. Suboptimal reduced-state MAP equalization algorithms, such as the M-algorithm and the T-algorithm [4], keep only the sequences of trellis states that contribute most to the APP calculation. The M-algorithm uses a subset of M states while the T-algorithm uses a variable number of states at different instants. This paper employs the variation of the M-algorithm proposed in [5], which is based on the implementation of the forward–backward algorithm in the log-domain and can offer improved performance.

Another popular approach reduces the channel impulse response to a shorter length before equalization [6]–[12]. The shortening methods vary from direct truncation of the channel memory to approaching a target impulse response by minimizing an error function. Typically a linear prefilter shortens

Paper approved by W. E. Ryan, the Editor for Modulation, Coding and Equalization of the IEEE Communications Society. Manuscript received November 20, 2000; revised June 12, 2001, and November 2, 2001. This paper was presented in part at the Conference on Information Sciences and Systems (CISS), Baltimore, MD, March 2001, and at the IEEE Communication Theory Workshop, San Diego, CA, May 2001.

C. Fragouli is with the National Capodistrian University of Athens, 173-43 Athens, Greece.

N. Al-Dhahir, S. N. Diggavi, and W. Turin are with AT&T Shannon Laboratory, Florham Park, NJ 07932 USA (e-mail: naofal@research.att.com).

Publisher Item Identifier S 0090-6778(02)05100-0.

[1]For the rest of this paper we denote sequences of finite length with the starting/ending time-index as the subscript/superscript, respectively.

0090-6778/02$17.00 © 2002 IEEE

the channel to a smaller number of taps and then the shortened channel is equalized by either MLSE or MAP. The shortening prefilter introduces residual ISI and has to avoid possible noise enhancement. The residual ISI becomes more pronounced as the length of the target channel, which determines the number of states of the trellis equalizer, decreases.

Considering the number of equalizer trellis states as a measure for receiver complexity, [11] proposes the combination of prefiltering and reduced-state sequence estimation to achieve a target complexity with the smallest degradation in performance possible. More specifically, a prefilter concentrates the channel power in a smaller number of taps than that of the original channel, thus reducing the total number of trellis states, but without shortening to a degree that causes severe residual ISI. The prefilter is followed by DDFSE which further reduces the number of trellis states to the desired level. This paper follows the general method in [11] but is distinct in the following key aspects:

1) A symbol-by-symbol reduced-state MAP estimator is used instead of DDFSE, which is less susceptible to error propagation and provides soft decisions at the equalizer output.
2) The prefilter design is optimized not only toward concentrating the channel power in a smaller number of taps without noise enhancement, but also toward shaping the target channel power-delay profile to optimize the performance of reduced-state decoding. In other words, the prefilter optimizes the target channel power-delay profile for the specific reduced trellis equalization.
3) Different enhancements of the basic structure (shortening prefilter followed by a reduced-state trellis equalizer) are investigated. These include using different prefilters for the forward and backward recursions, oversampling for finer tuning of the prefilter, and concatenating two trellis steps together during decoding.

The performance of the proposed receiver structure is evaluated for enhanced data for GSM evolution (EDGE). To increase the data rate, EDGE employs 8-PSK with a linearized GMSK pulse-shaping filter which introduces severe ISI. This makes the number of states needed for optimal trellis equalization prohibitive and thus motivates the application of reduced complexity techniques for this application. Implementing a space–time trellis code with two transmit antennas at the base station and one receive antenna at the mobile offers increased diversity and coding gains on the downlink which is the bottleneck in asymmetric applications such as web-browsing and downloading. Our focus in this paper is on space–time trellis codes. Equalization schemes for space–time block codes are discussed in [13].

The rest of this paper is organized as follows. Section II reviews the EDGE channel model. Although this model is presented in specific numerical detail for precise presentation of the simulations environment, the investigated methods are applicable to any frequency-selective quasi-static fading channel. Section III deals with the trellis equalization algorithms. Section IV is devoted to the prefilter design. Section V presents the simulation results and the paper is concluded in Section VI.

II. Channel Model

To achieve high information rates, EDGE [14] employs 8-PSK modulation with a linearized GMSK pulse over a 200-kHz time-division multiple access (TDMA) channel. For EDGE, the ideal frequency pulse is truncated from $(-2T_s, 2T_s)$, where T_s is the symbol duration and made causal. The truncated pulse introduces severe intersymbol interference (above -50 dB) of four consecutive symbols. The achieved symbol rate is 271 kb/s, or equivalently the symbol duration T_s is 3.69 μs. The physical channel we consider follows the typical urban (TU) power-delay profile [14], where every tap follows an independent complex Gaussian distribution.

The overall channel output $y(n)$ at time n is described as

$$y(n) = \sum_{i=0}^{L-1} h(i)s(n-i) + z(n) \tag{1}$$

where $s(n)$ is the input sequence, $z(n)$ is additive Gaussian noise, L is the channel length ($L = 4$ for the EDGE TU channel considered), and $h(n)$ are the overall channel taps that include the transmit filter and the physical channel. For the rest of the paper, we denote vectors with boldface, i.e., $\mathbf{h} = [h(0)\cdots h(L-1)]$. We further assume the following channel power normalization condition:

$$E\mathbf{h}\mathbf{h}^* = \sum_i E(h(i)h^*(i)) = 1 \tag{2}$$

where $\mathbf{h}^*$ is the conjugate transpose of $\mathbf{h}$. For the purpose of simulations, we assume that the noise is white. In practice, the presence of co-channel or adjacent channel interference results in correlated noise. To combat the wireless channel fading, space–time trellis codes jointly optimize the function of modulation, coding, and spatial diversity transmission to provide significant diversity and coding gains. Use of multiple antennas at the base-station transmitter provides diversity gains[2] for the downlink while placing most of the complexity burden at the base station. We consider two transmit and one receive antenna transmission scenarios, employing the 8-state 8-PSK space–time trellis code in [16] with the trellis description as shown in Fig. 1. In this figure, the output of the two transmit antennas are denoted by $s_1(n)$ and $s_2(n)$, respectively. We denote by $\mathbf{h_1} = [h_1(0)\cdots h_1(L-1)]$ and $\mathbf{h_2} = [h_2(0)\cdots h_2(L-1)]$ the channels from the first and second transmit antenna to the receiver antenna, respectively. For the space–time code that we consider, the received output at time n can be expressed as

$$\begin{aligned} y(n) &= \sum_{i=0}^{L-1} h_1(i)s(n-i) + \sum_{i=0}^{L-1} h_2(i)[s(n-i-1)]^{\tilde{5}} + z(n) \\ &= \sum_{i=0}^{L} h_{eq}(i)s(n-i) + z(n) \end{aligned} \tag{3}$$

[2]While we consider space–time trellis codes in this paper, the proposed methods are applicable to other multiple-antenna systems.

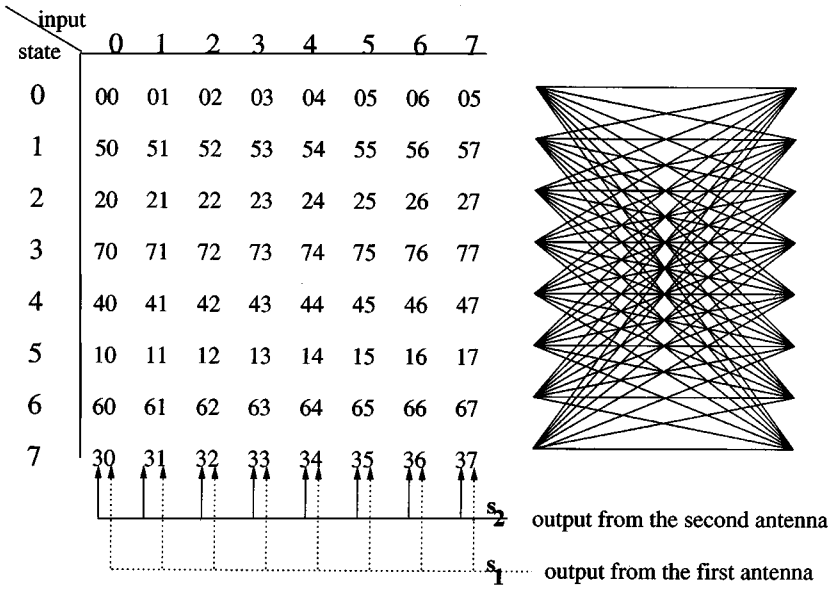


Fig. 1. Trellis and encoder matrix representation of the 8-state 8-PSK space–time trellis code proposed in [15].

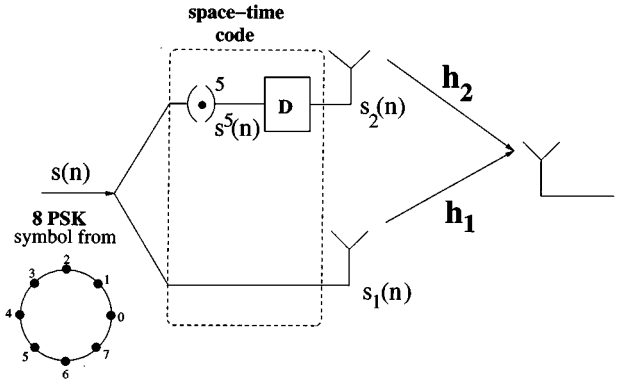


Fig. 2. Nonlinear implementation of the 8-state 8-PSK space–time trellis code proposed in [15].

where

$$h_{eq}(i) = \begin{cases} h_1(0), & \text{for } i = 0 \\ [s(n-L)]^4 h_2(L-1), & \text{for } i = L \\ h_1(i) + [s(n-i)]^4 h_2(i-1), & \text{for } 1 \le i \le L-1. \end{cases} \quad (4)$$

Note that, since $s(i)$ belongs to the 8-PSK constellation, $[s(i)]^4 = \pm 1$. It follows from (4) that we can incorporate the space–time code structure in the channel model to arrive at an equivalent single-input single-output channel which we denote by $\mathbf{h_{eq}} = [h_{eq}(0) \cdots h_{eq}(L)]$. This allows for joint channel equalization and decoding of the space–time code using a single trellis description based on $\mathbf{h_{eq}}$. The selected code allows a nonlinear implementation with one memory element, as Fig. 2 illustrates, thus increases the channel memory only by one. Note that this technique is *not* limited to the 8-state 8-PSK space–time trellis code and, in fact, every code described by a trellis can be incorporated into the channel trellis, allowing joint equalization and decoding. Exploiting the structure of the space–time trellis code in this manner requires in general C^{m+L} states, where m is the memory of the space–time trellis code and L the ISI channel length.

Finally, we make the following two assumptions.

1) The channels $\mathbf{h}_1$ and $\mathbf{h}_2$ remain constant over an EDGE block and vary independently from block to block. This assumption is based on the fact that each EDGE block has a time duration of 0.577 ms which is much smaller than the coherence time of most wireless channels (for example, for 60-mi/h mobile velocity, the coherence time is approximately 12.3 ms at a carrier frequency of 900 MHz [17]).
2) The receiver has perfect channel knowledge. In practice, the 26 training symbols in each EDGE block are used to estimate the channel taps.

Note that the overall channel $\mathbf{h}_{eq}$ seen by the trellis equalizer/decoder is not constant over a block, since its tap values depend on the input symbols. More specifically, $\mathbf{h}_{eq}$ can take 2^{L-1} possible values for each constant $\mathbf{h}_1$ and $\mathbf{h}_2$. These values occur in a time succession that can be modeled by a hidden Markov model (HMM). Fig. 3 shows, for 10 000 channel realizations, the average maximum, minimum, and mean power for the five channel taps. The power variation is significant for the second and third taps.

Given $\mathbf{h}_1$ and $\mathbf{h}_2$, a trellis equalizer/decoder can easily calculate the $\mathbf{h}_{eq}$ taps from (4) for a given input sequence, and thus the corresponding decoding metrics. For the rest of this paper, we denote the overall equivalent channel by $\mathbf{h}$ to simplify notation.

III. Soft-Output Trellis Decoding Algorithms

The(MAP symbol decoder maximizes the APP of symbol $s(n)$ after receiving a sequence $y_1^N = \{y(1) \cdots y(N)\}$, which is proportional to the so-called unnormalized APP $\Pr(s(n), y_1^N)$. To calculate this probability, we note that, according to (3), $y(n)$

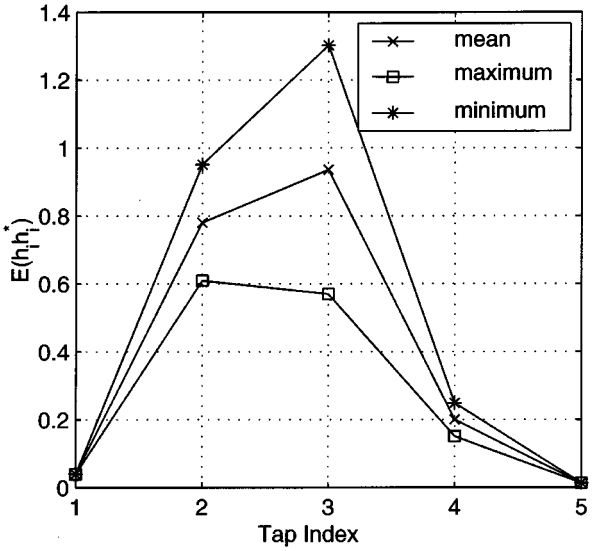


Fig. 3. Power distribution for the five $\mathbf{h}_{eq}$ channel taps calculated through averaging over 10 000 realizations of the EDGE TU channel.

represents an HMM with states $X_n = s_{n-L+1}^n$ and state-transition observation probability densities

$$b_{ij}(y(n)) = \Pr(y(n)|X_{n-1} = i,\, X_n = j) \propto \exp(\|y(n) - \sum_{i=0}^{L} h_{eq}(i)s(n-i)\|^2/2\sigma^2) \quad (5)$$

where $\|.\|$ denotes the Euclidean norm and σ^2 is the noise variance. Using these probabilities and the Markov chain transitional probabilities $p_{ij} = \Pr(X_n = j|X_{n-1} = i)$, we can define [18] an input–output state-transition probability matrix $\mathbf{P}(s(n),\, y(n))$ whose $(i,\, j)$ element is the conditional probability $\Pr(X_n = j,\, y(n)|X_{n-1} = i) = p_{ij}b_{ij}(y(n))$ of transferring from trellis state i at time $n-1$ to trellis state j at time n with input $s(n)$ and observed output $y(n)$. Using this matrix, we can express the unnormalized APP in the following matrix form [4], [18]:

$$\Pr(s(n),\, y_1^N) = \boldsymbol{\alpha}(y_1^{n-1})\mathbf{P}(s(n),\, y(n))\boldsymbol{\beta}(y_{n+1}^N) \quad (6)$$

where the forward probability vector $\boldsymbol{\alpha}(y_1^{n-1})$ at time $n-1$ depends on the observed sequence y_1^{n-1} and is defined as

$$\boldsymbol{\alpha}(y_1^{n-1}) = \boldsymbol{\pi}\mathbf{P}(y(1))\mathbf{P}(y(2))\cdots\mathbf{P}(y(n-1)). \quad (7)$$

The row vector $\boldsymbol{\pi}$ consists of the state initial probabilities. The ith element of this vector is the probability $\Pr(X_{n-1} = i,\, y_1^{n-1})$. The backward probability vector $\boldsymbol{\beta}(y_{n+1}^N)$ at time $n+1$ depends on the observed sequence y_{n+1}^N and is defined as

$$\boldsymbol{\beta}(y_{n+1}^N) = \mathbf{P}(y(n+1))\mathbf{P}(y(n+2))\cdots\mathbf{P}(y(N))\mathbf{1}. \quad (8)$$

The ith element of this vector is the probability $\Pr(y_k^N|X_k = i)$. In (7) and (8), we used the notation $\mathbf{P}(y(n)) = \sum_s \mathbf{P}(s(n) = s,\, y(n))$, and $\mathbf{1}$ is a column vector with all elements equal to one. It follows from (7) and (8) that the forward and backward probability vectors can be computed recursively as

$$\boldsymbol{\alpha}(y(1)) = \boldsymbol{\pi}\mathbf{P}(y(1)) \quad \boldsymbol{\alpha}(y_1^n) = \boldsymbol{\alpha}(y_1^{n-1})\mathbf{P}(y(n)) \quad (9)$$

$$\boldsymbol{\beta}(y(N)) = \mathbf{P}(y(N))\mathbf{1} \quad \boldsymbol{\beta}(y_n^N) = \mathbf{P}(y(n))\boldsymbol{\beta}(y_{n+1}^N). \quad (10)$$

The previous equations for computing the unnormalized APP represent the so-called forward–backward algorithm. For complexity and numerical precision reasons, these equations are usually implemented in the logarithmic domain [19].

We would like to point out that the forward–backward algorithm can be applied not only to the systems with the i.i.d. sources and additive white Gaussian noise (AWGN) considered in this paper, but also to more general communication systems in which information sources, coding, and channels have memory and can be described by HMMs [18], because in all these cases the APP can be computed by equations that have the form of (6).

A. M-Algorithm

The M-BCJR algorithm [4] proposes the following reduced-state trellis decoding technique: at each trellis step, keep only the M active states associated with the highest metrics and discard the rest. In other words, the forward vectors $\boldsymbol{\alpha}(y_1^{n-1})$ and backward vectors $\boldsymbol{\beta}(y_{n+1}^N)$, as calculated by (9) and (10), retain the M largest value components (normalized so that their sum is equal to one), and assign zero probability to the rest and, therefore, do not propagate paths from them.

Denote as $\mathcal{A}_n^M$ and $\mathcal{B}_n^M$ the set of M nodes selected at time n by the forward and backward recursions, respectively. Typically, the forward recursion chooses the tree of active nodes $\mathcal{A} = \bigcup_n \mathcal{A}_n^M$, while the backward recursion may only follow the selected active nodes: $\mathcal{B}_n^M \subseteq \mathcal{A}_n^M$ [4]. Decisions are taken using the edges in the common subtree.

This approach of combining the forward and backward vectors to form soft decisions is biased toward the correct selection of active nodes by the forward recursion. This bias leads to significant performance degradation [5] if the channel has low power in the first taps, as is the case for many channels, including TU EDGE.

We can explain this degradation by examining when the "correct" state sequence $X_{\text{corr}} = \{X_0^{N-1}\}$, corresponding to the sequence of transmitted symbols, is included in the active tree $\mathcal{A}$. Assume that the state that corresponds to the correct transmitted data at time $n-1$ is included in the set of active states

$$X_{n-1} \in \mathcal{A}_{n-1}^M, \qquad X_{n-1} \subset X_{\text{corr}}.$$

The metric associated with each of the next states of the form $X_n = (s(n-L+2)\cdots s(n-1)u)$, for each possible input u, depends on the distance

$$\|y(n) - (h(0)u + \sum_{i=1}^{L-1} h(i)s(n-i))\| = \|h(0)(s(n)-u) + z(n)\|.$$

It follows from this equation that the smaller is the energy of the first tap $h(0)$, the higher is the probability of not including the correct path in the active tree $\mathcal{A}$, which leads to performance degradation. Similarly, the energy in the last tap $h(L-1)$ plays

an important role in the error probability of the backward recursion.

Another problem that the M-algorithm might have is that the forward recursion may allow less than M possible paths for the backward recursion to follow at each trellis step. Thus, the backward recursion cannot fully take advantage of its allocated complexity of M states. Moreover, there might not be enough active edges to calculate soft metrics. All these effects become more pronounced as M decreases.

B. Modified M-Algorithm

The forward–backward algorithm is commonly implemented in the logarithmic domain to avoid multiplications of very small numbers that can cause machine precision problems, and to reduce complexity. To remove the bias caused by the selection of the active nodes by only one of the recursions, [5] takes advantage of the implementation of the MAP algorithm in the log-domain.

We allow the forward and backward recursions independently select trees of active nodes $\mathcal{A} = \bigcup \mathcal{A}^{\mathcal{M}}$ and $\mathcal{B} = \bigcup \mathcal{B}^{\mathcal{M}}$ without restricting one to be a subtree of the other. To form soft decisions at time n, we use all edges with at least one active node. As a metric for the nonactive nodes, we use instead of minus infinity (which corresponds to probability zero in the log domain), negative numbers I_A and I_B that express the reliability of selecting M nodes by the forward or the backward recursion, respectively. The proposed choice of values are [5]

$$I_A = \log\left(Q\left(\sqrt{|h(0)|^2 d_C^2/2\sigma^2}\right)\right) \tag{11}$$

$$I_B = \log\left(Q\left(\sqrt{|h(L-1)|^2 d_C^2/2\sigma^2}\right)\right) \tag{12}$$

where d_C is the minimum Euclidean distance between any two constellation points.

The nonlinearity introduced by deciding on the sets $\mathcal{A}_n^M$ and $\mathcal{B}_n^M$ has similarities with the DFE nonlinear hard decisions. Indeed, the M-algorithm with $M_A = 1$ and $M_B = 0$ reduces to a DFE structure with no feedforward filter.

In the minimum mean square error (MMSE) DFE framework, making decisions with a delay $\Delta > 0$ can result in significant performance improvement [20]. Similarly, during MAP decoding for a channel with L taps at time instance n, we can take soft decisions about any $s(n-\Delta)$ with $\Delta = 0 \cdots L-1$ by summing over $s_{n-L+1}^{n-\Delta-1}$ and $s_{n-\Delta+1}^{n}$ the probabilities of trellis edges connecting nodes $X_{n-1} = s_{n-L+1}^{n-1}$ and $X_n = s_{n-L+2}^{n}$. The optimal delay Δ depends on the reliability of the α and β metrics and the distribution of the channel's power among its taps. Forming decisions with delay Δ does not affect the performance of the forward–backward algorithm (MAP decoding) since it amounts to just a different implementation, but may significantly benefit a reduced-state algorithm [5].

C. Concatenating Two Trellis Steps

As discussed in the previous subsection, the reliability of the forward (backward) recursion depends upon the power of the first (last) tap. As Fig. 3 shows, the first and last taps of the EDGE TU channel have significantly less average power than the middle taps. A simple approach to alleviate this problem is to replace two of the decoder trellis steps

$$X_{n-2} \overset{s(n-1)}{\rightarrow} X_{n-1} \overset{s(n)}{\rightarrow} X_n$$

with one step $X_{n-2} \overset{s(n-1)s(n)}{\rightarrow} X_n$ and two input symbols.

Since each trellis step advances by $s(n-1)$ and $s(n)$ simultaneously, the decision metric of the forward (backward) recursion accumulates the power of the first (last) two taps. However, the noise variance is doubled because two outputs are observed. Such an arrangement benefits the EDGE TU channel where the average tap power is greater for the second tap. Again this different implementation of the forward–backward algorithm does not affect its performance, but improves the performance of the reduced-state decoder for channels whose middle taps are stronger than end taps. Moreover, it reduces the memory requirements of the forward–backward algorithm since a block is equalized with half the trellis length and the number of trellis states per trellis step remains the same.

IV. Prefilter

The performance of the reduced-state decoder depends on the number M of active states as compared to the total number of trellis states C^L. As $M \rightarrow C^L$, the bit error rate (BER) curve of the M-algorithm approaches the BER curve of MAP. The rate of convergence depends upon the power-delay profile of the channel.

A shortening prefilter may be used to concentrate the channel power in a smaller number of taps $L_s < L$, thus reducing the total number of trellis states to C^{L_s}. Shortening is a noninvertible operation that leads to some loss of information, so that decoding using all the states of the shortened trellis suffers from a performance loss. The performance loss can be evaluated in terms of the residual ISI after shortening and the noise power at the prefilter output.

Assume that the decoder can support a specific complexity of M trellis states. M-state decoding may provide better performance on the shortened C^{L_s}-state trellis than when applied to the initial C^L-state trellis, especially if the prefilter can also shape the power-delay profile of the channel to optimize reduced-state decoding.

Following the approach in [11], we propose the use of a prefilter to concentrate and shape the channel, up to a point where a small performance loss is incurred, and then apply reduced-state trellis equalization.

A. Prefilter Design

We describe the prefilter design starting from a constant channel and extend it following to our time-varying channel case.

Denote by $\mathbf{h}$ the impulse response of a channel of length L and by $\mathbf{w}$ the impulse response of a prefilter of length N_f. The impulse response of the effective channel $\mathbf{h}_{\text{eff}} = \mathbf{w} * \mathbf{h}$ has length $N_f + L - 1$ and can be expressed in matrix form as

$$\mathbf{h}_{eff} = \mathbf{H}\mathbf{w} \tag{13}$$

where $\mathbf{H}$ is the channel convolution matrix with the $(i+m, i)$th element equal to $h(m)$ for $m = 0, 1, \ldots, (L-1)$ and the rest of its elements are zeros. Let $\mathbf{h}_{\text{win}}$ represent the N_b+1 taps of $\mathbf{h}_{\text{eff}}$ to retain after shortening the channel, and $\mathbf{h}_{\text{wall}}$ the (N_f+L-N_b-2) remaining taps. Then

$$\mathbf{h}_{\text{win}} = \mathbf{J}_{\text{win}}\mathbf{h}_{\text{eff}} = \underbrace{\mathbf{J}_{\text{win}}\mathbf{H}}_{\mathbf{H}_{\text{win}}}\mathbf{w} = \mathbf{H}_{\text{win}}\mathbf{w} \tag{14}$$

where the matrix $\mathbf{J}_{\text{win}}$ is constructed using columns of the identity matrix that correspond to the indexes of the tap positions of $\mathbf{h}_{\text{win}}$ within $\mathbf{h}_{\text{eff}}$. Similarly,

$$\mathbf{h}_{\text{wall}} = \mathbf{J}_{\text{wall}}\mathbf{h}_{eff} = \underbrace{\mathbf{J}_{\text{wall}}\mathbf{H}}_{\mathbf{H}_{\text{wall}}}\mathbf{w} = \mathbf{H}_{\text{wall}}\mathbf{w} \tag{15}$$

where the matrix $\mathbf{J}_{\text{wall}}$ is constructed from the columns of the identity matrix corresponding to tap positions of $\mathbf{h}_{\text{wall}}$ within $\mathbf{h}_{\text{eff}}$.

The proposed prefilter design criterion maximizes the shortening signal-to-noise ratio (SSNR), which is defined as the signal power of the shortened channel divided by the residual ISI plus the noise power at the prefilter output. The corresponding optimization problem can be formulated as the following generalized eigenvector problem [9], [8], [11]:

$$\begin{aligned} &\max_{\mathbf{w}} \mathbf{w}^*\mathbf{B}\mathbf{w} \\ &\text{subject to } \mathbf{w}^*\mathbf{A}\mathbf{w} = 1 \end{aligned} \tag{16}$$

where $\mathbf{B} = \mathbf{H}^*_{\text{win}}\mathbf{H}_{\text{win}}$, $\mathbf{A} = \mathbf{H}^*_{\text{wall}}\mathbf{H}_{\text{wall}} + \mathbf{R}_{nn}$, and $\mathbf{R}_{nn}$ is the noise covariance matrix at the prefilter input. The optimal solution $\mathbf{w}_{\text{opt}}$ can be calculated [21] as

$$\mathbf{w}_{\text{opt}} = (\mathbf{L_A}^*)^{-1}\mathbf{u}_{\max} \tag{17}$$

where $\mathbf{A} = \mathbf{L_A}\mathbf{L_A}^*$ is the Cholesky factorization of matrix $\mathbf{A}$, and $\mathbf{u}_{\max}$ is the normalized eigenvector of matrix $(\mathbf{L_A})^{-1}\mathbf{B}(\mathbf{L_A})^{-1}$ that corresponds to its largest eigenvalue $\lambda_{\max}$. The resulting optimal SSNR is

$$\text{SSNR}_{\text{opt}} = 10\log_{10}\left(\frac{\mathbf{w}^*_{\text{opt}}\mathbf{B}\mathbf{w}_{\text{opt}}}{\mathbf{w}^*_{\text{opt}}\mathbf{A}\mathbf{w}_{\text{opt}}}\right) = 10\log_{10}(\lambda_{\max}). \tag{18}$$

Equation (17) provides the optimal prefilter for a time-invariant channel. As we pointed out before, the overall channel at the input of the prefilter, because it incorporates the space–time code, is data-dependent. In this case, we can design the prefilter to optimize the average power of the equivalent channel [11]. Using (4), the corresponding matrices $\mathbf{A}$ and $\mathbf{B}$ can be expressed as

$$\begin{aligned} \mathbf{B} &= E(\mathbf{H}^*_{\text{win}}\mathbf{H}_{\text{win}}) = (\mathbf{H}^1_{\text{win}})^*\mathbf{H}^1_{\text{win}} + (\mathbf{H}^2_{\text{win}})^*\mathbf{H}^2_{\text{win}} \\ \mathbf{A} &= E(\mathbf{H}^*_{\text{wall}}\mathbf{H}_{\text{wall}}) + \mathbf{R}_{nn} \\ &= (\mathbf{H}^1_{\text{wall}})^*\mathbf{H}^1_{\text{wall}} + (\mathbf{H}^2_{\text{wall}})^*\mathbf{H}^2_{\text{wall}} + \mathbf{R}_{nn} \end{aligned} \tag{19}$$

where $\mathbf{H}^i_{\text{win}}$ and $\mathbf{H}^i_{\text{wall}}$, $i = 1, 2$ are matrices corresponding to the constant channels $\mathbf{h}_1$ and $\mathbf{h}_2$ between the two transmit and the one receive antenna (see [11] for a more detailed derivation).

B. Forward–Backward Prefilter

Define the weighting matrix $\mathbf{M} = \text{diag}\{m_0, m_1, \ldots, m_{N_b}\}$, with $m_i \geq 0$ and consider the optimization problem

$$\begin{aligned} &\max_{\mathbf{w}} \mathbf{w}^*\mathbf{H}^*_{\text{win}}\mathbf{M}\mathbf{H}_{\text{win}}\mathbf{w} \\ &\text{subject to } \mathbf{w}^*\mathbf{H}^*_{\text{wall}}\mathbf{H}_{\text{wall}}\mathbf{w} + \mathbf{w}^*\mathbf{R}_{nn}\mathbf{w} = 1. \end{aligned} \tag{20}$$

This formulation scales the power in each of the N_b+1 taps of the shortened channel with a different coefficient m_i as follows:

$$\mathbf{w}^*\mathbf{H}^*_{\text{win}}\mathbf{M}\mathbf{H}_{\text{win}}\mathbf{w} = \mathbf{h}^*_{\text{win}}\mathbf{M}\mathbf{h}_{\text{win}} = \sum_{i=0}^{N_b} m_i h^*_{\text{win}}(i)h_{\text{win}}(i)$$

where $\mathbf{h}_{\text{win}} = \mathbf{H}_{\text{win}}\mathbf{w}$. Assigning a large weight to a tap favors the concentration of power to it. Thus, by appropriately selecting the weights, we may shape the power-delay profile of the shortened channel. Equation (16) is a special case of (20) for $\mathbf{M}$ equal to the identity matrix.

In the case a time-varying channel, weighting can be applied as follows:

$$\mathbf{A} = E(\mathbf{H}^*\mathbf{M}\mathbf{H}) = (\mathbf{H}^1_{\text{win}})^*\mathbf{M}_1\mathbf{H}^1_{\text{win}} + (\mathbf{H}^2_{\text{win}})^*\mathbf{M}_2\mathbf{H}^2_{\text{win}}$$

where $\mathbf{M} = \text{diag}\{[m_0, m_1, \ldots, m_{N_b}, m_{N_b+1}]\}$ is of dimension $(N_b+2)\times(N_b+2)$, and $\mathbf{M}_1 = \text{diag}\{[m_0, m_1, \ldots, m_{N_b}]\}$, $\mathbf{M}_2 = \text{diag}\{[m_1, \ldots, m_{N_b+1}]\}$ are of dimension $(N_b+1)\times(N_b+1)$. The solution $\mathbf{w}_{M\text{opt}}$ of the weighted optimization problem in (20) is calculated similarly to (17) but with $\mathbf{B} = \mathbf{H}^*_{\text{win}}\mathbf{M}\mathbf{H}_{\text{win}}$. The observed SSNR though is still given by

$$\begin{aligned} \text{SSNR}_{M\text{opt}} &= 10\log_{10}\left(\frac{\mathbf{w}^*_{M\text{opt}}\mathbf{H}^*_{win}\mathbf{H}_{\text{win}}\mathbf{w}_{M\text{opt}}}{\mathbf{w}^*_{M\text{opt}}\mathbf{w}_{M\text{opt}}}\right) \\ &\leq 10\log_{10}\left(\frac{\mathbf{w}^*_{\text{opt}}\mathbf{H}^*_{\text{win}}\mathbf{H}_{\text{win}}\mathbf{w}_{\text{opt}}}{\mathbf{w}^*_{\text{opt}}\mathbf{w}_{\text{opt}}}\right) = \text{SSNR}_{\text{opt}} \end{aligned}$$

where $\mathbf{w}_{\text{opt}}$ is the solution of (16). Imposing constraints on the target channel power-delay profile leads to a suboptimal SSNR. This loss can be compensated for if the shape of the power-delay profile allows the subsequent reduced-state trellis equalizer to achieve better performance with a smaller number of active states.

Fig. 4 provides an example of using different prefilters to shorten the EDGE TU channel from five to four taps. The average, maximum average, and minimum average power-delay profiles are provided. Case (4*a*) used the weight matrix $\mathbf{M_B} = \text{diag}\{[1\ 0\ 10]\}$ and target taps window $[6\ 7\ 8]$ while case (4*b*) used the weight matrix $\mathbf{M_A} = \text{diag}\{[10\ 0\ 1]\}$ and target taps window $[5\ 6\ 8]$.

Equations (11) and (12) imply that the target power-delay profile that favors the forward recursion has concentrated power in the first channel taps while it has concentrated power in the last taps for the backward recursion. Following the approach in [5], we propose to use a different prefilter for the forward and the backward recursions. Both prefilters shorten the channel to the same number of taps but create a different power-delay profile.

Let $y_f(n)$ and $y_b(n)$ be the forward and backward prefilter outputs at time n. We can use $y_f(n)$ for computing the forward probability vectors $\alpha((y_f)_1^{n-1})$ in (7), and $y_b(n)$ for computing

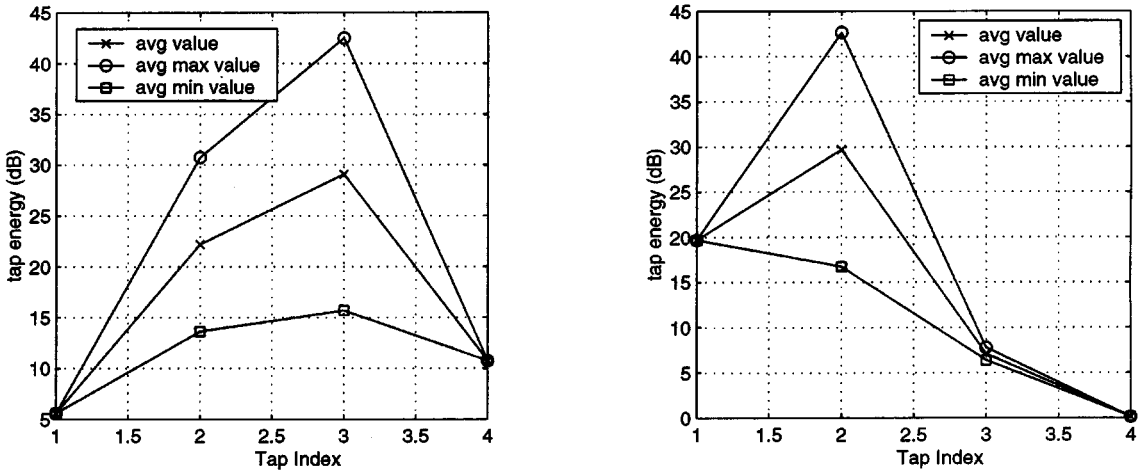


Fig. 4. Average power-delay profile for different prefilter choices for shortening the EDGE TU channel at SNR $= 20$ dB. (a) $\overline{M}_{\overline{B}}$. (b) $\overline{M}_{\overline{A}}$.

the backward probability vectors $\beta((y_b)_{n+1}^N)$ in (8). To form the soft decisions, we can use either $y_f(n)$ or $y_b(n)$ as follows:

$$\Pr(s(n),\, y_1^N) = \boldsymbol{\alpha}((y_f)_1^{n-1})\mathbf{P}(s(n),\, y_{\{f \text{ or } b\}}(n))\boldsymbol{\beta}((y_b)_{n+1}^N). \quad (21)$$

That is, we calculate

$$\Pr(s(n),\, y_f(1)\ldots y_f(n-1),\, y_{\{f \text{ or } b\}}(n),\, y_b(n+1)\cdots y_b(N)). \quad (22)$$

In [5], an all-pass filter was used to concentrate power at the first (last) channel taps before the forward (backward) recursion. The all-pass filter was IIR and ideally converted a channel to its minimum (maximum) phase without loss of information. The proposed approach uses FIR filters, which incur loss of information, but do not have stability problems. Moreover, FIR filters are amenable to adaptive implementations, and may more drastically concentrate power at the taps of interest.

C. Prefilter Design Choices

The prefilter design involves the following design parameters:

1) the number of prefilter taps N_f;
2) the length of the target channel $N_b + 1$;
3) the positions of the $N_b + 1$ taps included in the target channel;
4) the weighting matrix $\mathbf{M}$.

The choice of these parameters depends on the acceptable complexity level at the receiver, and the specific channel under consideration. The optimal values can be determined through exhaustive simulations. A computationally less demanding option is to relate some metrics to the desired prefilter performance and optimize parameters 1)–4) accordingly.

The loss due to residual ISI after shortening is reflected in the difference between SNR at the prefilter output and SSNR as follows:

$$\text{SNR} - \text{SSNR} = 10\log_{10}\left(\frac{E(\mathbf{h}_{\text{eff}}^*\mathbf{h}_{\text{eff}})}{E(\mathbf{w}^*\mathbf{R}_{nn}\mathbf{w})}\right) - 10\log_{10}\left(\frac{E(\mathbf{h}_{\text{win}}^*\mathbf{h}_{\text{win}})}{E(\mathbf{h}_{\text{wall}}^*\mathbf{h}_{\text{wall}} + \mathbf{w}^*\mathbf{R}_{nn}\mathbf{w})}\right) \quad (23)$$

where $\mathbf{h}_{\text{eff}} = \mathbf{h} * \mathbf{w}$ is the effective channel at the prefilter output.

The reliability I_A and I_B of the forward and backward recursions of the M-algorithm depends on the quantities

$$d_A = \frac{E(h_{\text{win}}^*(0)h_{\text{win}}(0))}{E(\mathbf{h}_{\text{wall}}^*\mathbf{h}_{\text{wall}} + \mathbf{w}^*\mathbf{R}_{nn}\mathbf{w})}$$

$$d_B = \frac{E(h_{\text{win}}^*(N_b)h_{\text{win}}(N_b))}{E(\mathbf{h}_{\text{wall}}^*\mathbf{h}_{\text{wall}} + \mathbf{w}^*\mathbf{R}_{nn}\mathbf{w})} \quad (24)$$

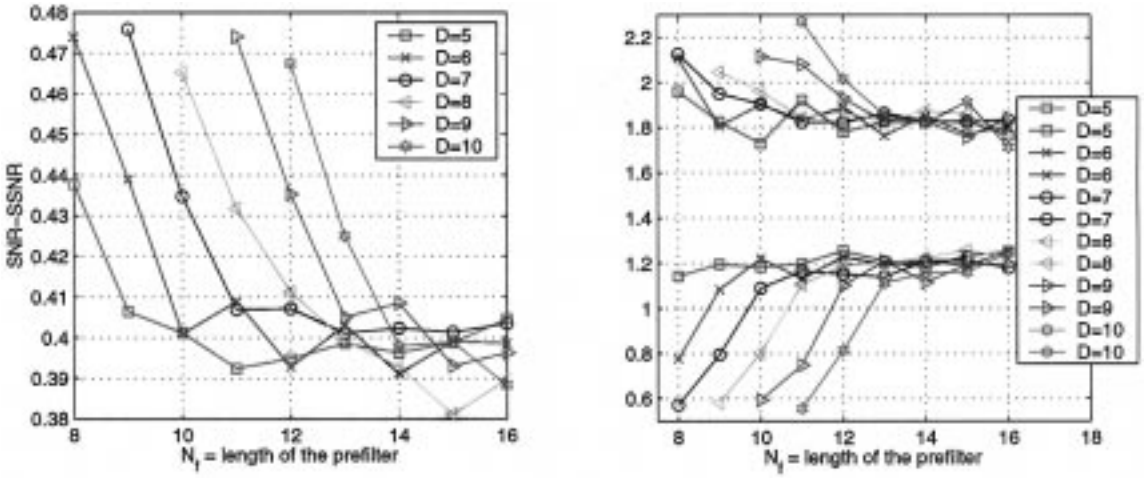
where $h_{\text{win}}(0)$ and $h_{\text{win}}(N_b)$ are the first and the last taps of the shortened channel, respectively.

Metrics (23) and (24) provide a measure of the prefilter effectiveness, and may be used to select parameter values. For example, Fig. 5 plots metrics (23) and (24) in (a) and (b), respectively, for different target tap positions of the form $[D\ D+1\ D+2]$ (described by the value of D), as a function of the prefilter length N_f. The plots represent average values after shortening 10 000 EDGE TU-channel realizations. Fig. 5 indicates that using a prefilter of length $N_f = 8$ with $D = 5$ should lead to the same performance as using a prefilter of length $N_f = 12$ with $D = 7$ (indeed this equivalence was verified through simulation).

D. Oversampling

This section investigates sampling of the receiver output at a rate greater than the baud rate T_s and downsampling after the prefilter.

Fig. 5 indicates that increasing the prefilter length to span more than a certain number of symbols (this number depends on the channel memory and the target response) does not offer further performance improvement. In [12], it was demonstrated that a prefilter operating with oversampling by a factor of Q, $Q > 1$, under some conditions on the prefilter length and the rank of a channel matrix, and assuming sufficient excess bandwidth, can always perfectly suppress the residual ISI, which is not true in the T_s case. The analysis in [12] shows that a prefilter operating at a higher sampling rate has more degrees of freedom available and thus can do a better job of concentrating the channel power in the taps of interest. The



(a) (b)

Fig. 5. Metrics (a) (23) and (b) (24) for different target tap positions of the form $[D\ D+1\ D+2]$ (described by D).

same arguments apply to the power-delay profile shaping of the target channel, oversampling may allow the prefilter to achieve a better fit to the desired target channel.

For simplicity, consider oversampling with a rate $T_s/2$. The prefilter taps are $T_s/2$-spaced; thus, to span the same number of symbols as when they are T_s-spaced, the number of prefilter taps is doubled. The trellis equalizer still operates on the T_s time scale. Let $y(t)$ be the continuous-time received output at time t, i.e., $y(t) = \sum_{n=0}^{L-1} s(n)h(t - nT_s) + z(t)$ where $h(t)$ is the impulse response of the channel. In the EDGE scenario, $h(t)$ may represent the overall impulse response between the two transmit and one receive antennas, incorporating the space–time code in the channel model, as was described in Section II.

The oversampled channel output, assuming no timing error, can be described as

$$\begin{bmatrix} y(nT_s) \\ y\left(nT_s + \dfrac{T_s}{2}\right) \end{bmatrix} = \begin{bmatrix} h(0) & h(T_s) & \cdots & h((L-1)T_s) \\ h\left(\dfrac{T_s}{2}\right) & h\left(\dfrac{3T_s}{2}\right) & \cdots & h\left(\dfrac{3(L-1)T_s}{2}\right) \end{bmatrix} \cdot \begin{bmatrix} s(n) \\ s(n-1) \\ \vdots \\ s(n-L+1) \end{bmatrix} + \begin{bmatrix} z(nT_s) \\ z\left(nT_s + \dfrac{T_s}{2}\right) \end{bmatrix}. \quad (25)$$

The convolution of the channel and the prefilter results in an equivalent $T_s/2$ channel related to the T_s-spaced channel input vector with an equation similar to (25). Downsampling before the trellis equalizer, i.e., keeping either $y(nT_s)$ or $y(nT_s + (T_s/2))$ results in observing half the taps of the equivalent channel, either $[h(0)\ h(T_s)\ \cdots\ h((L-1)T_s)]$ or $[h(T_s/2)\ h(3T_s/2)\ \cdots\ h(3(L-1)T_s/2)]$.

Thus, we can solve the optimization problem (20) using a $T_s/2$-spaced prefilter to achieve a target impulse response such that after downsampling, the observed channel taps maximize SSNR. One possible formulation is to have a weighting matrix $\mathbf{M}$ with zero weights corresponding to the taps that will not be observed after downsampling, for example, of the form: $\mathbf{M} =$

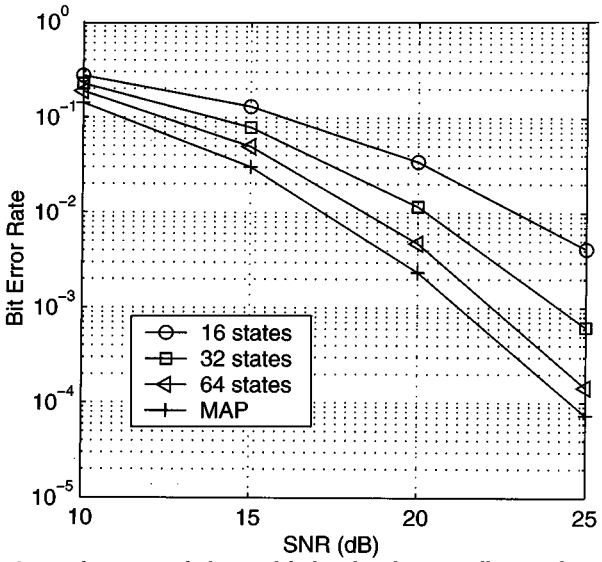


Fig. 6. Performance of the modified reduced-state trellis equalizer (as described in Section III-B) with $M = 16$, 32, and 64 states, over the EDGE TU channel.

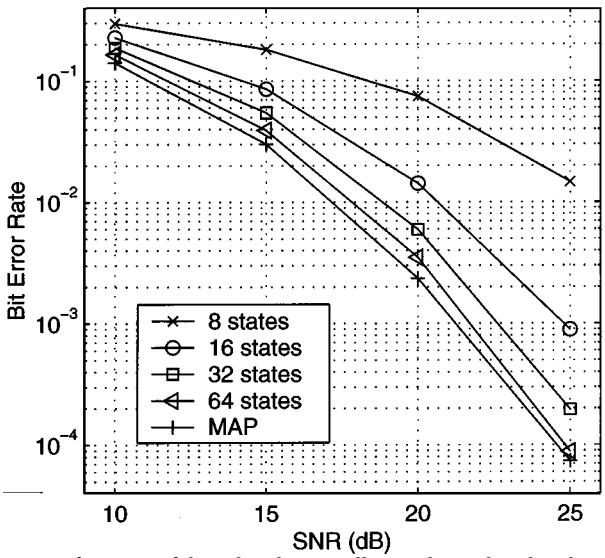


Fig. 7. Performance of the reduced-state trellis equalizer when decoding two trellis steps together with $M = 8$, 16, 32 and 64 states over the EDGE TU channel.

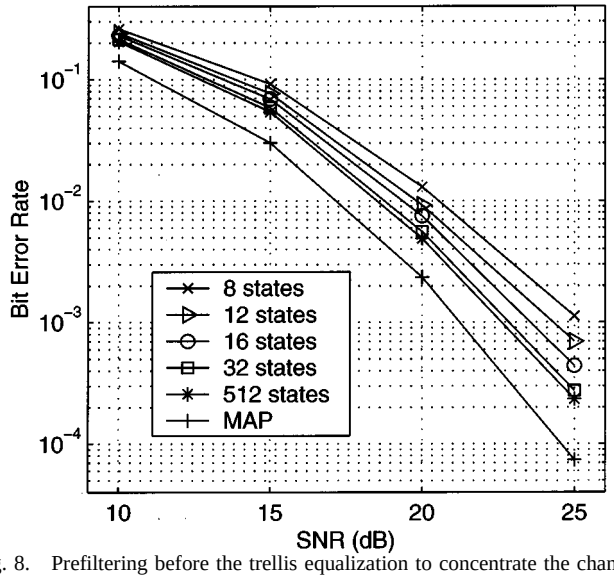


Fig. 8. Prefiltering before the trellis equalization to concentrate the channel power in fewer taps.

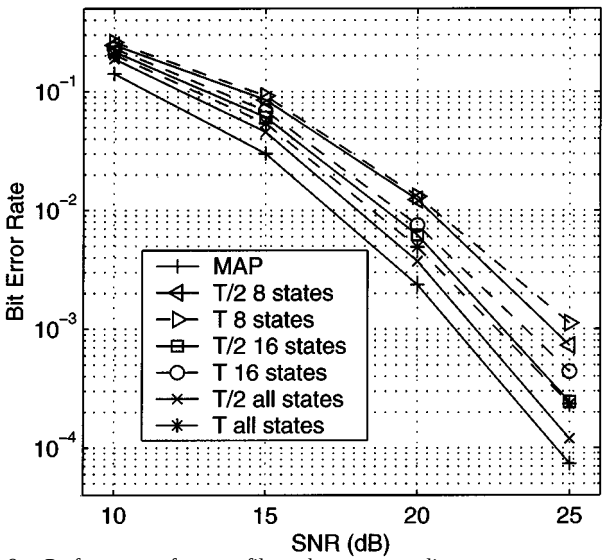


Fig. 9. Performance of one prefilter when oversampling.

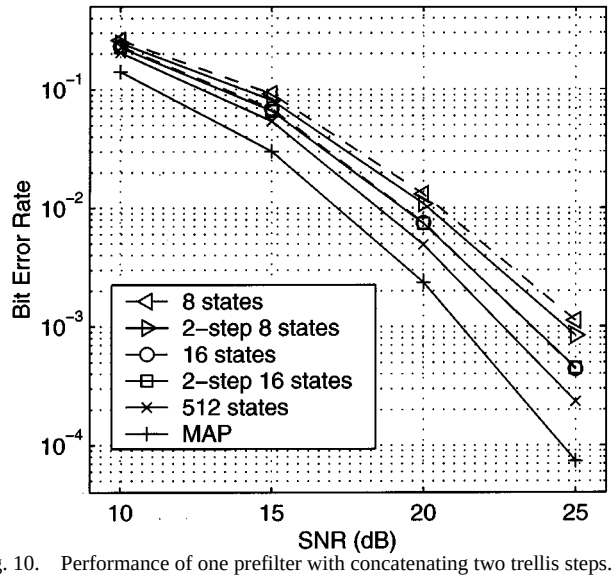


Fig. 10. Performance of one prefilter with concatenating two trellis steps.

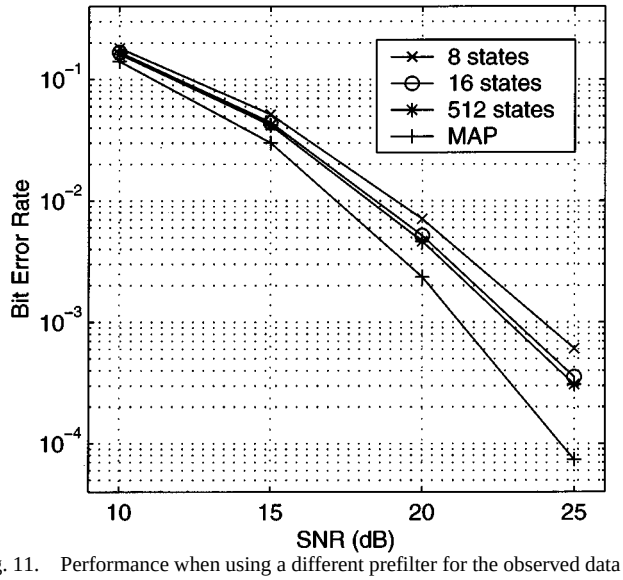


Fig. 11. Performance when using a different prefilter for the observed data of the forward and backward recursion of the trellis equalizer.

$\mathrm{diag}\{[m_0, 0, m_1, 0\cdots]\}$. The zero weights allow the prefilter to arbitrarily allocate power to the corresponding taps, and thus have more freedom to achieve better the target response in the taps of interest.

V. Simulation Results

This section provides simulation results for the EDGE TU channel. We assume two transmit and one receive antennas and the 8-state 8-PSK space–time trellis code that was described in Section II. Each EDGE frame consists of 114 information symbols. Each simulation was run until at least 100 frames were in error.

A. M-Algorithm

Fig. 6 compares the performance of the forward–backward algorithm and the modified M-algorithm described in Section III-B, over the EDGE TU channel. The forward–backward algorithm (denoted by "MAP" in Fig. 6) involves $8^4 = 4096$ trellis states. The reduced-state equalizer performance is within 1 and 3 dB from that of MAP equalization with 64 states and 16 states, respectively.

B. Two Concatenated Steps for M-Algorithm

Fig. 7 shows the performance of the reduced-state trellis equalization algorithm over the EDGE TU channel, when two trellis steps are decoded together, as described in Section III-C. The performance of MAP equalization is also plotted as a point of reference for our simulations. Concatenating two steps together allows us to achieve the same performance with half the number of active trellis states. The performance difference is within 0.5, 1, and 3 dB with 64, 32, and 16 states, respectively.

C. Prefilter

Fig. 8 plots the performance when a shortening prefilter precedes the trellis equalizer, as described in Section IV-A. The prefilter has $N_f = 8$ taps and was designed using (17). The target channel response length is $N_b + 1 = 3$ taps for each of the $\mathbf{h}_1$ and $\mathbf{h}_2$ transmit channels, or equivalently four taps for the

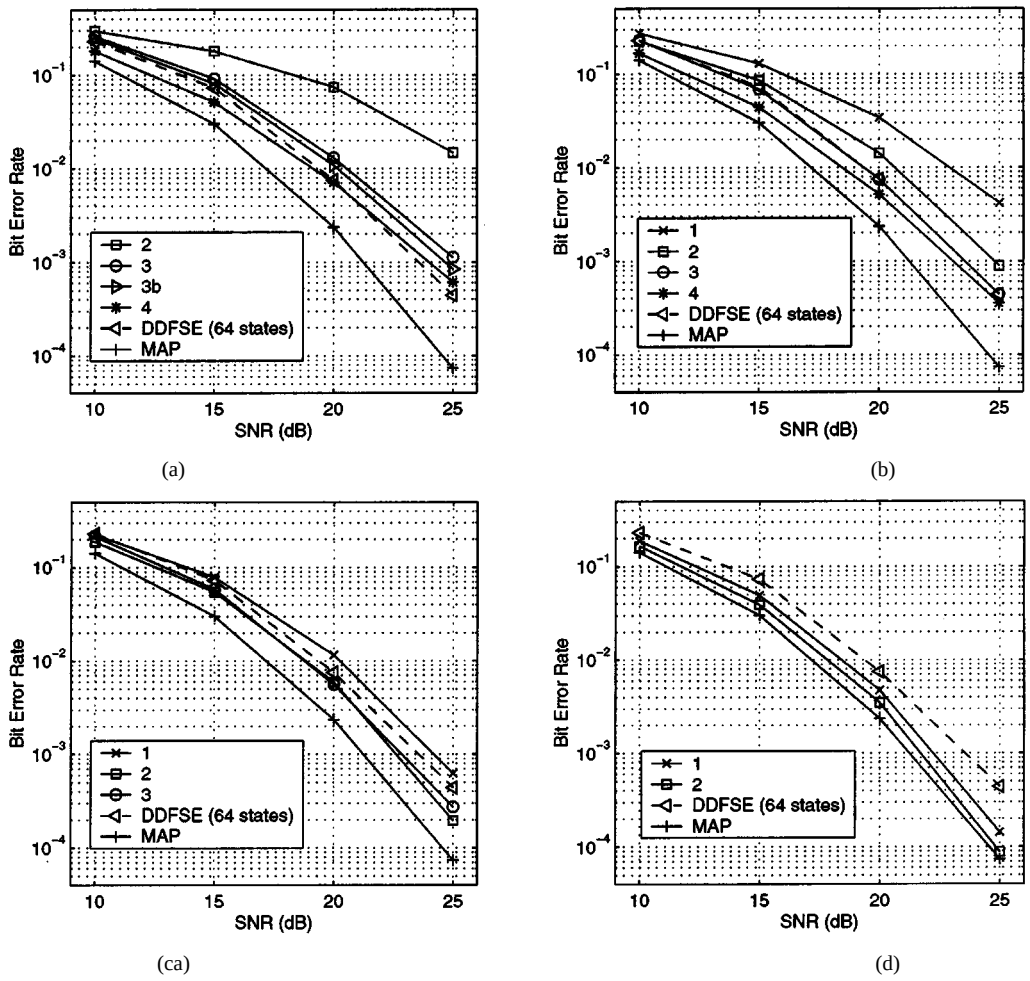


Fig. 12. Performance comparison for different schemes. (a) $M = 8$ states. (b) $M = 16$ states. (c) $M = 32$ states. (d) $M = 64$ states. Notation: $1 \rightarrow$ reduced-state equalizer (Section III-B), $2 \rightarrow$ reduced-state equalizer with two concatenated steps (Section III-C), $3 \rightarrow$ prefilter shortens the channel before the reduced-state equalizer (Section IV-A), $3b \rightarrow$ prefilter before the reduced-state equalizer with two concatenated steps, $4 \rightarrow$ a different prefilter shapes the observed sequence for the forward and backward recursion (Section IV-B).

overall channel. The tap positions window is $[5\ 6\ 7]$ for $\mathbf{h}_1$ and $\mathbf{h}_2$ channels, or equivalently $[5\ 6\ 7\ 8]$ for the overall $\mathbf{h}$ channel. Thus, the trellis after shortening has $8^3 = 512$ states.

The noise autocorrelation matrix at the prefilter output is $\mathbf{w}^*\mathbf{R}_{nn}\mathbf{w}$ where $\mathbf{R}_{nn}$ is the noise autocorrelation matrix at the prefilter input. Generally, the noise that corrupts the observed sequence of the trellis equalizer is colored. However, in our simulations, the trellis equalizer metrics were calculated assuming the noise white for simplicity of implementation, at the expense of some performance loss. Fig. 8 shows the performance for $M = 8$, 12, 16, 32, and 512 states. The 1–2-dB gap between MAP decoding without the shortening prefilter (4096 states) and MAP decoding after shortening (512 states) is due to ignoring the residual ISI and the noise correlation. This gap slightly increases with SNR.

The performance with 512 states is the best possible on the shortened trellis. The reduced-state trellis equalizer almost achieves it with 32 states. Moreover, the performance loss is less severe when the number of states is further reduced down to 8 states. As expected, the prefilter limits the best achievable performance, but allows the trellis equalizer to achieve it with a fewer number of active states.

D. Oversampling

Fig. 9 plots the performance when oversampling at the prefilter input and downsampling at its output, as described in Section IV-D. For comparison, we reproduced with dotted lines the curves of Fig. 8 (case of prefilter without oversampling). Oversampling offers a performance improvement that increases with SNR up to 1.5 dB.

E. Two Concatenated Steps With Prefilter

Fig. 10 plots the performance when employing both a prefilter and concatenation of two trellis steps of the following trellis. The dotted lines again reproduce the curves of Fig. 8 for comparison. Concatenating two trellis steps in this case can offer a small improvement, and only when employing $M = 8$ states. Indeed, the channel after the prefilter does not have a large difference of average power between the first (last) two taps, as the channel before the prefilter does. Thus, concatenating two trellis steps in this case does not offer much benefit.

F. Forward and Backward Prefilter

Fig. 11 plots the performance when using a different prefilter for the forward and backward recursions of the trellis equalizer,

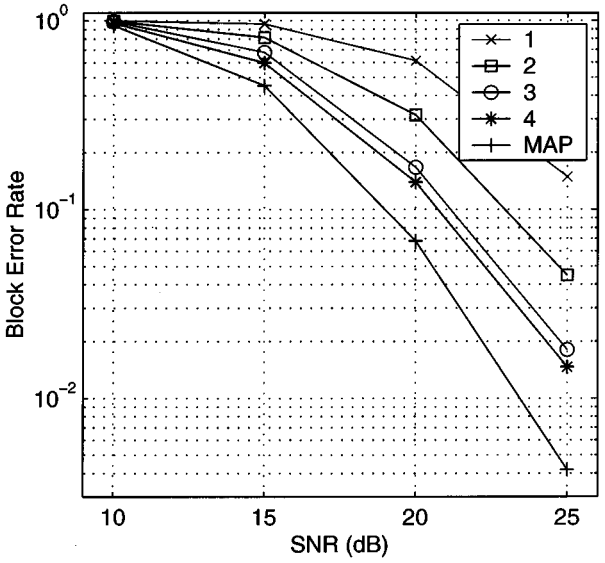


Fig. 13. Block error rate performance for the different schemes described in Fig. 12 for $M = 16$ active states.

as proposed in Section IV-B. Each prefilter has $N_f = 8$ taps and was designed according to (20). The forward prefilter used a weighting matrix $\mathbf{M} = \mathrm{diag}\{[1\ 0\ 0]\}$ and the backward a weighting matrix $\mathbf{M} = \mathrm{diag}\{[0\ 0\ 1]\}$. The trellis equalizer with $M = 16$ states almost achieves the best possible performance of 512 states on the shortened trellis.

G. Performance Comparison

Fig. 12 compares the performance achieved for $M = 8$, 16, 32, and 64 active states for the applicable equalization methods in each case. All plots also reproduce with a dotted line the performance of DDFSE with $M = 64$ active states and a length $N_f = 32$ shortening prefilter, as presented in [11]. Fig. 13 shows the block error rate for $M = 16$ active states.

For $M = 8$ and $M = 16$ states, a different prefilter for the forward and backward recursion leads to the best performance. For $M = 32$, the best performance is achieved when using a single prefilter. We get a slightly worse performance (without a prefilter) when concatenating two trellis steps. Finally, for $M = 64$ states, the best performance is obtained with no prefilter and concatenating two trellis steps.

VI. Conclusion

We presented soft trellis-based equalization algorithms for frequency-selective fading channels that employ transmit diversity. The basic receiver structure combines a shortening prefilter with a soft-decision reduced-state MAP equalizer/decoder. The task of reducing the complexity is divided among the prefilter and the trellis equalizer to minimize the performance loss.

The trellis equalizer employs a variation of the M-algorithm, based on the implementation of the forward–backward recursions in the logarithmic domain, that can offer improved performance. For channels with low power in the first or last tap, the performance of the reduced-state decoder may be improved by decoding two trellis steps jointly.

The prefilter design minimizes the residual ISI after shortening and the noise power at the prefilter output. It can also shape the power-delay profile of the target channel. This last property motivates the use of a different prefilter for the forward and the backward recursions to maximize their reliability. Oversampling at the receiver input and downsampling before the trellis equalizer allows the prefilter to better achieve a target channel impulse response when sufficient excess bandwidth is available. Metrics that quantify the prefilter performance were proposed and used to facilitate the choice of design parameters such as the prefilter length and the target channel length.

Acknowledgment

The authors would like to thank A. F. Naguib and W. Younis for many interesting discussions. They would also like to thank A. R. Calderbank and V. Vaishampayan for their continued support of the work.

References

[1] A. Duel-Hallen and C. Heegard, "Delayed decision-feedback sequence estimation," *IEEE Trans. Commun.*, vol. 37, pp. 428–436, May 1989.
[2] M. Eyuboglu and S. Qureshi, "Reduced-state sequence estimation with set partitioning and decision feedback," *IEEE Trans. Commun.*, vol. 36, pp. 13–20, Jan. 1988.
[3] J. Anderson and E. Offer, "Reduced-state sequence detection with convolutional codes," *IEEE Trans. Inform. Theory*, vol. 40, pp. 965–972, May 1994.
[4] V. Franz and J. Anderson, "Concatenated decoding with a reduced-search BCJR algorithm," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 186–195, Feb. 1998.
[5] C. Fragouli, N. Seshadri, and W. Turin, "On the reduced trellis equalization using the M-BCJR algorithm," in *Proc. CISS 2000*, Princeton, NJ, Mar. 2000, pp. WP28–WP33.
[6] S. U. Qureshi and E. E. Newhall, "An adaptive receiver for data transmission over time-dispersive channels," *IEEE Trans. Inform, Theory*, vol. IT-19, pp. 448–457, July 1973.
[7] D. D. Falconer and F. R. Magee, "Adaptive channel memory truncation for maximum likelihood sequence estimation," *Bell Syst. Tech. J.*, vol. 52, pp. 1541–1562, Nov. 1973.
[8] C. Yin and G. Yue, "Optimal impulse response shortening for discrete multitone transceivers," *Electron. Lett.*, vol. 34, no. 1, pp. 35–36, Jan. 1998.
[9] P. J. W. Melsa, R. C. Younce, and C. E. Rohrs, "Impulse response shortening for discrete multitone transceivers," *IEEE Trans. Commun.*, vol. 44, pp. 1662–1672, Dec. 1996.
[10] N. Al-Dhahir and J. M. Cioffi, "Efficiently computed reduced-parameter input-aided MMSE equalizers for NIL detection: A unified approach," *IEEE Trans. Inform. Theory*, vol. 42, pp. 903–915, May 1996.
[11] W. Younis and N. Al-Dhahir, "FIR prefilter design for MLSE equalization of space–time-codes transmission over multipath fading channels," in *Proc. ISCAS*, vol. 4, Sydney, Australia, May 2001, pp. 362–365.
[12] S. N. Diggavi and B. C. Ng, "On joint equalization and interference suppression for interference limited ISI channels,", preprint, 1999.
[13] N. Al-Dhahir, "Overview of equalization schemes for space–time-coded transmission with application to EDGE," in *Vehicular Technology Conf.*, Oct. 2001, vol. 2, pp. 1053–1057.
[14] A. Furuskar, S. Mazur, F. Muller, and H. Olofsson, "EDGE: Enhanced data rates for GSM and TDMA/136 evolution," *IEEE Personal Commun.*, vol. 6, pp. 56–66, June 1999.
[15] A. F. Naguib and N. Seshadri, "MLSE and equalization of space-coded signals," in *VTC-Spring*, vol. 3, May 2000, pp. 1688–1693.
[16] V. Tarokh, N. Seshadri, and A. R. Calderbank, "Space–time codes for high data rate wireless communications: Performance criterion and code construction," *IEEE Trans. Inform. Theory*, vol. 44, pp. 744–765, Mar. 1998.
[17] T. Rappaport, *Wireless Communications*. Englewood Cliffs, NJ: Prentice-Hall, 1996.
[18] W. Turin, *Digital Transmission Systems: Performance Analysis and Modeling*. New York: McGraw-Hill, 1998.

[19] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "A soft-input soft-output APP module for the iterative decoding of concatenated codes," *IEEE Commun. Lett.*, vol. 1, pp. 22–24, Jan. 1997.
[20] N. Al-Dhahir and J. M. Cioffi, "MMSE decision-feedback equalizers: Finite-length results," *IEEE Trans. Inform. Theory*, vol. 41, pp. 961–975, July 1995.
[21] D. Bertsekas, *Nonlinear Programming*. Belmont, MA: Athena Scientific, 1996.

Christina Fragouli (S'98–M'01) received the B.S. degree in electrical engineering from the National Technical University of Athens, Athens, Greece, in 1996 and the M.Sc. and Ph.D. degrees in electrical engineering from the University of California at Los Angeles in 1998 and 2000, respectively.

From September 2000 to September 2001, she was with AT&T Shannon Laboratory, Florham Park, NJ, as a Consultant. She is currently a Research Associate at the National Capodistrean University of Athens. Her research interests are in the area of channel coding and signal processing for communications.

Naofal Al-Dhahir (S'89–M'90–SM'98) received the M.S. and Ph.D. degrees from Stanford University, Stanford, CA, in 1990 and 1994, respectively, in electrical engineering.

He was an Instructor at Stanford University during Winter 1993. From August 1994 to July 1999, he was a member of the technical staff at the Communications Program at GE Corporate R&D Center in Schenectady, NY, where he worked on various aspects of satellite communication systems design and anti-jam GPS receivers. Since August 1999, he has been a principal member of technical staff at AT&T Shannon Laboratory in Florham Park, NJ. His current research interests include equalization schemes, space–time coding and signal processing, OFDM, and digital subscriber line technology. He has authored 38 journal papers and holds 7 U.S. patents in the areas of satellite communications and digital television. He is co-author of the book *Doppler Applications for LEO Satellite Systems* (Norwell, MA: Kluwer, 2002).

Dr. Al-Dhahir is a member of the IEEE SP4COM technical committee. He is Associate Editor for IEEE TRANSACTIONS ON SIGNAL PROCESSING (in the area of *Signal Processing for Communications*) and IEEE COMMUNICATIONS LETTERS.

Suhas N. Diggavi (S'93–M'93) received the B.Tech degree in electrical engineering from the Indian Institute of Technology, Delhi, India, and the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, in 1998.

Currently, he is a Senior Member pf Technical Staff in the Information Sciences Center, AT&T Shannon Laboratories, Florham Park, NJ. His research interests include wireless networks, information theory, source coding, and signal processing.

William Turin (SM'83) received the M.S. degree in mathematics from Odessa State University, Odessa, U.S.S.R., in 1958, and the Ph.D. degree in mathematics from the Institute for Problems of Mechanics of the Academy of Sciences of the U.S.S.R., Moscow, in 1966.

From 1960 to 1979, he was associated with the Moscow Institute of Electrical Engineering and Telecommunications, first as Assistant and later an Associate Professor. From 1980 to 1981, he was a Senior Research Scientist in the Department of Psychology, New York University, New York, NY. Since 1981, he has been a member of Technical Staff at AT&T Bell Laboratories, Holmdel and Murray Hill, NJ. Currently, he is a Technology Consultant at AT&T Labs-Research, Shannon Laboratory, Florham Park, NJ. His research interests include modeling channels with memory, digital signal processing, handwriting and speech recognition and compression, computer simulation, error correcting codes, microwave radio, and Markov processes. He is the author of five books and numerous papers.