Parallel Trellis Viterbi Algorithm for Sparse Channels

Nigel C. McGinty, Rodney A. Kennedy, *Member, IEEE*, and Peter Hoeher

***Abstract*—The Viterbi algorithm (VA), which normally operates using a single trellis, can be optimally reformulated into a set of independent trellises for a special class of sparse intersymbol interference (ISI) channels. These independent trellises operate in parallel and have less overall complexity than a single trellis. This trellis decomposition can be applied to a more general class of sparse channels approximately resulting in a suboptimal reduced complexity equalizer.**

***Index Terms*— Equalizers, intersymbol interference, sparse channels, Viterbi detection.**

I. System Model Definition

Consider the baseband equivalent communication model in Fig. 1. Binary phase-shift keying (BPSK) modulated symbols, a_k, propagate through the communications channel [1], which is modeled as a transversal filter with the impulse response $H(z) = \sum_{i=0}^{L} h_i z^{-i}$, or, equivalently in vector notation $H = [h_0, h_1, \cdots, h_L]'$ where the prime represents transposition, L parameterizes the channel length and k is the discrete time index.

The received signal y_k is given by

$$y_k = \sum_{i=0}^{L} h_i a_{k-i} + v_k \tag{1}$$

where the noise term, v_k, is zero-mean white Gaussian noise with variance σ_v^2. An equalizer, using y_k, generates estimates of the transmitted symbols, denoted by $\hat{a}_k$.

II. The Viterbi Algorithm (VA)

The VA is a breadth-first trellis-search algorithm [2]. The 2^L states in the trellis, at time k, are defined by

$$S_k = [a_{k-1}, a_{k-2}, \cdots, a_{k-L}]'. \tag{2}$$

Each temporal transition between states, S_k to S_{k+1}, under the input a_k, has the branch metric

$$\lambda(y_k, a_k, S_k) = |y_k - h_0 a_k - S_k' \Psi H|^2 \tag{3}$$

Manuscript received January 16, 1998. The associate editor coordinating the review of this letter and approving it for publication was Prof. C. D. Georghiades. This work was supported in part by the Australian Government under the Cooperative Research Centres Program.

N. C. McGinty is with the Communications Division, Defence Science and Technology Organization, Salisbury, SA, 5108, Australia.

R. A. Kennedy is with the Telecommunications Engineering Group and the Cooperative Research Centre for Robust and Adaptive Systems, Research School of Information Sciences and Engineering, The Australian National University, Canberra ACT 0200, Australia.

P. Hoeher is with the German Aerospace Center (DLR) Oberpfaffenhofen, Institute for Communications Technology, D-82230 Wessling, Germany.

Publisher Item Identifier S 1089-7798(98)04030-7.

where Ψ is a $L \times (L+1)$ matrix having zero elements except for $\psi_{i,\,i+1} = 1$ for $i \in \{1, 2, \cdots, L\}$. At S_{k+1} the most likely path is selected according to the path metric calculation

$$\Lambda(S_{k+1}) = \min_{S_k} |\lambda(y_k, a_k, S_k) + \Lambda(S_k)|^2 \tag{4}$$

where $\Lambda(S_k)$ is the path metric for state S_k. Each state at time k has a survivor sequence which contains the most likely path through the trellis to that node. If the 2^L survivor sequences are traced back in time, at some point they converge and a decision can then be made as to the transmitted symbol, this time is referred to as the decision delay and is equal to D.

III. Sparse Channel Model

A sparse channel consists of a high proportion of zero valued taps in its channel impulse response. These channels can exist due to multiple transmission paths in a sparse environment, such as in an aeronautical channel. The *zero-pad channel class* has the structure

$$H = [h_0, \underbrace{0, 0, \cdots, 0}_{(f-1)}, h_1, \underbrace{0, \cdots, 0}_{(f-1)}, \cdots, \underbrace{0, \cdots, 0}_{(f-1)}, h_g]' \tag{5}$$

where $g+1$ is the number of nonzero taps in the channel. This channel consists of an equal number of zero valued taps, $(f-1)$, between each nonzero tap. A *zero-pad* channel can be characterized by its nonzero parameters as

$$H_{\text{ZP}}(f) = [h_0, h_1, \cdots, h_g]'. \tag{6}$$

IV. Parallel Trellis Viterbi Algorithm

Consider the simple channel, $H = [h_0, 0, h_2]'$. Given the receipt of y_k, the VA determines the branch metric

$$\lambda(y_k, a_k, S_k) = |y_k - h_0 a_k - h_2 a_{k-2}|^2 \tag{7}$$

which is seen only to depend on alternate data symbols. Using this information two independent trellises can be constructed, one estimating the sequence $\{a_0, a_2, a_4, \cdots\}$ and the other estimating $\{a_1, a_3, a_4, \cdots\}$. This independent trellis decomposition can be generalized to an arbitrary zero pad channel as given by (5).

The *parallel trellis Viterbi algorithm* (PTVA) consists of f independent parallel trellises. Each trellis, numbered $n \in \{1, 2, \cdots, f\}$, consists of 2^g states which, at time k, are defined as

$$S_{k,\,n} = [a_{k-f}, a_{k-2f}, \cdots, a_{k-gf}]'. \tag{8}$$

For each time instance the trellis number n is incremented by modulo f. Note, that each trellis has a transition once

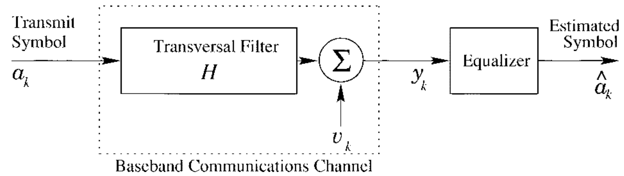

Fig. 1. A block diagram of a generalized communications system, where a_k propagates through a channel with the impulse response $H = [h_0, h_1, \cdots, h_L]'$, and an estimate of the transmitted symbol, $\hat{a}_k$, is made using the received sequence.

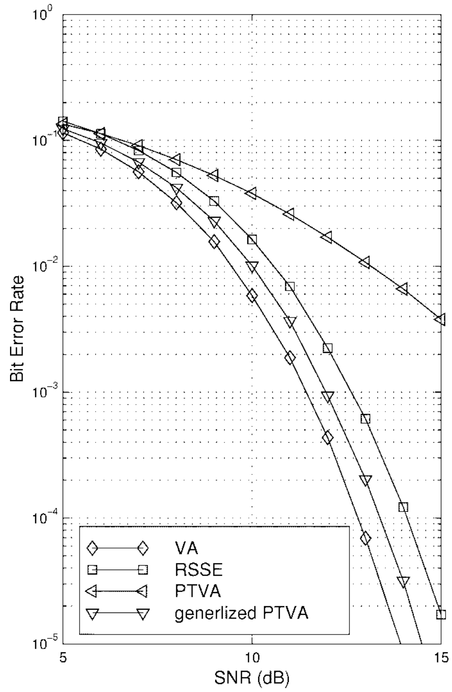

Fig. 2. BER versus SNR for the channel with coefficients $H = [0.6852, 0.2056, 0.5482, 0.1370, 0.4111]'$.

every f samples; the temporal state spacing is increased from baud interval, T, to fT, allowing more time between metric computations.

Continuing with the definition of the PTVA, at time k for a given state, $S_{k,n}$, with the received symbol y_k and under the input a_k, the branch metric can be calculated by

$$\lambda(y_k, a_k, S_{k,n}) = |y_k - h_0 a_k - S'_{k,n} \Psi H_{\text{ZP}}(f)|^2 \quad (9)$$

where Ψ is defined similarly to that in (3). Finally, for a given state pair, $S_{k+f,n}$ and $S_{k,n}$, the path metric, $\Lambda(S_{k+f,n})$, is defined according to

$$\Lambda(S_{k+f,n}) = \min_{S_{k,n}} |\lambda(y_k, a_k, S_{k,n}) + \Lambda(S_{k,n})|^2. \quad (10)$$

The PTVA reduces the complexity of the VA for the zero pad channel class with no loss of optimality. The complexity reduction can be attributed to the lower total number of states and the increased temporal state spacing. Table I compares the trellis and numerical requirements for the PTVA to the VA.

Recently a VA using multiple trellises to equalize sparse channels was proposed in [3]. The trellises for this approach are irregular (the state transitions are dependent on time). At any time, there are D parallel trellises of length D, where $D \gg f$ is the decision delay. Compared to [3], the regular trellis structure of the PTVA has a simpler and more straightforward implementation.

TABLE I
COMPARISON BETWEEN THE VA AND PTVA WHERE $L = fg$

	VA	PTVA
No. of trellises	1	f
States per trellis	2^L	2^g
State spacing	T	fT
Additions	2^{L+1}	2^{g+1}
Multiplications	$(L+1)2^{L+1}$	$(g+1)2^{g+1}$
Compare-select	2^L	2^g

V. GENERALIZED PTVA

A *general sparse channel* contains a similar structure to the zero pad channel class except the assumed zero valued taps of the zero pad channel may contain small nonzero values. When the PTVA equalizes general sparse channels a degree of correlated noise is introduced due to ignoring energy in the channel impulse response. A *generalized PTVA* is proposed for the equalization of such general sparse channels. This algorithm retains the PTVA trellis structure and uses state estimates from other trellises in the branch metric calculation for the low energy taps.

Given the parameters f and g, as defined for the zero pad channel class, a general sparse channel can be written as

$$H = [h_0, h_1, \cdots, h_{f-1}, h_f, h_{f+1}, \cdots, h_{2f}, \cdots, h_{gf}]'. \quad (11)$$

Define f subvectors of H, each containing g elements, as

$$H_i = [h_i, h_{f+i}, h_{2f+i}, \cdots, h_{(g-1)f+i}]' \quad (12)$$

indexed by $i \in \{1, 2, \cdots, f\}$. The branch metric for state $S_{k,n}$ under transition a_k can be written as

$$\lambda(y_k, a_k, S_{k,n}) = \left| y_k - h_0 a_k - S'_{k,n} H_f - \sum_{j=1}^{f-1} \hat{S}'_{k+j,q} H_j \right|^2. \quad (13)$$

The summation term in (13) incorporates the state estimates $\hat{S}_{k+j,q}$ at time $k+j$ from trellis q, where $q = (n+j) \bmod f$. The path metric is determined using (10).

In (13) a suitable set of state estimates at time $k+j$ can be efficiently computed by

$$\hat{S}_{k+j,q} = \arg \min_{S_{k+j,q}} |\Lambda(S_{k+j,q})| \quad (14)$$

for $j = \{1, 2, \cdots, f-1\}$. The state estimates are not binding as the final symbol estimates will be extracted from the survivor sequence of the appropriate trellis. The effect of an incorrect decision is generally only minimal, it will cause a decision error if an incorrect path obtains a lower metric than the most likely path due to that estimation error (as opposed to a channel noise induced error).

A. Performance Evaluation

The channel $H = [0.6852, 0.2056, 0.5482, 0.1370, 0.4111]'$ is considered where 0.2056 and 0.1307 are the low energy taps. The bit-error rate (BER) versus signal-to-noise ratio performance curves for the PTVA, generalized PTVA, VA and reduced state sequence estimator (RSSE) [4] (which is used as a comparative reduced complexity VA) are shown in Fig. 2. The relatively poor performance of the PTVA can be attributed to the low energy taps not being sufficiently small. The VA has a 0.62-dB better performance than the generalized PTVA, which is 0.71 dB better than the RSSE a BER of 10^{-4}.

The generalized PTVA's performance is dependent on the channel. This indicates that the general sparse channel must be close in structure to the zero pad channel class in order to minimize the algorithm's sensitivity to the quality of the preliminary state estimates.

REFERENCES

[1] J. G. Proakis, *Digital Communications,* 3rd ed. Singapore: McGraw Hill, 1995.

[2] G. D. Forney, "The Viterbi algorithm," *Proc. IEEE,* vol. 61, pp. 268–278, Mar. 1973.

[3] N. Benvenuto and R. Mar.esani, "The Viterbi algorithm for sparse channels," *IEEE Trans Commun.,* vol. 44, pp. 287–289, Mar. 1996.

[4] M. V. Eyuboǧlu and S. U. H. Qureshi, "Reduced-state sequence estimation with set partitioning and decision feedback," *IEEE Trans. Commun.,* vol. 36, pp. 13–20, Jan. 1988.