List Viterbi Decoding Algorithms with Applications

Nambirajan Seshadri, *Member IEEE* and Carl-Erik W. Sundberg, *Fellow IEEE*

Abstract—**A list Viterbi decoding algorithm (LVA) produces a rank ordered list of the L globally best candidates after a trellis search. Here, we present two such algorithms, (i) a parallel LVA that simultaneously produces the L best candidates and (ii) a serial LVA that iteratively produces the k^{th} best candidate based on knowledge of the previously found $k-1$ best paths. The application of LVA to a concatenated communication system consisting of an inner convolutional code and an outer error detecting code is considered in detail. Analysis as well as simulation results show that significant improvement in error performance is obtained when the inner decoder, which is conventionally based on the Viterbi algorithm (VA), is replaced by the LVA. An improvement of up to 3 dB is obtained for the additive white Gaussian noise (AWGN) channel due to an increase in the minimum Euclidean distance. Ever larger gains are obtained for the Rayleigh fading channel due to an increase in the time diversity. It is also shown that a 10% improvement in throughput is obtained along with significantly reduced probability of a decoding failure for a hybrid FEC/ARQ scheme with the inner code being a rate compatible punctured convolutional (RCPC) code.**

1. Introduction

The Viterbi algorithm (VA) [1-2] performs efficient maximum likelihood decoding of finite state signals observed in noise. In many situations, the state space is either too large to search or only a part of the signal is generated by a finite state machine. Two examples of such communication systems are shown in Figure 1. The first one is a concatenated coding system consisting of an outer error detecting code and an inner error correcting convolutional code. The second one is an outer speech coder followed by an inner error correcting convolutional code. In the first example, the joint state space can be defined but it may be too huge to search. In the second example, the combined source-channel coder state space is not precisely defined. Conventionally, the inner code is decoded first under the assumption that the input to the inner code is independent and identically distributed data. This is then followed by outer decoding. The inner decoder, which is normally based on the Viterbi algorithm (VA), searches for the best path through a trellis that is defined by the inner code. Considerable improvement in performance is obtained over this conventional decoding approach when the knowledge of $L > 1$ best paths through this trellis is utilized during subsequent decoding.

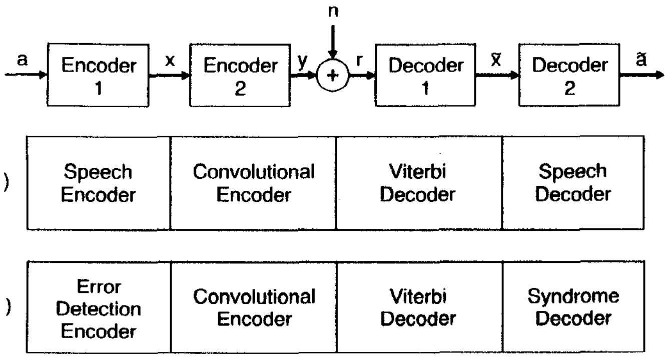

Fig. 1. Examples of communication systems with concatenated decoders.

Various generalizations of the VA have appeared in the literature. Forney [3] considered a list-of-2 maximum likelihood decoder for the purpose of analyzing sequential decoding algorithms [4]. Yamamoto and Itoh (Y-I) [5] proposed an ARQ algorithm where the decoder requests a frame repeat whenever the best path into every state at some trellis level is "too close" to the second best candidate into every state. However they do not explicitly make use of the second best candidate as we do. More recently, Hashimoto [6] has proposed a list type reduced-constraint generalization of the Viterbi algorithm which contains the Viterbi and the M-algorithm [7] as special cases. The purpose of this algorithm is to keep the decoding complexity to be no more than that of the Viterbi algorithm and to avoid error propagation due to reduced state decoding. Again, no explicit use of the survivors other than the best is made after decoding. The Y-I algorithm has been successfully employed in a concatenated coding scheme by Deng and Costello [8]. The inner decoder is a convolutional code with the Y-I decoding algorithm and the outer decoder is an errors and erasures decoder where the symbol erasure information is supplied by the Y-I algorithm. List decoding of block codes for the binary symmetric channel has recently been studied by Elias [9]. This work contains an extended reference list on list decoding of block codes for this channel. A new soft output Viterbi algorithm which gives analog reliability information associated with each decoded symbol from the VA has been proposed by Hagenauer and Hoeher [10,11]. Such algorithms are typically used in the inner decoding stage.

Paper approved by Sergio Benedetto, the Editor for Signal Design, Modulation and Detection of the IEEE Communications Society. Manuscript received March 4, 1991; revised March 31, 1992. This paper was presented in part at the Conference on Information Sciences and Systems, Princeton, NJ, March 1990 and the IEEE Global Telecommunications Conference, San Diego, CA, November 1989.

AT&T Bell Laboratories, Signal Processing Research Department, 600 Mountain Avenue, Murray Hill, NJ 07974

IEEE Log Number 9400913.

0090-6778/94$04.00 © 1994 IEEE

Here, we present two LVAs[1] that produce a rank ordered list of the L globally best candidates after a trellis search. The two algorithms are (i) a parallel LVA that simultaneously produces the L best candidates and (ii) a serial LVA that iteratively produces the k^{th} best candidate based on the knowledge of the previously found $k-1$ candidates. We consider in detail the application of this algorithm to a concatenated communication system consisting of an inner forward error correction (FEC) code and an outer (ideal) error detecting code. Later in the text, we briefly consider the application of this algorithm to other concatenated communication systems.

The parallel and serial LVA along with implementation and complexity details are described in Section 2. Section 3 considers the application of the LVA to hybrid forward error correction and automatic repeat request (FEC/ARQ) data transmission scheme. For a textbook treatment of hybrid FEC/ARQ schemes, we refer the reader to [12]. In this work, the inner code is either a fixed rate convolutional code or an incremental redundancy transmission code such as the rate compatible punctured convolutional (RCPC) code [13], and the outer code is assumed to be an ideal error detecting code. Asymptotic error performance and simulation results are presented for the Gaussian and Rayleigh fading channels. Section 4 concludes the work.

2. List Viterbi Decoding Algorithms

In this section we present the parallel and serial LVAs. For the parallel algorithm, the task of identifying the L best candidates is achieved in one pass through the trellis while the serial algorithm achieves the same result by L successive passes through a trellis. We begin by summarizing the Viterbi algorithm which also establishes the notations to be used subsequently.

2.1. *Summary of the Viterbi Algorithm*

An N state fully connected trellis is shown in Figure 2. (Some of the connections are non-existent for rates less than $\log_2 N$ bits/state transition.) The total number of sequences is N^2M where M is the total number of such trellis sections. The incremental cost (metric) associated with moving from state j at time $t-1$ to state i at time t is given by $c_t(j,i)$ (where $c_t(j,i)=\infty$ if j and i are not connected). The problem is to find the best state sequence (with minimum cost) through the trellis, starting from, for example, state 1 (at time 0) and ending at state 1 (at time M). Let the minimum cost to reach state j at time t from the known starting state be $\phi_t(j)$. Let the history of the best path be stored in the array $\xi_t(j)$. At time t, $\xi_t(j)$ is the state occupied by the best path into state j at time $t-1$. The Viterbi algorithm can be summarized as follows:

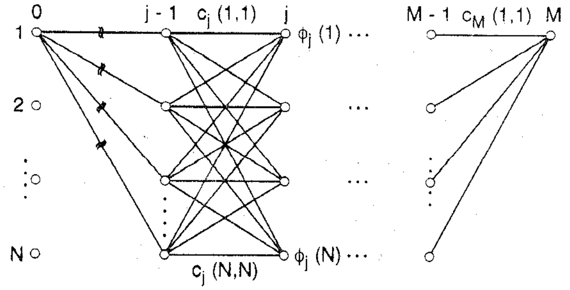

Fig. 2. Fully connected trellis with N states.

1. Initialization: $(t=1)$

$$\begin{aligned} \phi_1(i) &= c_1(1,i)\ , \qquad (1)\\ \xi_1(i) &= 1\ ,\\ 1 \le i &\le N\ . \end{aligned}$$

2. Recursion: $(1<t<M)$

$$\begin{aligned} \phi_t(i) &= \min_{1\le j\le N} [\phi_{t-1}(j)+c_t(j,i)]\ , \qquad (2)\\ \xi_t(i) &= \arg\min_{1\le j\le N} [\phi_{t-1}(j)+c_t(j,i)]\ ,\\ 1 \le i &\le N\ . \end{aligned}$$

3. Termination: $(t=M)$

$$\begin{aligned} \phi_M(1) &= \min_{1\le j\le N} [\phi_{M-1}(j)+c_M(j,1)]\ , \qquad (3)\\ \xi_M(1) &= \arg\min_{1\le j\le N} [\phi_{M-1}(j)+c_M(j,1)]\ . \end{aligned}$$

4. Path Backtracking:
The best state sequence is

$$(1, i_1, \ldots, i_{M-1}, 1)\ ,$$

where

$$\begin{aligned} i_t &= \xi_{t+1}(i_{t+1})\ , \qquad (4)\\ 1 \le t &\le M-1\ . \end{aligned}$$

2.2. *Parallel LVA*

The parallel LVA finds the L best paths simultaneously by computing the L best paths into each state at every time instant.

Parallel Algorithm:

Let $\phi_t(i,k), 1\le k\le L$, be the k^{th} lowest cost to reach state i at time t from state 1 at time 0. Similarly let $\xi_t(i,k)$ be the state (and $r_t(i,k)$ be the corresponding ranking) of the k^{th} best path at time $t-1$, when this path passes through state i at time t.

1. Initialization $(t=1)$

$$\begin{aligned} \phi_t(i,k) &= c_1(1,i),\\ \xi_t(i,k) &= 1, \qquad 1\le i\le N,\ \ 1\le k\le L. \end{aligned} \qquad (5)$$

[1] We have used the term generalized Viterbi algorithm (GVA) instead of List Viterbi algorithm (LVA) in conference versions of this paper.

2. Recursion $(1 < t < M)$

$$\phi_t(i,k) = \min_{\substack{1\le j\le N\\ 1\le l\le L}}{}^{(k)} \, [\phi_{t-1}(j,l) + c_t(j,i)], \quad 1 \le i \le N$$

where $\min^{(k)}$ denotes the k^{th} smallest value.

$$(j^*, l^*) = \arg\min_{\substack{1\le j\le N\\ 1\le l\le L}}{}^{(k)} \, [\phi_{t-1}(j,l) + c_t(j,i)], \ 1 \le i \le N. \tag{6}$$

Here j^* is the predecessor state for the k^{th} best path into state i and l^* is the corresponding ranking.

$$\xi_t(i,k) = j^*$$
$$\gamma_t(i,k) = l^*.$$

3. Termination $(t = M)$

$$\phi_M(1,k) = \min_{\substack{1\le j\le N\\ 1\le l\le L}}{}^{(k)} \, [\phi_{M-1}(j,l) + c_M(j,1)] \ ,$$

$$(j^*, l^*) = \arg\min_{\substack{1\le j\le N\\ 1\le l\le L}}{}^{(k)} \, [\phi_{M-1}(j,l) + c_M(j,1)] \tag{7}$$

$$\xi_t(i,k) = j^*$$
$$\gamma_t(i,k) = l^*.$$

4. Path Backtracking:
The k^{th} best state sequence is

$$(1, j_1, j_2, \ldots, j_{M-1}, 1) \ ,$$

where

$$\begin{aligned} j_t &= \xi_{t+1}(j_{t+1}, l_{t+1}) \ , \\ l_t &= r_{t+1}(j_{t+1}, l_{t+1}) \ , \\ j_{M-1} &= \xi_M(1,k), \end{aligned} \tag{8}$$

and

$$l_{M-1} = r_M(1,k) \ .$$

The parallel algorithm is illustrated in Figure 3 where at each state and at every time, the NL accumulated costs are computed and the L smallest accumulated cost paths along with their costs are stored. This algorithm requires maintaining a cost array of NL accumulated costs and a state array of $NL \times M$ which stores the path history for each time instant.

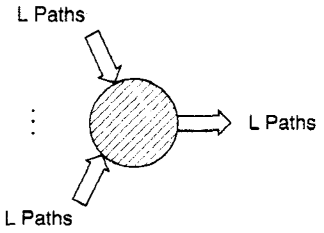

Fig. 3. Dynamic programming for finding the L-best paths implemented in a parallel manner.

It is easy to modify the parallel algorithm to operate in a continuous transmission mode rather than in a block mode by maintaining the L best paths into each state and by releasing the L best symbols (after tracking back D_p symbols where D_p is the decoding depth) at each instant corresponding to the best among the NL survivors, the second best among the NL survivors etc. A simplified parallel list-of-2 VA can be found in [14].

2.3. Serial LVA

The serial algorithm finds the L most likely candidates, one at a time, beginning with the most likely path. The main benefit of this algorithm is that the k^{th} best candidate is computed only when the previously found $k-1$ candidates are determined to be in "error". This avoids many of the unwanted computations of the parallel algorithm. We illustrate the serial algorithm for finding the 2nd and 3rd best before presenting the general algorithm. The best path is assumed to be found by the Viterbi algorithm. It is convenient to retain all the computations performed by the Viterbi algorithm including the cost associated with each locally best (partial) state sequence.

2nd Best Path: We make use of the fact that the globally 2nd best path after leaving the best path at some instant, merges with the best path at a later instant and never diverges again. This is because any subsequent divergence will result in a higher cost path. Figure 4 shows the admissible and inadmissible topology for the second best path. Using this fact, the following recursion can be written. Let $i^{(t)}$ be the state occupied by the best path at time t. Then

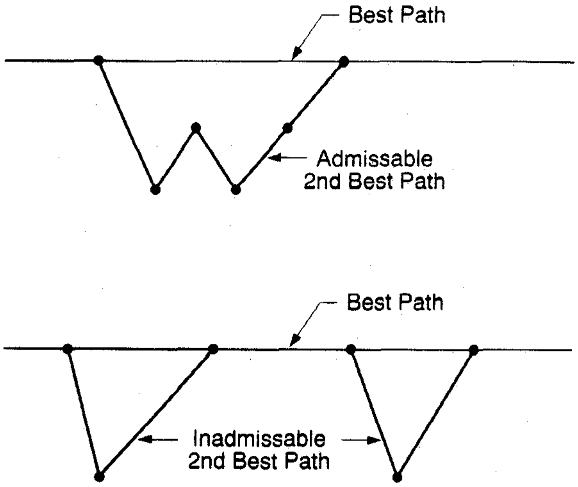

Fig. 4. Admissible and inadmissible topologies for the 2nd best path.

$$\phi_t(i^{(t)}, 2) \ = \ \min\left\{ \left[\phi_{t-1}(i^{(t-1)}, 2) + c_t(i^{(t-1)}, i^{(t)})\right] ; \ \min_{\substack{1\le l\le N\\ l\ne i^{(t-1)}}} \left[\phi_{t-1}(l,1) + c_t(l, i^{(t)})\right] \right\} , \tag{9}$$

The first term in the recursion of (9) is the second best path to reach state $i^{(t)}$, with the constraint that this path should have merged with the globally best by at least time $t-1$. The second term is the second best path to reach state $i^{(t)}$, with the constraint that this path should merge with the globally best at time t and not before. The one with the minimum cost remains in contention.

In order to release the second best path, the time instant at which the last best merge happened should be recorded as well as the corresponding predecessor state. Supposing this time instant is τ and the predecessor state is $l^{(\tau-1)}$. Then, the globally second best state sequence is the best path from state 1 (at time 0) to state $l^{(\tau-1)}$. The remainder is the same as the globally best state sequence. We note that, unlike the parallel algorithm, it is not necessary to keep track of $2N$ accumulated costs and $2N$ path histories. Furthermore, $\phi_{t-1}(l,1)$ in (9) is available by prior use of the Viterbi algorithm. Thus storing these values will significantly reduce the computational cost.

3^{rd} Best Path: Let us first consider the state $i^{(t)}$ occupied by the best path at time t. Let us assume that this state is not the one where the globally 2^{nd} best finally merges with the globally best path. A "final merge" means that this path has not merged with the globally best path at time $t-1$ and remains merged from time t onwards. If the globally third best candidate were to merge with the best candidate for the final time at time t, then this candidate must be the locally second best path into this state. Figure 5a shows such candidates. The locally 2^{nd} best candidate into this state is computed and its cost $\phi_t(i^{(t)},2)$ is added to the cost of the globally best path over the remaining time. The cost of this locally 2^{nd} best candidate is compared to that of the surviving candidate and the candidate with the lower cost remains in contention. This recursion is performed until the end of the trellis is reached.

Let us consider the time τ at which the globally 2^{nd} best path finally merges with the best. Then we note that the candidate for being the 3^{rd} best and finally merging with the best path at time τ is the second best to the globally 2^{nd} best path over the time span $t=0$ to $t=\tau$ and is the globally best candidate over the remaining time span. The cost of this candidate is compared to the lone survivor obtained previously and the lowest of the two remains in contention. Figure 5b shows the locally second best candidates to the globally 2^{nd} best.

We now summarize the serial algorithm. We also assume that the globally best sequence has been found by the Viterbi algorithm and that the locally best path into each state at every time instant is known along with its associated cost. Let the best state sequence be $(1, i^{(1)}, i^{(2)}, \ldots, i^{(j)}, \ldots, i^{(M-1)}, 1)$.

Serial Algorithm:

Initialization:

a. Initialize the number of candidates found to $k=1$.

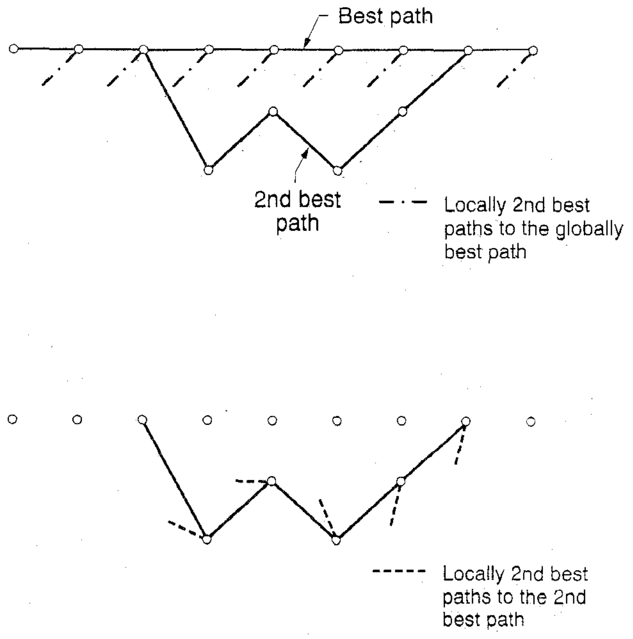
Best path
2nd best path
Locally 2nd best paths to the globally best path
Locally 2nd best paths to the 2nd best path

Fig. 5. Steps involved in serial list-of-3 VA. (a) Locally 2^{nd} best candidates to the globally best path. (b) Locally 2^{nd} best candidates to the globally 2^{nd} best path.

b. Set up a path matrix of size $L \times M$, to store the state sequences, where L corresponds to the maximum number of best state sequences that are to be found, and M is the length of the state sequence.

c. Form a "merge" count array of size M, where the j^{th} element C_j is the number of paths, among the previously found globally best paths that finally merge with the globally best path at time j. We initialize C_j to 1 for all j.

Recursion:

a. At time j, find the path that finally merges at time j, and one that is in contention for being the k^{th} best candidate. This candidate is the ${C_{j+1}}^{\text{th}}$ best path from time instant 0 (and state 1) to time instant j (and state M). The cost of this candidate is added to the cost of the globally best path over the remainder of the trellis and is compared to the cost of the surviving candidate. The lowest of the two remains in contention.

b. Update the ${C_j}^{\text{th}}$ entry of the merge count array, i.e., $C_j = C_j + 1$ for that time instant j when the final merge of the l^{th} best path with the globally best path happens.

c. Increment k by 1.

d. Loop back to (a) until $k = L$.

We note that further computational savings can be realized by storing the currently available L best candidates in a stack. Let us assume that the k^{th} best path is to be found. Then, it is clear that the k^{th} globally best candidate is at least the k^{th} best candidate in the stack. It can also be verified that the only other possibility is that the k^{th} candidate is the 2^{nd} best path to the $(k-1)^{\text{th}}$ globally best path. This candidate is compared to the k^{th} candidate in the stack and the lower of the two is the globally k^{th} best path. The stack is continuously updated in the process of finding the k^{th} best candidate. A tree-trellis search algorithm similar to this final version of the serial algorithm (i.e., using a stack) has been proposed by Soong and Huang [15]. They use the forward trellis search using the VA for finding the locally best candidate into each state at every instant followed by a *single* backward tree search (backwards stack) to find all the remaining $L-1$ best paths.

2.4. Computational and Storage Requirements

It is clear that the parallel algorithm requires L times more storage and computation than those of the VA. On the other hand, using the final version of the serial algorithm which stores all the intermediate computations, in order to find the k^{th} best path, it is sufficient to find the 2^{nd} best path to the $k-1^{\text{th}}$ globally best path. This requires the evaluation of M path metrics. If all the intermediate computations are stored, about N metric additions and N comparisons are required at each instant (by using eqn. (9)). Thus the total cost to find the k^{th} best path is NM additions and comparisons. Some additional cost is incurred in inserting a newly found candidate into the stack if its cost is lower than that of any candidate in the stack. In any event, the computational cost is much lower than that of the parallel list Viterbi algorithm. Also, the average computational cost for the serial algorithm may be smaller than that of the parallel algorithm. The serial algorithm finds the k^{th} best only if the $(k-1)^{\text{th}}$ best is in "error". On the other hand, the parallel LVA has to find the L best all the time. The storage requirement to store the accumulated cost into each state at every instant is about M times higher than the VA.

3. Applications to Data Transmission

Here, the outer code is an error detecting code and the inner code is a convolutional code. Conventionally, the inner decoder based on the VA releases the best decoded sequence and the outer decoder performs error detection on this decoded sequence. If an error is detected then the outer decoder, for example, may request the transmitter to retransmit. In this case the inner decoder has to perform a second Viterbi decoding etc. Here we propose that the decoder based on the LVA releases the L best candidates and the outer decoder (assumed ideal in the performance evaluation below) selects the correct one from the L best if it exists. The serial algorithm is ideally suited for this purpose. This is because the outer decoder performs error detection on the k^{th} best candidate, $k = 1, \ldots, L$ and requests the inner decoder for the $k+1^{\text{th}}$ best candidate only if the k^{th} best is found to be in error. A decoding failure is declared if $k = L$. We note that using the serial algorithm, the task of finding the 2^{nd} best, the third best etc., requires lower computational effort than performing a second Viterbi decoding as required in the conventional approach. Figure 6 shows the decoder operation for this concatenated coding system. We first evaluate the asymptotic error performance of this coding system in the presence of noise and fading. In particular, we are interested in the probability of the correct candidate not being among the L best. The results for the Gaussian channel are derived for any L and for any code (including the inner code being a coded modulation) while for the Rayleigh fading channel the results are derived for binary codes and $L = 2$.

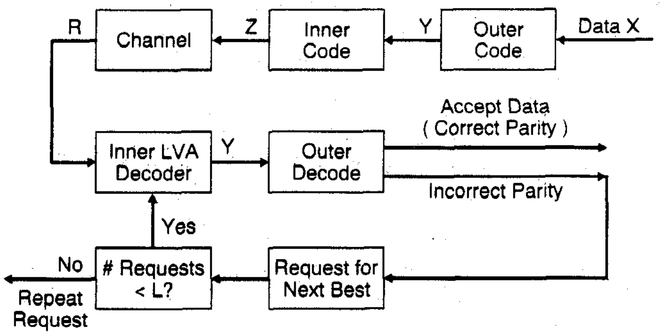

Fig. 6. Block diagram of data transmission system with LVA decoder.

3.1. Asymptotic SNR Gain with LVA Decoding for the AWGN Channel

The probability of incorrect decoding is evaluated for the additive white Gaussian noise channel when the LVA is used for decoding. This is the probability that the correct candidate is not in the list of the L best candidates. Our analysis makes use of signal space geometry. The results show that the worst case asymptotic coding gain is independent of the code, its rate, and the modulation scheme. It provides an indication of the gain that can be achieved by practical codes or modulation schemes at high channel SNRs. We will present simulation results with specific codes to show that practical gains are close to the theoretical predictions.

$L = 1$: When $L = 1$, i.e., when the VA is used for decoding, the error performance is mainly determined by the pair of signal points that are most likely to be confused by each other. This corresponds for the AWGN channel to pair of signal points (codewords) that are at the minimum Euclidean distance. Let $\underline{a}$ and $\underline{b}$ be two codewords, and let $s(\underline{a})$ and $s(\underline{b})$ be the corresponding signals that are actually transmitted on the channel. The minimum Euclidean distance is then given by

$$D_{\min} = \min_{\underline{a} \neq \underline{b}} D(\underline{a}, \underline{b}) \tag{10}$$

where

$$D^2(\underline{a}, \underline{b}) = \|s(\underline{a}) - s(\underline{b})\|^2 \tag{11}$$

where $\|(\cdot)\|$ is the squared Euclidean distance. The probability of error at very large signal to noise ratios is given by

$$P_e = \text{const. } Q\left(\sqrt{\frac{D^2_{\min}}{2N_0}}\right) \tag{12}$$

where

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-t^2/2}\, dt \tag{13}$$

and N_0 is the one-sided power spectral density of the noise.

$L = 2$: When $L = 2$, we are interested in finding the probability that the correct data sequence is not among the list of the two best decoder outputs. We now have three candidates $s(\underline{a})$, $s(\underline{b})$, and $s(\underline{c})$ corresponding to data sequences $\underline{a}$, $\underline{b}$ and $\underline{c}$. These three signal points form a triangle with the edges of the triangle being at least $D_{\min}$ in length. The closest point in the region of error to $s(\underline{a})$ is o as shown in Figure 7, and the Euclidean distance from $s(\underline{a})$ to o, D_{eq}, will dominate the error probability at large SNRs. This distance is given by

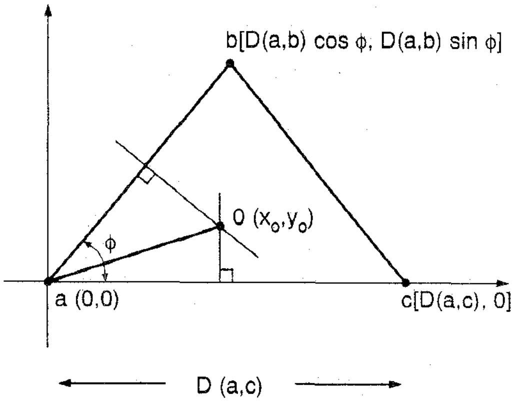

Fig. 7. Signal space geometry for LVA with $L = 2$.

$$D^2_{eq} = \frac{D^2(\underline{b},\underline{c})}{4\left(1 - \left(\frac{D^2(\underline{b},\underline{c}) - D^2(\underline{a},\underline{c}) - D^2(\underline{a},\underline{b})}{2D(\underline{a},\underline{c})D(\underline{a},\underline{b})}\right)^2\right)} \,. \tag{14}$$

This distance takes on its lowest value when all the three points $s(\underline{a})$, $s(\underline{b})$ and $s(\underline{c})$ are at $D_{\min}$ from each other. The coding gain G of the LVA ($L = 2$) over the VA ($L = 1$) is then given by

$$10\log_{10}(G) = 10\log_{10}\left(\frac{D_{eq}}{D_{\min/2}}\right)^2 = 10\log_{10}\left(\frac{4}{3}\right) \tag{15}$$

which is 1.25 dB.

$L = 3$: When $L = 3$, an error occurs whenever the correct candidate is not on the list of the three best decoded messages. The worst case signal space geometry that corresponds to the least gain is a tetrahedral packing. Here, every signal point is equidistant form the other three. The coding gain can then be evaluated as $10\log_{10}(G) = 10\log_{10}$ linebreak $(3/2)$ which is 1.76 dB.

General Result: We want to know the most likely event in which a GVA can output the globally L best candidates and not have the correct candidate on the list. For any L, the tightest packing of $L+1$ signal points in L dimensions, is the simplex [16], [17] where every signal point is equidistant, from each of the L points. Then, using the simplex geometry, and the result in [16, pp. 259-261], we get the worst case asymptotic gain for the GVA with L outputs over the VA as

$$10\log_{10}(G) = 10\log_{10}\left(\frac{2L}{L+1}\right) \,. \tag{16}$$

The gains with $L = 4$, 8, 16 are 2.04 dB, 2.50 dB, 2.75 dB respectively. For large values of L, the gain approaches 3 dB. The (small) loss due to the rate of the outer error detecting code is not taken into account in (16).

The worst case asymptotic gains presented here are somewhat optimistic for intermediate channel signal-to-noise ratios. The actual gain is often smaller when the number of set of L nearest neighbors is taken into account. What seems practical to achieve with a list size of $L = 2$ and 3 are gains of about 1.0 dB and 1.5 dB respectively as shown below by simulations.

Simulation Results: We have performed a series of simulations for rate $R = 1/2$, and $R = 8/10$ codes with memory $\nu = 4$. The generator matrix for the mother code, and the puncturing table for the $R = 8/10$ code is given in [13]. P_{BL} is the probability that the correct alternative is not among the L best produced by the LVA. The block error probabilities P_{BL}, $L = 1, 2$ and 3 have been simulated for the AWGN channel. Results in Figure 8 indicate that a gain of about 1 dB is gained for a $R = 1/2$ code with block size of 512 information bits when $L = 2$, and about 1.25 dB when $L = 3$. Similar results are obtained for the rate $R = 8/10$ code which is shown in Figure 8. The channel signal-to-noise ratio used in Figures 8 and 9 is E_c/N_0 where E_c is the energy per channel symbol. The energy per information bit E_b is given by $RE_b = E_c$ where R is the code rate.

3.2. Probability of Incorrect Decoding for the Rayleigh Fading Channel

We consider the decoding of coded data symbols subjected to independent Rayleigh fades from symbol to symbol and corrupted by additive white Gaussian noise. We assume a Rayleigh fading channel that permits coherent demodulation (ideal recovery of carrier phase) of binary PSK. Interleaving over many symbols (ideally infinite) is assumed to justify the assumption of independent fading from symbol to symbol. The received symbol r_k at time k is

$$r_k = a_k \cdot x_k \cdot \sqrt{E_c} + n_k \tag{17}$$

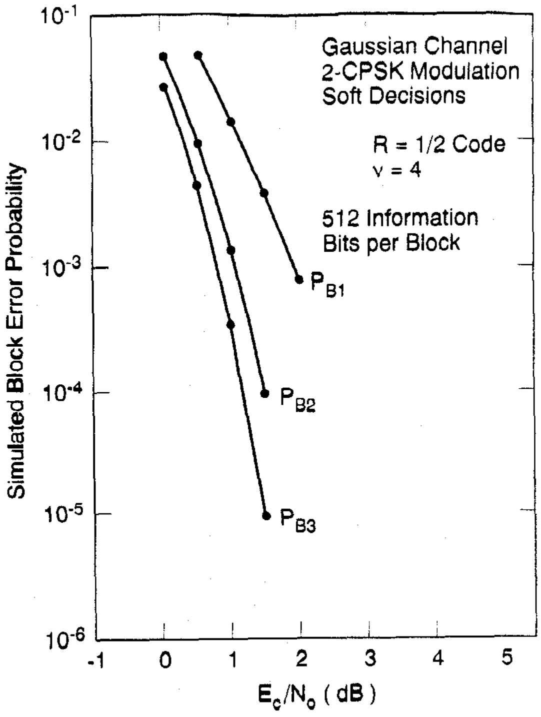

Fig. 8. Block error probability for the $L(\leq 3)$ best path LVA on the Gaussian channel. An inner $R = 1/2$, $\nu = 4$ code is used. The number of information bits per block is 512.

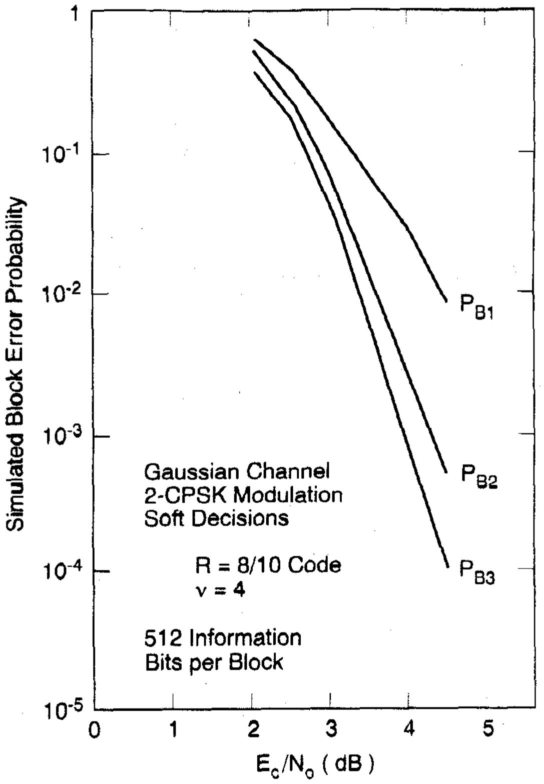

Fig. 9. Block error probability for $L(\leq 3)$ best path LVA on the Gaussian channel. An inner $R = 8/10$, $\nu = 4$ code is used. The number of information bits per block is 512.

where n_k is Gaussian variate with mean zero and variance N_0. x_k is the transmitted symbol taking on values ±1. a_k is Rayleigh distributed with probability density function

$$p_A(a_k) = \frac{a_k}{\sigma^2}\, e^{-a_k^2/2\sigma^2} \quad a_k \geq 0\,. \tag{18}$$

The instantaneous signal-to-noise ratio per received symbol γ is given by $\gamma = a_k^2 \cdot E_c/N_0$ which is distributed according to the pdf

$$pr(\gamma) = \frac{1}{\gamma} e^{-\gamma/\overline{\gamma}} \quad \gamma \geq 0\,. \tag{19}$$

$\overline{\gamma}$, the average signal-to-noise ratio per received symbol $(\overline{E}_c/N_0)$, is the mean of the random variable and is given by

$$\overline{\gamma} = \frac{E_c}{N_0} E\{a_k^2\} = \frac{E_c}{N_0} \cdot 2\sigma^2 \tag{20}$$

where $E\{\cdot\}$ denotes the expectation. $2\sigma^2$ is chosen to be 1 without loss of generality, so that the average SNR per received channel symbol, $\overline{E}_c/N_0$, is E_c/N_0. The performance of the decoder can be improved if a reliable estimate of the fade information (a_k's), also called the channel state information (CSI) is incorporated into the metric calculations [13].

$L = 1$: Provided the fade is independent from symbol to symbol, the asymptotic error probability with soft demodulation and Viterbi decoding is well approximated by

$$P_e \approx \text{const.} \left(\frac{1}{\overline{\text{SNR}}}\right)^D \tag{21}$$

where the time diversity $D = D_{free}$ (minimum Hamming distance of the code) and $\overline{\text{SNR}}$ is the average receiver channel signal-to-noise ratio.

$L = 2$: To find out the worst case when a transmitted codeword will not be on the list of the best two candidates from the decoder output, we will again consider three codewords $s(\underline{a})$, $s(\underline{b})$, and $s(\underline{c})$. However, instead of considering an ideal Rayleigh fading channel, we consider a simplified model where if the channel is in a fade, the received symbol is assumed to be an erasure. Otherwise, the received symbol is assumed to be demodulated perfectly. For this model, we can ask how many erasures can be made and have the correct candidate in the list of two with $L = 2$, LVA. Clearly when $L = 1$, $D_{free} - 1$ erasures can be made (leading to a diversity of $D = D_{free}$).

Each of the codewords $s(\underline{a})$, $s(\underline{b})$ and $s(\underline{c})$ differ in at least D_{free} positions from the other two. Let us assume that they differ in exactly D_{free} positions and that D_{free} is a even number. Without loss of generality, let the transmitted codeword $s(\underline{a})$ be the all zero codeword. Let $s(\underline{c})$ differ from $s(\underline{a})$ and $s(\underline{b})$ in $d(< D_{free})$ positions where the first two do not. Then, the minimum number of positions in which both $s(\underline{b})$ and $s(\underline{c})$ differ from $s(\underline{a})$ can be easily seen to be $\frac{3}{2}D_{free}$. Thus for $s(\underline{b})$ and $s(\underline{c})$ to be selected over $s(\underline{a})$, there should be at least $\frac{3}{2}D_{free} - 1$ erasures. The effective diversity is thus $(3/2)D_{free}$. When D_{free} is an odd number, the effective increase in diversity is at least $(D_{free} + 1)/2$. Note that an increase of D in (21) quickly yields significant gains in channel signal-to-noise ratio and in error probability. The largest relative gains occur for small values of D_{free}. This is confirmed by the simulations below for Rayleigh fading channel. To summarize, with list-of-2 decoding, the diversity is

$$D = \left\lceil \frac{3}{2} D_{free} \right\rceil . \qquad (22)$$

List Decoding With Arbitrary N_e Erasures: We obtain an upper bound to the list size in the presence of N_e erasures. Let $K(N, D_{free})$ be the maximum number of codewords in any binary code of length N and minimum Hamming distance D_{free}. Then, using the Hamming bound [18], the number of such codewords is upper bounded as

$$K(N, D_{free}) \leq \log_2 \left[\frac{2^N}{\sum_{i=0}^{e} \binom{N}{i}} \right] \qquad (23)$$

where $e = \lfloor \frac{D_{free}-1}{2} \rfloor$.

Let the number of erasures of a given binary code be N_e. Consider the worst case situation of all the codewords being identical except in the N_e erased positions. Since the minimum Hamming distance of the code is D_{free}, the number of such codewords is upper bounded by $K(N_e, D_{free})$. Thus an upper bound on the list size so that the correct candidate is on the list with N_e erasures is $K(N_e, D_{free})$.

Table 1 shows an upper bound (L^{**}) on the required list size to achieve a certain time Diversity (D) for $D_{free} = 3$. Also shown in the table is list size (L^*) as determined by the best known binary codes (from Appendix A of [18]) with $D_{free} = 3$ and code length equal to N_e.

Table 1

List Sized Needed With N_e Erasures for a Binary Code With $D_{free} = 3$

N_e	D	L^*	L^{**}
4	5	2	4
5	6	4	6
6	7	8	10
7	8	16	16
8	9	20	29
9	10	38	52
10	11	72	94
11	12	142	171

Simulation Results: We have simulated the performance of rate $R = 1/2$ and $8/10$ convolutional codes with memory $\nu = 4$ on a Rayleigh fading channel with BPSK modulation. The demodulation is assumed to be ideally coherent, and the fading is assumed to be independent from symbol to symbol. We have not used any channel state information (CSI) (the fade values) at the decoder and we refer to [13] for a detailed evaluation of the error performance with and without CSI. The metric used for decoding is the correlation metric (soft decision), the same as for the AWGN channel. In the simulations for the Rayleigh fading channel we have used binary phase shift keying, soft decisions and no CSI. The simulation results in Figure 10 for the $R = 1/2$ code, and in Figure 11 for the $R = 8/10$ code show a linear relationship between $\log P_{BL}$ and the $\overline{\text{SNR}}$. By calculating the slope, one can evaluate the diversity. It can be seen in Figure 10 that the diversity is about 5 when $L = 1$, and is about 9 when $L = 2$. Not that the diversity at large SNRs and with perfect CSI is 7 (value of D_{free}) when $L = 1$ and about 11 when $L = 2$ (based on the analysis above). These values are naturally lower for small SNRs and without CSI as the graphs indicate. Similar results are obtained for the rate $R = 8/10$ code. Note that the gain with the LVA ($L = 3$) is almost 4.5 dB in $\overline{E}_c/N_0$ over the VA ($L = 1$) case at $P_{BL} = 10^{-3}$.

3.3. Hybrid FEC/ARQ with RCPC and LVA Decoding

Combinations of forward error correction (FEC) and automatic repeat request (ARQ) systems are often referred to as hybrid ARQ schemes [12]. On very noisy channels like fading mobile radio channels, powerful FEC is needed. The most flexible and robust hybrid ARQ schemes uses inner rate compatible punctured convolutional (RCPC) codes. These codes can be decoded using soft Viterbi decoding. The main advantage of the RCPC codes over other FECs is their incremental redundancy transmission feature using the concept of rate compatible puncturing [13]. The hybrid FEC/ARQ scheme with RCPC codes and Viterbi decoding works as follows. A block of N_i data bits are augmented with N_c cyclic redundancy check (CRC) bits using an outer block code. For the purpose of analysis and simulations, we will assume that the probability of undetected error for this code is zero. ν known information bits for terminating the trellis are added to the $N_i + N_c$ bits before being encoded by the inner RCPC code. This block of $N_T = N_i + N_c + \nu$ bits are encoded by a family of RCPC codes with memory ν, and rates from 1 to $1/n$. In our examples, we will use a value of $n = 3$, and puncturing period $p = 8$. The possible rate are $p/(p + \lambda)$, $\lambda = 0, 1, 2, \ldots, (n-1)p$. The parameter λ is called the level of puncturing. The information bits are first encoded by the rate $1/n$ convolutional code. The output of the convolutional code is then punctured according to a rate compatible puncturing rule which for a given value of λ is in the form of a puncturing table $\mathbf{a}(\lambda)$ [13].

In a typical ARQ scenario, a subset of all possible λ is used, e.g., $\lambda = 1, 2, 4, 8$ and 16. The transmitter starts with the highest code rate possible, and will continue to transmit additional punctured bits corresponding to successively lower rate codes until it receives positive acknowledgement from the receiver. We assume an error free feedback channel. Two parameters are used to characterize the

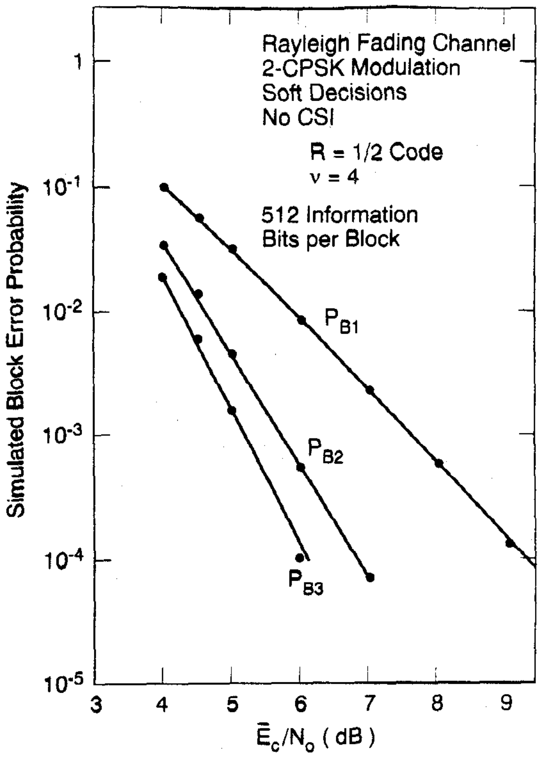

Fig. 10. Block error probability for $L(\leq 3)$ best path LVA on the fully interleaved Rayleigh fading channel. An inner $R = 1/2$, $\nu = 4$ code is used. The number of information bits per block is 512.

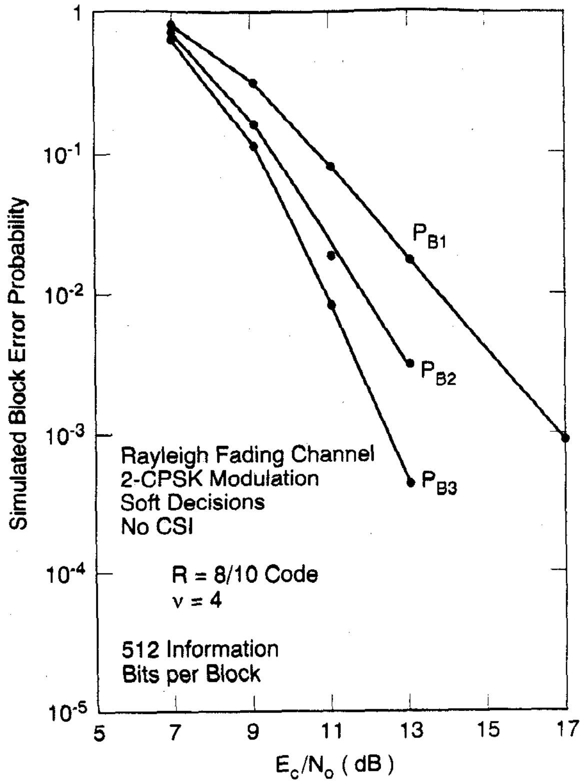

Fig. 11. Same as Figure 10 with a $R = 8/10$, $\nu = 4$ code.

performance of the hybrid FEC/ARQ system are the throughput S and the probability of a decoding failure P_F. The throughput S is the average number of received accepted bits over the average number of transmitted bits. There is a nonzero probability that correct decoding cannot be achieved at the lowest code rate. This is the probability of failure to decode, P_F. This quantity can be reduced by decreasing the code rate $1/n$ or through the process of code combining [13]. Both these methods will reduce the system throughput. We intend to use the LVA instead of the VA to reduce P_F and simultaneously increase the throughput (and implicitly decrease the overall delay).

We have simulated the complete hybrid FEC/ARQ system for the $\nu = 4$ RCPC codes with the LVA ($L = 3$). The puncturing period is $p = 8$. For reference, we have also simulated the same system using a conventional Viterbi decoder (VA). The results are shown in Figure 12 for the ideally interleaved Rayleigh fading channel with BPSK modulation and soft decision decoding. Ideal channel state information is also used. The RCPC codes chosen have $\lambda = 1$, 2, 4, 8, and 16 giving code rates of 8/9, 8/10, 8/12, 8/16, and 8/24 respectively. As an outer code, we chose an hypothetical block code with N_c of 34. The information block size N_i was chosen to be 382. The relative improvement of the throughput is about 10% over the signal-to-noise ratio range. This corresponds to about 1 dB in $\overline{E}_c/N_0$.

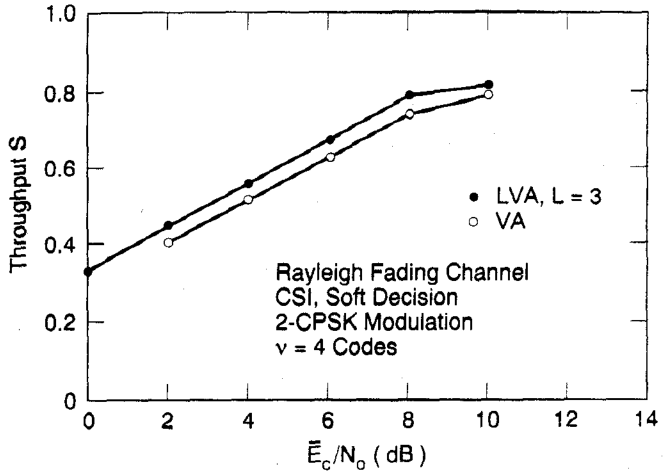

Fig. 12. Throughput versus average channel $\overline{\text{SNR}}$ $\overline{E}_c/N_0$ for a hybrid FEC/ARQ scheme with LVA and VA respectively.

Combined with the improvement in throughput comes the significant improvement in P_F. We did not explicitly simulate P_F for the $\nu = 4$, $R = 1/3$ code which is used as the final low rate code in the system in Figure 12. However this can be approximately estimated by comparing P_{B1} (VA) and P_{B3} (LVA, $L = 3$) in Figure 9 for the $\nu = 4$, $R = 1/2$ code. A gain in $\overline{E}_c/N_0$ of about 3 dB is achieved at $P_{BL} = 10^{-4}$.

4. Discussion and Conclusions

We have presented parallel and serial list Viterbi decoding algorithms [LVAs] that produce an ordered list of $L > 1$ globally best candidates after a trellis search. Novel methods for utilizing the LVA for concatenated communication systems are described.

We have analyzed and simulated the performance of the list decoding algorithm with multiple outputs for the Gaussian and the Rayleigh fading channels. With supporting theory, we find that the algorithm for the AWGN channel can yield 1.0-2.0 dB in practice. For the Rayleigh fading channel, list decoding results in an increase in the effective time diversity of the code. This in turn corresponds to an increasing coding gain with increasing $\overline{\text{SNR}}$, a typical characteristic of increased diversity. Gains of 3-4.5 dB are demonstrated at block error probabilities of about 10^{-4}.

We have demonstrated that the LVA improves *both* the throughput and the probability of failure to decode for hybrid FEC/ARQ schemes. The price paid is increased signal processing at the receiver. It is interesting to note that the use of the LVA is optional. One receiver may operate with VA while another may use $L = 2$, and yet another with $L = 8$ etc. It is also not necessary to use the LVA in every stage of decoding in the hybrid FEC/ARQ scheme that uses RCPC codes. If one were interested in solely reducing the probability of failure to decode, then the LVA needs to be used only in the last stage of decoding.

While the LVA has been applied to the decoding of convolutional codes in this paper, we also expect similar performance improvements when it is used for the decoding of combined coding and modulation schemes [19,20,21]. The analysis we have presented for the AWGN channel is directly applicable to these schemes. More work needs to be done in the presence of Rayleigh fading.

List decoding has also been applied to the problem of joint data and channel estimation [22]. Other applications include speech recognition [15,23] and combined speech and channel decoding [14,24].

References

[1] A. J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," *IEEE Transactions on Information Theory*, Vol. IT-13, pp. 260-269, 1967.

[2] A. J. Viterbi and J. K. Omura, *Principles of Digital Communication and Coding*, McGraw-Hill, NY, 1979.

[3] G. D. Forney, Jr., "Convolutional Codes II: Maximum Likelihood Decoding," *Information Control*, 25, pp. 222-266, July 1974.

[4] G. D. Forney, Jr., "Convolutional Codes III: Sequential Decoding," *Information Control*, 25, pp. 267-297, July 1974.

[5] H. Yamamoto and K. Itoh, "Viterbi Decoding Algorithm for Convolutional Codes with Repeat Request," *IEEE Transactions on Information Theory*, IT-26, pp. 540-547, September 1980.

[6] T. Hashimoto, "A List-Type Reduced-Constraint Generalization of the Viterbi Algorithm," *IEEE Transactions on Information Theory*, IT-33, pp. 866-876, November 1987.

[7] J. B. Anderson and S. Mohan, "Sequential Coding Algorithms: A Survey and Cost Analysis," *IEEE Transactions on Communications*, Vol. COM-32, pp. 169-176, Feb. 1984.

[8] R. H. Deng and D. J. Costello, Jr., "High Rate Concatenated Coding Systems Using Bandwidth Efficient Trellis Inner Codes," *IEEE Transactions on Communications*, Vol. 37, No. 5, pp. 420-427, May 1989.

[9] P. Elias, "Error-Correcting Codes for List Decoding," *IEEE Transactions on Information Theory*, Vol. 37, No. 1. pp. 5-12, January 1991.

[10] J. Hagenauer and P. Hoeher, "A Viterbi Algorithm with Soft-Decision Outputs and its Applications," *GLOBECOM '89*, Dallas, Texas, Nov. 1989, Conference Record pp. 1680-1686.

[11] P. Hoeher, "TCM on Frequency-Selective Fading Channels: A Comparison of Soft-Output Viterbi-Like Equalizers," *GLOBECOM '90*, December 3-5, San Diego. Conference Record, pp. 376-381.

[12] S. Lin and D. J. Costello, Jr., *Error Control Coding-Fundamentals and Applications*, Prentice-Hall, Inc., NJ, 1983.

[13] J. Hagenauer, "Rate Compatible Punctured Convolutional Codes (RCPC codes) and their Applications," *IEEE Transactions on Communications*, Vol. COM-36, pp. 389-400, April 1988.

[14] N. Seshadri and C-E. W. Sundberg, "Generalized Viterbi Algorithms for Error Detection with Convolutional Codes," *GLOBECOM '89*, Dallas, Texas, Nov. 1989, Conference Record pp. 1534-1538.

[15] F. K. Soong and E. F. Huang, "A Tree-Trellis Based Fast Search Algorithm for Finding the *N* Best Sentence Hypotheses in Continuous Speech Recognition," *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, 1991, pp. 705-708.

[16] J. M. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering*, John Wiley & Sons, NY, 1965.

[17] C. L. Weber, *Elements of Detection and Signal Design*, Springer Verlag, New York, 1987.

[18] F. J. MacWilliams and N.J.A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland Publishing Company, 1978.

[19] G. Ungerboeck, "Channel Coding with Multilevel/Phase Signals," *IEEE Transactions on Information Theory*, Vol. IT-28, No. 1, pp. 55-67, Jan. 1982.

[20] J. B. Anderson, T. Aulin and C-E. W. Sundberg, *Digital Phase Modulation*, Plenum Press, NY, 1986.

[21] N. Seshadri and C-E. W. Sundberg, "Multi-level Codes with Large Time Diversity for the Rayleigh Fading Channel," *IEEE Transactions on Communications*, Vol. 41, No. 9, pp. 1300-1310, September 1993.

[22] N. Seshadri, "Joint Data and Channel Estimation Using Fast Blind Trellis Search Techniques," *GLOBECOM '90*, December 3-5, San Diego. Conference Record, pp. 1659-1663, also to appear in IEEE Transactions on Communications, March 1994.

[23] C. H. Lee and L. R. Rabiner, "A Network-Based Frame Synchronous Level Building Algorithm for Connected Word Recognition," *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, 1988, pp. 410-413.

[24] W. C. Wong, N. Seshadri and C-E. W. Sundberg, "Estimation of Unreliable Packets in Sub-band Coding of Speech," *Proceedings of the IEE*, Part I, Vol. 138, No. 1, pp. 43-49, February 1991.

Nambirajan Seshadri (S'81, M'87) was born in India in 1961. He received the B.E. degree in Electronics and Communication from the University of Madras, India, in 1982, and the M.S. and Ph.D. degrees in Computer and Systems Engineering from the Rensselaer Polytechnic Institute, Troy, NY, in 1984 and 1986 respectively. Since November 1986, he has been a Member of Technical Staff at AT&T Bell Laboratories, Murray Hill, NJ in the Signal Processing Research Department. His research interests are in the general areas of communications and signal processing. He is the coauthor of *Blind Deconvolution* (Prentice Hall, NY, 1994). He holds two patents in the area of data communications.

Carl-Erik W. Sundberg (S'69 – M'75 – SM'81 – F'90) was born in Kariskrona, Sweden on July 7, 1943. He received the M.S.E.E. and Dr. Techn. degrees from the Lund Institute of Technology, University of Lund, Sweden, in 1966 and 1975, respectively. Currently he is a Distinguished Member of the Technical Staff at AT&T Bell Laboratories, Signal Processing Research Department, Murray Hill, NJ. Before 1976 he held various teaching and research positions at the University of Lund. During 1976, he was with the European Space Research and Technology Centre (ESTEC), Noordwijk, The Netherlands, as an ESA Research Fellow. From 1977 to 1984 he was a Research Professor (Docent) in the Department of Telecommunication Theory, University of Lund. He has held positions as Consulting Scientist at LM Ericsson, SAAB-SCANIA, Sweden, and at Bell Laboratories, Holmdel. His consulting company, SUNCOM, has been involved in studies of error control methods and modulation techniques for the Swedish Defense, a number of private companies and international organizations. His research interests include source coding, channel coding, digital modulation methods, fault-tolerant systems, digital mobile radio systems, spread-spectrum systems, digital satellite communications systems, and optical communications. He has published over 80 journal papers and contributed over 110 conference papers. He holds 13 US, Swedish and international patents. He is coauthor of *Digital Phase Modulation*, (New York: Plenum, 1986) and *Topics in Coding Theory*, (New York: Springer-Verlag 1989). Dr. Sundberg has been a member of the IEEE European-African-Middle East Committee (EAMEC) of COMSOC from 1977 to 1984. He is a member of COMSOC Communication Theory Committee and Data Communications Committee. He has also been a member of the Technical Program Committees for the International Symposium on Information Theory, St. Jovite, Canada, October 1983, for the International Conference on Communications, ICC'84, Amsterdam, The Netherlands, May 1984 and for the 5th Tirrenia International Workshop on Digital Communications, Tirrenia, Italy, September 1991. He has organized and chaired sessions at a number of international meetings. He has been a member of the International Advisory Committee for ICCS'88, ICCS'90 and ICCS'92 (Singapore). He served as Guest Editor for the IEEE Journal on Selected Areas in Communications in 1988-1989. He is a member of SER (Svenska Elektroingenjörening) and the Swedish URSI Committee (Svenska Nationalkommitten för Radiovetenskap). In 1986 he and his coauthor received the IEEE Vehicular Technology Society's Paper of the Year Award and in 1989 he and his coauthors were awarded the Marconi Premium Proc. IEE Best Paper Award.