MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

A Trellis-based Technique for Blind Channel Estimation and Equalization

Cao, L; Chen, C.W.

TR2003-131   December 2003

Abstract

In this paper, we present a trellis-based blind channel estimation and equalization technique coupling two kinds of adaptive Viterbi algorithms. First, the initial blind channel estimation is accomplished by incorporating the list parallel Viterbi algorithm with the least mean square (LMS) updating approach. In this operation, multiple trellis mappings are preserved simultaneously and ranked in terms of path metrics. Equivalently, multiple channel estimates are maintained and updated once a single symbol is received. Second, the best channel estimate from the above operation will be adopted to set up the whole trellis. The conventional adaptive Viterbi algorithm is then applied to detect the signal and further update the channel estimate alternately. A small delay is introduced for the symbol detection and the decision feedback to smooth the noise impact. An automatic switch between the above two operations is also proposed by exploiting the evolution of path metrics and the linear constraint inherent in the trellis mapping. Simulation has shown an overall excellent performance of the proposed scheme in terms of mean square error (MSE) for channel estimation, robustness to the initial channel guess, computational complexity, and channel equalization.

Publication History:

1. First printing, TR-2003-131, December 2003

A Trellis-based Technique for Blind Channel Estimation and Equalization

Lei Cao, Chang Wen Chen, Philip Orlik, Jinyun Zhang, and Daqing Gu

Abstract: **In this paper, we present a trellis-based blind channel estimation and equalization technique coupling two kinds of adaptive Viterbi algorithms. First, the initial blind channel estimation is accomplished by incorporating the list parallel Viterbi algorithm with the least mean square (LMS) updating approach. In this operation, multiple trellis mappings are preserved simultaneously and ranked in terms of path metrics. Equivalently, multiple channel estimates are maintained and updated once a single symbol is received. Second, the best channel estimate from the above operation will be adopted to set up the whole trellis. The conventional adaptive Viterbi algorithm is then applied to detect the signal and further update the channel estimate alternately. A small delay is introduced for the symbol detection and the decision feedback to smooth the noise impact. An automatic switch between the above two operations is also proposed by exploiting the evolution of path metrics and the linear constraint inherent in the trellis mapping. Simulation has shown an overall excellent performance of the proposed scheme in terms of mean square error (MSE) for channel estimation, robustness to the initial channel guess, computational complexity, and channel equalization.**

Index Terms: **Adaptive Viterbi algorithms, blind channel estimation and equalization, least mean square (LMS) updating.**

I. INTRODUCTION

Channel estimation and channel equalization in communications are two highly related technologies that are often considered jointly. For a finite discrete channel model $h(k), k = 0, 1, \ldots, L$, given the transmitted symbol sequence $x(n), n = 1, 2, \ldots, N$, where each symbol is an element of an alphabet set with size M, the received sequence $y(n), n = 1, 2, \ldots, N$ with inter-symbol interference (ISI) is

$$y(n) = \sum_{k=0}^{L} h(k)x(n-k) + v(n), \quad (1)$$

where $v(n)$ is the i.i.d. additive Gaussian noise with zero mean and variance σ^2. For a block of N received symbols, in which the channel response is assumed time-invariant, the probability density function of $y(n)$, conditioned on knowledge of $h(n)$ and $x(n), n = 1, \ldots, N$, is

$$P(\vec{y}|\vec{x}, \vec{h}) = \frac{1}{(2\pi\sigma^2)^{\frac{N}{2}}} \exp\Bigg(-\frac{1}{2\sigma^2}\sum_{n=1}^{N} |y(n) - \sum_{k=0}^{L} h(k)x(n-k)|^2\Bigg). \quad (2)$$

The problem of joint channel estimation and equalization now becomes an optimization problem. Essentially we want to find the estimates of the channel parameters $\vec{h}^*$ and the transmitted data $\vec{x}^*$ so that the above probability is maximized, i.e.,

$$(\vec{x}^*, \vec{h}^*) = \arg\max_{\vec{x},\vec{h}} P(\vec{y}|\vec{x}, \vec{h}) = \arg\min_{\vec{x},\vec{h}} \sum_{n=1}^{N} |y(n) - \sum_{k=0}^{L} h(k)x(n-k)|^2 \quad (3)$$

where arg max (or arg min) is the argument to maximize (or minimize) the following expression. When the input signal $\vec{x}$ is known, we may differentiate equation (3) over $\vec{h}$ and set it equal to zero to generate a set of linear equations. The maximum likelihood channel estimate can then be obtained by solving these linear equations. On the other hand, when the channel impulse response is known, the maximum likelihood sequence estimation (MLSE) of the source symbols can be accomplished by using Viterbi algorithm. However, difficulty arises when neither $\vec{h}$ nor $\vec{x}$ is known. This is the problem of blind channel estimation and equalization with only the knowledge of received observations $\vec{y}$. A direct approach is that for each possible source data sequence $\vec{x}$ we determine the maximum likelihood estimate of $\vec{h}$, and then, we select the data sequence $\vec{x}^*$ and the corresponding $\vec{h}^*$ so that equation (3) is satisfied. Unfortunately, the number of possible data sequences in this exhaustive search is huge and equals M^N. Therefore, this method is prohibitive due to its extreme complexity.

Another way to solve the problem is to use Hidden Markov Model (HMM). With this model, the Baum-Welch algorithm [1] can be applied to iteratively estimate $[\vec{h}, \sigma^2]$. This approach is actually an EM (Expectation-Maximum) algorithm that is implemented alternately between the calculation of a likelihood function $P(\vec{y}|\vec{h}, \sigma^2)$ and the maximization of it over the target variables $[\vec{h}, \sigma^2]$. It was proved [2] that the likelihood function is locally maximized as the iteration number increases. In [3], the trellis branch values rather than $\vec{h}$ itself are updated in each iteration to reduce the computational complexity in solving linear equations. In general, the Baum-Welch algorithm has very high computational complexity. Therefore, many researches have been focused on the Viterbi algorithm-based joint

Manuscript received September, 2002; approved for publication by Marco Lops, Division I Editor, October, 2003.

L. Cao is with the Department of Electrical Engineering, University of Mississippi, University, MS 38677, USA, e-mail: lcao@olemiss.edu.

C. W. Chen is with the Department of Electrical and Computer Engineering, Florida Institute of Technology, Melbourne, FL 32901, USA, e-mail: cchen@fit.edu.

P. Orlik, J. Zhang and D. Gu are with Mitsubishi Electric Research Laboratories, Cambridge, MA 02139, USA, e-mails: {porlik,jzhang,dgu}@merl.com

channel estimation and signal detection. A practical method to use Viterbi algorithm is illustrated in Figure 1. This kind of schemes generally includes three functionalities: channel estimation, Viterbi signal detection and channel updating. In typical non-blind schemes, channel estimation is usually completed through a training sequence. For blind channel estimation and signal detection, iterative algorithms proposed in [4] and [5] operate alternately between channel estimation, assuming that the symbols are known, and a Viterbi sequence detection, assuming the channel parameters are known. However, these algorithms are very sensitive to the initial channel guess [4]. In [6], a list parallel adaptive Viterbi algorithm (LPA-VA) was proposed to maintain multiple survivors for each state at any time instant along trellis diagram so that the multiple corresponding channel estimates are then maintained and updated individually based on LMS. A quantized channel estimation approach is also proposed in [7]. While these techniques have advantages in parallel structure and robustness to the initial channel guess, they have a major disadvantage of high complexity either by always using the parallel list Viterbi algorithm [6] or by simultaneously operating a set of Viterbi decoding algorithms [7].

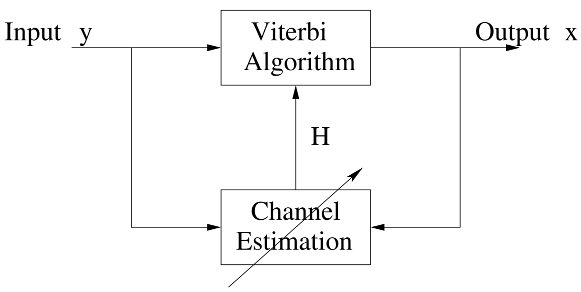

Fig. 1. ML based channel estimation and signal detection. Three functions include initial channel estimation, signal detection, decision feedback for channel updating.

In this paper, we propose a trellis based technique for blind joint channel estimation and signal detection. It is actually a two-mode process coupling two kinds of adaptive Viterbi algorithms. We first take advantage of the scheme proposed in [6] to achieve the robustness to the initial channel guess. Then, we propose to switch to a conventional adaptive Viterbi algorithm for signal detection and grained channel update. Noise impact is smoothed by introduction of a small delay in the signal detection and decision feedback. The switch between the two modes is designed by exploiting the linear constraint in the trellis mapping and the evolution of path metrics. This technique not only preserves the benefits of scheme of [6] such as robustness to initial channel guess, but also provides some other significant advantages. For example, the computational complexity and required memory resources are greatly reduced; the mean square error (MSE) of the channel estimates is decreased as well. The rest of the paper is organized as follows. Section 2 addresses the proposed technique in detail. We first describe the trellis mapping from the received symbols to the trellis branches, and derive the total number of mappings. Then, we discuss the list parallel adaptive VA and the conventional adaptive VA (CA-VA) for the operation of initial channel estimation and the operation of signal detection and channel update, respectively. The switch design between the two operations is also proposed. Section 3 presents the simulation results. Finally, Section 4 concludes this paper.

II. SYSTEM DESCRIPTION

The whole system couples two operations of Viterbi algorithms to complete different functions in channel estimation and equalization. First, the list parallel Viterbi algorithm with LMS adaptation is used to obtain the initial channel estimate. Second, the conventional adaptive Viterbi algorithm is employed to detect the symbols and finely update the channel estimate through feedback. In this section, we will introduce the mapping between the received symbols and the trellis structure determined by the channel impulse response, and the linear constraint in the mapping. We then describe LPA-VA and CA-VA, respectively. We also present the switch design between the two operations.

A. Trellis Mapping and Linear Constraint

For a binary signal $M = 2$ ($x(n) = 1$ or $-1, n = 1, \ldots, N$), and a channel impulse response $h(k), k = 0, \ldots, L$, the received symbols are the convolution of the both. Acting exactly the same as the code generators in convolutional codes, the channel impulse response determines the trellis structure. The length $(L+1)$ of the channel response determines the total number of states in the trellis structure to be 2^L, while the amplitudes of the each $h(k), k = 0, \ldots, L$ determine the branch values of the trellis. Figure 2 is an example of trellis structure for $M = 2$ and $L = 2$ (channel impulse response is thus $\{h_0, h_1, h_2\}$). As a result, in order to find the channel impulse response, we need to map the received symbols to the trellis branches.

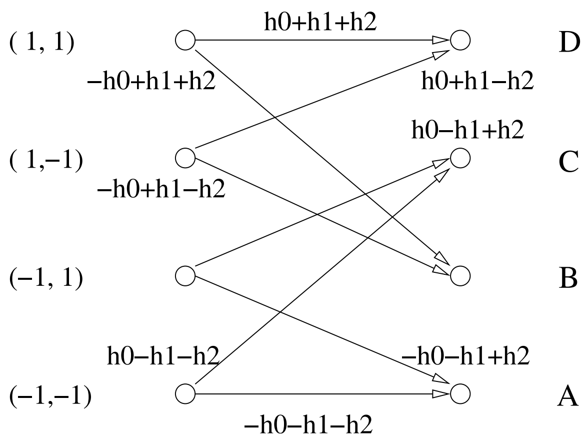

Fig. 2. The trellis for M=2 and L=2. Branch values are determined by the channel response.

In the absence of noise, the maximum number of different symbols received is 2^{L+1}, which is the same as the number of branches in the trellis structure. We also assume differential coding at the transmitter. The total number of trellis mappings that map these symbols to the branches can be formulated as

$$N_{trellis} = 2^L(2^{L+1} - 2)(2^{L+1} - 4) \cdots (2^{L+1} - 2L)$$

$$= \frac{2^L(2^L)!}{(2^L-L-1)!}. \quad (4)$$

This equation can be interpreted as follows. Suppose the received channel symbols are $\{c_1, c_2, \ldots, c_{2^L}, -c_{2^L}, \ldots, -c_2, -c_1\}$. The symbol c_1 can be mapped to one of 2^L branches (rather than 2^{L+1} because of differential coding). Then, $-c_1$ is assigned automatically due to the symmetry of the trellis branches. Therefore, symbol c_2 can be mapped to one of the following $2^{L+1}-2$ branches. Consequently, $-c_2$ will be assigned to the branch symmetric to the one with c_2. Similarly, c_3 will be mapped to one of $2^{L+1}-4$ branches. This process may continue to the final symbol. However, as long as symbols up to c_{L+1} are allocated, we are able to choose $L+1$ branches from the totally mapped $2(L+1)$ branches to set up $L+1$ linear equations. From these linear equations the $L+1$ components of the channel impulse response $\vec{h}$ can be solved uniquely. Consequently, the remaining symbols, $\{\pm c_{L+2}, \ldots, \pm c_{2^L}\}$, will be automatically set to their corresponding branches determined by the solved $\vec{h}$. This feature is the linear constraint in the trellis mapping. Notice that it has not been considered in [6] and the total number of trellis mappings represented by equation (10) in [6] is

$$N_{tre} = 2^L(2^{L+1}-2)(2^{L+1}-4)\cdots 4\cdot 2\cdot 1 = 2^{2^L}(2^L)!/2. \quad (5)$$

This value is much higher than the actual number considering the linear constraint. From equations (4) and (5), it is easy to find that $N_{tre}/N_{trellis} = 2^{2^L-L-1}(2^L-L-1)!$. A simple example to demonstrate the trellis mapping and linear constraint can be shown by Figure 2, where up to 8 different symbols $c_1, c_2, c_3, c_4, -c_4, -c_3, -c_2, -c_1$ will be received assuming a noise free channel. If c_1 is allocated to branch AC, $-c_1$ will be automatically assigned to branch DB due to the linear constraint. After we have mapped c_2 and c_3 to branches such as BA, CB, respectively, we can choose three linearly independent branches, such as AC, BA, and CB, so that h_0, h_1, and h_2 will be solved completely. Consequently, the c_4 and $-c_4$ will be automatically assigned according to the solved channel impulse response. Therefore the linear constraint significantly reduces the number of trellis mappings that need to be considered. We will show later that the linear constraint can also be exploited to design the switch between the operation of initial channel estimation and the operation of signal detection and further channel updating.

It should be pointed out that the assumption of differential coding in the beginning is not necessary. It only affects the allocation of the first symbol in trellis mapping. Without this assumption, the actual number of trellis mapping will be doubled, i.e., c_1 can be mapped to any one of 2^{L+1} branches rather than 2^L branches. As a result, the estimated channel and the detected data sequence will be either the original versions or the versions only with different signs. It also means for the blind channel estimation and signal detection without any knowledge of the transmitted sequence, we can not tell the difference between the true results and their "mirror" versions, because $X(z)H(z)$ and $(-X(z))(-H(z))$ produce the same output $Y(z)$.

Clearly, the number of mappings in equation (4) still grows large as L increases. It is impossible to compare all these mappings in order to find the best one. However, it is practical to consider a subset of the most appropriate mappings at the same time. This suboptimal method can be realized by incorporating the list parallel Viterbi algorithm with the LMS updating method [6]. Simulation results in [6] show that a good channel estimate occurs within the first 50 symbols for three considered channels in [8] (pp. 631) when SNR > 10dB.

B. List Parallel Adaptive Viterbi Algorithm

In order to maintain multiple trellis mappings (corresponding to multiple channel estimates) at the same time, the list Viterbi algorithm is adopted to retain the K best candidates for each state at each time instant. The corresponding channel estimates are updated whenever a single symbol has been received and detected according to the path metrics. As shown in Figure 2, from any time instant t to $t+1$, we will compare the received symbol $y(t)$ with each branch value and compute a set of path metrics for each state at $t+1$. Channel estimates will also be immediately updated for each state at time $t+1$, $t = 1, \ldots, N_1$, where $N_1(< N)$ is the number of stages for initial channel estimation. The whole process may be exemplified by considering operations at state C in Figure 2. Note that state C has 2 predecessors: states A and B. Let $M^t_{B,i}$ (or $M^t_{A,j}$) be the i^{th} (or j^{th}), $i, j = 1, \ldots, K$, smallest path metric in state B (or A) at time instant t. The corresponding channel estimate in state B (or A) at time t are $\vec{h}^t_{B,i}$ (or $\vec{h}^t_{A,i}$). At the next time instant $t+1$, for state C, we calculate its path metrics from both states B and A as

$$M^{t+1}_{CB,i} = M^t_{B,i} + \text{dist}\{\vec{h}^t_{B,i}(1,-1,1)', y(t)\} \quad (6)$$

$$M^{t+1}_{CA,j} = M^t_{A,j} + \text{dist}\{\vec{h}^t_{A,j}(1,-1,-1)', y(t)\} \quad (7)$$

where $i, j = 1, 2, \ldots, K$, $(\cdot)'$ is the transpose of vector $(\cdot)$, and dist$\{\cdot,\cdot\}$ is the distance between the two elements. Note that the last two digits in (1,-1,1) (or (1,-1,-1)) in the above equations represent the state B (or A), while the first digit "1" is the input bit that leads state B (or A) to state C. Then, the K smallest path metrics $M^{t+1}_{C,l}, l = 1, \ldots, K$ for state C at time $t+1$ will be selected from the union of $M^{t+1}_{CB,i}$ and $M^{t+1}_{CA,i}$

$$M^{t+1}_{C,l} = \text{arg } l^{th} \text{ min } \{M^{t+1}_{CB,i}, M^{t+1}_{CA,j}\}, \quad i, j = 1, \ldots, K, \quad (8)$$

where arg l^{th} min means to select the l^{th} minimum value from the following set. After the K minimum path metrics for each state are obtained at time instant $t+1$, the corresponding channel estimates can be updated. For example, if the r_1^{th} smallest path metric of state C at $t+1$ is calculated from the r_2^{th} smallest path metric in state B at t, we say that the r_1^{th} best channel estimate of C at $t+1$, $\vec{h}^{t+1}_{C,r_1}$, is updated from the r_2^{th} best channel estimate of B at t, $\vec{h}^t_{B,r_2}$. The update is computed using least mean square (LMS) method as

$$\vec{h}^{t+1}_{C,r_1} = \vec{h}^t_{B,r_2} + \Delta\cdot\text{dist}\{\vec{h}^t_{B,r_2}(1,-1,1)', y(t)\}(1,-1,1), \quad (9)$$

where Δ is the step size used in LMS.

C. Conventional Adaptive Viterbi Algorithm (VA)

After a good initial channel estimate is obtained at a certain time instant, continuously maintaining multiple survivors per state will consume more memories and computational resources than a conventional adaptive VA. Moreover, this more complicated algorithm does not guarantee the best convergence performance, though it generally finds a roughly good channel estimate very quickly. As stated in last section, at any time instant, we first compute K minimum path metrics for each state (equation (6)-(8)). Then, the updating of the channels is conducted immediately (equation (9)). The best channel estimate at a time instant is selected to be the one corresponding to the minimum path metric among all states. This LPA-VA operation has two direct consequences.

First, suppose that at time t the best channel is $\vec{h}^t$ corresponding to state A. When a symbol with additive noise is received, according to equation (6)-(8) and due to the noise and previous channel estimates, the smallest path metric at time $t+1$ may be in any state, assuming in state B. Therefore, the best channel estimate $\vec{h}^{t+1}$ at time $t+1$ is evolved from a channel estimate in state C or D but not the best one $\vec{h}^t$ in A at time t. This feature is very helpful in the beginning for channel estimation when we do not have any knowledge of the channel and the initial channel vector is generally set to $\vec{0}$. However, when a good channel estimate is obtained, this effect will divert the correct channel updating direction. That is, the best channel estimate in each time stage may not be updated from its own previous version, but from a previous version of another different channel estimate. Second, in the Viterbi algorithm for signal detection, even though the trellis is fixed without updating, a decoding depth is needed to reduce the truncation error that occurs when the information bits are released before the entire codeword has been processed. Forney [9] has shown that the probability of truncation error decreases exponentially with the decoding depth. At low SNR, the error is negligible if the decoding depth is 5 times more than the number of memories that equals the channel length L here. When using LPA-VA for signal decision, however, signals are detected immediately without any delay (zero coding depth) after one symbol is received. Therefore, any error in signal decision may divert the update of the channel estimate. This, in turn, further affects the following symbol detection. As a result, the list parallel adaptive VA that keeps a number of channel candidates and uses the zero decoding delay is appropriate for initial channel estimation but not for the signal detection and channel further updating when a good channel estimate has been achieved.

Therefore, in this research, we propose the use of a conventional adaptive Viterbi algorithm to alternately detect the signal and further update the channel estimate. When a good channel estimate is obtained, it is adopted to set up the whole trellis. Signal decision and channel updating are all started from this channel estimate. The path metric calculation is still the same with equations (6) (7) except that $K = 1$ and all the channel estimates are the same in this case. However, before the signal detection and decision feedback, a delay of small number of symbols is introduced. The delay here has the effect to smooth the noise in several delayed symbols so that the impact of a single high value noise is alleviated. As shown later in simulation, both better MSE convergence and less computational complexity are achieved with this method.

D. Switching Design Between Two Operations

As described above, when a good channel estimate is obtained from the list parallel adaptive VA, the operation should be switched to the conventional adaptive VA to finely update the channel estimate and detect the symbols. Each state sequence along the trellis actually represents one input bit sequence. The associated trellis path (branch sequence) is the convolution of the input sequence with channel impulse response. The blind channel estimation and equalization is to find the closest trellis path matched to the received symbols (including channel noise) with the minimum distance.

The linear constraint stated before for the mapping of channel symbols to trellis branches not only reduces the number of mappings to achieve good channel estimation with a short length of received symbols, but also provides a potential technique to design the switch between the two operations. As shown in Figure 2 with 8 different channel symbols $\{(c_1, -c_1), (c_2, , -c_2), (c_3, -c_3), (c_4, -c_4)\}$ in the absence of noise, a received symbol sequence with as few as 5 symbols may be enough to complete the channel estimation. For example, if the received symbol sequence is: $c_1, c_2, c_3, c_4, c_4, \cdots$, then the state sequence along the trellis must be $A \rightarrow C \rightarrow B \rightarrow A \rightarrow A \rightarrow \cdots$. This bit state sequence corresponds to the input bit sequence of $1 \rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow \cdots$. Any other sequence will violate the linear constraint in the trellis mapping. Therefore, by choosing three branches along the path to constitute three linear independent equations, the channel estimate $h(k), k = 0, 1, 2$ can be easily solved. For non-differential coding, there is another symmetrical state sequence $D \rightarrow B \rightarrow C \rightarrow D \rightarrow D \rightarrow \cdots$ corresponding to a channel estimate. This channel estimate is the same as above except each channel tap has its sign reversed.

Apparently, the best channel estimate is the one corresponding to a state sequence that has the smallest path metric. Furthermore, this state sequence must be along a valid trellis path due to the linear constraint. Otherwise, the best channel estimate at a certain time stage is not evolved from the previous version of itself. This characteristic can be exploited to decide whether a good channel estimate has been obtained. In the operation of LPA-VA, the total number of channel estimates is equal to the multiplication of the number of trellis states and the number of survivors per state. From the initial channel guess $\vec{0}$, these channel estimates are updated in different directions. From any time stage t to $t+1, t = 1, \ldots, N$, we can find a branch that causes a state at time $t+1$ having the smallest path metric. Since the path metrics are calculated from different channels, in the beginning steps of channel estimation, these branches are not likely to form a valid path, which means any two consecutive such branches may not have a common node (state). But as the algorithm continues, one good channel estimate will stand out. As such, the states with the smallest path metrics caused by this channel estimate will construct a single valid path. As a result, when the observed number of consecutive connected best states is larger than a pre-defined threshold, we know a roughly good channel estimate has been obtained and it is the time to switch the operation to the second mode. Actually when the smallest path metric stands out in LPA-VA, it will eventually dominate the following path metric calculation and hence affect the updating of other channel estimates. Eventually, after a certain steps many channel estimates will converge to a small neighborhood of the best one. This feature may also supplement the design of a switch.

It needs to point out that the switch based on the linear constraint relies only on heuristic considerations. As consequence, it is not guaranteed to take place always. A very small number of cases that it does not happen have been found in low SNR conditions. As a result, in a practical system, this "soft" switch determined by the linear constraint should be coupled with a "hard" switch that sets to a certain maximum number of processed symbols.

III. SIMULATION RESULTS

In this simulation, we use 2-PAM signal and test the proposed approach over three channels proposed in [8] (page 631). These channels are: a={-0.2, -0.5, 0.7, 0.36, 0.2}, b={0.407, 0.815, 0.407}, and c={0.227, 0.460, 0.688, 0.460, 0.227}. Channel b and c exhibits only amplitude distortion while channel a has both amplitude and phase distortion. Channel b has a spectral null at the band edge while channel c has an in-band null. The MSE of channel estimation is calculated as

$$MSE(t) = \frac{1}{Q}\sum_{i=1}^{Q}\{\frac{1}{L+1}\sum_{k=0}^{L}[\hat{h_i}^t(k) - h_{org}(k)]^2\}, t = 1, \cdots, N \quad (10)$$

where Q is the number of data blocks tested to get the average MSE value. N is the number of bits in each block. $L+1$ is the number of taps of channel impulse response. $\hat{h_i}^t(k)$ is the value of the k^{th} tap in the estimated channel response at the t^{th} time stage in the i^{th} data block. $h_{org}(k)$ is the actual channel the signal goes through. SNR is calculated as $10\log\{\sum_{k=0}^{L}|h(k)|^2/\sigma^2\}$, where σ^2 is the variance of i.i.d. white Gaussian noise. We set each block to have 200 symbols ($t \le 200$). A step size of 0.1 is used in the LMS adaptive method in both Viterbi algorithms.

First, for channel b we compare the LPA-VA having 4 survivors per state with the CA-VA for channel updating. Delays tested in CA-VA are 0 and 10 symbols, respectively. SNR is set to 13 dB. An initial channel estimate is assumed as $\{0.577, 0.577, 0.577\}$. Figure 3 shows the results as the average of 100 trials. Both algorithms converge in terms of channel estimation MSE. Since the LA-VA retains 4 channel estimates for each state, it shows the fastest initial channel estimation. The CA-VA with 0 delay makes a signal decision and feeds it back to update the channel estimate immediately after the current symbol is received. This method is especially vulnerable to isolated high noise and shows the worst performance in both initial channel estimation and final channel updating. The CA-VA with a delay of 10 symbols has achieved the lowest MSE in final channel estimation. In this case, the impact of isolated high noise is alleviated by smoothing within the delayed symbols. $\{0, 0, 0\}$ was also tested as the initial channel guess. We found LPA-VA with 4 survivors per state converged for all 100 trials, while the conventional adaptive VA with 10 symbols delay had 4 trials that converged to wrong channel estimates with the final square error larger than 0.1. This simulation demonstrates that compared with CA-VA, LPA-VA is more robust to initial channel guess and reaches a good channel estimate more quickly. However, it does not achieve the best convergence performance in the following channel updating.

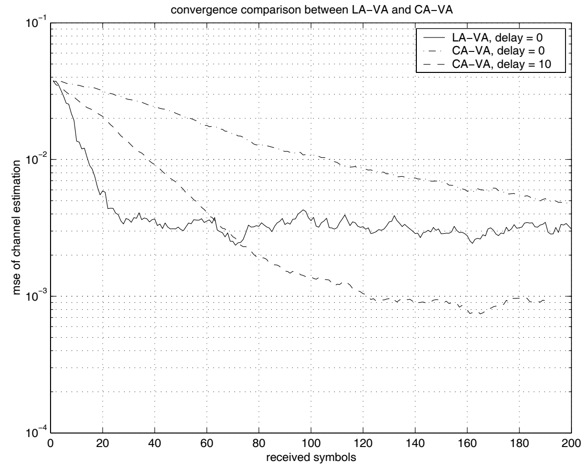

Fig. 3. Comparison between the LPA-VA and CA-VA with delay = 0, 10 for channel b estimation and updating. SNR is 13 dB.

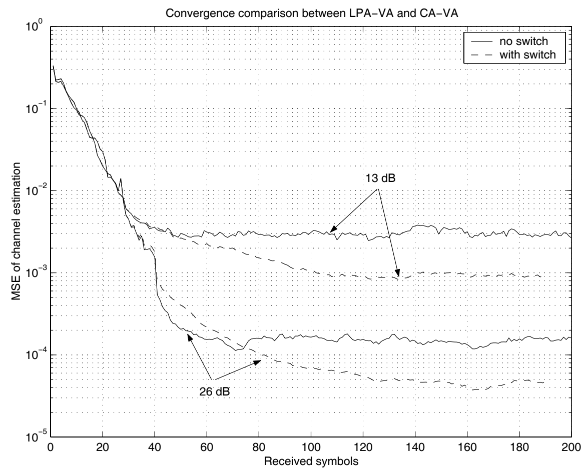

Fig. 4. Comparison between the proposed approach and the one that only uses LPA-VA with 4 survivors/state for channel b. SNRs are 13dB and 26dB, respectively. Average switch point for 13dB is the 49^{th} symbol, while that for 26dB is the 31^{th} symbols.

Second, we compare the proposed two-mode scheme with one that only uses LPA-VA for channel b. 4 survivors per state are adopted in LPA-VA. In our approach, the switch is set when the number of consecutive states with the minimum path metrics along a single path is greater than 20. The channel corresponding to the minimum path metric is then chosen to set the whole trellis. CA-VA is applied for signal detection and further channel update. 100 trials were conducted in conditions of 13dB and 26dB SNRs. Figure 4 shows the simulation results. Channel estimation results over channel a and c are also shown in Figure 5 and Figure 6, respectively. Eight survivors per state are used in LPA-VA in these two cases because of the longer impulse responses. Number of consecutive states along a valid path for switch is set to 30. The average switching positions are shown in Table 1. Apparently, the position number is smaller for a higher SNR value, which is consistent with the fact that good channel estimates are easy to achieve in high SNR situations. The channels with more taps also need more symbols to conduct the initial channel estimation. As shown in the results, The proposed scheme indeed takes advantages of LPA-VA and CA-VA. It obtains a good channel estimate very fast, and at the same time, achieves a better MSE performance in the following channel updating. Moreover, the computational complexity is significantly reduced in the proposed approach since only one channel estimate is maintained in the second mode of process. Other step sizes are also tested in LMS and all shows the lower MSE using CA-VA than using LPA-VA. In all above cases, the channel estimation start from vector of $\vec{0}$. This also shows that this scheme can also achieve the automatic gain control very well from the initial zero amplitude to the appropriate channel estimate.

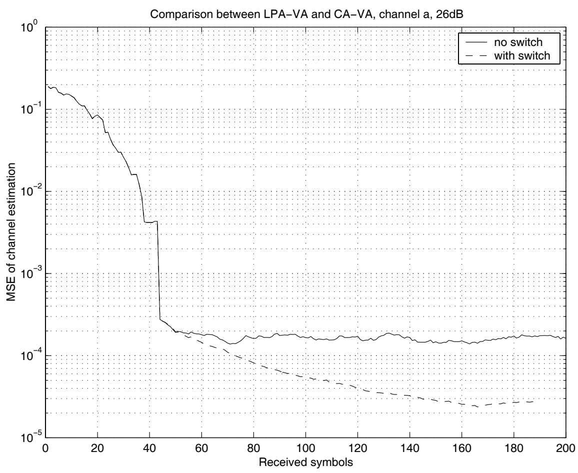

Fig. 5. Comparison between the proposed approach and the one that only uses LPA-VA with 8 survivors/state for channel a. SNR is 26dB. Average switch point is the 51^{th} symbols.

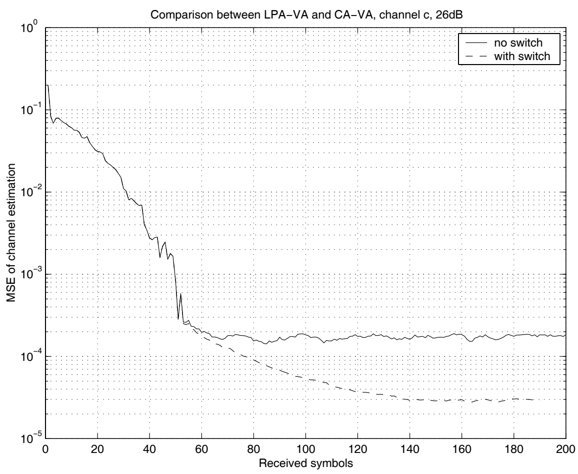

Fig. 6. Comparison between the proposed approach and the one that only uses LPA-VA with 8 survivors/state for channel c. SNR is 26dB. Average switch point is the 54^{th} symbols.

Table 1. Switch positions for various conditions.

	$Mean\_pos$	Min_pos	Max_pos	dB
channel b	48.6	26	103	13
channel b	30.8	26	60	26
channel a	50.8	38	73	26
channel c	53.5	36	85	26

We also show in Figure 7 the channel equalization results of the scheme over channel b compared to the MLSE method in which the channel impulse response is assumed known while signal detection is via the Viterbi decoding. In this simulation, we set each block have 1000 bits, bits starting from minimum number of 250 and the switch position are used for the BER calculations. We find the equalization results are very close to the MLSE signal detection with less than 0.5dB loss. The BER result for data without ISI is also shown in the figure for comparison.

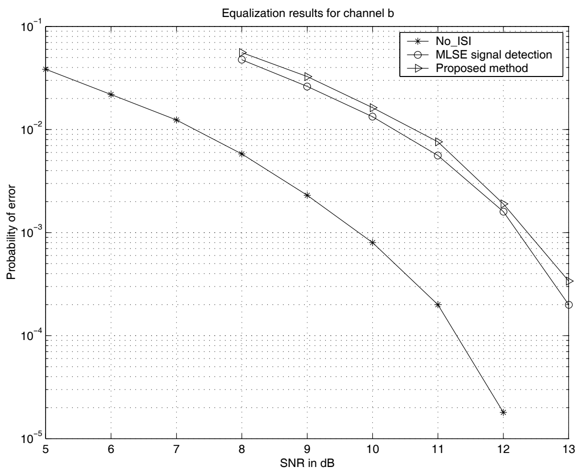

Fig. 7. Equalization result of the proposed approach for channel b.

IV. CONCLUSION

In this paper, we present a trellis based blind channel estimation and signal detection method. From the received symbols only, an initial channel estimation is conducted using a list parallel adaptive Viterbi algorithm in which multiple channel estimates are maintained simultaneously. Upon getting a good initial channel estimate, the process is switched to signal detection and decision feedback that is realized using a conventional adaptive Viterbi algorithm. Switching between these two operations is designed by exploiting the evolution status of the path metrics and the linear constraint in trellis mapping. Simulation results show that this method achieves a good overall performance in terms of lower MSE for channel estimation, robustness to initial channel estimation, less computational complexity, and good equalization results.

REFERENCES

[1] L. E. Baum, "An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes," *In-*

equalities, vol. 3, pp. 1–8, Jan. 1972.

[2] L. R. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[3] G.K. Kaleh and R. Vallet, “Joint parameter estimation and symbol detection for linear or nonlinear unknown channels,” *IEEE Trans. on Communications*, vol. 42, no. 7, pp. 2406–2413, July 1994.

[4] M. Feder and J. A. Catipovic, “Algorithms for joint channel estimation and data recovery - application to equalization in underwater communications,” *IEEE Journal of Oceanic Engineering*, vol. 16, no. 1, pp. 42–55, Jan. 1991.

[5] B.-H. Juang and L. R. Rabiner, “The segmental K-means algorithm for estimating parameters of hidden markov models,” *IEEE Trans. on Acoustics, Speech, And Signal Processing*, vol. 38, no. 9, pp. 1639–1641, Sep. 1990.

[6] N. Seshadri, “Joint data and channel estimation using blind trellis search techniques,” *IEEE Trans. on Communications*, vol. 42, no. 2/3/4, pp. 1000–1011, Feb./Mar./Apr. 1994.

[7] E. Zervas, J. Proakis, and V. Eyuboglu, “A ”quantized” channel approach to blind equalization,” *Proc. IEEE International Conference on Communications*, vol. 3, pp. 1539–1543, June 1992.

[8] J. G. Proakis, “Digital Communications”, *McGraw-Hill*, 4ed., 2000.

[9] J. G. D. Forney, “Convolutional codes II: Maximum likelihood decoding,” *Information and Control*, vol. 25, pp. 222–266, July 1974.

Lei Cao received the B.E. degree in electrical engineering from Hefei University of Technology, China in 1990, the M.S. degree in computer science from University of Science and Technology of China in 1993, and the Ph.D. degree in electrical engineering from University of Missouri-Columbia in 2002. Currently he is an assistant professor in the Electrical Engineering Department, the University of Mississippi. His research interest include multimedia processing, coding and wireless communications.

Chang Wen Chen received the B.S degree from the University of Science and Technology of China in 1983, the M.S.E.E. degree from University of Southern California, Los Angeles, in 1986, and Ph.D. degree from University of Illinois at Urbana-Champaign in 1992. His research areas include multimedia processing and compression, mobile wireless networking, biomedical image processing, telemedicine, visualization etc. Dr. Chen has received many research awards from NSF, NASA, Whitaker Foundation, DARPA, and several major corporations. He served as the header of the Interactive Media Group at Sarnoff Corporation. Currently he is Allen S. Henry Distinguished Professor, Director of Wireless Center of Excellence of the Department of Electrical and Computer Engineering in Florida Institute of Technology. He is serving as an Associated Editor for the IEEE Trans. on Circuits and Systems for Video Technology, IEEE trans. on Multimedia, and on the editorial boards of several other international Journals.

Philip Orlik received the B.E. degree in 1994 and the M.S. degree in 1997 both from the State University of New York at Stony Brook. In 1999 he earned his Ph. D. in electrical engineering also from SUNY Stony Brook. He joined Mitsubishi Electric Research Laboratories (MERL) in August 2000, and is currently a member of technical staff. His research interests include wireless and optical communications, networking, queuing theory, and analytical modeling.

Jinyun Zhang received her Ph.D. in Electrical Engineering from the University of Ottawa in the area of digital signal processing, where she was also a Visiting Scholar and worked on digital image processing. Prior to this she was a teacher/lecturer at Tsinghua University, Beijing, China. Dr. Zhang worked for Nortel Networks for the past ten years where she held engineering and most recently management positions in the areas of VLSI design, Advanced Wireless Technology Development, Wireless Networks and Optical Networks. She has a broad technical background, specializing in system design, DSP algorithms, and real-time embedded S/W for wireless communications and DWDM optical networks. Jinyun Zhang joined the Mitsubishi Electric Research Laboratories (MERL) in 2001.

Daqing Gu received the BE degree from Tsinghua University, Beijing, China in 1987; the MS and Ph.D. degrees in electrical engineering from the State University of New York at Stony Brook, Stony Brook, NY in 1996 and 1999, respectively. He joined Mitsubishi Electric Research Laboratories (MERL) in 1999, and is currently a principal member of technical staff. He has been involved in many wireless communications and networking projects, and has many publications in the field. His current research interests include IEEE802.11 standardizations, QoS in wireless communications and networks, multimedia home networking and MIMO-OFDM technologies.