

3-1995

## Optimal Sequence Estimation for Convolutionally Coded Signals With Binary Digital Modulation in ISI Channels

Fuqin Xiong

Cleveland State University, [f.xiong@csuohio.edu](mailto:f.xiong@csuohio.edu)

Follow this and additional works at: [https://engagedscholarship.csuohio.edu/enece\\_facpub](https://engagedscholarship.csuohio.edu/enece_facpub)

 Part of the [Signal Processing Commons](#)

**How does access to this work benefit you? Let us know!**

### Original Citation

Xiong, F. (1995). Optimal Sequence Estimation for Convolutionally Coded Signals with Binary Digital Modulation in ISI Channels. *Journal of The Franklin Institute*, 332(2), 125-140.

### Repository Citation

Xiong, Fuqin, "Optimal Sequence Estimation for Convolutionally Coded Signals With Binary Digital Modulation in ISI Channels" (1995). *Electrical Engineering and Computer Science Faculty Publications*. 57.  
[https://engagedscholarship.csuohio.edu/enece\\_facpub/57](https://engagedscholarship.csuohio.edu/enece_facpub/57)

This Article is brought to you for free and open access by the Electrical and Computer Engineering Department at EngagedScholarship@CSU. It has been accepted for inclusion in Electrical Engineering and Computer Science Faculty Publications by an authorized administrator of EngagedScholarship@CSU. For more information, please contact [library.es@csuohio.edu](mailto:library.es@csuohio.edu).

# *Optimal Sequence Estimation for Convolutionally Coded Signals with Binary Digital Modulation in ISI Channels*

by FUQIN XIONG

*Department of Electrical Engineering, Cleveland State University, Cleveland, OH 44115, U.S.A.*

**ABSTRACT:** *Decoding convolutional codes with binary digital modulation in intersymbol interference (ISI) channels is studied. The receiver structure is a whitened matched filter (WMF) whose transfer function is determined by the ISI channel. Decoding of the output sequence can be performed in two steps or one step. The two-step decoding first decodes the ISI corrupted coded sequence back to the ISI free coded sequence which is then decoded back to the uncoded message sequence. For one-step decoding, the entire encoder-channel-receiver system is modeled as a new encoder with combined memory length of the memory lengths of the original encoder and the channel, and followed by a weighted summation mapping from the binary symbols to real number symbols. The weighting coefficients are determined by the channel characteristic. In both two-step and one-step decoding, the Viterbi algorithm (VA) is used to perform the maximum likelihood decoding. Decoding error probability and complexity of both methods are analyzed, simulated and compared.*

## ***1. Introduction***

A maximum likelihood sequence estimator (MLSE) consisting of a whitened matched filter followed by a Viterbi decoder was developed by Forney (1) for PAM channels with ISI. Since then, many papers have proposed a variety of schemes to reduce the complexity of the Viterbi algorithm (2–5). But all the above schemes are for an uncoded input sequence. Unlike uncoded input symbols, the coded symbols are correlated. However, the derivation of the WMF does not require the independence of the input symbols (1). Therefore, WMF structure is also optimal to coded input sequence. This paper describes how to apply the MLSE to convolutionally coded signals.

Previously, two schemes of using the Viterbi algorithm (VA) to decode ISI channels with convolutionally coded input sequence have been proposed (6, pp. 284–287). One of them does not use a whitening filter, therefore the output noise samples are not statistically independent. This poses a great difficulty to the analysis of the receiver performance. As a result no bit error rate expression was presented. The other receiver which does use the WMF appeared as a problem in (6, p. 298). However, no analysis or simulation has been found in the literature.

After Ungerboeck invented the famous trellis coded modulation (TCM) (7),

several authors applied the Viterbi algorithm to decode the ISI contaminated TCM signals (2, 8–10). Their receivers use a WMF. Therefore, the Viterbi algorithm is applied to the combined super trellis to decode the received signal sequence. To reduce the computational complexity of the Viterbi algorithm in the super trellis, the above papers explored a variety of reduced-state Viterbi algorithms. Duell-Hallen and Heegard (2) proposed a delayed decision feedback algorithm which is a reduced-state Viterbi algorithm with feedback incorporated into the structure of path metric computations. Chevillat and Eleftheriou (8) and Eyuboğlu and Qureshi (9) used set partitioning to reduced the number of states of the super trellis. Zou and Weinrichter (10) simply truncated the ISI coefficients to reduce the number of states. All these algorithms can achieve a certain compromise between complexity and error performance. However, they all are dealing only with TCM signals in ISI channels.

This paper deals with the situation of decoding convolutional codes in ISI channels. This situation arises in systems that use convolutional codes and binary modulation such as BPSK, DPSK and BFSK. In these cases, a transmitted symbol corresponds to one bit of the coded bit sequence. The finite state machine (FSM) representing the transmitter–channel–WMF combination is a one-dimensional transversal filter. However, in the case of TCM, since a symbol is chosen from the two-dimensional signal constellation using Ungerboeck’s set partitioning method, the FSM is a complex (or two-dimensional) transversal filter. Therefore, despite the similarity between TCM and original convolutional codes, neither the methods nor the results in (2, 8–10) are applicable to the cases of convolutional codes with binary modulations.

This paper is to fill this gap. A well-structured combined FSM will be derived. This combined FSM will be expressed by a generator matrix which is derived from the code generator and the channel FSM model. This allows the results to be applied to any combination of convolutional codes of any code rate and channel FSM model. Based on this combined FSM model the super trellis can be easily constructed and error performance and complexity of the Viterbi algorithm can be easily evaluated. Any reduced-state Viterbi algorithm can also be applied and their performance can be easily analyzed using previous results in the literature.

The paper is organized as follows. In Section II, the receiver structure is presented. Section III describes the combined FSM model and the combined trellis for one-step decoding. Section IV is the performance analysis and comparison. Section V concludes the paper.

## ***II. Receiver Structure***

The maximum likelihood receiver structure for an ISI channel with uncoded inputs consists of a WMF and a Viterbi decoder (1). With a convolutional encoder at the front end the cascade of the encoder, channel (baseband) and the WMF is shown in Fig. 1.

The input and output symbols of the encoder are binary (0,1). In a practical system the (0,1) alphabet is converted to a (−1,1) alphabet since the latter needs 3 dB less average bit energy for the same error probability. However, it can be

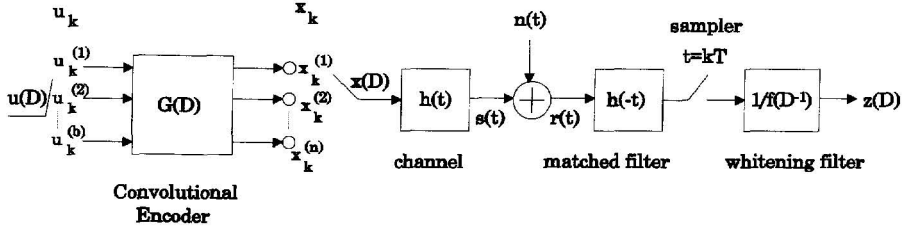


FIG. 1. Cascade of the encoder, the channel and the whitened matched filter.

shown that the effect of the input level shifting on the output of the FSM is merely a level shifting/scaling of the output signals, which does not affect the error performance analysis. Therefore, the converter is omitted in the model of this paper.

The channel is characterized by a square-integrable impulse response  $h(t)$  of length  $L$  symbol intervals, i.e.  $h(t) = 0$  for  $t > LT$  and  $t < 0$  and  $\|h\|^2 = \int_{-\infty}^{+\infty} h^2(t) dt < \infty$ . A white Gaussian noise  $n(t)$  of zero mean and double-sided spectral density  $N_0/2$  represents noise entering at various points of the channel. The transfer function of the whitening filter is  $1/f(D^{-1})$ , where  $f(D^{-1})$  is one of two factors of the spectrum of the channel:

$$r(D) = f(D)f(D^{-1}), \quad (1)$$

where

$$r(D) = \sum_{k=-v}^v r_k D^k \quad (2)$$

and  $v = L-1$ . We use  $r(D)$  to denote the  $D$ -transform of the autocorrelation function of  $h(t)$ ,

$$r_k = \begin{cases} \int_{-\infty}^{\infty} h(t)h(t-kT) dt & |k| \leq L-1 \\ 0 & |k| \geq L. \end{cases} \quad (3)$$

The cascade of the channel and the WMF can be lumped in a FSM model whose transfer function is  $f(D)$  of degree  $v$ , another factor of the  $r(D)$  (1):

$$f(D) = f_0 + f_1 D + \dots + f_v D^v \quad (4)$$

Thus the entire system can be shown as in Fig. 2.

The code is an  $(n, b, m)$  code of rate  $R = b/n$ . The transfer function matrix  $G(D)$  of the encoder is defined as

$$G(D) = \begin{bmatrix} g_1^{(1)}(D) & g_1^{(2)}(D) & \dots & g_1^{(n)}(D) \\ g_2^{(1)}(D) & g_2^{(2)}(D) & \dots & g_2^{(n)}(D) \\ \vdots & \vdots & & \vdots \\ g_b^{(1)}(D) & g_b^{(2)}(D) & \dots & g_b^{(n)}(D) \end{bmatrix} \quad (5)$$

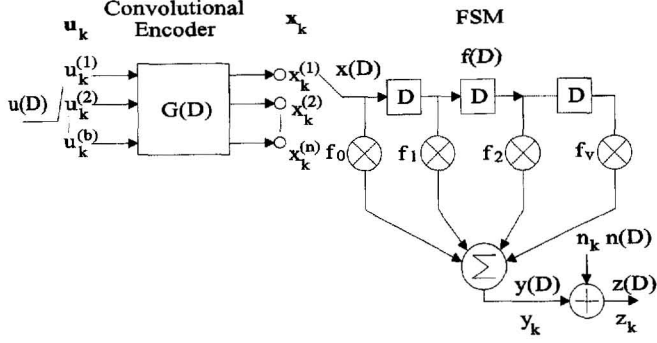


FIG. 2. Cascade of the encoder and the finite state machine model.

in which  $g_i^{(j)}(D)$  is the generator polynomial relating input  $u^{(i)}$  to output  $x^{(j)}$ , and  $g_i^{(j)}(D)$  has a maximum degree of  $m$ .

Corresponding to a  $b$ -dimensional input vector  $\mathbf{u}_k$ , there is an  $n$ -dimensional vector  $\mathbf{x}_k$ . Their relation, in  $D$ -transform, is

$$\mathbf{x}(D) = G^T(D)\mathbf{u}(D) \quad (6)$$

where all operations are in GF(2) field (modulo-2), and the superscript T denotes transpose of the matrix. The components of  $\mathbf{x}_k$  are shifted sequentially into the FSM. In other words, the input sequence to the FSM is a one-dimensional sequence  $x(D)$ , thus the output sequences are

$$y(D) = f(D)x(D) \quad (7)$$

and

$$z(D) = y(D) + n(D) \quad (8)$$

where all operations are in real number field  $\mathbf{R}$ . We use  $n(D)$  to denote a white Gaussian noise sequence with zero mean and variance  $N_0/2$ .

In time domain the relation is a convolution :

$$y_k = \sum_{i=0}^v f_i x_{k-i} \quad (9)$$

and

$$z_k = y_k + n_k. \quad (10)$$

### III. One-step Decoding

Based on the diagram in Fig. 2, a straightforward decoding method would be a two-step decoding method. The two-step decoder decodes sequence  $z(D)$  back to  $x(D)$  which is then decoded back to  $\mathbf{u}(D)$ . For maximum likelihood decoding, the first decoder (VA1) is a soft-decision Viterbi decoder (1) and the second one (VA2) is a hard-decision Viterbi decoder. VA2 cannot make use of any signal amplitude

information which has been lost after VA1. However, if we combine the encoder and the FSM together and apply a one-step Viterbi decoding to the super FSM, the signal amplitude information would be fully utilized in the decoding process.

In order to derive the one-step decoding algorithm, all inputs and outputs are grouped in vector forms. Corresponding to each  $b$ -dimensional input vector  $\mathbf{u}_k$ , there are  $n$ -dimensional vectors  $\mathbf{x}_k$ ,  $\mathbf{y}_k$  and  $\mathbf{z}_k$  at different stages' output. The relations between  $\mathbf{u}_k$  and  $\mathbf{x}_k$ ,  $\mathbf{y}_k$  and  $\mathbf{z}_k$  are straightforward (see Eqs (6), (8)).

However, the relation between  $\mathbf{y}_k$  and  $\mathbf{x}_k$  is not simple due to the sequential feeding of the  $\mathbf{x}'_k$ s components into the FSM. In fact, each  $\mathbf{y}_k$  involves current  $\mathbf{x}_k$  and at least one previous input, namely,  $\mathbf{x}_{k-1}$ . In general,  $\mathbf{y}_k$  is determined by  $\mathbf{x}_k$ ,  $\mathbf{x}_{k-1}, \dots, \mathbf{x}_{k-\lambda}$  as follows:

$$\begin{bmatrix} y_k^{(1)} \\ y_k^{(2)} \\ \vdots \\ y_k^{(\eta)} \end{bmatrix} = \begin{bmatrix} 0 & \dots & 0 & 0 & f_0 & \dots & f_v \\ 0 & \dots & 0 & f_0 & \dots & f_v & 0 \\ \vdots & \vdots & & & \vdots & & \vdots \\ f_0 & \dots & f_v & 0 & \dots & 0 & 0 \end{bmatrix} \begin{bmatrix} x_k^{(n)} \\ \vdots \\ x_k^{(1)} \\ x_{k-1}^{(n)} \\ \vdots \\ x_{k-1}^{(1)} \\ x_{k-\lambda}^{(n)} \\ \vdots \\ x_{k-\lambda}^{(\eta)} \end{bmatrix} \quad (11)$$

where  $\lambda = \lceil v/n \rceil$  and  $\eta = \lambda n - v + 1$ . Here,  $\lceil v/n \rceil$  denotes the smallest integer not less than  $v/n$ . In vector form the above equation can be written as

$$\mathbf{y}_k = \mathbf{F} \mathbf{x}'_k \quad (12)$$

in obvious notation.

Measured in number of input vectors, the memory length of the FSM is  $\lambda$ . The total memory length of the encoder-FSM cascade is therefore

$$m' = m + \lambda. \quad (13)$$

The dimension of the output vector  $\mathbf{x}'_k$  is

$$n' = n + v. \quad (14)$$

Thus the system can be equivalently thought of as a new encoder  $G'(D)$  with a longer memory  $m'$  and a larger dimensional ( $n'$ ) output. The number of states of the encoder is  $S' = 2^{bm'} = S2^{b\lambda}$ , where  $S = 2^{bm}$  is the number of states of the original encoder. The new encoder generates  $\mathbf{x}_k, \mathbf{x}_{k-1}, \dots, \mathbf{x}_{k-\lambda}$  simultaneously. They compose  $\mathbf{x}'_k$ . The new generator matrix is

$$G'(D) = [G_r(D), DG_r(D), \dots, D^\lambda \tilde{G}_r(D)], \quad (15)$$

where  $G_r(D)$  is obtained by reversing the order of  $G(D)$ 's columns:

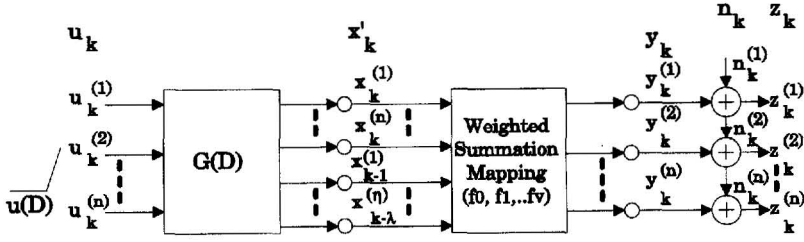


FIG. 3. Combined finite state machine model of the system of Fig. 2.

$$G_r(D) = \begin{bmatrix} g_1^{(n)}(D) & g_1^{(n-1)}(D) & \dots & g_1^{(1)}(D) \\ g_2^{(n)}(D) & g_2^{(n-1)}(D) & \dots & g_2^{(1)}(D) \\ \vdots & \vdots & & \vdots \\ g_b^{(n)}(D) & g_b^{(n-1)}(D) & \dots & g_b^{(1)}(D) \end{bmatrix}. \quad (16)$$

Then  $\tilde{G}_r(D)$  is formed from  $G_r(D)$  by dropping  $g_i^{(j)}(D)$  for  $i = 1, \dots, b$  and  $j < \eta$ . The new encoder can be represented by

$$\mathbf{x}'(D) = G'^T(D)\mathbf{u}(D). \quad (17)$$

Then  $\mathbf{x}'_k$  is mapped into  $\mathbf{y}_k$  through a weighted summation operation with the weighting coefficients  $f_0, f_1, \dots, f_v$  (Fig. 3). Clearly this equivalent model is an FSM. The output vector  $\mathbf{y}_k$  is determined by the state of the encoder  $G'(D)$  and the input vector  $\mathbf{u}_k$ . The sufficient statistic is  $\mathbf{z}_k = \mathbf{y}_k + \mathbf{n}_k$ , where  $\mathbf{n}_k$  is the white Gaussian noise vector. Its components are independent and one noise vector is independent from another.

#### Example 1

A simple ISI channel and a simple code are chosen in this example for the purpose of illustration. The channel is the one-pole channel

$$h(t) = \begin{cases} \beta e^{-\alpha t} & t \geq 0 \\ 0 & t < 0 \end{cases} \quad (18)$$

where  $\alpha > 0$ ,  $\beta > 0$ . It is a slowly decaying low-pass channel when  $\alpha$  is small. By choosing appropriate parameters and truncation (4), one possible transfer function  $F(D)$  of the FSM model could be

$$\begin{aligned} f(D) &= f_0 + f_1 D + f_2 D^2 \\ &= 1 + e^{-0.5} D + e^{-1.0} D^2 \\ &= 1 + 0.607 D + 0.368 D^2. \end{aligned} \quad (19)$$

Now consider a maximum  $d_{\text{free}}(3,2,1)$  code with

$$G(D) = \begin{bmatrix} 1+D & D & 1+D \\ D & 1 & 1 \end{bmatrix} \quad (20)$$

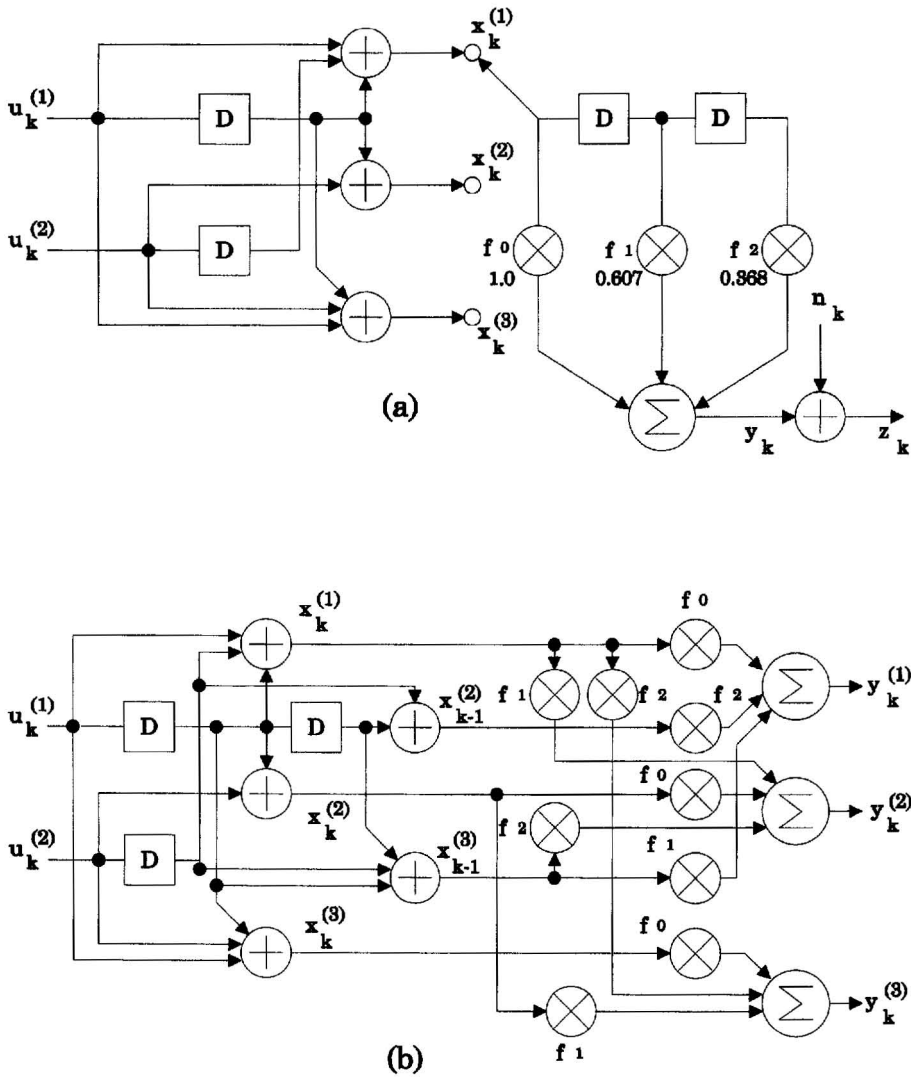


FIG. 4. Figures for Example 1: (a) Cascade of the encoder and the FSM; (b) Combined FSM.

and the encoder is cascaded with this channel as shown in Fig. 4(a), from which it is seen that

$$\begin{aligned}
 y_k^{(1)} &= f_0 x_k^{(1)} + f_1 x_{k-1}^{(3)} + f_2 x_{k-1}^{(2)}, \\
 y_k^{(2)} &= f_0 x_k^{(2)} + f_1 x_k^{(1)} + f_2 x_{k-1}^{(3)}, \\
 y_k^{(3)} &= f_0 x_k^{(3)} + f_1 x_k^{(2)} + f_2 x_k^{(1)}.
 \end{aligned} \tag{21}$$

The equivalent system model is shown in Fig. 4(b). The new encoder grows out of

the original encoder by adding one more memory element ( $\lambda = 1$ ) and repeating the connections for the  $x_k^{(2)}$  and  $x_k^{(3)}$  one stage later to form the connections for  $x_{k-1}^{(2)}$  and  $x_{k-1}^{(3)}$ . The generator of the new encoder is therefore

$$G'(D) = \begin{bmatrix} 1+D & D & 1+D & D+D^2 & D^2 \\ 1 & 1 & D & D & D \end{bmatrix}. \quad (22)$$

The code rate reduces to  $R' = 2/5$ . However, it is not the real rate in the system. The real rate is still  $2/3$ .

Given the equivalent finite state machine, maximum likelihood (ML) estimation can be applied to the output sequence  $\mathbf{z}(D)$  to estimate the input sequence  $\mathbf{u}(D)$ . Extending the one-dimensional case in (1) to the multiple-dimensional case, it is easy to show that maximizing the likelihood is equivalent to minimizing the following quantity,

$$M_p(\mathbf{u}(D)) = \sum_{k=1}^N M_b(\mathbf{u}_k, \mathbf{z}_k), \quad (23)$$

which is the path metric, where

$$M_b(\mathbf{u}_k, \mathbf{z}_k) = \sum_{i=1}^n (z_k^{(i)} - y_k^{(i)})^2 \quad (24)$$

is the branch metric. The path metric is the squared Euclidean distance between sequences  $\mathbf{y}(D)$  and  $\mathbf{z}(D)$ . The branch metric is the squared Euclidean distance between vectors  $\mathbf{y}_k$  and  $\mathbf{z}_k$ .

The Viterbi algorithm is an efficient means of implementing the strict ML estimation (6). It will use the above metrics. For the equivalent  $(n', b, m')$  FSM, the  $m'$  most recent inputs determine a state of the FSM. Denote a state at any time  $k$  as  $\mathbf{s}_k$ ; then

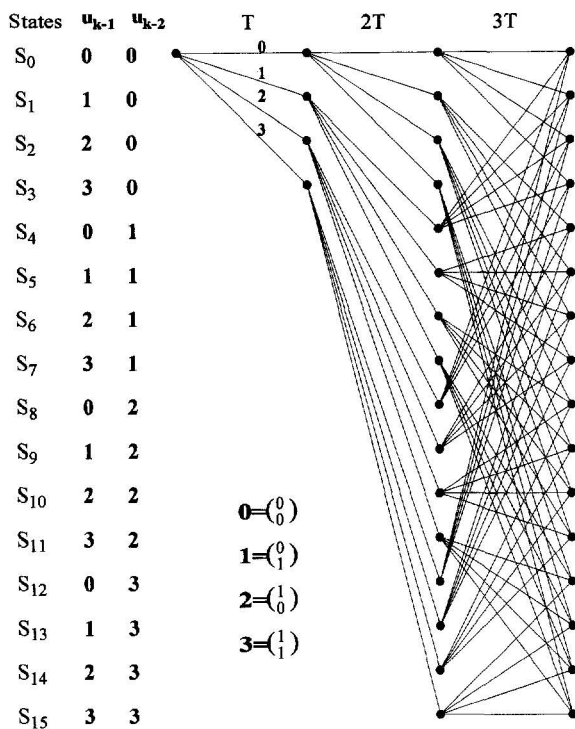
$$\mathbf{s}_k = (\mathbf{u}_{k-1}, \mathbf{u}_{k-2}, \dots, \mathbf{u}_{k-m'}). \quad (25)$$

The trellis of the Viterbi algorithm has  $2^{bm'}$  states. There are  $2^b$  transitions stemming from each state. Each transition is associated with an input vector  $\mathbf{u}_k$  and an output vector  $\mathbf{y}_k$  as denoted by

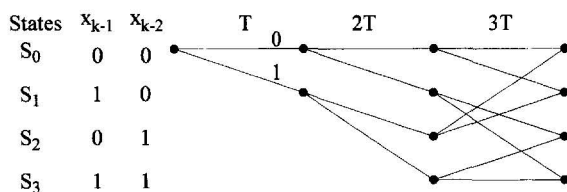
$$\mathbf{y}_k = y(\mathbf{u}_k, \mathbf{s}_k) = y(\mathbf{u}_k, \hat{\mathbf{u}}_{k-1}, \dots, \hat{\mathbf{u}}_{k-m'}). \quad (26)$$

To calculate  $\mathbf{y}_k$  one must first calculate  $\mathbf{x}'_k$  from  $(\mathbf{u}_k, \hat{\mathbf{u}}_{k-1}, \dots, \hat{\mathbf{u}}_{k-m'})$  using (17) (modulo-2 arithmetic), then calculate  $\mathbf{y}_k$  from  $\mathbf{x}'_k$  using (12) (real number arithmetic).

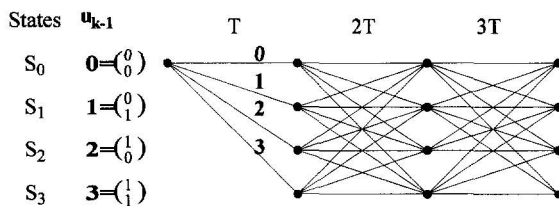
Figure 5 shows trellises of the FSMs of Example 1. The Viterbi algorithm searches through the trellis for the ML path. For details of the well known Viterbi algorithm the readers are referred to (11) or other books on coding.



(a) Trellis for one-step decoding



(b) Trellis for VA1 in two-step decoding



(c) Trellis for VA2 in two-step decoding

FIG. 5. Trellises for Example 1.

#### IV. Performance Analysis

##### 4.1. Error probability

The analysis for the one-step decoding is basically an extension from the one-dimensional uncoded case in (1, 12) to the coded  $b$ -dimensional input  $n$ -dimensional output FSM case.

Following a derivation procedure similar to that in (1, 12), the upper bound of the bit error probability is found as

$$P_b \leq \sum_{d \in D} Q \left[ \frac{d}{\sqrt{2N_0}} \right] \frac{1}{b} \sum_{\varepsilon \in E_d} W_H(\varepsilon) \left[ \prod_{i=0}^H \prod_{j=1}^b \frac{2 - |e_{u_i}^{(j)}|}{2} \right] \quad (27)$$

where  $\varepsilon$  denotes an error event. Here,  $E_d$  is the set of error events which cause all incorrect paths at distance  $d$ ,  $D$  is the set of all possible paths which are of distance  $d$  away from the correct path,  $W_H(\varepsilon)$  is the Hamming weight of the input error sequence  $\mathbf{e}_u(D)$  which has a degree of  $H$ , and  $e_{u_i}^{(j)}$  is the  $j$ th component of the  $i$ th term in the error sequence  $\mathbf{e}_u(D)$ . The  $Q(\cdot)$  function is the error function defined as

$$Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} \exp \left[ -\frac{x^2}{2} \right] dx.$$

A distance  $d$  is associated with an error event and is evaluated by using error sequence  $\mathbf{e}_y(D)$  in the following expression :

$$\begin{aligned} d^2(\varepsilon) &= \|\mathbf{e}_y\|^2 = \sum_{i=0}^K \|\mathbf{e}_{y_i}\|^2 \\ &= \sum_{i=0}^K \sum_{j=1}^n (e_{y_i}^{(j)})^2 = [\mathbf{e}_y^T (D^{-1}) \mathbf{e}_y(D)]_{D=0}. \end{aligned} \quad (28)$$

To evaluate the bound in (27), a simple but lengthy method is to perform a quasi-exhaustive search over main incorrect paths whose distances are close to the minimum distance  $d_{\min-u}$ . This is usually done by a computer. If the distance spectrum of the FSM is such that the  $d_{\min-u}$  is much smaller than other distances, then an approximation of Eq. (27) is given by

$$P_{b1} \approx K_1 Q \left( \frac{d_{\min-u}}{\sqrt{2N_0}} \right), \quad (29)$$

where

$$K_1 = \frac{1}{b} \sum_{\varepsilon \in E_{d_{\min-u}}} \left[ W_H(\varepsilon) \prod_{i=0}^H \prod_{j=1}^b \frac{2 - |e_{u_i}^{(j)}|}{2} \right]. \quad (30)$$

This expression is obtained by simply ignoring all the terms in Eq. (27) except those involving the minimum distance  $d_{\min-u}$  of  $d(\varepsilon)$ . This approximation is called an upper estimate. It was shown that the upper bound asymptotically tends to

upper estimate when  $N_0$  goes to zero (1). When  $d_{\min-u}$  is much smaller than other  $d(\varepsilon)$ , Eq. (27) is accurate. However when  $d_{\min-u}$  is not much smaller than other  $d(\varepsilon)$ , the estimate can be quite poor. Then a larger section of the distance spectrum needs to be included in the error probability calculation.

In two-step decoding, the BER expression for VA1 is simply a special case of Eqs (29) and (30). In this case,  $b = 1$  and the input sequence is  $x(D)$ . Denoting the BER of the VA1 as  $P_{b-VA1}$ , then

$$P_{b-VA1} \approx K_2 Q\left(\frac{d_{\min-x}}{\sqrt{2N_0}}\right), \quad (31)$$

where

$$K_2 = \sum_{\varepsilon \in E_{d_{\min-x}}} \left[ W_H(\varepsilon) \prod_{i=0}^H \frac{2 - |e_{x_i}|}{2} \right]. \quad (32)$$

The minimum Euclidean distance  $d_{\min-x}$  is different from  $d_{\min-u}$  in one-step decoding. The one-step decoding is conducted in the composite trellis which relates  $\mathbf{u}(D)$  to  $\mathbf{y}(D)$  directly. All the possible  $\mathbf{u}(D)$ 's, but not all the possible  $\mathbf{x}(D)$ 's, are tested. Thus  $d_{\min-u}$  is the minimum distance in the composite trellis. Here,  $d_{\min-u}$  corresponds to error events in  $\mathbf{u}(D)$  that produce the minimum distance between  $\mathbf{y}(D)$ 's. In two-step decoding VA1 and VA2 are conducted in two different trellises. The VA1 is conducted in the trellis relating  $x(D)$  to  $y(D)$ . All the possible  $x(D)$ 's are tested, in other words, there is no constraint on  $x(D)$ 's. The error probability is determined by the  $d_{\min-x}$  in the trellis. Also,  $d_{\min-x}$  corresponds to error events in  $x(D)$  that produce the minimum distance:  $d_{\min-x}$  is the minimum distance purely due to the ISI channel and uncoded input sequences. Due to encoding those  $d_{\min-u}$  error events in  $\mathbf{u}(D)$  do not necessarily generate those  $d_{\min-x}$  error events in  $x(D)$ .

The error probability expression for a hard decision Viterbi decoder is well known (11):

$$p_b \approx \frac{1}{b} B_{d_{\text{free}}} 2^{d_{\text{free}}} p^{d_{\text{free}}/2}, \quad (33)$$

where  $d_{\text{free}}$  is the minimum distance between any two code words in the code generated by  $G(D)$ . Since convolutional code is linear,  $d_{\text{free}}$  is also the minimum weight of the nonzero code words in the code. Here,  $B_{d_{\text{free}}}$  is the total number of nonzero message bits on all weight  $d_{\text{free}}$  paths, and  $p$  is the channel transition error probability. However, this expression is only valid for an input sequence with random errors. The input sequence to the VA2 is the sequence  $\hat{x}(D)$  which is the output sequence of VA1. The errors in the input sequence of VA1,  $z(D)$ , are random. But the errors in  $\hat{x}(D)$  are not completely random. As a matter of fact, the errors most likely occur as bursts of the length of the error events which cause  $d_{\min-x}$ . As a result, Eq. (33) is not accurate for VA2. Thus  $P_{b-VA2}$  will be less than  $P_{b-VA1}$  and greater than that expressed by (33). The actual  $P_{b-VA2}$  depends upon the randomness of the errors in sequence  $x(D)$  and the error correcting capability of the code. Due to the vast difference in values predicted by (31) and (33), these

two bounds are so loose that they are not useful. The evaluation of  $P_{b-VA2}$  thus is better done by simulation.

#### 4.2. Coding gains

To find coding gain (or loss) one needs to write all BER expressions in terms of the average bit energy to noise spectral density ratio  $E_b/N_0$ . The  $E_b = 0.5$  in our model since the input alphabet is (0, 1) where both symbols are equally likely. The bit-by-bit detection of a uncoded binary ISI-free PAM sequence has a BER of

$$P_b = Q(\sqrt{E_b/N_0}).$$

The MLSE of an uncoded ISI sequence has a BER of

$$P_b = K_2 Q(d_{\min-x}/\sqrt{2N_0}) = K_2 Q(d_{\min-x}\sqrt{E_b/N_0}). \quad (34)$$

For the coded sequence the  $E_s = RE_b$ , where  $R$  is the code rate and  $E_s$  is the average symbol energy. Thus the BER of the one-step decoding is

$$P_{b1} \approx K_1 Q(d_{\min-u}\sqrt{RE_b/N_0}). \quad (35)$$

Thus, by comparing (34) and (35), the asymptotical coding gain for one-step decoding is approximately ( $K_1$  and  $K_2$  are ignored)

$$\gamma_1 \approx 10 \log (Rd_{\min-u}^2/d_{\min-x}^2)(\text{dB}) \quad (36)$$

where  $R$  accounts for the transmission energy reduction due to coding for a fixed energy per information bit.

For two-step decoding, analytical coding gain cannot be obtained due to the lack of error probability expression.

In the following analytical and numerical simulation results of the  $P_b$  of two examples are given.

#### 4.3. Error probability of examples

##### Example 2

This example assumes the same one-pole lowpass channel as in Example 1. The code has a generator

$$G(D) = [1 + D + D^2 + D^3 + D^6 \quad 1 + D^2 + D^3 + D^5 + D^6]. \quad (37)$$

This is the popular (2, 1, 6) code which has been implemented as a VLSI chip by several manufacturers. This code has an  $d_{\text{free}} = 10$  and an asymptotical hard decision coding gain of 3.97 dB and an asymptotical soft decision coding gain of 6.97 dB.

##### Example 3

This example assumes the same code as in Example 2, but the channel is a magnetic recording channel or Lorentzian channel given by

## BER of Example 2

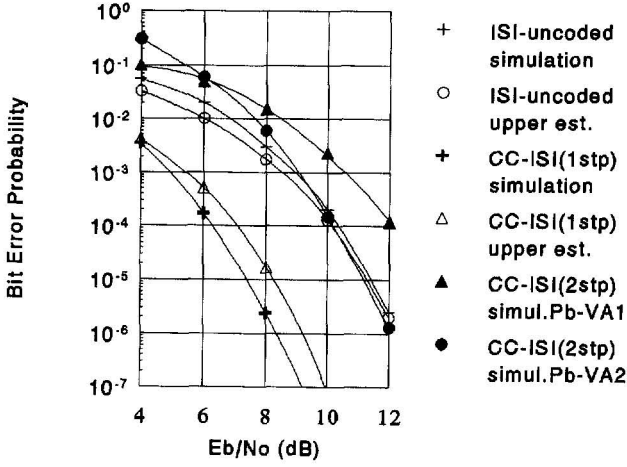


FIG. 6. Bit error probability of Example 2.

$$h(t) = \left[ 1 + \left( \frac{t}{\tau} \right)^2 \right]^{-1} - \left[ 1 + \left( \frac{t - T_b}{\tau} \right)^2 \right]^{-1} \quad (38)$$

where  $T_b$  is the bit period. Assuming  $T_b = \tau$  then the truncated FSM transfer function (2 ISI terms kept) is (4)

$$f(D) = 1 + 0.260D - 0.368D^2. \quad (39)$$

To determine the performance, the most important thing to do is to find all  $d_{\min}$ . The  $d_{\min-x}$  is determined by the channel. By computer search using (28),  $d_{\min-x}$  of Example 2 is found to be 1.161, which corresponds to two error events:  $\mathbf{e}_x(D) = \pm(1-D)$ . Here,  $K_2 = 1.0$  according to (32). The  $d_{\min-u}$  of Example 2 is found to be 2.335, which corresponds to two error events:  $\mathbf{e}_u(D) = \pm(1-D)$ . Here,  $K_1 = 1$ . The  $d_{\min-x}$  of Example 3 is found to be 1.097, which corresponds to two error events:  $\mathbf{e}_x(D) = \pm 1$ . Here,  $K_2 = 1.0$ . The  $d_{\min-u}$  of Example 3 is found to be 2.094, which corresponds to one error event:  $\mathbf{e}_u(D) = 1$ . Here,  $K_1 = 0.5$ . Using these parameters the BERs and the coding gains can be determined. The estimated coding gains for Examples 2 and 3 are 3.06 (dB) and 2.61 (dB), respectively, using one-step decoding. They are slightly poorer than that of ISI-free, hard decision decoding. But it is about 3 to 4 (dB) poorer than that of ISI-free soft decision decoding. This means that the ISI causes degradation in coding gain.

Figures 6 and 7 show the analytical and simulation results of Examples 2 and 3. From the figures it is seen that the simulated  $P_b$  of the one-step decoding is quite close to the prediction. Since the prediction is an upper estimate, the simulated  $P_b$  being slightly smaller is reasonable.

Figure 6 shows that the simulated asymptotical one-step coding gain for Example 2 is about 4 dB which is slightly larger than the prediction (3.06 dB). It also shows

### BER of Example 3

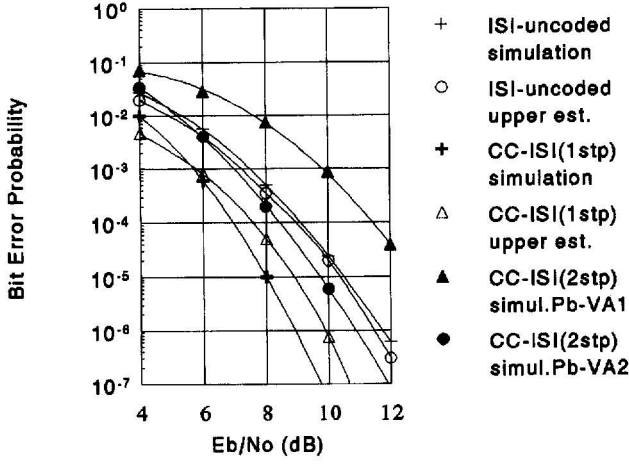


FIG. 7. Bit error probability of Example 3.

that the simulated two-step coding gain is about 0.2 dB at  $E_b/N_0$  over 10 dB. The VA2 improves the BER of the VA1 by 2 dB, but it still can not compensate for the loss of magnitude information after VA1. As a result, the one-step decoding is about 3.8 dB better than the two-step decoding in this example. However, when compared with ISI-free channel, soft decision case, the one-step decoding is still about 3 dB poorer. This 3 dB loss is caused by the ISI.

Figure 7 shows that the simulated asymptotical one-step coding gain for Example 3 is about 2.5 dB which is close to the prediction (2.61 dB). It also shows that the simulated two-step coding gain is about 1 dB at  $E_b/N_0$  over 10 dB. The VA2 improves the BER of the VA1 by about 3 dB. The one-step decoding is about 2 dB better than the two-step decoding in this example. However, when compared with ISI-free channel, soft decision case, the one-step decoding is still about 4.5 dB poorer. This 4.5 dB loss is caused by the ISI.

From above results it is seen that different code-ISI channel combinations entail different losses compared with ISI-free, soft-decision decoding case whether one-step decoding or two-step decoding is used. More loss is seen when two-step decoding is used. The coding gain difference between one-step and two-step decodings varies widely (3.8 dB for Example 2 and 2 dB for Example 3). Therefore not much can be said about the difference in general. However, it is evident that the one-step decoding is better than the two-step decoding in terms of error performance.

#### 4.4. Complexity

The complexity of a Viterbi decoder is basically determined by its number of states.

The two-step decoding needs two Viterbi decoders operating in a pipeline style. The number of states is  $2^v$  for VA1 and  $2^{b_m}$  for VA2. In VA1, decoding must be

performed on  $2^v$  states for every symbol which represents  $R = b/n$  information bits. In VA2, decoding on  $2^{bm}$  states is performed for every  $b$  information bits. Thus the number of states for each information bit is

$$S_2 = S_{21} + S_{22} = \frac{2^v}{R} + \frac{2^{bm}}{b}. \quad (40)$$

If decoding is implemented by hardware, VA1 and VA2 are operating in a pipeline style, the decoding speed is dictated by the slower one of the two. If decoding is implemented by software, VA1 and VA2 are operating serially, the decoding time is the sum of the time used by them.

The one-step decoding needs only one Viterbi decoder. The number of states in the trellis is at most  $2^{bm'}$ . Decoding must be performed on  $2^{bm'}$  states for every  $b$  information bits. Thus the number of states for each information bit is

$$S_1 = \frac{2^{bm'}}{b}. \quad (41)$$

One-step Viterbi decoder has more states than VA2 since its memory length  $m' = m + \lambda > m$ . It also usually has more states than VA1 ( $2^{bm'}$  vs.  $2^v$ ). Consequently its speed usually is slower than the two-step decoding for software implementation. But for hardware implementation, its speed is not necessarily slower if parallel processing is used.

The ratio of  $S_1/S_2$  for Examples 2 and 3 is  $128/72 = 1.778$  which is consistent with the recorded CPU time ratio of 1.771 in the simulation.

## V. Conclusion and Discussion

The WMF receiver structure is adopted to detect convolutionally coded signals with ISI so that an ML decoder using the Viterbi algorithm can be applied. The decoder can be a two-step decoder or a one-step decoder. The two-step decoder consists of a soft-decision Viterbi decoder VA1 and a hard-decision Viterbi decoder VA2. The VA1 decodes the ISI and noise contaminated signal sequence into a ISI-free and noise-free sequence. The VA2 further corrects the errors in the sequence. The one-step decoder consists only one Viterbi decoder. In order to apply the one-step decoding, an equivalent composite finite state machine model is derived. The bit error probability of the decoders is derived. Two code-channel combinations are evaluated and simulated. The analysis and simulation show that the one-step decoding has better BER than the two-step decoding due to its soft-decision nature. However, the ISI entails loss to the coding gain, so that the coding gain is smaller than that for an ISI-free channel. As to complexity, the onestep decoder is either slower or needs more hardware because it has more states.

Even though the examples studied are in favor of the one-step decoding, the choice between the one-step and two-step decoding need be carefully examined by balancing the error performance and complexity. Also it is seen that the con-

volitional code must be powerful enough to retain some coding gain in ISI channels. In other words, codes with short constraint length are not suitable for ISI channels.

## References

- (1) G. D. Forney, Jr, "Maximum likelihood sequence estimation of digital sequences in the presence of intersymbol interference", *IEEE Trans. Inform. Theory*, Vol. IT-18, pp. 363–378, May 1972.
- (2) Duel-Hallen and C. Heegard, "Delayed decision-feedback sequence estimation", *IEEE Trans. Commun.*, Vol. COM-37, pp. 428–436, May 1989.
- (3) M. Eyuboğlu and S. Qureshi, "Reduced-state sequence estimation with set partitioning and decision feedback", *IEEE Trans. Commun.*, Vol. COM-36, pp. 13–20, January 1988.
- (4) F. Xiong, A. Zerik and E. Shwedyk, "Sequential sequence estimation for channels with finite or infinite intersymbol interference", *IEEE Trans. Commun.*, Vol. COM-38, pp. 795–804, June, 1990.
- (5) F. Xiong and E. Shwedyk, "Sequential sequence estimation for multiple channel systems with both ISI and ICI", *IEEE Trans. Commun.*, Vol. COM-41, pp. 322–331, February 1993.
- (6) A. J. Viterbi and J. K. Omura, "Principles of Digital Communication and Coding", McGraw-Hill Inc., New York, NY, 1979.
- (7) G. Ungerboeck, "Channel coding with multilevel/phase signals", *IEEE Trans. Inform. Theory*, Vol. IT-28, pp. 55–67, January 1982.
- (8) P. Chevillat and E. Eleftheriou, "Decoding of trellis-encoded signals in the presence of intersymbol interference and noise", *IEEE Trans. Commun.*, Vol. COM-37, pp. 669–676, July 1989.
- (9) M. Eyuboğlu and S. Qureshi, "Reduced-state sequence estimation for coded modulation on intersymbol interference channels", *IEEE Trans. Commun.*, Vol. COM-37, pp. 898–995, August 1989.
- (10) G. Zou and H. Weinrichter, "Comparison of decoding algorithms for trellis coded modulation using Viterbi algorithm in the presence of intersymbol interference", *Electr. Lett.*, Vol. 24, pp. 541–542, April 1988.
- (11) S. Lin and D. J. Costello, "Error control coding: fundamentals and applications", Prentice-Hall, Englewood Cliffs, NJ, 1983.
- (12) G. Ungerboeck, "Adaptive maximum likelihood receiver for carrier-modulated data-transmission systems", *IEEE Trans. Commun.*, Vol. COM-22, pp. 624–636, May 1974.