

## ON THE APPLICATION OF SEQUENCE ESTIMATION ALGORITHMS IN THE DIGITAL COMPACT CASSETTE (DCC)

Volker Braun, Kees A. Schouhamer Immink, Milton A. Ribeiro, and Gijs J. van den Eenden

**Abstract**—In the current Digital Compact Cassette (DCC) recorder, a straightforward threshold detection scheme is applied which does not exploit the distance properties of the 8-to-10 modulation (ETM) code. More advanced detection schemes are investigated to see if they improve system performance. Two soft decision sequence estimation algorithms are considered: a fully-fledged Viterbi algorithm realizing Maximum Likelihood Sequence Estimation (MLSE) and a suboptimal trellis detection algorithm. With either of the algorithms, an asymptotic gain of 3 dB versus threshold detection level in the presence of additive white Gaussian noise (AWGN) is expected. The suboptimal detection algorithm is based on the Running Digital Sum (RDS) trellis underlying the ETM code. Performing the suboptimal algorithm requires a minimum of 48 additions, 19 comparisons, and a single table look-up for both detection and decoding of an ETM codeword, versus a minimum of 4234 additions, 421 comparisons, and 423 table look-ups for the fully-fledged algorithm. Compared to the fully-fledged scheme, the suboptimal algorithm leads to a large reduction in computational complexity. The performance of either of the algorithms in the presence of AWGN is determined by means of a computer simulation. The fully-fledged Viterbi algorithm achieves a gain of about 2.3 dB versus threshold detection level for symbol error rates in the order of  $10^{-4}$ ... $10^{-5}$ . In this range of the symbol error rate, the suboptimal algorithm shows a loss of less than a quarter of a dB compared to the fully-fledged scheme. The suboptimal trellis detection algorithm thus results in only a slight degradation in error rate performance compared to the fully-fledged scheme. The examination of experimental data showed the feasibility of both sequence estimation algorithms on the auxiliary data track of the DCC system.

### I. INTRODUCTION TO THE DIGITAL COMPACT CASSETTE - DCC

THE Digital Compact Cassette (DCC) system is a digital extension of the well-known analog Compact Cassette (CC) system [10]. Backwards compatibility to the CC system is achieved by using a similar cassette format and the same tape speed. The tape speed of 4.76 cm/s and the minimum wavelength of the tape material of about  $1 \mu\text{m}$  allow a maximum bit rate per track of 96 kbps (2 bits per wavelength). Eight parallel tracks carrying main (music) data result in an overall main data bit rate of 768 kbps. For reasons of robustness, 50% redundancy is included for overhead and coding. A signal resulting in CD audio quality requires an information rate of about 1.4 Mbps. In the DCC system, therefore, a data reduction scheme called

This work was presented in part at the IEEE International Symposium on Information Theory, Trondheim, Norway, June 27 - July 1, 1994.

V. Braun is with the Institute for Experimental Mathematics, Ellernstr. 29, 45326 Essen, Germany.

K.A. Schouhamer Immink, M.A. Ribeiro, and G.J. van den Eenden are with the Philips Research Laboratories, Prof. Holstlaan 4, 5656 AA Eindhoven, The Netherlands.

Contributed Paper

Manuscript received September 27, 1994

PASC is applied, having a reduction factor 4 [11]. A separate track carries auxiliary data such as track information, index number, time codes and table of contents. The auxiliary data is used for music search. The auxiliary data bit rate of 12 kbps is eight times lower than the main data bit rate. Stationary magneto-resistive heads are used for reading. Each channel is equalized to a full response characteristic. Saturation recording is applied. The coding scheme consists of a concatenation of interleaved error correcting codes, Reed-Solomon codes, and an inner modulation code called ETM. The principles of modulation and coding in magnetic recording are described in, for example, [6]. The following section focuses on the details of the ETM code as used in the DCC system.

### II. 8-TO-10 MODULATION - ETM

In the DCC system, a binary 8-to-10 modulation (ETM) code is used. Eight parallel source bits are translated by the channel encoder into codewords of ten channel symbols, which results in a *code rate* of 8/10. Let  $\{\mathbf{x}_k\} = \{\dots, \mathbf{x}_{-1}, \mathbf{x}_0, \dots, \mathbf{x}_k, \dots\}$  denote the encoded sequence of binary symbols  $\mathbf{x}_k \in \{0, 1\}$ . We define the *Running Digital Sum (RDS)* of the encoded sequence  $\{\mathbf{x}_k\}$  as

$$z_k = \sum_{i=-\infty}^k (2\mathbf{x}_i - 1). \quad (1)$$

From (1), we derive the following recursive property of the RDS:

$$z_k = z_{k-1} + 2\mathbf{x}_k - 1. \quad (2)$$

Hence, the RDS at time  $k$  is given by  $z_k = z_{k-1} + 1$  after transmission of a 'one' (i.e.  $\mathbf{x}_k = 1$ ), and by  $z_k = z_{k-1} - 1$  after transmission of a 'zero' (i.e.  $\mathbf{x}_k = 0$ ).

The ETM code has the property that the RDS of the encoded sequence remains within the range  $-3 \leq z_k \leq 2$  for any  $k$ . In other words, the RDS of the encoded sequence takes a finite number of values. Codes assuming a finite number of RDS values are called *RDS-constrained* codes. RDS-constrained codes are DC-free codes, i.e., the encoded sequences have high-pass characteristics with a spectral null at the zero frequency. Pierobon [5] showed that the power spectral density function of an encoded sequence  $\{\mathbf{x}_k\}$  vanishes at the zero frequency if, and only if, the encoder is a finite running digital sum encoder. A tutorial description of RDS-constrained codes and the ETM code in a worked out example can be found in [1].

It will be apparent to the reader that spectral shaping of a sequence by removing the low-frequency components can be accomplished only at the price of a certain rate loss. In other words, we are interested in the *capacity* of RDS-constrained codes, i.e., the maximum possible information transmission rate. The capacity of RDS-constrained codes is determined by the number of values the RDS takes [1]. This number is often called the *Digital Sum Variation (DSV)*. The DSV of the ETM code is equal to 6, resulting in a capacity of 0.85. In other words, the design of a rate 8/10 channel code occupying six RDS values is feasible (as opposed to a code occupying only 5 sum states and having the capacity 0.79). Before analyzing the ETM code in more detail, we briefly discuss the application of DC-free codes.

Binary DC-free codes have been widely applied in optical and magnetic recording systems. DC-free codes have a long history and their application is certainly not confined to recording practice. Since the early days of digital communication over cable, DC-free codes have been employed to counter the effects of low-frequency cut-off due to coupling-components, isolating transformers, etc. In optical recording, for example in the CD player, DC-free codes are employed to circumvent or reduce interaction between the data written on the disk and the servo systems that follow the track. Immunity against low-frequency disturbances, as caused by fingerprints on a CD, for example, is achievable with the aid of DC-free codes. In this case, the encoded sequence itself contains no low-frequency components, so errors of this type can be avoided by high-pass filtering.

ETM encoding is performed with the aid of a codebook comprised of two pages. The pages are associated with two encoder states, denoted by  $s_0$  and  $s_1$ . Each page has 256 entries which enables a one-to-one state-dependent encoding. Depending on the specific transmitted codeword, the encoder state either alters, from  $s_0$  to  $s_1$  (or from  $s_1$  to  $s_0$ ), or remains in the same state. State-dependent encoders can be described by trellis diagrams. The *trellis diagram* depicts the connections allowed between the encoder states along the time axis. The trellis diagram describing the ETM encoding rule is shown in Fig. 1. The

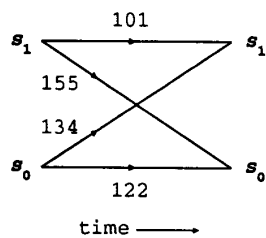


Figure 1: Trellis diagram of the ETM encoder

code page associated with encoder state  $s_0$ , for example, consists of 122 codewords keeping the encoder in  $s_0$ , and 134 codewords altering the encoder state to  $s_1$ . The number of allowed connections from state  $s_{i-1}$  to state  $s_j$  in the trellis diagram, where  $i, j \in \{1, 2\}$ , determines the

element contained in the  $i$ -th row and the  $j$ -th column of the connection matrix. The *connection matrix* of the ETM code (see Fig. 1) is given by

$$D_{ETM} = \begin{pmatrix} 122 & 134 \\ 155 & 101 \end{pmatrix}. \tag{3}$$

Up to now, we have considered the ETM state-dependent encoding. In the following, we will focus on the ETM micro-structure. We consider the RDS of the encoded sequence and we use the simple recursive relation (2) to construct the RDS trellis. The *RDS trellis* describes all fluctuations of the RDS which are allowed to occur in the encoded sequence. Each ETM codeword realizes a path through the RDS trellis shown in Fig. 2. The RDS trellis

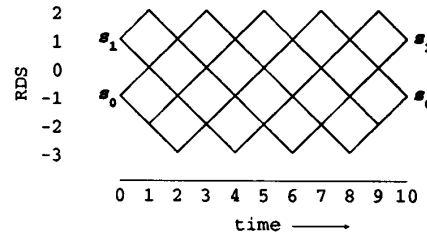


Figure 2: RDS trellis underlying the ETM code

in Fig. 2 has two principal states, denoted by  $s_0$  and  $s_1$ . The principal states of the RDS trellis correspond to the states of the ETM encoder; see Fig. 1. As we can see in Fig. 2, the ETM code occupies six sum states. Furthermore, we recognize that the ETM codewords which do not alter the encoder state contain equally many ‘zeros’ and ‘ones’. Some of these codewords are allowed to occur in both pages of the codebook (89 of these codewords occur in both code pages [1], thus the ETM code comprises 423 distinct codewords). The ETM codewords which alter the encoder state contain different numbers of ‘zeros’ and ‘ones’. They occur in only one page of the codebook. The number of allowed connections from the principal state  $s_{i-1}$  to  $s_j$  in the RDS trellis, where  $i, j \in \{1, 2\}$ , determines the  $ij$ -th element of the connection matrix. The connection matrix of the RDS trellis in Fig. 2 is given by

$$D_{RDS} = \begin{pmatrix} 197 & 155 \\ 155 & 131 \end{pmatrix}. \tag{4}$$

We have characterized the ETM encoder and the RDS trellis in Fig. 2 by connection matrices (3) and (4). Comparison of these matrices shows that each element in  $D_{ETM}$  is less or equal to its corresponding element in  $D_{RDS}$ . Obviously, there are more paths through the RDS trellis than there are ETM codewords. In other words, the ETM code is a subset of the set containing all paths through the underlying RDS trellis. About 20% (126 of 638) of the paths through the RDS trellis are not codewords and are not used for coding.

As we have seen, the ETM code is designed in order to exhibit a spectral null at the zero frequency. In addition, the ETM code is assembled to comply with the following design considerations:

- Simple encoding and decoding is possible, permitting simple low-cost design.
- The concatenated channel string takes on six digital sum values.
- The maximum run-length is 5 channel symbols, i.e., at most 5 consecutive like symbols can occur.
- State-independent decoding is possible. Decoding can be accomplished without using the actual value of the digital sum. In this way, error propagation is limited to 8 decoded source bits.
- Encoding and decoding can be realized by means of the codebook, i.e., via table look-ups. Enumerative techniques, as described in [1], can also be used.

Distance properties, as desired for error correcting codes, have not been considered in the code design. On the other hand, the construction of RDS-constrained codes already implies specific distance properties. The distance properties of the ETM code are found by inspection of the RDS trellis in Fig. 2. Two arbitrary ETM codewords differ in at least one bit position, i.e., the *minimum Hamming distance* is equal to 1. The minimum Hamming distance between two distinct encoded sequences, the *free Hamming distance*, is

$$d_{free}^H = 2. \quad (5)$$

The free Hamming distance is equivalent to the minimum Hamming distance between two distinct paths through the RDS trellis which emanate from a common sum state and terminate (after an arbitrary number of branches) in a common sum state. We can benefit from the free Hamming distance by using sequence estimation algorithms such as the well-known *Viterbi algorithm*. In later sections, we will match the Viterbi algorithm to the requirements of the ETM code. In the next section, we will consider the dominant disturbances encountered in the DCC system.

### III. PROBLEM DESCRIPTION

In the current DCC player, a straightforward hard decision threshold detection scheme is applied which does not exploit the distance properties of the ETM code. More advanced detection methods are considered which might improve the system performance. We start by analyzing the sources of possible errors. In the DCC system, three dominant error sources can be distinguished:

- In stationary-head digital magnetic tape recorders such as the DCC recorder, bit slips, i.e., insertions or deletions of bits, occur frequently [9]. As a result of a bit slip, the number of detected (or written) bits differs from the number of bits on tape. Bit slips are due to tape-to-head speed variations, as caused by shocks for example. The occurrence of bit slips depends on, among other things, the channel bit rate, the properties of mechanical system and clock recovery circuitry, and the system environment.
- In the current DCC player, no specific erase process is applied. Overwrite (and adjacent-track) noises amount to low-frequency additive noises [2].

- All other types of noise, such as those caused by read amplification and equalization or signal amplitude variations due to timing jitter, are modeled as additive white Gaussian noise (AWGN).

The Viterbi algorithm, as a realization of Maximum Likelihood Sequence Estimation (MLSE) in the presence of AWGN, is a well-suited detection method when additive, approximately white, Gaussian noise, or low-frequency additive noise (such as signal offsets) are the dominant sources of signal disturbances. In the next section, we give a brief tutorial introduction to MLSE, including an explanatory example of the Viterbi algorithm.

### IV. INTRODUCTION TO MAXIMUM-LIKELIHOOD DETECTION

The maximum-likelihood principle is a general method of decision and estimation, derived from probability theory. Provided all codewords are equally likely to occur, then applying the maximum-likelihood principle leads to minimum decision error rate. Tutorial descriptions of the maximum-likelihood principle can be found in, for example, [3], or [4]. We will use the maximum-likelihood principle for the extraction of a sequence of information symbols from a noisy sequence at the receiver, i.e., we consider Maximum Likelihood Sequence Estimation (MLSE).

We assume binary, antipodal signaling. Let  $\{c_k\} = \{c_0, c_1, \dots, c_k, \dots\}$ , where  $c_k \in \{-1, 1\}$ , denote the encoded sequence transmitted (i.e. a channel symbol  $\mathbf{x}_k$ , see (1), corresponds to  $c_k = 2\mathbf{x}_k - 1$  transmitted). We assume zero-mean AWGN as the only channel impairment. At the receiver, we obtain a sequence of analog samples  $\{r_k\} = \{c_k\} + \{n_k\}$ , where  $\{n_k\}$  is a sequence of uncorrelated noise samples induced by the AWGN channel. As an estimate for the transmitted sequence  $\{c_k\}$ , MLSE determines that sequence  $\{\hat{c}_k\}$  among the set  $\mathcal{C}$  of all allowed encoded sequences which is closest to the received sequence  $\{r_k\}$  in terms of squared Euclidean distance. In other words, MLSE determines the sequence  $\{\hat{c}_k\}$  satisfying

$$\min_{\{\hat{c}_k\} \in \mathcal{C}} \sum_{i=0}^k (r_i - c_i)^2. \quad (6)$$

Expanding the term  $(r_i - c_i)^2$  in (6) results in the expression  $r_i^2 - 2r_i c_i + c_i^2$ , where the term  $c_i^2$  is constant,  $c_i^2 = 1$ . The term  $r_i^2$  has no relevance in the sequence  $\{\hat{c}_k\}$  satisfying (6). Hence, the sequence  $\{\hat{c}_k\}$  satisfying (6) equivalently satisfies

$$\max_{\{\hat{c}_k\} \in \mathcal{C}} \sum_{i=0}^k r_i c_i, \quad (7)$$

i.e., it has maximum cross-correlation with the noisy sequence received.

We conclude that in the presence of AWGN, the maximum-likelihood receiver is equivalent to a receiver that minimizes the Euclidean distance to the received sequence. For binary, antipodal signaling, the minimum-distance receiver (6) is equivalent to the maximum-cross-correlation

receiver (7). Performing an algorithm which implements the maximum-cross-correlation receiver (7) requires add, compare, and select operations. Due to  $c_k \in \{-1, 1\}$ , no multiplications are required. The straightforward implementation of the maximum-likelihood receiver according to (7) results in a huge computational complexity for long data sequences, i.e., for  $k$  large. An algorithm which can be feasibly implemented was developed by Viterbi in 1967. We will present here an explanatory example of the Viterbi algorithm. Later, we will apply it to the special case of the ETM code. Tutorial descriptions of the Viterbi algorithm can be found in many textbooks on digital communications, for example, [4].

We assume that all allowed encoded sequences describe paths through a trellis. Trellis examples were presented in the description of the ETM code given in Section II. We are concerned with finding that trellis path whose corresponding encoded sequence has smallest Euclidean distance to the noisy sequence received. As an example, we consider the 2-state trellis depicted in Fig. 3. We start by defining

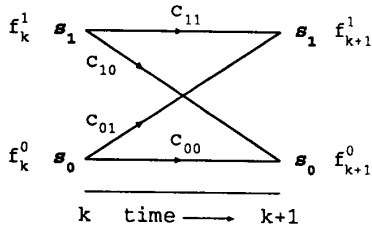


Figure 3: Path metrics of a 2-state trellis

a *path metric* for each of the trellis states. Let  $f_k^i$  denote the path metric associated with trellis state  $s_i$  at time  $k$ , where  $i \in \{0, 1\}$ . An essential characteristic of the Viterbi algorithm is the iterative calculation of the path metrics, i.e., the path metrics at time  $k+1$  are a function of the path metrics at time  $k$ . We consider the trellis branch emanating from state  $s_i$  and entering state  $s_j$ ,  $i, j \in \{0, 1\}$ , which represents the transmission of the channel symbol  $c_{ij}$ , where  $c_{ij} \in \{-1, 1\}$ . We define the path metric increment

$$\Delta f_{k+1}^{ij} = r_{k+1} c_{ij}, \quad (8)$$

associated with this trellis branch, where  $r_{k+1}$  is the sample received at time  $k+1$ . We have defined the path metric increment (8) in accordance with the maximum-cross-correlation receiver (7). Hence, performing the Viterbi algorithm does not require any multiplications. Now we are in the position to define the iterative rules for calculating the path metrics. At time  $k+1$ , the path metric associated with trellis state  $s_0$  is calculated according to

$$f_{k+1}^0 = \max\{f_k^0 + \Delta f_{k+1}^{00}, f_k^1 + \Delta f_{k+1}^{10}\}, \quad (9)$$

and the path metric associated with trellis state  $s_1$  according to

$$f_{k+1}^1 = \max\{f_k^0 + \Delta f_{k+1}^{01}, f_k^1 + \Delta f_{k+1}^{11}\}. \quad (10)$$

We can interpret (9) and (10) as follows: from all possible paths entering a trellis state, we select that path having maximum metric, i.e., the *survivor path*. We determine a survivor path for each of the trellis states. Looking backwards in time, the survivor paths tend to merge after a few branches into a common path. We use this common path for selecting the detector output. In other words, the Viterbi algorithm requires delay to make decisions. In the next section, we will derive the gains in error rate performance which can be expected for the ETM code when MLSE instead of threshold detection is applied.

## V. ASYMPTOTIC CODING GAIN

In order to fully exploit the distance properties of the ETM code, we consider MLSE as detection scheme. We start by approximating the *symbol error probabilities*, i.e., the probabilities that ETM codewords are erroneously detected, for MLSE and threshold detection. We assume the transmitted signal to be a rectangular wave having amplitude  $A$ . We further assume the received samples to be disturbed by zero-mean AWGN with noise variance  $\sigma^2$ . We define the signal-to-noise ratio in decibels as

$$20 \log_{10} (A/\sigma). \quad (11)$$

Let  $Q(x)$  be the tail integral of the Gaussian distribution defined by

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-y^2/2} dy. \quad (12)$$

The symbol error probability for threshold detection can then be approximated by

$$n Q(A/\sigma), \quad (13)$$

where  $n$  is the codeword length,  $n = 10$ . In the case of MLSE, the symbol error rate approximately equals

$$N Q(\sqrt{d_{free}^H} A/\sigma), \quad (14)$$

where the *error coefficient*  $N$  is the average number of sequences having free Hamming distance to any encoded sequence at any time. Both approximations in (13) and (14) fit well for moderate and high signal-to-noise ratios. Derivations of (13) and (14) can be found in, for example, [4].

We define the *coding gain* as the improvement in signal-to-noise ratio given a constant error rate when MLSE instead of threshold detection is applied. The *asymptotic coding gain* denotes the coding gain for signal-to-noise ratios approaching infinity. The error coefficient  $N$  in (14) has a serious influence on the symbol error rate in the range of moderate signal-to-noise ratios, between 5 dB and 10 dB, for example. In the range of high signal-to-noise ratios, the influence of the error coefficient on the symbol error rate is rather small. The asymptotic coding gain is not influenced by the error coefficient. Hence, the asymptotic coding gain is obtained by comparing the Gaussian tail integrals in (13) and (14). Using (5), the asymptotic coding gain is

$$10 \log_{10} d_{free}^H = 3 \text{ dB}. \quad (15)$$

In the following section, we will focus on the realization of two distinct sequence estimation algorithms. The first scheme considered comprises a fully-fledged Viterbi algorithm which realizes MLSE. The second algorithm implements a suboptimal detection scheme.

## VI. REALIZATION OF THE DETECTION ALGORITHMS

As we have seen in Section II, the ETM code is a subset of the complete set of paths through the RDS trellis in Fig. 2. Hence, a Viterbi algorithm based on the RDS trellis results in a detection scheme which is not optimal in the maximum-likelihood sense. A fully-fledged Viterbi algorithm realizing MLSE can be based on the ETM encoder trellis which was shown in Fig. 1. In the following, both detection schemes are considered and the computational complexities are estimated. The fully-fledged scheme turns out to be a wordwise algorithm. The suboptimal scheme will comprise a bitwise algorithm. We assume perfect synchronization for both schemes.

### A. Fully-Fledged Viterbi Algorithm

We start by considering the fully-fledged Viterbi algorithm which realizes MLSE. The fully-fledged Viterbi algorithm is based on the 2-state ETM encoder trellis shown in Fig. 1. Each stage of the trellis corresponds to the transmission of an ETM codeword, i.e., we realize MLSE with a wordwise Viterbi algorithm.

Let  $C$  denote the ETM codebook. We split the ETM codebook according to  $C = C_{00} \cup C_{01} \cup C_{10} \cup C_{11}$  into four subsets, where  $C_{ij}$  denotes that set of ETM codewords altering the encoder state from  $s_i$  to  $s_j$ ,  $i, j \in \{0, 1\}$ . Let  $r_k = (r_{k1}, r_{k2}, \dots, r_{kn})$ , where  $n = 10$ , denote the noise-corrupted ETM codeword received at time  $k$ . In order to realize the maximum-cross-correlation receiver (7), we define the path metric increment as

$$\Delta f_k = \sum_{j=1}^{n=10} r_{kj} c_j, \quad (16)$$

where  $c = (c_1, c_2, \dots, c_n)$ ,  $c_j \in \{-1, 1\}$ , denotes an ETM codeword (in bipolar notation), i.e.,  $c \in C$ . Now we define the iterative rules for calculating the path metrics. At time  $k + 1$ , the path metric associated with trellis state  $s_0$  is calculated according to

$$f_{k+1}^0 = \max\{f_k^0 + \max_{C_{00}}\{\Delta f_{k+1}\}, f_k^1 + \max_{C_{10}}\{\Delta f_{k+1}\}\}, \quad (17)$$

and the path metric associated with trellis state  $s_1$  according to

$$f_{k+1}^1 = \max\{f_k^0 + \max_{C_{01}}\{\Delta f_{k+1}\}, f_k^1 + \max_{C_{11}}\{\Delta f_{k+1}\}\}. \quad (18)$$

Calculating the path metric increment according to (16) requires ten additions per ETM codeword. The ETM code has the property that 423 of the 512 codewords are distinct [1]. Hence, at least 4230 additions are required for calculating all distinct path metric increments according to (16). Moreover, to calculate the path metric increments

(16), we have to read the 423 distinct ETM codewords from the code table. In order to evaluate (17) and (18), we have to determine the maxima among the path metrics (or increments) which requires at least 421 comparisons and further 4 additions. The fully-fledged algorithm can immediately combine detection and decoding, i.e., no separate decoding step, such as a table look-up, is required. We conclude that the maximum-likelihood detection per transmitted ETM codeword, including decoding, requires a computational complexity of at least 4234 additions, 421 comparisons, and 423 table look-ups.

### B. Suboptimal Trellis Detection

The suboptimal trellis detection scheme consists of a Viterbi algorithm which is based on the RDS trellis underlying the ETM code as shown in Fig. 2. In other words, we determine the most likely path through the RDS trellis. Since the ETM code is a subset of the set containing all paths through the RDS trellis, the most likely RDS trellis path is not necessarily a valid ETM codeword. Therefore, this detection scheme is not optimal in the maximum-likelihood sense. We can approximate the symbol error rate expected for the suboptimal detection algorithm in the presence of AWGN by using (14), where the error coefficient  $N$  is increased compared to the error coefficient for MLSE. The free Hamming distance exploited by the suboptimal detection algorithm given in (5) remains the same as for MLSE. In order to realize the suboptimal scheme, we split the RDS trellis in Fig. 2 into ten 1-bit trellises, which results in a bitwise Viterbi algorithm. Each 1-bit trellis has three states. Two different 1-bit trellises for odd and even instants are distinguished. As an example, two 1-bit trellises are shown in Fig. 4. At the codeword boundaries

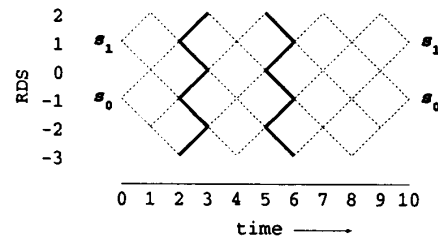


Figure 4: Two 1-bit trellises for suboptimal detection algorithm

only two survivor paths, ending in the principal states of the RDS trellis in Fig. 2, are considered. Since the detector output consists of a path through the RDS trellis, an additional decoding step is required.

The path metric calculation is realized as described in Section IV (see (7), (8), (9), and (10)), with the exception that the trellises here exhibit three states. Each 1-bit trellis in Fig. 4 comprises five branches. The RDS trellis in Fig. 2 consists of 48 branches. Performing the Viterbi algorithm requires a single addition per branch; see (7). The number of branches of the RDS trellis hence determines the minimum number of additions required to detect an ETM codeword. The minimum number of comparisons required

to detect an ETM codeword is determined by the number of branch pairs entering a common sum state. Both 1-bit trellises in Fig. 4 exhibit two branch pairs entering a common sum state. The RDS trellis in Fig. 2 exhibits 19 such branch pairs. We conclude that performing the suboptimal trellis detection algorithm requires a computational complexity of at least 48 additions and 19 comparisons per detected ETM codeword. No code table look-ups are required for detection. We assume a table look-up for decoding. Compared to the fully-fledged scheme, this suboptimal detection algorithm leads to a large reduction in computational complexity. Considering detection and decoding, reduction factors of about 85 for the number of additions, 20 for the number of comparisons, and 423 for the number of code table look-ups can be given. The use of RDS trellises for sequence estimation in order to decrease the computational complexity is also described in [7] and [8].

### VII. SIMULATION RESULTS

The error rate performances of the fully-fledged Viterbi algorithm and of the suboptimal trellis detection algorithm in the presence of AWGN were determined by means of computer simulations. Fig. 5 shows the resulting sym-

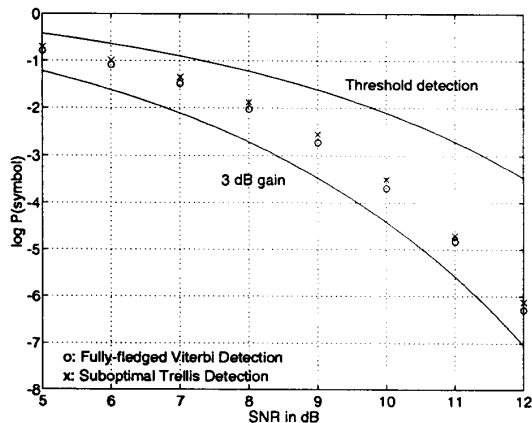


Figure 5: Symbol error probabilities for fully-fledged Viterbi detection (o) and suboptimal trellis detection (x)

bol error probabilities versus the signal-to-noise ratio, as defined by (11), in the range from 5 dB to 12 dB. The upper line in Fig. 5 represents, according to (13), the error rate performance expected for threshold detection. The line below corresponds to a gain of 3 dB versus the error rate performance expected for threshold detection; see (15). The error rate performance obtained for the fully-fledged Viterbi algorithm is given by the small circles in Fig. 5 and by the small crosses for the suboptimal detection algorithm. As we can see in Fig. 5, the fully-fledged Viterbi algorithm achieves a coding gain of approximately 2.3 dB for symbol error rates in the order of  $10^{-4} \dots 10^{-5}$ . The suboptimal detection algorithm results in only a slight

degradation in error rate performance compared to the fully-fledged scheme. For symbol error rates in the order of  $10^{-4} \dots 10^{-5}$ , less than a quarter of a dB are lost compared to the fully-fledged algorithm. Asymptotically, the symbol error rates for both algorithms appear to converge to coding gains of 3 dB.

### VIII. EXPERIMENTAL RESULTS

The application of sequence estimation algorithms is only advantageous if bit slips occur rarely on the channel. For the eight main data tracks of the DCC, this condition is not met [9]. The relatively low bit rate of the auxiliary data (see Section I) suggests that bit slips here should seldom occur. Therefore, only the auxiliary channel is considered. About  $10^6$  samples of auxiliary data measured from an overwritten recording were examined. All samples were taken from the same cassette. Therefore, the measured raw bit error rate (threshold detection) of about  $3 \cdot 10^{-5}$  is only representative for this recording. All errors found were pattern-dependent single additive errors, not random errors. No bit slips could be detected. By using the synchronization strategy described in [9], synchronization could always be realized. Both developed sequence estimation algorithms avoided all errors that would have been made with threshold detection.

### IX. CONCLUSIONS

Although the ETM code has not been designed in order to exhibit distance properties, an asymptotic gain of 3 dB versus threshold detection level can be achieved by applying soft decision sequence estimation algorithms (AWGN assumed). With MLSE an approximately 2.3 dB gain versus threshold detection level for symbol error rates in the order of  $10^{-4} \dots 10^{-5}$  is obtained. A suboptimal detection algorithm which is based on the RDS trellis underlying the ETM code was developed. Compared to MLSE, this suboptimal algorithm leads to a large reduction in computational complexity and only to a slight degradation in error rate performance. In the DCC system, either of the considered sequence estimation algorithms can be applied on the auxiliary data track where bit slips does not seem to be the dominant source of errors.

### ACKNOWLEDGMENTS

This research project was carried out at the Philips Research Laboratories in Eindhoven, The Netherlands. The first author would like to thank the members of the Magnetic Recording Group for fruitful discussions and comments.

### REFERENCES

- [1] K.A. Schouhamer Immink, *Coding Techniques for Digital Recorders*, Prentice Hall International (UK) Ltd, 1991.
- [2] C.D. Mee, and E.D. Daniel, *Magnetic Recording*, volume II, McGraw-Hill Book Company, New York, 1988.
- [3] J.M. Wozencraft, and I.M. Jacobs, *Principles of Communication Engineering*, John Wiley & Sons, Inc., New York, 1965.
- [4] R.E. Blahut, *Digital Transmission of Information*, Addison-Wesley Publishing Company, 1990.

- [5] G.L. Pierobon, "Codes for Zero Spectral Density at Zero Frequency," *IEEE Trans. Inform. Theory*, vol. IT-30, no. 2, pp. 435-439, March 1984.
- [6] P.H. Siegel, and J.K. Wolf, "Modulation and Coding for Information Storage," *IEEE Comm. Magazine*, vol. 29, no. 12, pp. 68-86, December 1991.
- [7] H. Thapar, J. Rae, C. Shung, R. Karabed, and P. Siegel, "On the Performance of a Rate 8/10 Matched Spectral Null Code for Class-4 Partial Response," *IEEE Trans. Magn.*, vol. MAG-28, no. 5, pp. 2883-2888, September 1992.
- [8] R. Wood, "Viterbi Reception of Miller-Squared Code on a Tape Channel," *Proc. of the Fourth Intern. Conf. on Video and Data Recording*, pp. 333-343, Southampton, April 1982.
- [9] M.A. Ribeiro, "Optimal Bit-Level Synchronization Strategy for Digital Magnetic Recorders," *Proc. of the Fourteenth Symposium on Information Theory in the Benelux*, pp. 222-227, May 1993.
- [10] G.C.P. Lokhoff, "dcc-Digital Compact Cassette," *IEEE Trans. Consumer Electronics*, vol. CE-37, no. 3, pp. 702-706, August 1991.
- [11] G.C.P. Lokhoff, "Precision Adaptive Subband Coding (PASC) for the Digital Compact Cassette (DCC)," *IEEE Trans. Consumer Electronics*, vol. CE-38, no. 4, pp. 784-789, November 1992.

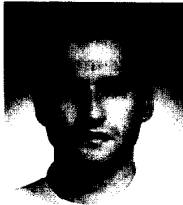


**Kees A. Schouhamer Immink (M'81-SM '86-F'90)** was born in Rotterdam, The Netherlands, on December 18, 1946. He received the B.S. degree from the Rotterdam Polytechnic in 1967, and the M.S. and Ph.D degrees from the Eindhoven University of Technology in 1974 and 1985.

Immink joined the Philips' Research Laboratories in Eindhoven in 1968, where his work primarily concerned the signal processing side of optical recording systems, later becoming responsible for the design and development of coding techniques for the Compact Disc, Compact Disc Video, experimental erasable optical audio discs, R-DAT, and DCC recorders.

Immink holds more than thirty patents, mainly in the area of digital recording, and he has written numerous papers in the field of coding techniques for optical and magnetic recorders. He is author of the monograph *Coding Techniques for Digital Recorders* and co-author of *Principles of Optical Disc Systems and Reed Solomon Codes: Theory and Application*.

He was elected a Fellow of the AES, IEE, IEEE, and was awarded the AES Silver Medal in 1992, and the IEE Sir J.J. Thomson Medal in 1993.



**Milton A. Ribeiro (S'90-M'92)** received his B.S.E.E. degree from the Instituto Tecnológico de Aeronautica (ITA), S.J. Campos, Brazil, in December 1989, and his M.S.E.E. degree from the Eindhoven International Institute (EII) in May 1991, both in Electronic Engineering.

From 1991 until 1993 he was a member of the technical staff of Philips Research Laboratories in Eindhoven, The Netherlands, where he took part in the development of the Digital Compact Cassette (DCC). He joined the

Philips Advanced Communication Enterprise in January, 1994, where he is involved with the development of agent-based communications for the european operators.

He is a member of the IEEE and its Communications society.



**Volker Braun (S'93)** was born in Neunkirchen, Germany, in 1967. He received the Dipl.-Ing. degree in electrical engineering from the University of Kaiserslautern in 1992.

In September 1992, he joined the Digital Communications Group at the Institute for Experimental Mathematics at the University of Essen. He is engaged in problems of modulation, coding, and detection, especially for recording systems.



**Gijs J. van den Enden** was born in Papendrecht, The Netherlands, on August 13, 1964. He received the B.S. degree from the Dordrecht Polytechnic in 1986.

Van den Enden joined the Philips Research Laboratories in Eindhoven in 1988. His work concerned digital magnetic recording systems such as HDTV studio recorders, DCC recorders and linear recorders.