FACTA UNIVERSITATIS (NIŠ)
Series: Electronics and Energetics vol. 1(1995), 57–72

DISTANCE SPECTRUM OF CHANNEL TRELLIS CODES ON PRECODED PARTIAL-RESPONSE 1-D CHANNEL

Miroslav Despotović and Vojin Šenk

Abstract. Distance spectrum is the main factor in determining the event error probability when maximum-likelihood decoding is used for convolutional codes [13]. In this paper we compute the distance spectrum for the best R=1/N channel codes obtained using convolutional (linear trellis) codes together with known offset sequence addition on precoded partial-response $1-D$ channel proposed in [4]. To this end we have modified the BIFAST algorithm [11] in order to enable the distance spectrum evaluation in the supper-trellis [1], a technique necessary for this type of channel codes.

1. Introduction

Recent works on achieving high data rates on intersymbol (ISI) channels analyse the use of a technique referred to as PRML (PR - partial-response signalling with ML - maximum likelihood decoding). For combatting ISI this approach, instead of keeping the transitions far apart using (d, k) codes [9], allows the transitions to be close together, and the received signal with its resulting ISI is equalized to a predetermined impulse response, like the one known to be present in the class-4 partial-response channel [5]. Since this channel is equivalent to a memoryless one preceded by a trellis code, the equalized signal may then be detected by an ML sequence estimator, e.g., Viterbi detector [3]. The pre-Viterbi detector equalization does not greatly increase the probability of erroneous symbol detection, while the complexity of the Viterbi detector is significantly reduced.

Magnetic recording may also be viewed as an ISI-corrupted transmission [10]. For magnetic recording, little equalization is required to force

Manuscript received February 4, 1995.
The authors are with the Faculty of Technical Sciences, Trg Dositeja Obradovića 6, 21000 Novi Sad.

the equalized channel to match the class-4 partial-response channel (PR4), whose impulse response is $h(D) = 1 - D^2$ (D - unit delay operator). This channel can be considered as two time-interleaved "dicode" partial response channels, each with polynomial $h(D) = 1 - D$, [10]. A simplified method for ML sequence detection for the PR4 channel would thus consist of deinterleaving the samples of the readback waveform and to apply Viterbi detectors matched to the $(1 - D)$ channel, one for each deinterleaved data stream.

The PR4 channel, viewed as a trellis code, does not have sufficient error-protecting capabilities for transmission (recording) purposes. In order to amend this situation, the input to this channel may itself be trellis-coded. The cascade of the two trellis codes is equivalent to a unique trellis code. Hole and Ytrehus [4] gave a nice review of papers on coding techniques for improving the reliability of digital transmission (recording) over a noisy PR channel. They also gave a number of new binary convolutional (linear trellis) codes that, together with known offset sequence addition and precoding, give powerful overall trellis codes when used on the $1 - D$ channel. It is the aim of this paper to determine the performance of these codes upon optimal (Viterbi) decoding.

The performance of a trellis code depends on the decoding algorithm employed and on the properties of the code. The distance spectrum is the property of the code that constitutes the main factor of the event error probability of a maximum-likelihood (optimum) decoder, if the distance is appropriately chosen for the coding channel used [13], [12]. For instance, Hamming distance and squared Euclidean distance **(s.E.d.)** are the appropriate distance measures for the binary symmetric channel (BSC) and the additive white Gaussian noise (AWGN) channel, respectively.

The minimum distance of a channel code (also called the free distance in the case of trellis codes) was used for a long time as the only parameter for judging the suitability of the code for information transmission over a noisy channel. The use of distance spectrum of a trellis code for bounding its performance [13], although known since 1971, has found widespread application only recently, after a number of efficient algorithms for its evaluation have been found [2], [8], [11]. The sole use of the free distance yields relatively good (lower) bounds on the error probability only at code rates that are much smaller than the channel capacity (usually designated as the high signal-to-noise ratio (SNR) region), but largely underestimate it otherwise. On the contrary, the (upper) bounds obtained using the distance spectrum are useful at a much wider range of code rates.

In Section 2 some introductory concepts and definitions are given. An illustrative example is given in Section 3. In Section 4 we give a brief description of super-trellis construction that is used for determining distance spectrum of non-regular codes [8]. Finally, Section 5 contains results for rate $R = 1/N$ channel encoders.

2. Convolutionally Coded PR Channel

We will now review some of definitions of the basic terms referring to the binary convolutionally coded partial-response channel.

Let F represent the finite field $GF(2)$. The configuration of a coded $1 - D$ PR channel is depicted in Figure 1. At time t the $(N, K, \mathcal{M}_c)$ binary convolutional encoder of memory $\mathcal{M}_c$ takes a K-tuple of bits $i_t = [i_t^1, \dots, i_t^K] \in F^K$ together with $\mathcal{M}_c$ previous K-tuples $i_{t-1}, \dots, i_{t-\mathcal{M}_c}$, and generates an N-tuple of coded bits $x_t = [x_t^1, \dots, x_t^N] \in F^N$. The binary input sequence to the encoder is represented by $I = [i_0, i_1, \dots]$, while the corresponding output sequence is similarly represented by $X = [x_0, x_1, \dots]$. The binary convolutional encoder is defined by a 1 by N generator matrix $G(D) = [G^1(D), \dots, G^N(D)]$, $\deg(G^n(D)) = \mathcal{M}_c$, [4]. For example, consider the $(5, 1, 1)$ convolutional code defined by the generator matrix $G(D) = [1, 0, 0, 0, D]$. This code will be used in the Section 3 as the underlying convolutional code for constructing a good overall trellis code (see Figure 2). Alternatively, this matrix may be given in the usual octal notation [6, pp. 329–337] as $G = [2, 0, 0, 0, 1]$, since $2_8 = 10_2 = 1 + 0 \cdot D$ and $1_8 = 01_2 = 0 + 1 \cdot D$. This compact notation will be used in Table 1.

Let C be any $(N, K, \mathcal{M}_c)$ convolutional code and let $A = [a_0, a_1, \dots]$, $a_i \in F^N$, be an arbitrary sequence that is not in C. The set

$$C \oplus A = \{X \oplus A | X \in C\}, \qquad A \neq 0 \tag{1}$$

is a coset of C ($\oplus$ denotes componentwise addition (in F) of binary sequences). The offset sequence A is a coset representative (not unique). In this paper we shall be interested in coset representatives of the form $A = \overline{p}$ where $p = [p^1, \dots p^N]$ is an N-tuple in F^N, and $\overline{p}$ denotes the period sequence of period N obtained by repeating the N-tuple p. A trellis represanting the sequence $C \oplus A$ is called the **coset trellis**.

In Figure 1 a coset representative $\overline{p}$ is added to the coded sequence X (to ensure short maximum zero-run at the channel output). The coset sequence $X_c = X \oplus \overline{p}$ is passed through the precoder with transfer function

$1/(1 \oplus D)$, resulting in the binary sequence X_p. This sequence is passed through a $1 - D$ partial-response channel producing the ternary sequence X_{PR} with coefficients in $\{0, \pm 1\}$. Zero mean i.i.d. Gaussian noise n is added to X_{PR} resulting in the real number sequence Y. The Viterbi decoder uses Y to determine a maximum likelihood estimate $\hat{I}$ of the input sequence I.

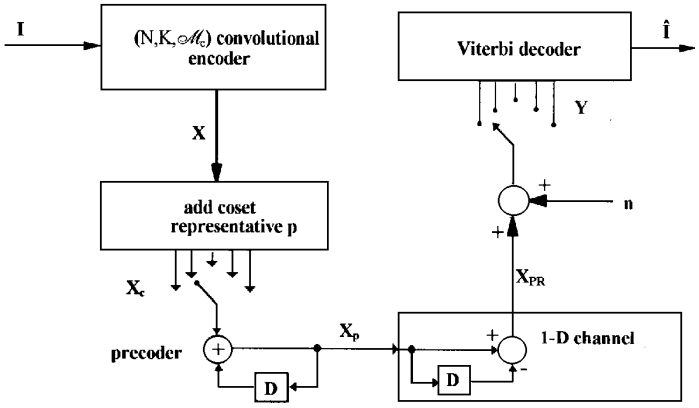

Fig. 1. Convolutionally coded PR chanel

The set of noiseless output sequences X_{PR} from $1-D$ channel in Figure 1 forms an overall ternary non-linear trellis code called the **channel code**. The trellis for the channel code is called the **decoder trellis**.

The free *s.E.d.* at the channel output (the minimum *s.E.d.* between any two paths through the decoder trellis) is lower bounded by the free Hamming distance of the $(N, K, \mathcal{M}_c)$ convolutional code [14]. The lower bound suggests the use of a convolutional code with maximal free Hamming distance for given rate R and number of decoder states S. Hole and Ytrehus [4] determined cosets of convolutional codes that, together with $1/(1 \oplus D)$ precoder, generate $(N, K, \mathcal{M}_c)$ trellis codes with free *s.E.d.* significantly larger then the free Hamming distance of the best $(N, K, \mathcal{M}_c)$ convolutional code with the same number of states in the Viterbi decoder (hereafter denoted as HY-codes). These cosets exploit the channel memory in such a way that the free *s.E.d.* is enhanced, i.e. the convolutional codes used *are not* optimized for Hamming distance. Also, it is important to note that codes tabulated in Section 5 are said to be **trellis matched** to the $1 - D$ channel [4], if the channel code memory is unexpanded, i.e. if $\mathcal{M} = \mathcal{M}_c$. The number of

decoder states is unaffected by the channel memory in this case.

The **distance spectrum** of a trellis code is a sequence of ordered pairs (d_i, M_i), $i = 1, 2, \ldots$, where $d_i < d_{i+1}$ is the i-th distance among all the paths that diverge from the same state (of the code memory) in the trellis in one moment, and remerge $\mathcal{M} + 1$ or more branches later ($d_1 = d_{free}$). One of these paths is called the **reference** path, the other the **concurrent** path.

A union bound on the first event error probability P_e of channel codes may be obtained by summing the error probability over all possible incorrect paths which remerge with all possible correct paths. At any time unit, P_e is bounded by [8], i.e.

$$P_e \leq \sum_{i=1}^{\infty} M_i Q\Big(\sqrt{\frac{d_i}{2\mathcal{N}_0}}\Big), \tag{2}$$

where d_i represents the i-th $s.E.d.$ between signal sequences, M_i is the (average) multiplicity of the i-th spectral line, i.e. of codewords at distance d_i from a specific codeword, $\mathcal{N}_0$ is the one-sided noise power spectral density, and $Q(\cdot)$ is the Gaussian integral function

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-\frac{t^2}{2}} dt. \tag{3}$$

As stated above, this upper bound is useful at moderate to high signal-to-noise ratios.

3. An Illustrative Example

As an example, let us consider the $(N, K, \mathcal{M}_c) = (5, 1, 1)$ convolutional code defined by the generator matrix $G(D) = [1, 0, 0, 0, D]$ whose $d_{Hfree} = 2$. A coset of the convolutional code is obtained by adding the coset representative $\overline{p}$, where $p = [0, 1, 1, 1, 1]$, to all codewords in the code. The convolutional encoder, together with coset, precoded and decoder trellises is shown in Figure 2. It can be seen that the generated $(5, 1, 1)$ channel code has free $s.E.d.$ equal to $d_1 = 26$ with multiplicity $M_1 = 0.25$. If the convolutional code were optimized for maximum free Hamming distance, $(G(D) = [1, 1+D, 1+D, 1+D, 1+D,])$, $d_{Hfree} = 9$, the free $s.E.d.$ for the equivalent channel code would be only $d_1 = 9$ with any coset representative.

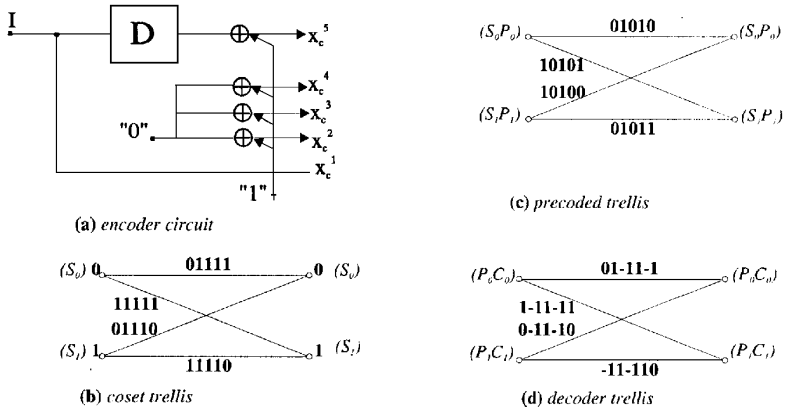

Fig. 2. Convolutional code with $R = 1/5$ $G = [2, 0, 0, 0, 1]$ on the $(1 - D)$ PR channel, S-encoder state, P precoder state and C-channel state

4. Super-trellis

Sometimes, computing the distance spectrum is not possible in the convenient manner searching the path closest to any reference path (usually the all-zero one), because the trellis code is neither regular nor quazi-regular [8]. In these cases, the search has to be made for all paths as the reference ones instead. For small encoder memories, the most efficient method is to create a super-trellis [1, pp. 461–467]. An example of a super-trellis construction for the simple two-state code trellis is given in Figure 3. The super-trellis has S^2 states, while the original trellis has S states.

Every super-state consists of a pair of states in which the first element is the state of the trellis that spans the set of reference paths (the reference trellis), and the second element is the state of the trellis containing concurrent paths (the concurrent trellis). The procedure is not to find the diverge/merge path with minimum accumulated distance from the unique reference path, but to find a path with minimum accumulated distance that diverges from a reference super-state (S_i, S_i) into some non-reference super-state and later remerge to a reference one (it need not be the same one as the breakout reference super-state). Super-trellis branches are labeled with the *s.E.d.* between signals assigned to the corresponding branches of the reference and concurrent trellis.

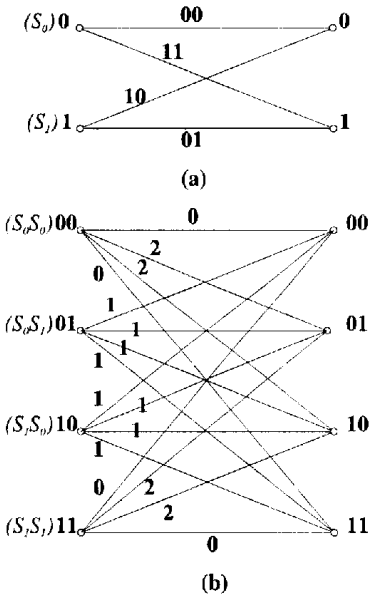

Fig. 3. (a) An example of the two-state trellis and (b) its corresponding super-trellis

For example, transition from the super-state (S_0, S_1) to the super-state (S_1, S_1) is labeled with the distance between signals on the branch $S_0 \to S_1$ (in the reference path trellis), i.e., signal 11, and branch signal $S_1 \to S_1$ (in the concurrent trellis), i.e., signal 01. This distance is $(1-0)^2 + (1-1)^2 = 1$. The super-trellis search is done with the aid of the modified BIFAST algorithm [11]. The original algorithm was designed for regular or quazi-regular trellises, where there exists only one reference path. If a regular code would be analysed via the super-trellis technique, each spectral line would be enumerated $S2^{Kl}$ times, where l is the length of the error event, otherwise not accounted for in evaluating the code distance spectrum. The average (over all reference paths) distance spectrum is obtained normalizing the results obtained with this number, and then summing over all l.

5. Results

Distance spectra results for $1/N$ HY-codes are given in Table 1. These results are illustrated in the next five figures giving some interesting comparisons.

Table 1. Distance spectrum of channel codes trellis matched tothe $1 - D$ channel

No.1 $p = 17$	$\mathcal{M}_c = 1, R = 1/5$ $G = [2, 0, 0, 0, 1]$	No.2 $p = 27$	$\mathcal{M}_c = 2, R = 1/5$ $G = [0, 4, 0, 1, 0]$	No.3 $p = 27$	$\mathcal{M}_c = 3, R = 1/5$ $G = [0, 15, 0, 0, 7]$
d_i	M_i	d_i	M_i	d_i	M_i
26	$2.500000000e-01$	28	$2.500000000e-01$	38	$5.000000000e-01$
28	$2.500000000e-01$	30	$6.250000000e-02$	42	$1.500000000e+00$
30	$7.500000000e-01$	32	$5.000000000e-01$	46	$1.125000000e+00$
32	$7.500000000e-01$	34	$5.000000000e-01$	48	$2.500000000e-01$
34	$1.000000000e+00$	36	$2.500000000e-01$	50	$6.250000000e-01$
36	$1.000000000e+00$	38	$8.750000000e-01$	52	$1.000000000e+00$
38	$1.000000000e+00$	40	$2.500000000e-01$	54	$8.750000000e-01$
40	$1.000000000e+00$	42	$5.000000000e-01$	56	$8.750000000e-01$
42	$1.000000000e+00$	44	$5.000000000e-01$	58	$1.125000000e+00$
44	$1.000000000e+00$	46	$3.125000000e-01$	60	$7.500000000e-01$
46	$1.000000000e+00$	48	$5.000000000e-01$	62	$3.500000000e+00$
48	$1.000000000e+00$	50	$6.250000000e-01$	64	$1.562500000e+00$
50	$1.000000000e+00$	52	$7.500000000e-01$	66	$5.125000000e+00$
52	$1.000000000e+00$	54	$1.250000000e+00$	68	$2.500000000e+00$
54	$1.000000000e+00$	56	$9.375000000e-01$	70	$4.632812500e+00$
56	$1.000000000e+00$	58	$2.000000000e+00$	72	$5.765625000e+00$
58	$1.000000000e+00$	60	$1.750000000e+00$	74	$6.335937500e+00$
60	$1.000000000e+00$	62	$1.625000000e+00$	76	$8.687500000e+00$
62	$1.000000000e+00$	64	$2.875000000e+00$	78	$8.636718750e+00$
64	$1.000000000e+00$	66	$1.875000000e+00$	80	$8.710937500e+00$
66	$1.000000000e+00$	68	$2.375000000e+00$	82	$1.435156250e+01$
68	$1.000000000e+00$	70	$3.187500000e+00$	84	$1.264843750e+01$
70	$1.000000000e+00$	72	$2.703125000e+00$	86	$2.439843750e+01$
72	$1.000000000e+00$	74	$3.625000000e+00$	88	$1.933984375e+01$
74	$1.000000000e+00$	76	$4.343750000e+00$	90	$2.968750000e+01$
76	$1.000000000e+00$	78	$5.015625000e+00$	92	$3.099609375e+01$
78	$1.000000000e+00$	80	$6.109375000e+00$	94	$3.914062500e+01$
		82	$6.343750000e+00$	96	$5.212011719e+01$
		84	$8.953125000e+00$	98	$5.715039062e+01$
				100	$6.838476562e+01$
				102	$8.222851562e+01$
				104	$9.068701172e+01$
				106	$1.294423828e+02$
				108	$1.348256836e+02$
				110	$1.827050781e+02$
				112	$1.962065430e+02$
				114	$2.451684570e+02$

Table 1. continued

No.4 $p=17$	$\mathcal{M}_c=4, R=1/5$ $G=[26,0,0,7,0]$	No.5 $p=6$	$\mathcal{M}_c=1, R=1/4$ $G=[2,0,0,1]$	No.6 $p=12$	$\mathcal{M}_c=2, R=1/4$ $G=[0,4,0,1]$
d_i	M_i	d_i	M_i	d_i	M_i
46	$1.875000000e-01$	18	$2.500000000e-01$	20	$2.500000000e-01$
50	$8.750000000e-01$	20	$2.500000000e-01$	22	$6.250000000e-02$
54	$1.250000000e+00$	22	$7.500000000e-01$	24	$5.000000000e-01$
56	$6.250000000e-02$	24	$7.500000000e-01$	26	$5.000000000e-01$
58	$6.250000000e-01$	26	$1.000000000e+00$	28	$2.500000000e-01$
60	$6.250000000e-01$	28	$1.000000000e+00$	30	$8.750000000e-01$
62	$9.375000000e-02$	30	$1.000000000e+00$	32	$5.000000000e-01$
64	$1.750000000e+00$	32	$1.000000000e+00$	34	$6.250000000e-01$
66	$7.500000000e-01$	34	$1.000000000e+00$	36	$1.031250000e+00$
68	$1.875000000e+00$	36	$1.000000000e+00$	38	$1.312500000e+00$
70	$4.156250000e+00$	38	$1.000000000e+00$	40	$7.500000000e-01$
72	$7.421875000e-01$	40	$1.000000000e+00$	42	$2.250000000e+00$
74	$8.875000000e+00$	42	$1.000000000e+00$	44	$1.968750000e+00$
76	$1.117187500e+00$	44	$1.000000000e+00$	46	$1.562500000e+00$
78	$8.468750000e+00$	46	$1.000000000e+00$	48	$3.578125000e+00$
80	$6.929687500e+00$	48	$1.000000000e+00$	50	$3.390625000e+00$
82	$3.507812500e+00$	50	$1.000000000e+00$	52	$2.812500000e+00$
84	$1.755468750e+01$	52	$1.000000000e+00$	54	$6.015625000e+00$
86	$3.242187500e+00$	54	$1.000000000e+00$	56	$6.171875000e+00$
88	$1.998046875e+01$			58	$4.656250000e+00$
90	$1.819531250e+01$			60	$1.056250000e+01$
92	$1.019140625e+01$				
94	$4.846142578e+01$				
96	$8.703125000e+00$				
98	$6.324267578e+01$				
100	$4.576562500e+01$				
102	$4.237841797e+01$				
104	$1.337663574e+02$				
106	$3.055810547e+01$				
108	$1.997133789e+02$				
110	$1.163720703e+02$				
112	$1.601523438e+02$				
114	$3.504694824e+02$				
116	$1.064584961e+02$				
118	$5.734952393e+02$				
120	$2.911203613e+02$				

All distance spectra contain spectral lines bounded with $3d_{free}$, except code No.4. For the sake of easier comparison of the codes we introduce the

Table 1. continued

No.7 $p=12$	$\mathcal{M}_c=3, R=1/4$ $G=[0,15,0,7]$	No.8 $p=12$	$\mathcal{M}_c=4, R=1/4$ $G=[0,2,0,25]$	No.9 $p=3$	$\mathcal{M}_c=1, R=1/3$ $G=[2,0,1]$
d_i	M_i	d_i	M_i	d_i	M_i
26	$5.000000000e-01$	32	$2.812500000e-01$	10	$2.500000000e-01$
30	$1.125000000e+00$	36	$9.218750000e-01$	12	$2.500000000e-01$
32	$2.500000000e-01$	40	$1.468750000e+00$	14	$7.500000000e-01$
34	$1.375000000e+00$	42	$2.656250000e-01$	16	$7.500000000e-01$
36	$5.000000000e-01$	44	$1.796875000e+00$	18	$1.000000000e+00$
38	$1.375000000e+00$	46	$1.093750000e+00$	20	$1.000000000e+00$
40	$8.750000000e-01$	48	$2.250000000e+00$	22	$1.000000000e+00$
42	$1.625000000e+00$	50	$2.421875000e+00$	24	$1.000000000e+00$
44	$1.312500000e+00$	52	$3.359375000e+00$	26	$1.000000000e+00$
46	$2.882812500e+00$	54	$4.359375000e+00$	28	$1.000000000e+00$
48	$2.328125000e+00$	56	$5.726562500e+00$	30	$1.000000000e+00$
50	$4.902343750e+00$	58	$7.250000000e+00$		
52	$4.726562500e+00$	60	$1.009375000e+01$		
54	$7.726562500e+00$	62	$1.212695312e+01$		
56	$7.882812500e+00$	64	$1.726562500e+01$		
58	$1.187500000e+01$	66	$2.121582031e+01$		
60	$1.179296875e+01$	68	$2.890234375e+01$		
62	$1.759375000e+01$	70	$3.719348145e+01$		
64	$1.842871094e+01$	72	$4.891894531e+01$		
66	$2.651367188e+01$	74	$6.437854004e+01$		
68	$3.001025391e+01$	76	$8.369458008e+01$		
70	$4.262792969e+01$	78	$1.104318848e+02$		
72	$4.951416016e+01$	80	$1.441513062e+02$		
74	$6.854296875e+01$	82	$1.882341614e+02$		
76	$8.122802734e+01$	84	$2.480677490e+02$		
78	$1.077687988e+02$	86	$3.226047974e+02$		
		88	$4.237944336e+02$		
		90	$5.536002197e+02$		
		92	$7.250024414e+02$		
		94	$9.445279541e+02$		
		96	$1.243921143e+03$		

cumulative distance distribution exponent $(CDDE)$ as

$$E_{cum}(d_l) = \frac{1}{N_c R}\mathrm{ld}\Big(\sum_{i\le l} M_i\Big), \tag{4}$$

where $N_c = (\mathcal{M}+1)N$ is the **constraint length** of the code [7].

This definition enables us to compare codes of different memory lengths

Table 1. continued

No.10 $p=5$	$\mathcal{M}_c=2, R=1/3$ $G=[0,2,7]$	No.11 $p=3$	$\mathcal{M}_c=3, R=1/3$ $G=[15,0,7]$	No.12 $p=3$	$\mathcal{M}_c=4, R=1/3$ $G=[25,0,37]$
d_i	M_i	d_i	M_i	d_i	M_i
12	$7.500000000e-01$	18	$1.000000000e+00$	20	$5.000000000e-01$
14	$2.500000000e-01$	22	$1.875000000e+00$	22	$2.500000000e-01$
16	$1.250000000e+00$	24	$1.125000000e+00$	24	$2.000000000e+00$
18	$7.500000000e-01$	26	$3.375000000e+00$	26	$8.750000000e-01$
20	$1.000000000e+00$	28	$2.500000000e+00$	28	$4.562500000e+00$
22	$1.281250000e+00$	30	$7.570312500e+00$	30	$3.257812500e+00$
24	$1.187500000e+00$	32	$7.359375000e+00$	32	$6.125000000e+00$
26	$2.250000000e+00$	34	$1.519921875e+01$	34	$1.064843750e+01$
28	$2.062500000e+00$	36	$1.857421875e+01$	36	$1.653710938e+01$
30	$3.687500000e+00$	38	$3.504687500e+01$	38	$2.096093750e+01$
32	$3.917968750e+00$	40	$4.434277344e+01$	40	$4.533007812e+01$
34	$5.355468750e+00$	42	$7.768359375e+01$	42	$5.813476562e+01$
36	$6.710937500e+00$	44	$1.078598633e+02$	44	$1.037343750e+02$
		46	$1.776022949e+02$	46	$1.601098633e+02$
		48	$2.526063232e+02$	48	$2.442608643e+02$
		50	$4.080000610e+02$	50	$3.908679199e+02$
		52	$5.969239502e+02$	52	$6.250312500e+02$
		54	$9.380477295e+02$	54	$9.368974609e+02$
				56	$1.544540283e+03$
				58	$2.369559082e+03$
				60	$3.715906250e+03$

and code rates on the same diagram. According to [12], $CDDE$ does not depend on these two code parameters, but on code excellence only. It may be shown [12] that $CDDE$ is essentially a logarithmic measure of the gain in error probability obtainable by an increase of Viterbi decoder complexity. For larger memories and distances ($\mathcal{M} > \mathcal{M}_L \approx 2$ and $d > d_L \approx 1.5 d_{free}$), $CDDE$ is approximated by [12],

$$E_{cum}(d_l) \cong a\frac{d_l}{N_c} + b, \qquad a \in \mathbb{R}^+,\ b \in \mathbb{R}. \tag{5}$$

This property is easily verified in Figs. (4-8). The better code is characterized by lower a and b, corresponding to the right-most plot in these figures. Since the slope, a, is approximately the same for all good codes, the vertical shift b could be taken as the sole measure of how much a code is matched to the $1-D$ channel. For good codes shift b should always be negative.

CDDE for several HY $R = 1/4$ codes is given in Figure 4. Increasing memory length, one obtains a more linear curve. The slope gradually increases with $\mathcal{M}_c$, with simultaneous slight increase of b.

In Figure 5 we compare the $CDDE$ for HY code No.11 and the channel code obtained using the maximal free Hamming distance convolutional code with same parameters and $G = [54, 64, 74]$, $d_{Hfree} = 10$ and $p = [1, 0, 1]$. Both codes are trellis matched. The $CDDE's$ for these two codes are both very linear with identical slopes. The $CDDE$ of the max. d_{Hfree} code is shifted upwards by approximately 0.6 so that the HY code is better not only when free $s.E.d.$'s are compared, but also throughout the whole distance spectrum.

In Figure 6 we compare the $CDDE$'s for different code rates and same memory lengths ($\mathcal{M}_c = 2$). Once again, the difference between $CDDE$'s compared is only in the vertical shift, the best code being the one with the smallest rate.

The influence of the coset representative on the $CDDE$ of the trellis matched codes obtained using the convolutional code with $G = [5, 7, 7]$, $\mathcal{M}_c = 2$, $d_{Hfree} = 8$ (this is the maximum d_{Hfree} code) and the code) and the convolutional code with $G = [0, 2, 7]$, $\mathcal{M}_c = 2$, $d_{Hfree} = 5$ are analysed in Figure 7. The best coset representative for both codes is $p = [1, 0, 1]$, and the worst one with $p = [0, 0, 0]$. Note that the second code with $p = [1, 0, 1]$ is actually the HY code used in Table 1 as the code No. 10. The influence of the coset representative is smaller for the maximum d_{Hfree} code, due to the fact the free $s.E.d.$ is bounded by d_{Hfree}, as mentioned above. It may also be observed that the free $s.E.d.$ of this code is not influenced by p, and that the only difference between channel codes obtained from it lies in the vertical shift b. The second convolutional code is much more sensitive to p. With the worst p its $CDDE$ is by far the least favorable one, including the worst free $s.E.d.$ of all four channel codes, while the best p gives the best $CDDE$ curve in the whole set.

Finally, $CDDE$'s for several HY codes with different code rates and memory lengths are compared in Figure 8.

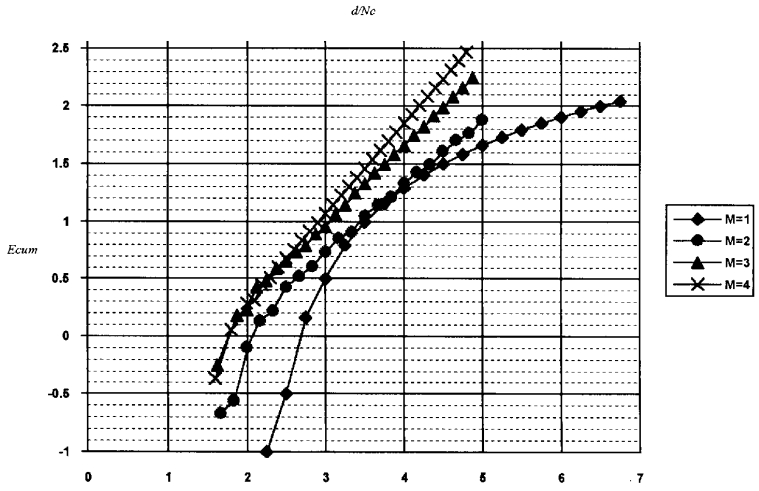

Fig. 4. Cumulative distance distribution exponent for HY codes with $R = 1/4$ and different memory lengths

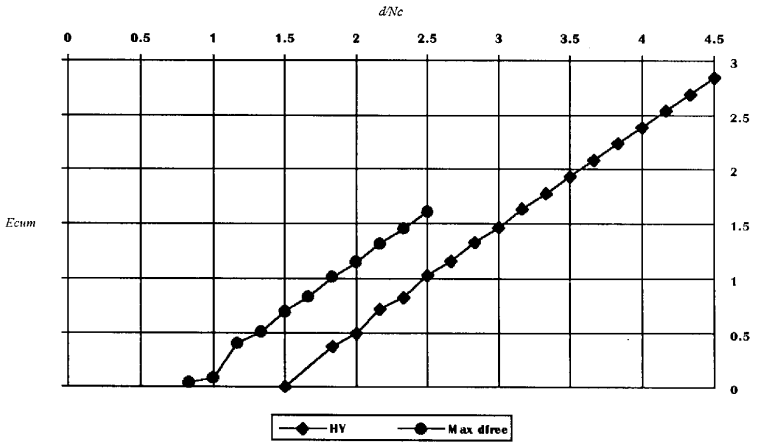

Fig. 5. Cumulative distance distribution exponent for HY code No. 11 and maximal free Hamming distance code with $R = 1/3$, $\mathcal{M} = 3$ and $p = [1, 0, 1]$

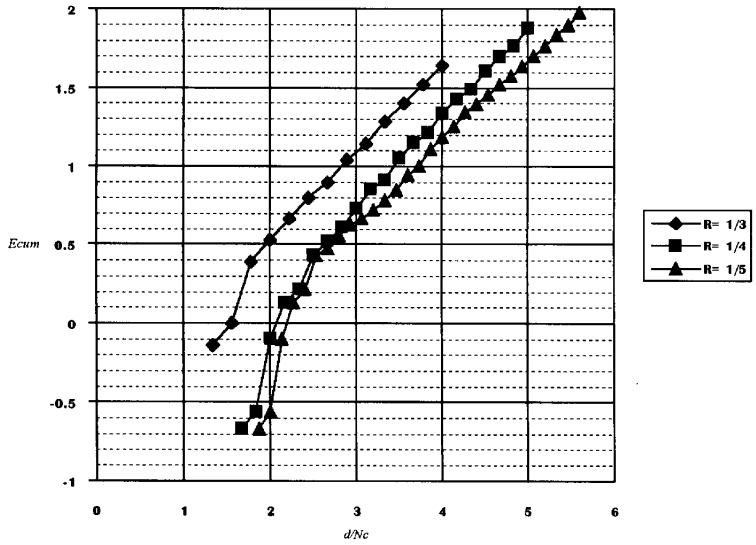

Fig.e 6. Cumulative distance distribution exponent for HY codes with $\mathcal{M} = 2$ and different code rates

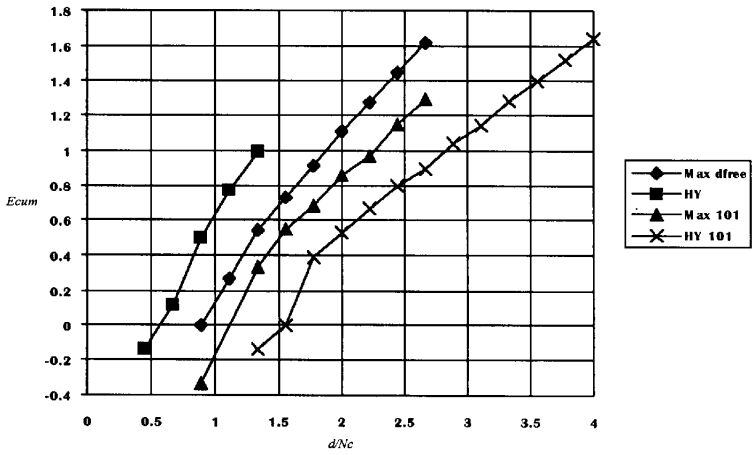

Fig. 7. Cumulative distance distribution exponent for HY codes and maximal d_{Hfree} codes and their cosets with $\mathcal{M} = 2$ and $R = 1/3$

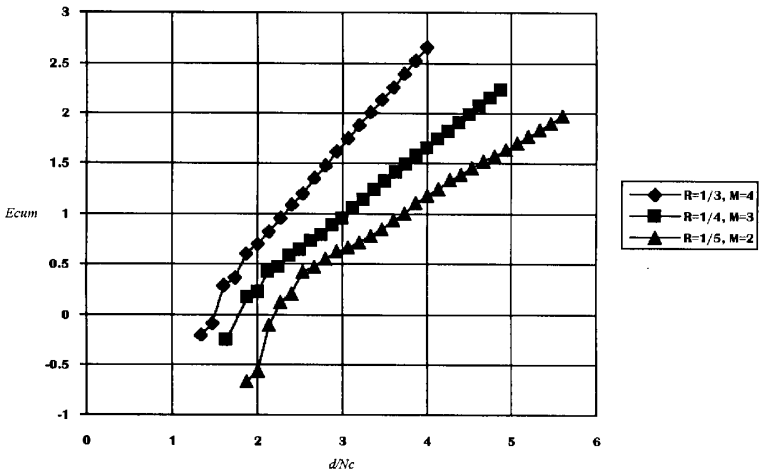

Fig. 8. Cumulative distance distribution exponent for HY codes with different code rates and memory lengths

REFERENCES

1. J. B. Anderson, T. Aulin, C. Sundberg: *Digital phase modulation.* Plenum Press, 1986.
2. M. Cedervall, R. Johannesson: *A FAST algorithm for computing distance spectrum of convolutional codes.* IEEE Trans.Comm., Vol.35. No.6., November 1989, pp. 1146–1159.
3. G. D. Forney,Jr.: *The Viterbi Algorithm.* Proc. IEEE, vo1.61, No.3, March 1973, pp. 268–78.
4. K. J. Hole, O. Ytrehus: *Improved coding techniques for precoded partial-response channels.* IEEE Trans. on Information Theory, Vol. 40, No.2, March 1994.
5. P. Kabal, S. Pasupathy: *Partial–Response Signaling.* IEEE Trans.Comm., vol. 23, No.9, Sept. 1975, pp. 921–934.
6. S. Lin, D. J. Costello: *Error control coding:Fundamentals and applications.* Prentice-Hall, 1983.
7. J. L. Massey: *Error bounds for tree codes, trellis codes and convolutional codes with encoding and decoding procedures.* CISM Courses and Lectures No.216, Coding and complexity, 1975.
8. M. Rouanne, D. J. Costello: *An algorithm for computing the distance spectrum of trellis codes.* IEEE Journal on Selected Areas in Comm., vol. SAC-7, Aug. 1989, pp. 929–940.

9. K. A. Schouhamer Immink: *Coding techniques for digital recorders.* Prentice-Hall, 1991.

10. P. H. Siegel, J. K. Wolf: *Modulation and coding for information storage.* IEEE Communication Magazine, Dec.1991, pp. 68–86.

11. V. ŠENK, D. E. Lazić, M. Despotović: *BIFAST - an algorithm for computing distance-spectrum of trellis codes.* Proc. of the 1991 MELECON conference, Ljubljana, Yugoslavia, 1991, pp. 602–605.

12. V. ŠENK: *The error exponent of specific families of error control codes and suboptimal decoding procedures that enable its attainment.* Ph.D. dissertation, Electrotehnical faculty, Belgrade, 1992.

13. A. J. Viterbi: *Convolutional codes and their performance in communication systems.* IEEE Trans.Commun.Technol., vol. COM-19, Oct. 1971, pp. 751–772.

14. J. K. Wolf, G. Ungerboeck: *Trellis coding for partial-response channels.* IEEE Trans.Comm., vol. COM-34, No. 8, Aug. 1986, pp. 765–773.