Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2010, Article ID 695237, 9 pages
doi:10.1155/2010/695237

Research Article

Enumerative Encoding of TMTR Codes for Optical Recording Channel

Hui-Feng Tsai

Department of Computer Science and Information Engineering, Ching Yun University, Jhongli City 32097, Taiwan

Correspondence should be addressed to Hui-Feng Tsai, hftsai@cyu.edu.tw

Received 26 July 2010; Accepted 19 September 2010

Academic Editor: Magnus Jansson

We propose a new time-varying maximum transition run (TMTR) code for DVD recording systems, which has a rate 8/11 higher than the EFMPlus code and a lower power spectral density (PSD) at low frequencies. An enumeration method for constructing the new TMTR code is presented. Computer simulations indicate that the proposed TMTR code outperforms the EFMPlus code in error performance when applied to partial response optical recording channels.

1. Introduction

In data storage systems, a modulation code is known as (d, k)-constrained code, where d and k represent the maximal and minimal number of zeros between two consecutive ones. The main function of a (d, k) modulation code is to improve the recording density and increase the storage capacity. The timing information could also be controlled using a (d, k) modulation code. For example, magnetic tape and disk systems often adopt $(1, 7)$ or $(2, 7)$ codes, while optical systems such as CD and DVD usually employ $(2, 10)$ EFM (Eight-to-Fourteen Modulation) or $(2, 10)$ EFMPlus [1] modulation codes.

Recent research on the (d, k) modulation code has focused on the time-varying maximum transition run (TMTR) code [2–9], which can be treated as a $(0, k)$ modulation code. The TMTR code matched to the partial response channel can delete some dominant error events and enhance the Euclidian distance of the partial response channel to the matched filter bound. As a result, a coding gain over the conventional scheme can be obtained when the time-varying Viterbi detector is applied to the TMTR-coded partial response channel. In a previous work [10], we proposed a new time-varying maximum transition run (TMTR) code with $(k_1^{\text{even}} = 1, k_1^{\text{odd}} = 2, k)$ constraint for DVD recording systems, which has rate 8/11 higher than the EFMPlus code and a lower power spectral density (PSD) at low frequencies. The $(k_1^{\text{even}} = 1, k_1^{\text{odd}} = 2, k)$ TMTR code was realized with a look-up table, and the k-constraint was not considered during construction. In this paper, instead of a look-up table we present an enumeration method for constructing the $(k_1^{\text{even}} = 1, k_1^{\text{odd}} = 2, k)$ codes. Based on this construction, a rate 8/11 code with $k = 7$ is found. The proposed code can achieve better timing recovery performance. We show that 387 surviving words exist with length 11 from the construction technique. This new method needs one bit of memory for encoding, but no memory is required for decoding. An enumerating algorithm is used for encoding/decoding, and a look-up table is not required.

The rest of this paper is organized as follows. In Section 2, we briefly describe the $(k_1^{\text{even}} = 1, k_1^{\text{odd}} = 2, k)$ TMTR codes for partial response (PR) optical recording channels. In Section 3, an outline of the design methodology for constructing a high-rate TMTR code is presented. We illustrate concatenation problem between codewords and provide a solution. Section 4 introduces an enumerative coding method for TMTR codes. In Section 5, the power spectral density (PSD) of the rate 8/11 code is evaluated and compared with the EFMPlus code. An error performance comparison between uncoded, TMTR-coded, and EFMPlus-coded EPRII optical recording channel is presented in Section 6. The conclusion is provided in Section 7.

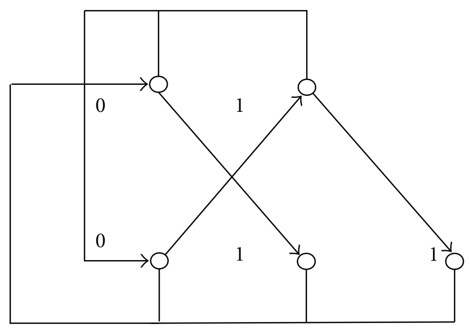

Figure 1: FSTD of TMTR constraint.

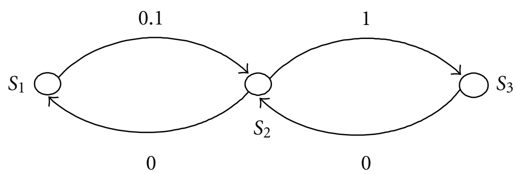

Figure 2: Simplified FSTD with TMTR constraint.

2. TMTR Codes for Partial Response Optical Recording Channels

The maximum transition run (MTR) method is a coding method, which limits the number of consecutive potential variations being not greater than k. The time-varying maximum transition run (TMTR) is a further modification of the MTR, which sets different constraints for the number of consecutive variations depending upon whether it starts at an odd or even position. For example, $(k_1^{\text{even}}, k_1^{\text{odd}})$ TMTR constraints mean that the number of consecutive 1s starting at an even position is not greater than k_1^{even} and the number of consecutive 1s starting at an odd position is not greater than k_1^{odd}. The method can increase the minimum distance of the encoded system to an upper matched filter bound (MFB); therefore, it has the distance enhancing property. The TMTR constraint can be described using a finite state transition diagram (FSTD), given in Figure 1. The vertices at the top of the diagram represent even positions, and the number of 1s starting at the even positions can be 1 only, satisfying the constraint of $k_1^{\text{even}} = 1$. The vertex at the bottom of the diagram represents odd positions, and the number of 1s starting at the odd positions can be 1 or 2, satisfying the constraint of $k_1^{\text{odd}} = 2$. Figure 2 shows a simplified FSTD with the $(k_1^{\text{even}} = 1, k_1^{\text{odd}} = 2)$ TMTR constraints.

Cideciyan et al. [11] suggested an advanced signal processing technique, the partial response and maximal likelihood (PRML) channel, to further increase the recording densities and reliability over that achieved by the conventional peak detector. The signal processing technique employing the PRML channel has become a standard widely used in most of today's data storage systems. The most popular partial response system for optical recording has the form $(1 + D)^n$, where n is a nonnegative integer. The PR systems with $n = 2$ and 3 are referred to as the PRII and EPRII systems, respectively. Karabed and Siegel [12] proposed a class of modulation codes that take advantage of the well-defined spectral nulls presented in partial response channels. The time-varying maximum transition run (TMTR) code [2–9], which can be treated as a $(0, k)$ modulation code, has recently been studied for partial response channels. The TMTR code matched to the partial response channel can delete some dominant error events and enhance the Euclidian distance of the partial response channel.

Vannucci and Foschini [13] described a powerful algorithm to search for the minimum Euclidean distance $d_{\min}^2$ for $(1 + D)^n$ partial response channels. They found that the shortest error event achieving $d_{\min}^2$ has the type of "$\ldots 0 + -0 \ldots$" for most of $(1 + D)^n$ partial response channels. As a matter of fact they found that those error events of the form "$\ldots 0 + -(+-)0 \ldots$" always have a distance less than the matched filter bound d_{MFB}^2 which is defined as the distance corresponding to the one-bit error event.

If the error event "$\ldots 0 + -0 \ldots$" can be forbidden to occur in coded sequences for $(1 + D)^n$ partial response channels, the minimum distance of the channels can be increased to d_{MFB}^2 resulting in a coding gain of $10 \log(d_{\text{MFB}}^2/d_{\min}^2)$ dB. With NRZI modulation there are four pairs of binary coded sequences, which could generate the error event "$\ldots 0 + -0 \ldots$," shown as follows:

$$\begin{matrix} 110 & \mathit{111} & 010 & \mathit{011} \\ \mathit{011} & 010 & \mathit{111} & 110 \end{matrix} \tag{1}$$

The TMTR modulation code with constraint $(0, k_1^{\text{even}} = 1, k_1^{\text{odd}} = 2)$ can be used to forbid the occurrence of sequences *111* and *011*, and as a result error event "$\ldots 0 + -0 \ldots$" would not occur in the detection of $(1 + D)^3$ channels, and a coding gain of 3 dB can be obtained. The channel capacity of the $(0, k_1^{\text{even}} = 1, k_1^{\text{odd}} = 2)$ TMTR code is equal to 0.7929, which indicates that a codeword with length 11 bits at least is required to encode or represent a byte (8-bit) message.

3. Construction for $(k_1^{\text{even}} = 1, k_1^{\text{odd}} = 2, k)$ TMTR Codes

A TMTR code is specified as $(k_1^{\text{even}} = 1, k_1^{\text{odd}} = 2, k)$ constraint, where k is the maximum number of consecutive zeros, k_1^{even} and k_1^{odd} constraints represent the maximum numbers of consecutive ones starting from an even position and an odd position, respectively. This construction is based upon $(k_1^{\text{even}} = 1, k_1^{\text{odd}} = 2, k; r_1, r_0, l_1, l_0)$ constraint. Both r_1 and r_0 represent the maximum number of ones after the last zero and the maximum number of zeros after the last one, respectively. In similar, parameters l_1 and l_0 represent the maximum number of ones before the first zero and the maximum number of zeros before the first one. Any two sequences satisfying $(k_1^{\text{even}} = 1, k_1^{\text{odd}} = 2, k; r_1, r_0, l_1, l_0) = (1, 2, k; 1, k - 1, 1, k)$ constraint can be freely concatenated without violating the $(k_1^{\text{even}} = 1, k_1^{\text{odd}} = 2, k)$ constraint. In order to reduce the consecutive zero length in the sequences after concatenation, the following substitution rule is applied: assume that a sequence $\mathbf{y}$ is followed by a sequence **x**, then

(1) if **x** has more than 2 zeros before the first one, and the last bit of **y** is a zero, then flip the first 2 bits of **x** into two ones, for example, $(000\ldots101) \rightarrow (110\ldots101)$;

(2) otherwise, use **x** as the encoder output.

In the case of (1), a long sequence of consecutive zeros is spread into two parts by "11" in the beginning of the 2nd sequence. Because $l_1 = 1$, a sequence beginning with "11" is not an original code.

Rates of some constructed codes using this method are listed in Table 1. As displayed, a rate 8/11 code with $k_1^{\text{even}} = 1$ and $k_1^{\text{odd}} = 2$ constraints is found, and this code has the $k = 7$ constraint. Sequences of even lengths satisfying the $(k_1^{\text{even}} = 1, k_1^{\text{odd}} = 2, k)$ constraint can be freely concatenated without violating the $(k_1^{\text{even}} = 1, k_1^{\text{odd}} = 2)$ constraint. Odd lengths sequences, however, cannot be freely concatenated without violating the $(k_1^{\text{even}} = 2, k_1^{\text{odd}} = 1)$ constraint. To solve this problem, assuming that there is a modulo-11 counter synchronized to the data, the two transitions in arrow can end at times 1, 2, 4, 6, 8, and 10 relative to counter. The even and odd positions in a codeword of 11 bits are given as $(e\,o\,e\,o\,e\,o\,e\,o\,e\,o\,e)$. For example, a sequence of 3 codewords will be

$$\begin{gathered}
\begin{pmatrix}10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1\end{pmatrix}, \quad \begin{pmatrix}10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1\end{pmatrix}, \quad \begin{pmatrix}10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1\end{pmatrix}\\
\begin{pmatrix}e & o & e & o & e & o & e & o & e & o & e\end{pmatrix}, \quad \begin{pmatrix}e & o & e & o & e & o & e & o & e & o & e\end{pmatrix}, \quad \begin{pmatrix}e & o & e & o & e & o & e & o & e & o & e\end{pmatrix}\\
\begin{pmatrix}1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 1\end{pmatrix}, \quad \begin{pmatrix}1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 1\end{pmatrix}, \quad \begin{pmatrix}1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 1\end{pmatrix}\\
\begin{pmatrix}x & e & o & e & o & e & o & e & o & e & o\end{pmatrix}, \quad \begin{pmatrix}x & e & o & e & o & e & o & e & o & e & o\end{pmatrix}, \quad \begin{pmatrix}x & e & o & e & o & e & o & e & o & e & o\end{pmatrix}
\end{gathered} \tag{2}$$

where the 1st line expresses the positions of the code bits. The 2nd line expresses the even/odd code bit positions. The 3rd line expresses the maximum number of consecutive "1" starting at the position. There is no two consecutive "2" in the 3rd line. It means that no dominant error event $\pm(1,-1)$ will occur. To obtain the coding gain of this encoder, a time-varying Viterbi detector is required. The trellis diagrams of the Viterbi detector for even and odd times are shown in Figure 7(c). The Viterbi detector for code bit stream positions will be the same as shifting the 2nd line to right by 1 position. The result is shown in the 4th line. The even or odd Viterbi detector properties must match the bit position shown in the 4th line.

4. Enumerative Encoding $(k_1^{\text{even}} = 1, k_1^{\text{odd}} = 2, k)$ TMTR Codes

Let us lexicographically order the binary sequences of length n by

$$\begin{gathered}
\underline{X} = (x_{n-1}, \ldots, x_1, x_0) \succ \underline{Y} = (y_{n-1}, \ldots, y_1, y_0)\\
\Longleftrightarrow \left(x_p > y_p\right) \text{ and} (x_i = y_i) \quad \forall p < i < n.
\end{gathered} \tag{3}$$

An enumerating encoder maps a set of consecutive integers onto a lexicographically ordered set of sequences. In order to describe the enumerating encoder/decoder, some notations will be defined as follows.

(D.1) A_n is the lexicographically ordered set of $(k_1^{\text{even}} = 1, k_1^{\text{odd}} = 2, k; r_1, r_0, l_1, l_0) = (1, 2, k; 1, k-1, 1, \infty)$ sequences of length n.

(D.2) $R(\underline{\mathbf{X}})$ is the number of sequences $\underline{Y} \in A_n$ such that $\underline{X} \succ \underline{Y}$.

(D.3) $R(\underline{\mathbf{0}}) = 0$, where $\underline{0}$ is the all zero sequence.

(D.4) $\text{res}(\underline{\mathbf{X}})$ is the sequence obtained by modifying the first nonzero bit of $\underline{X}$ to zero.

(D.5) $\underline{U}^i$ is the minimum sequence among sequences in A_n and having the first symbol one at position i.

(D.6) $\underline{M}^i$ is the maximum sequence among sequences in A_n and having the first symbol one at position i.

(D.7) $w_i = R(\underline{U}^i) - R(\text{res}(\underline{U}^i))$, then we have

$$R(\underline{\mathbf{X}}) = \sum_{i=0}^{n-1} x_i w_i. \tag{4}$$

(D.8) We have $t_i = R(\underline{M}^i)$.

By definitions (D.5) and (D.6), it is easy to see that

$$R\left(\underline{U}^i\right) = t_{i-1} + 1. \tag{5}$$

The w_i's and t_i's can be obtained by the following recursive relation with initial values $w_0 = 1, t_0 = 1$:

$$\begin{aligned}
w_i &= t_{i-1} + 1 - R\left(\text{res}\left(\underline{\mathbf{U}}^i\right)\right),\\
t_i &= R\left(\underline{\mathbf{M}}^i\right).
\end{aligned} \tag{6}$$

For illustration, consider $(k_1^{\text{even}} = 1, k_1^{\text{odd}} = 2, k; r_1, r_0, l_1, l_0) = (1, 2, 4; 1, 3, 1, \infty)$ sequences with length 4; one has

$$\begin{array}{ccl}
i & & 3\ \ 2\ \ 1\ \ 0,\\
w_i & \longrightarrow & 6\ \ 3\ \ 2\ \ 1,\\
t_i & \longrightarrow & 8\ \ 5\ \ 2\ \ 1,\\
R\left(\text{res}\left(\underline{\mathbf{U}}^i\right)\right) & \longrightarrow & 0\ \ 0\ \ 0\ \ 0,
\end{array} \tag{7}$$

Table 1: Practical code rates of $(k_1^{\text{even}} = 1, k_1^{\text{odd}} = 2, k)$ codes.

Block length	Code rate	$(k_1^{\text{even}} = 1, k_1^{\text{odd}} = 2, k)$
4	4/5	(1, 2, 4)
5	3/5	(1, 2, 4)
6	4/6	(1, 2, 5)
7	5/7	(1, 2, 6)
8	6/8	(1, 2, 6)
9	7/9	(1, 2, 7)
10	7/10	(1, 2, 6)
11	8/11	(1, 2, 7)
12	9/12	(1, 2, 7)
20	14/20	(1, 2, 6)
22	16/22	(1, 2, 7)
24	18/24	(1, 2, 7)

where

$$\begin{aligned} \underline{M}^0 &= (0001), \quad \underline{U}^0 = (0001), \\ \underline{M}^1 &= (0010), \quad \underline{U}^1 = (0010), \\ \underline{M}^2 &= (0110), \quad \underline{U}^2 = (0100), \\ \underline{M}^3 &= (1010), \quad \underline{U}^3 = (1000). \end{aligned} \tag{8}$$

Similarly, when using a codeword of length 11 and code rate of 8/11, the enumerating encoding method is as follows:

$$\begin{array}{rcccccccccccc} i & & 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0, \\ w_i & \longrightarrow & 234 & 157 & 79 & 53 & 27 & 18 & 9 & 6 & 3 & 2 & 1, \\ t_i & \longrightarrow & 470 & 236 & 158 & 79 & 53 & 26 & 17 & 8 & 5 & 2 & 1, \\ R\left(\text{res}\left(\mathbf{U}^i\right)\right) & \longrightarrow & 6 & 3 & 2 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0, \end{array} \tag{9}$$

where

$$\begin{aligned} \underline{M}^0 &= (00000001010), \quad \underline{U}^0 = (00000001000), \\ \underline{M}^1 &= (00000011010), \quad \underline{U}^1 = (00000010000), \\ \underline{M}^2 &= (00000101010), \quad \underline{U}^2 = (00000100000), \\ \underline{M}^3 &= (00001101010), \quad \underline{U}^3 = (00001000000), \\ \underline{M}^4 &= (00010101010), \quad \underline{U}^4 = (00010000001), \\ \underline{M}^5 &= (00110101010), \quad \underline{U}^5 = (00100000001), \\ \underline{M}^6 &= (01010101010), \quad \underline{U}^6 = (01000000010), \\ \underline{M}^7 &= (10110101010), \quad \underline{U}^7 = (10000000100). \end{aligned} \tag{10}$$

Let B_n be a subset of A_n consisting of sequences with no more than $k = 7$ leading zeros. Then the number of elements of B_{11} is given as

$$\begin{aligned} |B_{11}| &= R([10110101010]) - R([00000000110]) \\ &= w_{10} + w_8 + w_7 + w_5 + w_3 + w_1 - w_2 - w_1 \\ &= 387. \end{aligned} \tag{11}$$

```
β = β + t^(n−k−1) + 1          if    (x'_0 = 0)
i = n − 1                         if    (x_(n−1) = 0)
                                     if   (x_(n−2) = 0)
while i > 0                              x_(n−1) = 1
   if  β > t^(i−1)                       x_(n−2) = 1
       x_i = 1
       β = β − w_i                    end
   end                             end
end                              end
x_0 = β
```

Figure 3: Enumerative encoding for $(k_1^{\text{even}}, k_1^{\text{odd}}, k)$ codes.

Table 2: Codebook of rate 3/4 $(k_1^{\text{even}} = 1, k_1^{\text{odd}} = 2, k = 4)$ code.

Data	Codeword	Data	Codeword
$b_1 \cdots b_3$	$c_1 \cdots c_4$	$b_1 \cdots b_3$	$c_1 \cdots c_4$
000	0001or 1101	100	0110
001	0010	101	1000
010	0100	110	1001
011	0101	111	1010

We can then encode an integer β to a sequence $\underline{\mathbf{X}}$ using the enumerative algorithm given in Figure 3 (assuming the previous block is $\underline{\mathbf{X}'}$). The decoding is simply done by

$$\hat{\beta} = \sum_{i=0}^{n-1} x_i w_i - t^{n-k-1} - 1, \tag{12}$$

where $t^{n-k-1} + 1$ is the value of first codeword, $t_i = 0$ and $t_i = -1$ for $i < 0$.

For illustration, the mapping relationship between data and codewords of a rate 3/4 code satisfying $(k_1^{\text{even}} = 1, k_1^{\text{odd}} = 2, k = 4)$ constraint is given in Table 2. The data to codeword mapping for a rate 8/11 code satisfying $(k_1^{\text{even}} = 1, k_1^{\text{odd}} = 2, k = 7)$ constraint is listed in Table 3.

5. Power Spectral Density

In DVD systems the power spectral density at low frequency, referred to as the low-frequency content, of the encoded data sequences should normally be maintained as low as possible to alleviate interference with pilot and focus servo signals. For example, in addition to satisfying the (2, 10) constraint, the 8/16 EFMPlus code employed in the DVD system is also designed to achieve very low low-frequency content to reduce the interference between the written signal and the servo signal. Efficient low-frequency component suppression is a crucial criterion for the 8/11 $(0, k_1^{\text{even}} = 1, k_1^{\text{odd}} = 2, k = 7)$ TMTR code rate. The low-frequency content is based upon the running digital sum (RDS) Z given by

$$Z_i = \sum_{j=-\infty}^{i} X_j = Z_{i-1} + X_i, \tag{13}$$

Table 3: Data to codeword mapping for the rate 8/11 ($k_1^{\text{even}} = 1, k_1^{\text{odd}} = 2, k = 7$) code.

	Data	Codeword		Data	Codeword		Data	Codeword		Data	Codeword
	$b_1 \cdots b_8$	$c_1c_2 \cdots c_{11}$		$b_1 \cdots b_8$	$c_1c_2 \cdots c_{11}$		$b_1 \cdots b_8$	$c_1c_2 \cdots c_{11}$		$b_1 \cdots b_8$	$c_1c_2 \cdots c_{11}$
0	00000000	00000001000	42	00101010	00001100100	94	01011110	00100100100	178	10110010	01001000000
		or 11000001000			or 11001100100	95	01011111	00100100101	179	10110011	01001000001
1	00000001	00000001001	43	00101011	00001100101	96	01100000	00100100110	180	10110100	01001000010
		or 11000001001			or 11001100101	97	01100001	00100101000	181	10110101	01001000100
2	00000010	000000010	44	00101100	00001100110	98	01100010	00100101001	182	10110110	01001000101
		or 11000001010			or 11001100110	99	01100011	00100101010	183	10110111	01001000110
3	00000011	00000010000	45	00101101	00001101000	100	01100100	00101000000	184	10111000	01001001000
		or 11000010000			or 11001101000	101	01100101	00101000001	185	10111001	01001001001
4	00000100	00000010001	46	00101110	00001101001	102	01100110	00101000010	186	10111010	01001001010
		or 11000010001			or 11001101001	103	01100111	00101000100	187	10111011	01001010000
5	00000101	00000010010	47	00101111	00001101010	104	01101000	00101000101	188	10111100	01001010001
		or 11000010010			or 11001101010	105	01101001	00101000110	189	10111101	01001010010
6	00000110	00000010100	48	00110000	00010000001	106	01101010	00101001000	190	10111110	01001010100
		or 11000010100			or 11010000001	107	01101011	00101001001	191	10111111	01001010101
7	00000111	00000010101	49	00110001	00010000010	108	01101100	00101001010	192	11000000	01001010110
		or 11000010101			or 11010000010	109	01101101	00101010000	193	11000001	01001011000
8	00001000	00000010110	50	00110010	00010000100	110	01101110	00101010001	194	11000010	01001011001
		or 11000010110			or 11010000100	111	01101111	00101010010	195	11000011	01001011010
9	00001001	00000011000	51	00110011	00010000101	112	01110000	00101010100	196	11000100	01001100000
		or 11000011000			or 11010000101	113	01110001	00101010101	197	11000101	01001100001
10	00001010	00000011001	52	00110100	00010000110	114	01110010	00101010110	198	11000110	01001100010
		or 11000011001			or 11010000110	115	01110011	00101011000	199	11000111	01001100100
11	00001011	00000011010	53	00110101	00010001000	116	01110100	00101011001	200	11001000	01001100101
		or 11000011010			or 11010001000	117	01110101	00101011010	201	11001001	01001100110
12	00001100	00000100000	54	00110110	00010001001	118	01110110	00101100000	202	11001010	01001101000
		or 11000100000			or 11010001001	119	01110111	00101100001	203	11001011	01001101001
13	00001101	00000100001	55	00110111	00010001010	120	01111000	00101100010	204	11001100	01001101010
		or 11000100001			or 11010001010	121	01111001	00101100100	205	11001101	01010000001
14	00001110	00000100010	56	00111000	00010010000	122	01111010	00101100101	206	11001110	01010000010
		or 11000100010			or 11010010000	123	01111011	00101100110	207	11001111	01010000100
15	00001111	00000100100	57	00111001	00010010001	124	01111100	00101101000	208	11010000	01010000101
		or 11000100100			or 11010010001	125	01111101	00101101001	209	11010001	01010000110
16	00010000	00000100101	58	00111010	00010010010	126	01111110	00101101010	210	11010010	01010001000
		or 11000100101			or 11010010010	127	01111111	00110000001	211	11010011	01010001001
17	00010001	00000100110	59	00111011	00010010100	128	10000000	00110000010	212	11010100	01010001010
		or 11000100110			or 11010010100	129	10000001	00110000100	213	11010101	01010010000
18	00010010	00000101000	60	00111100	00010010101	130	10000010	00110000101	214	11010110	01010010001
		or 11000101000			or 11010010101	131	10000011	00110000110	215	11010111	01010010010
19	00010011	00000101001	61	00111101	00010010110	132	10000100	00110001000	216	11011000	01010010100
		or 11000101001			or 11010010110	133	10000101	00110001001	217	11011001	01010010101
20	00010100	00000101010	62	00111110	00010011000	134	10000110	00110001010	218	11011010	01010010110
		or 11000101010			or 11010011000	135	10000111	00110010000	219	11011011	01010011000
21	00010101	00001000000	63	00111111	00010011001	136	10001000	00110010001	220	11011100	01010011001
		or 11001000000			or 11010011001	137	10001001	00110010010	221	11011101	01010011010
22	00010110	00001000001	64	01000000	00010011010	138	10001010	00110010100	222	11011110	01010100000
		or 1100100000			or 11010011010	139	10001011	00110010101	223	11011111	01010100001
23	00010111	00001000010	65	01000001	00010100000	140	10001100	00110010110	224	11100000	01010100010
		or 11001000010			or 11010100000	141	10001101	00110011000	225	11100001	01010100100
24	00011000	00001000100	66	01000010	00010100001	142	10001110	00110011001	226	11100010	01010100101
		or 11001000100			or 11010100001	143	10001111	00110011010	227	11100011	01010100110
25	00011001	00001000101	67	01000011	00010100010	144	10010000	00110100000	228	11100100	01010101000
		or 11001000101			or 11010100010	145	10010001	00110100001	229	11100101	01010101001

Table 3: Continued.

	Data $b_1 \cdots b_8$	Codeword $c_1c_2 \cdots c_{11}$		Data $b_1 \cdots b_8$	Codeword $c_1c_2 \cdots c_{11}$		Data $b_1 \cdots b_8$	Codeword $c_1c_2 \cdots c_{11}$		Data $b_1 \cdots b_8$	Codeword $c_1c_2 \cdots c_{11}$
26	00011010	00001000110	68	01000100	00010100100	146	10010010	00110100010	230	11100110	01010101010
		or 11001000110			or 11010100100	147	10010011	00110100100	231	11100111	10000000100
27	00011011	00001001000	69	01000101	00010100101	148	10010100	00110100101	232	11101000	10000000101
		or 11001001000			or 11010100101	149	10010101	00110100110	233	11101001	10000000110
28	00011100	00001001001	70	01000110	00010100110	150	10010110	00110101000	234	11101010	10000001000
		or 11001001001			or 11010100110	151	10010111	00110101001	235	11101011	10000001001
29	00011101	00001001010	71	01000111	00010101000	152	10011000	00110101010	236	11101100	10000001010
		or 11001001010			or 11010101000	153	10011001	01000000010	237	11101101	10000010000
30	00011110	00001010000	72	01001000	00010101001	154	10011010	01000000100	238	11101110	10000010001
		or 11001010000			or 11010101001	155	10011011	01000000101	239	11101111	10000010010
31	00011111	00001010001	73	01001001	00010101010	156	10011100	01000000110	240	11110000	10000010100
		or 11001010001			or 11010101010	157	10011101	01000001000	241	11110001	10000010101
32	00100000	00001010010	74	01001010	00100000001	158	10011110	01000001001	242	11110010	10000010110
		or 11001010010	75	01001011	00100000010	159	10011111	01000001010	243	11110011	10000011000
33	00100001	00001010100	76	01001100	00100000100	160	10100000	01000010000	244	11110100	10000011001
		or 11001010100	77	01001101	00100000101	161	10100001	01000010001	245	11110101	10000011010
34	00100010	00001010101	78	01001110	00100000110	162	10100010	01000010010	246	11110110	10000100000
		or 11001010101	79	01001111	00100001000	163	10100011	01000010100	247	11110111	10000100001
35	00100011	00001010110	80	01010000	00100001001	164	10100100	01000010101	248	11111000	10000100010
		or 11001010110	81	01010001	00100001010	165	10100101	01000010110	249	11111001	10000100100
36	00100100	00001011000	82	01010010	00100010000	166	10100110	01000011000	250	11111010	10000100101
		or 11001011000	83	01010011	00100010001	167	10100111	01000011001	251	11111011	10000100110
37	00100101	00001011001	84	01010100	00100010010	168	10101000	01000011010	252	11111100	10000101000
		or 11001011001	85	01010101	00100010100	169	10101001	01000100000	253	11111101	10000101001
38	00100110	00001011010	86	01010110	00100010101	170	10101010	01000100001	254	11111110	10000101010
		or 11001011010	87	01010111	00100010110	171	10101011	01000100010	255	11111111	10001000000
39	00100111	00001100000	88	01011000	00100011000	172	10101100	01000100100			
		or 11001100000	89	01011001	00100011001	173	10101101	01000100101			
40	00101000	00001100001	90	01011010	00100011010	174	10101110	01000100110			
		or 11001100001	91	01011011	00100100000	175	10101111	01000101000			
41	00101001	00001100010	92	01011100	00100100001	176	10110000	01000101001			
		or 11001100010	93	01011101	00100100010	177	10110001	01000101010			

where $\{X_i\} = \{\ldots, X_{-1}, X_0, X_1, \ldots\} \in \{-1, +1\}$ represents the writing sequences. A lower RDS results in lower low-frequency content. The power spectral density of a sequence can be expressed as

$$H(w) = \lim_{M\to\infty} E\left[\frac{1}{2M+1}\left|\sum_{m=-M}^{M} X_m e^{-jmw}\right|^2\right]. \tag{14}$$

For a bounded RDS, that is, $|Z_i| = |\sum_{j=-\infty}^{i} X_j| \leq C < \infty$, C is a constant, and the DC content is then given as

$$H(0) = \lim_{M\to\infty} E\left[\frac{1}{2M+1}\left|\sum_{m=-M}^{M} X_m\right|^2\right] \approx 0, \tag{15}$$

that indicates a DC-free content. As shown in (11), the number of sequences with length 11 satisfying $(k_1^{\text{even}} = 1, k_1^{\text{odd}} = 2, k, r_1, r_0, l_1, l_0) = (1, 2, 7; 1, 6, 1, \infty)$ is 387, but it only requires 256 codewords in the rate 8/11 TMTR code. The surplus $387 - 256 = 131$ code sequences can be exploited for minimizing both k-constraint and low-frequency content. The 131 surplus code sequences are then used to suppress the low-frequency content by reducing the running digital sum (RDS). There are two tables (a main table and a substitute table) used in the encoder, as shown in Figure 4. The main table consists of 256 codewords (i.e., sequences $\underline{\mathbf{X}}$ corresponding to $132 \leq \beta \leq 387$ in the enumerating algorithm), and the substitute table with 131 codewords (i.e., sequences $\underline{\mathbf{X}}$ corresponding to $1 \leq \beta \leq 131$) is used to minimize the power spectral density at low frequencies.

β	Sequence	$\underline{X}$	β	Sequence	$\underline{X}$
132	0011000011	0	1	0000000100	0
	⋮			⋮ Substitute table ⋮	
	⋮				
262	1000100100	0	131	0011000010	1
	⋮ Main table ⋮				
387	1011010101	0			

Figure 4: Codebook of rate 8/11 ($k_1^{\text{even}} = 1, k_1^{\text{odd}} = 2, 7$) TMTR code.

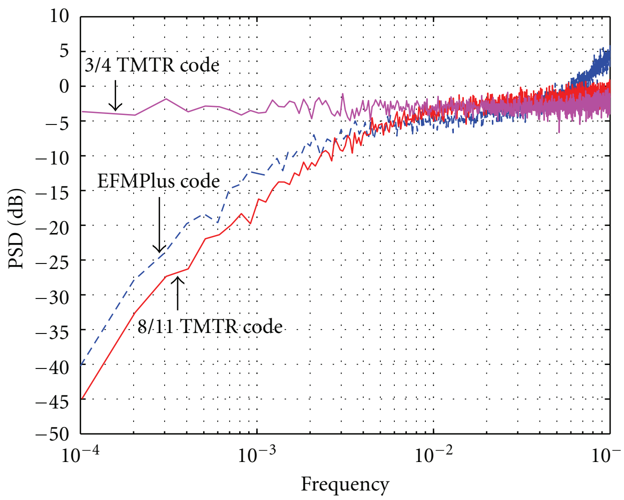

Figure 5: Power spectral density.

The PSD of both EFMPlus code and $(0, k_1^{\text{even}} = 1, k_1^{\text{odd}} = 2)$ TMTR code can be computed using the fast Fourier transform (FFT)

$$H(kw_s) \cong \sum_{n=1}^{\infty} X_n W_N^{(k-1)(n-1)}, \tag{16}$$

where $W_N^k = e^{-j(2k\pi/N)}$, $w_s = 2\pi/N$.

Figure 5 depicts the power spectral density (PSD) versus the normalized frequency for the 8/16 EFMPlus code, the rate 3/4 $(0, k_1^{\text{even}} = 1, k_1^{\text{odd}} = 2, k = 4)$ TMTR code, and the 8/11 $(0, k_1^{\text{even}} = 1, k_1^{\text{odd}} = 2, k = 7)$ TMTR code with method of enumerative, respectively. As shown, at frequency of 10^{-4}, the PSDs for both EFMPlus and TMTR codes are -40 dB, -4 dB and -45 dB, respectively. The result indicates that the 8/11 $(0, k_1^{\text{even}} = 1, k_1^{\text{odd}} = 2, k = 7)$ TMTR code achieves a lower low-frequency content (with 5 dB lower) than the EFMPlus code. In Figure 5, we notice that the rate 3/4 $(0, k_1^{\text{even}} = 1, k_1^{\text{odd}} = 2, k = 4)$ TMTR code has only -4 dB at frequency of 10^{-4} and it is not suitable for optical recording systems, although it has a higher code rate compared to the 8/11 $(0, k_1^{\text{even}} = 1, k_1^{\text{odd}} = 2, k = 7)$ TMTR code.

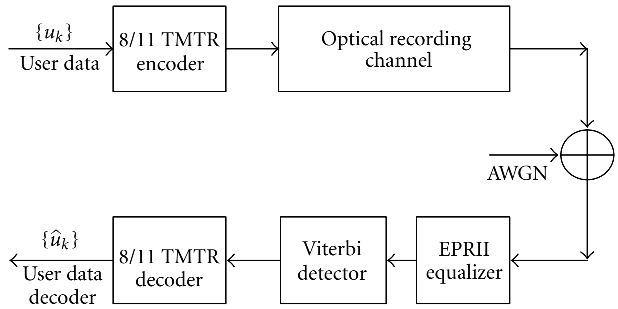

Figure 6: Optical recording system model.

6. Simulation Results

The superiority of the rate 8/11 $(0, k_1^{\text{even}} = 1, k_1^{\text{odd}} = 2)$ TMTR code over the EFMPlus code is also demonstrated on error performance through a computer simulation on the EPRII optical channel of the form $P(D) = (1 + D)^3$. Figure 6 depicts the optical recording system for simulation. An 8-state transition diagram, as depicted in Figure 7(a), can describe the EPRII optical channel. During simulation, a Gaussian optical recording channel was assumed with impulse response $m(t)$ given by

$$m(t) = \frac{2}{\sqrt{\pi}ST} \exp\left\{-\left(\frac{2t}{ST}\right)^2\right\}, \tag{17}$$

where S represents the user information density. T is the channel bit period assumed to be one $(T = 1)$ for uncoded systems in simulation. The optical recording channel is then corrupted with the additive white Gaussian noise (AWGN), and at receiver an ideal EPRII equalizer is employed to equalize the optical channel in the form of $P(D) = (1 + D)^3$.

The Viterbi detector is then employed on the 8-state trellis diagram of the EPRII optical channel, to recover binary-recorded data from the equalized and sampled output. Note that for the EFMPlus-coded EPRII channel the 8-state trellis diagram can be reduced to a 6-state trellis diagram due to the $(2, 10)$ constraint, as shown in Figure 7(b), while for the TMTR-coded EPRII channel the 8-state trellis diagram is still required with some branches deleted as depicted in Figure 7(c). Figure 8 demonstrates the bit error rate versus the signal to noise ratio (SNR) for TMTR-coded, EFMPlus-coded and uncoded EPRII optical recording channels at user density $S = 1.2$. The SNR in dB is defined in this paper as the ratio of the input complex waveform signal energy generated by a 127-bit pseudorandom binary sequence to the channel noise energy of the same duration. As shown both the TMTR-coded and EFMPlus-coded EPRII systems achieve better performance than the uncoded EPRII system.

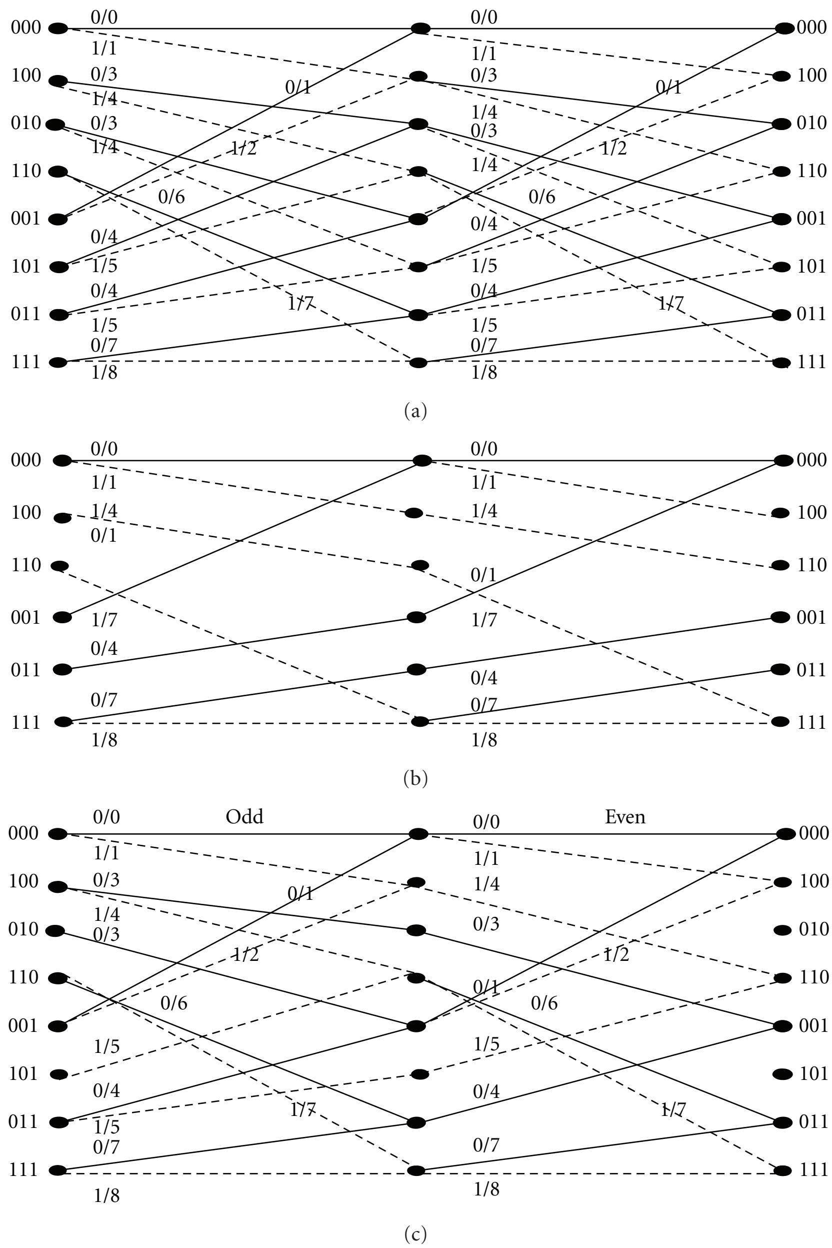

Figure 7: Trellis diagram. (a) EPRII channel. (b) EFMPlus-coded EPRII channel. (c) TMTR-coded EPRII channel.

This is because both coded EPRII systems have a coding gain of 3 dB when these modulation codes are considered in the 8-state trellis diagram of the EPRII system during detection. This leads to a coding gain of 3 dB. As also can be seen in this figure, the TMTR-coded EPRII system improves the EFMPlus-coded EPRII system by approximately 1 dB in bit error rate. The coding loss of the EFMPlus-coded EPRII system is due to the bit rate loss. Figure 9 shows the signal to noise (SNR) required to achieve a bit rate error of 10^{-5}, as a function of user density, at a rate of 8/11 $(0, k_1^{\text{even}} = 1, k_1^{\text{odd}} = 2)$ TMTR code and a rate of 8/16 EFMPlus code, applied to an EPRII optical channel. Figure 9 shows how the TMTR code provides little coding gain at user densities below 1.2 but increases coding gain at higher densities. At a user density of $S = 2.0$, the TMTR code on the EPRII optical channel provides nearly 2.7 dB of coding gain above the rate of 8/16 EFMPlus code on the EPRII optical channel. Therefore, from the performance comparison made in Figure 9, it can be seen that even a greater improvement in coding gain could be achieved for TMTR-coded EPRII system at higher user densities.

7. Conclusion

In this paper, we present an enumeration method for constructing $(k_1^{\text{even}} = 1, k_1^{\text{odd}} = 2, k)$ codes, and the look-up table described in a previous work is not required for the encoder/decoder. Based on the construction, a rate

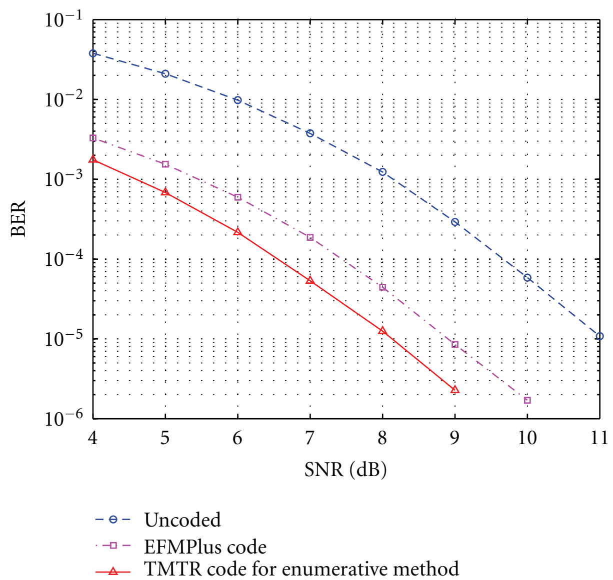

FIGURE 8: Error performance of TMTR-/EFMPlus-coded and uncoded EPRII systems.

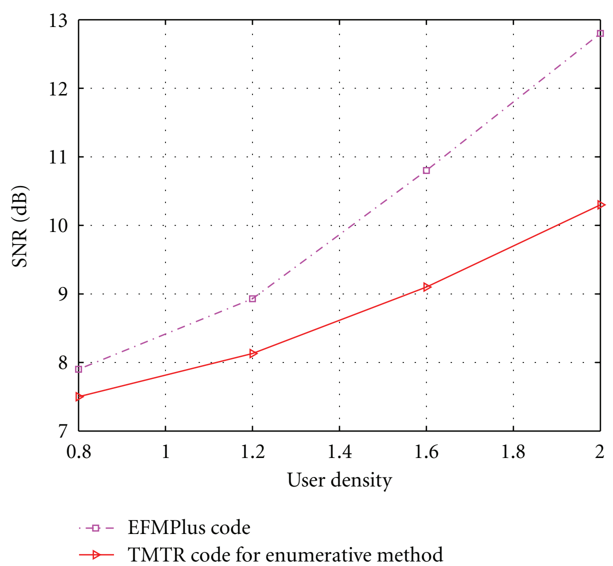

FIGURE 9: The SNR in dB required to achieve a 10^{-5} BER versus user density.

8/11 $(k_1^{\text{even}} = 1, k_1^{\text{odd}} = 2, 7)$ TMTR code is found. This code can achieve lower power spectral density at low frequencies compared to the EFMPlus code. In addition, computer simulations reveal that the rate 8/11 TMTR code outperforms the EFMPlus code in error performance when applied to partial response optical recording channels.

References

[1] K. A. S. Immink, "EFMPlus: the coding format of the multimedia compact disc," *IEEE Transactions on Consumer Electronics*, vol. 41, no. 3, pp. 491–497, 1995.

[2] J. Moon and B. Brickner, "Maximum transition run codes for data storage systems," *IEEE Transactions on Magnetics*, vol. 32, no. 5, pp. 3992–3994, 1996.

[3] B. E. Moision, P. H. Siegel, and E. Soljanin, "Distance-enhancing codes for digital recording," *IEEE Transactions on Magnetics*, vol. 34, no. 1, pp. 69–74, 1998.

[4] W. G. Bliss, "An 8/9 rate time-varying trellis code for high density magnetic recording," *IEEE Transactions on Magnetics*, vol. 33, no. 5, pp. 2746–2748, 1997.

[5] K. K. Fitzpatrick and C. S. Modlin, "Time-varying MTR codes for high density magnetic recording," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '97)*, pp. 1250–1253, Phoenix, Ariz, USA, November 1997.

[6] B. Nikolić, M. M.-T. Leung, and L. K.-C. Fu, "Rate 8/9 sliding block distance-enhancing code with stationary detector," *IEEE Transactions on Magnetics*, vol. 37, no. 3, pp. 1168–1174, 2001.

[7] R. D. Cideciyan and E. Eleftheriou, "Codes satisfying maximum transition run and parity-check constraints," in *Proceedings of the IEEE International Conference on Communications*, pp. 635–639, June 2004.

[8] I. Demirkan and Y. X. Lee, "The combined constraints for perpendicular recording channels," *IEEE Transactions on Magnetics*, vol. 42, no. 2, pp. 220–225, 2006.

[9] T. L. Poo and B. H. Marcus, "Time-varying maximum transition run constraints," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4464–4480, 2006.

[10] H.-F. Tsai and Y. Lin, "Turbo decoding for a new DVD recording system," *IEEE Transactions on Consumer Electronics*, vol. 51, no. 3, pp. 864–871, 2005.

[11] R. D. Cideciyan, F. Dolivo, R. Hermann, W. Hirt, and W. Schott, "A PRML system for digital magnetic recording," *IEEE Journal on Selected Areas in Communications*, vol. 10, no. 1, pp. 38–56, 1992.

[12] R. Karabed and P. H. Siegel, "Matched spectral-null codes for partial-response channels," *IEEE Transactions on Information Theory*, vol. 37, no. 3, pp. 818–855, 1991.

[13] G. Vannucci and G. J. Foschini, "The minimum distance for digital magnetic recording partial responses," *IEEE Transactions on Information Theory*, vol. 37, no. 3, pp. 955–960, 1991.