SIMULATION STUDY OF THE PERFORMANCE OF THE VITERBI DECODING ALGORITHM FOR CERTAIN M-LEVEL LINE CODES

Khmaies Ouahada [*]

[*] *Department of Electrical and Electronic Engineering Science, University of Johannesburg, South Africa. Email: kouahada@uj.ac.za.*

Abstract: In this paper we study the performance of different classes of M-level line codes under the Viterbi decoding algorithm. Some of the presented M-level line codes inherited the state machine structure by using the technique of distance mappings which preserve the properties of binary convolutional codes. Other M-level line codes were enforced to have the state machine structure to make use of the Viterbi decoding algorithm. The technique of spectral shaping was combined with distance mappings to generate spectral null distance mappings (SNDM) M-level line codes.
The 2-dB gain between soft and hard decisions decoding for the different classes of M-level line codes is investigated. The standard technique for assessing the stability and the accuracy of any decoding algorithm, which is the error propagation is used to analyze the stability and the accuracy of the Viterbi decoding algorithm of the M-level line codes.
The obtained results have shown advantages and outperformance of SNDM codes compared to the rest of line codes presented in this paper.

Key words: Viterbi decoder, Soft/Hard decision, Error propagation, Line codes

1. INTRODUCTION

In literature, many authors' works contributed towards the development of the design of multi-level line codes and the improvements of their error-correction capabilities [1–4]. For certain applications, researchers have shown that M-level line codes may be preferable to binary codes for high speed digital transmission as in the case of the optical fiber channel [5–8]. The additional signal levels or symbols in a pulse amplitude modulated signal sequence can be used to reduce the symbol rate and hence the bandwidth of the coded signal [9]. The lower switching rate required can also be used to obtain higher data-transfer rates in an optical local area network (LAN) system where the transmission rate is limited by complementary metal oxide semiconductor (CMOS) technology [10].

It is still common practice to use combinational logic decoders for M-level line codes. As these codes are considered to be non-linear codes, we make use of the technique of distance mappings [12–14] to map permutation sequences to the outputs of convolutional codes that can give our new M-level line codes the trellis structure and thus make use of the Viterbi decoding algorithm.

The spectral shaping technique used in this paper is to create nulls at certain specific frequencies including the lowest ones, which can give our new designed M-level line codes another advantage to overcome some communications problems like channels not transmitting zero frequency components.

This paper is organized as follows. Section 2 introduces briefly the techniques of spectral shaping and distance mappings and presents a few examples of algorithms for the design of the related class of M-level line codes. Section 3 investigates the implementation of the Viterbi decoding to our designed codes and to a range of different published M-level line codes. Simulation results for the bit error rate (BER) performance of these M-level line codes in soft and hard decision to verify the 2-dB gain are also presented. The Viterbi decoding error propagation for M-level line codes for the assessment of the stability and the accuracy of our Viterbi decoding algorithm is investigated in Section 4. Finally a conclusion is presented in Section 5 to compare between the obtained results and present the advantages and disadvantages of codes inheriting the convolutional codes structures.

2. DESIGN OF M-LEVEL LINE CODES

The techniques of spectral shaping and distance mappings are combined and implemented to permutation sequences to generate our new designed M-level line codes [15]. As line codes are usually DC-free codes, we make use of the spectral shaping technique to shape the spectrum of our M-level line codes to suit certain applications. The use of distance mappings technique is actually for the purpose of having codes with better error correction capability inheriting the trellis structure from the base codes which are the convolutional codes. This makes the use of the Viterbi decoding algorithm possible.

2.1 Spectral Shaping M-level Line Codes

Shaping the spectrum of any sequence whether it is binary or non-binary to create nulls at certain frequencies is the same as forcing the power spectral density (PSD) function to zero at those corresponding frequencies [16]. The spectral shaping technique is usually applied to baseband data stream, which is represented by the vector $y = (y_1, y_2, \ldots, y_M)$. We make use in this paper of spectral null equations to create nulls at rational submultiples of the symbol frequency. The reason of the use of non binary or permutation sequences is to be able to generate multilevel pulse amplitude modulated signals. The design of DC-free M-level line codes is also considered in our work.

For a codeword of length, M, there exists an integer multiple of k, where

$$M = ks.$$

The frequency of value $f = r/k$ represents the spectral nulls at rational sub multiples r/k, with r as an integer. To generate nulls at those frequencies, we have to satisfy [17]

$$A_1 = A_2 = \cdots = A_k, \tag{1}$$

If all the codewords in a codebook satisfy these equations, the codebook will exhibit nulls at the required frequencies. This is true for binary and non binary sequences despite shaping non-binary sequences is much complicated than the binary ones since more constraints should be taken into considerations. Hence, we have to choose a suitable permutation sequences with suitable pulse amplitude channel symbols to satisfy the spectral shaping equation in (1).

Each permutation symbol (PS) is mapped to a channel symbol (CS), which represents the level of the signal. In general, for odd values of M, the symbol mapping is

$$\begin{array}{lcccccccc} \text{PS:} & 0 & 1 & \cdots & \frac{M-1}{2} & \cdots & M-2 & M-1 \\ & \downarrow & \downarrow & & \downarrow & & \downarrow & \downarrow \\ \text{CS:} & -\frac{M-1}{2} & -\frac{M-3}{2} & \cdots & 0 & \cdots & +\frac{M-3}{2} & +\frac{M-1}{2} \end{array}$$

and for even values of M, the symbol mapping is

$$\begin{array}{lccccccccc} \text{PS:} & 0 & 1 & \cdots & \frac{M-2}{2} & \frac{M}{2} & \cdots & M-2 & M-1 \\ & \downarrow & \downarrow & & \downarrow & \downarrow & & \downarrow & \downarrow \\ \text{CS:} & -\frac{M}{2} & -\frac{M-2}{2} & \cdots & -1 & +1 & \cdots & +\frac{M-2}{2} & +\frac{M}{2} \end{array}$$

As an example, if we allocate the channel symbols of -3 -1 $+3$ $+1$ to the permutation sequence 0132, then we will guaranty nulls at frequencies 0, $1/2$ and 1 since $M = 4$ and therefore $k = 2$.

2.2 Decoding Algorithm for M-level Line Codes

As was mentioned in the introduction, M-level line codes usually use combinational logic decoders. To get benefit of the Viterbi decoder, we make use of the distance mappings technique to present our new designed M-level line codes in a state machine [18] form. The technique is simply mapping the outputs of a convolutional code to other codewords from a code with lesser error-correction capabilities. In our case we use the spectrally shaped permutation sequences. This mapping will allow us to obtain suitably well shaped output code sequences and better decoded by using the Viterbi algorithm [13, 19, 20]. Our new codes are also called as M-level line trellis codes in view of their trellis structure that is inherited from the base codes, the convolutional codes.

The technique of mapping was introduced by Ferreira *et al* in their papers [12] and [13], where they have shown how the output binary n-tuple code symbols from an $R = m/n$ convolutional code can be mapped to non-binary M-tuple permutation code symbols, thereby creating a permutation trellis code.

Ferreira *et al* have introduced two types of matrices $\mathbf{D} = [\mathbf{d_{ij}}]$ and $\mathbf{E} = [\mathbf{e_{ij}}]$, which are respectively related to the Hamming distances between the codewords of the base code, the convolutional code and the mapped code, the M-level line code.

As an example, we take the mapping of the set of binary 2-tuple code symbols, $\{00, 01, 10, 11\}$ to a set of 4-tuples spectral null codewords, $\{0123, 0231, 3102, 3201\}$.

For this mapping we have

$$\mathbf{D} = \begin{bmatrix} 0 & 1 & 1 & 2 \\ 1 & 0 & 2 & 1 \\ 1 & 2 & 0 & 1 \\ 2 & 1 & 1 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{E} = \begin{bmatrix} 0 & 2 & 2 & 4 \\ 2 & 0 & 4 & 2 \\ 2 & 4 & 0 & 2 \\ 4 & 2 & 2 & 0 \end{bmatrix}.$$

It is clear that $e_{ij} \geq d_{ij} + 1$, $\forall i \neq j$, and this guarantees an increase in the distance of the resulting code.

In general, if $e_{ij} \geq d_{ij} + \delta$, $\delta \in \{1, 2, \ldots\}$, $\forall i \neq j$ we call such mappings *distance-increasing mappings* (DIMs). In the case where $e_{ij} \geq d_{ij}$, $\forall i \neq j$ and equality achieved at least once, we have *distance-conserving mappings* (DCMs). Finally, if $e_{ij} \geq d_{ij} + \delta$, $\delta \in \{-1, -2, \ldots\}$, $\forall i \neq j$, we have *distance-reducing mappings* (DRMs).

Since we make use of spectral null technique combined with distance mapping technique, we denote our new designed M-level line codes as SNDM-line codes.

2.3 Examples of Designed Codes

To design our M-level line codes with well shaped spectrum, we need to start with a permutation sequence that will lead to the construction of our multilevel codebook. By using the property of commutativity for addition between the variables in (1) to permute the channel symbols of these variables and to keep the spectral null property satisfied, we can generate our spectrally shaped M-level line codewords or sequences. The swapping of our permutation symbols needs a special construction algorithm that will help in avoiding repetitive sequences which cause loss on the calculation of distances between the generated codewords of our code and therefore the error correction capability will be less. Here we make use of the cube graph construction [21], which has proven to be optimum.

As was explained before, in the spectral null equation and for sequences of length $M = ks$, we have k groupings with

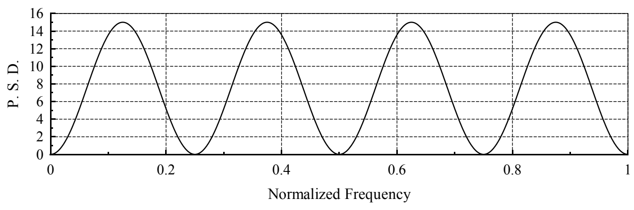

Figure 1: PSD for $M = 8$ and $k = 4$

s symbols in each grouping. Hence, our mappings will consist of several smaller mappings. For each A_i we have a mapping of length s that is used to permute the y_i in the grouping. For all the A_i as a grouping we have another mapping of length k that is used to permute the A_i. The following example illustrates this.

Example 1 For $M = 8$, with $k = 4$ and $s = 2$, we have the following spectral null equation:

$$\underbrace{\underbrace{\overbrace{y_1+y_5}^{s=2}}_{A_1} = \underbrace{\overbrace{y_2+y_6}^{s=2}}_{A_2} = \underbrace{\overbrace{y_3+y_7}^{s=2}}_{A_3} = \underbrace{\overbrace{y_4+y_8}^{s=2}}_{A_4}}_{k=4} \tag{2}$$

Let $\texttt{swap}(y_a,y_b)$ denote the swapping of symbols in the variables y_a and y_b. The following sequences can then be obtained from the original SN sequence. All swaps are presented in (3).

The swaps in (3), can be written in an algorithm form to show in details all steps in the generation of our new designed M-level line codes. The inputs x_i represent the outputs of the convolutional codes.

```
Input: $(x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8)$
Output: $(y_1,y_2,y_3,y_4,y_5,y_6,y_7,y_8)$
$(y_1,y_2,y_3,y_4,y_5,y_6,y_7,y_8) \leftarrow (0,1,2,3,7,6,5,4)$
begin
  if $x_1 = 1$ then swap$(y_1,y_5)$
  if $x_2 = 1$ then swap$(y_2,y_6)$
  if $x_3 = 1$ then swap$(y_3,y_7)$
  if $x_4 = 1$ then swap$(y_4,y_8)$
  if $x_5 = 1$ then swap$(y_1,y_2)(y_5,y_6)$
  if $x_6 = 1$ then swap$(y_3,y_4)(y_7,y_8)$
  if $x_7 = 1$ then swap$(y_1,y_3)(y_5,y_7)$
  if $x_8 = 1$ then swap$(y_2,y_4)(y_6,y_8)$
end.
```

□

We can see from the algorithm that the permutation sequence that we start our swapping with is 01237654 and this to make sure that the corresponding channel symbols for this sequence satisfies the spectral null equation (2). The designed M-level line code will generate spectral nulls at frequencies 0, 1/4, 1/2, 3/4 and 1 as depicted in Fig. 1.

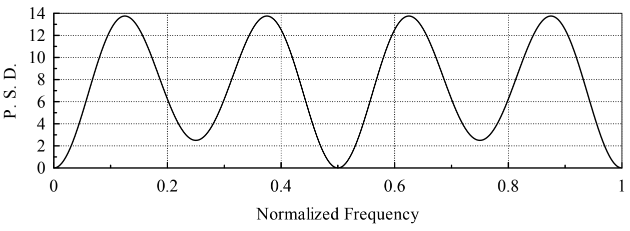

Figure 2: PSD for $M = 8$ and $k = 2$

Example 2 For the case of $M = 8$, with $k = 2$ and $s = 4$, we have the following spectral null equation:

$$\underbrace{\underbrace{\overbrace{y_1+y_3+y_5+y_7}^{s=4}}_{A_1} = \underbrace{\overbrace{y_2+y_4+y_6+y_8}^{s=4}}_{A_2}}_{k=2} \tag{4}$$

The corresponding algorithm to generate our M-level line code is as follows:

```
Input: $(x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8,x_9)$
Output: $(y_1,y_2,y_3,y_4,y_5,y_6,y_7,y_8)$
$(y_1,y_2,y_3,y_4,y_5,y_6,y_7,y_8) \leftarrow (0,1,3,2,4,5,7,5)$
begin
  if $x_1 = 1$ then swap$(y_1,y_3)$
  if $x_2 = 1$ then swap$(y_5,y_7)$
  if $x_3 = 1$ then swap$(y_1,y_5)$
  if $x_4 = 1$ then swap$(y_3,y_7)$
  if $x_5 = 1$ then swap$(y_2,y_4)$
  if $x_6 = 1$ then swap$(y_6,y_8)$
  if $x_7 = 1$ then swap$(y_2,y_6)$
  if $x_8 = 1$ then swap$(y_4,y_8)$
  if $x_9 = 1$ then swap$(y_1,y_2)(y_3,y_4)\cdots$
                              $\cdots(y_5,y_6)(y_7,y_8)$
end.
```

When the corresponding channel symbols for this sequence satisfies the spectral null equation (4), the designed M-level line code will generate spectral nulls at frequencies 0, 1/2, and 1 as depicted in Fig. 2. □

In general, we have to conduct the swaps based on the k-cube construction algorithm [21] to respect the distance between the indices of the permutation sequences y_i. And the general algorithm for our codes mapping is summarized as follows:

1. Comparison between n (convolutional outputs codewords length) and M (permutation sequence length).
   (a) $n > M$: Reducing mappings
   (b) $n < M$: Increasing mappings
   (c) $n = M$: Conserving mappings

$$
\begin{array}{lccccccc}
 & \overbrace{y_1 + y_5}^{A_1} & = & \overbrace{y_2 + y_6}^{A_2} & = & \overbrace{y_3 + y_6}^{A_3} & = & \overbrace{y_4 + y_8}^{A_4} \\
\text{SN sequence} & -7 + +7 & = & -5 + +5 & = & -5 + +5 & = & -5 + +5 \\
\hline\hline
\mathtt{swap}(y_1,y_5) & +7 + -7 & = & -5 + +5 & = & -5 + +5 & = & -5 + +5 \\
\mathtt{swap}(y_2,y_6) & -7 + +7 & = & +5 + -5 & = & -5 + +5 & = & -5 + +5 \\
\mathtt{swap}(y_1,y_5) & +7 + -7 & = & -5 + +5 & = & -5 + +5 & = & -5 + +5 \\
\mathtt{swap}(y_2,y_6) & -7 + +7 & = & +5 + -5 & = & -5 + +5 & = & -5 + +5 \\
\mathtt{swap}(y_1,y_2)(y_5,y_6) & -5 + +5 & = & -7 + +7 & = & -5 + +5 & = & -5 + +5 \\
\mathtt{swap}(y_1,y_2)(y_5,y_6) & -5 + +5 & = & -7 + +7 & = & -5 + +5 & = & -5 + +5 \\
\mathtt{swap}(y_1,y_2)(y_5,y_6) & -5 + +5 & = & -7 + +7 & = & -5 + +5 & = & -5 + +5 \\
\mathtt{swap}(y_1,y_2)(y_5,y_6) & -5 + +5 & = & -7 + +7 & = & -5 + +5 & = & -5 + +5 \\
\hline\hline
\end{array}
\tag{3}
$$

2. Mappings
   - x_i, $1 \leq n$
   - y_j, $1 \leq j \leq 2^{\lceil log_2 M \rceil}$
   - for $i = 1 : n$ if $x_i = 1$ then swap (y_j, y_{j+1}) end.

3. VITERBI DECODING M-LEVEL LINE CODES

In the literature, it was shown that with soft decisions Viterbi decoding, we have an improvement of 2 dB gain over hard decisions [22].

Following are a few examples of published M-level line codes including our designed codes. We run our simulation for soft and hard decisions for each of these M-level line codes and see if all of them have achieved the 2 dB gain difference between soft and hard decisions.

All M-level line codes investigated in this section are represented in a state machine form and this for the sake of using the Viterbi decoding algorithm.

The values of bit error rate (BER) corresponding to different values of signal to noise ration (SNR) for the mapped code are less than those of the base code. Besides that we have to emphasize the fact that mapped code using the Euclidean distance is slightly outperforming the mapped code using the Hamming distance. This is almost related to the fact that the Euclidean distance is used to refine the distance metrics and thus a soft decision could be used.

3.1 Three-Level Line Codes: Ternary Line Codes

Ternary line codes [24, 25] are often used on channels such as PCM metallic cable systems with transformer decoupling and repeaters. As the first world countries have started moving to high speed DSL technologies, using other modulation techniques, the developing countries however, the already installed digital subscriber loops, utilizing line codes such as AMI or HDB3, will still have to function for many years to come.

High Density Bipolar n (HDBn): This class of ternary line codes, used in European countries, has a maximum number of consecutive zeros to be limited to n. HDBn codes are considered to be good codes for better synchronization of receiver and transmitter and with low frequency cut-off point provided in power spectral density function. Fig. 3 shows the general form of the state machine of this class of codes.

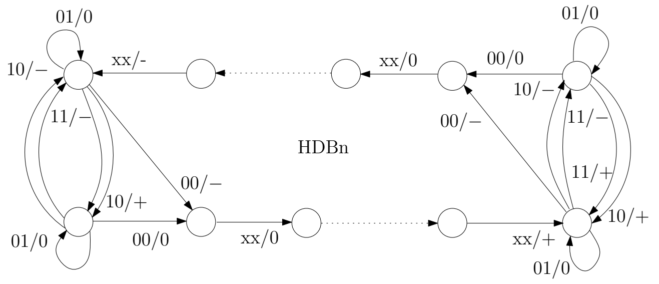

Figure 3: State Machine of HDBn.

If we take the case of $n = 3$, the digital data in HDB3 encoding is represented in almost identical fashion to AMI except for allowances made to accommodate certain violation as will be explained later.

The patterns of HDB3 codes are described as there is no changes in voltage for a sequence of 0s is solved by changing any incidence of four consecutive '0' bits into

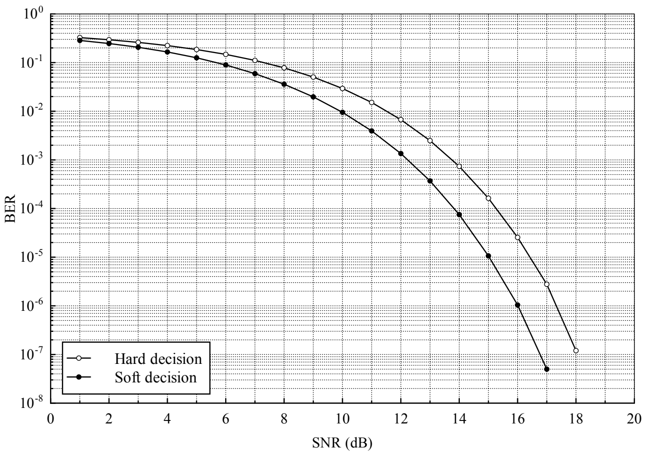

Figure 4: HDB3: Viterbi decoding soft and hard decisions.

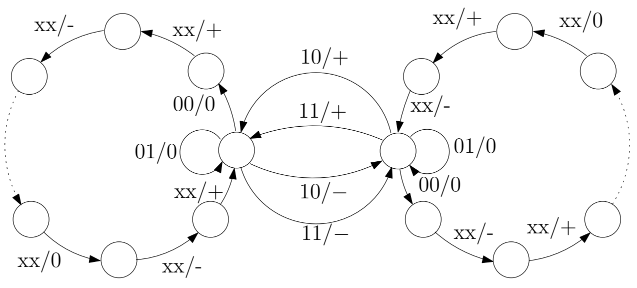

Figure 5: State Machine of BnZS.

a stream containing 000V, where the polarity of the V bit is the same as the previous non-0 voltage (opposite to a '1' bit, which causes a V signal with an alternate voltage according to the previous one). But a new problem arises - because the polarity of the non-zero bits is the same, a non-zero DC level is formed. This is overcome by changing the polarity of the V bit to the opposite of the previous V bit. This changes the bit stream to B00V, where the polarity of the B bit is the same as the polarity of the V bit. The change "fools" the receiver into thinking a received B bit is a '1' bit, but when it receives the V bit (with the same polarity), it understands the B and the V bits as a '0'. In HDB3, the maximum number of consecutive zeros allowed in the substituted string is 3.

Using the same simulation setup as previously done with AMI, we found that the difference between Soft and Hard decisions decoding is near the 2 dB gain at the $BER = 10^{-6}$ as depicted in Fig. 4.

Binary n Zeros Substitution (BnZS): The encoder for BnZS codes uses the 0VB0VB filling pattern. In Fig. 5, the convention of using a bold transition arrow labeled XX/output has been introduced to indicate that transitions and outputs for all four input combinations are the same. Otherwise, the input of the encoder is arranged in the same way as the output of detection, where the most significant bit represents the delayed data and the least significant bit represents the all-zeros flag.

A simple modification to the output code converts the filling sequence to B0VB0V, or indeed any desired filling pattern. Care must be taken only to ensure that the first signed pulse of the filling sequence is indeed a B or V pulse as required.

Using the VBVB filling pattern, the B4ZS line code consists of 18 states arranged symmetrically around a horizontal center line.

Every transition from a state in the upper-half, following a data '1', has its destination in the lower half, and vice versa. This feature corresponds to adherence to the bipolar alternation rule. The pair of states at the left-hand side of the state diagram is occupied whenever the data contains a long string of consecutive data 1s. They can be considered to be the remnants of the parent bipolar encoder, with its data '0' self-loops replaced by the remaining 16 states.

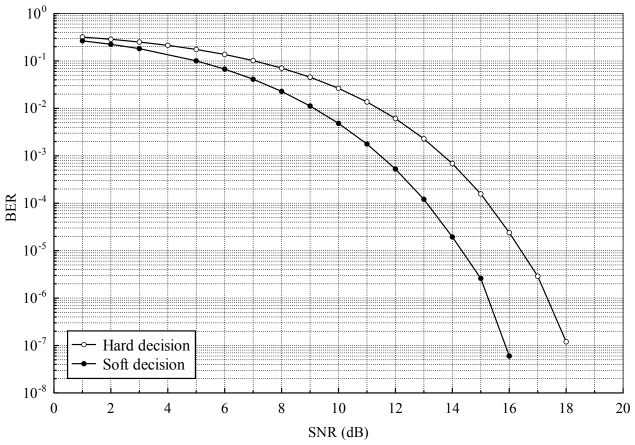

Figure 6: B4ZS: Viterbi decoding soft and hard decisions.

One of the pair of states at the right-hand side of the diagram is occupied whenever a data '1' is followed by 3 consecutive data 0s. Exiting from them on a data '0' corresponds to commencing the filling sequence. Considering just the upper state of the pair, it is entered only by an arc associated with the previous output +, and on data '0', it begins producing the output sequence $+--+$, that is VBVB.

Using the same simulation setup, Fig. 6 shows the 2 dB gain between hard and soft decisions Viterbi decoding.

SNDM-3Binary 6Ternary Line Code: (SNDM-3B6T): As was explained earlier, this class of codes is actually the combination of two techniques, which are the spectral shaping and distance mappings techniques. The SNDM-3B6T code is a ternary line code where we map 3 binaries to six ternaries. Our base code which is the convolutional code has a code rate of $R = 3/4$ and a constraint length of $K = 3$. The four bit outputs will be mapped onto six permutation symbols which are in fact repetitive symbols for the sake to drop the pulse amplitude modulated levels to three as it will be depicted in the following algorithm.

We consider the case of $M = 6$ with $k = 2$ and $s = 3$. The channel symbols must satisfy

$$\underbrace{\underbrace{\overbrace{y_1 + y_3 + y_5}^{s=3}}_{A_1} = \underbrace{\overbrace{y_2 + y_4 + y_6}^{s=3}}_{A_2}}_{k=2}. \tag{5}$$

We can see from (5) that we can assign two input bits to swap y_i in each equation, which means that we need four input bits in total to swap all symbols. Since we have chosen by purpose that all symbols are the same in each equation, then we do not need to swap A_1 and A_2 and therefore no need for extra input bits for swapping. Thus the convolutional base code may have a rate of $R = 3/4$.

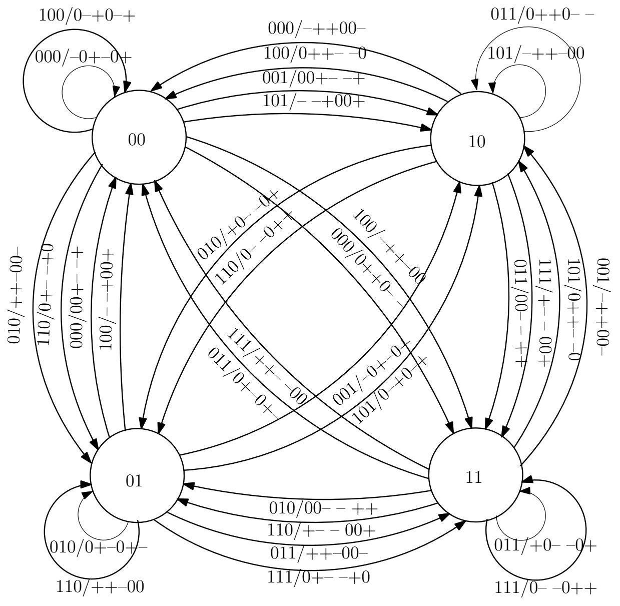

Figure 7: State machine of SNDM-3B6T.

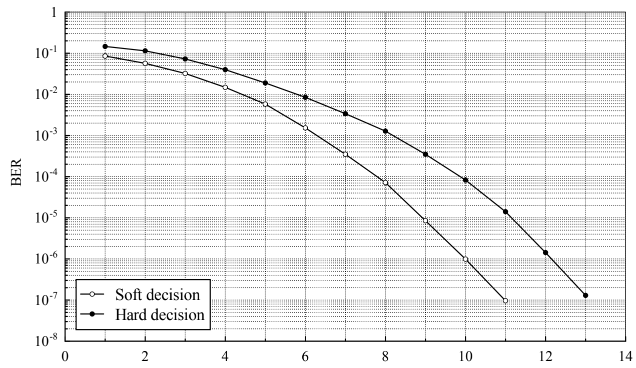

Figure 8: SNDM-3B6T: Viterbi decoding soft and hard decisions.

The mapping algorithm for the SNDM-3B6T code is described below.

```
Input: $(x_1,x_2,x_3,x_4)$
Output: $(y_1,y_2,y_3,y_4,y_5,y_6)$
$(y_1,y_2,y_3,y_4,y_5,y_6) \leftarrow (0,1,2,0,1,2)$
begin
  if $x_1 = 1$ then swap$(y_1,y_3)$
  if $x_2 = 1$ then swap$(y_1,y_5)$
  if $x_3 = 1$ then swap$(y_2,y_4)$
  if $x_4 = 1$ then swap$(y_2,y_6)$
end.
```

The state machine of the resultant SNDM-3B6T code, is presented in Fig. 7.

The BER performance of this code is depicted in Fig. 8. We can see clearly the 2 dB gain between soft and hard decisions at $BER = 10^{-6}$.

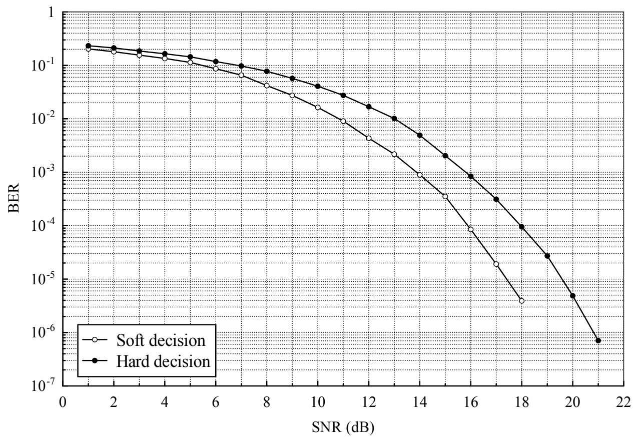

Figure 9: SNDM-7B8Q: Viterbi decoding soft and hard decisions.

3.2 *Four-Level Line Codes: Quaternary Line Codes*

Quaternary line codes, as the 2B1Q line code, have been used in transmission and also been used in modem ISDN circuits. Following are a few examples of quaternary line codes.

SNDM-7Binary 8Quaternary (SNDM-7B8Q): We consider the example of permutation sequences of length $M = 8$ for the case of $k = 2$ and $s = 4$. The symbols in each grouping A_i are equal or repeating and the resultant permutation sequence is an eight symbol sequence with four channel symbols, which will be used to generate a 4-Level line code. As explained previously, we need 8 bits from the convolutional code's output to be able to swap all the y_i symbols. Since the two groupings A_1 and A_2 are equal, then there is no need for input bits to swap them. Therefore the convolutional base code may have a rate of $7/8$. The mapping algorithm for the SNDM-7B8Q code is described below.

```
Input: $(x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8)$
Output: $(y_1,y_2,y_3,y_4,y_5,y_6,y_7,y_8)$
$(y_1,y_2,y_3,y_4,y_5,y_6,y_7,y_8) \leftarrow (0,0,1,1,2,2,3,3)$
begin
  if $x_1 = 1$ then swap$(y_1,y_3)$
  if $x_2 = 1$ then swap$(y_3,y_7)$
  if $x_3 = 1$ then swap$(y_1,y_5)$
  if $x_4 = 1$ then swap$(y_3,y_7)$
  if $x_5 = 1$ then swap$(y_2,y_4)$
  if $x_6 = 1$ then swap$(y_6,y_8)$
  if $x_7 = 1$ then swap$(y_2,y_6)$
  if $x_8 = 1$ then swap$(y_4,y_8)$
end.
```

The BER performance of this code is depicted in Fig. 9. We can see clearly the 2 dB gain between the soft and hard decisions.

Table 1: Encoder for 2B1QI line code

	Inputs							
	00		01		10		11	
State	Outputs	Next State	Outputs	Next State	Outputs	Next State	Outputs	Next State
A	0	B	-1	C	+1	D	+2	E
B	-2	A	-1	C	+1	D	+2	E
C	-2	A	0	B	+1	D	+2	E
D	-2	A	-1	C	0	B	+2	E
E	-2	A	-1	C	+1	D	0	B

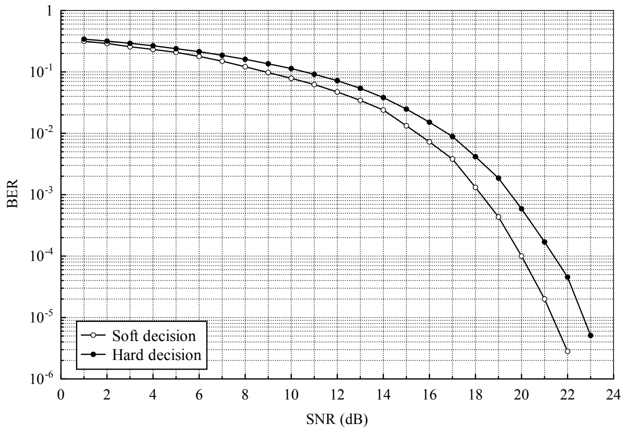

Figure 10: 2B1QI: Viterbi decoding soft and hard decisions.

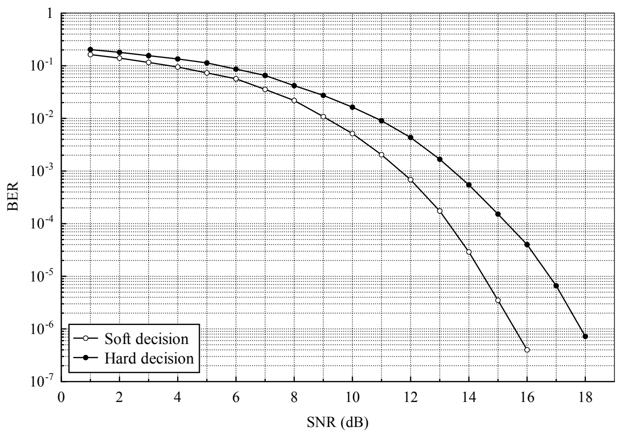

Figure 11: SNDM-4B6QI: Viterbi decoding soft and hard decisions.

3.3 *Five-Level Line Codes*

2 Binary 1 Quaternary Inverse Line Code: (2B1QI): This code, as its encoder is presented in Table 1, is considered to be similar to the well known 2B1Q quaternary line code [9], except that this line code is with 5-levels and which give him a favorably built-in properties for clock extraction.

The BER performance of this code is depicted in Fig. 10. We can see clearly the 2 dB gain between the soft and hard decisions.

SNDM-4Binary 6Quaternary Inverse (SNDM-4B6QI): We take the case of $M = 6$ with $k = 2$ and $s = 3$. The channel symbols must satisfy (5). By repeating only one symbol in both A_1 and A_2, we can make the number of symbols equal to five. The corresponding five channel-level symbols can generate a 5-Level line code. To chose the convolutional base code rate, we can see that we need 4 bits to swap all the symbols y_i and just 1 bit to swap A_1 and A_2 since $k = 2$. In total, we need 5 bits from the outputs of the corresponding convolutional base code, which means that we need a convolutional code with a rate of $R = 4/5$.

The new designed code belongs to the quaternary line codes. This code is called a 4Binary 6Quaternary Inverse and denoted by SNDM-4B6QI. The mapping algorithm for the SNDM-4B6QI code is described below.

```
Input: $(x_1,x_2,x_3,x_4,x_5)$
Output: $(y_1,y_2,y_3,y_4,y_5,y_6)$
$(y_1,y_2,y_3,y_4,y_5,y_6) \leftarrow (4,0,2,4,1,3)$
begin
  if $x_1 = 1$ then swap$(y_1,y_3)$
  if $x_2 = 1$ then swap$(y_1,y_5)$
  if $x_3 = 1$ then swap$(y_2,y_4)$
  if $x_4 = 1$ then swap$(y_2,y_6)$
  if $x_5 = 1$ then swap$(y_1,y_2)(y_3,y_4)(y_5,y_6)$
end.
```

The BER performance of this code is depicted in Fig. 11. We can see clearly the 2 dB gain between the soft and hard decisions.

3.4 *Six Level Line Codes*

4B2H Line Code: This new line code [9] with encoder presented in Table 2, generates 6-level from permutation sequences. Different to the published results, we make use of the Viterbi decoding algorithm since a state machine presentation was given to this code. The BER performance of this code is depicted in Fig. 12. We can see clearly the 2 dB gain between the soft and hard decisions.

Table 2: Encoder for 4B2H Line Code

	State									
	A		B		C		D		E	
Input	Outputs	Next State	Outputs	Next State	Outputs	Next State	Outputs	Next State	Outputs	Next State
0000	-1+1	A	-1+1	B	-1+1	C	-1+1	D	-1+1	E
0001	+1-1	A	+1-1	B	+1-1	C	+1-1	D	+1-1	E
0010	+3-3	A	+3-3	B	+3-3	C	+3-3	D	-5-3	A
0011	+3+5	E	-3+3	B	-3+3	C	-3+3	D	-3+3	E
0100	+5+3	E	-3+1	A	-3+1	B	-3+1	C	-3+1	D
0101	+3+3	D	+3+3	E	-1-3	A	-1-3	B	-1-3	C
0110	+1+5	D	+1+5	E	+1-5	A	+1-5	B	+1-5	C
0111	+5+1	D	+5+1	E	-3+5	D	-3+5	E	-3-5	A
1000	+1+3	C	+1+3	D	+1+3	E	-5+3	C	-5+3	D
1001	+3+1	C	+3+1	D	+3+1	E	+3-5	C	-5-1	B
1010	-1+5	C	-1+5	D	-1+5	E	-3-1	B	-3-1	C
1011	+5-1	C	+5-1	D	+5-1	E	-5+1	B	-5+1	C
1100	+1+1	B	+1-3	A	+1-3	B	+1-3	C	+1-3	D
1101	+3-1	B	+3-1	C	+3-1	D	+3-1	E	-3-3	B
1110	-1+3	B	-1+3	C	-1+3	D	-1+3	E	-1-1	D
1111	+5-3	B	+5-3	C	+5-3	D	-1-5	A	-1-5	B

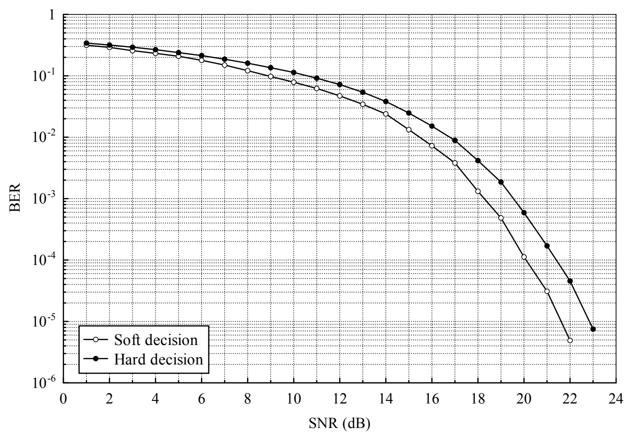

Figure 12: 4B2H: Viterbi decoding soft and hard decisions.

3.5 *High-Multilevel Line Codes*

For the generation of higher levels codes, we make use of our previous techniques the spectral shaping and distance mappings in order to generate pulse amplitude modulated line codes with m binary inputs and M symbol outputs or M channel levels, we denote them by SNDM-mBML. We study the case of $M = 12$ to explain our technique.

Example 3 For $M = 12$, with $k = 3$ and $s = 4$, we have

$$\begin{aligned} A_1 &= \overbrace{y_1 + y_4 + y_7 + y_{10}}^{s=4} \\ A_2 &= y_2 + y_5 + y_8 + y_{11} \\ A_3 &= y_3 + y_3 + y_6 + y_{12} \end{aligned} \qquad (6)$$

In this case where all symbols are not repeated, we need 4 bits to swap all symbols in each grouping. This will lead to 12 bits since we have three groupings. On the other side we need 2 bits to swap the three groupings as it is based on the cube construction. Therefore we need in total 14 bits to swap all the channel symbols. The corresponding convolutional base code rate should be convolutional code with a rate of $R = 13/14$. The corresponding mapping algorithm for this high level line codes is presented below. The designed code has nulls at the frequencies 0, 1/3, 2/3 and 1 as depicted in Fig. 13.

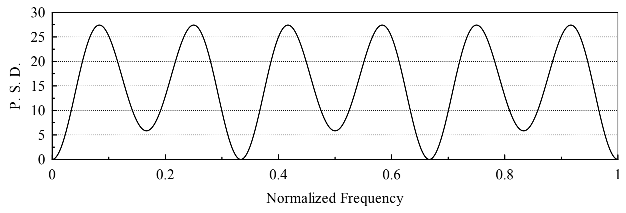

Figure 13: PSD of SNDM-13B12L Line code

```
Input: $(x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8,x_9,x_{10},x_{11},x_{12},$
$x_{13},x_{14})$
Output: $(y_1,y_2,y_3,y_4,y_5,y_6,y_7,y_8,y_9,y_{10},y_{11},y_{12})$
$(y_1,y_2,y_3,y_4,y_5,y_6,y_7,y_8,y_9,y_{10},y_{11},y_{12})$
$\leftarrow (0,1,2,5,4,3,6,7,8,11,10,9)$
begin
  if $x_1 = 1$ then swap$(y_1,y_4)$
  if $x_2 = 1$ then swap$(y_7,y_{10})$
  if $x_3 = 1$ then swap$(y_1,y_7)$
  if $x_4 = 1$ then swap$(y_4,y_{10})$
  if $x_5 = 1$ then swap$(y_2,y_5)$
  if $x_6 = 1$ then swap$(y_8,y_{11})$
  if $x_7 = 1$ then swap$(y_2,y_8)$
```

Table 3: Some high-level line codes

M	k	s	Frequency Nulls	Base Code Rate	Multilevel line code
8	4	2	0,1/4,1/2,3/4,1	7/8	SNDM-7B8L
8	2	4	0,1/2,1	8/9	SNDM-8B8L
12	4	3	0,1/4,1/2,3/4,1	11/12	SNDM-11B12L
12	3	4	0,1/3,2/3, 1	13/14	SNDM-13B12L
12	2	6	0,1/2,1	14/15	SNDM-14B12L
15	5	3	0,1/5,2/5,3/5,4/5,1	14/15	SNDM-14B15L
15	3	5	0,1/3,2/3, 1	16/17	SNDM-16B15L
16	4	4	0,1/4,1/2,3/4,1	19/20	SNDM-19B16L
16	8	2	0,1/8,1/4,3/8,1/2,5/8,3/4,7/8,1	19/20	SNDM-19B16L
16	2	8	0,1/2,1	24/25	SNDM-24B16L
18	2	9	0,1/2,1	26/27	SNDM-26B18L
18	9	2	0,1/9,2/9,1/3,4/9,5/9,2/3,7/9,8/9,1	20/21	SNDM-20B18L
18	3	6	0,1/3,2/3, 1	22/23	SNDM-22B18L
18	6	3	0,1/6,1/3,1/2,2/3,5/6,1	18/19	SNDM-18B18L
20	2	10	0,1/2,1	30/31	SNDM-30B20L
20	10	2	0,1/10,1/5,3/10,2/5, 1/2, 3/5,7/10,4/5,9/10,1	22/23	SNDM-22B20L
20	4	5	0,1/4,1/2,3/4,1	23/24	SNDM-23B20L
20	5	4	0,1/5,2/5,3/5,4/5,1	24/25	SNDM-24B20L

```
    if $x_8 = 1$ then swap$(y_5, y_{11})$
    if $x_9 = 1$ then swap$(y_3, y_6)$
    if $x_{10} = 1$ then swap$(y_9, y_{12})$
    if $x_{11} = 1$ then swap$(y_3, y_9)$
    if $x_{12} = 1$ then swap$(y_6, y_{12})$
    if $x_{13} = 1$ then swap$(y_1, y_2)(y_4, y_5)\cdots$
                                   $\cdots(y_7, y_8)(y_{10}, y_{11})$
    if $x_{14} = 1$ then swap$(y_1, y_3)(y_4, y_6)\cdots$
                                   $\cdots(y_7, y_9)(y_{10}, y_{12})$
end.
```

In view of the large number of states for the base codes, we will not be able to present their state machines in view of the space restriction in this paper. □

Table 3 presents a few examples of high-level line codes that we are able to design, taking into consideration the rates of the corresponding base codes.

4. VITERBI DECODING ERROR PROPAGATION

Viterbi algorithm is based on the calculation of the distances between the received and the expected transmitted information data in each branch of the trellis diagram designed from the state machine of the convolutional base code. Previous published work [18] has shown the modeling of line codes for the sake of having the state machine presentation, which will be used for the Viterbi algorithm.

A simulation experiment was conducted to analyze and prove that the coding gain is predominantly determined by the error propagation when Viterbi decoding fails. The experiment [28] is simply based on the generation of

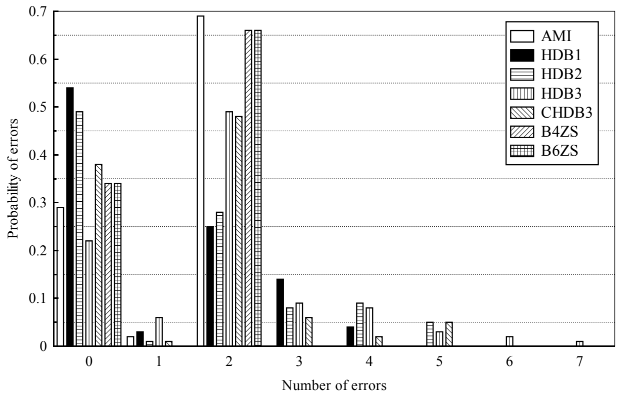

Figure 14: Error propagation of certain ternary line codes, due to a random single isolated channel error.

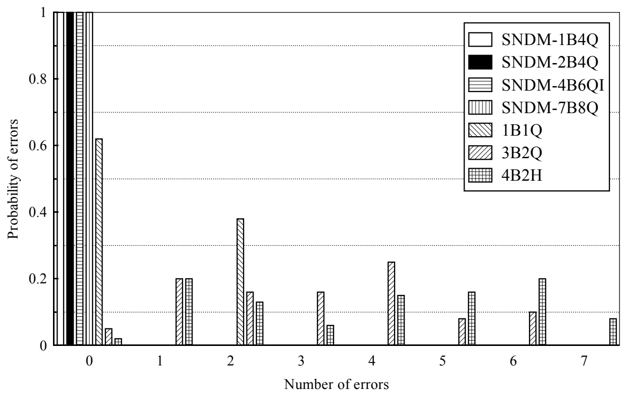

Figure 15: Error propagation of certain quaternary line codes, due to a random single isolated channel error.

Table 4: Probability of *i* errors

Multilevel Line Codes	Number of propagation errors *i*							
	0	1	2	3	4	5	6	7
AMI	0.29	0.02	0.69	0	0	0	0	0
HDB1	0.54	0.03	0.25	0.14	0.04	0	0	0
HDB2	0.49	0.01	0.28	0.08	0.09	0.05	0	0
HDB3	0.22	0.06	0.49	0.09	0.08	0.03	0.02	0.01
CHDB3	0.38	0.01	0.48	0.06	0.02	0.05	0	0
B4ZS	0.34	0	0.66	0	0	0	0	0
2B2T	0.17	0.11	0.02	0.16	0.54	0	0	0
2B2TA	0.16	0.07	0.58	0.19	0	0	0	0
SNDM-3B6T	1	0	0	0	0	0	0	0
SNDM-5B9T	1	0	0	0	0	0	0	0
1B1Q	0.62	0	0.38	0	0	0	0	0
3B2Q	0.05	0.2	0.16	0.16	0.25	0.08	0.1	0
4B2H	0.02	0.2	0.13	0.06	0.15	0.16	0.2	0.08
SNDM-1B4Q	1	0	0	0	0	0	0	0
SNDM-2B4Q	1	0	0	0	0	0	0	0
SNDM-4B6QI	1	0	0	0	0	0	0	0
SNDM-7B8Q	1	0	0	0	0	0	0	0

Table 5: Number of levels vs error propagation

Number of Codes Levels	Multilevel Line Codes	Gain using 3-bit quantization	Expected number of propagated errors	Maximum number of propagated errors
	AMI	1.90	1.40	3
Three	B4ZS	2.05	1.32	2
	B6ZS	2.00	1.34	2
	HDB1	1.85	1.40	5
	HDB2	1.55	1.42	6
	HDB3	1.45	1.48	6
	CHDB3	1.35	1.66	7
	2B2T	1.70	1.8	3
	2B2TA	1.50	2.79	4
	SNDM-3B6T	2.10	0	0
	SNDM-5B9T	2.00	0	0
	1B1Q	2.00	0.76	2
	2B1Q	1.40	3.17	6
Four	3B2Q	1.50	3.06	6
	SNDM-2B4Q	2.00	0	0
	SNDM-4B6Q	2.00	0	0
	SNDM-7B8Q	2.00	0	0
Five	2B1QI	1.30	3.2	6
	SNDM-4B6QI	2.1	0	0
Six	4B2H	1.1	3.8	7

widely separated single random errors between the levels of the code's symbols, and the observation of the number of errors propagated.

Table 4 shows that our designed codes have zero error propagation and this is expected in view of the convolutional codes properties that our codes are generated from. We can see also these results from another angle from Fig. 14 and 15.

Table 5 shows the variation of the number of error propagation and the values of the expected number of propagated error in terms of the number of the code's level.

So from Tables 4 and 5, we can summarize that the increase of the number of levels in line codes and the increase of the complexity of the pattern within the same class of line codes are the major factors in the increase of the error propagation of the Viterbi decoding. This is always not true if the codes are designed from the convolutional codes as in the case of our new designed M-level line codes.

It is clear from the previous results that the expected number of error propagation increases for the following reasons:

- When the complexity of the pattern increases, which is true even within the same class of line codes, the value of the expected error propagation increases. As an example, the error propagation for HDBn line codes, the expected number of error propagation is higher than the one for BnZS line codes. Similar case when we compare 2B2T line codes and 2B2TA line codes.
- When the number of levels increases, even within the same number of states, the expected number of error propagation increases. As an example, if we take the case of 2B1QI line code, which has five levels, we can see that the expected number of error propagation is higher than the one for 4B2H line codes since it has six levels. Both line codes have similar number of states which is five.

It is important to notice that the number of states is not an important criteria in the increase of the expected number of error propagation. This is clear with the following two examples. In the 2B1QI and 4B2H line codes which have similar number of states, we can see that they differ with the number of expected number of error propagation. In the case of 1B1Q and 3B2Q line codes, we can see that 1B1Q code has less expected number of error propagation than 3B2Q code despite it has more number of states.

5. CONCLUSION

We have combined two techniques, spectral shaping and distance mappings to design M-level line codes. This class of codes inherited the same state structure and distance properties as convolutional codes. Thus the use of the Viterbi algorithm as a decoding solution is possible.

The possible 2-dB gain is investigate and the results verified some theoretical assumption that made SNDM code outperforming other M-level line codes.

The obtained results have shown that the error propagation Viterbi decoding increases with the complexity of the pattern of the code as the case between the HDBn codes and BnZS line codes even both classes belong to the same class of M-level line codes which is the ternary line codes. As the number of levels, M, increase the complexity of coding increase as well and therefore the error propagation of the decoding algorithm increase. This problem is reduced and simplified with the design SNDM line codes.

High rate M-level line codes were also achieved with well shaped spectrum. Table 3 presents their corresponding spectral nulls frequencies. It is clear from Table 3 and from all previously presented PSD figures, that all our designed M-level line codes are also DC-free codes [29], which will help to overcome some communications problem like zero frequency components.

REFERENCES

[1] H. Imai and S. Hirakawa, "A new multilevel coding method using error-correcting codes", in *IEEE Transactions on Information Theory*, vol. 23, no. 3, pp. 371–377, May. 1977.

[2] U. Wachsmann and R. F. H. Fisher and J. B. Huber,"Multilevel codes: Theoretical concepts and practical design rules", in *IEEE Transactions on Information Theory*, vol. 45, no. 5, pp. 1361–1391, Jul. 1999.

[3] Y. Kofman and E. Zehavi and S. Shamai,"Performance analysis of a multilevel coded modulation system", in *IEEE Transactions on Communications*, vol. 42, no. 234, pp. 299–312, Feb. 1994.

[4] J. Wu and D. J. Costello Jr.,"New Multilevel Codes Over GF(q)", in *IEEE Transactions on Information Theory*, vol. 38, no. 3, pp. 933–939, May. 1992.

[5] A. R. Calderbank and M. A. Herro and V. Telang,"A Multilevel Approach to the Design of DC-free Line Codes", in *IEEE Transactions on Information Theory*, vol. 35, no. 3, pp. 579–583, May. 1989.

[6] R. M. Brooks and A. Jessop,"Line coding for optical fibre systems", in *International Journal of Electronics*, vol. 55, no. 1, pp. 81–120, 1983.

[7] S. D. Personick,"*Optical Fiber Transmission Systems*",New York: Plenum, 1981.

[8] C. Matrakidis and J. J. O'Reilly, "A Block decodable line code for high speed optical communication", in *Proceedings of the International Symposium on*

Information Theory, pp. 221, Ulm, Germany, June 29–Jul 4, 1997.

[9] L. Botha, H. C. Ferreira and I. Broere, "Multilevel sequences and line codes," in *IEE Proc. I Commun. Speech Vis.*, 1993, vol. 140, no. 3, pp. 255–261.

[10] J. K. Pollard, "Multilevel data communication over optical fiber," in *IEE Proc. I Commun. Speech Vis.*, 1991, vol. 138, no. 3, pp. 162–168.

[11] N. J. Lynch-Aird,"Review and Analytical comparison of Recursive and Nonrecursive Equalization Techniques for PAM Transmission Systems", in *International Journal on Electronics*,vol. 55, no. 1, pp. 81–120, 1983.

[12] H. C. Ferreira and A. J. H. Vinck, "Interference cancellation with permutation trellis codes," in *Proc. IEEE Veh. Technol. Conf. Fall 2000*, Boston, MA, Sep. 2000, pp. 2401–2407.

[13] H. C. Ferreira, A. J. H. Vinck, T. G. Swart and I. de Beer, "Permutation trellis codes," *IEEE Trans. Commun.*, vol. 53, no. 11, pp. 1782–1789, Nov. 2005.

[14] T. G. Swart and H. C. Ferreira, "A generalized upper bound and a multilevel construction for distance-preserving mappings," *IEEE Trans. Inf. Theory*, vol. 52, no. 8, pp. 3685–3695, Aug. 2006.

[15] K. Ouahada, T. G. Swart and H. C. Ferreira, "Permutation sequences and coded PAM signals with spectral nulls at rational submultiples of the symbols," accepted for publication in *Communications and Cryptography*, Springer, 2011.

[16] E. Gorog, "Alphabets with desirable frequency spectrum properties," *IBM J. Res. Develop.*, vol. 12, pp. 234–241, May 1968.

[17] K. A. S. Immink, *Codes for mass data storage systems*, Shannon Foundation Publishers, The Netherlands, 1999.

[18] D.B. Keogh, "Finite-State Machine Descriptions of Filled Bipolar Codes," *A.T.R.* vol. 18, no. 2, pp. 3-12, June 1984.

[19] T. G. Swart, I. de Beer, H. C. Ferreira and A. J. H. Vinck, "Simulation results for permutation trellis codes using M-ary FSK," in *Proc. Int. Symp. on Power Line Commun. and its Applications*, Vancouver, BC, Canada, Apr. 6–8, 2005, pp. 317–321.

[20] G. David Forney, JR., "The Viterbi Algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, Mar. 1973.

[21] K. Ouahada and H. C. Ferreira, "*k*-Cube construction mappings from binary vectors to permutation sequences," in proceedings of *IEEE International Symposium on Information Theory*, Seoul, South Korea, June 28July 3, 2009.

[22] K. Ouahada and H. C. Ferreira, "Simulation study of the performance of ternary line codes under Viterbi decoding," in *IEE Proc-Commun.*, vol. 151, no. 5, pp. 409–414, 2004.

[23] A. Viterbi and J. Omura, "*Principles of Digital Communication and Coding*, McGraw-Hill Kogakusha LTD, Tokyo Japan, 1979.

[24] Buchner J. B. "Ternary Line Codes," *Philips Telecommunications Review*, vol. 34, no. 2, pp. 72–86, June 1976.

[25] H. C. Ferreira, J. F. Hope, and A. L. Nel, "On Ternary Error Correcting Line Codes," *IEEE Trans. Commun.* vol. 37 no. 5, pp. 510–515, May 1989.

[26] A. Croisier, "Introduction to Pseudo-ternary Transmission Codes," *IBM J. Res. Develop.* , vol. 14, pp. 354–367, Jul. 1970.

[27] V. I. Johannes, "Comments on Compatible-High-Density Bipolar Codes: An Unrestricted transmission Plan for PCM Carriers," *IEEE Trans. Commun. Tech.*, vol. COM-20, no. 1, pp. 78–79, Feb. 1972.

[28] N. Q. Duc, "Line coding techniques for baseband digital transmission," in *Australian Telecommunications Research*, vol. 9, no. 1, pp. 351–357, 1977.

[29] G. L. Pierobon, "Codes for zero spectral density at zero frequency," *IEEE Trans. Inf. Theory*, vol. 30, pp. 435–439, Mar. 1984.

Notes

Notes

Notes