

# New DC-free Multilevel Line Codes With Spectral Nulls at Rational Submultiples of the Symbol Frequency

Khmaies Ouahada, Hendrik C. Ferreira and Theo G. Swart  
 Department of Electrical and Electronic Engineering Science  
 University of Johannesburg, P.O. Box 524, Auckland Park, 2006, South Africa  
 Email: ktw@ing.rau.ac.za, hcferreira@uj.ac.za, ts@ing.rau.ac.za

**Abstract**—A new technique of designing multilevel line codes is presented. Distance-preserving mappings from binary sequences to permutation sequences are used, where the permutation sequences also have spectral nulls properties. The resultant line codes use the structure of the convolutional codes and thus have certain advantages compared to other published line codes.

## 1. Introduction

Mappings from a set of binary sequences to permutation sequences with spectral nulls properties are considered.

Permutation mappings originated with the suggestion by Ferreira and Vinck [1] of using permutation trellis codes, combined with M-FSK modulation, in power-line communications. They combined convolutional codes with permutation codes by using a distance-preserving mapping. Since then, much research have been done on permutation mappings to find new algorithms and better mappings [2]–[5]. Ouahada *et al.* [6] have shown that binary spectral nulls codes are useful to compensate for fading in a channel.

Previous investigations have shown that multilevel line codes are preferable to binary codes for high speed digital transmission as the case for optical fiber channel [7]. The additional signal levels or symbols in a multilevel sequence can be used to reduce the symbol rate and hence the bandwidth of the coded signal. The lower switching rate required can also be used to obtain higher data-transfer rates in an optical LAN system where the transmission rate is limited by CMOS technology [8].

In this paper we will present the combination of distance-preserving mappings (DPMs) with spectral nulls (SN) codes. The resulting codes are called SN-DPM codes.

The technique of designing a baseband data stream to have a spectrum with nulls occurring at certain frequencies is the same as having the power spectral density (PSD) function equal to zero at those frequencies [9]. We present the codeword length,  $M$ , as an integer multiple of  $k$ , then

$$M = ks,$$

where  $f = r/k$  represents the spectral nulls at rational

submultiples  $r/k$  [10]. We have to satisfy

$$A_1 = A_2 = \dots = A_k, \quad (1)$$

where

$$A_i = \sum_{\lambda=0}^{s-1} y_{i+\lambda k}, \quad i = 1, 2, \dots, k. \quad (2)$$

If all the codewords in a codebook satisfy these equations, the codebook will exhibit nulls at the required frequencies.

The spectral nulls codebook for  $k = 2$  and  $s = 2$  is:

$$C = \begin{Bmatrix} -1 & -1 & -1 & -1 \\ -1 & -1 & +1 & +1 \\ -1 & +1 & +1 & -1 \\ +1 & -1 & -1 & +1 \\ +1 & +1 & -1 & -1 \\ +1 & +1 & +1 & +1 \end{Bmatrix}$$

This is also true for non-binary codebooks, as will be the case in this paper.

In Section II we provide an overview on the concept of DPMs and defining three types of DPMs. In Section III we present few examples of the spectral nulls sequence, which is used to get the required spectrum. In Section IV we present a multilevel line code's construction based on the DPMs from binary sequences to permutation sequences. Two examples of multilevel line codes, which are quaternary line codes, are presented in Section V. We conclude with some final remarks in Section VI.

## 2. Distance-Preserving Mappings

Distance-preserving mappings is a coding technique, which maps the outputs of a convolutional code to other codewords from a code with lesser error-correction capabilities. The purpose behind this technique is to firstly obtain suitably constrained output code sequences and secondly to exploit the error correction characteristics of the new code with the use of the Viterbi algorithm [2], [11].

In [1] and [2] it was 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 as depicted in Fig. 1.

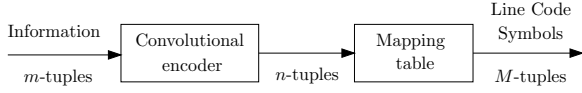


Fig. 1. Encoding process for a distance-preserving spectral nulls code

In this paper a similar mapping will be considered, but instead of a convolutional codes' output symbols, we will map all possible binary codewords of a certain length to permutation codewords, with the additional constraint that the permutation codewords are from a spectral nulls codebook. The mappings are applied to convolutional codes to create new permutation trellis codes with spectral nulls.

The matrices  $\mathbf{D} = [d_{ij}]$  and  $\mathbf{E} = [e_{ij}]$  are respectively the Hamming distance matrices between the codewords of the convolutional base code and the resulting mapped code (see [1] for an expanded definition).

**Example 1** We can map the set of binary 2-tuple code symbols,  $\{00, 01, 10, 11\}$  to a set of 4-tuples SN codewords,  $\{1243, 1342, 4213, 4312\}$  as shown in Fig. 2.

Taking the channel symbols into account (see the following section), we have a spectral nulls codebook as

$$\left\{ \begin{array}{l} (1) (2) (4) (3), \\ (1) (3) (4) (2), \\ (4) (2) (1) (3), \\ (4) (3) (1) (2) \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} (-3) (-1) (+3) (+1), \\ (-3) (+1) (+3) (-1), \\ (+3) (-1) (-3) (+1), \\ (+3) (+1) (-3) (-1) \end{array} \right\}.$$

This set guarantees nulls at frequencies 0,  $1/2$  and 1. To simplify the presentation in the rest of the paper, 1243 will be used instead of  $-3 -1 +3 +1$ .

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.  $\square$

In general, if  $e_{ij} \geq d_{ij} + \delta, \delta \in \{1, 2, \dots\}, \forall i \neq j$  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  $e_{ij} \geq d_{ij} + \delta, \delta \in \{-1, -2, \dots\}, \forall i \neq j$ , we have *distance-reducing mappings* (DRMs).

We use  $\mathcal{N}_k(M, n, \delta)$  to denote the SN-DPMs from binary sequences of length  $n$  to permutation sequences of length  $M$  with spectral nulls at multiples of  $1/k$ . The type of mapping is indicated by  $\delta$ , with  $\delta > 0$  for DIMs,  $\delta = 0$  for DCMs and  $\delta < 0$  for DRMs.

### 3. Spectral Nulls Sequence

The mapping sequence that we have to start with to satisfy the spectral nulls equation in (1) is called *SN*

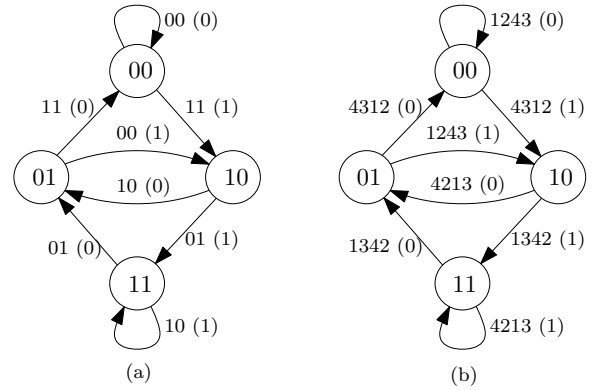


Fig. 2. State systems for the (a) convolutional code and (b) SN-DPM code

*sequence*. Each permutation symbol (PS) is mapped to a channel symbol (CS), which could represent the voltage level. In general, for odd  $M$  the symbol mapping is

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

and for even  $M$  the symbol mapping is

$$\begin{array}{cccccc} \text{PS:} & 1 & 2 & \dots & \frac{M}{2} & \frac{M+2}{2} & \dots & M \\ & \downarrow & \downarrow & & \downarrow & \downarrow & & \downarrow \\ \text{CS:} & -\frac{M}{2} & -\frac{M-2}{2} & \dots & -1 & +1 & \dots & +\frac{M}{2} \end{array}$$

Then it is just a matter of placing the symbols in such a way that the SN equations are satisfied.

**Example 2** Consider an SN sequence for  $M = 8$  and  $k = 4$ , with channel symbols assigned as follows

$$\begin{array}{cccccc} \text{PS:} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ \text{CS:} & -4 & -3 & -2 & -1 & +1 & +2 & +3 & +4 \end{array}$$

From (1) and (2) we must satisfy

$$y_1 + y_5 = y_2 + y_6 = y_3 + y_7 = y_4 + y_8, \quad (3)$$

with each grouping having two elements with indices differing by  $k$ . This means that if  $y_i$  is mapped to a negative channel symbol, the corresponding positive channel symbol must be mapped to  $y_{i+k}$ , as follows

$$\begin{array}{cccccc} \text{PS:} & 1 & 2 & 3 & 4 & 8 & 7 & 6 & 5 \\ & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ \text{CS:} & -4 & -3 & -2 & -1 & +4 & +3 & +2 & +1 \\ & y_1 & y_2 & y_3 & y_4 & y_5 & y_6 & y_7 & y_8 \end{array}$$

Clearly this satisfies (3) and the required SN sequence that generates nulls at frequencies 0,  $1/2$  and 1 is therefore 12348765.  $\square$

**Example 3** Next, consider an SN sequence for  $M = 9$  with  $k = s = 3$ . From (1) and (2) we must satisfy

$$y_1 + y_4 + y_7 = y_2 + y_5 + y_8 = y_3 + y_6 + y_9.$$

We left shift the symbols in groupings of three, as follows

$$\begin{array}{ccccccccc} \text{PS:} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ \text{CS:} & -4 & -3 & -2 & -1 & 0 & +1 & +2 & +3 & +4 \\ & \underbrace{\hspace{2em}}_{0\text{-shift}} & & \underbrace{\hspace{2em}}_{1\text{-shift}} & & & \underbrace{\hspace{2em}}_{2\text{-shifts}} & & & \\ & & & & \downarrow & & & & & \\ \text{PS:} & 1 & 2 & 3 & 5 & 6 & 4 & 9 & 7 & 8 \\ & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ \text{CS:} & -4 & -3 & -2 & 0 & +1 & -1 & +4 & +2 & +3 \\ & y_1 & y_2 & y_3 & y_4 & y_5 & y_6 & y_7 & y_8 & y_9 \end{array}$$

The SN sequence that generates nulls at frequencies 0, 1/3, 2/3, and 1 then is 123564978.  $\square$

**Example 4** We take the case of  $M = 10$ , with  $k = 2$  and  $s = 5$ . We cyclic-shift our permutation symbols with a step of  $s$  as depicted in Fig. 3.

$$\begin{array}{cccccccccc} \text{PS:} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ & & & & & \downarrow & & & & & \\ \text{shift} & 1 & 5 & 9 & 3 & 7 & 2 & 6 & 10 & 4 & 8 \\ & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ \text{CS:} & -5 & -1 & +4 & -3 & +2 & -4 & +1 & +5 & -2 & +3 \\ & y_1 & y_2 & y_3 & y_4 & y_5 & y_6 & y_7 & y_8 & y_9 & y_{10} \end{array} \quad \square$$

The SS sequence that generates nulls at frequencies 0, 1/2 and 1 then is (1)(5)(9)(3)(7)(2)(6)(10)(4)(8).

Table I presents selected SN sequences and the corresponding spectral nulls for different values of  $k$  and  $s$ .

## 4. Line Codes Construction

The property of commutativity for addition between the variables in (1) and (2), is used to permute the

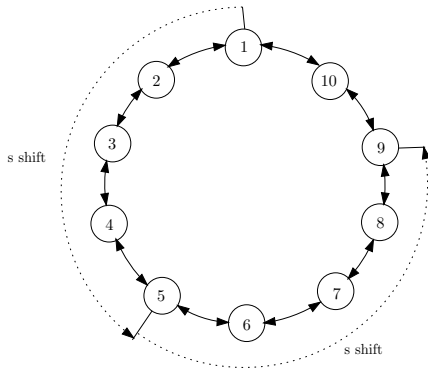


Fig. 3. Spectral nulls Sequence for  $k = 2$  and  $s = 5$

TABLE I  
SELECTED SN SEQUENCES FOR MULTILEVEL LINE CODES

$M$	$k$	Nulls	Sequence
4	2	$0, \frac{1}{2}, 1$	(1)(2)(4)(3)
6	2	$0, \frac{1}{2}, 1$	(1)(2)(4)(3)(5)(6)
6	3	$0, \frac{1}{3}, \frac{2}{3}, 1$	(1)(2)(3)(6)(4)(5)
8	4	$0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1$	(1)(2)(3)(4)(8)(7)(6)(5)
8	2	$0, \frac{1}{2}, 1$	(1)(2)(4)(3)(5)(6)(8)(7)
9	3	$0, \frac{1}{3}, \frac{2}{3}, 1$	(1)(2)(3)(5)(6)(4)(9)(7)(8)
10	5	$0, \frac{1}{5}, \frac{2}{5}, \frac{3}{5}, \frac{4}{5}, 1$	(1)(2)(3)(4)(5)(10)(9)(7)(8)(6)

symbols in these variables to satisfy the SN property and to generate our SN-DPMs codes.

Since  $M = ks$ , there are  $k$  groupings with  $s$  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  $A_i$ . Any of the previous mapping algorithms can be used for the smaller mappings, in this paper we make use of the multilevel construction [5]. The following example illustrates this.

**Example 5** For  $M = 8$ , with  $k = 4$  and  $s = 2$ , we have

$$\underbrace{\underbrace{y_1 + y_5}_{s=2}}_{A_1} = \underbrace{\underbrace{y_2 + y_6}_{s=2}}_{A_2} = \underbrace{\underbrace{y_3 + y_7}_{s=2}}_{A_3} = \underbrace{\underbrace{y_4 + y_8}_{s=2}}_{A_4} \quad (4)$$

Using the commutativity property of addition and the equalities, we can have different permutations of the variables which still satisfy the equations. Thus,  $y_1 + y_5$  is the same as  $y_5 + y_1$  and  $A_1 = A_2$  is the same as  $A_2 = A_1$ . But instead of permuting the actual variables, the channel symbols in each variable is permuted.

Let  $\text{swap}(y_a, y_b)$  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 (for clarity we use the channel symbols instead).

$$\begin{array}{l} \text{SN sequence} \\ \text{swap}(y_1, y_5) \\ \text{swap}(y_2, y_6) \\ \text{swap}(y_1, y_2)(y_5, y_6) \end{array} \quad \begin{array}{l} \underbrace{y_1 + y_5}_{A_1} = \underbrace{y_2 + y_6}_{A_2} \\ -7 + +7 = -5 + +5 \\ +7 + -7 = -5 + +5 \\ -7 + +7 = +5 + -5 \\ -5 + +5 = -7 + +7 \end{array}$$

The last swap is equivalent to “ $\text{swap}(A_1, A_2)$ ”. Any of these possible swaps can then be used in combination with another.

Consider an  $\mathcal{N}_4(8, 8, 0)$  mapping with spectral nulls at frequencies 0, 1/4, 1/2, 3/4 and 1 obtained from the

algorithm below.

```

Mapping algorithm for  $\mathcal{N}_4(8, 8, 0)$ 
Input:  $(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8)$ 
Output:  $(y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8)$ 
 $(y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8) \leftarrow (1, 2, 3, 4, 8, 7, 6, 5)$ 
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.

```

The power spectral density is shown in Fig. 4.  $\square$

**Example 6** For  $M = 8$ , with  $k = 2$  and  $s = 4$ , we have

$$\underbrace{\underbrace{y_1 + y_3 + y_5 + y_7}_{A_1}}_{k=2} = \underbrace{\underbrace{y_2 + y_4 + y_6 + y_8}_{A_2}}_{k=2} \quad (5)$$

Consider an  $\mathcal{N}_2(8, 9, -1)$  mapping with spectral nulls at frequencies 0, 1/4, 1/2, 3/4 and 1 obtained from the algorithm below.

```

Mapping algorithm for  $\mathcal{N}_2(8, 9, -1)$ 
Input:  $(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9)$ 
Output:  $(y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8)$ 
 $(y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8) \leftarrow (1, 2, 4, 3, 5, 6, 8, 7)$ 
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( $y_1, y_2$ )( $y_3, y_4$ )...
  ...( $y_5, y_6$ )( $y_7, y_8$ )
end.

```

The power spectral density is shown in Fig. 5.  $\square$

Examples 5 and 6 show that for the same  $M$  value, we have different numbers of inputs and this related to the values of  $k$  and  $s$ .

## 5. Quaternary Line Codes

We present in this section a comparison study between quaternary line codes, which are designed by our DPMs technique, and others published in the literature [7]. We study their performances from an error propagation perspective and the differences between their spectra.

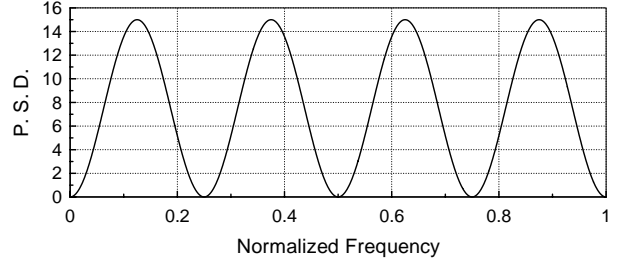


Fig. 4. Power spectral density of  $\mathcal{N}_4(8, 8, 0)$

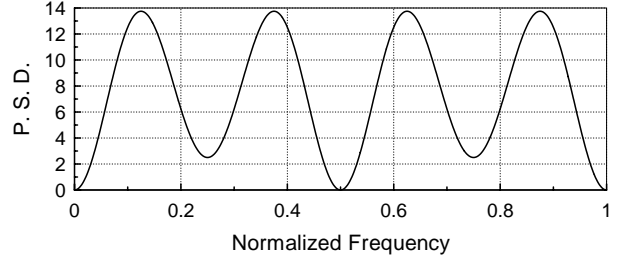


Fig. 5. Power spectral density of  $\mathcal{N}_2(8, 9, -1)$

### 5.1. 1Binary 4Quaternary: 1B4Q

This class of codes have for each one input bit, four quaternary output symbols. Using the DPMs technique, it is clear that the number of input bits will determine the cardinality of the quaternary line code. Thus the number of input bits is actually the outputs of the convolutional base code, and this is determined by its rate, we have chosen in this case a rate of  $R = 1/2$ .

We consider the case of  $M = 4$  with  $k = 2$ , and  $s = 2$ . The channel symbols must satisfy

$$\underbrace{\underbrace{y_1 + y_3}_{A_1}}_{k=2} = \underbrace{\underbrace{y_2 + y_4}_{A_2}}_{k=2} \quad (6)$$

We can see that from (6) we can assign two input bits to swap  $y_i$  in each equation. Thus we can create an algorithm for  $\mathcal{N}_2(4, 2, 2)$  as described below.

```

Mapping algorithm for  $\mathcal{N}_2(4, 2, 2)$ 
Input:  $(x_1, x_2)$ 
Output:  $(y_1, y_2, y_3, y_4)$ 
 $(y_1, y_2, y_3, y_4) \leftarrow (1, 2, 4, 3)$ 
begin
  if  $x_1 = 1$  then swap( $y_1, y_3$ )
  if  $x_2 = 1$  then swap( $y_2, y_4$ )
end.

```

The mapping will result in the following codebook

$$\begin{Bmatrix} 00 \\ 01 \\ 10 \\ 11 \end{Bmatrix} \rightarrow \begin{Bmatrix} 1243 \\ 1342 \\ 4213 \\ 4312 \end{Bmatrix} \rightarrow \begin{Bmatrix} -3 - 1 + 3 + 1 \\ -3 + 1 + 3 - 1 \\ +3 - 1 - 3 + 1 \\ +3 + 1 - 3 - 1 \end{Bmatrix}.$$

As was introduced previously, the finite state machine of the resultant line code will have the same finite state machine of the base code, which in this case is the convolutional code with  $K = 3$  and  $R = 1/2$  as depicted in Fig. 6.

## 5.2. 2Binary 4Quaternary: 2B4Q

Using a convolutional code with rate  $R = 2/3$  as a base code, we can have  $2^3 = 8$  outputs, which are mapped into 8 codewords of the quaternary sequences.

The following algorithm illustrate the three distinct swaps, based on the swaps of the  $y_i$  and  $A_i$ , and therefore we create an algorithm for  $\mathcal{N}_2(4, 3, 1)$  as depicted below.

```

Mapping algorithm for  $\mathcal{N}_2(4, 3, 1)$ 
Input:  $(x_1, x_2, x_3)$ 
Output:  $(y_1, y_2, y_3, y_4)$ 
 $(y_1, y_2, y_3, y_4) \leftarrow (1, 2, 4, 3)$ 
begin
  if  $x_1 = 1$  then swap( $y_1, y_3$ )
  if  $x_2 = 1$  then swap( $y_2, y_4$ )
  if  $x_3 = 1$  then swap( $y_1, y_2$ )( $y_3, y_4$ )
end.

```

This results in the mapping

$$\left\{ \begin{array}{l} 000 \\ 001 \\ 010 \\ 011 \\ 100 \\ 101 \\ 110 \\ 111 \end{array} \right\} \rightarrow \left\{ \begin{array}{l} 1243 \\ 4213 \\ 1342 \\ 4312 \\ 2134 \\ 2431 \\ 3124 \\ 3421 \end{array} \right\} \rightarrow \left\{ \begin{array}{l} -3 - 1 + 3 + 1 \\ +3 - 1 - 3 + 1 \\ -3 + 1 + 3 - 1 \\ +3 + 1 - 3 - 1 \\ -1 - 3 + 1 + 3 \\ -1 + 3 + 1 - 3 \\ +1 - 3 - 1 + 3 \\ +1 + 3 - 1 - 3 \end{array} \right\}.$$

After mapping the spectral nulls symbols to the channel symbols, it can be verified that each sequence satisfies the SN property. The distance matrices will also verify that it is indeed an increasing mapping with  $\delta = 1$ . The power spectral density of the 2B4Q code is depicted in Fig. 7.

We use a convolutional code with rate  $R = 2/3$ , and number of its states is 16. In view of the space limitation

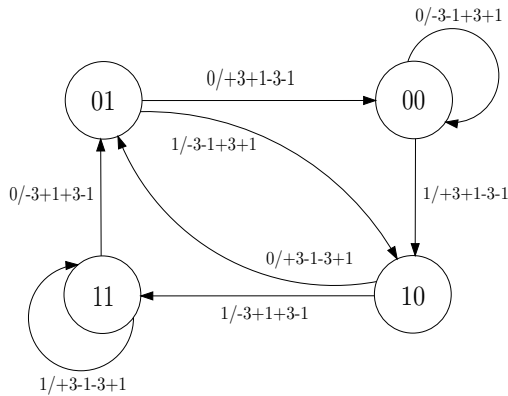


Fig. 6. Finite State Machine of 1B4Q Line Code

of the paper, we just present the encoder of the 2B4Q line code in Table II.

We compare our line code to others published in the literature [7]. We can see from Fig. 8 that our designed line codes have the advantage of having nulls at rational submultiples of the symbol frequency and this is advantageous for other applications [6].

## 5.3. Error Propagation of Quaternary Line Codes

As is known, the Viterbi algorithm is based on the calculation of the metrics in a trellis diagram for a convolutional code and this needs a finite state machine presentation. Previous investigations on the modeling of the line code to have state presentation and then be able to implement the Viterbi algorithm to get benefit of the maximum-likelihood that Viterbi offer.

The error propagation, which is the study of a single error on the performance of the Viterbi decoder, shows how perfect the Viterbi decoder is.

Fig. 9 shows that our designed codes have zero error propagation and this is expected in view of the DPMs properties. Other quaternary line codes [7] show signs of error propagation. The 3B2Q line code has three states and this causes problems for the Viterbi decoder to calculate the metrics properly, this is why this code has the highest error propagation. Also both 1B1Q and 3B2Q have more error propagation than our codes and this is because of the structure of their trellises, which do not have all zeros path.

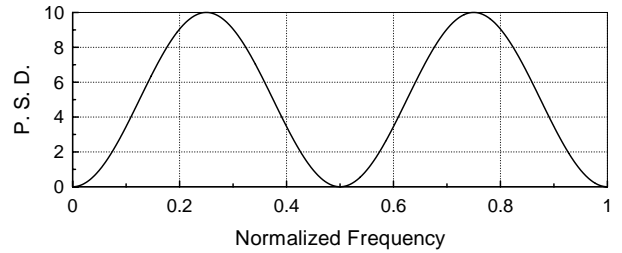


Fig. 7. Power spectral density of  $\mathcal{N}_2(4, 3, 1)$

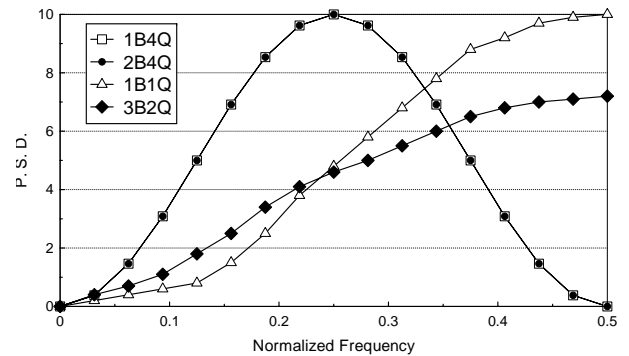


Fig. 8. Power spectral density of certain Quaternary Line Codes

TABLE II  
ENCODER OF THE 2B4Q LINE CODE

State	Inputs								
	00		01		10		11		
	Outputs	Next State	Outputs	Next State	Outputs	Next State	Outputs	Next State	
A	0000	-3-1+3+1	A	+1+3-1-3	I	+1+3-1-3	C	-3-1+3+1	K
B	0001	-1+3+1-3	A	-3+1+3-1	I	-3+1+3-1	C	-1+3+1-3	K
C	0010	+3+1-3-1	B	-1-3+1+3	J	-1-3+1+3	D	+3+1-3-1	L
D	0011	+1-3-1+3	B	+3-1-3+1	J	+3-1-3+1	D	+1-3-1+3	L
E	0100	-1+3+1-3	A	-3+1+3-1	I	-3+1+3-1	C	-1+3+1-3	K
F	0101	-3-1+3+1	A	+1+3-1-3	I	+1+3-1-3	C	-3-1+3+1	K
G	0110	+1-3-1+3	B	+3-1-3+1	J	+3-1-3+1	D	+1-3-1+3	L
H	0111	+3+1-3-1	B	-1-3+1+3	J	-1-3+1+3	D	+3+1-3-1	L
I	1000	+3+1-3-1	E	-1-3+1+3	M	-1-3+1+3	G	+3+1-3-1	O
J	1001	+1-3-1+3	E	+3-1-3+1	M	+3-1-3+1	G	+1-3-1+3	O
K	1010	-3-1+3+1	F	+1+3-1-3	N	+1+3-1-3	H	-3-1+3+1	P
L	1011	-1+3+1-3	F	-3+1+3-1	N	-3+1+3-1	H	-1+3+1-3	P
M	1100	+1-3-1+3	E	+3-1-3+1	M	+3-1-3+1	G	+1-3-1+3	O
N	1101	+3+1-3-1	E	-1-3+1+3	M	-1-3+1+3	G	+3+1-3-1	O
O	1110	-1+3+1-3	F	-3+1+3-1	N	-3+1+3-1	H	-1+3+1-3	P
P	1111	-3-1+3+1	F	+1+3-1-3	N	+1+3-1-3	H	-3-1+3+1	P

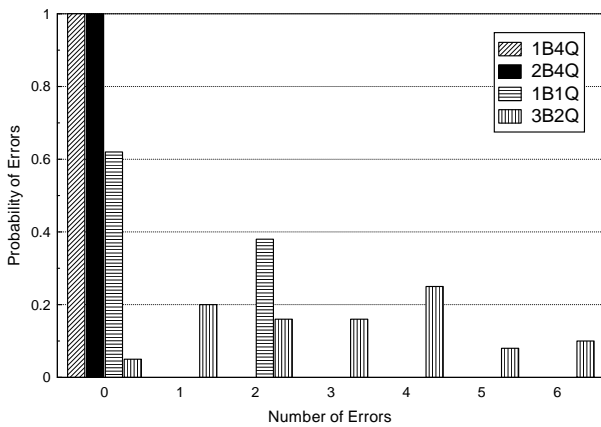


Fig. 9. Error Propagation of certain Quaternary Line Codes

## 6. Conclusion

We have introduced a new construction of multilevel line codes based on the technique of distance-preserving mappings. These codes will have the same properties as convolutional codes and thus the use of the Viterbi algorithm will be very useful as a solution for decoding.

Besides this advantage, we designed DC-free line codes with nulls at rational submultiples of the symbol frequency. To our knowledge, this is the first design to do so.

The fact that our line codes have the structure of trellis codes, we have the benefit of no error propagation.

In the future we will investigate the use of Euclidean distances instead of Hamming distances and determine the performance of these multilevel line codes in such cases.

## References

- [1] 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.
- [2] 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.
- [3] J.-C. Chang, R.-J. Chen, T. Kløve and S.-C. Tsai, "Distance-preserving mappings from binary vectors to permutations," *IEEE Trans. Inf. Theory*, vol. 49, no. 4, pp. 1054–1059, Apr. 2003.
- [4] J.-C. Chang, "Distance-increasing mappings from binary vectors to permutations," *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 359–363, Jan. 2005.
- [5] 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.
- [6] K. Ouahada, H. C. Ferreira, A. J. H. Vinck and W. A. Clarke, "Combined higher order spectral nulls codes and OFDM modulation for power line communications," in *Proc. Int. Symp. on Power Line Commun. and its Applications*, Orlando, FL, Mar. 26–29, 2006, pp. 122–127.
- [7] 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.
- [8] J. K. Pollard, "Multilevel data communication over optical fiber," in *IEE Proc. I Commun. Speech Vis.*, 1991, vol. 138, no. 3, pp. 162–168.
- [9] E. Gorog, "Alphabets with desirable frequency spectrum properties," *IBM J. Res. Develop.*, vol. 12, pp. 234–241, May 1968.
- [10] K. A. S. Immink, *Codes for mass data storage systems*, Shannon Foundation Publishers, The Netherlands, 1999.
- [11] 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.