A Multilevel Construction for Mappings from Binary Sequences to Permutation Sequences

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

***Abstract*— A multilevel construction is introduced to create distance-preserving mappings from binary sequences to permutation sequences. It is also shown that for certain values, the new mappings attain the upper bound on the sum of Hamming distances obtainable for such mappings, and in the other cases improve on those of previous mappings.**

I. Introduction

Vinck [1] renewed the interest in permutation codes when he suggested it for coding on a power-line communications system. Ferreira *et al* [2], [3] then used the idea of distance-preserving mappings (DPMs) [4] in creating mappings of binary sequences to permutation sequences to construct permutation trellis codes, while preserving the distance amongst the codewords. They showed how new mappings could be constructed by making use of a prefix method. Chang *et al* [5] extended this further by presenting several constructions for creating distance-preserving mappings. More recent constructions have been proposed by Lee [6] and Chang [7]. Swart, de Beer and Ferreira [8] presented an upper bound on the sum of the Hamming distances in such mappings, using simulation results to show that mappings attaining the upper bound does indeed perform better than mappings that do not. Wadayama and Vinck [9] presented a multilevel construction for permutation codes, combining constant weight binary codes to form permutation codes. Our construction in this paper is based on this idea.

II. Preliminaries

First, a brief overview of related definitions and a description of DPMs to permutation sequences will be given.

Definition 1 A binary code $\mathcal{C}_b$ consists of $|\mathcal{C}_b|$ sequences of length n, where every sequence contains 0s and 1s as symbols. □

Definition 2 A permutation code $\mathcal{C}_p$ consists of $|\mathcal{C}_p|$ sequences of length M, where every sequence contains the M different integers $1, 2, \ldots, M$ as symbols. □

Definition 3 The symmetric group, S_M, consists of the sequences obtained from permuting the symbols $1, 2, \ldots, M$ in all the possible ways, with $|S_M| = M!$. □

For our mappings $\mathcal{C}_b$ will consist of all the possible binary sequences of length n with $|\mathcal{C}_b| = 2^n$ and $\mathcal{C}_p$ will consist of some subset of S_M with $|\mathcal{C}_p| = |\mathcal{C}_b|$. In addition, the distances between sequences for one set is preserved amongst the sequences of the other set.

For binary sequences, let $\mathbf{x}_i$ be the i-th binary sequence in $\mathcal{C}_b$. The Hamming distance $d_H(\mathbf{x}_i, \mathbf{x}_j)$ is defined as usual as the number of positions in which the two sequences differ. Construct a matrix $\mathbf{D}$ whose entries are the distances between binary sequences in $\mathcal{C}_b$, where

$$\mathbf{D} = [d_{ij}] \text{ with } d_{ij} = d_H(\mathbf{x}_i, \mathbf{x}_j). \quad (1)$$

Similarly for permutation sequences, let $\mathbf{y}_i$ be the i-th permutation sequence in $\mathcal{C}_p$. The Hamming distance $d_H(\mathbf{y}_i, \mathbf{y}_j)$ is also defined as the number of positions in which the two sequences differ. Construct a matrix $\mathbf{E}$ whose entries are the distances between permutation sequences in $\mathcal{C}_p$, where

$$\mathbf{E} = [e_{ij}] \text{ with } e_{ij} = d_H(\mathbf{y}_i, \mathbf{y}_j). \quad (2)$$

Example 1 The following is a possible mapping of $n = 2 \to M = 3$ (for subsequent mappings the binary sequences, which will follow the usual lexicography, will be omitted)

$$\{00, 01, 10, 11\} \to \{123, 132, 321, 312\}.$$

Using (1) and (2), we have for the above mapping

$$\mathbf{D} = \begin{bmatrix} 0 & 1 & 1 & 2 \\ 1 & 0 & 2 & 1 \\ 1 & 2 & 0 & 1 \\ 2 & 1 & 1 & 0 \end{bmatrix} \text{ and } \mathbf{E} = \begin{bmatrix} 0 & 2 & 2 & 3 \\ 2 & 0 & 3 & 2 \\ 2 & 3 & 0 & 2 \\ 3 & 2 & 2 & 0 \end{bmatrix}.$$

In this case all entries had an increase in distance (except the main diagonal where there is always zero distance). □

Previously, the three different mapping types were defined in terms of trellis codes [3], but since only the mappings themselves are considered, we will redefine it in terms of distances between the binary and permutation sequences only.

- *Distance-conserving mapping* (DCM): guarantees conservation of the binary sequences' Hamming distance, such that $e_{ij} \geq d_{ij}$, for all $i \neq j$.
- *Distance-increasing mapping* (DIM): guarantees that the permutation sequences' distance will always have some

increase above the binary sequences' distance, such that $e_{ij} \geq d_{ij} + \delta$, $\delta \in \{1, 2, \ldots\}$ for all $i \neq j$.

- *Distance-reducing mapping* (DRM): the permutation sequences' distance has a distance loss which is guaranteed to be not more than a fixed amount compared to the binary sequences' distance, such that $e_{ij} \geq d_{ij} + \delta$, $\delta \in \{-1, -2, \ldots\}$ for all $i \neq j$.

In general δ defines the type of DPM, with $\delta = 0$ indicating a DCM, $\delta > 0$ indicating a DIM and $\delta < 0$ indicating a DRM. We now introduce the notation $\mathcal{M}(n, M, \delta)$ to indicate DPMs from n-binary sequences to M-permutation sequences with δ indicating the lower bound on the distance change and the mapping type. The mapping in Example 1 would thus be a $\mathcal{M}(2, 3, 1)$ mapping.

III. Binary Multilevel Representation of a Permutation

Any permutation can be written using a binary multilevel representation, as in

$$0123 \rightarrow \begin{Bmatrix} 0101 \\ 0011 \end{Bmatrix} \text{ or } 34201 \rightarrow \begin{Bmatrix} 10001 \\ 10100 \\ 01000 \end{Bmatrix},$$

where each symbol's binary value is used as a column. Note that for convenience the symbols $0, 1, \ldots, M-1$ are used instead of $1, 2, \ldots, M$. The rows then form the different levels that will be used. For length M sequences we will need $L = \lceil \log_2 M \rceil$ levels to represent the sequence.

To obtain different permutations, we swap (or transpose) the columns according to certain rules for each level, starting at level 1 and working down to level L. When considering the k-th level, two columns, say a and b, can only be swapped if there are different symbols (i.e. 0 and 1) in positions a and b on level k and all the symbols on the levels $k+1, k+2, \ldots, L$ in columns a and b are the same. This will be further illustrated in the next example. For brevity, we will use $\texttt{swap}(a, b)$ to indicate swapping of columns a and b.

Example 2 Start with the $M = 5$ identity element, which is

$$01234 \rightarrow \begin{Bmatrix} 01010 \\ 00110 \\ 00001 \end{Bmatrix}.$$

For the first level, we can have no swaps, $\texttt{swap}(1, 2)$, $\texttt{swap}(3, 4)$ and $\texttt{swap}(1, 2)(3, 4)$, resulting in $\{01010, 10010, 01100, 10100\}$.

For the second level, we can have no swaps, $\texttt{swap}(1, 3)$, $\texttt{swap}(2, 4)$, $\texttt{swap}(1, 4)$, $\texttt{swap}(2, 3)$ and $\texttt{swap}(1, 3)(2, 4)$, resulting in $\{00110, 10010, 01100, 10100, 01010, 11000\}$.

For the third level, we can have no swaps, $\texttt{swap}(4, 5)$, $\texttt{swap}(3, 5)$, $\texttt{swap}(2, 5)$ and $\texttt{swap}(1, 5)$, resulting in $\{00001, 00010, 00100, 01000, 10000\}$.

The *multilevel permutations*, P_k, are used to represent the set of possibilities on the k-th level. In this case we have

$$\begin{aligned} P_1 &= \{01010, 10010, 01100, 10100\} \\ P_2 &= \{00110, 10010, 01100, 10100, 01010, 11000\} \quad (3) \\ P_3 &= \{00001, 00010, 00100, 01000, 10000\}. \end{aligned}$$

□

Since a subset from all the possible permutation sequences must be chosen to do a mapping, we must know whether using the multilevel representation will yield all the sequences in S_M.

Proposition 1 *If the multilevel permutations, P_k, $1 \leq k \leq L$ are used for the multilevel construction, then for M it generates all the permutation sequences from the symmetric group, S_M, with $|S_M| = M!$.* □

Briefly, by enumerating the number of possibilities for each level and using induction, it can be proved that $|P_1| \times |P_2| \times \cdots \times |P_L| = M!$.

Returning to Example 2, from (3) it is clear that $|P_1| = 4$, $|P_2| = 6$ and $|P_3| = 5$, and when multiplied together is equal to $|S_5| = 5! = 120$.

IV. Multilevel Construction

The construction by Wadayama and Vinck [9] was limited to the case where the input sequences was of length $n = 2^m$, with m any positive integer, and consisted of binary constant weight codes. One can regard this as a mapping from binary constant weight sequences to permutation sequences. Our construction will be mapping all binary sequences of length n to permutation sequences, as well as being valid for any value of n.

From the multilevel permutations in the previous section, a suitable subset is chosen to create the *multilevel components*, C_k, for the k-th level, where $C_k \subseteq P_k$.

The idea of the multilevel construction is then to first create a distance-preserving mapping from the binary subsequences, $n_1, n_2, \ldots, n_L$, to the binary sequences of the multilevel components, which can then be transformed into permutation sequences. This is illustrated in Fig. 1, where L is the number of levels, and for $1 \leq k \leq L$, n_k is the number of input bits assigned to level k, with $n_1 + n_2 + \cdots + n_L = n$ and C_k are the multilevel components for level k.

Although any combination of swaps can be used on a level, as was seen in Example 2, matters are simplified by only using sequences that was obtained by combinations of the following

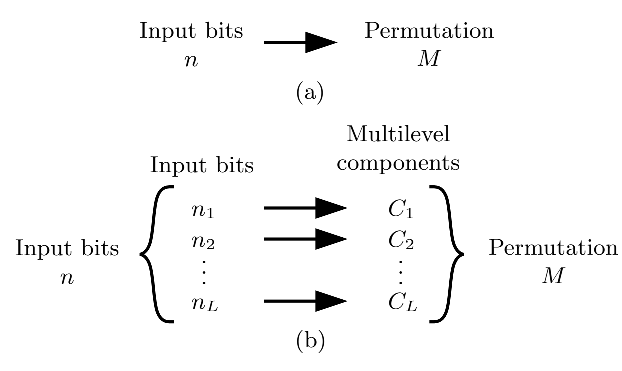

Fig. 1. Comparison of (a) mapping using standard lexicography and (b) mapping using the multilevel construction

swaps

$$\begin{aligned}
&\text{Level 1: } \texttt{swap}(1,2)(3,4)(5,6)(7,8)\cdots \\
&\text{Level 2: } \texttt{swap}(1,3)(2,4)(5,7)(6,8)\cdots \\
&\text{Level 3: } \texttt{swap}(1,5)(2,6)(3,7)(4,8)\cdots \\
&\text{Level 4: } \texttt{swap}(1,9)(2,10)(3,11)(4,12)\cdots \\
&\qquad\vdots
\end{aligned} \tag{4}$$

In general, the $(i2^{k-1}+j)$-th swap on the k-th level, where $1 \leq j \leq 2^{k-1}$, $i \geq 0$, is given by $\texttt{swap}(i2^k+j, i2^k+j+2^{k-1})$.

Example 3 To create an $\mathcal{M}(5,5,0)$ mapping, multilevel components are chosen from the multilevel permutations in (3). These should be chosen in such a way that distance is preserved between the input subsequences and the component sequences. However, the distance between the component sequences should be equal to or larger than twice the distance between the corresponding input subsequences. This will be further clarified in Proposition 2.

We choose $n_1 = 2$, $n_2 = 2$, $n_3 = 1$ and

$$\begin{aligned}
C_1 &= \{01010, 10010, 01100, 10100\}, \\
C_2 &= \{00110, 10010, 01100, 11000\}, \\
C_3 &= \{00001, 10000\}.
\end{aligned} \tag{5}$$

Let $\mathbf{D}_k$ be the distance matrix between the input subsequences on level k and let $\mathbf{E}_k$ be the distance matrix between the components on level k. Then,

$$\mathbf{D}_1 = \begin{bmatrix} 0 & 1 & 1 & 2 \\ 1 & 0 & 2 & 1 \\ 1 & 2 & 0 & 1 \\ 2 & 1 & 1 & 0 \end{bmatrix} \qquad \mathbf{E}_1 = \begin{bmatrix} 0 & 2 & 2 & 4 \\ 2 & 0 & 4 & 2 \\ 2 & 4 & 0 & 2 \\ 4 & 2 & 2 & 0 \end{bmatrix}$$

$$\mathbf{D}_2 = \begin{bmatrix} 0 & 1 & 1 & 2 \\ 1 & 0 & 2 & 1 \\ 1 & 2 & 0 & 1 \\ 2 & 1 & 1 & 0 \end{bmatrix} \qquad \mathbf{E}_2 = \begin{bmatrix} 0 & 2 & 2 & 4 \\ 2 & 0 & 4 & 2 \\ 2 & 4 & 0 & 2 \\ 4 & 2 & 2 & 0 \end{bmatrix}$$

$$\mathbf{D}_3 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad \mathbf{E}_3 = \begin{bmatrix} 0 & 2 \\ 2 & 0 \end{bmatrix},$$

producing the required distance-preserving mappings. □

With the multilevel components, we can now map binary sequences to permutation sequences. The binary input sequence is broken into L subsequences of lengths $n_1, n_2, \ldots, n_L$. Each subsequence is mapped to a multilevel component, and the corresponding L multilevel components are used to determine the required permutation sequence. We start with the identity element and swap columns (using the swaps set out in (4)) until the first level is the same as the chosen multilevel component. Then proceed to the next level and repeat, until all the levels have been done, then convert each column to a decimal number to obtain the permutation sequence. The next example illustrates this procedure, as well as showing a more elegant method to present the mapping.

Example 4 Consider the binary input 10111 which is broken into subsequences 10, 11 and 1. From (5) the corresponding multilevel components are 01100, 11000 and 10000.

Start with the identity element for $M = 5$ in the multilevel representation,

$$\begin{matrix} 01010 \\ 00110. \\ 00001 \end{matrix}$$

For the first level, we require $\texttt{swap}(3,4)$, for the second level $\texttt{swap}(1,3)(2,4)$ and for the third level $\texttt{swap}(1,5)$. Thus, the identity element is transformed as

$$\begin{matrix} 01010 \\ 00110 \\ 00001 \end{matrix} \rightarrow \begin{matrix} 01100 \\ 00110 \\ 00001 \end{matrix} \rightarrow \begin{matrix} 10010 \\ 11000 \\ 00001 \end{matrix} \rightarrow \begin{matrix} 00011 \\ 01001, \\ 10000 \end{matrix}$$

remembering that the entire column is swapped, otherwise it will not be a valid permutation. This $\mathcal{M}(5,5,0)$ mapping maps 10111 to 42013.

Since the mapping only makes use of swaps or combinations of swaps in (4), it is easy to set up an algorithm, where $x_1, \ldots, x_5$ represents the binary sequence and $y_1, \ldots, y_5$ represents the permutation sequence, as follows

```
Input: (x1, x2, x3, x4, x5)
Output: (y1, y2, y3, y4, y5)
begin
   (y1, y2, y3, y4, y5) ← (0, 1, 2, 3, 4)
   if x2 = 1 then swap(y1, y2)
   if x1 = 1 then swap(y3, y4)
   if x4 = 1 then swap(y1, y3)
   if x3 = 1 then swap(y2, y4)
   if x5 = 1 then swap(y1, y5)
end.
```

Using 10111 as input again, we obtain

$$\begin{array}{lclcl}
 & & & & 01234 \\
x_2 = 0 & \rightarrow & - & \rightarrow & 01234 \\
x_1 = 1 & \rightarrow & \texttt{swap}(y_3, y_4) & \rightarrow & 01324 \\
x_4 = 1 & \rightarrow & \texttt{swap}(y_1, y_3) & \rightarrow & 31024 \\
x_3 = 1 & \rightarrow & \texttt{swap}(y_2, y_4) & \rightarrow & 32014 \\
x_5 = 1 & \rightarrow & \texttt{swap}(y_1, y_5) & \rightarrow & 42013,
\end{array}$$

resulting in the same permutation sequence. □

Let $\mathbf{x}$ be the binary sequence of length n and break it up into subsequences of input bits for each level, with the i-th input subsequence of length n_k on level k being denoted by $\mathbf{x}_{i,k}$. Assign $c_{i,k}$ to denote the i-th multilevel component for level k, such that $C_k = \{c_{0,k}, c_{1,k}, \ldots, c_{2^{n_k}-1,k}\}$. Thus, we have a mapping of $\mathbf{x}_{i,k} \rightarrow c_{i,k}$ on level k.

Proposition 2 *For $1 \leq k \leq L$, choose a subset of the multilevel permutations, P_k, to form the multilevel components, C_k, such that $C_k \subseteq P_k$, and map input bits of length n_k to each possibility in C_k, such that $|C_k| = 2^{n_k}$ with $n_1 + n_2 + \cdots + n_L = n$. A DPM from binary sequences to permutation sequences is obtained if*

1) $d_H(c_{i,k}, c_{j,k}) \geq 2d_H(\mathbf{x}_{i,k}, \mathbf{x}_{j,k})$, *for DRMs and DCMs,*
2) $d_H(c_{i,k}, c_{j,k}) \geq 2d_H(\mathbf{x}_{i,k}, \mathbf{x}_{j,k}) + \delta - 1$, *for DIMs,*

for all $i \neq j$. □

The following example is to illustrate how the input bits can be assigned differently.

Example 5 Consider the following three $\mathcal{M}(6,6,0)$ mappings

$$\mathcal{M}_1(6,6,0): \begin{array}{l} C_1 = \left\{\begin{array}{l}010101, 100101, 011001, 101001, \\ 010110, 100110, 011010, 101010\end{array}\right\}, \\ C_2 = \{001100, 100100, 011000, 110000\}, \\ C_3 = \{000011, 110000\}, \end{array}$$

$$\mathcal{M}_2(6,6,0): \begin{array}{l} C_1 = \{010101, 100101, 011010, 101010\}, \\ C_2 = \{001100, 100100, 011000, 110000\}, \\ C_3 = \{000011, 100001, 010010, 110000\}, \end{array}$$

$$\mathcal{M}_3(6,6,0): \begin{array}{l} C_1 = \left\{\begin{array}{l}010101, 100101, 011001, 101001, \\ 010110, 100110, 011010, 101010\end{array}\right\}, \\ C_2 = \{001100, 110000\}, \\ C_3 = \{000011, 100001, 010010, 110000\}. \end{array}$$

The input bits were assigned as $n_1 = 3$, $n_2 = 2$ and $n_3 = 1$, $n_1 = 2$, $n_2 = 2$ and $n_3 = 2$, and $n_1 = 3$, $n_2 = 1$ and $n_3 = 2$ respectively. To illustrate the difference between the mappings, the $\mathbf{E}$ distance matrices are visualized in Fig. 2. The figure clearly shows that the distances are distributed differently in each mapping. Despite the differences, in each case the mapping still satisfies the property of distance-preserving.

For all three mappings $|\mathbf{E}| = 19456$ and $|\mathbf{E}_{\max}| = 20472$ (refer to the next section for the definitions of $|\mathbf{E}|$ and $|\mathbf{E}_{\max}|$). □

V. Upper Bound on Distance

In [8], an upper bound was presented on the sum of the Hamming distances that can be attained in a permutation mapping. The sum of the Hamming distances in the $\mathbf{E}$ distance matrix is

$$|\mathbf{E}| = \sum_{i=1}^{2^n} \sum_{j=1}^{2^n} e_{ij}. \tag{6}$$

The upper bound is denoted by $|\mathbf{E}_{\max}|$, with

$$|\mathbf{E}_{\max}| = M[2^{2n} - (2\alpha\beta + \beta + \alpha^2 M)], \tag{7}$$

where $\alpha = \lfloor 2^n/M \rfloor$ and $\beta = 2^n \mod M$, with $\lfloor . \rfloor$ producing the integer part after division and *mod* producing the remainder after division.

Proposition 3 *Any multilevel DPM with $M = 2^l$ and l any positive integer will attain the upper bound $|\mathbf{E}_{\max}|$, provided the maximum distances are achieved between all the multilevel components.* □

The maximum distances between multilevel components are only achieved when each component's complement is also a component, and this is only possible for values of $M = 2^l$ and l any positive integer. The following example illustrates this.

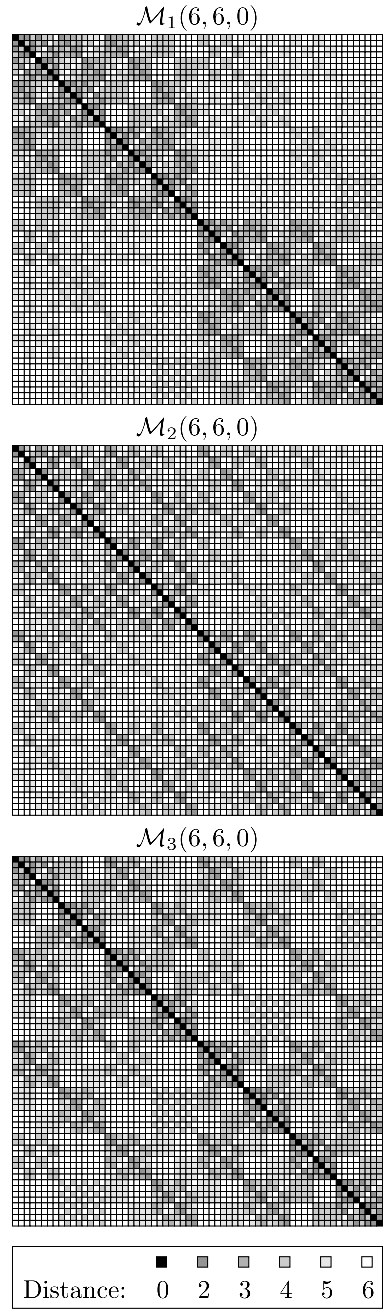

Fig. 2. Visualization of $\mathbf{E}$ distance matrices for different $\mathcal{M}(6,6,0)$ mappings

Example 6 Consider an $\mathcal{M}(8,8,0)$ mapping with

$$C_1 = \left\{\begin{array}{l} 01010101, 01010110, 01011001, 01011010, \\ 01100101, 01100110, 01101001, 01101010, \\ 10010101, 10010110, 10011001, 10011010, \\ 10100101, 10100110, 10101001, 10101010 \end{array}\right\},$$

$$C_2 = \left\{\begin{array}{l} 00110011, 00110110, 01100011, 01100110, \\ 10011001, 10011100, 11001001, 11001100 \end{array}\right\}.$$

It is easy to verify that the complement of each component in

TABLE I
COMPARISON OF DISTANCES FOR VARIOUS MAPPINGS

M	$\lvert\mathbf{E}_{\max}\rvert$	Prefix [2]	Construction 2 [5]	Construction 3 [5]	Construction [6]	Construction 2 [7]	Multilevel construction
4	768	732	768	768	–	768	768
5	4090	3616	3712	–	3872	4020	3712
6	20472	17072	17536	18432	–	18432	19456
7	98294	78528	81024	–	91016	88064	94208
8	458752	355840	367744	393216	–	413312	458752
9	2097144	–	1645696	–	1911000	1802240	1982464
10	9437160	–	7281792	7864320	–	8110080	9043968
11	41943022	–	31923328	–	37741432	36330496	40108032
12	184549344	–	138878080	150994944	–	154927104	180355072

C_1 and C_2 is also in C_1 and C_2.

Any of the following C_3 components generates a DCM when combined with the above C_1 and C_2. Alongside each C_3 we also list the respective $|\mathbf{E}|$ values,

$$\begin{aligned} C_3 &= \{00001111, 10000111\}, \quad |\mathbf{E}| = 409600, \\ C_3 &= \{00001111, 11000011\}, \quad |\mathbf{E}| = 425984, \\ C_3 &= \{00001111, 11100001\}, \quad |\mathbf{E}| = 442368, \\ C_3 &= \{00001111, 11110000\}, \quad |\mathbf{E}| = 458752. \end{aligned}$$

One can see that as the distance between the C_3 components increases, the distance of the entire mapping also increases. For this case $|\mathbf{E}_{\max}| = 458752$ and this is attained by the last C_3, the only one that contains the components' complements as well. □

In Table I we compare the $|\mathbf{E}|$ values of previous DCMs with those of our new mappings. For both $M = 4$ and $M = 8$ our new mappings attain the upper bound, as the proposition states. Even though our new mappings for other values of M do not attain the upper bound, they are still an improvement over the previous known mappings. Our new DIMs and DRMs also attain the upper bound when $M = 2^l$ and l is any positive integer.

VI. CONCLUSION

We introduced a new multilevel construction to create mappings from binary sequences to permutation sequences, while preserving the corresponding distances between the sequences. Although it is not a construction for M in general, it simplifies the process for finding mappings by breaking it into smaller mappings. As example, to map $n = 16$ to $M = 16$ requires one to choose 65536 permutation sequences, while preserving the distance between all those sequences. With this new construction one only needs four $n = 4$ to $M = 16$ mappings, requiring 16 binary sequences to be chosen for each smaller mapping.

Even though the mapping lacks generality, it has great flexibility, as was shown in Example 5. In [3] it was observed that different mappings that attain the upper bound can have differing performance results when used in permutation trellis codes. Using the flexibility one can obtain a mapping that is optimal when combined with a certain trellis code.

To our knowledge, this is also the first construction that can be used for distance-increasing and distance-reducing mappings, in addition to distance-conserving mappings. It is also the first construction to attain the upper bound on the sum of the Hamming distances in such mappings for certain lengths of permutations. (Note that constructions in [7] are not increasing mappings according to the definitions presented here.)

Finally, this construction is also applicable to permutation sequences with repeating symbols, e.g. 123344. The same procedure is used in obtaining the mappings, and several have been constructed so far for various lengths and various repeating symbols.

ACKNOWLEDGMENT

This work was supported in part by the NRF under grant number 2053408. The authors would like to thank the anonymous reviewer for useful comments.

REFERENCES

[1] A. J. H. Vinck, "Coded modulation for powerline communications," *AEÜ International Journal of Electronics and Communications*, vol. 54, no. 1, pp. 45–49, Jan. 2000.

[2] H. C. Ferreira and A. J. H. Vinck, "Interference cancellation with permutation trellis codes," in *Proceedings of the IEEE Vehicular Technology Conference, Fall 2000*, Boston, MA, U.S.A., Sept. 2000, pp. 2401–2407.

[3] H. C. Ferreira, A. J. H. Vinck, T. G. Swart and I. de Beer, "Permutation trellis codes," *IEEE Transactions on Communications*, vol. 53, no. 11, p. 1782–1789, Nov. 2005.

[4] H. C. Ferreira, D. A. Wright and A. L. Nel, "Hamming distance preserving mappings and trellis codes with constrained binary symbols," *IEEE Transactions on Information Theory*, vol. 35, no. 5, pp. 1098–1103, Sept. 1989.

[5] J.-C. Chang, R.-J. Chen, T. Kløve and S.-C. Tsai, "Distance preserving mappings from binary vectors to permutations," *IEEE Transactions on Information Theory*, vol. 49, no. 4, pp. 1054–1059, Apr. 2003.

[6] K. Lee, "New distance-preserving mappings of odd length," *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2539–2543, Oct. 2004.

[7] J.-C. Chang, "Distance-increasing mappings from binary vectors to permutations," *IEEE Transactions on Information Theory*, vol. 51, no. 1, pp. 359–363, Jan. 2005.

[8] T. G. Swart, I. de Beer and H. C. Ferreira, "On the distance optimality of permutation mappings," in *Proceedings of the International Symposium on Information Theory (ISIT 2005)*, Adelaide, Australia, Sept. 4–9, 2005, pp. 1068–1072.

[9] T. Wadayama and A. J. H. Vinck, "A multilevel construction of permutation codes," *IEICE Transactions on Fundamentals*, vol. E84-A, no. 10, pp. 2518–2522, Oct. 2001.