Permutation Trellis Codes

Hendrik C. Ferreira, *Member, IEEE*, A. J. Han Vinck, *Fellow, IEEE*, Theo G. Swart, and Ian de Beer

***Abstract*—We introduce the new concept of permutation trellis codes and present a generalized construction procedure, applying our technique of distance-preserving mappings. Minimum-distance decoding follows naturally, using the Viterbi algorithm. We furthermore investigate the performance of these codes when combined with multitone frequency-shift keying modulation and noncoherent detection in a diversity scheme, to make transmissions robust against narrowband, broadband, and background noise disturbances, such as those encountered in power-line communications.**

***Index Terms*—Channel coding, convolutional codes, frequency-shift keying (FSK), interference suppression.**

I. Introduction

THIS LETTER reports the results of a study on the construction of permutation trellis codes, as well as the performance of these new codes when combined with multitone frequency-shift keying (M-FSK) on channels hampered by severe noise restrictions, such as those on the power-line communications (PLC) channel [1]–[3]. One application foreseen is a very robust low speed (e.g., 2400 b/s) modem functioning within a relatively wideband channel (e.g., within the 150-kHz bandwidth of the CENELEC PLC band [3]).

It is important to note that bandwidth is not the most important limitation here, but rather noise, and also not the widely assumed additive white Gaussian noise (AWGN), but rather an unusual, unpredictable, and widely varying mixture of noise, including additive background noise, impulse noise, and permanent frequency disturbers [3]. The emphasis of our combined coding and modulation scheme is thus on robustness, rather than on data rate or bandwidth use.

A previous study of one of the authors [1] reported on the robustness of permutation codes, specifically block codes, when combined with M-FSK modulation on the PLC channel, and also explained the assumptions and model in detail. Due to length restrictions, we now only summarize the previous material in the second part of this new letter.

In this letter, we specifically emphasize a new concept, namely encoding with permutation trellis codes, and hence, elected to present the corresponding material first. In the second part, we discuss the decoding procedures, returning to PLC channel issues. Finally, we present some performance results.

Paper approved by R. D. Wesel, the Editor for Coding and Communication Theory of the IEEE Communications Society. Manuscript received August 11, 2003; revised November 10, 2004. This work was supported in part by the Deutsche Forschungsgemeinschaft (DFG) and in part by the National Research Foundation (NRF) under Grant 2053408. This paper was presented in part at the IEEE Vehicular Technology Conference, Boston, MA, September 2000, and in part at the IEEE International Symposium on Information Theory, Washington, DC, June 2001.

H. C. Ferreira, T. G. Swart, and I. de Beer are with the Department of Electrical and Electronic Engineering Science, University of Johannesburg, Auckland Park 2006, South Africa (e-mail: hcf@ing.rau.ac.za; ts@ing.rau.ac.za; idb@ing.rau.ac.za).

A. J. H. Vinck is with the Institute for Experimental Mathematics, University Duisburg-Essen, Essen 45326, Germany (e-mail: vinck@iem.uni-due.de).

Digital Object Identifier 10.1109/TCOMM.2005.858683

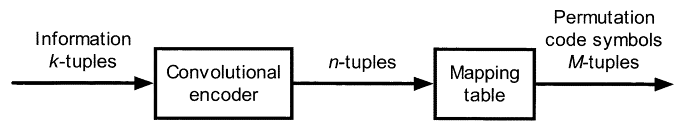

Fig. 1. Encoding process for a distance-preserving permutation trellis code.

To start off, we now briefly present some coding preliminaries and notation, also motivating our study of permutation trellis codes.

The topics of permutation block codes and of permutation arrays have been known and studied for some time, see, e.g., [4]–[7]. Permutation block codes, however, have the disadvantage that the construction of long block codes is a difficult mathematical problem. Furthermore, a general decoding algorithm is not known. We thus introduce permutation trellis codes, which overcome these disadvantages. First, the following definition.

Definition 1: A permutation code C consists of $|C|$ codewords of length M, where every codeword contains the M different integers $1, 2, \ldots, M$ as symbols.

Let $|C|$ denote the cardinality of the code. The cardinality of permutation block codes is upper bounded by [1], [7]

$$|C| \leq \frac{M!}{(d_{\min}-1)!}. \quad (1)$$

Some code constructions for block codes with $d_{\min} > 2$ can be found in, e.g., [5] and [8]. However, no results on permutation trellis codes could be found in the literature, prior to our preliminary results in [9]. Note that we now use the same general code construction and main ideas as in [10]. We here deepen and expand on both the analytical and numerical results in [9], and also present for the first time some performance results. Perhaps the most important result of our work is a generalized construction procedure for new trellis codes capable of achieving various coding rates, constraint lengths, and free distances, and with complexity commensurate with that of binary convolutional codes.

II. Construction of Permutation Trellis Codes: An Example

The trellis-code construction principle, using our distance-preserving mapping technique (also applied in [10]), can be explained by referring to Fig. 1.

The mapping table in Fig. 1 maps the output binary n-tuple code symbols from an $R = k/n$ convolutional code (henceforth called the base code) into integer M-tuples, which in this letter are codewords from a permutation code. The key idea is to find

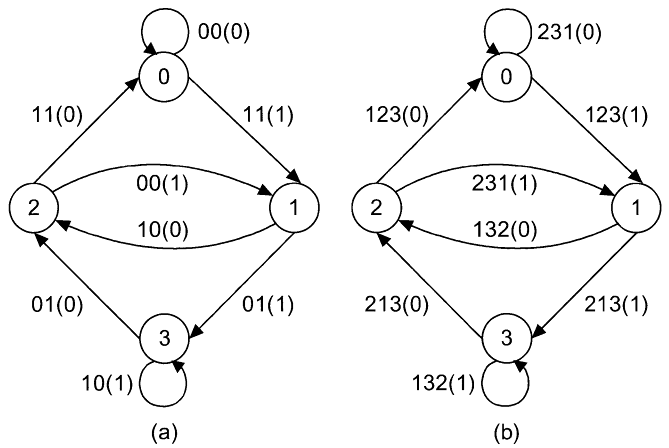

Fig. 2. State systems. (a) Convolutional base code. (b) Permutation trellis code.

an ordered subset of $2^n M$-tuples, out of the full set of permutation M-tuples (with cardinality $M!$), such that the Hamming distance between any two permutation M-tuples is at least as large as the distance between the corresponding convolutional code's output n-tuples which are mapped onto them. This property was previously called distance preserving in [10], since the Hamming distance of the base code is at least conserved, and may sometimes even be increased in the resulting trellis code.

To illustrate this idea, we first present an example. We use the simple, "generic textbook example" four-state, binary $R = 1/2, \nu = 2, d_{\text{free}} = 5$ convolutional code with octal generators 5 and 7 (see, e.g., [11]) as base code. At the output of the encoder, we can map the set of binary 2-tuple code symbols {00, 01, 10, 11} onto a set of permutation M-tuples {231, 213, 132, 123}. The corresponding state systems appear in Fig. 2. Note that, in general, the information transmission rate of the resulting permutation trellis-coded scheme will be k/M bits per channel use.

The property of distance preserving can be verified by setting up the matrices $\mathbf{D} = [d_{ij}]$ and $\mathbf{E} = [e_{ij}]$. We here represent the base code's 2-b binary output code symbols with integers. Briefly, let d_{ij} be the Hamming distance between the binary code symbols i and j, where $0 \leq i, j \leq 2^n - 1$. The key to the code-construction technique is then to find an ordered subset of 2^n permutation M-tuples such that $e_{ij} \geq d_{ij}$, for all $i \neq j$, where e_{ij} is the Hamming distance between the ith and jth M-tuples in the subset.

For our example code in Fig. 2

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

Here, since $e_{ij} = d_{ij} + 1$ for all $i \neq j$, the map of {00, 01, 10, 11} onto {231, 213, 132, 123} guarantees an increase of one unit of distance per step between any two unremerged paths in the trellis diagram of the resulting permutation trellis code, when comparing it with the base code.

For the base code in our example, $d_{\text{free}} = 5$ before the mapping. The shortest remerging paths in the trellis diagram, which for this code is known to determine the free distance [11], have length $\tau = 3$ steps. It is, thus, easy to see that $d'_{\text{free}} = 8$ for the $R' = 1/3$ code obtained after the mapping. This example in fact represents a distance-increasing map.

Note that the same mapping can be applied to any $R = 1/2$ convolutional base code. By using more powerful base codes, we can easily construct more powerful trellis codes, achieving larger free distances.

III. Generalization of Hamming Distance Mappings

Prompted by the distance increase that we observed in our example in Section II, we expand and generalize our previous concept of a distance-preserving mapping in [10].

Definition 2: Distance-Conserving Mappings (DCMs): A DCM only guarantees conservation of the base code's free distance. Hence, for at least one $i \neq j, e_{ij} = d_{ij}$, while for all other $i \neq j, e_{ij} \geq d_{ij}$.

Following the example in the previous section, we can now formally introduce a distance-increasing mapping.

Definition 3: Distance-Increasing Mappings (DIMs): A DIM guarantees that the resulting trellis code's distance will always have some increase above the base code's free distance, for any base code. Hence, $e_{ij} \geq d_{ij} + 1$ for all $i \neq j$.

In order to complete our generalization, we finally introduce a controlled distance-reducing mapping, which yields a trellis code having a distance lower than that of the base code, although the distance decrease is controlled.

Definition 4: Distance-Reducing Mappings (DRMs): A DRM has a distance loss which is guaranteed to be not more than a fixed amount per step between any two unremerged paths in the trellis diagram of the resulting trellis code, i.e., for at least one $i \neq j, e_{ij} < d_{ij}$, and for all other $i \neq j, e_{ij} \geq d_{ij} - \delta$, where here δ is some small positive integer.

Note that one type of DRM with which the distance decrease can be readily controlled is the class which only allows distance reduction to occur when $d_{ij} \geq 2$, and which has $\delta = 1$. For example, for some convolutional base codes, the free distance occurs between the shortest remerging paths, or stated more formally mathematically (see, e.g., [11, p. 114]), for these codes, the free distance is equal to the first-order row distance. Using such a base code, the free distance of the resulting trellis code d'_{free} is then lower bounded by

$$d'_{\text{free}} \geq d_{\text{free}} - \tau \cdot \delta \tag{3}$$

where τ is the length (number of steps) of the shortest remerging paths in the trellis diagram.

We obtained and present, in the next section and in Table I, some explicit examples of mappings.

IV. Mappings to Construct Permutation Trellis Codes With $3 \leq M \leq 8$

Convolutional codes of rate $R = k/n, 2 \leq n \leq 8$ suffice for many practical applications and often have decoder implementations of acceptable complexity. Consequently, we restricted this investigation to mappings intended to be used with base codes of these rates.

TABLE I
SOME DISTANCE PRESERVING MAPPINGS $Q(M, n, \delta)$ FOR CONSTRUCTING PERMUTATION TRELLIS CODES

M	Type	Q	Mapping
3	DIM	$Q(3,2,1)$	{231, 213, 132, 123}
4	DCM	$Q(4,4,0)$	{1234, 1243, 1324, 1342, 1423, 1432, 2134, 2143, 3214, 3241, 2314, 2341, 3421, 3412, 3124, 3142}
4	DIM	$Q(4,3,1)$	{1234, 1342, 1423, 3241, 4132, 2314, 2431, 2143}
5	DCM	$Q(5,5,0)$	{51234, 51243, 51324, 51342, 51423, 51432, 52134, 52143, 53214, 53241, 52314, 52341, 53421, 53412, 53124, 53142, 41235, 41253, 41325, 41352, 41523, 41532, 42135, 42153, 43215, 43251, 42315, 42351, 43521, 43512, 43125, 43152}
5	DIM	$Q(5,4,1)$	{12345, 13452, 14523, 15234, 23514, 25143, 21435, 24351, 31542, 32154, 34215, 35421, 52413, 53241, 51324, 54132}
5	DRM	$Q(5,6,-1)$	{12345, 12354, 12543, 12534, 13245, 13254, 13542, 13524, 14325, 14352, 14523, 14532, 15324, 15342, 15423, 15432, 21345, 21354, 21543, 21534, 23145, 23154, 23541, 23514, 24315, 24351, 24513, 24531, 25314, 25341, 25413, 25431, 42153, 42135, 42513, 42531, 41253, 41235, 41523, 41532, 45123, 45132, 45213, 45231, 43251, 43215, 43521, 43512, 32154, 32145, 32514, 32541, 31254, 31245, 31524, 31542, 35241, 35214, 35421, 35412, 34251, 34215, 34521, 34512}
6	DCM	$Q(6,6,0)$	{651234, 651243, 651324, 651342, 651423, 651432, 652134, 652143, 653214, 653241, 652314, 652341, 653421, 653412, 653124, 653142, 641235, 641253, 641325, 641352, 641523, 641532, 642135, 642153, 643215, 643251, 642315, 642351, 643521, 643512, 643125, 643152, 351264, 351246, 351624, 351642, 351426, 351462, 352164, 352146, 356214, 356241, 352614, 352641, 356421, 356412, 356124, 356142, 341265, 341256, 341625, 341652, 341526, 341562, 342165, 342156, 346215, 346251, 342615, 342651, 346521, 346512, 346125, 346152}
6	DIM	$Q(6,4,2)$	{123456, 213654, 421365, 132564, 241536, 435216, 624513, 145623, 264351, 452631, 543162, 362415, 354126, 615432, 536421, 316524}

We are first interested in obtaining DCMs, thus extending our previous work on such mappings in [10]. Furthermore, we set out to find one mapping for each $M, 3 \leq M \leq 8$, in order to exploit up to $M = 8$ degrees of frequency diversity with M-FSK, as explained in the next section.

For small values of M, namely $M = 3$ and 4, we intended to maximize the resulting permutation trellis code's rate, $R' = k/M$, by using an $R = (n-1)/n$ base code with n as large as possible. We thus obtained mappings, involving a subset of 2^n M-digit permutation codewords from the full set of $M!$ permutation codewords, where we maximize n, by letting

$$n = \lfloor \log_2 M! \rfloor \tag{4}$$

where $\lfloor \cdot \rfloor$ represents the floor function.

Note that for $M > 4, n = \lfloor \log_2 M! \rfloor > M$, we then use $R = k/n$ base codes with $n = M$, otherwise it will be impossible to find a DCM. This is due to the fact that $\max\{d_{ij}\} = n$ and $\max\{e_{ij}\} = M$, thus we cannot find a DCM if $n > M$. [Note also that for $5 \leq M \leq 8$, the highest R' that can be achieved using an $R = (n-1)/n$ base code with $n = M$ is reduced by less than 6% from the value, as in (4).]

We next obtain explicit mappings for the values of $M, 3 \leq M \leq 8$. In all our mappings, we assume the same ordering, i.e., a mapping from binary symbols represented with $\{0, 1, \ldots, 2^n - 1\}$ onto the subset of M-tuples that we present. We now introduce the notation $Q(M, n, \delta)$ for our mappings. Here, M and n are as defined previously, while δ represents the lower bound on the distance increase ($\delta > 0$) or the upper bound on the distance reduction ($\delta < 0$) per step in the trellis diagram. For a DCM, $\delta = 0$.

We start with $M = 3$ and $n = 2$. Any mapping will at least be distance-conserving, since $\max\{d_{ij}\} = 2$, while for permutation block codes, $\min\{e_{ij}\} = d_{\min} = 2$. Several distance-increasing maps, similar to the one in Section III, can easily be found by inspection for $M = 3$. Stating our result from Section II

$$Q(3,2,1) = \{231, 213, 132, 123\}. \tag{5}$$

For larger values of M, we can revert to the tree search, as in [10], or to recursive procedures.

For $M = 4$, many different mappings can be found by tree search. We present the following distance-conserving map, using 16 out of the possible 24 permutation code symbols, and found by the tree search as an example

$$Q(4,4,0) = \left\{ \begin{matrix} 1234, 1243, 1324, 1342, \\ 1423, 1432, 2134, 2143, \\ 3214, 3241, 2314, 2341, \\ 3421, 3412, 3124, 3142 \end{matrix} \right\}. \tag{6}$$

For still larger values of M, we can use some map for $M = 4$ as a core mapping or "kernel," and easily extend it recursively. This recursion can be briefly explained as follows.

The set of binary $(n+1)$-tuples can be ordered following normal lexicography, i.e., setting up the standard table of $(n+1)$-bit binary numbers. Note that the first $2^n (n+1)$-bit binary numbers are obtained by prefixing the set of n-bit binary numbers with a most significant bit (MSB) 0, and the second $2^n (n+1)$-bit binary numbers by prefixing the set of binary n-bit binary numbers with an MSB 1. This table can thus be partitioned according to the prefix bit into two subsets, each containing 2^n elements. Within each subset, the intrasubset distance between elements is determined by the $n \times n$ $\mathbf{D}$ matrix, and stays the same. However, the binary prefixes of 0 and 1 account for an additional one unit of distance between two elements from the two different subsets.

As stated, for $M = 3$ and 4, we maximize n with $n = \lfloor \log_2 M! \rfloor$, which, in fact, yields $n = M$, while for $5 \leq M \leq 8$, we also use $n = M$, since for $M \geq 5$, using $n = \lfloor \log_2 M! \rfloor$ will lead to $n > M$, making it impossible to find a DCM. Consequently, once we have found a map for some M, we can extend the M-tuples to obtain a map for $(M+1)$-tuples, in the same way that the set of binary n-tuples is extended into the set of binary $(n+1)$-tuples. We thus use two subsets in Q, each ordered the same as the map for M, and furthermore, let each subset have a different prefix.

For our permutation $(M+1)$-tuples, we can prefix the first subset with the integer $M+1$. We can then choose one of the integers $1, \ldots, M$ as prefix for the second subset, and replace it with integer $M+1$ inside each M-tuple. Note that this procedure will account for an additional two units of distance between two elements from different subsets. The intradistance within a subset is not changed. Using this recursive procedure and the map in (6), we obtained distance-conserving maps for $M = 5$ with prefixes 5 and 4, for $M = 6$ with prefixes 6 and 3, for $M = 7$ with prefixes 7 and 2, and for $M = 8$ with prefixes 8 and 1.

Up to now, we have emphasized DCMs in order to achieve a high-rate $R' = k/M$ for the resulting permutation trellis code. Such codes may then be considered for channels where bandwidth use is important. In other applications, it may be possible to lower the information rate or use more bandwidth in order to obtain either more robustness against interference, or the reduced Viterbi decoder complexity which results when k is smaller. On these channels, it is possible to start the design of a mapping with a smaller set of permutation M-tuples, having a larger $d_{\min}$. In this way, a code with a lower rate R', but with a higher d'_{free}, may be obtained. Here we can use code books with good $d_{\min}$ from, e.g., [1] and [8]. We give the following two examples.

For $M = 4$ and $n = 3$, we constructed a DIM as follows. Refer back to the matrix $\mathbf{D}$, of which an example is given in (2). The maximum values of d_{ij}, that is, $d_{ij} = n$, will always appear on the skew diagonal. For $n = 3$, there are four pairs of 3-b binary tuples which yield these maximum-valued $d_{ij} = 3$. Consider now the $M = 4, d_{\min} = 3$ permutation code book in [1], which has $|C| = 12$. In this code book, we can find four pairs of permutation codewords with Hamming distance $d = 4$ for each pair, and map the four pairs of 3-b binary tuples with $d_{ij} = 3$ in $\mathbf{D}$ onto them. In the $\mathbf{E}$ matrix, all other $e_{ij} \geq 3$ if $i \neq j$, since the code book has $d_{\min} = 3$, while in $\mathbf{D}$, the corresponding $d_{ij} \leq 2$. Any permutation trellis code obtained by using this mapping will show an increase of at least one unit of distance per step above the convolutional base code, since $e_{ij} = 4$ will correspond to all the maximum-valued $d_{ij} = 3$, while for all other $i \neq j, e_{ij} \geq d_{\min} = 3$.

For $M = 6$ and $n = 4$, we can proceed in a similar way, starting with an $M = 6, d_{\min} = 5$ code book from [8] with $|C| = 18$, and find eight pairs of permutation codewords with Hamming distance $d = 6$ for each pair. We can thus obtain an $\mathbf{E}$ matrix such that $e_{ij} = 6$ will correspond to $d_{ij} = 4$, while for all other $i \neq j, e_{ij} \geq d_{\min} = 5$. We thus obtain a distance increase of two units per step.

Finally, we investigate DRMs to better exploit available bandwidth when $M \geq 5$. First, for $M = 5$ and $n = 6$, we constructed a DRM with a loss of distance of at most one. Comparing the $\mathbf{D}$ and $\mathbf{E}$ matrices indicates that 14% of the elements showed a distance loss of one. Furthermore, 73% of the elements showed an increase of one or more. The same structure of the binary code was mostly used in the permutation code for this mapping, though some fine-tuning was necessary by trial and error. By using the same prefix method as described previously, other DRMs can be obtained for larger M values. This has been verified for $7 \leq n \leq 9$ mapped onto $6 \leq M \leq 8$.

All results obtained in this section for $3 \leq M \leq 6$ are summarized in Table I. Due to space limitations, we do not explicitly give the DCMs for $M = 7$ and 8 or DRMs for $6 \leq M \leq 8$: it is easy to write these down, starting with the mappings with $n = 6$ and applying the recursion described above.

We now show that the above concerns are of lesser importance, since first, the best mapping may vary from base code to base code, and second, the difference in performance between different mappings for one base code is not pronounced.

V. Comparing Distance-Preserving Mappings and Base Code Dependencies

Experience in this study, as well as in [10], indicated that if one example of a distance-preserving mapping can be found, a multitude of mappings usually exists. When comparing different mappings, the optimal distance-preserving mapping amongst these for a specific base code can be defined as the mapping for which globally, the best free distance in the resulting trellis code is obtained. This opens the questions of choosing the best mapping, as well as the difference in coding performance obtained when using a nonoptimal mapping. Our experience indicated that it might be a very difficult mathematical study, or a computationally intractable computer search, to find an optimal mapping for large values of M.

The particular base code selected plays a role in the effectiveness of a distance-preserving mapping. The dependence of convolutional codes on the rate, constraint length, and generator polynomials results in differences in trellis structure from one convolutional code to another. This also implies that one particular distance-preserving mapping will not always be optimal for every convolutional base code.

Base-code dependencies can be equalized to a degree by evenly distributing the Hamming distance increment/decrement achieved by the 2^n permutation M-tuples above/below the n-tuples of the base code. This will ensure that the increase/decrease in Hamming distance will be distributed evenly throughout the trellis structure when using any convolutional base code, in spite of structural differences between different base codes. In this way, a mapping can be found which performs similarly for many different base codes, and which may be near-optimal for several base codes. For this reason, we call such mappings near-optimal mappings.

It is interesting to note that DCMs from binary vectors to permutations were also investigated by Chang *et al.* [12] after the initial concept was first presented in [9]. However, our performance simulations showed that the mappings found by employing the recursion method in [12] for $M = 5$ and thus, a higher value of M, are not optimal.

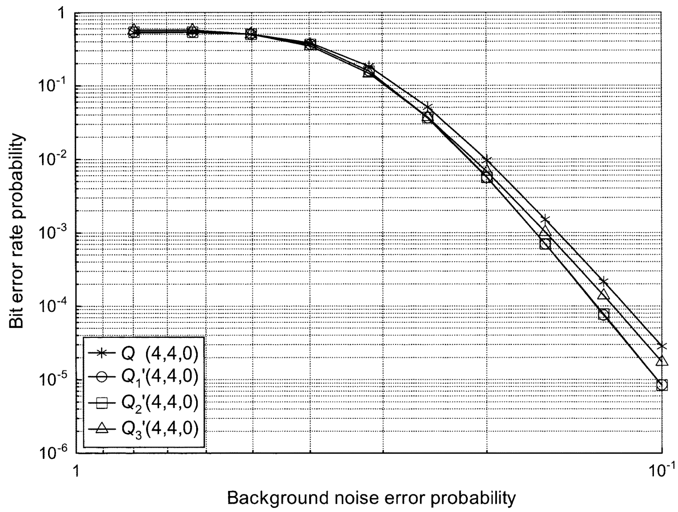

Fig. 3. Bit-error rate (BER) for background noise error probability comparing different mappings, $M = 4$, $R = 1/4$ base code. (See Section VII for probabilistic channel model.)

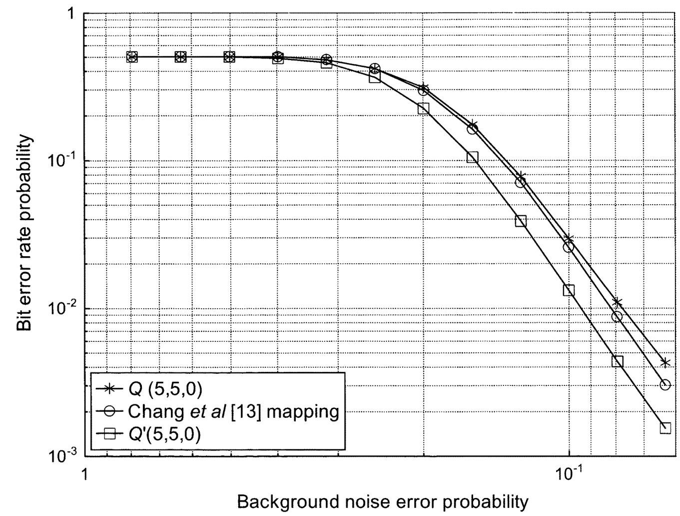

Fig. 4. BER for background noise error probability comparing different mappings, $M = 5$, $R = 4/5$ base code. (See Section VII for probabilistic channel model.)

After much experimentation, computer searches, and an empirical procedure based on the idea of set partitioning, we found the following examples of alternative near-optimal mappings:

$$Q_1'(4,4,0) = \left\{\begin{array}{c} 1234, 1243, 1324, 1342, \\ 2134, 2143, 3124, 3142, \\ 4231, 4213, 4321, 4312, \\ 2431, 2413, 3421, 3412 \end{array}\right\} \tag{7}$$

$$Q_2'(4,4,0) = \left\{\begin{array}{c} 1234, 1243, 1324, 1342, \\ 4231, 4213, 4321, 4312, \\ 2134, 2143, 3124, 3142, \\ 2431, 2413, 3421, 3412 \end{array}\right\} \tag{8}$$

$$Q_3'(4,4,0) = \left\{\begin{array}{c} 1234, 1243, 2134, 2143, \\ 1324, 1342, 3124, 3142, \\ 4231, 4213, 2431, 2413, \\ 4321, 4312, 3421, 3412 \end{array}\right\} \tag{9}$$

$$Q'(5,5,0) = \left\{\begin{array}{c} 12534, 21435, 13254, 24153, \\ 21354, 12345, 23514, 23145, \\ 15243, 51423, 25134, 53241, \\ 41325, 21543, 31524, 35142, \\ 14235, 12453, 34251, 54132, \\ 42513, 32415, 34512, 43152, \\ 54321, 52431, 45231, 35421, \\ 52314, 45312, 43521, 53412 \end{array}\right\}. \tag{10}$$

The near-optimal mapping $Q_3'(4,4,0)$ is equivalent to the one used in [12]. Using the above new mappings, a comparison with $Q(4,4,0)$ from Table I is plotted for a channel with background noise (refer to Section VII) in Fig. 3, using an $R = 1/4, \nu = 2, d_{\text{free}} = 8$ convolutional code as a base code. It is evident from Fig. 3 that the performance differences are small.

In Fig. 4, an $R = 4/5, \nu = 2, d_{\text{free}} = 2$ punctured convolutional code was used as a base code for comparing $Q(5,5,0)$ from Table I, the above $Q'(5,5,0)$, and the $M = 5$ mapping found in [12]. The performance graphs in Fig. 4 are plotted for the same type of channel, and shows that the $M = 5$ mapping published in [12] is not optimal, as we stated above. This indicates that any near-optimal mapping has to be obtained by starting the search process from scratch. However, although mappings found by recursion may not be optimal, the performance graphs in Figs. 3 and 4 indicate that their performance is close enough to those of near-optimal mappings.

VI. Permutation Trellis Codes Combined With M-ary FSK

We now investigate combining our permutation trellis codes with M-ary FSK modulation for application in channels with narrowband, broadband, and background noise disturbances. The power-line communication (PLC) channel is an example of such a channel. For a discussion of the various types of noise that prevail on PLC channels, refer, e.g., to [2] and [3]. Related previous work involving permutation block codes was reported in [1].

Constant envelope signal modulation, such as M-ary FSK, allows a transmitter's power amplifier to operate at or near saturation levels. We show that a combination of M-ary FSK modulation and permutation coding can provide for a constant envelope modulation signal, frequency spreading to avoid bad parts of the frequency spectrum, and time spreading to facilitate correction of frequency disturbances and impulse noise simultaneously. Frequency disturbances like narrowband noise are permanent over a long period of time, and of great importance in communication systems. On the other hand, impulse noise may put energy in larger parts of the spectrum in use for a short period.

In an M-ary FSK modulation scheme, symbols are modulated as one of the sinusoidal waves described by

$$s_i(t) = \sqrt{\frac{2E_s}{T_s}} \cos(2\pi f_i t), \quad 0 \le t \le T_s$$
$$f_i = f_0 + \frac{i-1}{T_s}, \quad 1 \le i \le M \tag{11}$$

where $i = 1, 2, \ldots, M$ and E_s is the signal energy per modulated symbol. The signals are orthogonal, and for noncoherent reception, the frequencies are spaced by $1/T_s$ Hz, this being the transmission rate. To avoid abrupt switching from one frequency to another, the information-bearing signal may modulate a single carrier whose frequency is changed continuously

(CPFSK). The demodulation may be accomplished using $2M$ correlators, two per signal waveform. The "classical" noncoherent demodulator detects M envelopes, and outputs as estimate for the transmitted frequency the one that corresponds to the largest envelope. This makes the demodulation process simple and independent from the channel attenuation.

However, there are several channel disturbances that degrade the performance:

- the "classical" noncoherent demodulation is not optimum in case of a simplified model of a frequency-selective channel, see [13];
- narrowband noise may cause single large envelopes to occur at the demodulator output;
- impulse noise has a broadband character and could lead to a multiple of large envelopes.

To be able to handle the above types of noise processes, we propose to "modify" the demodulator in such a way that the M detected envelopes can be used in the decoding process of permutation trellis codes. For this, every envelope detector $i, 1 \leq i \leq M$, is followed by a threshold T_i. A practical value for a received symbol energy E_i could be $0.6\sqrt{E_i}$. For values above the threshold, we output a one, otherwise, a zero. Hence, instead of a single demodulator output, we now have M outputs per transmitted symbol. A transmitted codeword of length M thus leads to $M \times M$ binary outputs. These outputs are placed in a binary $M \times M$ matrix $\mathbf{Y} = [y_{ij}]$, where i indicates the output of the detector for frequency f_i (symbol i) and $j, 1 \leq j \leq M$, the position or time in the codeword.

The trellis encoder outputs M symbols from a permutation codeword that belong to the particular transition in the trellis. The M-ary symbols are transmitted in time as the corresponding frequencies, and thus, the transmitted signal has a constant envelope.

The modified demodulator outputs a matrix $\mathbf{Y}$ for every received M symbols. For example, suppose that the permutation trellis encoder outputs the four-vector (4, 1, 2, 3). For a noiseless channel, the matrix

$$\mathbf{Y}_{\text{noiseless}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \tag{12}$$

occurs at the output of the "modified" demodulator. Every permutation codeword corresponds to a unique matrix $\mathbf{Y}_{\text{noiseless}}$. An impulsive noise component at time instant $j = 3$, or a narrowband disturbance at frequency f_2, corrupts $\mathbf{Y}_{\text{noiseless}}$, and may result in

$$\mathbf{Y}_{\text{impulse}} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

$$\mathbf{Y}_{\text{narrowband}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \tag{13}$$

respectively.

The permutation block decoder compares the $\mathbf{Y}_{\text{noiseless}}$ for every codeword with the output $\mathbf{Y}$ of the "modified" demodulator and outputs the codeword that leads to a minimum number of differences (minimum distance). The complexity of decoding grows with the number of codewords. To reduce the decoding complexity, we use the Viterbi decoding algorithm for the permutation trellis codes.

In the trellis diagram, every transition between states corresponds to a codeword from the permutation code, and thus a particular $\mathbf{Y}_{\text{noiseless}}$. The trellis decoder compares the particular $\mathbf{Y}_{\text{noiseless}}$ with the output $\mathbf{Y}$ of the "modified" demodulator, and outputs the number of differences (Hamming distance) for the transition. We call this the transition or branch metric in the Viterbi decoding process. More specifically, the branch metric is computed as

$$M - \left[\sum_{1 \leq i,j \leq M} (y_{ij} \wedge y'_{ij}) \right] \tag{14}$$

where $\wedge$ represents the binary AND operation, y_{ij} are the elements in the $\mathbf{Y}_{\text{noiseless}}$ matrix representing the branch, and y'_{ij} are the elements in the $\mathbf{Y}_{\text{received}}$ matrix, representing the output of the "modified" demodulator. The Viterbi decoder gives as an output the path in the trellis that has the minimum overall number of differences. An error event may occur if for a transmitted path in the trellis, the overall number of differences with the demodulator outputs is larger than or equal to that of a competing path. Note that the number of differences for a transition is M minus the number of agreements. Thus, equivalently, we can state that an error event may occur if the number of agreements of the correct path is less than or equal to that of a competing path. The minimum number of differences between any two paths is called the free distance d_{free}.

We now describe the influence of the different noise types on the decoding process.

- *Impulse noise*: Due to the broadband character, impulse noise may cause the demodulator to output the presence of all frequencies in a column (an all-one column) of $\mathbf{Y}$. An incorrect path thus has an agreement at the particular time instant (column). Depending on the signaling rate, more columns may be affected.
- *Permanent narrowband noise*: A permanent disturbance (narrowband noise) present at the subchannel for frequency f_i may lead to a matrix $\mathbf{Y}$, where row i is an all-one row. Since every incorrect transition in the trellis has a corresponding $\mathbf{Y}_{\text{noiseless}}$ with exactly one agreement in the particular row i, the number of agreements for each of these transitions is increased by one.
- *Background noise*: Degrades performance by introducing unwanted (called insertions) demodulator outputs or by causing the absence (called deletion) of a transmitted frequency in the demodulator output. The transmitted codeword (4, 1, 2, 3) for a particular transition may lead to the matrix

$$\mathbf{Y}_{\text{background}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \tag{15}$$

where we have an instant of background noise at $(i, j) = (3, 2)$ and $(2, 3)$, respectively. An insertion may increase the number of agreements for an incorrect transition by one (decreasing the number of differences by one). A deletion increases the number of differences (metric) of the correct transition, or the incorrect transition, as well, by one.

When comparing two trellis paths that are different in d positions, $d - 1$ errors of type impulse or background noise still allow correct decoding. However, for two paths of length L, the narrowband noise may increase the number of agreements for the incorrect path by L. As long as L plus the number of errors of the previous type is less than d, the Viterbi decoder gives the correct output. It is, therefore, important to have permutation trellis codes with a large free distance.

VII. Performance Results

As stated previously, a PLC channel may have an unpredictable and widely varying mixture of noise components, including additive background noise, impulse noise, and permanent frequency disturbers [3]. This may vary from channel to channel, and also, within the same channel, there may be a dependency on time of day or other longer term time variations. Consequently, there is no widely agreed upon "benchmark test" for new coding and modulation schemes such as the one for bandlimited AWGN channels.

However, we obtained some representative results by using a simple model, which generates errors in the received matrix according to certain error parameters. The error parameters were assumed to be equal for all frequency subbands. The different types of noise are generated as follows.

- *Impulse noise*: Each time slot in $\mathbf{Y}_{\text{received}}$ has a certain probability of resulting in an impulse noise. If an impulse disturbance occurs, the entire corresponding column in the matrix would have ones as elements. This is similar to the model of a binary symmetric channel (BSC) when each time slot is seen as an element for M bits.
- *Permanent narrowband noise*: This type of noise is generated by permanently letting a one output occur in a frequency slot. The effect is that all $\mathbf{Y}_{\text{received}}$ would have ones in the row that corresponds to the chosen frequency.
- *Background noise*: Each element in $\mathbf{Y}_{\text{received}}$ has a certain probability of being in error. Thus, a received one has a probability of p_b changing to a zero and vice versa. This is similar to the model of a BSC.

Three different DCM codes, covering a fairly wide range of base code parameters [11], were investigated. For $M = 3$, an $R = 1/2, \nu = 2, d_{\text{free}} = 5$ convolutional code was used as a base code, using the mapping discussed previously. For $M = 4$, an $R = 1/4, \nu = 2, d_{\text{free}} = 8$ convolutional code was used as a base code. Last, for $M = 5$, an $R = 4/5, \nu = 2, d_{\text{free}} = 2$ punctured convolutional code was used as a base code. For comparison, the equivalent uncoded data was sent at the same rate as the overall system for each case, respectively.

Simulation results are presented in Figs. 5–7. It can be seen that for all three interference and noise scenarios studied, our new permutation trellis codes offer a significant performance improvement over uncoded communications. Decoded BER improvements are dependent on the codes' free distances, as can be expected.

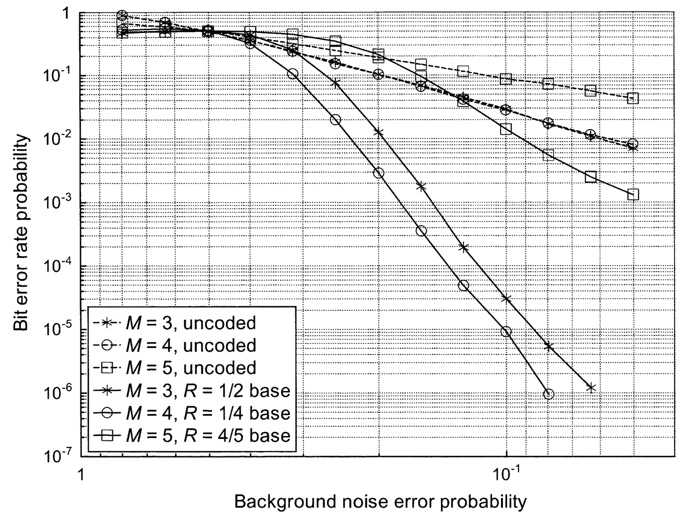

Fig. 5. BER for background noise.

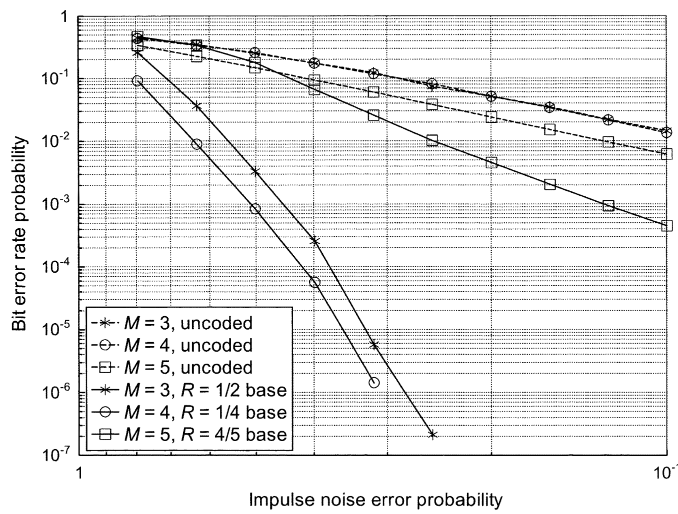

Fig. 6. BER for impulse noise.

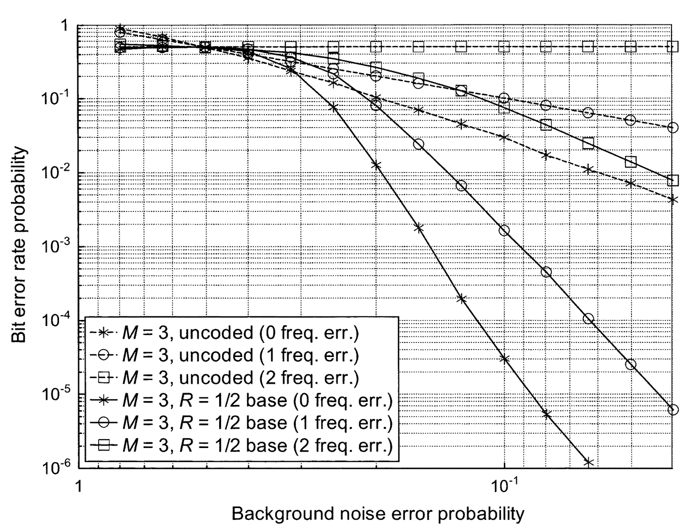

Fig. 7. BER for permanent frequency noise and background noise.

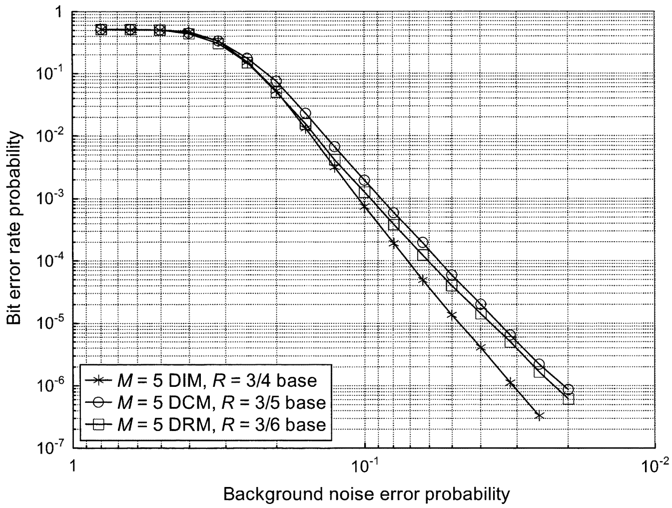

Fig. 8. BER for background noise, comparing DIM, DCM, and DRM.

In addition, the three different mapping types were investigated as follows (using mappings from Table I).

- *DIM*: Use $Q(5, 4, 1)$ to map $M = 5$ onto an $R = 3/4, \nu = 2, d_{\text{free}} = 3$ punctured convolutional base code.
- *DCM*: Use $Q(5, 5, 0)$ to map $M = 5$ onto an $R = 3/5, \nu = 2, d_{\text{free}} = 4$ punctured convolutional base code.
- *DRM*: Use $Q(5, 6, -1)$ to map $M = 5$ onto an $R = 3/6, \nu = 2, d_{\text{free}} = 5$ punctured convolutional base code.

These rates were used to ensure that the overall system rate was the same in all three cases. Also, the guaranteed free distance after mappings for all three codes is $d'_{\text{free}} = 4$. Note that the mappings in these cases are not optimized. The results for permanent frequency noise and background noise are shown in Fig. 8, the impulse noise was omitted as similar trends were observed. It can be seen that the DIM is performing the best. This is attributed to the vast number of mappings one can construct by choosing 16 $M = 5$ codewords from a possible 120, and thus being able to maximize the increase in distance.

VIII. Conclusion

We introduced the new concept of permutation trellis codes, overcoming the disadvantages of permutation block codes. Furthermore, we expanded and generalized our previous concept of a distance-preserving mapping in [10] to now include a DCM, a DIM, and a controlled DRM.

No results on permutation trellis codes preceding our preliminary work in [9] could be found in the literature. Subsequently, inspired by [9], a mathematical theory to construct or enumerate DCMs, similar to those that we constructed with the simple excursion, was developed and very recently published in [12].

With this letter, we presented and applied a code-construction procedure capable of constructing powerful permutation trellis codes, and also gave several explicit results for DCM, DIM, and DRM mappings. By using our construction procedure and the explicit mappings, future researchers can take advantage of the many results on, and vast literature covering, good convolutional codes.

In communications systems, our new permutation trellis codes can be applied in a modulation/coding scheme using M-FSK modulation. The scheme is very robust and simple to implement, and is, thus, attractive for application in PLC. Some simulation results, giving an indication of the performance of the scheme, were also presented.

Acknowledgment

The authors would like to thank the anonymous reviewers for their suggestions to improve the letter.

References

[1] A. J. H. Vinck, "Coded modulation for powerline communications," *A.E.Ü. Int. J. Electron. Commun.*, vol. 54, no. 1, pp. 45–49, 2000.
[2] O. Hooijen, "A channel model for the residential power circuit used as a digital communication medium," *IEEE Trans. Electromagn. Compat.*, vol. 40, no. 4, pp. 331–336, Nov. 1998.
[3] H. C. Ferreira, H. M. Grove, O. Hooijen, and A. J. H. Vinck, "Power line communication," in *Wiley Encyclopedia of Electrical and Electronics Engineering*, J. G. Webster, Ed. New York: Wiley, 1999, vol. 16, pp. 706–716.
[4] D. Slepian, "Permutation modulation," *Proc. IEEE*, vol. 53, no. 3, pp. 228–236, Mar. 1965.
[5] I. F. Blake, "Permutation codes for discrete channels," *IEEE Trans. Inf. Theory*, vol. IT-20, no. 1, pp. 138–140, Jan. 1974.
[6] I. F. Blake, G. Cohen, and M. Deza, "Coding with permutations," *Inf. Control*, vol. 43, no. 1, pp. 1–19, Oct. 1979.
[7] M. Deza and S. A. Vanstone, "Bounds for permutation arrays," *J. Statist. Planning Inference*, vol. 2, no. 2, pp. 197–209, 1978.
[8] T. Kløve, "Classification of permutation codes of length 6 and minimum distance 5," in *Proc. IEEE Int. Symp. Inf. Theory Appl.*, Honolulu, HI, Nov. 2000, pp. 465–468.
[9] H. C. Ferreira and A. J. H. Vinck, "Interference cancellation with permutation trellis codes," in *Proc. IEEE Veh. Technol. Conf.*, Boston, MA, Sep. 2000, pp. 2401–2407.
[10] H. C. Ferreira, D. A. Wright, and A. L. Nel, "Hamming distance preserving mappings and trellis codes with constrained binary symbols," *IEEE Trans. Inf. Theory*, vol. 35, no. 5, pp. 1098–1103, Sep. 1989.
[11] R. Johannesson and K. Zigangirov, *Fundamentals of Convolutional Coding*. Piscataway, NJ: IEEE Press, 1999.
[12] 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.
[13] T. Schaub, "Spread frequency shift keying," *IEEE Trans. Commun.*, vol. 42, no. 2–4, pp. 1056–1064, Feb.–Apr. 1994.