Nonbinary Convolutional Codes and Modified M-FSK Detectors for Power-Line Communications Channel

Khmaies Ouahada

Abstract: **The Viterbi decoding algorithm, which provides maximum-likelihood decoding, is currently considered the most widely used technique for the decoding of codes having a state description, including the class of linear error-correcting convolutional codes. Two classes of nonbinary convolutional codes are presented. Distance preserving mapping (DPM) convolutional codes (*DPM*-CC) and *M*-ary convolutional codes (*M*-CC) are designed, respectively, from the distance-preserving mappings technique and the implementation of the conventional convolutional codes (*C*-CC) in Galois fields (GF) of order higher than two. We also investigated the performance of these codes when combined with a multiple frequency-shift keying (*M*-FSK) modulation scheme to correct narrowband interference (NBI) in power-line communications (PLC) channel. The modification of certain detectors of the *M*-FSK demodulator to refine the selection and the detection at the decoder is also presented. *M*-FSK detectors used in our simulations are discussed, and their chosen values are justified. Interesting and promising obtained results have shown a very strong link between the designed codes and the selected detector for *M*-FSK modulation. An important improvement in gain for certain values of the modified detectors was also observed. The paper also shows that the newly designed codes outperform the conventional convolutional codes in a NBI environment.**

Index Terms: **Convolutional codes, Viterbi decoder, *M*-FSK modulation, envelope detector, power line communications, narrowband interference.**

I. INTRODUCTION

A power-line communications (PLC) channel is considered a hostile channel because three different main types of noise are present [1]. The background noise is equivalent to Gaussian white noise and is usually caused by natural sources such as the radiation from the sun. The impulse noise can be present for a short duration and can affect many or all time slots. This type of noise is usually caused by switching equipment or, sometimes, lightning strikes. The permanent narrow-band noise can affect some frequencies and can also be present over a period of time. The source of this type of noise could be television sets or computer terminals.

To overcome noise in a power-line channel, Vinck *et al.* [2] introduced the concept of coded modulation for PLC. Specifically, they combined *M*-FSK [3] with permutation codes to make transmissions robust against permanent frequency disturbances, impulse noise, and background noise.

We present in this paper the combination of two different non-conventional convolutional codes and *M*-FSK modulation

Khmaies Ouahada is with the Department of Electrical and Electronic Engineering Science, University of Johannesburg, Auckland Park, 2006, South Africa, email: kouahada@uj.ac.za

scheme for use in PLC channel and we investigate their performances to correct the narrowband interference.

The idea behind using *M*-FSK modulation and noncoherent detection in a diversity scheme, which conforms to the European Committee for Electrotechnical Standardization Norms (CEN-ELEC), EN 50065.1, part 6.3.2, is to make transmissions robust against narrowband, broadband, and background noise disturbances in hostile channels [1].

The designed nonconventional convolutional codes with nonbinary outputs are DPM convolutional codes and *M*-ary convolutional codes. The *M*-CC code is considered as an *M*-ary to *M*-ary code based on the same concept and construction as *C*-CC codes [4], [5], but instead in a Galois field of order higher than two. The second type of designed code is the implementation of a distance-preserving mappings technique that maps *C*-CC outputs onto permutation sequences of length *M*.

In addition to the investigation of the performance of the two designed codes in an additive white Gaussian noise (AWGN) channel and in the presence of narrowband noise, a comparative simulation study will be conducted in this paper. This simulation study will present the impact of modifying certain detectors to refine the decision of the envelope detector algorithm.

Narrowband noise causes continuous outputs from our demodulator (and will be discussed in detail later). The outputs represent burst errors to our decoder. Because convolutional codes are sequential codes and have difficulty in combating burst errors, which are presented as continuous outputs from the demodulator as a result of the presence of the narrowband noise, we make use of interleavers [6], [7] to spread out the sequences and thus minimize the effect of the narrowband interference.

The paper is organized as follows. The design of new nonbinary convolutional codes is discussed in Section II. Channel modeling of narrowband interference, an *M*-FSK modulation scheme, and modified detectors are presented in Section III. Simulation results for the combination of nonbinary convolutional codes, and the *M*-FSK modulation scheme, are presented in Section IV. We conclude the paper by comparing the performance of the two designed codes, and presenting the advantages and disadvantages of each.

II. NON-BINARY CONVOLUTIONAL CODES DESIGN

A. DPM Convolutional Codes

Distance-preserving mappings is a coding technique that maps the outputs of a convolutional code to other codewords (e. g., codewords from a permutation codebook with fewer error-correction capabilities).

Ferreira *et al.* [8] made use of distance-preserving mappings from binary sequences to permutation sequences to construct permutation trellis codes and make use of the maximum-
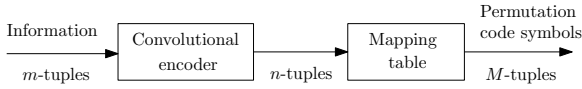
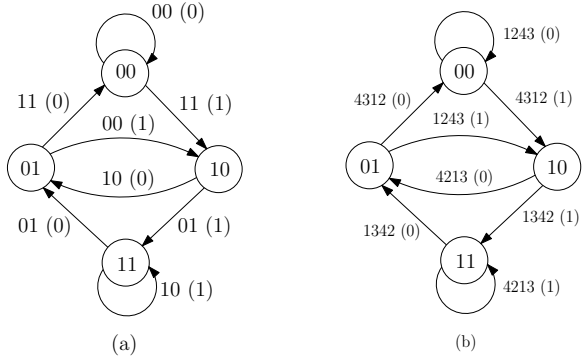Fig. 1. Encoding process for distance-preserving mapping codes

Fig. 2. State systems for (a) convolutional base code and (b) DPM convolutional code

likelihood Viterbi decoding algorithm [8]–[10]. Ferreira *et al.* also combined the permutation trellis codes with *M*-FSK modulation for certain applications in PLC [10]–[15].

In [8] and [11] it was shown how the output binary n-tuple code symbols from an $R = m/n$ convolutional code could be mapped to nonbinary *M*-tuple permutation code symbols to create a DPM convolutional code. This is depicted in Fig. 1.

To explain the mapping technique in a simple way, we present an example where we use the convolutional code with rate $R = 1/2$ and constraint length $K = 3$ as a base code [5]. The output of the encoder, which is a set of binary 2-tuple code symbols, can be mapped to a set of permutation *M*-tuples. The corresponding state systems appear in Fig. 2. Note that, in general, the information transmission rate of the resulting DPM convolutional coded scheme will be in bits per channel use.

The matrices $\mathbf{D} = [\mathbf{d_{ij}}]$ and $\mathbf{E} = [\mathbf{e_{ij}}]$ are, respectively, the Hamming distance matrices between the codewords of the binary input sequences, and the resulting mapped code (see [11], [12] for an expanded definition). The sum of all distances in $\mathbf{E}$ is denoted by $|\mathbf{E}|$, and plays a role in error-correcting capabilities, as explained in [16].

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

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 $e_{ij} \geq d_{ij} + 1$, $\forall i \neq j$. This guarantees an increase in the distance of the resulting code. In addition, we can see the increase in the Hamming distances from the fact that $|\mathbf{D}| = \mathbf{16}$ and $|\mathbf{E}| = \mathbf{32}$. □

In general, three types of DPMs can be obtained, depending on how the Hamming distance is preserved.

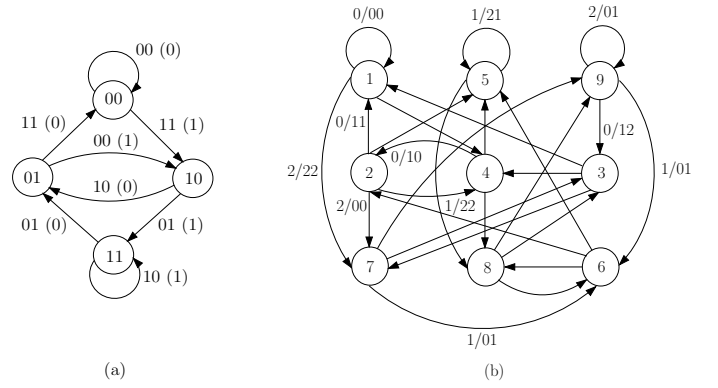• In the case where $e_{ij} \geq d_{ij} + \delta$, $\delta \in \{1, 2, \ldots\}$, $\forall i \neq j$, we call such mappings distance-increasing mappings (DIMs).

• In the case where $e_{ij} \geq d_{ij}$, $\forall i \neq j$ and equality achieved at least once, we have distance-conserving mappings (DCMs).

• In the case where $e_{ij} \geq d_{ij} + \delta$, $\delta \in \{-1, -2, \ldots\}$, $\forall i \neq j$, we have distance-reducing mappings (DRMs).

We use $\mathcal{Q}(\mathcal{M}, n, \delta)$ to denote the DPMs from binary sequences of length n to permutation sequences of length *M*, making use of the k-cube graph construction [17]. The type of mapping is indicated by δ, with $\delta > 0$ for DIMs, $\delta = 0$ for DCMs and $\delta < 0$ for DRMs.

Table 1 presents a few designed DPM codebooks used in the generation of our DPM convolutional codes using k-cube graph construction.

B. M-ary Convolutional Codes

This type of convolutional code is similar to *C*-CC codes but is defined in a Galois field *GF(q)* with $q > 2$. To understand the concept of this class of convolutional codes, we make use of a published example of the *C*-CC where rate $R = 1/2$ and constraint length $K = 3$.

Because our *M*-ary convolutional codes are defined in *GF (q)* with $q > 2$, the two modulo-q adders in the block diagram of the encoders of the new designed codes are based on operations in *GF (q)*. Only a few changes are expected in the properties of the *M*-ary convolutional codes-such as the number of states, which will increase exponentially because it is equal to q^{K-1} for a convolutional encoder in *GF (q)*, where $q > 2$ and *K* is the constraint length of the code.

Example 2 We take the example of $q = 3$, which means that we are working in *GF (3)* with symbols $\{0, 1, 2\}$. These symbols are the input of our convolutional code, and the operations will be addition in *GF (3)*. We have chosen a *C*-CC with of rate $R = 1/2$ and constraint length $K = 3$ to be able to compare its state machine to an *M*-ary convolutional code of the same rate and constraint length.

Fig. 3(a) presents the state machine of a conventional convolutional code with four states. For the same code characteristics but in *GF (3)*, the state machine has nine states, as shown in Fig. 3(b).

We denote $\mathbf{D}^{GF(2)}$ as the distance matrix that compares all possible outputs of the *C*-CC. $\mathbf{D}^{GF(3)}$ is the distance matrix of the *M*-ary convolutional code in *GF (3)*. These distance matrices

Fig. 3. State machine $R = 1/2$, $K = 3$ for (a) binary convolutional code and (b) *M*-ary convolutional code in $GF(3)$

Table 1. Some Constructed *DPM*-CC Outputs based on k-Cube graph Construction

$\mathcal{Q}(\mathcal{M},n,\delta)$	Mapping
$\mathcal{Q}(4,3,1)$	$\{1243, 1342, 4213, 4312, 1234, 1432, 3214, 3412\}$
$\mathcal{Q}(4,4,0)$	$\{1243, 1342, 4213, 4312, 1234, 1432, 3214, 3412, 2143, 2341, 4123, 4321, 2134, 2431, 3124, 3421\}$
$\mathcal{Q}(5,5,0)$	$\{$12345, 52341, 14325, 54321, 32145, 52143, 34125, 54123, 12435, 52431, 13425, 53421, 42135, 52134, 43125, 53124, 21345, 51342, 24315, 54312, 31245, 51243, 34215, 54213, 21435, 51432, 23415, 53412, 41235, 51234, 43215, 53214$\}$
$\mathcal{Q}(6,5,1)$	$\{$135246, 531246, 435216, 431256, 125346, 521346, 425316, 165243, 561243, 465213, 461253, 165342, 561342, 465312, 312564, 352164, 342561, 342165, 213564, 253164, 243561, 612534, 652134, 642531, 642135, 613524, 653124, 643521, 421356, 461352, 243165, 643125$\}$
$\mathcal{Q}(8,8,0)$	$\{$12345678, 56781234, 12347856, 78561234, 14325678, 56781432, 14327856, 78561432, 32145678, 56783214, 32147856, 78563214, 34125678, 56783412, 34127856, 78563412, 12345687, 56871234, 12348756, 87561234, 14325687, 56871432, 14328756, 87561432, 32145687, 56873214, 32148756, 87563214, 34125687, 56873412, 34128756, 87563412, 12346578, 65781234, 12347865, 78651234, 14326578, 65781432, 14327865, 78651432, 32146578, 65783214, 32147865, 78653214, 34126578, 65783412, 34127865, 78653412, 12346587, 65871234, 12348765, 87651234, 14326587, 65871432, 14328765, 87651432, 32146587, 65873214, 32148765, 87653214, 34126587, 65873412, 34128765, 87653412, 12435678, 56781243, 12437856, 78561243, 13425678, 56781342, 13427856, 78561342, 42135678, 56784213, 42137856, 78564213, 43125678, 56784312, 43127856, 78564312, 12435687, 56871243, 12438756, 87561243, 13425687, 56871342, 13428756, 87561342, 42135687, 56874213, 42138756, 87564213, 43125687, 56874312, 43128756, 87564312, 12436578, 65781243, 12437865, 78651243, 13426578, 65781342, 13427865, 78651342, 42136578, 65784213, 42137865, 78654213, 43126578, 65784312, 43127865, 78654312, 12436587, 65871243, 12438765, 87651243, 13426587, 65871342, 13428765, 87651342, 42136587, 65874213, 42138765, 87654213, 43126587, 65874312, 43128765, 87654312, 21345678, 56782134, 21347856, 78562134, 24315678, 56782431, 24317856, 78562431, 31245678, 56783124, 31247856, 78563124, 34215678, 56783421, 34217856, 78563421, 21345687, 56872134, 21348756, 87562134, 24315687, 56872431, 24318756, 87562431, 31245687, 56873124, 31248756, 87563124, 34215687, 56873421, 34218756, 87563421, 21346578, 65782134, 21347865, 78652134, 24316578, 65782431, 24317865, 78652431, 31246578, 65783124, 31247865, 78653124, 34216578, 65783421, 34217865, 78653421, 21346587, 65872134, 21348765, 87652134, 24316587, 65872431, 24318765, 87652431, 31246587, 65873124, 31248765, 87653124, 34216587, 65873421, 34218765, 87653421, 21435678, 56782143, 21437856, 78562143, 23415678, 56782341, 23417856, 78562341, 41235678, 56784123, 41237856, 78564123, 43215678, 56784321, 43217856, 78564321, 21435687, 56872143, 21438756, 87562143, 23415687, 56872341, 23418756, 87562341, 41235687, 56874123, 41238756, 87564123, 43215687, 56874321, 43218756, 87564321, 21436578, 65782143, 21437865, 78652143, 23416578, 65782341, 23417865, 78652341, 41236578, 65784123, 41237865, 78654123, 43216578, 65784321, 43217865, 78654321, 21436587, 65872143, 21438765, 87652143, 23416587, 65872341, 23418765, 87652341, 41236587, 65874123, 41238765, 87654123, 43216587, 65874321, 43218765, 87654321$\}$

are presented as follows:

$$\mathbf{D}^{GF(2)} = \begin{array}{c} \\ 00 \\ 01 \\ 10 \\ 11 \end{array}\begin{array}{cccc} 00 & 01 & 10 & 11 \\ \left[\begin{array}{cccc} 0 & 1 & 1 & 2 \\ 1 & 0 & 2 & 1 \\ 1 & 2 & 0 & 1 \\ 2 & 1 & 1 & 0 \end{array}\right] \end{array},$$

$$\mathbf{D}^{GF(3)} = \begin{array}{c} \\ 00 \\ 01 \\ 02 \\ 10 \\ 11 \\ 12 \\ 20 \\ 21 \\ 22 \end{array}\begin{array}{ccccccccc} 00 & 01 & 02 & 10 & 11 & 12 & 20 & 21 & 22 \\ \left[\begin{array}{ccccccccc} 0 & 1 & 1 & 1 & 2 & 2 & 1 & 2 & 2 \\ 1 & 0 & 1 & 2 & 1 & 2 & 2 & 1 & 2 \\ 1 & 1 & 0 & 2 & 2 & 1 & 2 & 2 & 1 \\ 1 & 2 & 2 & 0 & 1 & 1 & 1 & 2 & 2 \\ 2 & 1 & 2 & 1 & 0 & 1 & 2 & 1 & 2 \\ 2 & 2 & 1 & 1 & 1 & 0 & 2 & 2 & 1 \\ 1 & 2 & 2 & 1 & 2 & 2 & 0 & 1 & 1 \\ 2 & 2 & 2 & 2 & 1 & 2 & 1 & 0 & 1 \\ 2 & 2 & 1 & 2 & 2 & 1 & 1 & 1 & 0 \end{array}\right] \end{array}$$

The size of *M* in our modulation scheme must be the same as the size of the Galois field in which we generate our *M*-ary convolutional code. For example, generating an *M*-ary convolutional code in *GF (4)* is due to the use of 4-FSK modulation. Because the number of states of our *M*-ary convolutional code, equal to q^{K-1}, grows exponentially and rapidly becomes very large for higher values of the Galois field, it will definitely have an impact on the complexity of the Viterbi decoder. Therefore, for better clarity and presentation of the problem investigated in this paper, we limit our studies to *GF (4)* and *GF (8)* to have a lower number of states and to make our Viterbi decoding algorithm simpler [21]. □

III. POWERLINE COMMUNICATION CHANNEL AND *M*-FSK MODULATION

A. Channel Description for Narrowband Noise

In this paper, the narrowband noise is added to the AWGN channel [18], [19]. Its source has a probability $\boldsymbol{p}_n$ to be present for a duration of T_n symbols, where $T_n \in [0,\infty)$. For our
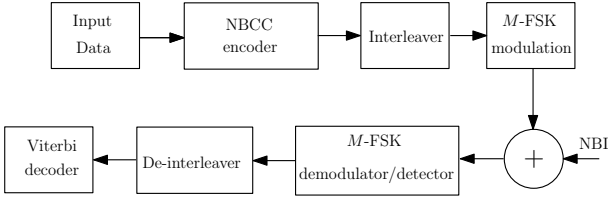
Fig. 4. Transmission system for non-binary convolutional codes with *M*-FSK
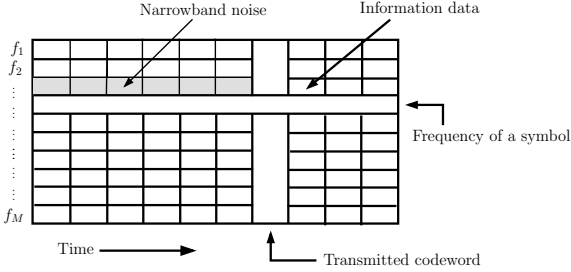

Fig. 5. *M*-FSK time-frequency grid presentation

model, we assume that the narrowband noise signal has an energy E_{ns}, which is much higher than the energy of the transmitted symbols E_s, and a frequency $f_{ns} = f_i$, where f_i corresponds to one of the transmission frequencies. The energy of the narrowband noise, E_s, when added to the transmitted symbol energy n_i that is corrupted by the AWGN, will cause a total saturation of the signal at the corresponding frequency.

B. M-FSK in the Presence of Narrowband Noise

Fig. 4 shows a general transmission system for nonbinary convolutional codes, where the encoded data is interleaved and then combined with the *M*-FSK modulation scheme. We assume noncoherent detection, where the detector bases its detection on the squared envelopes of the received signal samples [20].

In the case of *M*-ary convolutional codes, the duration of the narrowband noise can be longer or shorter than the traceback depth or the delay length, which is $5K$ for the Viterbi decoder [5]. For DPM convolutional codes, we consider a duration that is a multiple of the permutation sequences' length *M*, and is approximately in the range of $5K$ for fair comparisons later.

Fig. 5, a two-dimensional representation of the time-frequency grid of a transmitted sequence, shows how the narrowband noise affects the transmitted data by a total saturation of envelopes, which results in a burst of errors. Therefore, the addition of an interleaver [21] to the transmission system is necessary to minimize the impact of the bursts of errors that the Viterbi decoder cannot usually handle.

C. Non-coherent Detection of Orthogonal Signals with M-ary FSK Modulation

The *M*-ary FSK is considered to be an orthogonal frequency modulation scheme that is identical to OFDM modulation. In our communications system we consider using *M* orthogonal waveforms to transmit information symbols s_i in time, presented by $s_1(t), s_2(t), \ldots, s_M(t)$ and with energy per symbol denoted by $E_{s_i}, 1 \leq i \leq M$.

At the *M*-FSK demodulator, we have $2M$ correlators. Because there are two correlators for each possible transmitted frequency, the receiver for noncoherent demodulation does not have to know the phase. The simplest form of noncoherent demodulation is envelope detection, where the demodulator computes *M* envelopes and chooses the largest envelope as an estimated output for the transmitted frequency.

The time-frequency grid shown in Fig. 5 can be presented as an $M \times L$ matrix denoted by A, with *M* rows corresponding to the number of transmitted symbols and their corresponding *M*-FSK frequencies, and *L* the length of the transmitted symbols with $L > M$. For any integers i, j, with $1 \leq i \leq M$ and $1 \leq j \leq L$, we have $A(i, j) = E_{s_i}$ as the output energy of the transmitted symbol $s_i \in \{1, 2, 3, \ldots, M\}$ at the corresponding frequency $f_i \in \{f_1, f_2, f_3, \ldots, f_M\}$. Because we are using nonbinary convolutional codes in our work, the symbols s_i are represented by a permutation sequence.

Because the *M*-FSK modulation scheme is combined with codes to combat certain types of noise in PLC channel, it is important to introduce the three types of detector/decoder combinations of threshold detection that we have used in our work.

1. The *envelope detector* (ED) computes the largest value of the envelope of each signal. We allocate the value 1 to the corresponding element in the entries of the matrix A. Hence, it is assumed that the corresponding frequency has been transmitted.
2. In this type of *threshold detector* (TD), we refine our decision, and the optimization depends on the energy E_s. We usually choose the TD value of $\tau \times \sqrt{E_s}$, and for practical reasons we choose $0.6 \leq \tau \leq 0.9$.
3. Viterbi [22], [23] proposed a threshold called the *ratio threshold test* (VRTT) for mitigating the partial-band or multitone interferences when invoked as an erasure insertion scheme.

D. Modification of TD and VRTT Detectors to Envelope Detection

The decisions made at the *M*-FSK demodulator by both TD and VRTT detectors can be considered as soft decisions. Although these two detectors are mainly used for erasure channels due to the multiple decisions in their algorithms (as will be detailed later), we can also consider them as a soft-decision detection using the concept of the ED detector.

Because from our previous introduction we are dealing only with a burst of errors caused by the presence of narrowband noise and we are not interested in erasure, we limit our interest to the modification of TD and VRTT detectors to obtain the benefits of the soft-decision property. The changes made for these two detectors play the role of the softening of the ED detector at the *M*-FSK demodulator.

D.1 Modified Threshold Detector

Using the definition of the threshold detector introduced by Vinck [1] and presented previously, we denote by τ the chosen value for the threshold, which we have to compare to the maximum value of the output symbols' energy at the *M*-FSK demodulator.

The original algorithms for the TD detector, and the modified algorithm for it to perform as an envelope detector, are presented as follows:

1. Original Threshold Detector:
- $E_{s_i} = \max(E_{s_1}, \ldots, E_{s_M})$ with $1 \le i \le M$,
- $E_{s_t} = \max(E_{s_1}, \ldots, E_{s_M})$ with $1 \le t \le M, t \neq i$
- $E_{s_i} > E_{s_t}$
- If $E_{s_i} \ge \tau$ and $E_{s_t} < \tau$, then we have an additive error and the output symbol is s_i,
- If $E_{s_i} \ge \tau$ and $E_{s_t} \ge \tau$ then we have an erasure.
- If $E_{s_i} < \tau$, $1 \le i \le M$ then we have an erasure.

2. Modified Threshold Detector:
- $E_{s_i} = \max(E_{s_1}, \ldots, E_{s_M})$ with $1 \le i \le M$,
- We organize the symbols energies in a decreasing order: $E_{s_1} > E_{s_2} > E_{s_3} > \cdots > E_{s_M}$,
- The algorithm is:

```
Algorithm for modified TD
  if E_s1 < τ then choose s1 else
  if E_s2 < τ then choose s1 else
  if E_s3 < τ then choose s2 else
  if E_s4 < τ then choose s3 else
      ⋮         ⋮          ⋮
  E_sM < τ then choose s_{M-1} else
      choose s_M
  end.
```

D.2 Modified Viterbi Ratio Threshold Test Detector

We denote by $\lambda < 1$ the chosen value for the VRTT detector, where we have to compare the second maximum energy of the output symbols from the M-FSK demodulator to the product of the maximum energy of the output symbols to the coefficient λ of the VRTT detector. If the comparison reveals that the second maximum symbol energy is smaller than the main one, then we have an additive error. In contrast, an erasure is considered.

The modified VRTT detector is based on the idea of making an approach of the old VRTT detector to the ED detector. Thus, in the case of erasure detection, we force the detector to take the decision of an ED as presented in the modified VRTT detector, but with extra conditions that help making a soft decision (as will be detailed later). The original algorithms for the VRTT detector and the modified detector are as follows:

1. Original Viterbi Ratio Threshold Tester detector:
- $E_{s_i} = \max(E_{s_1}, \ldots, E_{s_M})$ with $1 \le i \le M$,
- $E_{s_t} = \max(E_{s_1}, \ldots, E_{s_M})$ with $1 \le t \le M$ with $t \neq i$. This is also called the second maximum,
- If $\lambda E_{s_i} \ge E_{s_t}$ then we have an additive error and the output symbol is s_i,
- If $\lambda E_{s_i} < E_{s_t}$ then we have an erasure.

2. Modified Viterbi Ratio Threshold Tester detector:
- $\lambda \in (0, 1]$,
- $E_{s_i} = \max(E_{s_1}, \ldots, E_{s_M})$ with $1 \le i \le M$,
- $E_{s_t} = \max(E_{s_1}, \ldots, E_{s_M})$ with $1 \le t \le M, t \neq i$
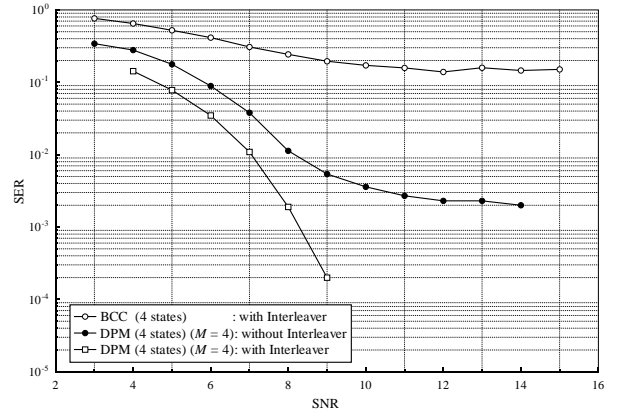- We organize the symbols energies in decreasing order: $E_{s_1} > E_{s_2} > E_{s_3} > \cdots > E_{s_M}$,

Fig. 6. Performance comparison for M-ary convolutional code and C-CC in the presence of NBI

- The algorithm is:

```
Algorithm for modified VRTT
From j = 1 to j = M-1
  if λE_sj > E_s(j+1) then choose s_j
      else choose s_(j+1)
  end
end.
```

The simulation results presented in the following sections are based on using the modified detectors. The use of the modified detectors is simply for studying of the utility and effectiveness of modifying detectors in different channels, such as the narrowband channel.

IV. COMBINED CODING AND M-FSK MODULATION FOR NARROWBAND CHANNEL

When narrowband noise interferes, the output energy of the narrowband noise source exceeds the output energy of the modulator. In this case, the demodulator will have a symbol that is "dominating," which will cause a problem for the Viterbi decoder in decoding these kinds of bursts of errors. Hence, the necessity arises of using an interleaver to combat the burst errors caused by the narrowband interference, as shown in the
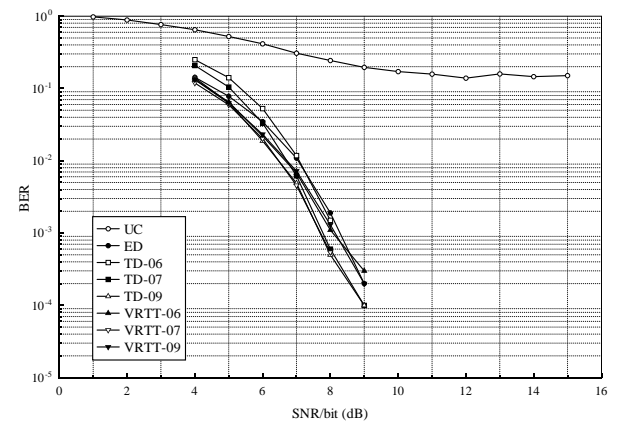

Fig. 7. Performance comparison of DPM convolutional code in the presence of NBI with different detectors and 4-FSK

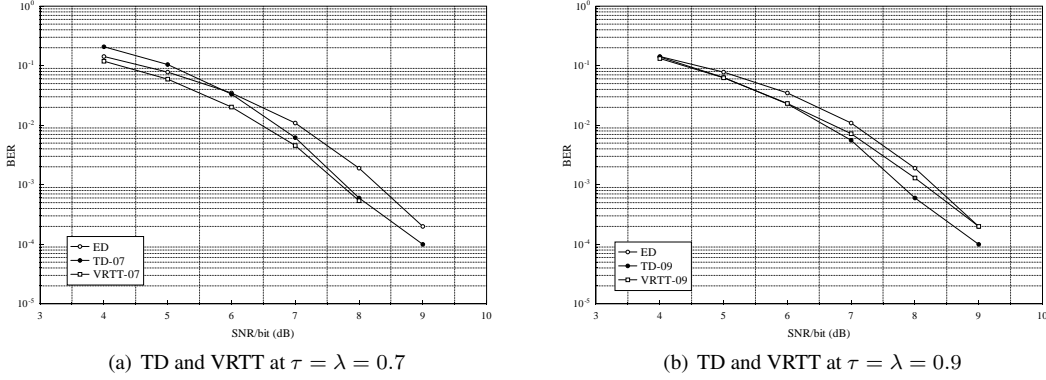(a) TD and VRTT at $\tau = \lambda = 0.7$ (b) TD and VRTT at $\tau = \lambda = 0.9$

Fig. 8. Performance comparison of DPM convolutional code for $M = 4$

general transmission system for nonbinary convolutional codes presented in Fig. 4.

We first run our simulation in the presence of a narrowband interferer with two different convolutional codes and the same number of states, but with outputs from different Galois fields. We combine these two codes with a 4-FSK modulation to compare their performances in the presence of NBI. The simulation was conducted with and without an interleaver.

It is clear from the results in Fig. 6 that the *DPM*-CC code overcomes the effects of narrowband interference, whereas the *C*-CC code fails. The importance of using an interleaver in our simulations is also clear from the differences between the curves.

A. Combined DPM Convolutional Codes and M-FSK

As stated previously, the purpose of combining the permutation codes and *M*-ary FSK modulation is frequency allocation of the permutation symbols when they are transmitted.

In this section we investigate the cases of $M = 4$ and $M = 8$ for our permutation sequences' lengths and their corresponding frequency modulations. We also emphasize the benefit of the mapping when the sum on the Hamming distances comparing the base code is increased using the cube graph construction.

Because in this section we use the mapping of permutation sequences with different lengths of *M*, we will expect a difference in the calculation of the signal-to-noise ratio per bits (SNR/bit) compared to the SNR per symbol, as depicted in the following formula [14]:

$$\frac{1}{M} \times log_2|C| \times E_b = E_s$$

where $|C|$ is the cardinality of the permutation subset.

Example 3 Consider the standard $R = 1/2$, $\nu = 2$, $d_{\text{free}} = 5$ convolutional code, which can be used to map $n = 2$ to $M = 4$ by applying the following mapping:

$$\left\{ \begin{matrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{matrix} \right\} \rightarrow \left\{ \begin{matrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \\ 3 & 4 & 1 & 2 \\ 4 & 1 & 2 & 3 \end{matrix} \right\}.$$

Let **D** and **E** represent, respectively, the Hamming distance matrices for the binary outputs and the permutation codewords

(see [8]), we then obtain

$$\mathbf{D} = \begin{matrix} & 00 & 01 & 10 & 11 \\ 00 \\ 01 \\ 10 \\ 11 \end{matrix} \left[\begin{matrix} 0 & 1 & 1 & 2 \\ 1 & 0 & 2 & 1 \\ 1 & 2 & 0 & 1 \\ 2 & 1 & 1 & 0 \end{matrix} \right],$$

$$\mathbf{E} = \begin{matrix} & 1234 & 2341 & 3412 & 4123 \\ 1234 \\ 2341 \\ 3412 \\ 4123 \end{matrix} \left[\begin{matrix} 0 & 4 & 4 & 4 \\ 4 & 0 & 4 & 4 \\ 4 & 4 & 0 & 4 \\ 4 & 4 & 4 & 0 \end{matrix} \right].$$

This clearly shows that the distance between any two distinct permutation codewords and the corresponding binary outputs increases by more than one. Thus, we are dealing with an increasing mapping.

When we combine this with an *M*-FSK modulator, every symbol, $1, 2, \ldots, M$ corresponds uniquely to one of the *M* frequencies. The *M*-ary symbols are transmitted in time as the corresponding frequencies; thus, the transmitted signal has a constant envelope. For example, if for $M = 4$ the codeword $(1, 4, 2, 3)$ is sent, then the frequencies (f_1, f_4, f_2, f_3) are sent consecutively over the channel. The corresponding bit error rate (BER) curves for each detector are presented in the following simulation results.

Fig. 7 compares our results using the ED detector, the TD detector for different values of the threshold τ, and the VRTT detector for different values of λ.   □

To see the effect of each detector on the error-correction performance of the DPM convolutional code, and to see which one is considered a better choice, we compare our results using the ED detector to the TD detector at the value of $\tau = 0.7$, and to the VRTT detector at the value of $\lambda = 0.7$, as shown in Fig. 8(a). It can be seen that the best choice is the VRTT detector at the value of $\lambda = 0.7$.

In Fig. 8(b) we compare our results for the ED detector to the TD detector at $\tau = 0.9$, and to the VRTT detector at $\lambda = 0.9$. The best value for the threshold for the TD detector is at the value of $\tau = 0.9$.

Taking into consideration the results in Fig. 8, and combining the results in both figures, we can see from Fig. 9 that the VRTT
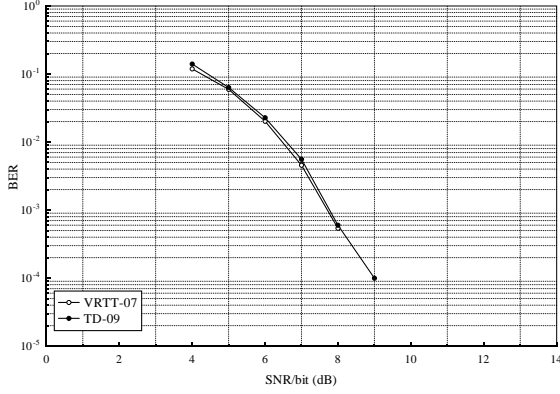
Fig. 9. Performance comparison of DPM convolutional code for $M = 4$ for TD and VRTT detectors
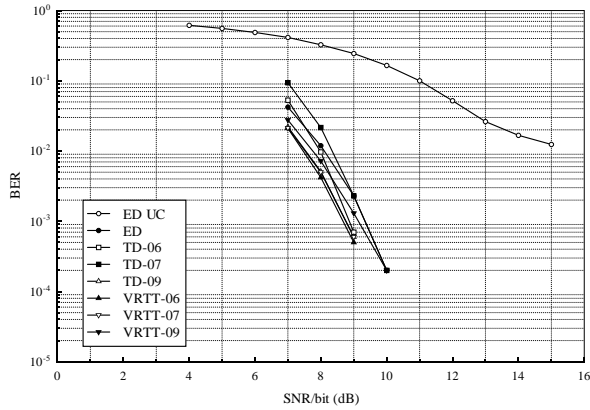

Fig. 10. Performance of DPM convolutional code in the presence of NBI with different detectors and 8-FSK

detector at the value of $\lambda = 0.7$ has the best performance. Thus, we can choose the VRTT detector at the value of $\lambda = 0.7$.
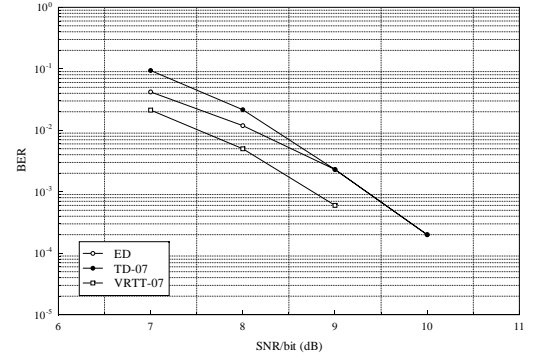
Example 4 In the case of 8-FSK, the standard $R = 1/2$, $\nu = 2$, $d_{\text{free}} = 5$ convolutional code can be used to map $n = 2$ to $M = 8$ by applying the following mapping:

$$\left\{ \begin{array}{cc} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{array} \right\} \rightarrow \left\{ \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 2 & 3 & 4 & 5 & 6 & 7 & 8 & 1 \\ 3 & 4 & 5 & 6 & 7 & 8 & 1 & 2 \\ 4 & 5 & 6 & 7 & 8 & 1 & 2 & 3 \end{array} \right\}.$$
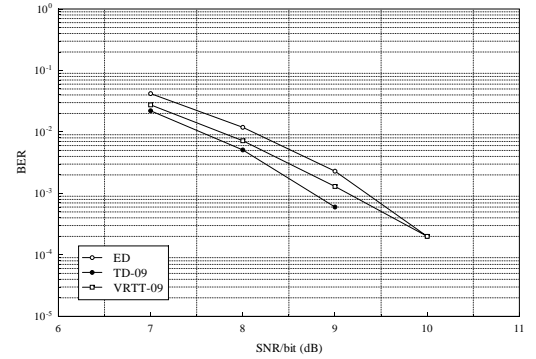
The distance matrices for this case are

$$\mathbf{D} = \begin{array}{c} \\ 00 \\ 01 \\ 10 \\ 11 \end{array} \begin{array}{c} \begin{array}{cccc} 00 & 01 & 10 & 11 \end{array} \\ \left[\begin{array}{cccc} 0 & 1 & 1 & 2 \\ 1 & 0 & 2 & 1 \\ 1 & 2 & 0 & 1 \\ 2 & 1 & 1 & 0 \end{array} \right], \end{array}$$

$$\mathbf{E} = \begin{array}{c} \\ 12345678 \\ 23456781 \\ 34567812 \\ 45678123 \end{array} \begin{array}{c} \begin{array}{cccc} 123...78 & 234...81 & 345...12 & 456...23 \end{array} \\ \left[\begin{array}{cccc} 0 & 8 & 8 & 8 \\ 8 & 0 & 8 & 8 \\ 8 & 8 & 0 & 8 \\ 8 & 8 & 8 & 0 \end{array} \right]. \end{array}$$

(a) TD and VRTT at $\tau = \lambda = 0.7$

(b) TD and VRTT at $\tau = \lambda = 0.9$

Fig. 11. Performance comparison of DPM convolutional code for $M = 8$

It is clear that we have an increasing mapping because the distance between any two distinct permutation codewords and the corresponding binary outputs increases by more than one.

Fig. 10 shows the performance of our distance-increasing mapping code combined with M-FSK for $M = 8$ using the ED detector, the VRTT detector with various values of λ, and the TD detector with various threshold values of τ. $\quad \square$

In Fig. 11(a), we compared our results using the ED detector to the TD detector at the value of $\tau = 0.7$, and to the VRTT detector at the value of $\lambda = 0.7$. It can be seen that the best choice is the VRTT detector at $\lambda = 0.7$.

Fig. 11(b) shows the comparative results between those obtained for the ED detector, those for the TD detector at $\tau = 0.9$, and those for the VRTT detector at $\lambda = 0.9$. It clear that the best results are obtained with the TD at the value of $\tau = 0.9$.

Taking into consideration the comparative results in Fig. 11, it can be seen that the TD detector or the VRTT detector can be good detectors for our DPM convolutional code for $M = 8$. Combining the results in Fig. 11 for the TD and VRTT detectors, we can see in Fig. 12 that the TD and VRTT detectors at values of 0.9 and 0.7, respectively, have the same performance. If we consider the previous results for the DPM convolutional code in the case of $M = 4$, where the best detector is the VRTT detector at the value of $\lambda = 0.7$, we can then choose the VRTT detector to be the best detector that can be used for DPM convolutional code at the specific value of $\lambda = 0.7$.

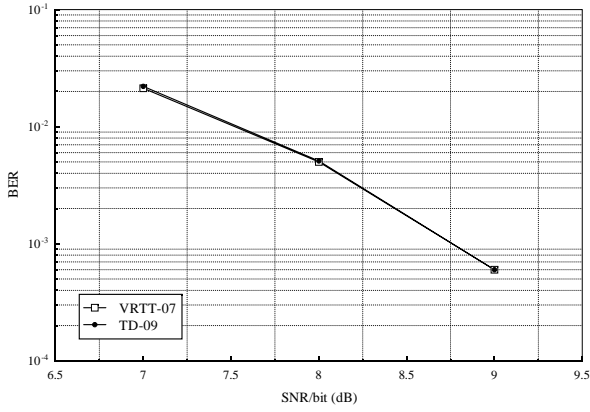The selected values of the detectors for the corresponding

Fig. 12. Performance comparison of DPM convolutional code for $M = 8$ for TD at $\tau = 0.9$ and VRTT at $\lambda = 0.7$
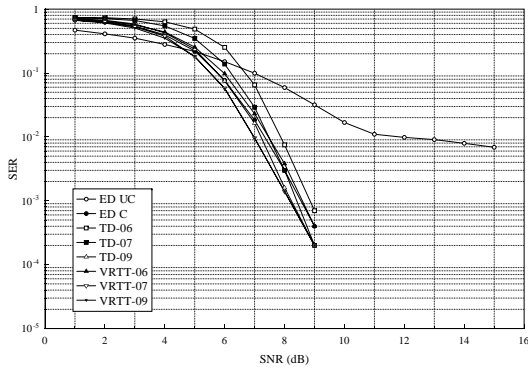

Fig. 13. Performance of M-ary convolutional code in $GF(4)$ in the presence of NBI with different detectors
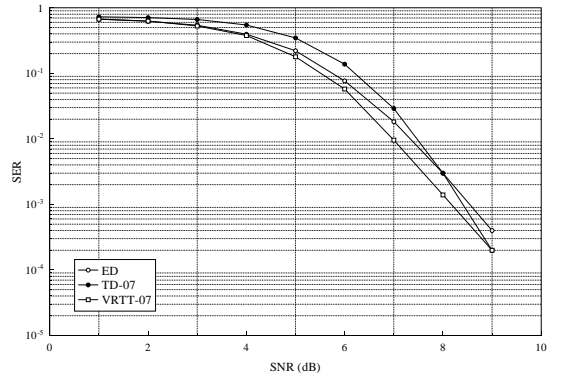

(a) TD and VRTT at $\tau = \lambda = 0.7$

(b) TD and VRTT at $\tau = \lambda = 0.9$

Fig. 14. Performance comparison of M-ary convolutional code in $GF(4)$

nonbinary convolutional codes are presented in Table 2.
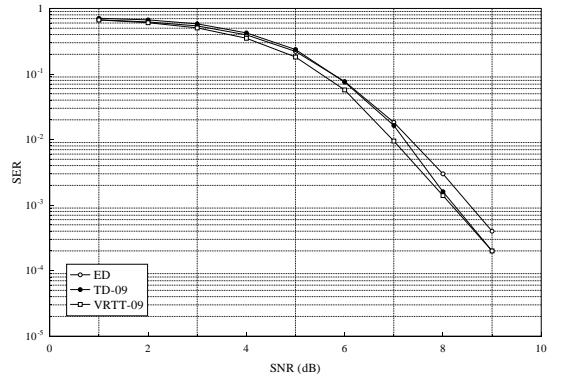
B. Combined M-ary Convolutional Codes and M-FSK

B.1 *M*-ary convolutional code in GF(4)

The modulation scheme plays an important role in the coding technique because the detectors can help in the selection of the output symbols from the demodulator. We run our simulations for uncoded (UC) data using only the envelope detector (ED UC) and our designed code, subject to all different detectors presented previously, to see the effect of each one of them on the error-correction performance of the *M*-ary convolutional code. The idea behind this is to find the optimum detector that can contribute to the minimization of errors. This optimum detector will be the right choice for our modulation scheme. The corresponding symbol error rate (SER) curves for each detector are presented in the following simulation results. As was explained previously about the modifications made for TD and VRTT detectors, the comparison that we are making now is between different ED detectors with different algorithms of selections at the demodulator. The detector that performs best is the one that is considered to be the optimum ED detector.

Fig. 13 shows the simulation results for the ED detector and for different values of τ and λ, respectively, for the TD and VRTT detectors. In our analysis, we will compare the obtained results for the ED detector to the obtained results for similar values of τ and λ that correspond, respectively, to the TD and VRTT detectors.

From the results in Fig. 13, we compare the results for the ED detector and the TD for the value of $\tau = 0.7$ and the VRTT detector for the value of $\lambda = 0.7$. It is clear from the results presented in Fig. 14(a) that the best performance is obtained with the VRTT detector.

A similar comparison was made, but with the obtained results using the ED detector and the TD for the value of $\tau = 0.9$, and the VRTT detector for the value of $\lambda = 0.9$. We can see from Fig. 14(b) that the best performance is obtained with the VRTT detector.

Taking into consideration the comparative results in Fig. 14, we can see that the VRTT detector is the best choice for our *M*-ary convolutional code in *GF(4)*. Combining the results presented in Fig. 14 for the VRTT detector, we can see from Fig. 15 that the VRTT detector performs in a similar manner for the values $\lambda = 0.7$ and $\lambda = 0.9$. Thus, we can choose the VRTT detector for the value of $\lambda = 0.7$.

B.2 *M*-ary convolutional code in GF(8)

As we explained from the results of *M*-ary convolutional code in *GF(4)*, the envelope detector is used for a hard-decision decoding to detect only the best outputs. In the case of soft-decision decoding, we detect multivalued outputs, which are achieved by modifying the threshold detectors (see [14]). These
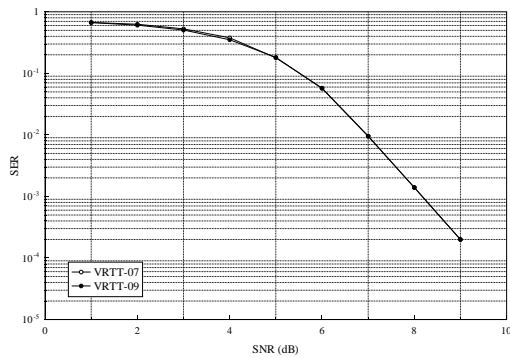
Fig. 15. Performance comparison of *M*-ary convolutional code in $GF(4)$ for VRTT detector
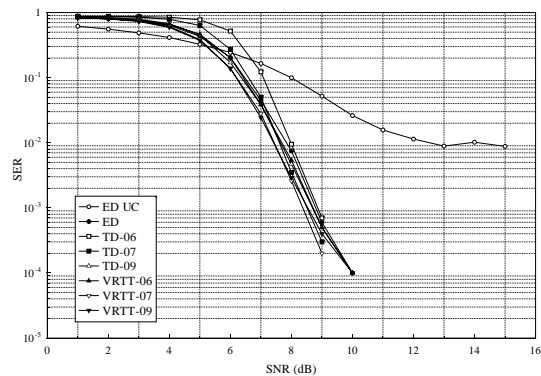

Fig. 16. Performance of *M*-ary convolutional code in $GF(8)$ in the presence of NBI with different detectors

(a) TD and VRTT at $\tau = \lambda = 0.7$

(b) TD and VRTT at $\tau = \lambda = 0.9$

Fig. 17. Performance comparison of *M*-ary convolutional code in $GF(8)$

Fig. 18. Performance comparison of *M*-ary convolutional code in $GF(8)$ for VRTT detector
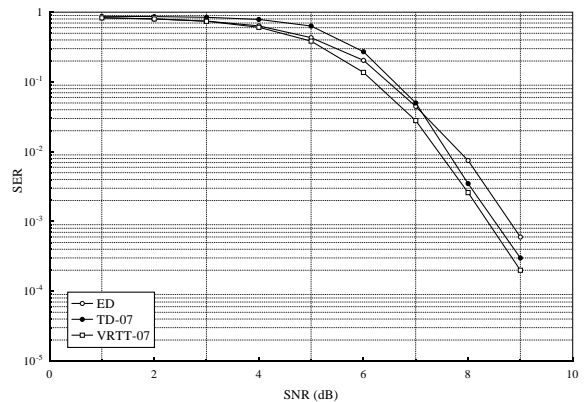
results are depicted in Fig. 16.

Similar to previous analyses, we compare our results for the curves of the ED detector to those of the TD detector at $\tau = 0.7$, and those of the VRTT detector at $\lambda = 0.7$. It is clear from certain selected results presented in Fig. 17(a) that the best choice for a detector is the VRTT at $\lambda = 0.7$.

In the case of certain results presented in Fig. 17(b), where we compare the curves for the ED detector to those of the TD detector at $\tau = 0.9$, and those of the VRTT detector at $\lambda = 0.9$, we see that the best choice is the VRTT detector at the value of $\lambda = 0.9$.
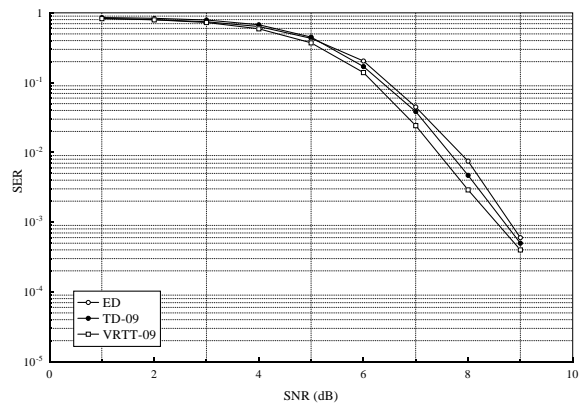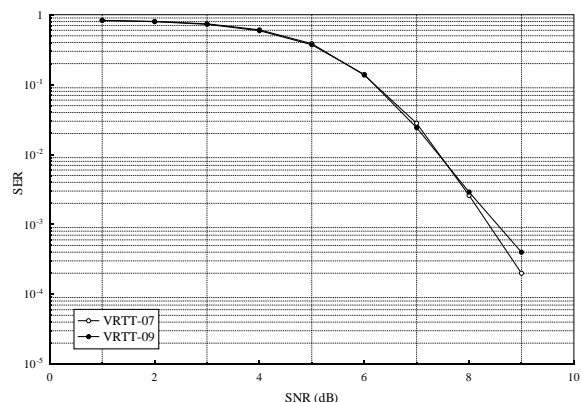
Taking into consideration the comparative results in Fig. 17 and adding results for the VRTT detector, we can see from Fig. 18 that the VRTT detector values of $\lambda = 0.7$ and $\lambda = 0.9$ have almost the same performance, except in the case of higher values of SNR, where the VRTT detector at the value of $\lambda = 0.7$ is considered to be the best. Thus, the VRTT detector is the best choice for *M*-ary convolutional code in *GF(8)* and, more precisely, for the value of VRTT at $\lambda = 0.7$.

V. CONCLUSION

M-FSK modulation has been considered a robust modulation scheme for PLC. Allocating permutation symbols to each frequency will help make easier the error-correction process at the demodulator. To achieve this, we combined two newly designed nonbinary convolutional codes with M-FSK modulation to combat narrowband interference, and to make the information transmission over the PLC channel more robust. Interesting results were achieved.

With regard to the *M*-FSK modulation, we simply modified two detectors that were originally designed for erasures to obtain the benefits of the soft-decision property in their detection algorithms. The obtained results showed an improvement in the performance of the ED detector.

Regarding the newly designed *M*-ary convolutional codes, we can say that they have shown good performance and error-correction capabilities with our different detectors. Unfortunately, they are not that simple to implement, especially with

Table 2. Selected M-FSK Detectors for non-binary convolutional codes

Codeword length (M)	Detector	
	M-ary Convolutional Code	DPM Convolutional Code
4	VRTT-0.7	VRTT-0.7
8	VRTT-0.7	VRTT-0.7

the exponential increase in the number of states. If we punctured the convolutional code, this could reduce the complexity of the Viterbi decoder, and thus, it may be possible to implement the M-ary convolutional codes in hardware.

A very interesting result obtained from the modified detectors shows that the best performance of our nonbinary convolutional codes, combined with the M-FSK modulation in the PLC channel and subject to the narrowband noise, is achieved with the use of the VRTT detector at the value of $\lambda = 0.7$. This is summarized in Table 2. This can be explained and justified by the fact that Viterbi has used the same concept in his decoding algorithm, which resulted in the best decoding algorithm for convolutional codes.

As we know, the selection of the survivor path in the trellis is based on the selection of the smallest metric in each branch, which is actually the accumulation of the previous metrics along the trellis. This gives better accuracy in the selection of the decoded data because the information is built up, upon one another, throughout the trellis. In our case, when we look at the algorithms of the TD detector and the VRTT detector, we can see that the selection of the symbols' energies in the VRTT detector is based on the comparison of the symbols' energies with the λ factor, which creates a dependency between them (as was explained in the algorithm). By contrast, the TD detector deals with the symbols' energies separately, and therefore, the final decision is not that accurate, as in the case of the VRTT detector. This provides a better explanation as to why we found that the VRTT detector is the best option for our nonbinary convolutional codes-which, in fact, have the same structure as the C-CC codes.

Although we have used distance-increasing mapping codes combined with 4-FSK and 8-FSK modulations, we can make use of other types of mappings such as distance-reducing mappings. The used length of the mapped codewords M is usually shorter than the length of the convolutional base code output codewords n, which will save frequency bandwidth.

REFERENCES

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

[2] A. J. H. Vinck, J. Haering, and T. Wadayama, "Coded M-FSK for power line communications," *in Proceedings of the IEEE International Symposium on Information Theory*, Sorrento, Italy, June 25–30, 2000, p. 137.

[3] A. Matache and J. A. Ritcey, "Optimum code rates for noncoherent M-FSK with errors and erasures decoding over Rayleigh fading channels," *in Conference Record of the Thirty-First Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA USA, Nov. 2–5, 1997, pp. 62–66.

[4] K. Ouahada, H. C. Ferreira, A. J. H. Vinck, and D. J. J. Versfeld, "Combined nonbinary codes and M-FSK modulation for power line communications," *in Proceedings of the International Symposium on Power-Line Communications and its applications*, Orlando, Florida, USA, Mar. 26–29, 2006, pp. 110–115.

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

[6] G. D. Forney Jr., "Burst-correcting codes for classic bursty channel," *IEEE Transactions on communications*, vol. 19, pp. 772–781, Oct. 1971.

[7] K. Andrews, C. Heegard, and D. Kozen, "A theory of interleavers," Ithaca, NY, USA, Tech. Rep., 1997.

[8] 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.

[9] G. D. Forney Jr., "The Viterbi algorithm," in *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–277, Mar. 1973.

[10] 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.

[11] 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.

[12] H. C. Ferreira, I. de Beer, and A. J. H. Vinck, "Distance preserving mappings onto convolutional codes revisited," *Proceedings of the IEEE Information Theory Workshop*, Breisach, Germany, pp. 23–26, Apr. 26–29, 2002.

[13] A. J. H. Vinck and J. Haering, "Coding and modulation for power line communications," *Proceedings of the International Symposium on Power-Line Communications and its Applications*, Limerick, Ireland, pp. 265–272, Apr. 5–7, 2000.

[14] A. J. H. Vinck, "Coding for Non-AWGN channels," *8-th International Symposium on Communication Theory and its Applications 2005*, Ambleside, Lake District, UK, pp. 258–261, July, 17–22, 2005.

[15] K. W. Shum, "Permutation coding and M-FSK modulation for frequency selective channel," in *IEEE Personal, Indoor and Mobile Radio Communications*, vol. 13, no. 5, pp. 2063–2066, Sept. 2002.

[16] 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.

[17] K. Ouahada and H. C. Ferreira, "k-Cube construction mappings from binary vectors to permutation sequences," *Proceedings of the IEEE International Symposium on Information Theory*, Seoul, South Korea, pp. 630–634, June 28–July 3, 2009.

[18] H. C. Ferreira, H. M. Grove, O. Hooijen, and A. J. H. Vinck, "Power line Communication," *Wiley Encyclopedia of Electrical and Electronics Engineering*, J. G. Webster Editor, vol. 16, John Wiley and Sons, New York, pp. 706–716, 1999.

[19] D. Galda and H. Rohling, "Narrow Band interference reduction in OFDM-based power line communication systems," *Proceedings of the International Symposium on Power-Line Communications and its Applications*, Malmö, Sweden, pp. 345–351, Apr. 4–6, 2001.

[20] John G. Proakis and Masoud Salehi, *Communication Systems Engineering*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1994.

[21] P. Sweeney, *Error Control Coding: From Theory to Practice*. England: John Wiley and Sons, Ltd., 2002.

[22] A. J. Viterbi, "A robust ratio-threshold technique to mitigate tone and partial band jamming in coded M-FSK systems," *in Proceedings IEEE of the International Conference on Military Communications*, Oct. 1982, pp. 22.4.1–22.4.5.

[23] L. F. Chang and R. J. M. Eliece, "A study of Viterbi's ratio threshold AJ technique," *in Proceedings of the International Conference on Military Communications*, Oct. 1984, pp. 11.2.1–11.2.5.