Permutation Modulation for Fading Channels

Alessandro Nordio and Emanuele Viterbo
Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy
email: nordio@polito.it, viterbo@polito.it

Abstract— **Permutation codes are special spherical codes designed for the band-limited Gaussian channel. Here, we investigate their application to fading channels: we present a simple maximum likelihood decoding algorithm and compute expressions for the codeword error probability.**

Keywords: Permutation modulation, fast fading, slow fading.

I. Introduction

Permutation modulation has been proposed by David Slepian in 1965 [3]. A Variant I permutation modulation is the set of codewords (signals) obtained by taking all permutations of an initial vector in the n-dimensional Euclidean space. A Variant II permutation modulation is the set of codewords obtained by taking all permutations and sign changes of the components of an initial vector in the n-dimensional Euclidean space. Good permutation modulations may be designed by appropriately selecting the initial vector and may be very efficiently decoded in the AWGN channel by essentially applying a sorting algorithm to the received signal vector [3]. In the literature the performance of such modulations was analyzed for AWGN channels only, in this paper we study their performance over fast and slow fading channels.

This article is organized as follows: the next section recalls the basic notation of permutation modulations; Section III provides an efficient algorithm for mapping and demapping permutation codes; a description of the considered channel models is given in Section IV. In Section V the Maximum Likelihood (ML) decoding algorithm for AWGN channel is recalled; in Sections VI and VII a ML decoding algorithm for fast fading channels and its analytical performance are derived. Moreover, in section VIII a Zero Forcing (ZF) suboptimal approach is considered. Finally the performance of permutation modulations for slow fading channels are analyzed in section IX. Examples of performance in terms of error probability are then given in section X, along with conclusive remarks.

II. Basic definitions

Let $\{\mu_1, \ldots, \mu_k\}$ be a set of distinct real numbers with $0 \leq \mu_1 < \mu_2 < \cdots < \mu_k$ and let $\{m_1, \ldots, m_k\}$ be a set of positive integers such that $n = \sum_{j=1}^{k} m_j$. Consider the initial vector with components sorted in ascending order

$$\mathbf{x}_0 = [\underbrace{\mu_1, \ldots, \mu_1}_{m_1}, \underbrace{\mu_2, \ldots, \mu_2}_{m_2}, \quad \cdots \quad , \underbrace{\mu_k, \ldots, \mu_k}_{m_k}] \ . \quad (1)$$

A Variant I permutation code consists of the set $\mathcal{S}$ of vectors obtained by permuting the components of the initial vector $\mathbf{x}_0$. The total number of codewords in such a code is

$$M_{\mathrm{I}} = \frac{n!}{m_1!\, m_2! \cdots m_k!} \ . \quad (2)$$

The Variant I code with $k = 2$, $m_1 = n-1$, $m_2 = 1$ and $\mu_1 = 0$ is the well known PPM or orthogonal modulation.

A Variant II permutation code consists of the set $\mathcal{S}$ of vectors obtained by permuting and applying all possible sign changes to the components of the initial vector $\mathbf{x}_0$. The total number of codewords in this code is

$$M_{\mathrm{II}} = \frac{2^h n!}{m_1!\, m_2! \cdots m_k!} \quad (3)$$

where $h = n - m_1$, if $\mu_1 = 0$ and $h = n$, if $\mu_1 > 0$. The Variant II code with $k = 2$, $m_1 = n-1$, $m_2 = 1$ and $\mu_1 = 0$ results in the well known biorthogonal modulation. The Variant II code with $k = 1$, $m_1 = n$, $\mu_1 \neq 0$ yields an n-bit PCM. In this case the points of $\mathcal{S}$ correspond to the 2^n vertices of an n dimensional hypercube of edge length $2\mu_1$, centered at the origin.

It is clear that all codewords of both Variant I and II codes lie on a hypersphere of radius $\sqrt{n\mathcal{P}}$ centered at the origin, where $\mathcal{P} = \frac{1}{n}\sum_{j=1}^{k} m_j \mu_j^2$.

III. Mapping and Demapping

Let us consider the permutation code $\mathcal{S}$ with parameters n, k with $\mathbf{x}_0$ specified by $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_k]$ with multiplicities $\mathbf{m} = [m_1, \ldots, m_k]$. The total number of available codewords $M = |\mathcal{S}|$ corresponds to a theoretical spectral efficiency of $\log_2 M/n$ bit/s/Hz. In general, since M is not a power of 2, a maximum of $B = \lfloor \log_2 M \rfloor$ bits can be mapped to each codeword yielding a spectral efficiency of B/n bit/s/Hz. The $M - 2^B$ remaining codewords will be purged from the codebook.

In the following we present a mapping/demapping algorithm that uniquely associates a vector $\mathbf{b} = [b_1, \ldots, b_B]$ containing B binary digits to a codeword of the permutation code. This algorithm is based on the fact that the elements of $\boldsymbol{\mu}$ are ordered in increasing order and it is possible to list all the codewords in lexicographic order as shown in Figure 1. In practice the first element of the list, with index 0 is

$$\mathbf{x}_0 = [\underbrace{\mu_1, \ldots, \mu_1}_{m_1}, \ldots, \underbrace{\mu_k, \ldots, \mu_k}_{m_k}] \quad (4)$$

while the last one is

$$\mathbf{x}_{M-1} = [\underbrace{\mu_k, \ldots, \mu_k}_{m_k}, \ldots, \underbrace{\mu_1, \ldots, \mu_1}_{m_1}] \quad (5)$$

0-7803-7661-7/03/$17.00©2003 IEEE.

and has index $M-1$. The mapping/demapping algorithm associates the q^{th} codeword of the list with the binary vector $\mathbf{b}$ whose decimal representation is q, for $q = 0, 1, \dots, 2^B - 1$. Clearly, the last $M - 2^B$ elements of the list cannot be mapped or demapped.

A. Mapping

Given the codeword index q, corresponding to the binary vector $\mathbf{b}$, the mapping function works sequentially in n steps, finding one by one the codeword components. In the first step the algorithm splits the ordered list in k contiguous subsets labeled $\mathcal{S}_j$, for $j = 1, \dots, k$, each one characterized by the first component of the codeword. That is, the subset $\mathcal{S}_j$ contains all codewords whose first component is μ_j, as shown in Figure 1. The cardinality of the subset $\mathcal{S}_j$ is

$$|\mathcal{S}_j| = \frac{(n-1)!}{(m_j-1)!\prod_{i=1,i\neq j}^{k} m_i!} = \frac{m_j}{n}M \qquad (6)$$

that corresponds to the number of codewords of a permutation code with parameters $n-1, k$ with multiplicity $[m_1, \dots, m_j - 1, \dots, m_k]$. Moreover from (6) notice that $|\mathcal{S}_j|$ can be simply written as a function of the total number of permutation M.

The subset $\mathcal{S}_j$ contains codewords with indexes q in the range

$$\sum_{i=1}^{j-1} |\mathcal{S}_i| \leq q \leq \sum_{i=1}^{j} |\mathcal{S}_i| - 1 \ , \qquad (7)$$

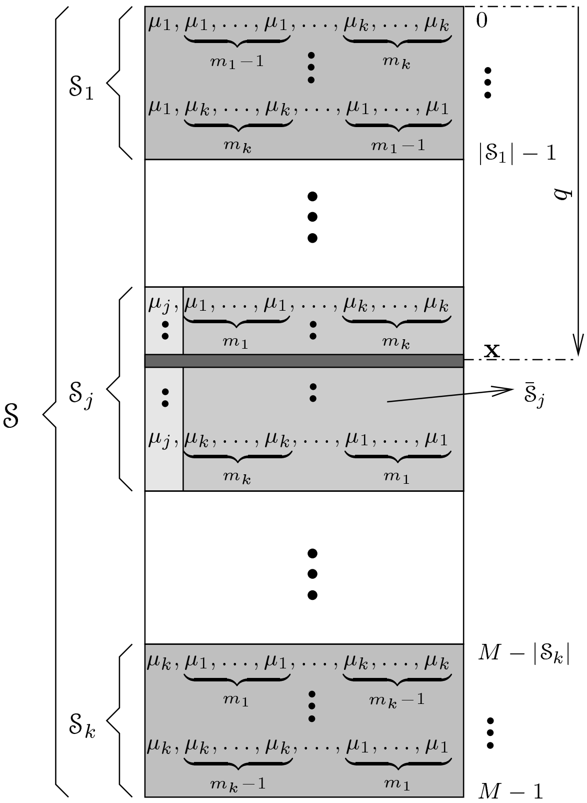
so, if q is in this range then the first element of the corresponding codeword is μ_j.

Suppose now that the subset $\mathcal{S}_j$ contains the index q, then the first component of the codeword, $x_1 = \mu_j$, is known. The next step proceeds by searching through the set $\mathcal{S}_j$ and neglecting the first element of the codewords. In practice it focuses on the set of codewords corresponding to a permutation code with parameters $n-1$, k, $\boldsymbol{\mu} = [\mu_1, \dots, \mu_k]$ and multiplicities $[m_1, \dots, m_j - 1, \dots, m_k]$, which is denoted by $\bar{\mathcal{S}}_j$, in the figure. This repeats until the last element of the codeword is found.

The pseudo code corresponding to the mapping function is given below:

```
Function $\mathbf{x}$ = Mapping($q$)
  Let $\bar{\mathbf{m}} = \mathbf{m}$
  Let $M^{(1)} = M$
  for $i = 1, \dots, n$
    let $s_l = q - \sum_{p=1}^{l} \bar{m}_p M^{(i)}/(n-i)$
         $l = 0, \dots, k-1$
    $j = \arg\min_l(s_l), \quad s_l > 0$
    $x_i = \mu_j$
    $q = s_j$
    $M^{(i+1)} = \bar{m}_j M^{(i)}/(n-i)$
    $\bar{m}_j = \bar{m}_j - 1$
  end
```

This function has an average complexity proportional to $nk/2$.



$\mathcal{S}$
$\mathcal{S}_1$
$\mathcal{S}_j$
$\mathcal{S}_k$
$\mu_1, \underbrace{\mu_1, \dots, \mu_1}_{m_1-1}, \dots, \underbrace{\mu_k, \dots, \mu_k}_{m_k}$
$\mu_1, \underbrace{\mu_k, \dots, \mu_k}_{m_k}, \dots, \underbrace{\mu_1, \dots, \mu_1}_{m_1-1}$
$\mu_j, \underbrace{\mu_1, \dots, \mu_1}_{m_1}, \dots, \underbrace{\mu_k, \dots, \mu_k}_{m_k}$
$\mu_j, \underbrace{\mu_k, \dots, \mu_k}_{m_k}, \dots, \underbrace{\mu_1, \dots, \mu_1}_{m_1}$
$\mu_k, \underbrace{\mu_1, \dots, \mu_1}_{m_1}, \dots, \underbrace{\mu_k, \dots, \mu_k}_{m_k-1}$
$\mu_k, \underbrace{\mu_k, \dots, \mu_k}_{m_k-1}, \dots, \underbrace{\mu_1, \dots, \mu_1}_{m_1}$
0
$|\mathcal{S}_1| - 1$
q
$\mathbf{x}$
$\bar{\mathcal{S}}_j$
$M - |\mathcal{S}_k|$
$M - 1$

Fig. 1. Codewords of the permutation code $\mathcal{S}$ ordered in lexicographic order

A similar mapping/demapping algorithm for permutation modulations has been proposed in [2]. However, because of (6), the algorithm proposed in this paper does not need the storage of the coefficients $|S_j|$ that would require an amount of memory exponential with k.

B. Demapping

The demapping algorithm finds the codeword index q given the codeword $\mathbf{x}$. As well as for the mapping algorithm it proceeds in n steps through the elements of $\mathbf{x}$. If the first element of the codeword is $x_1 = \mu_j$ then the codeword $\mathbf{x}$ lies in the set $\mathcal{S}_j$ and the index q is in the range defined by (7). The search then proceeds through the subset $\bar{\mathcal{S}}_j$ by reducing at each step the cardinality of the subset that contains the codeword $\mathbf{x}$ until the index q is found. The corresponding pseudo code of this algorithm is:

```
Function $q$ = Demapping(x)
  Let $\bar{\mathbf{m}} = \mathbf{m}$
  Let $M^{(1)} = M$
  Let $q = 0$
  for $i = 1, \dots, n$
    for $j = 1, \dots, k$
      if $x_i == \mu_j$ then break
    end
    let $q = q + \sum_{h=1}^{j-1} \bar{m}_h M^{(i)}/(n-i)$
    $M^{(i+1)} = \bar{m}_j M^{(i)}/(n-i)$
    $\bar{m}_j = \bar{m}_j - 1$
```

```
end
```

The demapping function has an average complexity proportional to $nk/2$.

IV. Channel models

Digital transmission over the band-limited fading channel is commonly modeled in the n-dimensional Euclidean space $\mathbb{R}^n$ as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}$$

where $\mathbf{y}$ is the received signal vector, $\mathbf{x} = [x_1, \ldots, x_n]$ is the transmitted signal vector (or codeword) taken from a finite signal constellation (or codebook) $\mathcal{S}$, $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \frac{N_0}{2}\mathbf{I})$ is a real Gaussian random vector with i.i.d. components and $\mathbf{H} = \text{diag}(h_1, \ldots, h_n)$ is the diagonal channel fading matrix. We distinguish two cases:

- *slow fading*, where the fading matrix is $\mathbf{H} = h\mathbf{I}$ and h changes independently from one codeword to the next;
- *fast fading*, where the fading coefficients h_i change independently in each component.

Finally, we note that when $\mathbf{H} = \mathbf{I}$ is constant, we fall back into the AWGN channel model. Assuming that the channel state information is available at the receiver, ML detection is given by

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathcal{S}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 = \arg\min_{\mathbf{x}'\in\mathcal{S}'} \|\mathbf{y} - \mathbf{x}'\|^2 \qquad (8)$$

where $\mathcal{S}'$ is the "faded signal constellation" $\mathbf{H}\mathcal{S}$. We note that the complexity of the ML receiver greatly depends on the structure of the code $\mathcal{S}$. In the worst case a total of $M = |\mathcal{S}|$ Euclidean distances must be computed. For large values of M this may be impractical, hence it is common to trade some of the performance for a reduced decoding complexity. Many classical forward error correcting codes have been selected for applications because they have simple decoding algorithms. In the case of AWGN channel an efficient decoding algorithm, recalled in the following section, was proposed by Slepian in [3].

V. ML Decoding for AWGN channel

Let us consider ML decoding of Variant I codes. We need to find the minimum of the the quantities

$$\|\mathbf{y} - \mathbf{x}_i\|^2 = \|\mathbf{y}\|^2 + n\mathcal{P} - 2(\mathbf{y}, \mathbf{x}_i) \qquad (9)$$

for $i = 0, \ldots, M-1$ where $(\mathbf{y}, \mathbf{x}_i)$ denotes the scalar product of the two vectors. Since $\|\mathbf{y}\|$ is independent of i the ML decoder may simply maximize the scalar product between the received vector and the codewords, i.e.

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}\in\mathcal{S}} \sum_{k=1}^{n} x_k y_k \ . \qquad (10)$$

This maximization problem may be solved as follows. Given the received vector $\mathbf{y}$, replace the smallest m_1 components by the values μ_1, replace the smallest m_2 remaining components with μ_2, etc. Until all the components have been replaced.

For Variant II codes ML decoding can be performed as follows.

1) Take the absolute value of the components of the received vector $\mathbf{y}$, i.e. let

$$\mathbf{y}' = [|y_1|, |y_2|, \ldots, |y_n|]$$

2) Apply the decoder of Variant I codes to $\mathbf{y}'$ to make a first decision $\mathbf{x}'$.
3) The final decision is given by

$$\hat{\mathbf{x}} = [\text{sgn}(y_1)x'_1, \ \text{sgn}(y_2)x'_2, \ \ldots, \ \text{sgn}(y_n)x'_n]$$

where $\text{sgn}(x) = +1$, if $x \geq 0$ and $\text{sgn}(x) = -1$, if $x < 0$.

It can be shown that the above algorithm is equivalent to solving the maximization problem (10).

The complexity of these decoding algorithms is rather small if compared to the brute force exhaustive search. In particular, it is enough to perform a sorting algorithm on the n components of the received vector and to keep track of the final index permutation. This permutation uniquely identifies the ML decoded codeword and the corresponding information bit label may be easily recovered by the demapping algorithm. Sorting can be performed with a complexity of $O(n\log(n))$, whereas exhaustive decoding requires Mn multiplications and $M(n-1)$ additions.

The average codeword error probability with ML detection in the AWGN channel is given by

$$\Pr(e) = \frac{1}{M} \sum_{i=0}^{M-1} \int_{\overline{\mathcal{R}_i}} \frac{e^{-\|\mathbf{x}-\mathbf{x}_i\|^2/(2\sigma^2)}}{(2\pi\sigma^2)^{n/2}} \, d\mathbf{x} \qquad (11)$$

where $\overline{\mathcal{R}_i} = \mathbb{R}^n \backslash \mathcal{R}_i$ is the complement of the decision region corresponding to the codeword $\mathbf{x}_i$, defined as

$$\mathcal{R}_i = \{\mathbf{z} \in \mathbb{R}^n \ : \ \|\mathbf{z} - \mathbf{x}_i\| < \|\mathbf{z} - \mathbf{x}_j\|, \ \ \forall j \neq i\} \ . \qquad (12)$$

These regions are also known as minimum distance or ML regions.

VI. ML Decoding for fast fading channels

In a fast fading channel the ML estimate $\hat{\mathbf{x}}$ of the transmitted codeword $\mathbf{x}$ can be found by computing

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathcal{S}} ||\mathbf{y} - \mathbf{H}\mathbf{x}||^2 = \arg\min_{\mathbf{x}\in\mathcal{S}} \sum_{i=1}^{n} \lambda_i(x_i) \qquad (13)$$

where

$$\lambda_i(x_i) \triangleq |y_i - h_i x_i|^2 = y_i^2 + h_i^2 x_i^2 - 2y_i h_i x_i \qquad (14)$$

is the contribution of the i-th component to the total metric.

In this paper we restrict our investigation to the case $k = 2$, with $\mathbf{m} = [m_1, m_2]$ and $\boldsymbol{\mu} = [\mu_1, \mu_2]$. We consider both Variant I and II permutation modulation.

A. Variant I permutation modulation

For Variant I modulations $x_i \in \{\mu_1, \mu_2\}$, so we have

$$\lambda_i(x_i) = \begin{cases} y_i^2 + h_i^2\mu_1^2 - 2y_ih_i\mu_1 & \text{if } x_i = \mu_1 \\ y_i^2 + h_i^2\mu_2^2 - 2y_ih_i\mu_2 & \text{if } x_i = \mu_2 \end{cases} \tag{15}$$

If we subtract a constant or we divide by a constant the argument of (13) the result does not change, that is,

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathcal{S}} \sum_{i=1}^{n} \bar{\lambda}_i(x_i) \tag{16}$$

where

$$\bar{\lambda}_i(x_i) \triangleq \frac{\lambda_i(x_i) - \lambda_i(\mu_1)}{\mu_2 - \mu_1} \tag{17}$$

Now we have

$$\bar{\lambda}_i(x_i) = \begin{cases} 0 & \text{if } x_i = \mu_1 \\ h_i^2(\mu_2 + \mu_1) - 2y_ih_i & \text{if } x_i = \mu_2 \end{cases} \tag{18}$$

Since choosing $x_i = \mu_1$ gives no contribution to the total metric, the minimization of (13) is obtained by setting $\hat{x}_i = \mu_2$ for the indexes i that correspond to the m_2 smallest metrics $\bar{\lambda}_i(\mu_2)$. For the remaining m_1 indexes we set $\hat{x}_i = \mu_1$. In practice we find a permutation π that sorts the metrics $\bar{\lambda}_i(\mu_2)$ in non decreasing order, such that $\bar{\lambda}_{\pi_1}(\mu_2) \le \bar{\lambda}_{\pi_2}(\mu_2) \le \ldots \le \bar{\lambda}_{\pi_{n-1}}(\mu_2) \le \bar{\lambda}_{\pi_n}(\mu_2)$ and we set

$$\hat{x}_{\pi_i} = \begin{cases} \mu_2 & \text{for } i = 1, \ldots, m_2 \\ \mu_1 & \text{for } i = m_2 + 1, \ldots, n \end{cases} \tag{19}$$

Theorem 1: The above algorithm produces a ML estimate.

Proof: Suppose that the solution found by the described algorithm does not minimize the metric (13). Then there exists at least a permutation of two indexes i and j that further reduce the metric. If $\bar{\lambda}$ is the total metric given by the algorithm, the new metric becomes $\bar{\lambda}' = \bar{\lambda} - \bar{\lambda}_i(\mu_2) + \bar{\lambda}_j(\mu_2) < \bar{\lambda}$. By solving the inequality we obtain $\bar{\lambda}_i(\mu_2) > \bar{\lambda}_j(\mu_2)$. But since we have chosen $\bar{\lambda}_i(\mu_2) \le \bar{\lambda}_j(\mu_2)$, this contradicts the hypothesis. ■

B. Variant II permutation modulations

The considerations for Variant I modulations hold Variant II. However, in this case, $x_i \in \{\pm\mu_1, \pm\mu_2\}$ and

$$\lambda_i(x_i) = \begin{cases} y_i^2 + h_i^2\mu_1^2 - 2y_ih_i\mu_1 & \text{if } x_i = +\mu_1 \\ y_i^2 + h_i^2\mu_1^2 + 2y_ih_i\mu_1 & \text{if } x_i = -\mu_1 \\ y_i^2 + h_i^2\mu_2^2 - 2y_ih_i\mu_2 & \text{if } x_i = +\mu_2 \\ y_i^2 + h_i^2\mu_2^2 + 2y_ih_i\mu_2 & \text{if } x_i = -\mu_2 \end{cases} \tag{20}$$

Clearly, since $y_i^2 + h_i^2\mu_1^2$ and $y_i^2 + h_i^2\mu_2^2$ are positive, the sign of x_i that minimizes these metrics is

$$\text{sgn}(x_i) = \text{sgn}(y_i). \tag{21}$$

The metric for Variant II modulations then reduces to

$$\lambda_i(x_i) = \begin{cases} y_i^2 + h_i^2\mu_1^2 - 2|y_i|h_i\mu_1 & \text{if } x_i = \text{sgn}(y_i)\mu_1 \\ y_i^2 + h_i^2\mu_2^2 - 2|y_i|h_i\mu_2 & \text{if } x_i = \text{sgn}(y_i)\mu_2 \end{cases} \tag{22}$$

and as in Variant I

$$\bar{\lambda}_i(x_i) = \begin{cases} 0 & \text{if } x_i = \text{sgn}(y_i)\mu_1 \\ h_i^2(\mu_2 + \mu_1) - 2|y_i|h_i & \text{if } x_i = \text{sgn}(y_i)\mu_2 \end{cases} \tag{23}$$

The decoding algorithm procedes as for Variant II using first the metrics (23) and the deciding on the signes using (21).

The pseudo code of the decoding algorithm is given below:

```
Function x̂ = Decoding(y)
  for $i = 1, \ldots, n$
    $\bar{\lambda}_i(\mu_2) = h_i^2(\mu_2 + \mu_1) - 2y_ih_i$ (V. I) or
    $\bar{\lambda}_i(\mu_2) = h_i^2(\mu_2 + \mu_1) - 2|y_i|h_i$ (V. II)
  end
  $\bar{\lambda}_\pi = \text{sort}(\bar{\lambda})$
  for $i = 1, \ldots, m_2$
    $\hat{x}_{\pi_i} = \mu_2$ (V. I) or
    $\hat{x}_{\pi_i} = \text{sgn}(y_{\pi_i})\mu_2$ (V. II)
  end
  for $i = m_2 + 1, \ldots, n$
    $\hat{x}_{\pi_i} = \mu_1$ (V. I) or
    $\hat{x}_{\pi_i} = \text{sgn}(y_{\pi_i})\mu_1$ (V. II)
  end
```

The complexity of this algorithm is O($n \log n$) (i.e the complexity of the sorting function).

VII. The codeword error probability with fast fading

We present here the closed form expression for the codeword error probability, for $k = 2$ and Variant I modulations. We consider an arbitrary permutation of the initial vector $\mathbf{x}_0$ represented by the vector $\mathbf{x}_j = \pi_j(\mathbf{x}_0) = [x_{1,j}, \ldots, x_{n,j}]$ and we denote by $\mathcal{I}_1$ the set of indexes i for which $x_{i,j} = \mu_1$ and by $\mathcal{I}_2$ the set of indexes i for which $x_{i,j} = \mu_2$. Moreover, in the following, we denote by $f_\xi(s)$ and $F_\xi(s)$ the probability density function (pdf) and the cumulative density function (cdf) of the random variable ξ, respectively.

We want to compute the average codeword error probability with respect to the channel, i.e.,

$$\Pr(e) = \frac{1}{M} \sum_{j=0}^{M-1} \mathbb{E}_{\mathbf{h}} [\Pr(e|\mathbf{h}, \mathbf{x}_j)] \tag{24}$$

The following lemma greatly simplifies the evaluation of (24).

Lemma 1: For any permutation code $\mathcal{S} = \{\mathbf{x}_j = \pi_j(\mathbf{x}_0) | \pi_j \in \mathcal{G}\}$ given by the orbit of an initial vector $\mathbf{x}_0$ under a permutation group $\mathcal{G}$ [1],

$$\Pr(e) = \mathbb{E}_{\mathbf{h}} [\Pr(e|\mathbf{h}, \mathbf{x}_0)] \tag{25}$$

if the fading coefficients h_i are mutually independent.

Proof: This lemma implies that it is possible to compute the word error probability by only assuming the initial vector $\mathbf{x}_0$ is transmitted, similarly to the case of group codes for AWGN channel [4].

Let region $\mathcal{R} = \{0 \le h_1 < h_2 < \cdots < h_n\}$ be a subset of the positive orthant $(\mathbb{R}^+)^n$ and S_n be the symmetric group of degree n and order $|S_n| = n!$. We have

$$\bigcup_{\sigma \in S_n} \sigma(\mathcal{R}) = (\mathbb{R}^+)^n \ .$$

where $\sigma(\mathcal{R}) = \{0 \le h_{\sigma(1)} < h_{\sigma(2)} < \cdots < h_{\sigma(n)}\}$ is the region with vector components permuted by σ. The error probability is then

$$\begin{aligned}
\Pr(e) \ &= \int_{\left(\mathbb{R}^+\right)^n} \frac{1}{M} \sum_{j=0}^{M-1} \Pr\left(e|\mathbf{h}, \pi_j(\mathbf{x}_0)\right) f(\mathbf{h}) \, \mathrm{d}\mathbf{h} \\
&= \frac{1}{M} \sum_{j=0}^{M-1} \sum_{k=1}^{n!} \int_{\sigma_k(\mathcal{R})} \Pr\left(e|\mathbf{h}, \pi_j(\mathbf{x}_0)\right) f(\mathbf{h}) \, \mathrm{d}\mathbf{h} \\
&= \frac{1}{M} \sum_{j=0}^{M-1} \sum_{k=1}^{n!} \int_{\mathcal{R}} \Pr\left(e|\sigma_k(\mathbf{h}), \pi_j(\mathbf{x}_0)\right) f(\sigma_k(\mathbf{h})) \, \mathrm{d}\mathbf{h} \\
&= \frac{1}{M} \sum_{j=0}^{M-1} \sum_{k=1}^{n!} \int_{\mathcal{R}} \Pr\left(e|\sigma_k \circ \pi_j(\mathbf{h}), \mathbf{x}_0\right) f(\mathbf{h}) \, \mathrm{d}\mathbf{h} \\
&= \frac{1}{M} \sum_{j=0}^{M-1} \sum_{\ell=1}^{n!} \int_{\mathcal{R}} \Pr\left(e|\sigma_\ell(\mathbf{h}), \mathbf{x}_0\right) f(\mathbf{h}) \, \mathrm{d}\mathbf{h} \\
&= \int_{\left(\mathbb{R}^+\right)^n} \Pr\left(e|\mathbf{h}, \mathbf{x}_0\right) f(\mathbf{h}) \, \mathrm{d}\mathbf{h} = \mathop{\mathbb{E}}_{\mathbf{h}} \left[\Pr\left(e|\mathbf{h}, \mathbf{x}_0\right)\right]
\end{aligned} \tag{26}$$

where we used the following facts: i) the composition of permutations $\sigma_k \circ \pi_j = \sigma_\ell$ runs through S_n since $\mathcal{G}$ is a subgroup of S_n, ii) $f(\mathbf{h}) = \prod f(h_i) = f(\sigma_k(\mathbf{h}))$. ■

As shown in the previous sections, the ML detection algorithm makes decisions using the metrics $\lambda_i = \bar{\lambda}_i(\mu_2)$. The codeword $\mathbf{x}_j$ is successfully decoded if all the metrics associated with the indexes $i \in \mathcal{I}_2$ (i.e where $x_i = \mu_2$) are lower than the metrics with indexes $i \in \mathcal{I}_1$ (i.e. where $x_i = \mu_1$). In other words if we define

$$\zeta \triangleq \min_{i \in \mathcal{I}_1} \lambda_i \quad \text{and} \quad \eta \triangleq \max_{i \in \mathcal{I}_2} \lambda_i \tag{27}$$

the error probability given that the codeword $\mathbf{x}_j$ is transmitted and conditioned to $\mathbf{h}$, is given by

$$\Pr\left(e|\mathbf{h}, \mathbf{x}_j\right) = 1 - \Pr\left(\eta < \zeta|\mathbf{h}, \mathbf{x}_j\right). \tag{28}$$

From (28) and (25) we have

$$\Pr(e) = 1 - \mathop{\mathbb{E}}_{\mathbf{h}} \left[\int_{-\infty}^{+\infty} \Pr\left(\eta < z|\mathbf{h}\right) f_{\zeta|\mathbf{h}}(z) \, \mathrm{d}z\right] \tag{29}$$

where, due to the independence of fading,

$$\Pr\left(\eta < z|\mathbf{h}\right) \ = \ \prod_{i \in \mathcal{I}_2} F_{\lambda_i|h_i}(z) \tag{30}$$

$$f_{\zeta|\mathbf{h}}(z) \ = \ -\frac{\mathrm{d}}{\mathrm{d}z} \prod_{i \in \mathcal{I}_1} \left(1 - F_{\lambda_i|h_i}(z)\right) \tag{31}$$

Then we can write the codeword error probability by using (29) and the metrics cdf $F_{\lambda_i|h_i}$, as

$$\begin{aligned}
\Pr(e) \ &= \ 1 + \int_{-\infty}^{+\infty} \prod_{i \in \mathcal{I}_2} \mathop{\mathbb{E}}_{h_i} \left[F_{\lambda_i|h_i}(z)\right] \cdot \\
&\qquad \frac{\mathrm{d}}{\mathrm{d}z} \prod_{i \in \mathcal{I}_1} \left(1 - \mathop{\mathbb{E}}_{h_i} \left[F_{\lambda_i|h_i}(z)\right]\right) \mathrm{d}z
\end{aligned} \tag{32}$$

where the equality derives from the mutual independence of the h_i's.

For Variant I permutation modulations $\lambda_i = h_i^2(\mu_2 + \mu_1) - 2y_i h_i$. Then, for a given h_i, y_i is a Gaussian random variable with distribution $y_i \sim \mathcal{N}(h_i \mu_i, N_0/2)$, hence the metrics λ_i are also normally distributed as $\lambda_i \sim \mathcal{N}(v_i, 2h_i^2 N_0)$ where, by letting $\delta = \mu_2 - \mu_1$,

$$v_i = \begin{cases} +h_i^2 \delta & \text{for} \quad i \in \mathcal{I}_1 \quad (x_i = \mu_1) \\ -h_i^2 \delta & \text{for} \quad i \in \mathcal{I}_2 \quad (x_i = \mu_2) \end{cases} \tag{33}$$

The average of the cdf of the i-th metric, with respect to the i-th component channel fading h_i is

$$\begin{aligned}
F_{\lambda_i}(z) \ &= \ \mathop{\mathbb{E}}_{h_i} \left[F_{\lambda_i|h_i}(z)\right] \\
&= \int_0^{+\infty} \int_{-\infty}^{z} \frac{f_{h_i}(h)}{2h\sqrt{\pi N_0}} \exp\left(-\frac{(x \pm h^2 \delta)^2}{4 N_0 h^2}\right) \mathrm{d}x \, \mathrm{d}h
\end{aligned} \tag{34}$$

where in the case of Rayleigh fading channel each coefficient h_i is Rayleigh distributed with pdf $f_{h_i}(h) = 2h \exp(-h^2)$, so that $\mathbb{E}[h_i^2] = 1$.

After some algebra we obtain

$$F_{\lambda_i}(z) = \begin{cases} c_1 \exp(b_1 z) & \text{if } z < 0 \\ 1 + c_2 \exp(b_2 z) & \text{if } z \ge 0 \end{cases} \qquad \text{for} \quad i \in \mathcal{I}_1 \tag{35}$$

and

$$F_{\lambda_i}(z) = \begin{cases} -c_2 \exp(-b_2 z) & \text{if } z < 0 \\ 1 - c_1 \exp(-b_1 z) & \text{if } z \ge 0 \end{cases} \qquad \text{for} \quad i \in \mathcal{I}_2 \tag{36}$$

where

$$\begin{aligned}
b_1 \ &= \ \frac{\delta + \sqrt{4N_0 + \delta^2}}{2N_0} \qquad (> 0) \\
b_2 \ &= \ \frac{\delta - \sqrt{4N_0 + \delta^2}}{2N_0} \qquad (< 0) \\
c_1 \ &= \ \frac{1}{b_1 \sqrt{4N_0 + \delta^2}} \qquad (> 0) \\
c_2 \ &= \ \frac{1}{b_2 \sqrt{4N_0 + \delta^2}} \qquad (< 0)
\end{aligned}$$

Finally, by plugging (35) and (36) in (32) the codeword error probability becomes

$$\begin{aligned}
\Pr(e) \ &= \ 1 - \\
&\quad m_1 (-c_2)^{m_2} \sum_{k=0}^{m_1 - 1} \frac{\binom{m_1 - 1}{k} (-c_1)^k}{(k+1)/c_1 - m_2/c_2} - \\
&\quad m_1 (-c_2)^{m_1 - 1} \sum_{k=0}^{m_2} \frac{\binom{m_2}{k} (-c_1)^k}{k/c_1 - m_1/c_2}
\end{aligned} \tag{37}$$

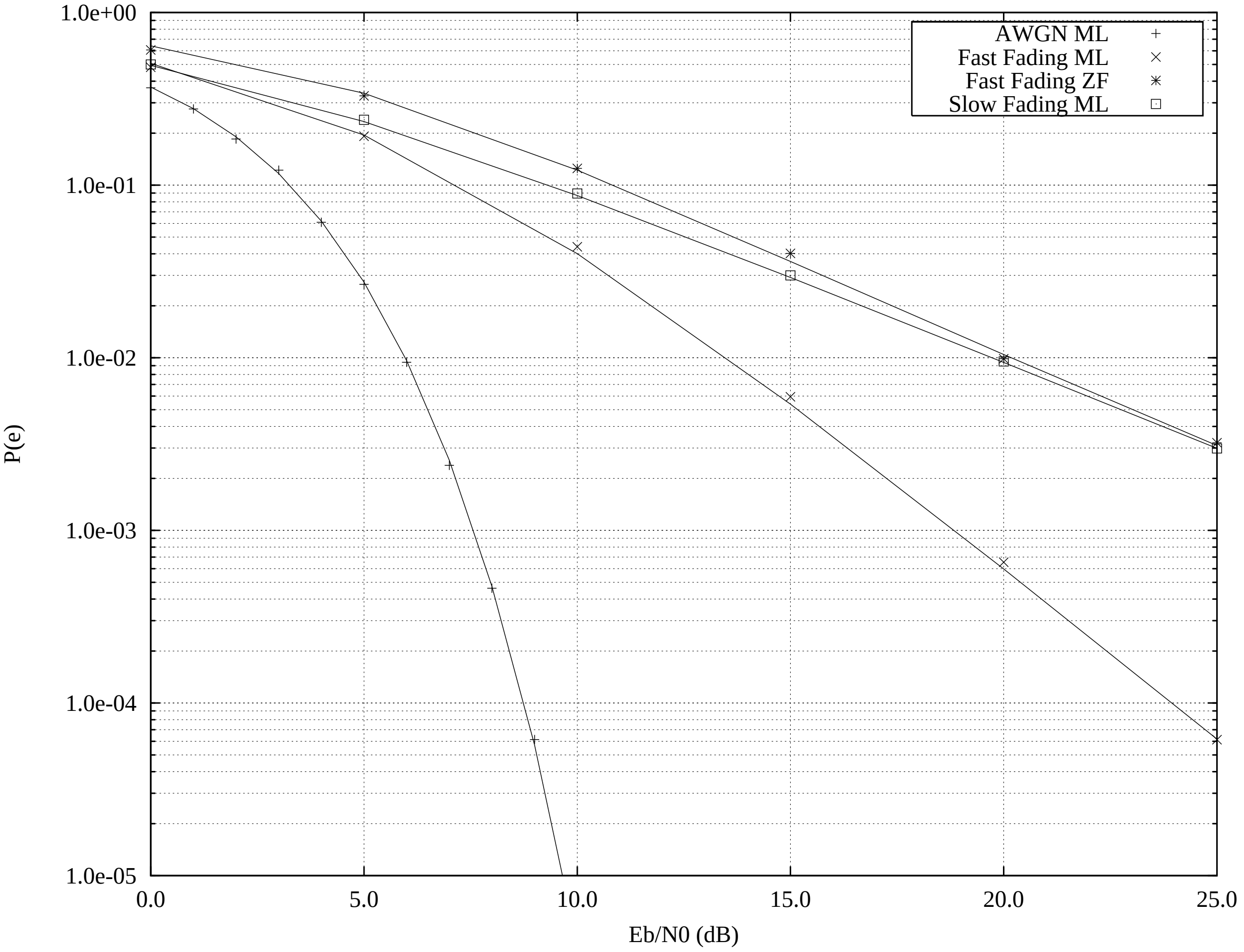


Fig. 2. Performance of the Variant I permutation modulation with $k = 2$, $n = 9$, $\mathbf{m} = [7, 2]$, $\alpha = 0$

It is convenient to write the coefficients c_1 and c_2 as a function of the signal to noise ratio E_s/N_0, where $E_s = m_1\mu_1^2 + m_2\mu_2^2$. In fact, if we set $\mu_1 = \alpha\mu_2$, with $0 \le \alpha < 1$ a system parameter, we obtain

$$c_1 = \frac{1}{2}\left(1 - \sqrt{\frac{\beta\gamma}{4 + \beta\gamma}}\right), \quad c_2 = c_1 - 1, \tag{38}$$

where

$$\beta = (1-\alpha)^2/(m_1\alpha^2 + m_2), \;\; \gamma = \frac{E_s}{N_0} \tag{39}$$

A few comments are appropriate here about the choice of the code parameters. We first note that $\Pr(e)$ is a decreasing function of $\beta\gamma$, hence it is convenient to minimize β. For fixed m_1, m_2 the best choice is $\alpha = 0$, which implies $\mu_1 = 0$. For a given α and a given $M = n!/(m_1!m_2!)$, the best choice for m_1, m_2 is when $m_1 > m_2$. This fact implies that codes having codewords with higher peak to average power ratio perform better in fast fading.

VIII. A Zero Forcing decoder for fast fading

For a fast fading channel the ZF approach clearly represents a suboptimal and simpler solution compared to the ML algorithm. In order to provide a comparison with the ML performance, in this section we derive the analytical expression of its $P(e)$.

The ZF approach inverts the channel by computing

$$\mathbf{z} = \mathbf{H}^{-1}\mathbf{y} = \mathbf{x} + \mathbf{H}^{-1}\mathbf{n}$$

where, for a given h_i, the random variable $z_i = y_i/h_i$ is normally distributed with mean x_i and variance $N_0/(2h_i^2)$. The estimate of the transmitted codeword is given by minimizing

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathcal{S}} \|\mathbf{z} - \mathbf{x}\|^2$$

This problem can be solved by applying the decoding algorithm for AWGN channel described in [3] that provides the codeword error probability

$$\begin{aligned}\Pr(e) &= 1 + \int_{-\infty}^{+\infty} \prod_{i\in\mathcal{I}_1} \mathop{\mathbb{E}}_{h_i}\left[F_{z_i|h_i}(z)\right] \cdot \\ &\qquad \frac{\mathrm{d}}{\mathrm{d}z} \prod_{i\in\mathcal{I}_2}\left(1 - \mathop{\mathbb{E}}_{h_i}\left[F_{z_i}(z)\right]\right) \mathrm{d}z \\ &= 1 + \int_{-\infty}^{+\infty} t\left(z + \sqrt{\gamma\beta}\right)^{m_1} \frac{\mathrm{d}}{\mathrm{d}z}\left(1 - t(z)\right)^{m_2} \mathrm{d}z\end{aligned} \tag{40}$$

where $t(z) = \frac{1}{2}\left(1 + \frac{z}{\sqrt{1+z^2}}\right)$, and β and γ are defined in (39). In terms of complexity the ZF algorithm differs from the ML algorithm only in the metric computation. In both cases, however, the complexity remains proportional to $O(n \log n)$.

IX. Error probability for slow fading channel

In the presence of slow fading channel the fading coefficient h remains constant within the codeword, that is $\mathbf{y} = h\mathbf{x} + \mathbf{n}$ where $y_i \sim \mathcal{N}(hxi, N_0/2)$. The probability of error is then given by

$$\begin{aligned}
\Pr(e) &= 1 \\
&+ \int_{-\infty}^{+\infty} \mathop{\mathbb{E}}_{h}\left[\prod_{i\in\mathcal{I}_1} F_{y_i|h}(z) \frac{\mathrm{d}}{\mathrm{d}z} \prod_{i\in\mathcal{I}_2} \left(1 - F_{y_i|h}(z)\right) \mathrm{d}z\right] \\
&= 1 - \frac{2m_2}{\sqrt{\pi}} \int_{-\infty}^{+\infty} \mathrm{e}^{-z^2} (1 - t(z))^{m_2 - 1} \cdot \\
&\quad \cdot \int_{0}^{+\infty} h\mathrm{e}^{-h^2} t\left(z + h\sqrt{\gamma\beta}\right)^{m_1} \mathrm{d}h\, \mathrm{d}z \qquad (41)
\end{aligned}$$

where $t(z) = \frac{1}{2}(1 + \mathrm{erf}(z))$.

X. Simulation results

Figure 2 shows the performance of the Variant I permutation modulation with parameters $n = 9$, $k = 2$, $\mathbf{m} = [7, 2]$ and $\alpha = 0$, for various channels. The solid lines are obtained by the exact computation of (37) in the ML fast fading case, and by numerical integration of (40) and (41) for the ZF fast fading and slow fading cases, respectively. The points superimposed to the solid lines have been obtained by simulation. The performance of the AWGN case is also shown. Notice that the ML algorithm for fast fading channel provides a non negligible gain with respect to the ZF approach even at very low signal-to-noise ratio. The performance for the slow fading channel is similar to the ZF fast fading case. We observe that permutation modulations exhibit a modulation diversity $L = 2$, since the minimum Hamming distance between any two codewords is 2, when they are obtained by any two component exchange [5]. With fast fading the full diversity can be exploited only if ML decoding is performed.

XI. Conclusions

In this paper we have presented some new results on permutation modulations for transmission over fading channels. In the case $k = 2$ we have devised an efficient ML decoder of Variant I and II permutation modulations in fast fading. An exact expression for the codeword error probability of Variant I permutation modulations for independent Rayleigh fading channels is given. A similar result is expected for Variant II permutation modulations.

Future work will concentrate on extending these results to permutation modulations with $k > 2$.

References

[1] J. K. Karlof, “Permutation codes for the Gaussian channel”, *IEEE Transactions on Information Theory*, n. 4, vol. 35, July 1989, pp. 726–732.

[2] B. M. King, M. A. Neifeld, “Low-complexity maximum-likelihood decoding of shortened enumerative permutation codes for holographic storage”, *IEEE Journal on Selected Areas in Communications*, n. 4, vol. 19, April 2001, pp. 783–790.

[3] D. Slepian, “Permutation modulation”, *Proceedings of the IEEE*, March 1965, pp. 228–236.

[4] D. Slepian, “Group codes for the Gaussian channel”, *Bell System Technical Journal*, n. 4, vol. 47, April 1968, pp. 575–602.

[5] J. Boutros and E. Viterbo, “Signal Space Diversity: a power and bandwidth efficient diversity technique for the Rayleigh fading channel”, *IEEE Transactions on Information Theory*, vol. 44, n. 4, pp. 1453–1467, July 1998.