Trellis Decoding of Permutation Modulations

Sḿinaire de Théorie des Nombres, Algorithmique et Cryptographie

Emanuele Viterbo

Dipartimento di Elettronica

Politecnico di Torino, Italy

`viterbo@polito.it`

Slides available at:

`http://www.tlc.polito.it/~viterbo/slides.html`

Introduction

- Permutation modulations are particular spherical codes proposed by David Slepian in 1965.
- A Variant I permutation modulation is the set of codewords obtained by taking all permutations of the components 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.
- Efficiently decoded in the AWGN channel and block fading channel by essentially applying a sorting algorithm to the received signal vector.
- In this talk we present a new trellis based decoder for ML detection over fast fading channels.

The codebook

- Digital transmission over the band-limited channels is commonly modeled in the n-dimensional Euclidean space $\mathbf{R}^n$.
- The *codebook* or *signal constellation* is a set of M points (vectors) in $\mathbf{R}^n$

$$\mathcal{S} = \{\mathbf{x}_i\}_1^M$$

- A codeword $\mathbf{x} = (x_1, \ldots, x_n) \in \mathcal{S}$ conveniently represents a band-limited signal to be transmitted by the source.
- The source labels each point with $\log_2(M)$ bits.

Channel Models

- Different types of channels may be described using the following model

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

- $\mathbf{y} = (y_1, \ldots, y_n) \in \mathbf{R}^n$ received signal vector
- $\mathbf{x} = (x_1, \ldots, x_n)$ transmitted codeword taken from a codebook $\mathcal{S}$
- $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \frac{N_0}{2}\mathbf{I})$ real Gaussian random vector with i.i.d. components
- $\mathbf{H} = \mathrm{diag}(h_1, \ldots, h_n)$ diagonal channel fading matrix
  1. *AWGN*: $\mathbf{H} = \mathbf{I}$
  2. *Slow (or block) fading*: $\mathbf{H} = h\mathbf{I}$ and h changes independently from one codeword to the next
  3. *Fast fading*: h_i change independently in each component

ML decoding

- Assuming that the channel state information is available at the receiver, ML detection is given by
$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathcal{S}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 = \arg\min_{\mathbf{x}'\in\mathcal{S}'} \|\mathbf{y} - \mathbf{x}'\|^2$$
where $\mathcal{S}' = \mathbf{H}\mathcal{S}$ is the "faded signal constellation".

- The complexity of the ML receiver greatly depends on the structure of the code $\mathcal{S}$.

- In the case of AWGN and slow fading the error probability is asymptotically dependent on the *minimum Euclidean distance* of the signal set.

- In the case of fast Rayleigh fading the performance of the signal set is dominated by the minimum Hamming distance between any two signal vectors, also known as *modulation diversity*, denoted by L.

Permutation Modulation

- $\{\mu_1, \ldots, \mu_k\}$ distinct real numbers with $0 \leq \mu_1 < \mu_2 < \cdots < \mu_k$.
- $\{m_1, \ldots, m_k\}$ positive integers such that $n = \sum_{j=1}^{k} m_j$.
- Initial vector with components sorted in ascending order

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

- A Variant I permutation modulation consists of the set $\mathcal{S}$ of vectors obtained by permuting the components of the initial vector $\mathbf{x}_0$. Total number of codewords:

$$M_{\mathrm{I}} = \frac{n!}{m_1! \, m_2! \cdots m_k!} \; .$$

- A Variant II permutation modulation consists of the set $\mathcal{S}$ of vectors obtained by permuting and applying all possible sign changes to the components of the initial vector $\mathbf{x}_0$. Total number of codewords:

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

where $h = n - m_1$, if $\mu_1 = 0$ and $h = n$, if $\mu_1 > 0$.

ML decoding PM for AWGN

- Let us consider ML decoding of Variant I codes in the AWGN case. We need to find the minimum of

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

  for $i = 0, \ldots, M-1$ where $(\mathbf{y},\mathbf{x}_i)$ denotes the scalar product of the two vectors and

$$\mathcal{P} = \frac{1}{n}\sum_{j=1}^{k} m_j \mu_j^2.$$

- It is equivalent to

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

- 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.

The trellis of permutation codes

- The trellis of the permutation code is a multidimensional finite length trellis.

$$\mathcal{T} = \{(s_1, \ldots, s_k) \in \mathbf{Z}^k \mid 0 \leq s_1 \leq m_1, \cdots, 0 \leq s_k \leq m_k\}$$

- A point in this set corresponds to a *state* in the trellis and will be labelled by the integer vector $(s_1, \ldots, s_k)$. There are a total of $\prod_{l=1}^{k}(m_l + 1)$ states throughout the trellis. We define the *initial state* as the one labelled by $(0, \ldots, 0)$ and the *final state* as the one labelled by $(m_1, \ldots, m_k)$.

- The trellis *branches* are all the segments of length one connecting the states. We can see this trellis as a multidimensional grid. We label with μ_l all trellis branches connecting state $(s_1, \ldots, s_l, \ldots, s_k)$ to state $(s_1, \ldots, s_l + 1, \ldots, s_k)$.

- The codewords correspond to paths on this trellis that stem from the initial state and end into the final state, without ever going backwards.

- In order to define the *trellis sections* we define the following state subsets:

$$\mathcal{T}^{(K)} = \{(s_1, \ldots, s_k) \in \mathcal{T} \mid \sum_{l=1}^{k} s_l = K\} \qquad K = 0, \ldots, n$$

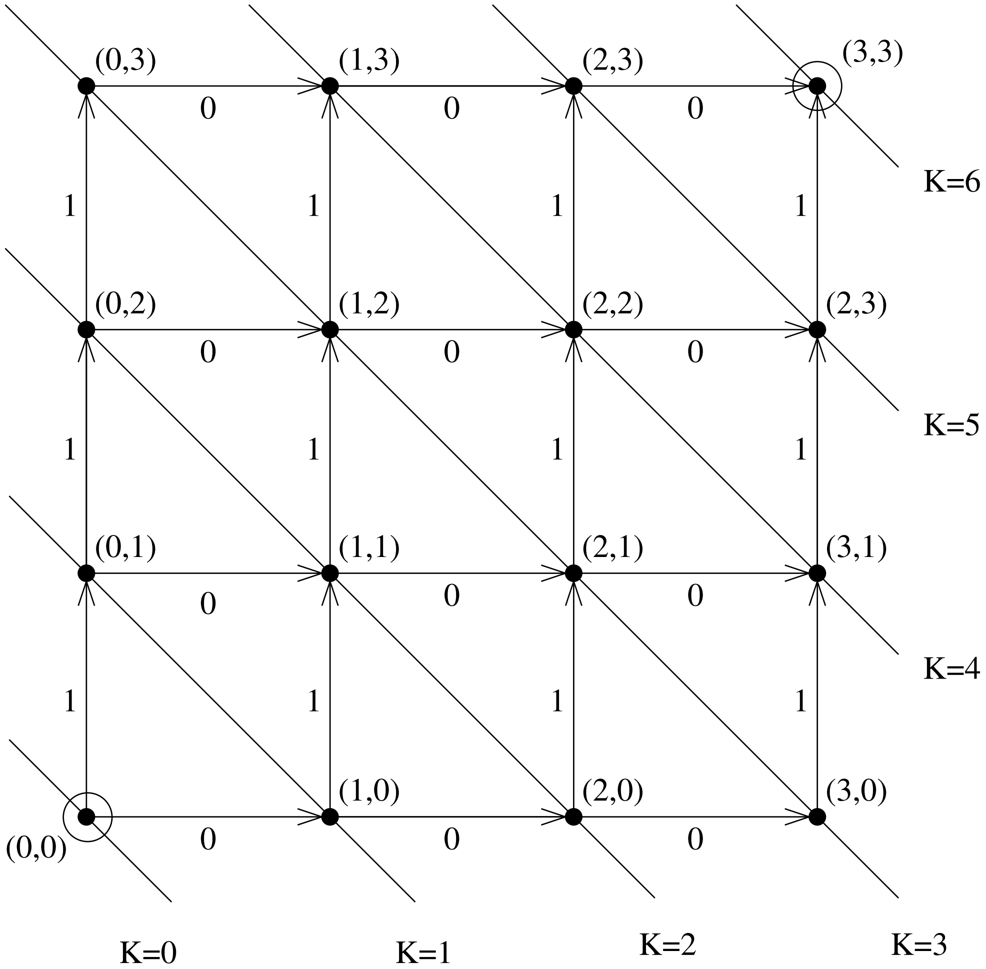
Examples

- Let $k = 2$, $m_1 = 3$ and $m_2 = 3$, $\mu_1 = 0$ and $\mu_2 = 1$, then $n = 6$ and $M = 20$. It can be seen that the number of states in each section are $|\mathcal{T}^{(0)}| = 1$, $|\mathcal{T}^{(1)}| = 2$, $|\mathcal{T}^{(2)}| = 3$, $|\mathcal{T}^{(3)}| = 4$, $|\mathcal{T}^{(4)}| = 3$, $|\mathcal{T}^{(5)}| = 2$, $|\mathcal{T}^{(6)}| = 1$.

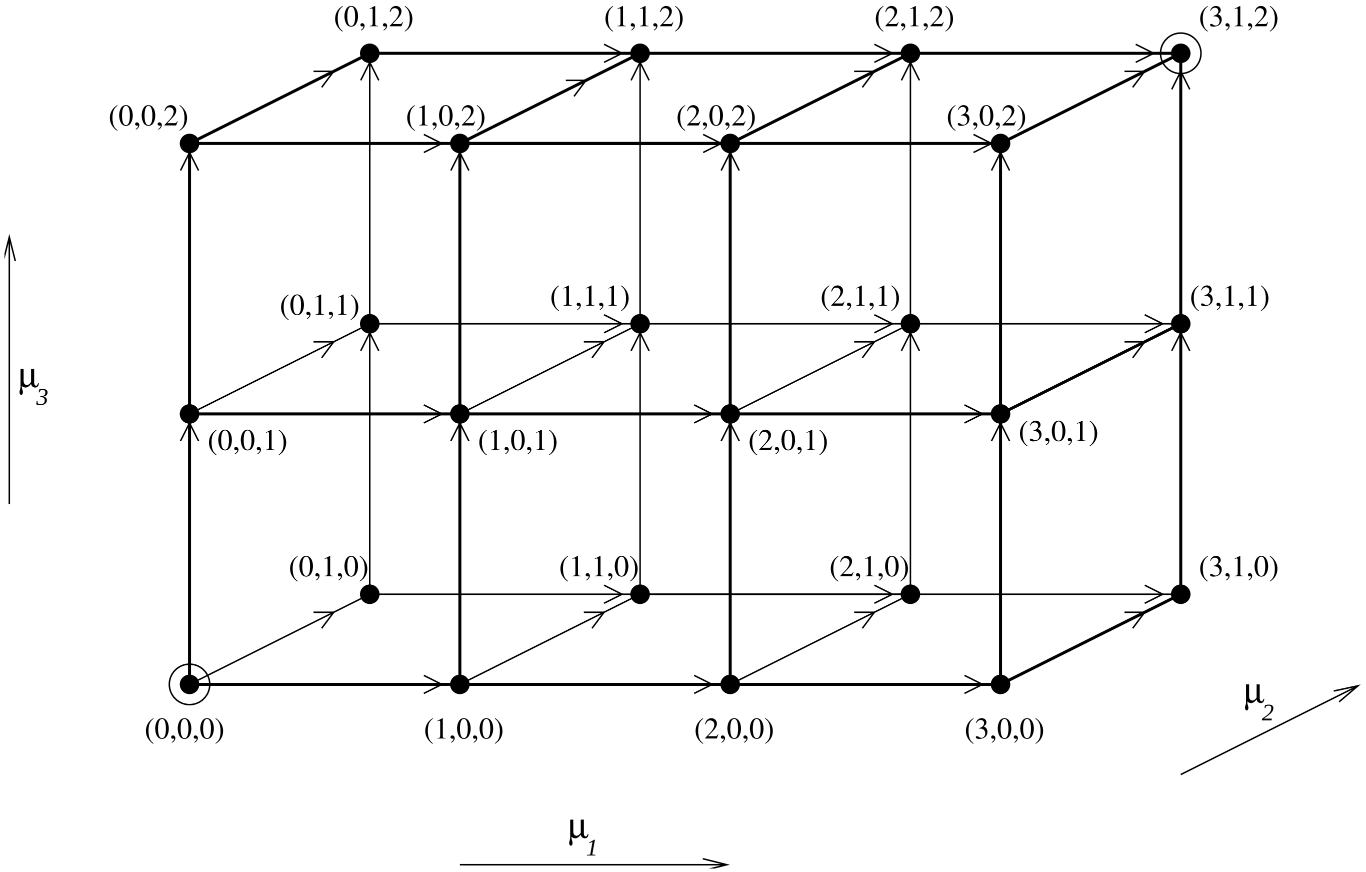
- Let $k = 3$, $m_1 = 3$, $m_2 = 1$, $m_3 = 2$ and the initial vector $\mathbf{x}_0 = (\mu_1, \mu_1, \mu_1, \mu_2, \mu_3, \mu_3)$, then $M = 60$. The total number of states is 24 and we have the following trellis sections

$$
\begin{aligned}
\mathcal{T}^{(0)} &= \{(0,0,0)\} \\
\mathcal{T}^{(1)} &= \{(0,0,1),(0,1,0),(1,0,0)\} \\
\mathcal{T}^{(2)} &= \{(0,0,2),(0,1,1),(1,0,1),(1,1,0),(2,0,0)\} \\
\mathcal{T}^{(3)} &= \{(0,1,2),(1,0,2),(1,1,1),(2,0,1),(2,1,0),(3,0,0)\} \\
\mathcal{T}^{(4)} &= \{(1,1,2),(2,0,2),(2,1,1),(3,0,1),(3,1,0)\} \\
\mathcal{T}^{(5)} &= \{(2,1,2),(3,0,2),(3,1,1)\} \\
\mathcal{T}^{(6)} &= \{(3,1,2)\}
\end{aligned}
$$

Example of 2D trellis



Example of 3D trellis



Viterbi decoding of grid codes

- Given the received word $\mathbf{y}$, we compute the branch metrics $\lambda_l = (h_l \mu_l - y_l)^2$ and use them to label to branches pointing along the l-th dimension for $l = 1, \dots, k$.

- The states in each section can be easily enumerated of the fly for each new section in the forward operation.

- For the add/compare/select operation in a given state $(s_1, \dots, s_k)$, the incoming branches are identified by subtracting 1 to each component of the state label, and verifying if it remains positive.

- It is convenient to store the path memory for the traceback information in a k dimensional array of sizes $(m_l + 1)$ for $l = 1, \dots k$ although this can be done efficiently using a one-dimensional vector of length $\prod_{l=1}^{k} (m_l + 1)$.

Conclusions an future work

- Trellis decoder for permutation modulation.
- Using the Viterbi algorithm it is possible to efficiently perform ML detection also in the presence of fast fading.
- Extend the trellis decoding to more general permutation codes.
- We can define a very large family of codes *grid codes* that have a multidimensional trellis representation.