

6

Códigos de Permutação para Modulação

Neste capítulo, códigos de permutação escalar e vetorial para transmissão em um canal gaussiano são apresentados. A modulação utilizando códigos de permutação escalar (SPM) foi introduzida por Slepian em 1965 [34] e subseqüentemente desenvolvida por Biglieri e Elia [5], Karloff [21] e Ingemarsson (1990) [20]. Diversas aplicações desse sistema têm sido propostas, como, por exemplo, gravação magnética [38], transmissão em canais com desvanecimento (2003) [28, 32, 37] e armazenamento óptico [23].

Neste trabalho, introduzimos uma nova modulação baseada em códigos de permutação vetorial (VPM). Assim como no contexto de compressão de fontes, códigos de permutação vetorial são obtidos a partir dos códigos de permutação escalar agrupando-se símbolos escalares em um vetor L -dimensional. Nossa definição no contexto de modulação é motivada pelo fato de que, quando usados para compressão, estes códigos possuem desempenho superior ao de sua versão escalar (*cf.* Capítulo 4). Além disso, é um fato conhecido que as modulações multidimensionais atingem melhores desempenhos que as unidimensionais; em um espaço de L -dimensões, constelações cujos símbolos podem variar livremente preenchem mais eficientemente o espaço do que a grade retangular de pontos resultante do produto cartesiano de L constelações unidimensionais.

Este capítulo está organizado da seguinte forma. A Seção 6.1 apresenta a definição VPM, a qual inclui a definição de SPM como caso particular. Na Seção 6.2, é discutida a decodificação ótima de SPM e VPM em canal AWGN e em canais com imperfeições. O projeto de SPM e VPM é abordado na Seção 6.3, na qual é apresentada uma generalização para o projeto de vetor de composição e discutido o projeto de alfabeto através de reticulados. Por fim, nossos resultados relativos ao desempenho de VPM em comparação com SPM são exibidos na Seção 6.4.

6.1

Definição

Seja $\{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \dots, \boldsymbol{\mu}_K\}$ um conjunto de vetores L -dimensionais reais distintos e seja $\{m_1, \dots, m_K\}$ um conjunto de inteiros positivos tais que

$$n = \sum_{k=1}^K m_k. \quad (6-1)$$

Considere a seguinte *palavra de referência*, de comprimento n , formada por m_k repetições de cada símbolo $\boldsymbol{\mu}_k$ em ordem crescente de índice:

$$\underline{\mathbf{x}}_1 = \left(\underbrace{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_1}_{m_1}, \underbrace{\boldsymbol{\mu}_2, \dots, \boldsymbol{\mu}_2}_{m_2}, \dots, \underbrace{\boldsymbol{\mu}_K, \dots, \boldsymbol{\mu}_K}_{m_K} \right)^T. \quad (6-2)$$

Um *código de permutação vetorial* é o conjunto $\mathcal{C} = \{\mathbf{x}_i\}_{i=1}^M$ formado pelas palavras obtidas quando se permuta os componentes de $\mathbf{x}_1 = (\mathbf{x}_{11}, \dots, \mathbf{x}_{1n})^T$ de todas as maneiras possíveis. Os conjuntos $\{\boldsymbol{\mu}_k\}_{k=1}^K$ e $\{n_k\}_{k=1}^K$ são chamados, respectivamente, de *alfabeto* e *vetor de composição*. No caso particular $L = 1$, este código é chamado de *código de permutação escalar*. Quando usados para modulação, códigos de permutação escalar e vetorial são denotados, respectivamente, por SPM (*scalar permutation modulation*) e VPM (*vector permutation modulation*).

Note que uma palavra de um VPM tem um total de $N = nL$ dimensões. A taxa de informação do código, em bits por dimensão, é portanto

$$R = \frac{1}{nL} \log M \quad (6-3)$$

onde o número M de palavras do código é dado por

$$M = \frac{n!}{\prod_{k=1}^K m_k!}. \quad (6-4)$$

Como todas palavras têm a mesma norma, a energia média por dimensão é

$$E = \frac{1}{nL} \|\mathbf{x}_1\|^2 = \frac{1}{nL} \sum_{k=1}^K m_k \|\boldsymbol{\mu}_k\|^2. \quad (6-5)$$

Por sua vez, a distância mínima entre as palavras é dada por

$$d_{\min} = \min_{i,j, i \neq j} \|\mathbf{x}_i - \mathbf{x}_j\| = \min_{k,\ell, k \neq \ell} \sqrt{2} \|\boldsymbol{\mu}_k - \boldsymbol{\mu}_\ell\| \quad (6-6)$$

uma vez que as duas palavras mais próximas só diferem em dois componen-

tes (transposição de símbolos). A distância mínima entre as palavras de um VPM pode também ser escrita como $d_{\min} = \sqrt{2} d_{\min}^*$, onde

$$d_{\min}^* = \min_{k, \ell, k \neq \ell} \|\boldsymbol{\mu}_k - \boldsymbol{\mu}_\ell\| \quad (6-7)$$

é a distância mínima entre os símbolos do alfabeto.

6.2

Detecção Ótima

6.2.1

Canal AWGN

A detecção de máxima verossimilhança em canal AWGN é realizada buscando-se a palavra $\hat{\mathbf{x}} \in \mathcal{C}$ que está à menor distância euclidiana da palavra recebida $\underline{\mathbf{y}} = (\mathbf{y}_1 \dots \mathbf{y}_n)$, onde $\mathbf{y}_j \in \mathbb{R}^L$, isto é, buscando-se

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x} \in \mathcal{C}} \|\underline{\mathbf{y}} - \mathbf{x}\|^2. \quad (6-8)$$

Por definição, todas as palavras $\underline{\mathbf{x}}_i$ de um código de permutação vetorial \mathcal{C} estão relacionadas à palavra de referência $\underline{\mathbf{x}}_1$ através de uma permutação. Reescrevendo (6-8) para explicitar essa relação, obtemos que o receptor ótimo estima a palavra transmitida $\underline{\mathbf{x}}_i$ por

$$\hat{\mathbf{x}} = (\mathbf{x}_{1, \hat{\pi}(1)}, \dots, \mathbf{x}_{1, \hat{\pi}(n)}) \quad (6-9)$$

onde $\hat{\pi}$ é uma função de permutação em Π_n dada por

$$\hat{\pi} = \arg \min_{\pi \in \Pi_n} \frac{1}{n} \sum_{j=1}^n \|\mathbf{y}_j - \mathbf{x}_{1, \pi(j)}\|^2. \quad (6-10)$$

De forma análoga ao desenvolvimento da Seção 4.2, podemos considerar que $\delta_{j, \ell} = \|\mathbf{y}_j - \mathbf{x}_{1, \ell}\|^2$ corresponde ao “custo” de se atribuir o vetor \mathbf{y}_j ao vetor $\mathbf{x}_{1, \ell}$. Assim, a equação (6-10) se torna

$$\hat{\pi} = \arg \min_{\pi \in \Pi_n} \frac{1}{n} \sum_{j=1}^n \delta_{j, \pi(j)} \quad (6-11)$$

isto é, a tarefa do receptor ótimo é encontrar a permutação π tal que o custo total $\sum_{j=1}^n \delta_{j, \pi(j)}$ seja minimizado. O problema da detecção ótima de VPM se reduz, portanto, a um *problema de atribuição* (cf. Seção 4.2), o qual pode

ser resolvido com complexidade $O(n^2\sqrt{n}\log n)$ através do algoritmo CSA. No caso particular de SPM ($L = 1$), o problema se reduz ao de uma simples ordenação de números reais, cuja complexidade é $O(n\log n)$ (cf. Seção 3.2).

6.2.2

Canal com imperfeições

Em um espaço de $N = nL$ dimensões, a detecção de máxima verossimilhança em canal com imperfeições pode ser escrita como

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x} \in \mathcal{C}} \|\underline{\mathbf{y}} - H\mathbf{x}\|^2 \quad (6-12)$$

onde a matriz H é dada por

$$H = \begin{pmatrix} \mathbf{h}_{11} & \mathbf{h}_{12} & \cdots & \mathbf{h}_{1n} \\ \mathbf{h}_{21} & \mathbf{h}_{22} & \cdots & \mathbf{h}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{h}_{n1} & \mathbf{h}_{n2} & \cdots & \mathbf{h}_{nn} \end{pmatrix} \quad (6-13)$$

e cada $\mathbf{h}_{j\ell}$ é uma matriz real $L \times L$. A detecção ótima em um canal com imperfeições genérico é um problema difícil, porque, a princípio, envolve o teste de todas as palavras de transmissão.

Suponha agora um caso particular em que o canal possui a seguinte matriz de imperfeições:

$$H = \begin{pmatrix} \mathbf{h}_1 & 0 & \cdots & 0 \\ 0 & \mathbf{h}_2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \mathbf{h}_n \end{pmatrix}. \quad (6-14)$$

Situações de interesse que podem ser caracterizadas dessa forma incluem sistemas que utilizam funções de base que não se sobrepõem no tempo (cf. seções 5.1.1 e 5.3) operando em canais com desvanecimento (lento ou rápido).

Aplicando (6-14) em (6-12), podemos reescrever a expressão para a detecção ótima de VPM como

$$\hat{\mathbf{x}} = (\mathbf{x}_{1,\hat{\pi}(1)}, \dots, \mathbf{x}_{1,\hat{\pi}(n)}) \quad (6-15)$$

onde

$$\hat{\pi} = \arg \min_{\pi \in \Pi_n} \frac{1}{n} \sum_{j=1}^n \|\mathbf{y}_j - \mathbf{h}_j \mathbf{x}_{1,\pi(j)}\|^2. \quad (6-16)$$

Assim, vemos que o mesmo método descrito acima para a recepção de VPM em canal AWGN pode ser aplicado na solução desse problema, bastando que seja feita uma pequena modificação no cálculo dos custos a fim de incorporar os coeficientes \mathbf{h}_j :

$$\delta_{j\ell} = \|\mathbf{y}_j - \mathbf{h}_j \mathbf{x}_{1\ell}\|^2. \quad (6-17)$$

A complexidade dessa operação é idêntica à da recepção em canal AWGN, ou seja, $O(n^2 \sqrt{n} \log n)$.

No caso particular da recepção de SPM, outras soluções existem para a situação em que a matriz H é diagonal. Em canais com desvanecimento lento, a recepção de SPM pode ser realizada pelo mesmo algoritmo de ordenação usado para recepção em canal AWGN, uma vez que a constelação “desvanecida” gerada por $H\mathbf{x}$ em (5-7) continua sendo um código de permutação.

O mesmo não acontece, entretanto, no caso de desvanecimento rápido. Em [28], Nordio e Viterbo abordam esse problema, apresentando um algoritmo eficiente para o caso particular $K = 2$. Viterbo elimina essa restrição em [37], desenvolvendo uma descrição de SPM através de uma estrutura de treliça, sobre a qual o algoritmo de Viterbi é aplicado. Aparentemente, esse é o único método proposto até o momento para o problema de desvanecimento rápido sem restrições em K . Entretanto, uma análise simples revela que a complexidade desse método é $O(\prod_{i=1}^K m_i) = O(\prod_{i=1}^K p_i n) = O(n^K)$, a qual rapidamente se torna proibitiva com o aumento de K . Portanto, em canais com desvanecimento rápido, uma alternativa mais eficiente para SPM com $K \geq 3$ é realizar a recepção sob a estrutura de VPM.

6.3

Projeto de SPM e VPM

Dados a dimensão de alfabeto L , o tamanho de bloco n e o número de componentes K , os parâmetros que caracterizam um VPM são seu alfabeto $\{\mu_k\}_{k=1}^K$ e seu respectivo vetor de composição $\{m_k\}_{k=1}^K$. Nesta seção, procuramos projetar esses parâmetros de forma a otimizar as grandezas de interesse na comparação entre códigos, as quais são a taxa, a energia e a distância mínima.

Conforme discutido na Seção 5.2, o compromisso ótimo entre essas

grandezas corresponde a maximizar a taxa R (a qual determina a eficiência espectral) e minimizar a razão E/d_{\min}^2 (a qual, para R fixo, determina o desempenho); isto é, deseja-se otimizar o lagrangiano envolvendo essas duas quantidades

$$J_\lambda(\{m_k\}, \{\boldsymbol{\mu}_k\}) = -R(\{m_k\}) + \lambda \frac{E(\{m_k\}, \{\boldsymbol{\mu}_k\})}{d_{\min}^2(\{\boldsymbol{\mu}_k\})}. \quad (6-18)$$

Note que, no caso dos códigos de permutação, o vetor de composição influencia tanto R quanto E , mas não d_{\min} , enquanto o alfabeto tem impacto apenas em E e d_{\min} , mas não em R .

Assim, dividimos o projeto em duas partes: a primeira consiste em fixar o alfabeto — e conseqüentemente a distância mínima — e buscar os valores ótimos para o vetor de composição, isto é, aqueles que otimizam o compromisso entre taxa e energia. A segunda etapa consiste em fixar o vetor de composição — e portanto a taxa — e buscar os valores do alfabeto que, para uma dada distância mínima, minimizem a energia. A combinação dessas duas etapas leva a VPM's que possuem uma boa relação entre taxa R e razão E/d_{\min}^2 (ou seja, entre eficiência espectral e desempenho).

6.3.1 Projeto de vetor de composição

O desenvolvimento a seguir é uma generalização do desenvolvimento para SPM obtido por Ingemarson [20]. Dados L , K e o alfabeto $\{\boldsymbol{\mu}_k\}_{k=1}^K$, desejamos encontrar o vetor de composição $\{m_k\}_{k=1}^K$ que maximiza R e minimiza E sob uma restrição em n .

Temos

$$E = \frac{1}{nL} \sum_{k=1}^K m_k \|\boldsymbol{\mu}_k\|^2 \quad n = \sum_{k=1}^K m_k \quad (6-19)$$

$$R = \frac{1}{nL} \log M = \frac{1}{nL} \left[\sum_{\ell=1}^n \log \ell - \sum_{k=1}^K \sum_{\ell=1}^{m_k} \log \ell \right] \quad (6-20)$$

Para minimizar $-R$ e E sob uma restrição em n , formamos o lagrangiano

$$f(m_1, \dots, m_K) = \sum_{k=1}^K \sum_{\ell=1}^{m_k} \log \ell + \lambda \sum_{k=1}^K m_k \|\boldsymbol{\mu}_k\|^2 + \nu \left(\sum_{k=1}^K m_k - n \right) \quad (6-21)$$

onde $\lambda \geq 0$. Note que as constantes em (6-19) e (6-20) foram omitidas da

expressão acima; em particular, o valor de n é garantido constante através da escolha de ν . Quando m_k cresce, a função $f(\cdot)$ está passando por um mínimo quando as seguintes inequações são satisfeitas:

$$f(m_1, \dots, m_k + 1, \dots, m_K) - f(m_1, \dots, m_k, \dots, m_K) > 0 \quad (6-22)$$

$$f(m_1, \dots, m_k - 1, \dots, m_K) - f(m_1, \dots, m_k, \dots, m_K) \geq 0 \quad (6-23)$$

Substituindo (6-21), temos

$$\log(m_k + 1) + \lambda \|\boldsymbol{\mu}_k\|^2 + \nu > 0 \quad (6-24)$$

$$\log(m_k) + \lambda \|\boldsymbol{\mu}_k\|^2 + \nu \leq 0 \quad (6-25)$$

Combinando (6-24) e (6-25), temos que os valores ótimos para m_k são

$$m_k = \left\lfloor 2^{-\lambda \|\boldsymbol{\mu}_k\|^2 - \nu} \right\rfloor = \left\lfloor \beta \cdot 2^{-\lambda \|\boldsymbol{\mu}_k\|^2} \right\rfloor, \quad k = 1, \dots, K \quad (6-26)$$

onde $\lfloor x \rfloor$ denota a parte inteira de x . O valor $\beta > 0$ é ajustado para satisfazer a restrição em n , enquanto λ é ajustado de forma que uma energia média E (ou, equivalentemente, uma taxa R) seja obtida. Conclui-se, portanto, que o vetor de composição ótimo é dado pelos valores arredondados de uma distribuição de Maxwell–Boltzmann (*cf.* [24]).

6.3.2

Projeto de alfabeto

O projeto de alfabeto consiste em, dados L , n , K e qualquer vetor de composição $\{m_k\}_{k=1}^K$, encontrar o alfabeto $\{\boldsymbol{\mu}_k\}_{k=1}^K$ que minimiza a energia E para uma dada distância mínima entre palavras d_{\min} .

Este problema está resolvido para SPM: em [5], Biglieri e Elia mostram que, *independentemente* do vetor de composição, os valores ótimos para μ_k estão igualmente espaçados:

$$\mu_k = k - \frac{K + 1}{2}, \quad k = 1, \dots, K. \quad (6-27)$$

Este resultado, no entanto, não é extensível para o caso L -dimensional.

Outra forma de colocar este problema é: fixada a distância mínima entre símbolos do alfabeto $d_{\min}^* = d_{\min}/\sqrt{2}$, projetar os vetores L -dimensionais $\boldsymbol{\mu}_k$ de forma a minimizar a energia $E = \frac{1}{L} \sum_{k=1}^K p_k \|\boldsymbol{\mu}_k\|^2$, onde $p_k = m_k/n$ é a probabilidade do vetor $\boldsymbol{\mu}_k$. O projeto de alfabeto de VPM é, portanto,

equivalente ao projeto de constelações L -dimensionais não-equiprováveis [24].

Não se conhece uma solução fechada ou um algoritmo ótimo para este problema. Uma abordagem tipicamente encontrada na literatura é a de considerar constelações baseadas em *reticulados* (*lattices*), os quais são arranjos regulares de pontos cobrindo todo o espaço L -dimensional. Em particular, escolhem-se os reticulados mais *densos* conhecidos. Este conceito faz referência ao problema de *empacotamento de esferas* (*sphere packing*), o qual consiste em encontrar o arranjo de esferas maciças que preenche de forma mais eficiente¹ um determinado espaço. A densidade do arranjo é a proporção de espaço ocupado pelas esferas. Observe que se o raio das esferas é fixo em $d_{\min}^*/2$, então a distância mínima entre os centros das esferas (os quais nesse caso correspondem aos pontos do reticulado) não pode ser menor que d_{\min}^* .

Uma constelação com número finito K de pontos pode ser encontrada a partir de um reticulado fazendo-se a intersecção deste com uma região finita do espaço (por exemplo, uma esfera), preferencialmente centrada na origem. Obtém-se assim uma constelação finita e compacta, com distância mínima d_{\min}^* .

Aparentemente, esta abordagem não é necessariamente ótima, pois minimizar o volume ocupado pelo sistema de esferas e minimizar a energia da constelação associada são problemas provavelmente diferentes, ainda que relacionados. No entanto, esta construção através de reticulados densos mostra-se extremamente prática, e tem sido usada com sucesso no projeto de constelações multidimensionais não-equiprováveis [24]. Neste trabalho, aplicamos tal construção também para o projeto de alfabeto de VPM. Após a escolha do alfabeto, os valores ótimos para o vetor de composição são calculados através de (6-26), conforme discutido anteriormente.

A seguir, as figuras 6.1 e 6.2 apresentam exemplos de alguns dos alfabetos usados neste trabalho. Em duas e três dimensões, utilizou-se como base os reticulados A_2 (hexagonal) e D_3 (cúbico de face centrada), respectivamente. Mais informações sobre estes reticulados podem ser encontradas em [10].

¹Considere, por exemplo, o problema de empacotar laranjas em uma caixa de feira. Qual arranjo permite que se coloque o maior número de laranjas na caixa?

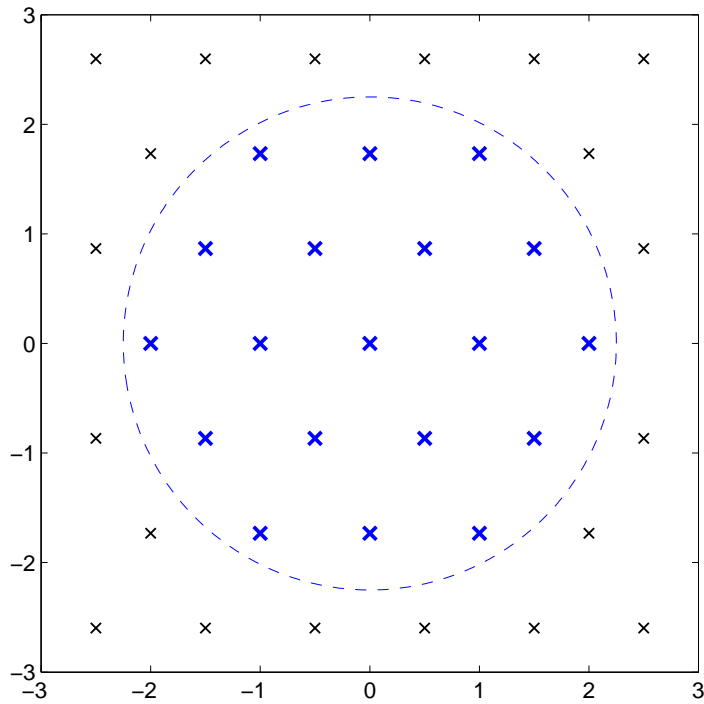


Figura 6.1: Exemplo de alfabeto de VPM baseado no reticulado A_2 . Parâmetros: $L = 2$, $K = 19$ e $d_{\min}^* = 1$.

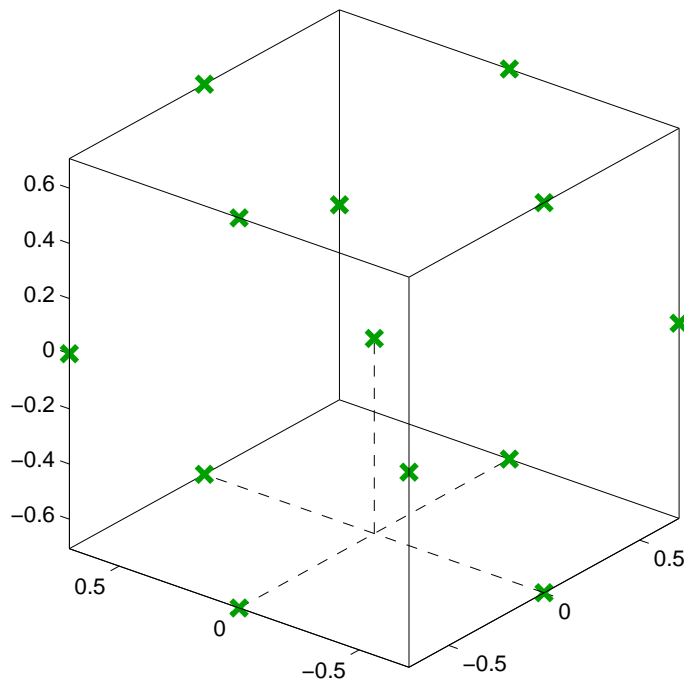


Figura 6.2: Exemplo de alfabeto de VPM baseado no reticulado D_3 . Parâmetros: $L = 3$, $K = 13$ e $d_{\min}^* = 1$.

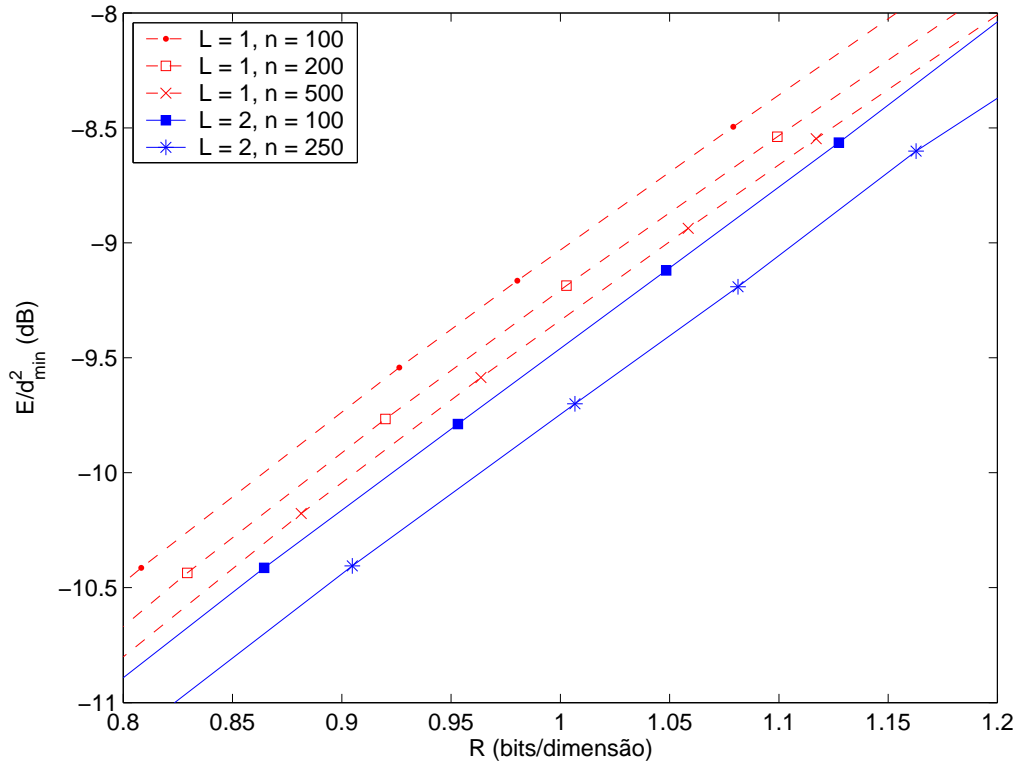


Figura 6.3: Desempenho de VPM's com $L = 1$ e 2 .

6.4 Desempenho

Nesta seção, aplicamos o projeto ótimo de vetor de composição sobre alfabetos com $L = 1, 2$ e 3 dimensões construídos conforme descrito na seção anterior. Inicialmente, partimos de um alfabeto com um grande número de pontos e calculamos o vetor de composição ótimo considerando um pequeno valor de λ , o que resulta em um código com altos valores de taxa e energia. Para n fixo, à medida que λ é aumentado, os valores das repetições m_k vão se tornando cada vez mais desiguais e concentrados em símbolos próximos à origem, o que resulta em códigos com menores valores de energia e taxa. A partir de um certo valor de λ , alguns símbolos μ_k são associados a $m_k = 0$ (probabilidade nula) e portanto são descartados do alfabeto. Para $L = 1, 2$ e 3 , utilizamos alfabetos com número de símbolos K variando, respectivamente, de 5 até 3, de 19 até 7 e de 55 até 13.

O desempenho de alguns VPM's obtidos é mostrado nas figuras 6.3 e 6.4, onde pode ser visto que o desempenho melhora tanto com o aumento do comprimento n quanto com o aumento da dimensão de alfabeto L . A Fig. 6.5 mostra que o desempenho assintótico de um VPM pode ser alcançado por outro VPM com n finito e maior L . O desempenho assintótico de um

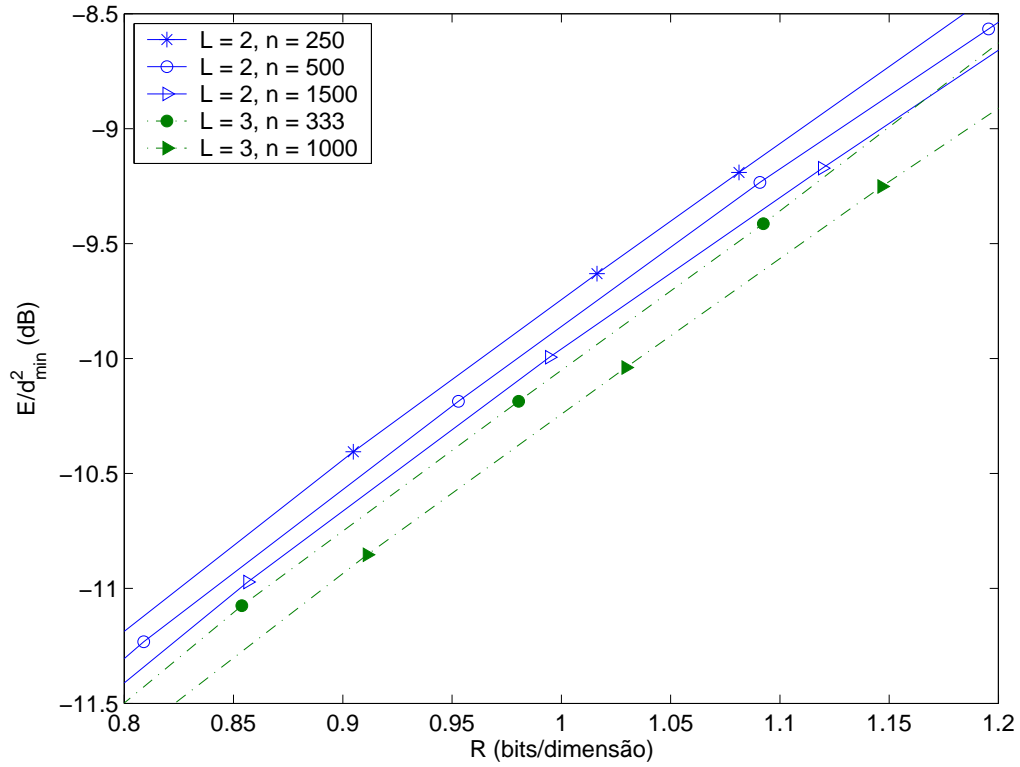


Figura 6.4: Desempenho de VPM's com $L = 2$ e 3.

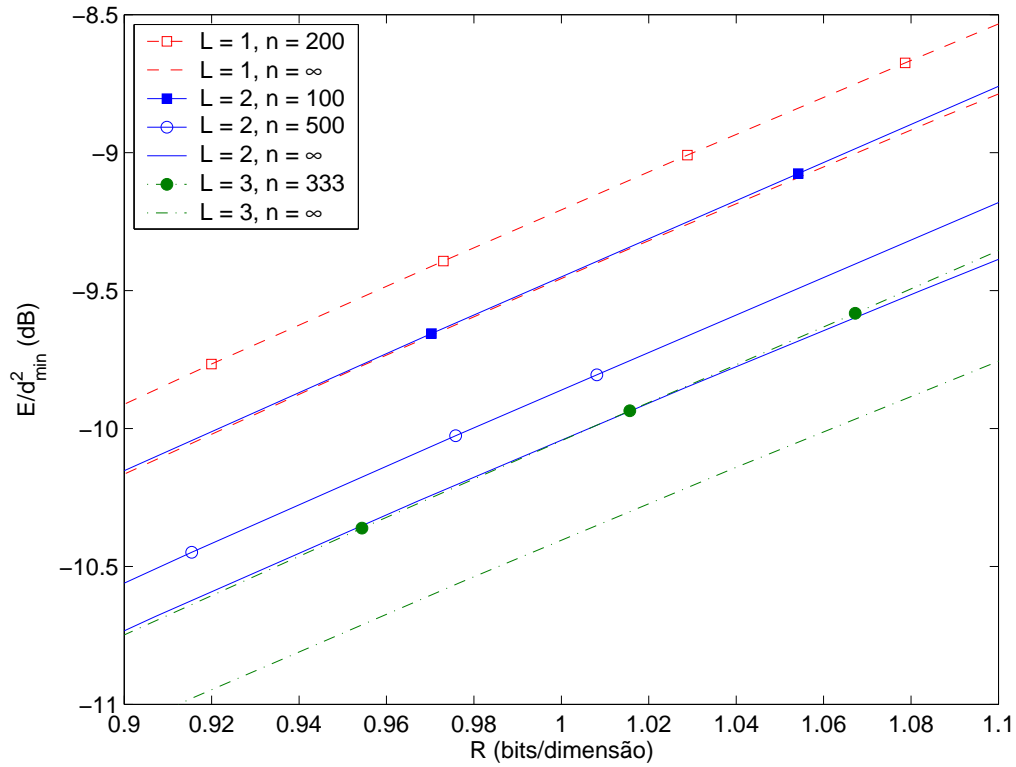


Figura 6.5: Desempenho de VPM's com $L = 1, 2$ e 3. Curvas para $n = \infty$ correspondem ao desempenho assintótico de cada código.

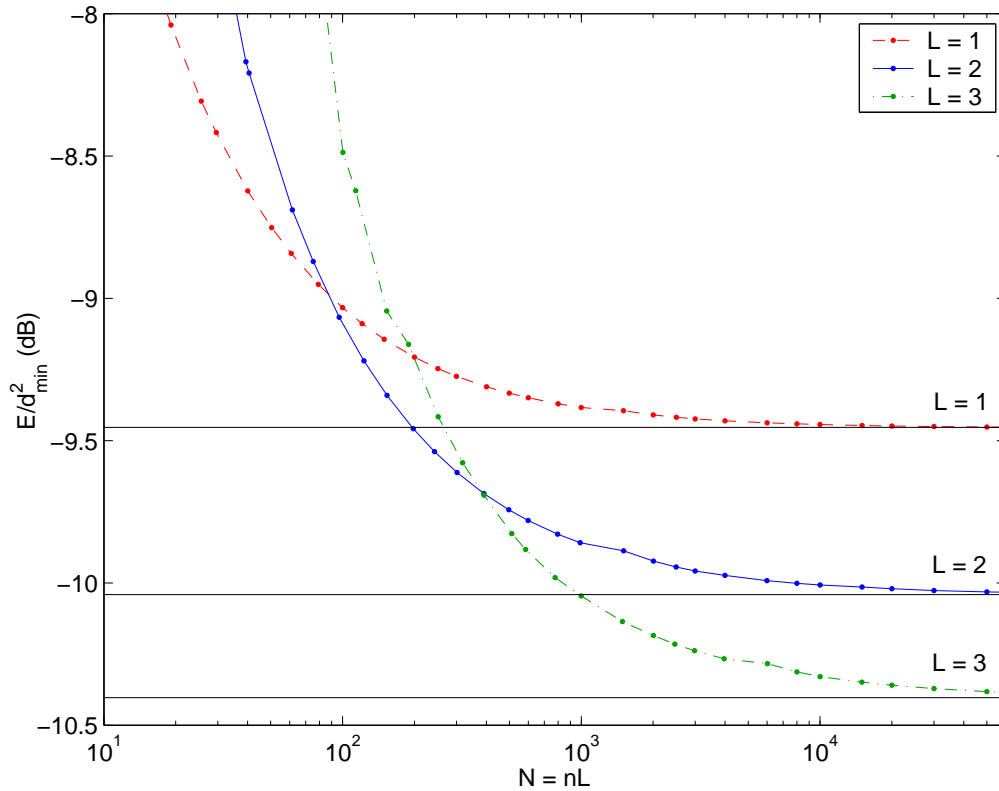


Figura 6.6: Desempenho de VPM's com $L = 1, 2$ e 3 para uma taxa $R = 1$ em função do número total de dimensões $N = nL$.

VPM é calculado fazendo-se $n \rightarrow \infty$ nas expressões para taxa, energia e vetor de composição ótimo, resultando em

$$R = -\frac{1}{L} \sum_{k=1}^K p_k \log p_k \quad (6-28)$$

$$E = \frac{1}{L} \sum_{k=1}^K p_k \|\boldsymbol{\mu}_k\|^2 \quad (6-29)$$

$$p_k = \frac{2^{-\lambda \|\boldsymbol{\mu}_k\|^2}}{\sum_{k=1}^K 2^{-\lambda \|\boldsymbol{\mu}_k\|^2}}, \quad k = 1, \dots, K. \quad (6-30)$$

A Fig. 6.6 mostra, para $R = 1$, a variação do desempenho de VPM's com $L = 1, 2$ e 3 em função do número total de dimensões $N = nL$. O desempenho assintótico de cada código é também mostrado. Observa-se que, a partir de um certo valor do produto nL , o desempenho de um SPM ou VPM é superado pelo desempenho de outro VPM com maior L . Além disso, observamos que o desempenho assintótico ($n \rightarrow \infty$) de um SPM ou VPM pode ser alcançado por outro VPM com maior L e comprimento n apenas moderado.

6.5 Comentários

É interessante notar que as expressões (6-29), (6-28) e (6-30) são idênticas às que podem ser obtidas para modulações L -dimensionais não-equiprováveis [24]. Isto significa que o desempenho assintótico de um VPM é equivalente ao desempenho de sua constelação não-equiprovável associada. Entretanto, é importante ressaltar que, de forma análoga ao caso do ECVQ para compressão de fontes, a técnica da sinalização não-equiprovável, apesar de bastante poderosa, possui taxa variável. Isto resulta em problemas de *overflow* e *underflow* de buffer e de propagação de erros, os quais restringem a utilização desse tipo de sistema somente a aplicações onde estes problemas são toleráveis. O VPM, por outro lado, permite a obtenção de ganhos semelhantes aos que se obtém com a sinalização não-equiprovável, com a vantagem de que a taxa é mantida fixa.