SOBRE A IMPLEMENTAÇÃO DE RETÍCULOS:
UM ALGORITMO DE DECODIFICAÇÃO / DEMAPEAMENTO COMBINADOS

A. IBRAHIN IRSHAID SHARI'A e H. MAGALHÃES DE OLIVEIRA

UNIVERSIDADE FEDERAL DE PERNAMBUCO
CODEC - Grupo de Comunicações DES/UFPE C.P.7800 - 50.711-970 - Recife - PE
Tel.: +(081) 2718210 FAX: +(081) 2718215 e-mail: 69hmo@npd.ufpe.br

Resumo

Retículos constituem uma das técnicas eficientes de modulação codificada. A implementação prática de retículos limitados envolve o mapeamento de seqüências binárias em pontos do retículo, a decodificação de vetores ruidosos em pontos do retículo e o demapeamento dos pontos em seqüências binárias. Apresenta-se um algoritmo para (de)codificação-(de)mapeamento combinados de reticulados obtidos de qualquer construção código. O processo é simplificado pela decodificação de vetores ruidosos diretamente em seqüências binárias. A idéia é explorar configurações que, do ponto de vista prático, permitam implementações atrativas. O desempenho do algoritmo é avaliado por simulação. Para retículos obtidos a partir da construção A, a decodificação é de máxima verossimilhança e para as demais construções, ela é por distância cotada.

Abstract

Lattices are efficient Coded Modulation schemes. The practical implementation of bounded lattices deals with a mapping of binary strings into lattice points, decoding noisy vectors into a lattice point and demapping it into binary sequences. We introduce a combined demapping-decoding algorithm for lattices obtained from any generalized code formula. The process is simplified by decoding a noisy vector directly into a binary stream. The aim is to exploit attractive configurations from the practical point of view. The algorithm performance is evaluated by simulation: It is maximum likelihood for lattices from construction A, and it is bounded distance in any other case.

1. Introdução

A correspondência entre um sistema de comunicação e algumas idéias básicas da Geometria foi brilhantemente formulada por Shannon [1], em 1948, através do Teorema da Codificação de Canais. Uma maneira de se projetar um conjunto de sinais que se aproxime dos padrões prometidos na Teoria de Shannon é representar cada sinal como um ponto em um espaço de n dimensões. O processo de projetar um conjunto de palavras código pode ser reduzido a um "problema geométrico de alocação de pontos em uma região de um espaço".

Há muito que os reticulados, no sentido matemático e cristalográfico do termo,foram propostos como estratégias de codificação sobre o canal Gaussiano [2]. Entretanto, somente após recentes publicações, eles suscitaram um maior interesse [3,4]. Adicionalmente, De Oliveira-Battail [5] e De Buda [6] demonstraram independentemente a existência de retículos que podem atingir a capacidade do canal. O largo sucesso da codificação de canal foi conseguido somente após a introdução da modulação codificada por Ungerboeck [7] e os códigos de retículo constituem numa destas técnicas. Há hoje um consenso que eles constituem uma ferramenta importante na codificação de canal [8]. Ganhos de codificação sem sacrificar a taxa de informação podem ser obtidos por códigos de bloco ou por códigos de treliça.

A confirmação da existência de embalagens densas de esferas em espaços de alta dimensão desencadeou vínculos entre o estudo de empacotamentos e a teoria de codificação. J. Leech e N.J.A. Sloane [9] estabeleceram as primeiras relações entre empacotamento de esferas e os códigos corretores de erro; eles usaram estes códigos para construir novas embalagens não reticuladas, na maioria dos casos, e ao mesmo tempo desenvolveram a "construção código" de muitos retículos conhecidos. A busca de empacotamentos densos como métodos de modulação codificada teve sucesso com o trabalho de Forney *et al*. [2] em 1984, no qual sistemas de modulação codificada em bloco foram construídos usando-se retículos densos nas dimensões 4, 8, 16 e 24- com ganhos de codificação em torno de 1.5, 3.0, 4.5, 6.0 dB, respectivamente [2]. A idéia similar foi descoberta separadamente por Calderbank e Sloane em 1987 [4], utilizando códigos de treliça. Existem outros sistemas que utilizam retículos em dimensões definidas, como aquele de N. Secord e de Buda [10], empregando o retículo de Gosset, e aquele de R. Lang e M. Longstaff [11], utilizando o retículo de Leech [12].

A utilização de retículos densos em altas dimensões tem a finalidade de aproximar-se do ideal de Shannon, que garante a possibilidade de transmitir com potência cerca de 9dB menor do que àquela em PAM, com uma probabilidade de erro controlada, no caso de ruído Gaussiano [2].

O sistema prático mais complexo, utilizando retículos, é baseado no retículo de Leech em 24 dimensões o qual oferece 6 dB de ganho em potência [11]. Do ponto da vista de complexidade, torna-se bastante difícil construir um sistema para oferecer ganhos adicionais, no estado da arte atual. Portanto, muitas pesquisas estão interessadas em simplificar os métodos de implementação dos sistemas já existentes, ou construí-los de maneira alternativa. Um dos principais desafios da codificação de canal consiste na concepção de algoritmos eficientes para decodificação.

A implementação de um sistema deve incluir métodos para simplificar três processos: 1)- mapear as seqüências binárias nos pontos do retículo; 2)- decodificar um ponto recebido com ruído em um ponto do retículo; e 3)- demapear este ponto do retículo a uma seqüência binária adequada.

Normalmente, o 2º processo é tratado separadamente dos demais e é o mais estudado. Neste trabalho, desenvolve-se um sistema de codificação e decodificação que permite simplificação na complexidade dos processos juntos.

Vários retículos densos são examinados, tais como Schäfli, Gosset e Leech, analisando-se o desempenho no canal Gaussiano. Um enfoque especial é dado ao algoritmo.Demostra-se como adaptá-lo às construções generalizadas-Fórmulas código de Forney [13].

A novidade dos resultados diz respeito à implementação prática de retículos, especialmente explorando o fato que as implementações não lidam com retículos, na acepção estrita do termo, mas com um subconjunto próprio dele extraído, em uma região limitada formada pela concatenação de constelações constituintes bidimensionais. O problema do (de)mapeamento de pontos em seqüências binárias (rotulação binária de pontos do retículo) é realizado de forma eficaz, levando em conta a finitude do conjunto de sinais, o que comumente não é explorado nos algoritmos até então propostos.

CONSTRUÇÃO CÓDIGO DE RETÍCULOS:

Existem quatro tipos de construções códigos de retículos, as quais são designadas por construções A, B, C e D. A construção A é a mais utilizada nas dimensões de 1 a 8, enquanto a B é efetiva nas dimensões de 8 a 24. A construção C é uma generalização de A e B e é efetiva nas dimensões $n=2^m$ [11], m=1,2,...,k, e finalmente a construção D é interessante em dimensões como 36 e 64 [14].

Construção Código A. Suponha que C é um código binário (n,m,d). A construção seguinte especifica uma embalagem de esferas em $\Re^n$: $\underline{X} = (x_1,x_2,.....,x_n)$ é um ponto na embalagem se e somente se $\underline{X}$ for congruente (mod 2) com uma palavra código em C [10]. Em outras palavras, os retículos que podem ser representados pela fórmula

$$\Lambda = 2Z^n + C_0 . \qquad (1)$$

Construção Código B. Seja C um código binário (n,m,d) com a propriedade que cada palavra tenha peso par; então a construção seguinte especifica o conjunto de pontos que formam uma embalagem em $\Re^n$: $\underline{X} = (x_1,x_2,...,x_n)$ é um ponto no retículo se e somente se $\underline{X}$ for congruente (mod 2) com uma palavra código e Σx_i for congruente (mod 4) com 0 [9]. Estes retículos são representados pela fórmula

$$\Lambda = 4Z^n + 2C_1 + C_0 , \qquad (2)$$

onde C_0 é um subcódigo de C_1 são incluídos na construção B [13].

Construção Código Generalizada. Suponha que C_{k-1}, C_{k-2},...,C_0 são códigos binários obedecendo à condição $C_j \subseteq C_{j+1}$. Então o retículo Λ inclui todos os pontos $\underline{X}$, tais que $\underline{X} \equiv 2^k Z^n + 2^{k-1}c_{k-1} + ... + 2^2 c_2 + 2c_1 + c_0$, onde c_j é uma palavra do código C_j. Representa-se isto pela fórmula código [8,13]

$$\Lambda \equiv 2^k Z^n + 2^{k-1}C_{k-1} + ... + 2^2 C_2 + 2C_1 + C_0. \qquad (3)$$

Tal construção é conhecida como construção D. No caso em que os códigos acima não são lineares ou $C_j \not\subset C_{j+1}$, a construção é conhecida como construção C. Os empacotamentos gerados neste caso são, geralmente, não reticulados.

2. Métodos de (De)codificação.

Nos canais ruidosos e limitados em banda passante, os retículos são usados como métodos de modulação codificada com a finalidade de aumentar a taxa de transmissão. Os pontos transmitidos são perturbados por um vetor de ruído aleatório multi-dimensional, o qual desloca os pontos ao espaço $\Re^n$. Um sistema de codificação e decodificação pode ser visto como um (conjunto de) algoritmo(s) que realiza(m) três funções: i) a função de codificação (mapeamento) $\xi(x)$. Esta função mapeia uma seqüência binária da fonte $\underline{b}=(b_1,b_2,..,b_n)$ em um ponto $\underline{y}=(y_1,y_2,...,y_n)$ do retículo Λ, i.e., $\underline{y} = \xi(\underline{b})$. ii) a função de decodificação $\phi(\underline{x})$. Como o ruído desloca o ponto transmitido para um ponto do espaço $\Re^n$, esta função encontra o ponto $\underline{y}= (y_1,y_2,..,y_n)$ mais próximo do retículo ao ponto recebido $\underline{x}=(x_1,x_2,..,x_n)$ do espaço $\Re^n$, i.e. $\underline{y}=\phi(\underline{x})$. iii) a função de demapeamento $\lambda(x)$. Esta função mapeia um ponto $\underline{y}=(y_1,y_2,...,y_n)$ do retículo Λ na seqüência binária $\underline{b}=(b_1,b_2,...,b_n)$ correspondente, $\underline{b}=\lambda(\underline{y})$. Ela deve ser o inverso da função de mapeamento, $\lambda(\underline{y}) = \xi^{-1}(\underline{x})$.

A pesquisa de métodos eficientes de decodificação é uma busca para diminuir a complexidade (aumentar sua velocidade), diminuir o volume da memória, e facilitar a sua implementação.

Muitos trabalhos tem sido realizados após o artigo pioneiro de J.H. Conway e N.J. Sloane [15] os quais propuseram algoritmos de decodificação dos retículos D_n, E_6, E_7, E_8, e seus duais.

Explorando a construção código dos retículos, J.H. Conway e N.J. Sloane [16] apresentaram um algoritmo de decodificação dos retículos obtidos pela construção A. Para retículos obtidos através da

aplicação da construção B, o problema é mais complexo, pois existem dois códigos dependentes que devem ser tratados. Considerando Λ como união de k classes laterais, Forney [17] propôs um algoritmo para decodificá-lo.

Baseado no conceito de Ungerboeck "partição de conjuntos" Forney *et al.* [2] apresentaram em linhas gerais como seria a codificação e decodificação dos retículos D_4, E_8, Λ_{16}, e Λ_{24}.

Devido a importância do retículo de Leech, muitos algoritmos foram propostos para decodificá-lo, tais como: o de J.H. Conway e N.J. Sloane [15] baseado no fato que Λ_{24} contém D_{24} como sub-retículo; o de Forney [17], o qual considerou Λ_{24} como duas classes laterais de H_{24}; o de G.R. Lang e F.M. Longstaff [11], os quais propuseram um sistema de codificação e decodificação deste retículo, baseado na construção código apresentada por Forney [13]. Este último, foi adotado em um MODEM 19.200 bits/seg da MOTOROLA [8].

Na decodificação por máxima verossimilhança (MLD), compara-se o ponto recebido com todos os pontos do retículo, usando-se a distância Euclidiana como métrica. A complexidade deste processo foi diminuída por algoritmos como aqueles supra mencionados. A maioria destes algoritmos trata os retículos **ilimitados** (sentido estrito) no espaço infinito.Nenhum destes algoritmos dedica-se a reduzir a complexidade do problema de (de)mapeamento entre as seqüências binárias e pontos do retículo. Pretende-se desenvolver um algoritmo que aborde ambos problemas simultaneamente.

Neste trabalho, os retículos são construídos através de constelações bidimensionais constituintes cujas coordenadas assume um dos valores na grade: $\pm1, \pm3, \pm5, ..\pm(2^m-1)$.

De Oliveira apontou em [18] que uma palavra binária de comprimento 2m é atribuída a cada ponto, onde m=3. Inicialmente, cada dois bits são acoplados como um vetor bidimensional $\underline{b}_i=(b_{i1}, b_{i2})$, definindo m vetores, de tal modo que a palavra binária $\underline{b}$ pode ser escrita como:

$$\underline{b} = (b_{m-1},b_{m-2},.....,b_1,b_0). \quad (4)$$

A conversão da informação binária em ponto de sinalização pode ser feita pelo procedimento seguinte.

Algoritmo (1)

etapa 1

Gerar uma nova palavra **B** pela troca dos 0's por -1's, definindo assim:

$$\underline{\mathbf{B}} = (\mathbf{B}_{m-1},\mathbf{B}_{m-2},.....,\mathbf{B}_1,\mathbf{B}_0) . \quad (5)$$

etapa 2

O ponto da constelação é dado por:

$$\underline{P} = \mathbf{B}_{m-1}2^{m-1}+...+\mathbf{B}_{m-1}*\mathbf{B}_{m-2}*...*\mathbf{B}_1 2^1+ \mathbf{B}_{m-1}*\mathbf{B}_{m-2}*...*\mathbf{B}_1*\mathbf{B}_0\, 2^0, \quad (6)$$

onde * denota a multiplicação vetorial (componente a componente) definida por:

$\underline{P}_1*\underline{P}_2=(x_1x_2,y_1y_2)$, onde $\underline{P}_i=(x_i,y_i) \in \Re^2$.

A interpretação deste método de rotulação binária é simples. Os primeiros dois bits (10) selecionam o quarto quadrante da constelação de 64 pontos, reduzindo-a a uma constelação de 16 pontos cuja origem está no ponto (4,-4). Os segundos dois bits (00) escolhem o terceiro quadrante da nova constelação, reduzindo-a a 4 pontos, cuja origem é no ponto (-4,4)+(-2,2)=(2,-2). Os últimos dois dígitos escolhem o ponto (1,-1) nesta última constelação.

A designação dos símbolos binários aos quadrantes é feita de acordo um código de Gray 2D, onde a constelação de sinais é dividida em quatro sub-constelações menores equivalentes. Uma mudança para diminuir a complexidade do algoritmo assume que os arcos sempre tenham a mesma direção. Deste modo, re-rotulam-se os pontos da constelação. Portanto a eq. (6) será:

$$\underline{P} = \mathbf{B}_{m-1}\, 2^{m-1}+...+ \mathbf{B}_1\, 2^1+\mathbf{B}_0\, 2^0. \quad (7)$$

O demapeamento é similar ao algoritmo em [18], exceto pelas conseqüências da modificação anterior. Suponha que **sgn**(.) seja um operador definido sobre $\Re^2$, que indica o sinal das duas coordenadas, i.é.,

$$\mathbf{sgn}(\underline{R}_i) = (\text{sgn } x_i , \text{sgn } y_i),$$

dado $\underline{R}_i =(x_i ,y_i) \in \Re^2$. Nesta seção, $\mathbf{B}_i$ denota o vetor bidimensional obtido por:

$$\mathbf{B}_i = \mathbf{sgn}\,(\underline{R}_i). \quad (8)$$

Suponha que $\underline{R}$ é um vetor recebido em $\Re^2$. O algoritmo seguinte mapeia este vetor em uma seqüência binária $\underline{b}$, que corresponde ao ponto mais próximo de uma constelação de 2^{2m} pontos:

Algoritmo (2)

Faça i = m-1; $\underline{R}_i= \underline{R}$ (condições iniciais)

etapa 1

$\mathbf{B}_i = \mathbf{sgn}\,(\underline{R}_i)$;

se i = 0, então termine o processo.

etapa 2

$\underline{R}_{i-1} = \underline{R}_i - 2^i\, \mathbf{B}_i$; i = i-1;

vá à etapa 1.

A seqüência binária decodificada é aquela que corresponde ao vetor **B** após a troca de -1's por 0's.

Este algoritmo pode ser aplicado a uma constelação cúbica n dimensional; a única mudança ocorre no vetor $\underline{b}_i$ o qual será definido em n dimensões: $\underline{b}_i= (b_{i1},b_{i2},...,b_{in})$.

Tal representação de pontos em constelações cúbicas de n-dimensões pode ser generalizada aos retículos n-dimensionais construídos através de constelações bidimensionais. Mostrou-se em [19] que um ponto do retículo se apresenta na forma:

$$\underline{P}=\mathbf{B}_{m-1}2^{m-1}+\mathbf{B}_{m-2}\, 2^{m-2}+...+\mathbf{B}_1 2^1+\mathbf{B}_0\, 2^0 \quad (9)$$

onde $\mathbf{B}_i =(B_{i1},B_{i2},....,B_{in})$, sendo $B_{ij}=\pm1$ e n é a dimensão do retículo.

Nos retículos da construção A, $\mathbf{B}_0$ é uma palavra do código C_0 especificado para cada retículo. Nos retículos da construção B, os vetores $\mathbf{B}_0$ e $\mathbf{B}_1$ são palavras de dois códigos C_0 e C_1. É claro que eq.(9) eqüivale à generalização da eq.(7) em n dimensões, exceto que: Na construção A, $\mathbf{B}_0$ é uma palavra codificada; na construção B, $\mathbf{B}_0$ e $\mathbf{B}_1$ ambos são palavras codificadas; na construção generalizada $\mathbf{B}_0,\mathbf{B}_1,...\mathbf{B}_g$, g<m são palavras codificadas.

Desde que as construções A e B são casos especiais da construção generalizada, são exibidos apenas algoritmos de (de)codificação para a última construção.

3.Codificação para Retículos Obtidos via Construção Código Generalizada.

Suponha que

$$\Lambda + 2^g Z^n + 2^{g-1}C_{g-1} + \ldots + 2^2 C_2 + 2^1 C_1 + C_0, \quad (10)$$

é um retículo obtido pela construção código generalizada; os vetores $\mathbf{B}_0,\mathbf{B}_1,\ldots\mathbf{B}_g$, devem pertencer a $C_0(n,k_0,d_0),C_1(n,k_1,d_1),\ldots,C_{g-1}(n,k_{g-1},d_{g-1})$. Portanto, um retículo assim obtido é capaz de transmitir a uma taxa $n(m-g)+k_{g-1}+\ldots+k_1+k_0$ bits/ponto, usando-se uma constelação de 2^{2m} pontos.

Mapeamento bit a ponto. Uma seqüência binária $\underline{b}$ com $n(m-g)+k_{g-1}+\ldots+k_1+k_0$ bits pode ser mapeada em um ponto no retículo Λ, construído a partir de uma constelação bidimensional com 2^{2m} pontos como segue.

Algoritmo (3)

etapa 1:

começando do último dígito da seqüência $\underline{b}$, cada k_i bits, onde i= 0,1,...g-1, são codificados em uma seqüência de n dígitos, utilizando o código $C_i(n,k,d)$, gerando, no total, uma seqüência de nm bits.

etapa 2:

dividir os nm bits em blocos $\underline{b}_i$ de n bits cada um; Gerar uma nova palavra $\mathbf{B}_i$ pela troca dos 0's por -1's, definindo assim: $\underline{\mathbf{B}}=(\mathbf{B}_{m-1},\mathbf{B}_{m-2},\ldots\ldots,\mathbf{B}_1,\mathbf{B}_0)$.

etapa 3:

O ponto do retículo é dado por (9).

Exemplo 1

Vamos mapear a seqüência (100110100011) em um ponto no retículo E_8. Como a fórmula código de E_8 é $2Z^8+C(8,4,4)$, e como há 12 bits para especificar um ponto, deve-se usar uma constelação de 2^4 pontos (m=2). Portanto o algoritmo acima resulta em:

1) usando o código (8,4,4), os últimos 4 bits (0011) são codificados em (11110000), gerando uma nova seqüência binária $\underline{b}$ = (1001101011110000);

2) tem-se então a seqüência

$\mathbf{B}$=(1,-1,-1,1,1,-1,1,-1,1,1,1,1,1,-1,-1,-1,-1) = $(\mathbf{B}_1,\mathbf{B}_0)$, então, P=(2,-2,-2,2,2,-2,2,-2)+(1,1,1,1,-1,-1,-1,-1) = (3,-1,-1,3,1,-3,1,-3).

Mapeamento Ponto A Bit. O algoritmo (2) pode ser generalizado para decodificar qualquer retículo obtido usando-se a construção código generalizada [13]. Suponha que o retículo Λ tenha a fórmula código dada em (3), onde $\forall i$ C_i é um código de bloco, $C_j \subseteq C_{j+1}$. O algoritmo seguinte codifica um vetor $\underline{R} \in \Re^n$ em uma seqüência binária corresponde a um ponto mais próximo a este vetor no retículo Λ.

Algoritmo (4).

faça i = m-1; $\underline{R}_i = (r_{i1},\ldots,r_{in}) = \underline{R}$; j=0,

defina o vetor $\underline{Q} = (q_1,\ldots,q_n) = |\underline{R}|$

etapa 1 :

$\mathbf{B}_i$ = **sgn** ($\underline{R}_i$);

se $|r_{ij}| < q_j$, então $q_j = |r_{ij}|$, j=1,...,n,

se i= j, vá à etapa 3.

etapa 2 :

$\underline{R}_{i-1} = \underline{R}_i - 2^i \mathbf{B}_i$; i = i-1;

volte à etapa 1

etapa 3 :

(i) decodifique o vetor $\underline{R}_j = (\underline{Q} * \mathbf{B}_j)/2^j$ numa palavra do código $C_j(n,k,d)$ usando um dos métodos de decodificação suave ótima, gerando um novo vetor B_j;

(ii) $\underline{R} = (\underline{R} - 2^j \mathbf{B}_j)$,

se j = g-1 vá à etapa 4,

j = j+1,

i = m-1; $\underline{R}_i = \underline{R}$; $\underline{Q} = \underline{R}$

vá à etapa 1

etapa 4 :

faça i = m-1; $\underline{R}_i = \underline{R}$ (condição inicial)

(i) $\mathbf{B}_i$ = **sgn** ($\underline{R}_i$);

se i= g, vá à etapa (iii).

(ii) $\underline{R}_{i-1} = \underline{R}_i - 2^i \mathbf{B}_i$;

i = i-1; volte à etapa (i);

(iii) O ponto decodificado é

$P = \mathbf{B}_i 2^i + \mathbf{B}_{i-1}2^{i-1} + \ldots + \mathbf{B}_1 2^1 + \mathbf{B}_0 2^0$.

Em cada $\mathbf{B}_i$ troque os -1's por 0's gerando novas seqüências $\underline{b}_i$. Em cada $\underline{b}_i$, encontre os dígitos de informação utilizando o código C_i para i<g. Os dígitos de informação nos vetores $\underline{b}_0,\underline{b}_1,\ldots,\underline{b}_{g-1}$, juntamente com $\underline{b}_{m-1},\underline{b}_{m-2},\ldots,\underline{b}_g$,formam a seqüência binária correspondente ao ponto P, calculado em (iii, etapa 4).

Exemplo 2

O retículo Λ_{16} pode ser obtido pela construção B, com fórmula código $4Z^n+2C_1(16,15,2)+C_0(16,5,8)$. O código C(16,5,8) é um código de Reed-Muller de 1ª ordem, enquanto o código C(16,15,2) é um código de um único dígito de paridade. Suponha que Λ_{16} seja construído usando-se uma constelação bidimensional de 2^8 pontos (m=4). Deseja-se decodificar no ponto mais próximo um vetor ruidoso recebido, e.g.

$\underline{R}$ = (12.8,-8.9,0.5,-3.7,5.8,-7.3,-11.1,2.2,3.4,12.2, -9.3, -4.2,6.3,8.5,-1.1,-5.1)

As etapas 1, 2 e 3 do algoritmo (4) produzem:

$\mathbf{B}_0$=(-1-1-1-1-1-1-1-1 1 1 1 1 1 1 1 1).

Logo subtraindo $\mathbf{B}_0$ do vetor $\underline{R}$, $\underline{R} \leftarrow \underline{R}-\mathbf{B}_0$,

R=(13.8,-7.9,1.5,-2.7,6.8,-6.3,-10.1,3.2,2.4,11.2, -10.3,-5.2,5.3,7.5,-2.1,-6.1). Usando este vetor, a etapa 3 (ii) do algoritmo (4) resulta:

i		R_i / $\mathbf{B}_i$							
3	R_3	13.8	-7.9	1.5	-2.7	6.8	-6.3	-10.1	3.2
		2.4	11.1	-10.3	-5.2	5.3	7.5	-2.1	-6.1
	$\mathbf{B}_3$	1	-1	1	-1	1	-1	-1	1
		1	1	-1	-1	1	1	-1	-1
2	R_2	5.8	0.1	-6.5	5.3	-1.2	1.7	-2.1	-4.8
		-5.6	3.2	-2.3	2.8	-2.7	-0.5	5.9	1.9
	$\mathbf{B}_2$	1	1	-1	1	-1	1	-1	-1
		-1	1	-1	1	-1	-1	1	1
1	R_1	1.8	-3.9	-2.5	1.3	2.8	-2.3	1.9	-0.8
		-1.6	-0.8	1.7	-1.2	1.3	3.5	1.9	-2.1

O vetor $\mathbf{B}_1$ é um palavra em $C_1(16,15,2)$. A etapa 4 do mesmo algoritmo resulta:

i		R_i $\mathbf{B}_i$							
3	R_3	10.8	-5.9	3.5	-4.7	4.8	-4.3	-12.1	5.2
		4.4	13.1	-12.3	-3.2	3.3	5.5	-4.1	-4.1
	$\mathbf{B}_3$	1	-1	1	-1	1	-1	-1	1
		1	1	-1	-1	1	1	-1	-1
2	R_2	2.8	2.1	-4.5	3.3	-3.2	3.7	-4.1	-2.8
		-3.6	5.2	-4.3	4.8	-4.7	-2.5	2.9	3.9
	$\mathbf{B}_2$	1	1	-1	1	-1	1	-1	-1
		-1	1	-1	1	-1	-1	1	1

Portanto o ponto decodificado será

$\underline{P}=\mathbf{B}_3 2^3+\mathbf{B}_2 2^2+\mathbf{B}_1 2^1+\mathbf{B}_0 2^0$

$= (\ 1\ -1\ 1\ -1\ 1\ -1\ -1\ 1\ 1\ 1\ -1\ -1\ 1\ 1\ -1\ -1)\ 2^3$

$+ (\ 1\ \ 1\ -1\ 1\ -1\ 1\ -1\ -1\ -1\ 1\ -1\ 1\ -1\ -1\ 1\ 1)\ 2^2$

$+ (\ 1\ -1\ -1\ 1\ 1\ -1\ 1\ -1\ -1\ -1\ 1\ -1\ 1\ 1\ \ 1\ -1)\ 2^1$

$+ (-1\ -1\ -1\ -1\ -1\ -1\ -1\ -1\ \ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1)\ 2^0$

$= (11,-7,1,-3,5,-7,-11,3,3,11,-9\ ,-5,7,11,-1,-5).$

$\underline{b}_0=(\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1)$

$\underline{b}_1=(\ 1\ 0\ 0\ 1\ 1\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 1\ 1\ 1\ 0)$

$\underline{b}_2=(\ 1\ 1\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 1\ 1)$

$\underline{b}_3=(\ 1\ 0\ 1\ 0\ 1\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 1\ 1\ 0\ 0)$

Os dígitos de informação em $\underline{b}_0$ são (01000) e em $\underline{b}_1$ são (100110100010111), portanto a seqüência binária decodificada corresponde ao ponto $\underline{P}$ será:
b=(0100010001101000101111101010001010011101 0100111001100)

4. Complexidade e Desempenho.

Os algoritmos apresentados são inéditos em mapear seqüências binárias aos pontos de retículo, ou decodificar vetores ruidosos às seqüências binárias correspondentes adequadas sem nenhuma utilização de tabelas (*look up tables*). Estes algoritmos exploram o fato que os retículos são composição de códigos binários. Isto abre caminho para a utilização da distância de Hamming no processo de decodificação.

Esta designação de rótulos binários aos pontos do retículo tem a vantagem de que, associa menos bits de informação (mais redundância) a parte do ponto que é mais provável de ser alterada pelo ruído. O quadrado da distância Euclidiana entre os dois pontos é

$$d^2(\mathbf{P}-\mathbf{P}')=4*d_H(\mathbf{B}_{m-1},\mathbf{B}'_{m-1})2^{2(m-1)}+$$
$$4d_H(\mathbf{B}_{m-2},\mathbf{B}'_{m-2})2^{2(m-2)}+\ldots+$$
$$4d_H(\mathbf{B}_1,\mathbf{B}'_1)2^2+4d_H(\mathbf{B}_0,\mathbf{B}'_0)2^0,$$

onde d_H é a distância de Hamming entre as énuplas.

Esta equação revela que se as seqüências que correspondem aos dois pontos diferem somente em um bit em B_j, a distância euclidiana entre eles seria 4.2 . Para os retículos obtidos pela construção A,

$$d_{H(min)}(\mathbf{B}_j,\mathbf{B}'_j)=1,\ j = 1,2,\ldots,m-1,\ \text{e}$$
$$d_{H(min)}(\mathbf{B}_0,\mathbf{B}'_0) = d_{H(min)}(C_0) = d_0.$$

O quadrado da distância mínima de um retículo construído aplicando-se construção A é $d^2_{min}(\Lambda)=$mín $[16, 4\ d_{H(min)}(C_0)]=$mín $[16, 4d_0]$.
Da mesma maneira, mostra-se que para um retículo obtido aplicando-se a construção B, o quadrado da distância mínima é $d^2_{min}(\Lambda)=$mín$[64,16\ d_{H(min)}(C_1), 4d_{H(min)}(C_0)]=$mín$[64,16d_1,4d_0]$, e para um retículo obtido aplicando-se a construção código generalizada, $d^2_{min}(\Lambda)=$mín$[4^{g+1},4^g d_g,\ldots,16d_1,4d_0]$ onde d_i é a distância mínima de Hamming do código C_i. Estes resultados confirmam aqueles em [2] e [13].

A equação acima revela que, se um erro de intensidade $d^2_{min}(\Lambda)/2$ atingir um ponto, no máximo 1 bit em $\mathbf{B}_1$ pode ser alterado, ou d_0 bits em $\mathbf{B}_0$. Portanto é conveniente colocar menos bits de informação em $\mathbf{B}_0$. O mesmo pode ser concluído para retículos obtidos aplicando-se outras construções.

Como os algoritmos que (de)codificam os retículos obtidos pela construção A ou B são casos particulares daqueles de retículos obtidos aplicando-se a construção código generalizada, avalia-se apenas a complexidade do último.

Seja Λ um retículo obtido aplicando-se a construção código generalizada, utilizando uma constelação bidimensional de 2^{2m} pontos. A complexidade pode ser avaliada da seguinte maneira;

1-Para decodificar uma palavra de C_0 são necessárias mn subtrações adicionado ao nº de operações da decodificação suave.

2-Para decodificar uma palavra de C_1 são necessárias n((m-1)+1) subtrações, além do nº de operações da decodificação suave.

3- em geral, para decodificar uma palavra do código C_i são necessárias n((m-i)+1) subtrações, além do nº de operações da decodificação suave, i=0,1,...,g-1.

4- Para decodificar o vetor $2^g Z^n$ são necessárias n(m-g+1) subtrações.

Portanto, são necessárias S =n(mg+m+g/2- g^2+1/2) +SC operações para mapear um ponto em uma seqüência binária, onde SC é o número de operações para obter a decodificação suave utilizando os códigos C_i, onde i= 0, 1,.., g-1.

No caso da codificação, os algoritmos precisam de n(m-1) adições, além do nº de operações para codificar $k_{g-1}+\ldots+k_1+k_0$ bits de informação nos códigos correspondentes.

A figura (1) compara a probabilidade de erro por bit (BER) e por ponto para o retículo E_8, construído usando uma constelação bidimensional de 16 pontos, usando o algoritmo e uma rotulação por tabela (*look up table*). Os resultados obtidos foram indistinguivéis.

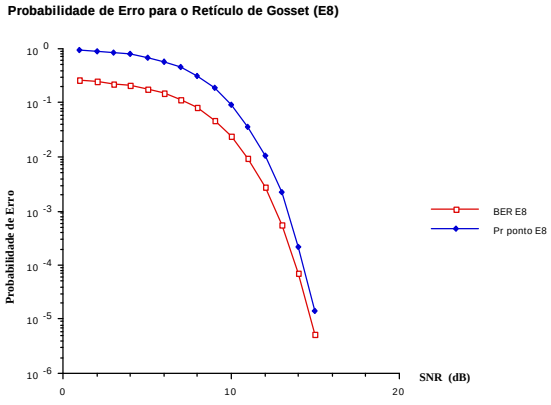

Figura 1. BER e Prob. De Erro por ponto para retículo E8, usando o algoritmo 4; simulação.

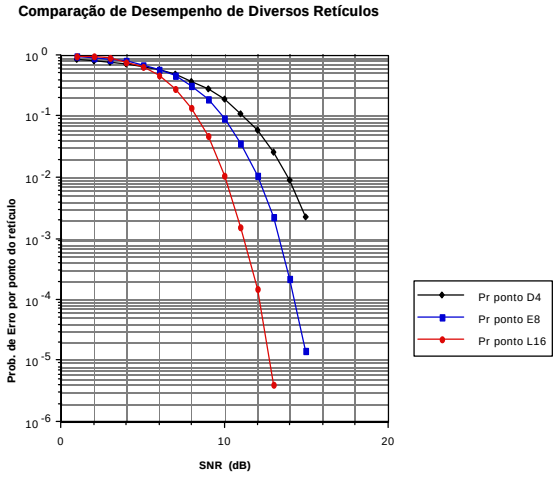

Figura 2. Desempenho de diversos retículos com decodificação MLD e pelo algoritmo 4; simulação.

Como mostram as figuras (1 e 2), o desempenho do algoritmo de decodificação para retículos obtidos aplicando-se a construção código A é idêntico àquele de qualquer algoritmo de decodificação por máxima verossimilhança. Como observado, a decodificação para retículos da construção código generalizada, não realiza MLD, porém as perdas em desempenho inferiores a 0,5 dB. Na simulação, foram considerados intervalos de confiança a 95%.

5. Conclusões

Partindo da "decodificação rápida QAM" estabelecida para constelações bidimensionais (Retículos cúbicos Z^2), foram introduzidos alguns algoritmos de codificação e decodificação para códigos obtidos à partir de qualquer construção código. Estes algoritmos diferem dos demais em:

1- Nos algoritmos de codificação, as seqüências binárias são mapeadas diretamente em pontos do retículo, sem a utilização de *look up tables*.

2- Nos algoritmos de decodificação, os pontos ruídosos são demapeados em seqüências binárias, sem a necessidade de calcular os pontos que correspondem a estas seqüências.

3- Estes algoritmos aproveitam o fato de que os retículos são composições de códigos binários. Portanto, os processos de (de)codificação de retículos são reduzidos a um problema equivalente com códigos binários.

O desempenho dos algoritmos de decodificação é avaliado por simulação Monte Carlo e comparado com a decodificação por máxima verossimilhança, utilizando como exemplos os retículos conhecidos D_4, E_8, Λ_{16} e Λ_{24}. Dos resultados, conclui-se que: (i) Para retículos obtidos pela construção código A, estes algoritmos são de decodificação por máxima verossimilhança. (ii) Para os retículos obtidos pela construção código generalizada (exceto a construção A), estes algoritmos são de decodificação por distância cotada [21].

Com base nos resultados de simulação, conclui-se que a decodificação por distância cotada é, praticamente, equivalente àquela por máxima verossimilhança. Como os algoritmos apresentados reduzem o problema de decodificação num problema equivalente com códigos binários, sugere-se investigar a utilização da decodificação algébrica no processo de decodificação no intuito de analisar o compromisso desempenho × complexidade.

Agradecimentos- O primeiro autor agradece o suporte da Capes. Este trabalho recebeu apoio parcial do *Conselho Nacional de Desenvolvimento Científico e Tecnológico* - CNPq.

Referências

[1]-C.E. Shannon,"Communication in Presence of Noise", *Proc. IRE*, **37**,10-21,Jan.,1948.

[2]-I.F. Blake, "The Leech Lattice as a code for the aussina Channel", *Info. Contr.*,1 **9**,66-74,1971.

[3]- G.D. Forney Jr et al.,"Efficient Modulation for Band-limited Channels",*IEEE Select. Areas Commun.*,**SAC2**, 632-645,Sep.,1984.

[4]- A.R. Calderbank and N.J.A. Sloane, "New Trellis Codes Based on Lattices and Cosets", *IEEE Trans.*, **IT33**,177-195,Mar.,1987.

[5]- H.M. de Oliveira and G. Battail, "A Capacity Therorem for Lattices Codes on the Gaussian Channel", SBT/IEEE Int. Symp. Telecomm.,ITS,Sept., Rio de Janeiro, 1990.

[6]- R. de Buda,"Some Optimal Codes Have Structure", *IEEE Select. Areas Comm.*,**SAC7**,893-899,1989.

[7]-G. Ungerboeck, "Channel Coding with Multilevel/Phase Signals", *IEEE Trans. Info. Theory*,**IT28**,55-67,Jan.,1982.

[8]- J.H. Conway and N.J.A. Sloane, *Sphere Packings, Lattices and Groups*,NY:Springer-Verlag,1988.

[9]-J. Leech and N.J.A. Sloane,"Sphere Packing and Error-Correcting Codes", *Canad. J. Math.*,**23**,718-745,1971.

[10]-N.P. Secord and R. de Buda,"A Two Stage Sequential Demodulator for the Gosset Lattice",*IEEE Select. Areas Commun.*,**SAC7**,974-981,Aug.,1989.

[11]-G.R. Lang and F.M. Longstaff," A Leech Lattice Modem",*IEEE Select. Areas Commun.*,**SAC7**,968-973,Aug.,1989.

[12]- J. Leech,"Notes on Sphere Packings", *Canad. J. Math.*,**1 9**,251-267,1967.

[13]-G.D. Forney Jr,"Coset Codes-Part I: Introduction and Geometrical Classification",*IEEE Trans.*,**IT34**,1123-1151, Sep.,1988.-Part II: Binary Lattices and Related Codes",*IEEE Trans.*,**IT34**,1152-1187,Sep.,1988.

[14]-E.S. Barnes and N.J.A. Sloane,"New Lattice Packings of Spheres",*Canad. J. Math.*, **35**,117-130,1983.

[15]-J.H. Conway and N.J.A. Sloane,"Fast Quantizing and Decoding Algorithms for Lattice Quantizers and Codes",*IEEE Trans. Info. Theory*, **IT28**,227-231,Mar.,1982.

[16]-J.H. Conway and N.J.A. Sloane,"Soft Decoding Techniques for Codes and Lattices, including Golay Code and the Leech Lattice",*IEEE Trans..*,**IT32**,41-50,Jan.,1988.

[17]-G.D. Forney Jr,"A Bounded-distance Decoding Algorithm for the Leech Lattice",*IEEE Trans.*,**IT35**,906-909,1989.

[18]-H.M. de Oliveira and G. Battail,"On Generalized 2-Dimensional Constellations and the Opportunistic Secondary Channel",*Ann. Télécomm.*,**47**,n.5-6,202-213,1992.

[19]-A.I. Shari´a, "Algoritmos de Codificação e Decodificação para Retículos",Dissertação de Mestrado,UFPE.

[20]-H.M. de Oliveira and G. Battail,"Performance of Lattice Codes over the Gaussian Channel",*Ann. Télécomm.*, **47**,n.7-8,293-305,1992.

[21]- M.A.O. Costa e Silva, R.Palazzo Jr.,"A Bounded Distance Decoding Algorithm for Lattices Obtained from a Generalized Formula",*IEEE Trans.*,**IT40**, 2075-82,1994.