UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ELETRÔNICA E SISTEMAS
COORDENAÇÃO DE MESTRADO

ALGORITMOS DE CODIFICAÇÃO E DECODIFICAÇÃO PARA CÓDIGOS DE RETÍCULOS

ATEF IBRAHIM IRSHAID SHARI'A

Dissertação apresentada à Pós-Graduação em Engenharia Elétrica da UFPE como parte dos requisitos para a obtenção do Título de Mestre em Engenharia Elétrica

Orientador: Prof. Dr. Hélio Magalhães de Oliveira

RECIFE PE - BRASIL
1994

SERVIÇO PÚBLICO FEDERAL

UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA
COORDENAÇÃO DO MESTRADO EM ENGENHARIA ELÉTRICA

PARECER DA COMISSÃO EXAMINADORA DE DEFESA DE
TESE DE MESTRADO
DE

ATEF IBRAHIM IRSHAID SHARI'A

T í T U L O

"ALGORITMOS DE CODIFICAÇÃO E DECODIFICAÇÃO *i:* LlA CÓDIGOS DE RETÍCULOS"

A Comissão Examinadora composta pelos professores : HÉLIO MAGALHÃES DE OLIVEIRA, DES/UFPE, RICARDO MENEZES CAMPELLO DE SOUZA, DES/UFPE e WALTER GODOY JÚNIOR, CEFET/PR, sob a presidência do primeiro, consideram o candidato ATEF IBRAHIM IRSHAID SHARI'A APROVADO COM DISTINÇÃO.

Recife, 28 de fevereiro de 1994

HÉLIO MAGALHÃES DE OLIVEIRA

RICARDO MENEZES CAMPELLO DE SOUZA

WALTER GODOY JÚNIOR

DEDICO ESTE TRABALHO

MINHA ESPOSA ALLANA.

AGRADECIMENTOS

Ao professor Hélio Magalhães de Oliveira pela orientação, tornando possível a realização deste trabalho.

Aos professores Mareia Campei1o, Ricardo Campello e Valdemar da Rocha Júnior por todos os ensinamentos recebidos no decorrer do mestrado.

Aos colegas de mestrado, em especial o amigo Hermano Cabral, pela colaboração.

A Coordenação do Mestrado em Engenharia Eletrônica.

A CAPES pelo auxilio concedido sob a forma de Bolsa de Estudo.

RESUMO

Os retículos constituem uma das técnicas da codificação de canal, onde o código é projetado de forma interligada à modulação em um processo conhecido como *"modulação codificada"*. Desde o monumental trabalho de SHANNON, ficou clara uma estreita relação entre a transmissão em altas taxas e a construção de estruturas densas em espaços com alta dimensionalidade. De Buda e De Oliveira-Battail demonstraram independentemente a existência de retículos que podem atingir a capacidade em um canal gaussiano. A dificuldade na implementação de retículos reside em três operações: O mapeamento das seqüências binárias nos pontos do retículo, a decodificação de vetores ruidosos em pontos do retículo, e o demapeamento dos pontos do retículo em seqüências binárias. A maioria dos algoritmos propostos a *priori* são dedicados ao segundo processo, ou são destinados a um retículo particular. Partindo de uma idéia simples, a decodificação via baricentros estabelecida por .De Oliveira para constelações bi-dimensionais, apresenta-se como um processo completo de codificação e decodificação de retículos obtidos a partir de qualquer construção código. Neste algoritmo, o problema de (de)codificação é reduzido a um problema de (de)codificação de códigos binários. Além disto, o problema de decodificação e demapeamento reduz-se a um único problema: a decodificação de vetores com ruído diretamente em seqüências binárias. O desempenho deste algoritmo é avaliado por simulação Monte Carlo e comparado com outros (e.g. máxima verossimilhança) . Conclui-se que, para os retículos obtidos pela construção código A, este algoritmo é de máxima verossimilhança, e para as demais construções, ele é de decodificação por "distância cotada".

ABSTRACT

Lattices are one of channel coding techniques where modulation and coding are jointly designed in an approach known as Coded Modulation. Ever since *Shannon's* monumental work, it was clear that a strong relationship exists between achieving high transmission rates and the construction of dense structures in high-dimensional spaces. *De Buda* and *De Oliveira-Battail* have proved the existence of lattices which can achieve capacity in Gaussian channels. The difficulties in lattice implementations are concerned with mapping binary sequences into lattice points, decoding noisy vectors in lattice points, and demapping points into binary streams. Most algorithms conceived until now focus just the second point, or deal with a specific lattice. Starting with a simple idea, namely Baricenter Decoding for two-dimensional constellations, we introduce coding and decoding processes of lattices obtained from any code construction. In such algorithms the problem of (de)coding is reduced to a problem of (de)coding binary codes. Furthermore, decoding and demapping are condensed into a single problem: Decoding noisy signal vectors directly into binary sequences. The performance of this new algorithm is evaluated by Monte Carlo simulation and compared with other different decoders (e.g. maximum likelihood). We conclude that, for lattices built from construction A, this algorithm is maximum likelihood, and for those derived from any other generalized construction, it is a bounded distance one.

RÉSUMÉ

Les réseaux sont des techniques de codage du channel où les codes sont projetés d'une façon liée à la modulation dans une approche nommée de modulation codée. Dès les travaux magnifiques de *Shannon,* il a été clair l'étroite relation entre l'obtention de débits élevés et la construction de structures denses dans une espace à haute dimension. *De Buda* et *De Oliveira-Battail* ont démontré, independentemment, l'existence de réseaux qui peuvent atteindre la capacité d'un channel gaussien. Les difficultées de mise en œuvre des réseaux concernnent trois tâches: Le *mapping* (application) de séquences binaires dans des points du réseau, le décodage d'un vecteur bruité dans un point du réseau, et le *demapping* de points dans des trains binaires de données. La plupart des algorithmes propposés auparavent ont été dédiés au deuxième processus, ou ne sont valables que pour un réseau particulier. D'après une idée assez simple, le décodage par baricentres établie pour *De Oliveira* dans le cadre des jeux de signaux bidimensionnels, on a aboutit à un algorithme de codage/décodage de réseaux conçues à partir d'une *formule code* quelconque. Dans cet algorithme, le problème de (dé)coder est remit au problème de (dé)codage de codes binaires. D'allieurs, le problème de "décodage suivi de *demapping"* est condensé dans un procédé unique: le décodage du signal bruité directement dans des données binaires. Les performances de ce nouveau algorithme sont évaluées par simulation sur ordinnateur et comparées avec celles d'autres méthodes de décodage (e.g., maximum de vraisemblance). On montre que, pour des réseaux obtenues pour la construction A, cet algorithme est de maximum de vraisemblance, alors qu'il s'agit de décodage borné en distance pour les réseaux conçues à partir d'autres constructions généralisées.

INDICE

INTRODUÇÃO 1

.1 - INTRODUÇÃO 1

1.2- EMBALAGEM DE ESFERAS EM UM SISTEMA DE COMUNICAÇÃO 6

1.3- MODULAÇÃO CODIFICADA 22

EMBALAGEM DE ESFERAS E RETÍCULOS 27

2.1- INTRODUÇÃO 27

2.2- REPRESENTAÇÃO MATEMÁTICA DE ESFERAS 28

2.3- EMBALAGENS RETICULADAS 29

2.4- DEFINIÇÕES 30

2.5- EXEMPLOS DE RETÍCULOS 34

2.6- CONSTRUÇÃO CÓDIGO DE RETÍCULOS 40

2.7- RETÍCULOS E CODIFICAÇÃO 46

INTRODUÇÃO AOS MÉTODOS DE CODIFICAÇÃO E DECODIFICAÇÃO PARA RETÍCULOS 63

3.1- MÉTODOS GERAIS DE DECODIFICAÇÃO 65

3.2- MÉTODOS ESPECIAIS DE CODIFICAÇÃO E DECODIFICAÇÃO 68

3.3- MÉTODOS DE DECODIFICAÇÃO SUAVE PARA CÓDIGOS BINÁRIOS 76

NOVOS ALGORITMOS PARA CODIFICAÇÃO E DECODIFICAÇÃO PARA RET1CULOS 84

4.1- INTRODUÇÃO

4.2- DECODI FI CAÇÃO

4.3- CODIFICAÇÃO

DISCUSSÃO E SUGESTÕES

5.1- COMPARAÇÃO

5.2- DISTÂNCIA MÍNIMA

5.3- COMPLEXIDADE

5.4- DESEMPENHO

5.5- MODULAÇÃO CIFRADA

CONCLUSÕES

APÊNDICE A

APÊNDICE B

APÊNDICE C

REFERÊNCIAS BIBLIOGRÁFICAS

CAPÍTULO 1

INTRODUÇÃO

1.1 - INTRODUÇÃO

Nas comunicações digitais confiáveis, um dos objetivos maiores do projetista é a construção de uma lista de palavras código, as quais podem ser transmitidas simultaneamente com confiabilidade máxima e potência mínima. Cada palavra código pode ser representada por um vetor de n dígitos, onde cada um destes pode assumir q valores. Por exemplo, n pode ter o valor 8, ou seja, cada palavra possui oito dígitos, e cada um dos dígitos pode assumir um dos valores {0,0.5,1,-1,-0.5}, ou seja, q=5. Em princípio, este sistema pode gerar 5^8 palavras código diferentes. Como a diferença entre diversos pares destas palavras é muito pequena, tal sistema é vunerável ao ruído do canal e pode estar sujeito a um grande número de erros aleatórios na transmissão. Por exemplo, a diferença entre a palavra (1,1,1,1,1,1,1,1) e a palavra (1,1,1,1,1,1,1,0.5) é tão pequena que, se as duas palavras forem usadas em canal ruidoso, existe uma grande possibilidade, em função

do nível do ruído, de que as duas sejam confundidas. Uma maneira alternativa de se colocar o problema é: se a diferença entre duas palavras é tão pequena como no caso das duas palavras acima, uma grande potência é necessária para garantir que as duas palavras possam ser distinguidas em presença de ruído.

A relação matemática entre a distinção das palavras código e a potência requerida para transmiti-las com confiabilidade arbitrariamente fixada foi brilhantemente formulada por Shannon [1] em 1948.

TEOREMA (1:1) (CODIFICAÇÃO DE CANAL)

Seja P a potência média de transmissão, e suponha que o ruído é aditivo, Gaussiano e branco de potência N na banda W. É possível por um sistema de codificação suficientemente complicado, transmitir dígitos em uma taxa

(1:1) $C = W \log \frac{P}{N}$ bits/seg,

com freqüência de erros tão pequena quanto se deseje. Em contrapartida, não é possível, por qualquer método de codificação, transmitir em uma taxa maior e ainda se ter uma freqüência de erros arbitrariamente baixa.

novo eixo (y^c^) a fim de realizar a primeira etapa da segunda fase, e nos eixos (y -c), i = 2,3,...,n, para a realização do resto das etapas desta fase.

TEOREMA 4:1. Suponha que após a (j-l)-ésima fase da divisão do retículo Z^n, temos o centro $(c_1, c_2, \ldots, c_n)$ como centro do grupo que contém o vetor recebido, então o centro do grupo escolhido após a k-ésima etapa da j-ésima fase é uma das combinações $(c_1 \pm 2^{m-j}, c_2 \pm 2^{m-j}, \ldots, c_k \pm 2^{m-j} > c_{k+1} \ldots c_n)$ onde **k. = 1**,2,....**n.**

•

PROVA. Como notamos na prova do lema (4:1), consideremos o caso de um eixo Z. Na primeira fase, as coordenadas de Z foram divididas em dois subconjuntos: negativas e positivas. Cada subconjunto tem 2^{m-1} coordenadas. Na segunda fase, cada subconjunto foi dividido em mais dois subconjuntos com relação aos centros $c^1 = \pm 2^{m-1}$. Cada novo subconjunto tem 2^{m-2} coordenadas da constelação. Em geral após a j-ésima fase da divisão, as coordenadas serão divididas em 2^j subconjuntos, cada subconjunto tendo 2^{m-j} coordenadas, as quais são exatamente: C **±1**, $c^j \pm 3, \ldots, c^j \pm (2^{m-j*1} - 1)$, onde c^j é a média aritmética das coordenadas de cada subconjunto que está atualmente no processo de divisão. As coordenadas de cada subconjunto serão divididas em relação ao centro c^j, concluindo dois subconjuntos: o primeiro

tendo os valores $c^j+1, c^j+3, \ldots\ldots, c^j+(2^{m-j+1}-1)$; e o segundo tendo os valores $c^j-1, c^j-3, \ldots\ldots, c^j-(2^{m-j+1}-1)$. O centro de cada subconjunto é c^{j+1}, onde

$$C^{J+1} = \frac{c^j+1 + c^j+3 + \ldots\ldots + c^j+(2^{m-j+1}-1)}{2^{m-j}}$$

$$c = c + \frac{1+3+\ldots\ldots+(2^{m-J+1}-1)}{2^{m-j}}$$

Como

$$1+3+5+\ldots\ldots+(2n-3)+(2n-1)=n^2,$$

fazemos

$$2^{m-j+1}-1 = 2n-1,$$

$$n = 2^{\quad},$$

Então:

$$(4:2) \qquad c^{j+1} = c^j + \frac{2^{2(m-j)}}{2^{B'j}} = c^j + 2^{m-j}.$$

Q.E.D.

COROLÁRIO 4:1 Os centros dos pontos de cada grupo concluído após a j-ésima etapa da divisão de qualquer retículo, onde $j=1,2,3,\ldots,m$, são da forma:

(4:3) $C^{J} = B_{m-l}\,2^{m}\sim' + B_{m-2}\,2^{m}\sim^{2} + \ldots + B_{m-J}\,2^{*}\sim>,$

onde $B_k = (h^{\wedge}, \ldots .'^{b}{}_{n}{}^{\wedge}{}'$ sendo $b^{\wedge} - \pm 1$.

COROLÁRIO 4:2 Qualquer ponto do retículo A_n pode ser definido da seguinte forma:

(4:4) $P = B_{m-1}\,2^{m}{}''^{1} + B_{m-2}\,2^{m}{}''^{2} + \ldots + B_{1}\,2^{*} + B_{0}\,2^{\circ},$

onde $j = l, 2, 3, \ldots, m$, e $B_k = (b_j, \ldots .^{/b}{}_{n}{}^{\wedge}{}'$ sendo $b_i = \pm 1$

DEFINIÇÃO 4:3- Vamos definir os centros que se apresentam sob a forma

(4:5) $C^{2} := B_{m-1}\,2^{a'1*} B_{m-2}\,2^{B}{}''^{2} + \ldots + B_{3}\,2^{3} + B_{2}\,2^{2},$

como centros pré-finais. Claro que;

$C^{2} = 0 \pmod 4$.

DEFINIÇÃ*0 4:4- Vamos definir os centros que se apresentam sob a forma:

(4:6) $C^{1} := B_{m-1} 2^{m-1} + B_{m-2} 2^{m-2} + \ldots + B_{2} 2^{2} + B_{1} 2^{1}$,

como centros finais. Claro que,

$$C^{1} = 0 \pmod 2 .$$

DEFINIÇÃO 4:5- Vamos definir os centros que se apresentam sob a forma:

(4:7) $C^{m-j} = B_{m-1} 2^{m-1} + B_{m-2} 2^{m-2} + \ldots + B_{m-j} 2^{m-j}$,

onde $j=1,2,3,\ldots\ldots(m-3)$, e $B_k = (b_i, \ldots\ldots b_0)$ sendo $b_i = \pm 1$-

como centros iniciais. Claro que,

$$C^{m-j} = 0 \pmod{2^{n-j}} .$$

A tabela (4:1) mostra os valores das coordenadas de cada tipo destes centros.

Podemos usar esta simplificação na estrutura dos centros e esta representação dos pontos do retículo para propor um sistema de codificação e decodificação simples.

Tabela (4:1). Coordenadas dos centros definidos anteriormente

m	centros iniciais	centros pré-finais	centros finais	constelação
1				±1
2			±2	±1, ±3
3		±4	±2, ±6	±1,±3,±5±7
4	±8	±4,±12	±2,±6,±10,±12	±1,±3,±5,±7 ±9,±11,±13, ±15
5	±8,±16 ±24	±4,±12,±20 ±28	±2,±6,±10,±14 ±18,±22,±26, ±30.	±1,±3,±5,±7 ±9,±11,±13, ±15,___,±29, ±31.

4:2) **DECODIFICAÇÃO.**

É fácil observar que a equação (4:4) é a representação simbólica da construção código de retículos. Portanto, partindo das simples idéias da composição de pontos do retículo por centro de gravidade, reencontramos uma relação com a fórmula código do retículo. Para os retículos obtidos aplicando-se a construção A, o vetor $\mathbf{B}_q$ deve ser uma palavra de um código $\mathbf{C}_q$ definido para cada retículo A_n, onde

$$A_n = 2Z^n + \mathbf{C}_o(n, k_o, d_o),$$

sendo $TL = \pm1, \pm3, \ldots, \pm(2^{n} - 1)$. Para retículos obtidos aplicando-se a construção B, os vetores $\mathbf{B}_0$ e $\mathbf{B}_q$ devem ser palavras de dois códigos $\mathbf{C}_0$ e $\mathbf{C}_q$, respectivamente, definidos para cada retículo Λ, onde

$$\Lambda = 4Z^{n} + 2\mathbf{C}_1(n, k_1, d_1) + \mathbf{C}_0(n, k_0, d_0),$$

sendo $1 = \pm1, \pm3, \ldots, \pm(2^{n} - 1)$. Em geral, para retículos obtidos aplicando-se a construção código generalizada, os vetores $\mathbf{B}_0$, $\mathbf{B}_1, \ldots$, e $\mathbf{B}_{r-1}$ devem ser palavras código dos códigos $\mathbf{C}_0$, $\mathbf{C}_1, \ldots$, e $\mathbf{C}_{r-1}$, onde

$$\Lambda = 2^{r}Z^{n} + 2^{r-1}\mathbf{C}_{r-1} + \ldots + 2\mathbf{C}_1 + \mathbf{C}_0.$$

sendo $1 = \pm1, \pm3, \ldots, \pm(2^{m} - 1)$. Vamos discutir o sistema de decodificação para cada construção.

4:2:1) CONSTRUÇÃO A.

Para retículos obtidos pela aplicação desta construção, o último termo da equação (4:4) é redundante e deve ser uma palavra de um código definido para cada retículo. A decodificação é constituída das seguintes etapas:

1 - redução do vetor recebido **R** a um vetor $\mathbf{R}_q$ e [-2,2], onde

$\mathbf{R} = \mathbf{R}_q \pmod{2Z^n}$,

2- decodificação de $\mathbf{R}_q$ em uma palavra $\mathbf{B}_q$ do código V

3- Subtração de $\mathbf{B}_q$ do vetor **R**. O resultado $\mathbf{R}-\mathbf{B}_0$ s decodificado em um centro final;

$C = \mathbf{B}_{m-1} 2^{m-1} + \mathbf{B}_{m-2} 2^{m-2} + \ldots + \mathbf{B}_2 2^2 + \mathbf{B}_1 2^1$.

4- O ponto $\mathbf{P} = C + \mathbf{B}_q$ é o ponto decodificado.

A redução de R a $R_q \pmod{2Z^n}$ é realizada de acordo com o processo na figura (4:1).

Tendo em vista esta discussão, o seguinte algoritmo encontra o ponto do retículo $A^{\wedge} = 2Z^n + C_0(n,k,d)$ mais próximo do vetor recebido $R = (r_1, \ldots, r_n)$ e R^n, (veja figura 4:2).

Algoritmo C4:l).

faça i = m-1, $R = C_t r_{vi}, \ldots, r_{in}$ D= R

define o vetor $o = <q, \ldots$ %?" I R I

etapa 1

$B_i = sgn(R_i) D$, onde sgn(R) sinal R ;

se $|r_i| < q_i$, então, $q_i = |r_i|$, para j= 1 . . . . n,

se i= 0, vá à etapa 3.

etapa 2 :

$\mathbf{R} = \mathbf{R} - 2^i * B$; i = i - 1 ;

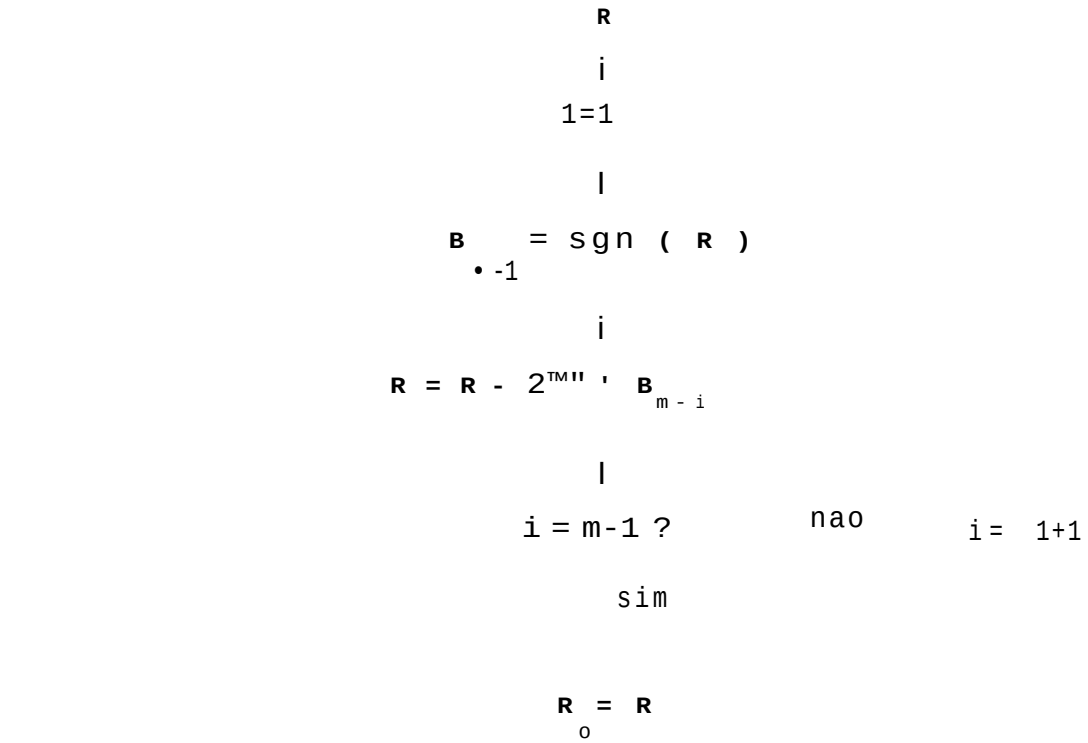


Figura (4:1) Reduzir **R** a $\mathbf{R}_q$ (mod $2Z^n$)

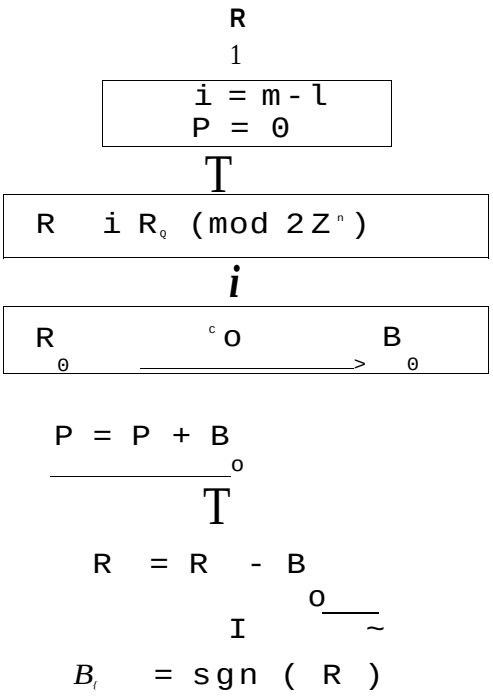


Figura (4:2) Algoritmo para decodificação de retículos aplicando a construção código A.

etapa 3 :

3:1D decodifique o vetor $R = B * Q$ em uma palavra do código $C^{Crwk.dD}$ usando um dos métodos de decodificação suave ótima, gerando um novo vetor B.

3:2,) $R = R - B$

etapa 4

faça $i = m - l$; $R^{\wedge} = R$ (condição inicial)

4:1) $B_i = sgn(R_i)$, onde $sgn(R_{i,J}) := sinal\ R_{i,J}$;

se i= 1, vá à etapa (4:3).

4:2) $R_{ti} = R_{,} 2^{l} * B_{,}$;

$i = 1 - 1j$

volte à etapa (4:1);

4:3) o ponto

$$P = B_i 2^{*4} B_{i-1} 2^{i-1} + \ldots + B_1 2^{*} + B_0 2^{\circ},$$

é o mais próximo do vetor recebido R.

A última etapa (4:3) pode ser anulada no caso que queremos mapear o vetor recebido em uma seqüência binária correspondente ao ponto **P,** como veremos no algoritmo de codificação. Exemplos dos retículos D_4 e E_8 serão dados após uma discussão da construção B.

É bem conhecido que esta taxa crítica de transmissão é chamada de "capacidade do canal de transmissão".

Infelizmente, o teorema de Shannon não mostra como construir tal sistema, embora confirme sua existência. Tal sistema requer uma potência cerca de **9** dB inferior a um sistema não codificado com a mesma freqüência de erros [2]. Muitos sistemas de codificação foram construídos, contudo nenhum deles tem se aproximado das promessas de Shannon. Como veremos na discussão dos sistemas de modulação codificada, o. sistema mais complexo e prático pode diminuir a potência de transmissão em cerca de 6 dB.

Uma maneira de se projetar um conjunto de sinais que se aproxime dos padrões prometidos no teorema de Shannon é representar cada sinal ou palavra de n dígitos como um ponto em um espaço de n dimensões. Por exemplo, considere uma seqüência qualquer de oito números pertencente ao sistema de sinais anterior. Fisicamente, cada número corresponde a um nível de voltagem na linha de transmissão, e assim cada palavra pode ser mapeada em um gráfico bidimensional como um série de oito pulsos distintos cujas alturas são especificadas ao longo do eixo do tempo. Matematicamente, um ponto em um espaço de oito dimensões pode representar a mesma informação que esta série possa ter. Suponha que o primeiro número de cada seqüência seja o valor da primeira coordenada do ponto, e o segundo número seja a segunda coordenada, e assim por diante. Desde que cada ponto em oito dimensões é determinado pela fixação dos valores de todas as oito coordenadas,

4:2:2) CONSTRUÇÃO B.

Em retículos obtidos aplicando-se a construção **B,** onde A_n é um retículo se e somente se $A_n = 4Z^n + 2C_1 + C_0$, ambos B_1 e B_q, na equação (4:4), devem ser palavras dos códigos e C_q respectivamente. Como revela a equação (4:4), o processo de decodificação envolvendo o termo B_j é uma função de B_q e vice-versa, portanto não há a possibilidade de decodificar nenhum deles sem conhecer o outro. Como para um vetor recebido R, ambos são desconhecidos, então uma maneira de se obter a decodificação por máxima verossimilhança é passar em todas as palavras do código C_q decodificando o vetor

$$R = R - (B_1 2^* + B_{i-1} 2^{i-1} + \ldots + B_o 2^\circ) \qquad (4:13)$$

em uma palavra do código $C^\wedge$, onde B_q é uma palavra do código C_q. Para cada combinação de B_q e B_I, encontramos um ponto decodificado e escolhemos aquele que tem a menor distância do vetor R.

Ora, passar através de todas as palavras do código C_q torna o processo de decodificação complexo e lento. A outra alternativa é decodificar o vetor R_q em uma palavra mais próxima de C_q, e a partir desta palavra decodificar o vetor R_j definido na equação (4:13) em um vetor mais próximo em . Tal redução não altera significantemente o desempenho do algoritmo, que ainda é virtualmente equivalente àquele de decodificação por máxima

verossimilhança, como mostraremos para o retículo Λ_{16} * T a l resultado é confirmado em **[22]**.

A partir desta simplificação, propomos o algoritmo seguinte que decodifica um vetor recebido R = (r , ..., r j e Rⁿ em um ponto no retículo $A_n = 4Z^n + 2C_1 + C_0$ (veja figura 4:3). Resultados de simulação indicam que a probabilidade de que este ponto não seja o mais próximo de R é quase zero.

Algoritmo C4:2).

faça i = m-1 ; R=C r. , . . , r D= R; g = O,

define o vetor Q • Cq . . ,q |R|

etapa **1** :

B. = sgnCR.D, onde sgnC R. } : « sinal R

se |r..|<q., entSo, q. = Ir. I, para j= l, . . . , n

se i= g, vá à etapa 3.

etapa 2 :

$R_{i-} = R - 2^{*} B$;

i = i - 1 ; volte à etapa 1,

etapa 3 :

3: 12) decodifique o vetor R = CQ* B D/2 em uma palavra do código C Cn.k.dD usando um dos métodos de decodificação suave ótima, gerando um novo vetor B ;

3:25 R = CR - 2 * B 5/2.

se g=l vá á etapa 4,

g=i.

i = m-1; ***R=*** R_j ***O. • »***

vá à etapa 1

etapa 4 :

faça i = m-1; ***R^= R*** (condição inicial)

4:1) B = sgn(R) . onde sgn(R) := sinal R ;

se i= 2, vá à etapa (4:3).

4:2) R_t j = R_4- 2^l* B_j;

i = i-l;

volte à etapa.(4:1);

4:3) O ponto

P = B ***2^*** B 2^{i-1}+ . . . .+ B ***2^+*** B 2^0,
i i-1 í o

é o ponto decodificado.

Como na construção A, a última etapa (4:3) pode ser eliminada caso desejemos decodificar diretamente em uma seqüência binária que corresponde ao ponto P.

4:2:3) CONSTRUÇÃO CÓDIGO GENERALIZADA

O algoritmo (4:2) pode ser generalizado para decodificar qualquer retículo obtido usando-se a construção código generalizada [22]. Suponha que o retículo A_n tem a fórmula código seguinte

(4:15) $A = 2^g Z^n + C_{g-1} 2^{g-1} + \ldots + C_1 2^1 + C_0 2^0$,

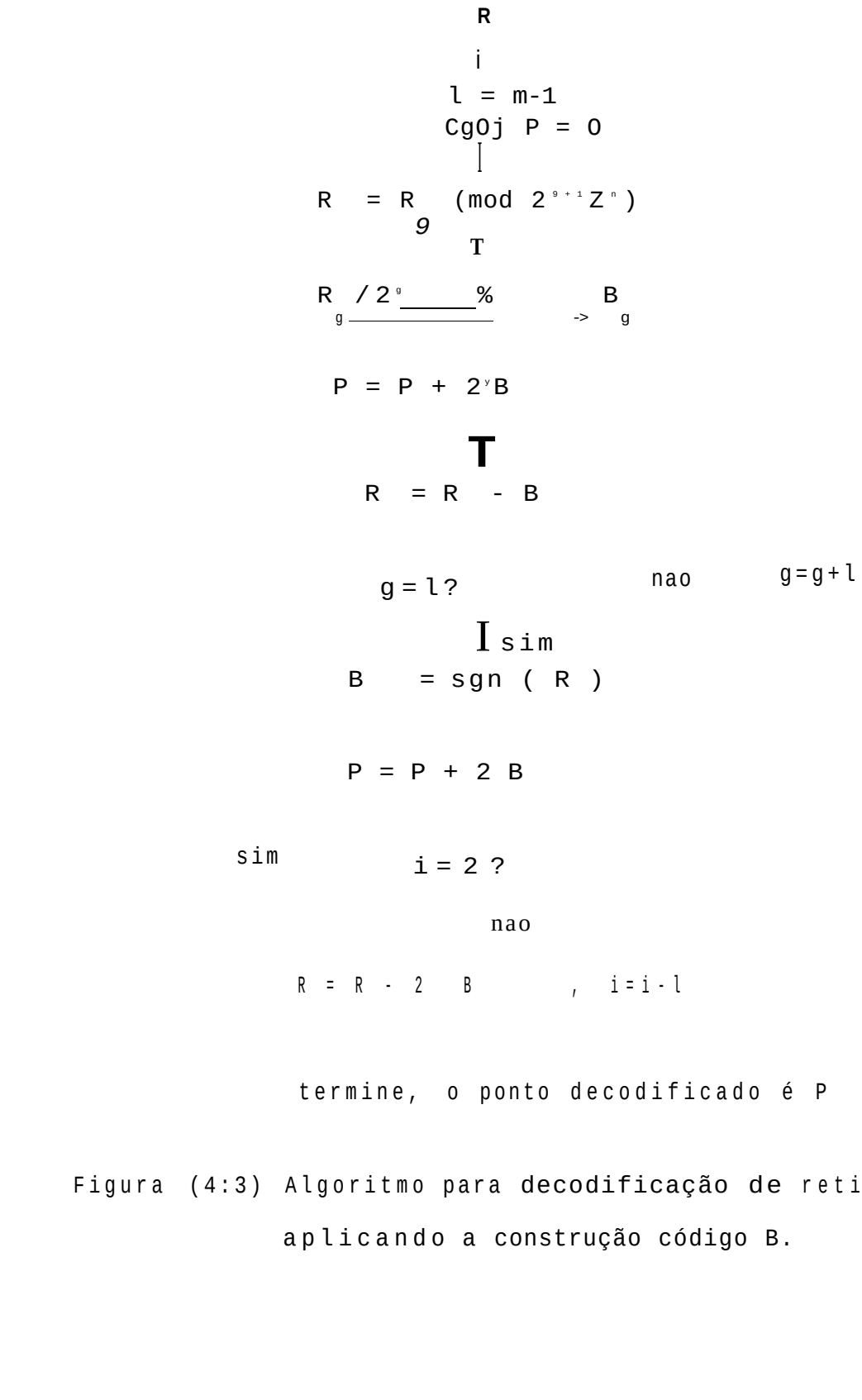


Figura (4:3) Algoritmo para decodificação de reticulos aplicando a construção código B.

onde C_t é um código de bloco, $C_t \subset C_{t-1} \subset C_L$. O algoritmo seguinte codifica um vetor R e R^n em um ponto no retículo A_n Cveja figura 4:45.

Algoritmo C4:35.

faça i = m-1 ; R =C r_{ii}. . . . ,r 5= R; j = O,

define o vetor Q = Cq. . . . .q)= |R|

etapa **1**

B_t = sgnCR $_i$) , onde sgnC $R_{t,j}$ 5 : •» sinal $R_{V.j}$.;

se |r..|<q., então, q. = |r..|, para j = l,...,n

se i= j, vá à etapa 3.

etapa 2

R_{L-1} = R_l - 2^l* **B** $_t$;

i = i - 1 ;

volte á etapa 1

etapa 3 :

3:15 decodifique o vetor R_J= CQ* **B** $_J$ 5/2^j em uma palavra do código C.Cn,k,d5 usando um dos métodos de decodificaçSo suave ótima, gerando um novo vetor **B^;**

3: 25 R = CR - 2^j* **B** $_J$ 5 ,

se j = g-1 vá à etapa 4,

j - j + i .
i = m-1 ; ***R*=** R, Q= R

vá à etapa 1

etapa 4 :

faça i = m-1; R" R (condição inicial)

4:1) B= sgn(R), onde sgn(R_t) := sinal R ;

se i= g, vá à etapa (4:3).

4:2) $R_{i-1} = R_i - 2^i * B_i$;

i = i-l;

volte à etapa (4:1);

4:3) O ponto

$P = B_i 2^i * B_{i-1} 2^{i-1} + \quad + B_1 2^* + B_0 2^\circ$,

é o ponto decodificado.

4:2:4) EXEMPLOS

1) O RETÍCULO D_4

Este retículo pode ser obtido aplicando-se a construção A,

$D_n = 2Z^n + C_0(4,3,2)$,

onde o código C_0 é um código de um único dígito de paridade, o qual pode ser decodificado pela troca de sinal da coordenada menos confiável se o teste de paridade falhar.

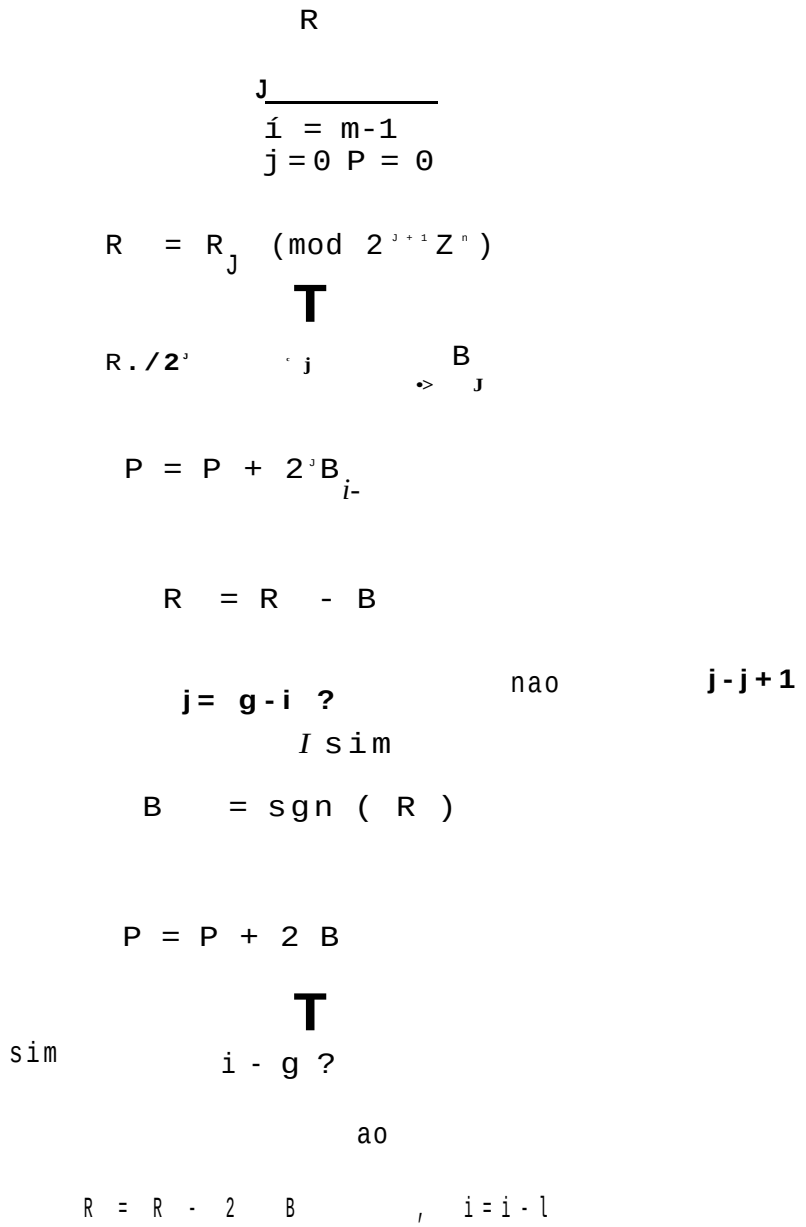


Figura (4:4) Algoritmo para decodificação de reticulos aplicando construção código generalizado.

EXEMPLO 4:3

Suponha que este retículo seja construído usando-se uma constelação bidimensional de 2^8 pontos (m = 4). Queremos, por exemplo, decodificar o vetor

R= (12.8. -8.9, 0.5, -3.7),

em um ponto mais próximo. As etapas 1, 2, e 3 do algoritmo (4:1) anterior resuitam em :

1	R /B			
3	R3	12.8,	-8.9,	0.5, -3.7
	Ba	1	-1	1 -1
2	R2	4.8	-0.9	-7.5 4.3
	B2	1	-1	-1 1
1	Ri	0.8	3.1	-3.5 0.3
	Bi	1	1	-1 1
0	Ro	-1.2	1.1	-1.5 -1.7
	Bo	- 1	\	-1 -1

A palavra do código $C_oC4_r3,a5$ mais próxima ao vetor R_o =Q = C-.8, .9. -.5 , -. 3D é B = C -1 , +1 , -1 , +15. A etapa quatro do mesmo algoritmo resulta em:

1		
3	R3	13.8, -9.9, 1.5, -M.7
	Ba	1 -1 1 -1
2	R2	5.8 -1.9 -8.5
	B2	1 -1 -1 1
1	Ri	1.8 2.1 -2.5 -l.rj
	Bi	1 1 -1 -1

Portanto o ponto decodificado é:

$$P = B_3 2^3 + B_2 2^2 + B_1 2^* + B_0 2^\circ$$

$$= (1,-1,\ 1,-1)^* 2^3 + (1,-1,-1,1)^* 2^2$$

$$+ (1,1,-1,-1)^* 2^* + \quad 1)^* 2^\circ.$$

$$= (\ 13,-9,\ 1,-\ )$$

2) O RETÍCULO E_8

Este retículo pode ser obtido aplicando-se a construção A,

$$E_8 = 2Z^8 + \mathbf{C}\ (8,4,4),$$

onde o código $\mathbf{C}_q(8,4,4)$ é um código de Reed-Muller de Iª ordem, o qual pode ser decodificado usando-se o método apresentado no capítulo anterior.

EXEMPLO 4:4

Suponha que este retículo seja construído a partir

de uma constelação bidimensional de 2 pontos (m = 4). Queremos, por exemplo, decodificar o vetor

R = (12.8, -8.9, 0.5, -3.7, 5.8, -7.3, -11.1, 2.2)

em um ponto mais próximo. As etapas 1, 2 e 3 do algoritmo (4:1) anterior resultam em :

1		R /B							
3	R_9	12.8,	-8.9,	0.5 ,	-3.7,	5.8,	-7.3,	-11.1,	*2.2*
	B_3	1	-1	1	. -1	1	-1	-1	1
2	R_2	4.8	-0.9	-7.5	4.3	-2.2	0.7	-3.1	-5.8
	B_2	1	-1	-1	1	-1	1	-1	-1
1	R_1	0.8	3.1	-3.5	0.3	1.8	-3.3	0.9	-1.8
	B_1	1	1	-1	1	1	-1	1	-1
0	R_0	-Q.8	o.q	*-0.6*	-0.3	*-0.2*	-0.3	*-o.q*	*o.l*
	6o	- 1		- \					

Usando um decodificador para C (8,4,4), o vetor R é decodificado no vetor B_0 = (-1,1,-1,1,-1,1,-1,1). A etapa 4 do mesmo algoritmo resulta em:

1		R /B V_i							
3	R	13.8,	-9.9,	- Q.5,	-2.7,	4-8,		-10.1,	1.2
	B_3	1	-1	1	-1	1	-1	-1	1
2	R_2	5.8	-1.9	5	5-3	-3-2	- L i	-2.1	-6.8
	B_2	1	-1	1	1	-1	1	-1	-1
1	R_1	1.8	2.1	-3. 5	1.3	f .8		1.9	-1.8
	B_1	1	1	í	1	1	-1	1	-1

Finalmente o ponto decodificado é

cada palavra código do sistema pode ser representada como um ponto distinto em um espaço de oito dimensões.

Quando as palavras código são representadas como pontos em um espaço, dois aspectos importantes de um conjunto de palavras código podem ter uma interpretação matemática; primeiro, lembre-se que as palavras código devem ser distinguidas confiávelmente entre si. Em um espaço de oito dimensões, por exemplo, isto sugere que os pontos que representam as palavras código devam ser separados por uma distância mínima elevada. Como a distância Euclidiana entre dois pontos em um espaço de oito dimensões mede a distinção entre duas palavras código?. Para determinar a distância Euclidiana, a diferença entre os dois valores de cada coordenada elevada ao quadrado. Pequenas diferenças entre os valores das duas coordenadas (i.e. diferenças menores do que 1) serão reduzidas, enquanto diferenças maiores (i.e. maiores do que 1) serão ampliadas. Como as diferenças pequenas em níveis de voltagem são as mais prováveis causadoras de confusão entre palavras código, então a distância Euclidiana é uma medida razoável para a distinção entre sinais perturbados por ruído gaussiano [31].

Considere, por exemplo, as duas palavras (1,1,1,1,1,1,1,1) e (0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5). O quadrado da distância entre os dois pontos é a soma dos quadrados da diferença entre as coordenadas das duas palavras, cada um sob a forma $(1-0.5)^2$. Assim a distância entre os dois pontos é -I 2 .De acordo com a métrica Euclidiana, as duas palavras (1,1,1,1,1,1,1,1)

$$P = B_3 2^3 + B_2 2^2 + B_1 2^1 * B_0 2^0$$

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

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

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

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

$$= (13\ -9\ -1\ -3\ 7\ -5\ -11\ 3).$$

3) O RETÍCULO A_{16}

Este retículo pode ser obtido aplicando-se a construção B, onde sua fórmula código é

$$A_{16} = 2Z^8 + C_1(16,15,2) + C_0(16,5,8).$$

O **código C** (16,5,8) **é um código de** Reed-Muller **de** 1* **ordem, enquanto o código C (16,15,2) é um código de um único dígito de paridade.**

EXEMPLO 4:5

Suponha que este retículo seja construído usando-se uma constelação bidimensional de 2 **pontos (m =** 4) **. Queremos, por exemplo, decodificar o vetor ruidoso**

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)

em um ponto mais próximo.

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

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

Logo subtraímos o vetor **B** do vetor **R**, ou seja

R = R- B

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 (3:2) do algoritmo (4:2) resulta:

1	R /B r i							
3	R_3	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
	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
	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

um palavra no código C_1 (16,15,2).

A etapa 4 do mesmo algoritmo resulta:

		R /B r i
3	R_3	10.8, -5.9, 3.5, -4.7, 4.8, -4.3, -12.1, 5.2 - 4.4, 13.2, -12.3, -3.2, 3.3, 5.5, -4.1, -4.1
	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
	B_2	1 1 -1 1 -1 1 -1 -1 -1 1 -1 1 -1 -1 1 1

Portanto o ponto decodificado será

$$B\,2^{3} + B\,2^{2} + B\,2^{*} + B2^{\circ}$$

$$(1\ -1\ 1\ -1\ \ 1\ -1\ \ -1\ 1\ 1\ 1\ -1\ -1\ 1\ 1\ -1\ -1)\ 2^{3}$$

$$(1\ \ 1\ -1\ 1\ -1\ 1\ -1\ -1\ -1\ 1\ -1\ 1\ -1\ -1\ 1\ 1\ 1)\ \ 2^{2}$$

$$(1\ -1\ -1\ 1\ \ 1\ -1\ 1\ -1\ -1\ -1\ 1\ -1\ 1\ 1\ 1\ 1\ -1)\ \ 2^{1}$$

$$(\ -1\ -1\ -1\ -1\ -1\ -1\ -1\ \ 1\ \ 1\ 1\ \ 1\ 1\ 1\ 1\ 1)\ \ 2^{\circ}$$

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

4:3) **CODIFICAÇÃO**

Aplicamos os algoritmos anteriores de decodificação de retículos construídos utilizando-se constelações bidimensionais, para propor um algoritmo que mapeia diretamente o vetor ruidoso recebido em uma seqüência binária correspondente em um ponto mais

próximo daquele vetor.

Este algoritmo é baseado em um outro [6] que mapeia uma seqüência binária em um símbolo de uma constelação QAM senso estrito (quadrada), e demapeia um vetor R € IR² em uma seqüência binária que corresponde ao símbolo mais próximo da constelação.

4:3:1) CODIFICAÇÃO PARA CONSTELAÇÕES BIDIMENSIONAIS

4:3:1:1) MAPEAMENTO BIT A SÍMBOLO

Vamos considerar uma constelação bidimensional quadrada de M = 2^{2m} pontos, tal como na figura (4:5), com coordenadas sob a forma: ±1, ±3, ± 5 (2^{m} - l). Uma palavra binária de comprimento 2m é atribuída a cada ponto, como indica a figura (4:5), onde m = 3. Inicialmente, cada dois bits são acoplados como um vetor bidimensional b • $(b^{\wedge}_{1}$, k>$_{12}$) / definindo m vetores, de tal modo que a palavra binária b pode ser escrita como:

(4:16) b - (b , to ,_____, b to).

A conversão da informação binária em ponto de sinalização pode ser feita de acordo com o procedimento seguinte.

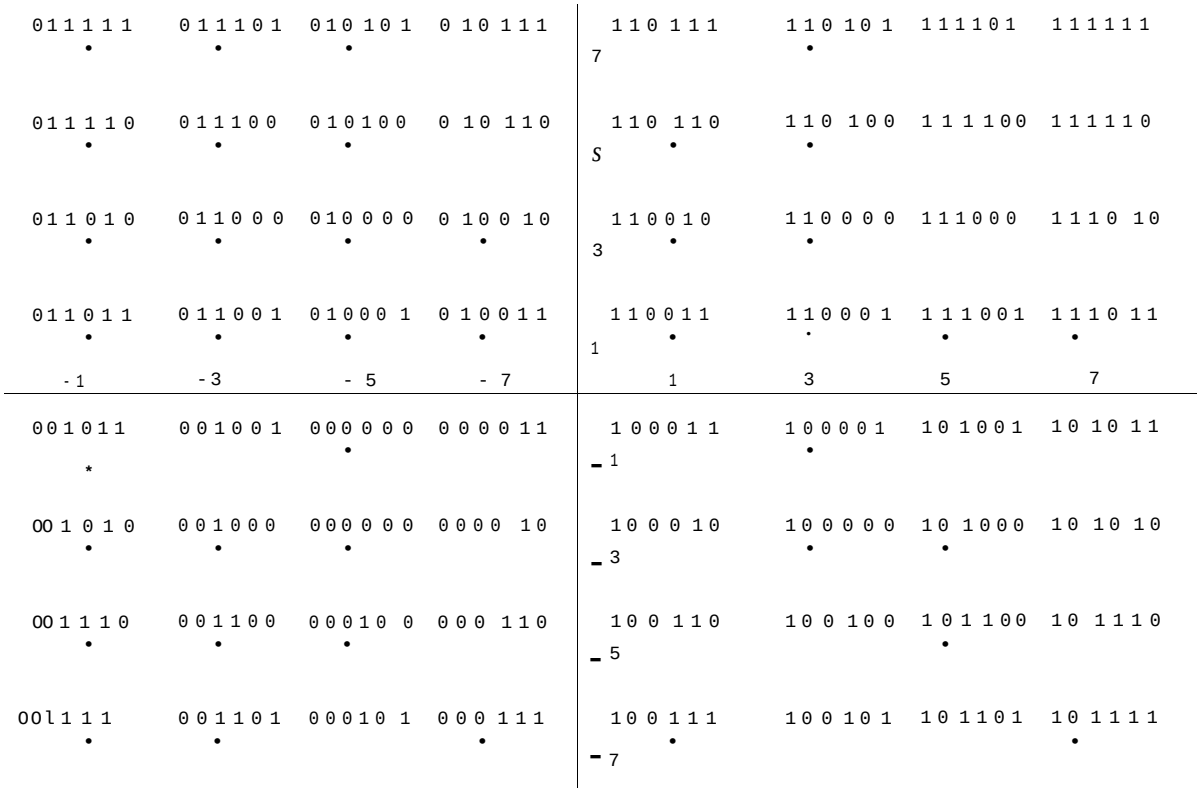


Figura (4:5). Esta figura mostra a atribuição de bit-a-símbolo de uma constelação de 64 pontos de acordo com [6].

Algoritmo (4:4)

etapa 1

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

(4:17) $B = (\ldots, b_2, b_1, b_0)$

etapa 2

O ponto da constelação é dado por :

(4:18) $P = B_{m-1} * 2^{m-1} + \ldots + B_{m-1} * B_{m-2} * \ldots * B_1 2^1 + B_{m-1} * B_{m-2} * \ldots * B_1 2^1 * B_0 2^0,$

onde * denota a multiplicação vetorial definida da seguinte maneira:

(4:19) $P_1 * P_2 := (x_1 x_2, y_1 y_2)$

onde $P_i = (x_i, y_i)$, $i = 1, 2$.

EXEMPLO 4:6

Consideremos uma constelação de M = 4 pontos (m = 3). Como exemplo, vamos encontrar o ponto da constelação associado à seqüência binária (100011). Temos :

$$B = (1\ -1\ -1\ -1\ 1\ 1) = (B_2, B_1, B_0)$$

assim :

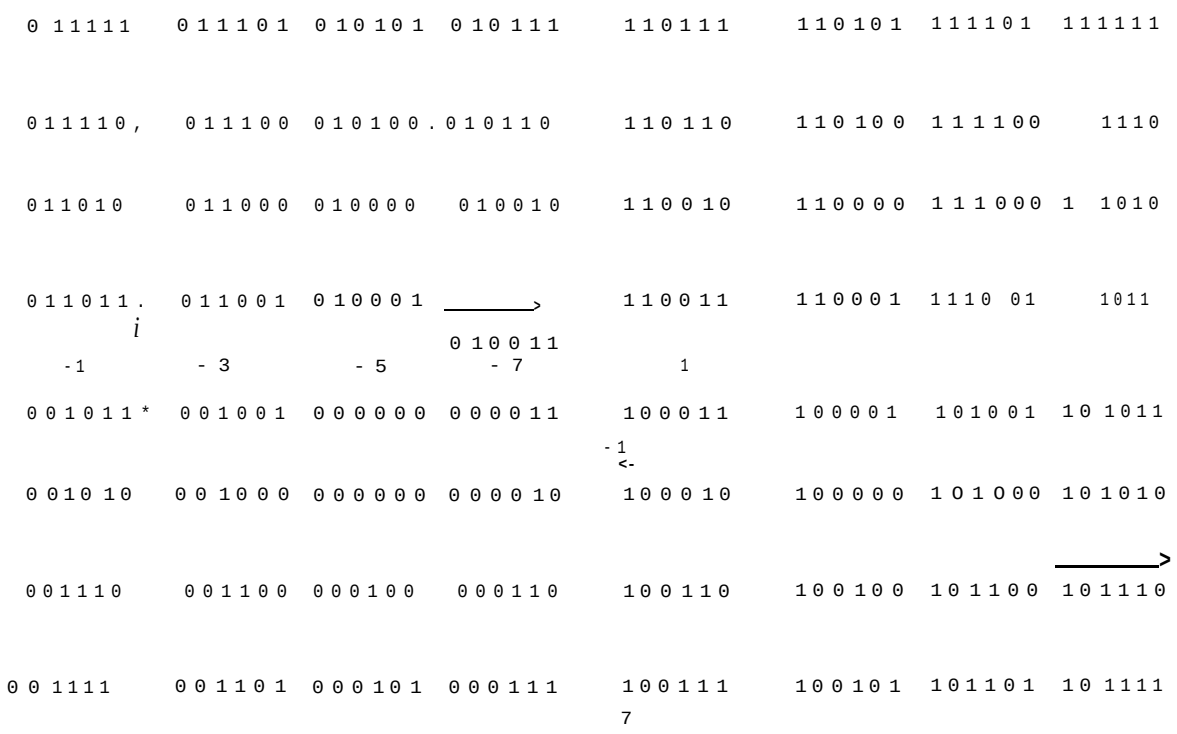
$$P = B_2 2^2 + B_2 * B_1 2^* + B_2 * B_1 * B_0 2^\circ$$

$$= (4, -4) + (-2, 2) + (-1, 1) = (1, -1),$$

como confirma a figura (4:5).

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 uma constelação de 4 pontos, cuja origem está 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 com a figura (4:6), onde a constelação de sinais é dividida em 4 constelações menores equivalentes. Partindo da origem do sistema de eixos, se o quadro cair na direção do arco horizontal, designamos 1 no primeiro bit, e 0 caso contrário. Da mesma maneira, o arco vertical define o segundo bit. Observe que os arcos sempre partem da origem, o que garante que a distância de Hamming entre os dois pontos mais próximos, no sentido Euclidiano é um bit. Tal vantagem pode ser sacrificada para diminuir a complexidade do algoritmo que pretendemos construir. A mudança que fazemos implica que os arcos sempre tenham a mesma direção, como mostra a figura (4:7). Deste modo, re-rotulamos os pontos da



Figura(4:6). Designação de bits aos quadrantes da cada constelação. Observe o sentido dos eixos. [6],

constelação nas seqüências binárias, como mostra a figura (4:8) Portanto a eq. (4:18) será :

(4:20) $p = B_{m-1} * 2^{m-1} + ... + B_1 2^* + B_0 2^\circ.$

EXEMPLO 4:7

Consideremos uma constelação de $M = 2^6$ pontos ($m = 3$). Como exemplo, vamos encontrar o.ponto da constelação designado à seqüência binária (100011). Temos :

$B = (1 -1 -1 -1 1 1) = (B_2, B_1, B_0).$

Assim,

$p = B_2 2^2 + B_1 2^1 + B_0 2^\circ$

$= (4, -4) + (-2, -2) + (1, 1) = (3,-5),$

como confirma a figura (4:8).

4:3:1:2) DEMAPEAMENTO SÍMBOLO A BIT

O demapeamento é similar ao algoritmo em [6], exceto pelas conseqüências da modificação anterior. Suponha que sgn(.) seja um operador definido sobre IR , que indica o sinal das duas coordenadas, isto é,

(4:21) $sgn(R_t) = (sgn\ x\ ^\wedge\ sgn$, dado $R\ ^\wedge\ x\ ^\wedge\ y\ ^\wedge$ e IR^2

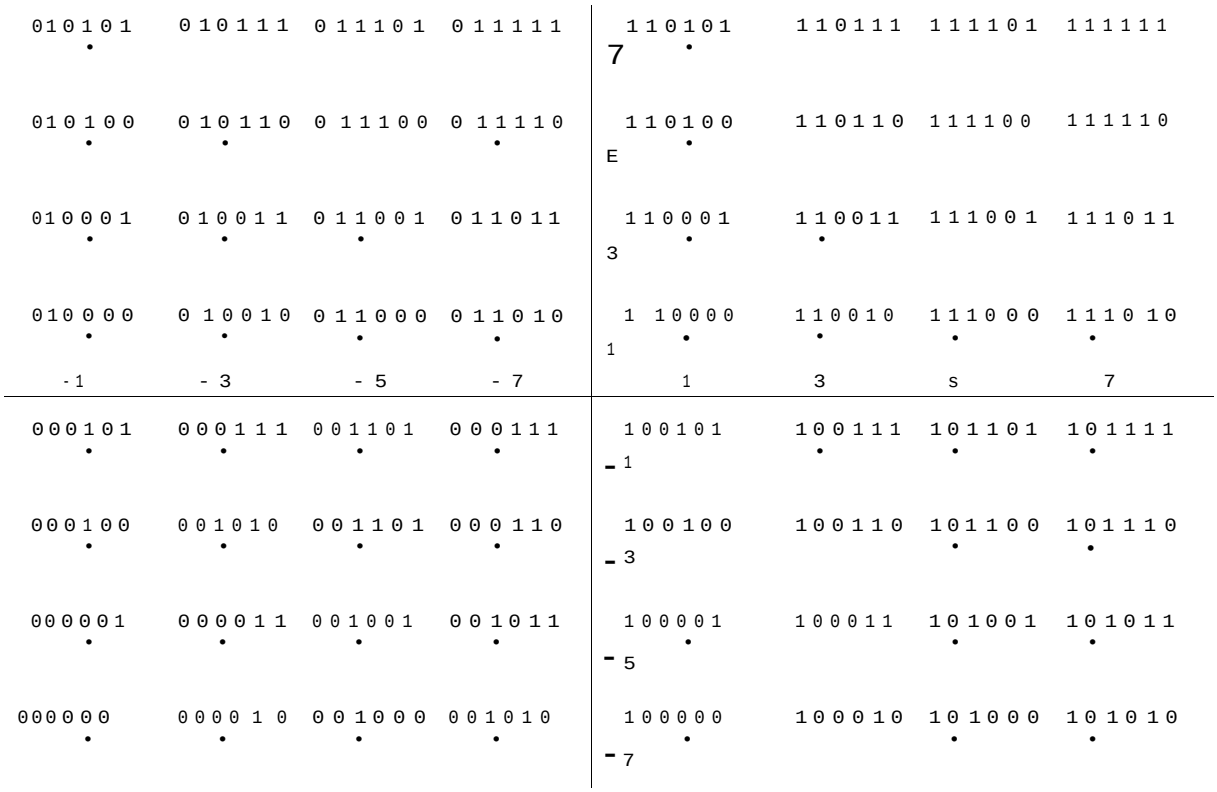


Figura (4:7). Esta figura mostra a atribuição de bit-a-símbolo de uma constelação de 64 pontos, de acordo com a modificação proposta.

e (0,0,1,1,1,1,1,1), que diferem somente nas primeiras duas coordenadas, também se afastam em *4* 2 unidades, e assim têm a mesma distinção entre si tanto quanto as duas palavras anteriores.

O segundo aspecto requerido a qualquer conjunto de palavras código é que a potência necessária para transmiti-las deve ser minimizada. A definição elementar de potência e de voltagem implica que a potência seja diretamente proporcional ao quadrado da voltagem. (e.g., para um circuito simples, a potência é o quadrado da voltagem dividida pela resistência do circuito). A potência total necessária para transmitir uma palavra código de oito dígitos é a soma dos quadrados destes dígitos. Esta soma é o quadrado da distância entre o ponto, o qual representa a palavra em um espaço de oito dimensões, e a origem (0,0,0,0,0,0,0,0).

O processo de projetar um sistema de palavras código, que seja confiável e que use eficientemente a potência, pode ser reduzido a um problema geométrico de alocação de pontos em uma região de um espaço, de uma maneira que não sejam tão próximos entre si, mas que fiquem o mais perto possível da origem do espaço. Se os pontos estiverem afastados por uma distância de -I 2 no mínimo, a questão será equivalente ao problema de se encontrar a embalagem de esferas cujos raios são metade desta distância, «J0.5. Um outro problema relacionado é encontrar todas as palavras código que têm a mesma potência, o que equivale a colocação de um maior número possível de esferas idênticas de n dimensões de modo que todas toquem uma esfera central de mesmo raio. Este é conhecido

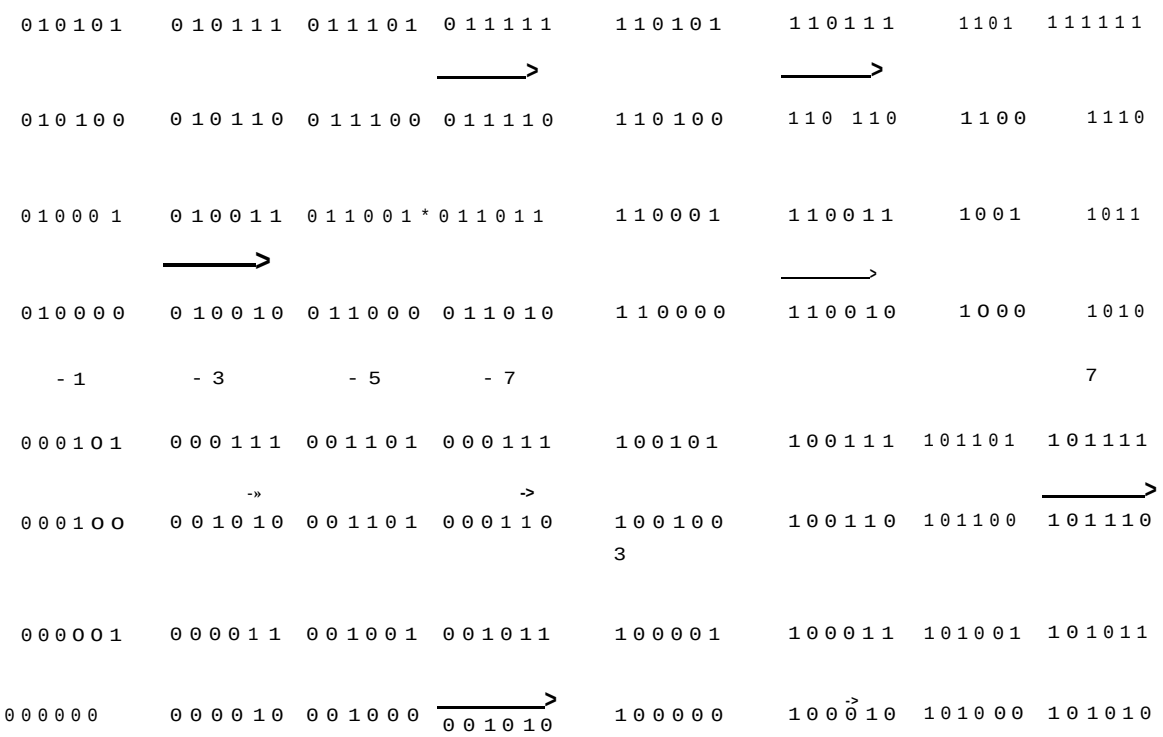


Figura (4:8). Designação de bits aos quadrantes da constelação, de acordo com a mudança.

Nesta seção, B_j denota o vetor bidimensional obtido por:

(4:22) $B_{i=}$ sgn (R) .

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

Algoritmo (4:5)

Faça i = m-1; ***R=*** R (condições iniciais)

etapa 1

B_t = sgntRj)

se i = 0, então termine o processo.

etapa 2

$R_{i-1} = R_i - 2^i * B_i$;

i = i-1;

vá à etapa 1.

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

EXEMPLO 4:8

Decodifique o vetor recebido R = (-3.2, 4.9), em um ponto de uma constelação de 64 pontos. O algoritmo acima resulta em:

R_2 = (-3.2, 4.9) * B_2 = (-1, 1)

R_1 = (0.8, 0.9) » B_1 = (1, 1)

R_0 = (-1.2,-1.1) * B_0 = (-1, -D

Então B = (-1,1,1,1,-1,-1), e a seqüência binária decodificada é (011100), que corresponde ao ponto mais próximo P = (-3, 5) da constelação.

Tais algoritmos .podem ser aplicados a uma constelação cúbica de n dimensões, a única mudança ocorre no vetor o qual será definido em n dimensões no lugar de duas dimensões, ou seja:

$$b = (b_{11}, b_{12}, \ldots, b_{1n}).$$

Tal mudança é trivial, mas é útil para ser aplicada em constelações multidimensionais codificadas, i.e. retículos, como a seguir.

No processo de partição do retículo em subconstelações com relação aos seus centros de gravidade, observamos que os pontos do retículo se apresentam na forma :

(4:23) $$C = B_{m-1}\, 2^{m-1} + B_{m-2}\, 2^{m-2} + \ldots + B_1\, 2^1 + B_0\, 2^0,$$

onde

$$B = (b_{11}, b_{12}, \ldots, b_{1n}),$$

sendo b_{IJ} = ±1 e n a dimensão do retículo. Nos retículos da

construção **A,** o vetor $\mathbf{B}_q$ é uma palavra do código $\mathbf{C}_q$ especificado para cada retículo. Nos retículos da construção **B,** os vetores $\mathbf{B}_q$ e $\mathbf{B}_I$ são palavras de dois códigos, $\mathbf{C}_q$ e , respectivamente, especificados para cada retículo. É claro que (4:23) é equivalente a generalização de equação (4:20) em n dimensões, com exceção de:

1) na construção **A,** $\mathbf{B}_q$ em (4:23) é uma palavra codificada;

2) na construção **B,** $\mathbf{B}_q$ e $\mathbf{B}_J$ em (4:23) ambos são palavras codificadas,

Considerando estes fatos, generalizamos os algoritmos anteriores para retículos multi-dimensionais.

4:3:2) CODIFICAÇÃO PARA RETÍCULOS OBTIDOS PELA CONSTRUÇÃO A.

Um retículo obtido aplicando-se a construção **A,**

$$\mathbf{A}_n = 2Z^n + \mathbf{C}(n,k,d),$$

capaz de transmitir n(m -l)+k bits por ponto, onde n é o número de dimensões do espaço sobre o qual o retículo está definido, $2^{2a>}$ é o número de pontos da constelação, e k é a dimensão do código **C.** Portanto, os n(m -1) bits identificam os m-1 primeiros termos em (4:23), os quais foram chamados de centro final mais próximo a **P,** e os últimos k bits escolhem um ponto do retículo partindo daquele

centro e usando o código C (n,k,d).

4:3:2:1) MAPEAMENTO BIT A PONTO.

Vamos considerar uma constelação quadrada com 2^m pontos. Uma seqüência binária de n(m-l)+k bits identifica um poipo no retículo A_n da seguinte maneira:

Algoritmo (4:6)

etapa 1

Codifique os últimos k bits em uma seqüência de n bits usando o código C (n,k,d), adicionando-a aos n(m-l) bits iniciais gerando uma nova seqüência binária de nm bits.

etapa 2

Utilize o algoritmo que mapeia uma seqüência de mn bits em um ponto em Z^n (constelação cúbica n-dimensional), para mapear esta nova seqüência a um ponto no retículo A_n.

EXEMPLO 4:9

Vamos mapear a seqüência (100110100011) em um ponto no retículo E_8.

Como a fórmula código do retículo E_8 é

$\mathbf{E_8} = 2Z^8 + C(8,4,4)$,

e como temos 12 bits para especificar um ponto, devemos 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) serão codificados em (11110000), gerando uma nova seqüência binária

b = (1001101011110000);

2) teremos então a .seqüência

$\mathbf{B}$ = (1,-1,-1,1,1,-1,1,-1,1,1,1,1,-1,-1,-1,-1)

$= (\mathbf{B_1}, \mathbf{B_0})$,

então,

$\mathbf{P}$ = (2.-2,-2,2,2,-2,2,-2)+ (1,1,1,1,-1,-1,-1,-1)

= (3,-1,-1,3,1,-3,1,-3).

4:3:2:2) **DEMAPEAMENTO PONTO A BIT.**

O demapeamento é equivalente àquele de Z^n com exceção de que o último ponto $\mathbf{R}_8$ deve ser decodificado em $\mathbf{B}_8$ usando-se o código $C_8(n,k,d)$.

Suponha que $\mathbf{R} = (r_1, r_2, \ldots, r_n)$ é um vetor recebido, $R \in R^n$, o mapeamos em uma seqüência binária correspondente ao ponto mais próximo do retículo $A = 2Z^n + C(n,k,d)$ da seguinte maneira:

Algoritmo (4:7)

As mesmas etapas do algoritmo (4:1) serão aplicadas neste algoritmo, excetuando-se a etapa (4:3), a qual é substituida por uma etapa que encontra os digitos de informação do vetor código $\mathbf{B}_q$, os quais formam com os vetores; $\mathbf{B}_{m-l}$, $\mathbf{B}_{m-2}$ . . . , e $\mathbf{B}_1$, a seqüência binária correspondente ao ponto P calculado na etapa (4:3).

i

EXEMPLO 4:10

Consideremos o retículo $\mathbf{D}=_4 2Z^4 + C_0(4,3,2)$ construído usando-se uma constelação bidimensional de 2 pontos (m = 4). Queremos, por exemplo, decodificar o vetor **R** (12.8.-8.9,0.5,-3.7) (exemplo 4:3) em uma seqüência binária b correspondente ao ponto mais próximo.

Q

Do exemplo 4:3, temos as seguintes seqüências decodificadas:

$\mathbf{B}_3$= (1,-1,-1, -1),

$\mathbf{B}_2$= (1, -1, 1, 1),

B = (1, 1, 1, 1), e

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

Os três primeiros dígitos em $\mathbf{B}_q$, (-1,1,1), correspondem aos bits de informação, portanto temos:

B = (1,-1,-1,-1,1,-1,1,1,1,1,1,1,-1,1,1),

a qual corresponde à seqüência binária b = (1000101111111011).

4:3:3) CODIFICAÇÃO PARA RETÍCULOS OBTIDOS PELA CONSTRUÇÃO B.

A diferença entre a construção **B** e construção A reside no fato de que os vetores $\mathbf{B}_j$ e $\mathbf{B}_q$ em (4:23) devem ser palavras dos códigos $C_j(n,k_i,d_i)$ e $C_o(n,k_Q,d_o)$, enquanto na construção A somente $\mathbf{B}_q$ é codificado. Portanto um retículo obtido aplicando-se esta construção é capaz de transmitir $n(m-2)+k+k$ bits por ponto, usando-se uma constelação de 2^{2m} pontos.

4:3:3:1) MAPEAMENTO BIT A PONTO.

Uma seqüência binária b de $n(m-2)+k_i+k_o$ bits pode ser mapeada a um ponto no retículo A_n, construído usando-se uma constelação bidimensional de 2^{2n} pontos como se segue.

Algoritmo (4:8)

etapa **1** :

Codifique os últimos k_Q bits em uma seqüência de n bits usando o código C (n^k—>d-), e codifique os penúltimos k^ bits em uma seqüência de n bits usando o código C (n,k ,d) gerando, no total, uma seqüência de nm bits.

etapa **2**

Utilize o algoritmo que mapeia uma seqüência de mn bits a um ponto em Z^n (constelação cúbica a n dimensões) para mapear esta nova seqüência em um ponto no retículo A_n.

4:3:3:2) MAPEAMENTO PONTO A BIT.

Modificamos o algoritmo **(4:2)** para mapear o vetor recebido em uma seqüência binária correspondente ao ponto **P** calculado na última etapa.

Algoritmo **(4**:9).

As mesmas etapas do algoritmo (4:2) serão aplicadas para este algoritmo, exceto a etapa (4:3), a qual será substituída por uma etapa que encontra os dígitos de informação nos vetores $\mathbf{B}_q$ e **B^**, os quais formam com os vetores $\mathbf{B}_{m-l}$, $\mathbf{B}_{m-2}$ **e B** seqüência binária correspondente ao ponto P, calculado em (4:3).

4:3:4) CODIFICAÇÃO PARA RETÍCULOS OBTIDOS 1 > CONSTRUÇÃO CÓDIGO GENERALIZADA.

Suponha que

$$\mathbf{A} = 2^{g} Z^{n} + \mathbf{C}_{g-1} 2^{g-1} + \ldots + \mathbf{C}_{1} 2^{*} + \mathbf{C}_{0} 2^{\circ},$$

é um retículo obtido aplicando-se esta construção, os veto s $B_{g-l}, \ldots, B_{1}$ e B_{0} em (4:23) devem ser palavras dos códigos $\mathbf{C}_{g-l}(n,k_{g-l},d_{g-l}), \ldots, \mathbf{C}_{1}(n,k_{1},d_{1})$ e $\mathbf{C}_{0}(n,k_{0},d_{0})$. Portanto um retículo obtido aplicando-se esta construção é capaz de transmitir $n(m-g)+ k_{g-l}+\ldots+k_{1}+k_{0}$ bits por ponto, usando-se uma constelação de 2^{21}" pontos.

4:3:4:1) MAPEAMENTO BIT A PONTO.

Uma seqüência binária b de $n(m-g)+ k_{g-l}+\ldots+k_{1}+k_{0}$ bits pode ser mapeada a um ponto no retículo $\mathbf{A}^{,}$ construído usando-se uma constelação bidimensional de 2^{2m} pontos como se segue.

Algoritmo (4:10)

etapa 1 :

começando do último dígito da seqüência b, cada $k^{}_{i}$ bits, onde i = 0,1,...g-l, serão codificados em uma

como o "problema de número das esferas tocantes" *(kissing number problem)* **[3].**

A correspondência entre um sistema' de comunicação e algumas idéias básicas da geometria foi estabelecida por Shannon **[1]**. Esta correspondência é sumarizada na tabela (1:1), onde W é a banda do sinal em um intervalo T. Um maior esclarecimento da relação entre o estudo de espaços geométricos e a construção e a concepção de sistemas de codificação pode ser obtido no exemplo na figura (1:1).

1.2 - EMBALAGENS DE ESFERAS EM UM SISTEMA DE COMUNICAÇÃO

Um sistema de comunicação de sinais, tal como aquele da figura (1:2), envolve a resolução eficiente de alguns problemas de maneira que o sinal da saída (1) seja uma réplica quase perfeita da entrada (1), e se o sinal for uma seqüência binária na entrada (2) , queremos que a seqüência na saída (2) seja uma réplica desta seqüência. Cada um destes problemas está ligado a cada bloco do sistema de transmissão.

1:2:1- AMOSTRAGEM :

O teorema da Amostragem de Shannon **[1][36]** garante que caso exista uma função f(t) que não possua nenhuma freqüência maior que W HZ, a mesma será completamente determinada

seqüência de n dígitos, utilizando o códijo C(n,k,d), gerando, no total, uma seqüência de m bits.

etapa 2 :

Utilize o algoritmo que mapeia uma seqüência de mn bits a um ponto em Z^n (constelação cúbica a n dimensões) para mapear esta nova seqüência em um ponto no retículo A_n.

4:3:4:2) MAPEAMENTO PONTO A BIT.

Modificamos o algoritmo (4:3) para mapear o vetor recebido a uma seqüência binária correspondente ao ponto P calculado na última etapa.

Algoritmo (4:11)

As mesmas etapas do algoritmo (4:3) serão aplicadas para este algoritmo, fora a etapa (4:3), a qual será substituída por uma etapa que encontra os dígitos de informação no vetor B_0, B_j, . . . , $\wedge_{g-1}$, os quais juntamente com os vetores B_{m-l}, B_{m-2} . . . , e B_g, formam a seqüência binária correspondente ao ponto P, calculado em (4:3).

CAPITULO 5

DISCUSSÃO E SUGESTÕES

5:1) COMPARAÇÃO

Os algoritmos apresentados no capítulo 4 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. Tal vantagem abre caminho para a utilização da distância de Hamming no processo de decodificação.

5:2) DISTÂNCIA MÍNIMA

Tal designação de rótulos binários aos pontos do retículo tem a vantagem de que, associa menos bits de informação (mais redundância) à parte do ponto que é mais provável de ser alterada pelo ruído. Para esclarecer este fato, considere dois

pontos:

$$P^{l} = B^{1}_{m-1} Z^{m'} \setminus B^{1}_{m-2} 2'' \ldots B^{1}_{1} 2^{l} + B^{0}_{0}, \text{ e}$$
$$P^{2} = B^{2}_{m-1} z''' . B^{2}_{m-2} 2'' \ldots + B^{2}_{1} 2' + 13^{2}2^{\circ}.$$

O quadrado da distância Euclidiana entre os dois pontos é

(5:1)
$$d^{2}(P^{1}-P^{2}) = 4*d_{H}(B^{1}_{m-1}, B^{2}_{m-1})* 2^{2(m-1)} + 4*d_{H}(B^{1}_{m-2}, B^{2}_{m-2})* 2^{2(m'^{2})} +$$

$$4*d_{H}(B^{1}_{1}, B^{2}_{1})* 2^{2} +$$
$$4*d_{H}(B^{1}_{0}, B^{2}_{0})* 2^{\circ},$$

onde d_{H} significa a distância de Hamming.

Esta equação revela que se as seqüências que correspondem aos dois pontos diferem somente em um bit ei B_{j} a distância euclidiana entre eles seria 4*2 • Portanto, o quadrado da distância Euclidiana mínima entre os dois pontos é

(5:2)
$$d^{2}_{in} = \min (4*d_{Huin}(B;. B^{2})* 2^{2J}), \quad j = o,..,m-1$$

Para os retículos obtidos aplicando-se a construção A,

$$d_{H(min)}(B^1, B^2) - d_{H(min)}(C) = d_0.$$

Concluímos que a distância ao quadrado mínima de um retículo A construído aplicando-se construção A é

(5:3) $$d^2_{min}(A) = \min[16, 4 * d_u, (C)] = \min[16, 4 * d_0].$$

Da mesma maneira, podemos mostrar que para um retículo A obt ' lo aplicando-se a construção B, o quadrado da distância mínima é

(5:4) $$d^2_{min}(A) = \min[64, 16 * d_u, (C), 4 * d_u, (C)] = \min[64, 16 * d_j, 4 * d_Q].$$

e para um retículo

$$A_n = 2^g Z^n + C_{g-1} 2^{g-1} + \ldots + C_1 2' + C_0 2^\circ,$$

obtido aplicando-se a construção código generalizada, o quadrado da distância mínima é

(5:5) $$d^2_{min}(A_n) = \min[4^{g+1}, 4^g * d_g, \ldots, 16 * d_1, 4 * d_0]$$

onde dj é a distância mínima de Hamming do código C_j. Estes resultados confirmam aqueles em **[2]** e [15].

A equação (4:24) revela que, se um erro de potência

igual a $d^2_{min}(A_n)/2$ atingir um ponto, no máximo 1 bit em B_1 pode ser alterado, ou d_o bits em B_o. Portanto é conveniente colocar menos bits de informações em B_o. 0 mesmo pode ser concluido sobre retículos obtidos aplicando-se a construção B ou a construção códigogeneralizada.

A Tabela (5:1) compara a probabilidade de erro por bit (BER) para o retículo E_8, construído usando uma constelação bidimensional de 16 pontos, usando esta atribuição e uma rotulação por tabela (look up table), como aquela no capítulo 3.

5:3) A COMPLEXIDADE.

Como os algoritmos que codificam e decodificam os retículos obtidos aplicando-se a construção A ou B são casos particulares daqueles que decodificam retículos obtidos aplicando-se a construção código generalizada, vamos calcular a complexidade do último.

Seja

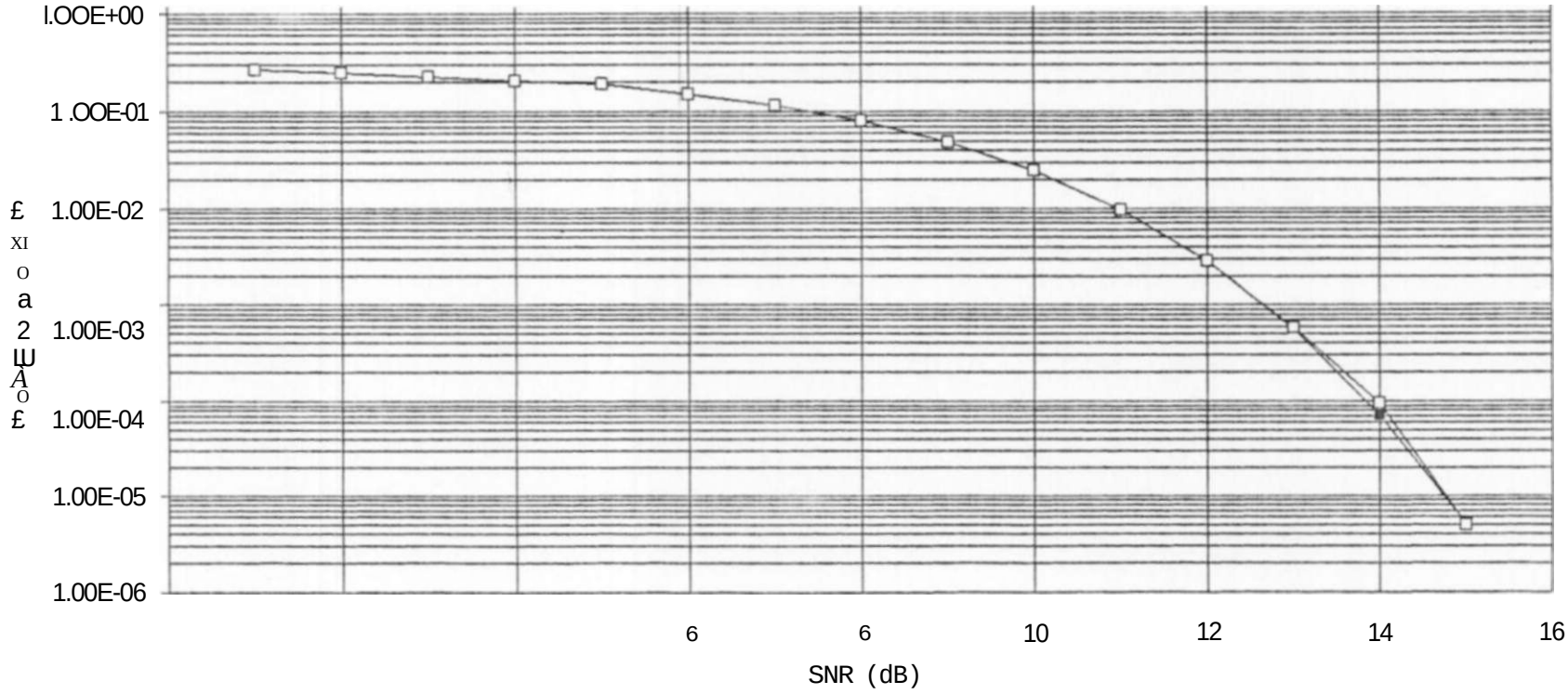
$$A_n = 2^0 Z^n + C_{g-1} 2^{g-1} + \ldots + C_1 2^1 + C_0 2^0$$

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

Tabela (5:1). Probabilidade de erro por bit (BER) para o mapeamento do algoritmo (4:8) e um tradicional, como aquele do algoritmo (3:5), usando o retículo E_8, construído usando-se uma constelação de 16 pontos.

sinal a ruído dB	probabilidade de erro para algoritmo (3:5).	probabilidade de erro para algoritmo (4:8).
1	2.66 e-01	2.66 e-01
2	2.49 e-01	2.49 e-01
3	2.28 e-01	2.28 e-01
4	2.06 e-01	2.06 e-01
5	1.80 e-01	1.80 e-01
6	1.49 e-01	1.49 e-01
7	1.15 e-01	1.15 e-01
8	8.03 e-02	8.03 e-02
9	4.86 e-02	4.86 e-02
10	2.45 e-02	2.45 e-02
11	9.66 e-03	9.66 e-03
12	2.83 e-03	2.85 e-03
13	5.72 e-04	5.75 e-04
14	7.08 e-05	6.90 e-05
15	5.14 e-06	5.00 e-06

Figura (5.3). Probabilidade de erro por bit do Algoritmo (4:8) e Algoritmo (3:5) para o retículo E8 construído à partir de uma constelação de 16 pontos.



1- Para decodificar uma palavra do código $\mathbf{C}_{q}$ SO necessárias mn subtrações, adicionado ao número de operaçôV--. da decodificação suave.

2- Para decodificar uma palavra do **código** lo necessárias n((m-l)+l) subtrações, além do número **de operaçõi** s da decodificação suave.

3- em geral, para decodificar uma palavra do código C são necessárias n((m-i)+l) subtrações, além do número de operações da decodificação suave.

4- Para decodificar uma palavra do < ódigo $^{c}{}_{q\,1}{}^{:}$ ão necessárias n(m-g-l)) subtrações, além do número de operaçõ' de decodificação suave.

5- Para decodificar o vetor $2^{9}Z^{n}$ ão necessáriasn(m-g+1) subtrações.

Portanto, no total 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 necessárias para obter a decodificação **suave** utilizando os códigos C^, onde i = 0, 1, . . , g-1.

No caso de codificação, os algoritmos precisam de n(m-l) adições, além do número de operações para **Codificar** $\mathbf{k}_{g\,i}$**+**. . . . **.+k**$_{i}$**+k**$_{Q}$ bits de informaçãos nos códigos correspondentes.

5:4) **DESEMPENHO**

Como mostra a tabela (5:2) e (5:3), 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. E, como notamos anteriormente, a decodificação para retículos obtidos aplicando-se a construção código generalizada, não realiza decodificação por máxima verossimilhança, mas como mostra a tabela (5:4), tem o mesmo desempenho.

Tabela (5:2). Comparação da probabilidade de erro por ponto entre o algoritmo (4:1) e um algoritmo de decodifi< o por "máxima verossimilhança" para o retículo D , construído usando-se Uma constelação de 16 pontos.

sinal a r ujLd o d B	probabilidade de er- ro usando um algorit mo de m&x ima veros- slmilhanca.	probabilidade de erro usando o al- goritmo (4:1).
1	0.85060000	0.85060000
2	0.80449998	0.80449998
3	0.76380000	0.76380000
4	0.71130000	0.71130000
5	0.64330000	0.64330000
6	0.55909997	0.55909997
7	0.47666666	0.47666666
8	0.37279999	0.37279999
9	0.27645001	0.27645001
10	0.18765432	0.18765432
11	0.10980000	0.10980000
12	0.05971605	0.05971605
13	0.02532500	0.02532500
14	0.00887167	0.00887167
15	0.00225000	0.00225000
16	0.00040167	0.00040167
17	0.00004000	0.00004000

TABELA (1:1). correspondência entre um sistema de comunicação e idéias de geometria [1].

Sistema de comunicação	Entidade geométrica .
0 conjunto de possíveis sinais.	Um espaço de 2TW dimensões.
Um sinal particular.	Um ponto no espaço.
Distorção no canal.	Um empacotamento do canal.
Ru{d o no canal.	Uma região de incerteza ao redor de cada ponto.
A energia média do sinal.	-1 (2WT) vezes o quadrado da distancia entre a origem e o ponto.
0 conjunto dos sinais de energia P.	0 conjunto dos pontos numa esfera de raio ^21 WP.
0 conjunto das mensagens possíveis.	Um espaço de 2T W dimensões. 1 1
0 conjunto das mensagens reais.	Um espaço de D dimens_^oes obtido através de considerar todos as mensagens equivQ.1CO_ – tes como um ponto, e de1e- tar as mensagens que a fon- te nao produz.
Uma mensagem.	Um ponto no espaço.
0 transmissor.	Um mapeamento das mensagens nos pontos do espaço.
0 receptor.	Um mapeamento dos pontos do espaço no espaço das mensa- gens.

Figura (5:2). Probabilidade de Erro por ponto do Algoritmo (4:1) para o Retículo D4 construído à partir de uma constelação de 16 pontos.

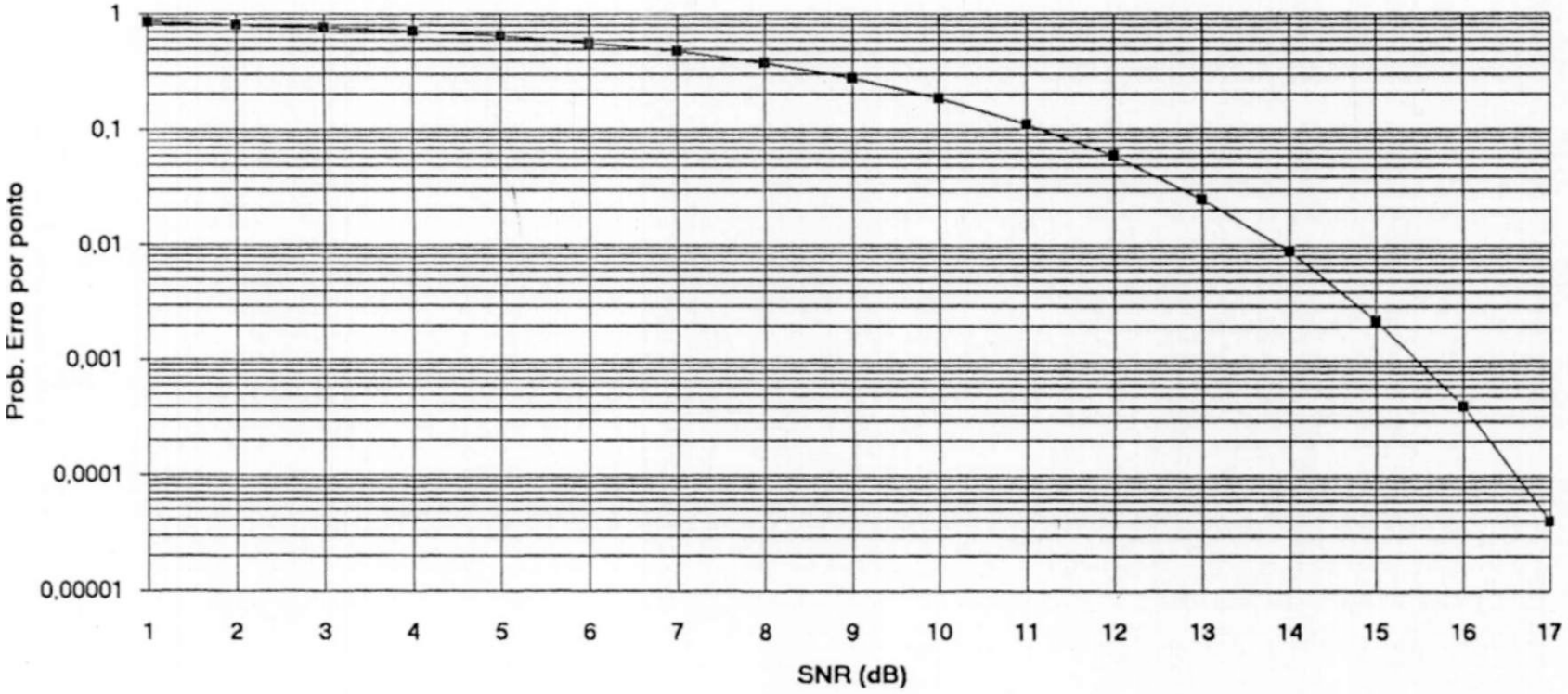


Tabela (5:3). Comparação da' probabilidade de erro por pont entre algoritmo (4:1) e um algoritmo de decodificaçã por "máxima verossimilhança" para o retículo E construído usando-se uma constelação de 16 pontos.

sinal a / *v* u>td o d B	probabilidade de erro usando um algorlt mo de *mdix* ima veros-siml lhanca.	probabilidade de erro usando algo-ritmo (4:1).
1	9.308e-01	9.308e-01
2	8.862e-01	8.862e-01
3	8.426e-01	8.426e-01
4	7.848e-01	7.848e-01
5	6.897e-01	6.897e-01
6	5.791e-01	5.791e-01
7	4.489e-01	4 .489e-01
8	3.092e-01	3.092e-01
9	1.863e-01	1.863e-01
10	9.150e-02	9.150e-02
11	3.627e-02	3.627e-02
12	1.048e-02	1.048e-02
13	2.191e-03	2.191e-03
14	2.170e-04	2.170e-04
15	1.400e-05	1.400e-05
16	4.414e-07	4.414e-07

Figura (53). Probabilidade de Erro po ponto do Algoritmo (4:1) para o Retículo E8 construído à partir de uma constelação de 16 pontos.

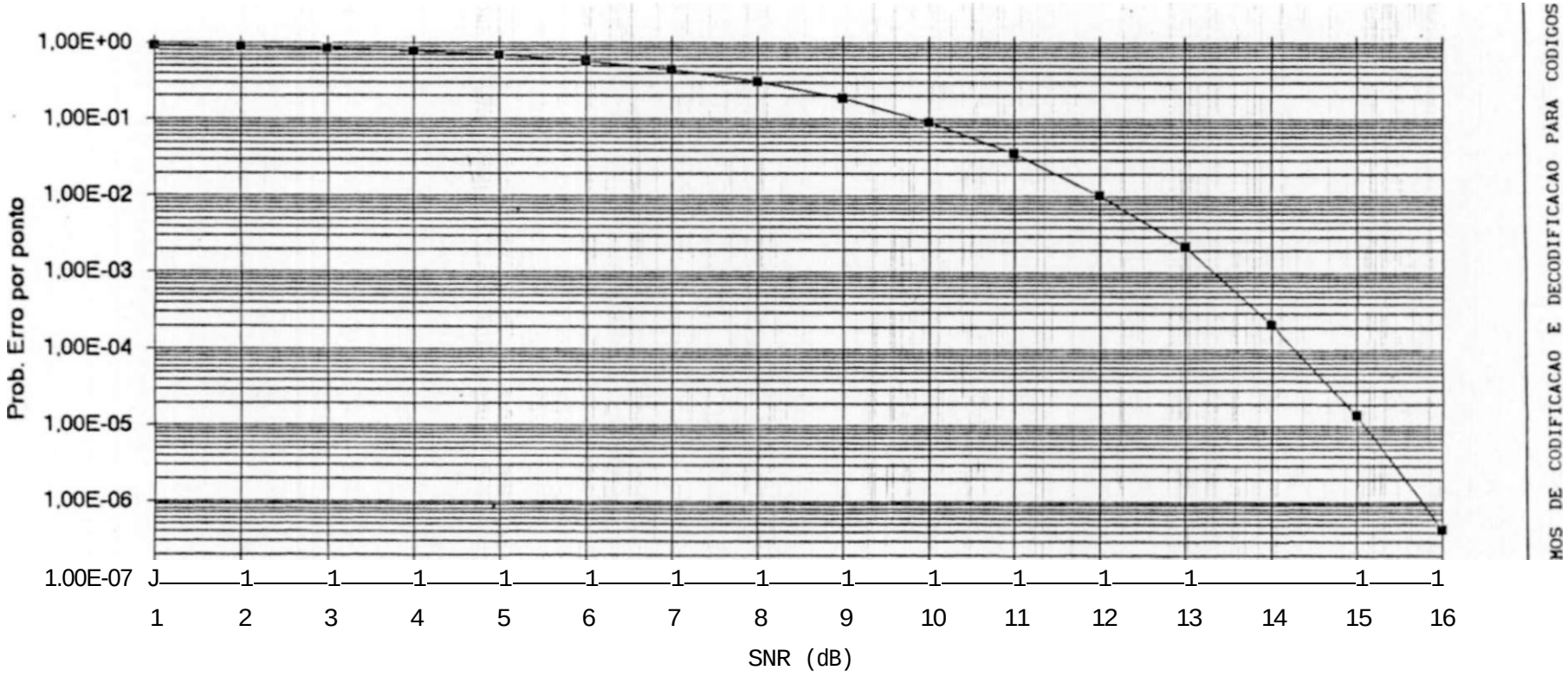
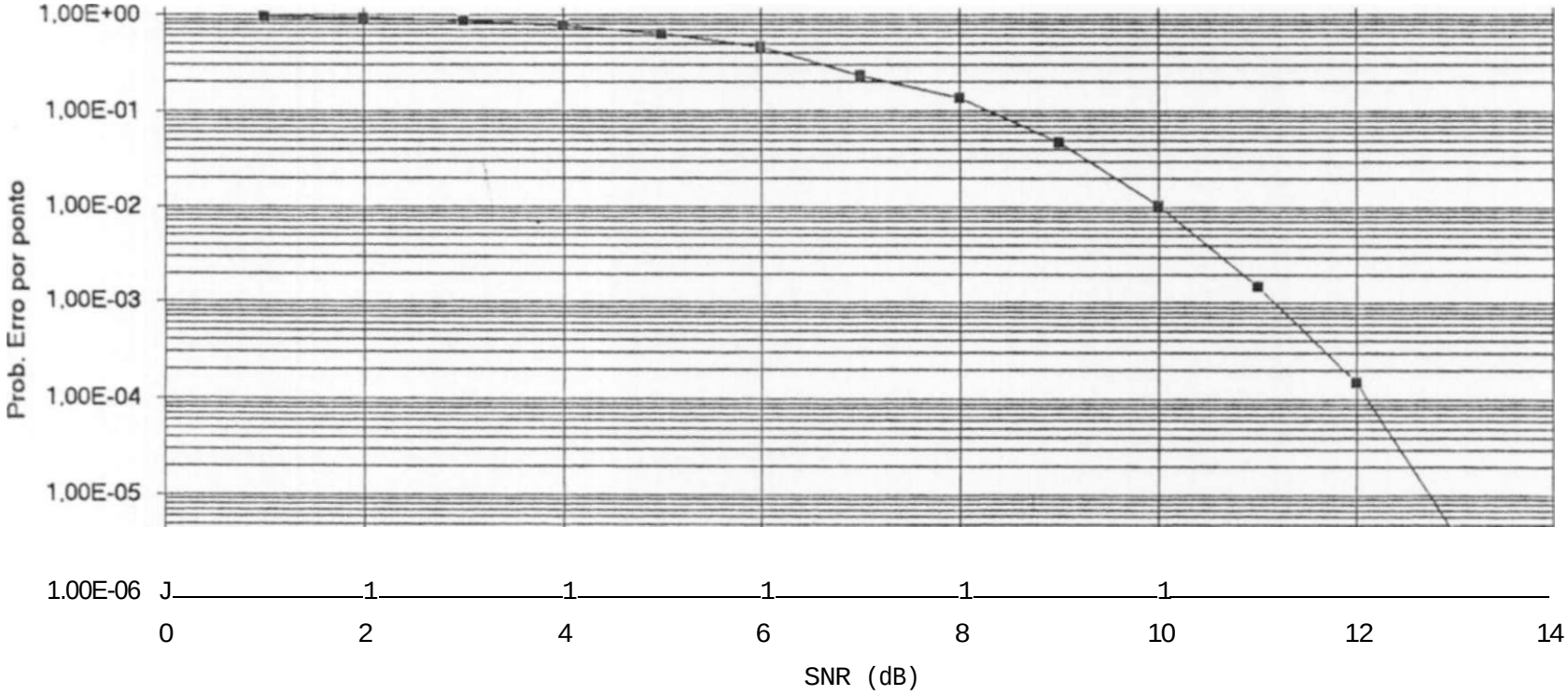


Tabela (4:5). Comparação da probabilidade de erro por ponto entre algoritmo *(4:2)* e um algoritmo de decodificação por "máxima verossimilhança" para o retículo A_{j6}, constuido usando-se uma constelação de 16 pontos.

sinal a ruA-d o dB	probabilidade de erro usando um algoritmo de deco d 1 f 1 caç£o intima- • verossimi lhanca	probabl 1 idade de erro usando o a1- gorltmo (4:2) • na 1 b prflx i mo.	Prob ab 1 1 Idadô de n^o deco d.L ~~ f 1car o pontL>
1	9.653e-01	9.653e-01	4.5e-5
2	9.352e-01	9.352e-01	3.Oe-5
3	8.740e-01	8.740e-01	2.Oe-5
4	7.765e-01	7.765e-01	4.Oe-5
5	6.381e-01	6.381e-01	1.5e-5
6	4.669e-01	4.669e-01	0.5e-5
7	2.836e-01	2.836e-01	0.5e-5
8	1.365e-01	1.365e-01	0.OeOO
9	4.710e-02	4.710e-02	0.OeOO
10	1.050e-02	1.050e-02	0.OeOO
11	1.500e-03	1.500e-03	0.0e00
12	1.500e-04	1.500e-04	0.0e00
13	4.100e-06	4.100e-06	0.0e00

Figura (5:W). **Probabilidade de Erro** por ponto do Algoritmo (4:2) para o Retículo construído à partir de uma constelação de 16 pontos.



5:5) MODULAÇÃO CIFRADA .

Trata-se de uma nova proposta, de combi r simultaneamente a modulação digital com um cifragem (e possivelmente um código de canal). Apenas introduzimos o tópico de pesquisa.

Considerando a fórmula código

$$A = C + 2C + 4C + \underline{\quad} + 2^{k-1}C + 2^{k}Z^{k},$$
$$\quad n \quad 0 \quad 1 \quad 2 \qquad\qquad k-1$$

qual a possibilidade de se obter um sistema de cifragem com chave privada utilizando os retículos?

Uma alternativa consiste em utilizar as várias combinações dos códigos binários como a chave do sistema.

Suponha que os códigos:

$$C_{o^{i}}(N, M . d)_{o' *},$$
$$C(N, M, d),$$

$$C_{m-1}(N, M . d)_{m-1^{\circ\circ}},$$

formam a chave do sistema. Suponha que o texto claro é

$$S = [*_{lt} \ldots, s_{t}),\ t = M_{Q} + M_{j} + \ldots + M_{mi}.$$

A sequência S será dividida em m sequências menores S^onde o comprimento da sequência é

$$S_1 \xrightarrow{C} B_i$$

Cada sequência é codificada em uma sequência usando o **código**

Temos agora duas alternativas para obter o texto cifrado:

1- Para cada B_j adicionamos um vetor de ruído $E_{|\#}$ onde

$w(E_j) < \hat{U}J2$, onde $w(E^)$ é o peso de ,

obtendo o vetor

$B_{\acute{i}} = B_{\acute{i}}^{*} E_{\acute{i}}$ (em notação -1, 1).

o texto cifrado seria o ponto

$$\overset{A}{P} = \overset{A}{B}_{m-l} 2^{m-1} + \overset{A}{B}_{m-2} 2^{m-?} + \_\_\_\_ + \overset{A}{B}_{1} 2 + \overset{A}{B}_{0},$$

O erro total adicionado deve manter a distância entre P e P **menor**

que a metade da distância mínima d_{min} do retículo, ou seja

$$d^2 < P - f > <$$

onde

$$P = B_{m-1} 2^{m-1} + B_{m-2} 2^{m-2} + \_\_\_\_ + B_1 2 + B_0,$$

De equação (5:1)

(5:6) $$d^2(P - P) = 4 * d(B_n B_{m-l}) * 2^{2(m}$$

$$4 * d(B_H, B_{m-2})* 2^{2(m''^2)} +$$

$$4 * d(B_H, B_l)* 2 4$$

$$4 * d(B_H . B_0)* 2°,$$

(5:7) $$d^2(P - P) = 4 * 2^{2(m-1)} * w(E_{m-1}) +$$

$$_4 * _2{}^{2(\text{œ}''^2)} * w(E_{m-2}) +$$

$$+ 4 * 4\ w(E)$$

$$+ 4 * w(E)$$

(5:8) $$d^2(P - P) = 2 * w(E_{m-l}) +$$

$$2^{2<-1\}} * w(E_{m-2})4$$

$$+ 16 * w(E)$$

$$+ 4 * w(E_{o})$$

portanto

(5:8) $2^{2m} * W(E) + ... + 16. * w(l) + 4 * w(E_{o}) < d_{min}/4$

Para obter o texto claro, simplesmente fazemos:

(5:9) $\hat{B}_{tn-1} = sgn(\hat{P} - (\hat{B}_{m-1} 2 + ... + B_{m-11*1)} 2^{-11*1}»))$

$$B_1 \quad \overset{\cdot}{|}_{decodificada} \quad \hat{S}_i$$

2- A outra alternativa é calcular o ponto P usando os sequências B_t:

(5:10) $P = B_{m-1} 2^{m-1} + B_{m-2} 2^{m"2} + \_\_\_\_ + B\ 2 + B_0$.

O ponto P será perturbado por um ruido F

(5:11) P = P + F, onde

$$F = (f_1, \ldots, f_n),$$

sendo

$$f_1^2 + f_2^2 + \ldots + f_n^2 < d^2(A)/4$$

usando o algoritmo em [1], este ponto pode ser decodificado na seqüência binária S.

Escolhemos a primeira alternativa para sugerir um sistema de cifragem. Suponha que os códigos

$$C(N, M, d),$$
$$C(N, M, d),$$

$$C(N.M, d),$$

formam o reticulo

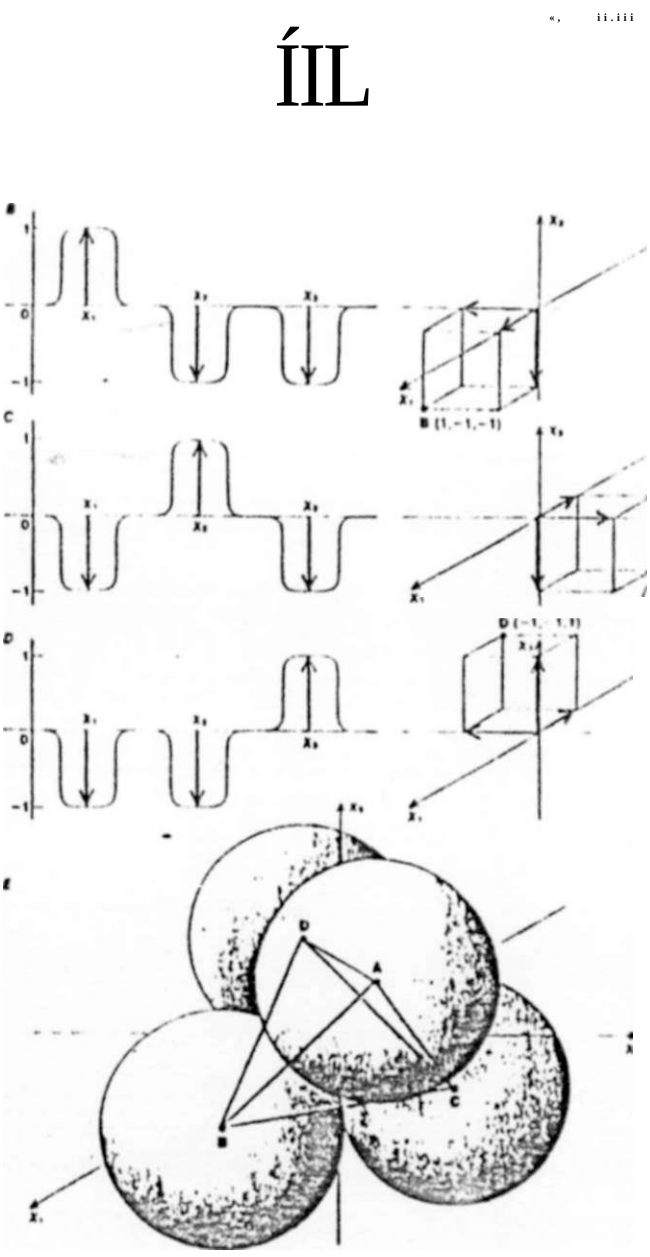
(5:12) $$A = C_{m-1}\,2^{m-1} + C_{m-2}\,2^{m-2} + \ldots + C_1 2 + C_0$$

o ponto

(5:13) $$P = B_{m-1}\,2^{m-1} + B_{m-2}\,2^{m-2} + \ldots + B_1 2 + B_0$$

é um ponto no retículo, onde

Figura (1:1).



O projeto de um código para uma transmissão eficiente de informações é um problema relacionado à embalagem de esferas. O código é um conjunto finito de sinais chamadas palavras código, as quais devem ser distintas entre si e não perdem potência. Se cada palavra é uma seqüência de três níveis discretos de voltagem, cada uma pode ser representada por um ponto num espaço de três dimensões. A potência necessária para transmitir cada é a soma dos quadrados de suas coordenadas. Para minimizar esta potência, os pontos devem estar o mais próximo possível da origem. Por outro lado, para serem distintas entre si devem ser afastadas por uma distância d. Isto é equivalente a colocar esferas de raio d/2 ao redor da origem [3],

B_{m-1} e C_{m-1}

O ponto P pode ser definido por uma matriz

(5:14) $$B = \begin{vmatrix} B_{m-1} \\ B_{m-2} \\ \vdots \\ B_{1} \\ B_{0} \end{vmatrix} \quad \begin{vmatrix} b_{1,m-1} & b_{2,m-1} & \cdots & b_{n,m-1} \\ b_{1,m-2} & b_{2,m-2} & & b_{n,m-2} \\ \vdots & & & \\ b_{1,1} & b_{2,1} & & b_{n,1} \\ b_{1,0} & b_{2,0} & & b_{n,0} \end{vmatrix}$$

Inicialmente cada $(M_{m-1} + M_{m-2} + \ldots + M_1 + M_0)$ dígitos binários são codificados em uma matriz B. O texto claro consiste de

(5:15) $$T^* (M_{m-1} + M_{m-2} + \ldots + M_1 + M_0)$$

bits binários. Portanto temos T matrizes codificadas

(5:16) $$B = \begin{matrix} B_{m-1} \\ B_1 \\ B_0 \end{matrix} \qquad B^2 = \begin{matrix} B_{m-1}^2 \\ B_1^2 \\ B_0^2 \end{matrix} \qquad B^T = \begin{matrix} B_{m-1}^T \\ B_1^T \\ B_0^T \end{matrix}$$

Re-ordenamos estas matrizes para obter as seguintes m matrizes

(5:17) B^{m-1} B^{m-2} B^{0} =

T	T	T
m-1	m-2	0
T-1	T-1	T-1
m-1	m-2	0
1	1	1
m-1	m-2	0
0	0	0
m-1	m-2	0

A partir destas, construimos T matrizes da seguinte maneira;

Suponha que queremos construir a matriz B^{i}, i= 1 - T. A primeira linha desta matriz consiste da primeira linha de uma matriz B^{j}. A segunda linha consiste da segunda linha de outra matriz B^{l}, e assim por diante. Uma linha de uma matriz B não pode ser escolhida mais de uma vez. Temos, portanto as seguintes matrizes:

(5:18) B^{1} = B^{2} =

B^{1}_{m-1}	B^{2}_{m-1}	S^{7}_{m-1}
B^{1}_{m-2}	B^{2}_{m-2}	B^{1}_{m-2}
B^{1}_{1}	B^{2}_{1}	B^{1}_{1}
B^{1}_{0}	B^{2}_{0}	B^{7}_{0}

A cada matriz adicionamos um erro binário de acordo com a. equação (5:8), obtendo

(5:19) $B^{*}_{i} = \begin{matrix} B_{m-1} \\ B^{1}_{m-2} \\ \vdots \\ B \end{matrix} \quad B^{2} = \begin{matrix} 2_{m-1} \\ 2_{m-2} \\ \vdots \\ 1 \\ B^{2} \end{matrix} \quad B^{T} = \begin{matrix} i-1 \\ m-2 \\ \vdots \\ 1 \\ 0 \end{matrix}$

Cada matriz B^{1} é utilizada para calcular um ponto

$$P^{1} = B_{m-1} z''' * B_{m-2} 2^{-2} + \ldots + B_{i} * 2 + B_{o},$$

onde

$$P = P + E$$

sendo P um ponto do retículo e E o erro adicionado na equação (5:19).

Dos T pontos calculados formamos uma matriz

$$(5:20) \qquad P = \begin{matrix} P^1 \\ \\ \\ \end{matrix} \begin{bmatrix} p_1^1 & p_2^1 & \cdots & p_n^1 \\ p_1^2 & p_2^2 & \cdots & p_n^2 \\ \vdots & \vdots & & \vdots \\ p_1^{t-1} & p_2^{t-1} & \cdots & p_n^{t-1} \\ p_1^t & p_2^t & \cdots & p_n^t \end{bmatrix}$$

A matriz P será multiplicada por uma matriz X de permutações antes da transmitissão.

Podemos observar que a chave deste sistema é dividido em três partes:

1- A construção do retículo (i.e. os códigos binários involvidos),

2- A primeira permutação para obter as matrizes B^1, e

3- A segunda permutação aplicada na matriz P.

Outra permutação equivalente àquela da Equação (5:18) pode ser aplicada às matrizes (5:19). Basicamente, este sistema é de chave privade, mas pode ser de chave publica se for possível encontrar uma matriz (ou um conjunto de matrizes) que incluía as operações de codificação e permutação.

r CAPITULO 6

CONCLUSÕES

Na introdução deste trabalho, estudamos a justificativa da utilização de retículos em sistemas 2 comunicações digitais. Observamos que os retículos podem *r* empregados como métodos de modulação codificada (códigos de canal), ou como métodos de quantização.

As embalagens reticuladas em geral, seus parâmetros, alguns retículos em particular, e a relação entre eles e os códigos binários, foram os assuntos principais do segundo capítulo.

Após o estudo de alguns algoritmos de codificação e decodificação para retículos no capítulo 3, concluímos a necessidade de um sistema completo para estes processos.

Partindo de uma idéia simples, a "decodificação via baricentros" estabelecida por De Oliveira para constelações bidimensionais, desenvolvemos 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 tabelas *(look up tables).*

2- Nos algoritmos de decodificação, os pontos ruidosos 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 codificação e decodificação 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 aqueles de decodificação por máxima verossimilhança,utilizando como exemplos os retículos conhecidos D_4, E_8 e A_{16}'.

Dos resultados obtidos, concluímos que:

1- Para retículos obtidos pela construção código A, estes algoritmos são de decodificação por máxima verossimilhança.

2- Para os retículos obtidos pela construção código generalizada (exceto a construção A), estes algoritmos são de decodificação por distância cotada.

Com base nos resultados de simulação utilizando o retículo A_{j6}, concluímos que a decodificação por distância cotada' é, praticamente, equivalente àquela por máxima verossimilhança.

Apresenta-se no final do trabalho, uma proposta para aplicar os retículos como métodos de cifragem com a vantagem de ser

projetado conjuntamente com a modulação digital. O sistema apresentado é apenas uma proposta inicial e pode ser modificado e aperfeiçoado,o que é sugerido como tópico de pesquisa.

Como os algoritmos apresentados reduzem o problema de decodificação a um problema equivalente com códigos binários, sugere-se uma pesquisa sobre a utilização da distância de Hamming (e,g, decodificação algébrica) no processo de decodif.icação no intuito de analisar o compromisso desempenho X complexidade.

r APÊNDICE A

DIVISÃO DE UMA REGIÃO DO RETÍCULO D_4

As etapas básicas do algoritmo de decodificação por centro de gravidade foram aplicadas ao retículo **D4** limitado para transmitir 3.5 bits/simbolo (1.75 bits/dim). A tabela seguinte mostra os resultados do processo da divisão, onde cada grupo K resultante de uma divisão anterior será dividido nos subgrupos 2K e 2K+1. (Por exemplo, o espaço **D4** inteiro, no.l, tem o centro (0,0,0,0) e será dividido nos subgrupos 2 e 3, etc).

Tabela (A:l)

numero do grupo	centro <io grupo	subgrupos
1	(o 0 0 0)	2 3
2	(-2 0 0 0)	4 5
3	(2 0 0 0)	6 7
4	(-2 -2 0 0)	8 9
5	("2 2 0 0)	10 11
6	(2-2 0 0)	12 13
7	(2 2 0 0)	14 15
8	("2 -2 -2 0)	16 17
9	("2 -2 2 0)	18 19
10	("2 2-2 0)	20 21
11	("2 2 2 0)	22 23
12	(2-2 2 0)	24 25
13	(2-2 2 0)	26 27
14	(2 2-2 0)	28 29
15	(2 2 2 0)	30 31

16	(-2-2-2-2)	32 33
17	(-2-2-2 2)	34 35
18	(-2-2 2-2)	36 37
19	(-2-2 2 2)	38 39
20	(-2 2-2-2)	40 41
21	(-2 2-2 2)	42 43
22	(-2 2 2-2)	44 45
23	(-2 2 2 2)	46 47
24	(2-2-2-2)	48 49
25	(2-2-2 2)	50 51
26	(2-2 2-2)	52 53
27	(2-2 2 2)	54 55
28	(2 2-2-2)	56 57
29	(2 2-2 2)	58 59
30	(2 2 2-2)	60 61
31	(2 2 2 2)	62 63
32	(-3-2-2-2)	64 65
33	(-1-2-2-2)	66 67
34	(-3-2-2 2)	68 69
35	(-1-2-2 2)	70 71
36	(-3-2 2-2)	72 73
37	(-1-2 2-2)	74 75
38	(-3-2 2 2)	76 77
39	(-1-2 2 2)	78 79
40	(-3 2-2-2)	80 81
41	(-1 2-2-2)	82 83
42	(-3 2-2 2)	84 85
43	(-1 2-2 2)	86 87
44	(-3 2 2-2)	88 89
45	(-1 2 2-2)	90 91
46	(-3 2 2 2)	92 93
47	(-1 2 2 2)	94 95
48	(1-2-2-2)	96 97
49	(3-2-2-2)	98 99
50	(1-2-2 2)	100 101
51	(3-2-2 2)	102 103
52	(1-2 2-2)	104 105
53	(3-2 2-2)	106 107
54	(1-2 2 2)	108 109
55	(3-2 2 2)	110 111
56	(1 2-2-2)	112 113
58	(1 2-2 2)	116 117
59	(3 2-2 2)	118 119
60	(1 2 2-2)	120 121
61	(3 2 2-2)	122 123
62	(1 2 2 2)	124 125
63	(3 2 2 2)	126 127
64	(-3-3-2-2)	128 129
65	(-3-1-2-2)	130 131
66	(-1-3-2-2)	132 133
67	(-1-1-2-2)	134 135
68	(-3-3-2 2)	136 137

69	("3 -1-2 2)	138 139
70	(-1 -3-2 2)	140 141
71	(-1 -1-2 2)	142 143
72	("3 -3 2-2)	144 145
73	("3 -1 2-2)	146 147
74	(-1 -3 2-2)	148 149
75	(-1 - 1 2-2)	150 151
76	("3 -3 2 2)	152 153
77	("3 - 1 2 2)	154 155
78	(-1 -3 2 2)	156 157
79	(-1 - 1 2 2)	158 159
80	(-3 1-2-2)	160 161
81	("3 3-2-2)	162 163
82	(-1 1-2-2)	164 165
83	(-1 3-2-2)	166 167
84	(-3 1-2 2)	168 3.69
85	.(-3 3-2 2)	170 171
86	(-1 1-2 2)	172 173
87	(-1 3-2 2)	174 175
88	("3 1 2-2)	176 177
89	("3 3 2-2)	178 179
90	(-1 1 2-2)	180 181
91	(-1 3 2-2)	182 183
92	("3 1 2 2)	184 185
93	("3 3 2 2)	186 187
94	(-1 1 2 2)	188 189
95	(-1 3 2 2)	190 191
96	(1-3-2-2)	192 193
97	(1-1-2-2)	194 195
98	(3-3-2-2)	196 197
99	(3-1-2-2)	198 199
100	(1-3-2 2)	200 201
101	(1-1-2 2)	202 203
102	(3-3-2 2)	204 205
103	(3-1-2 2)	206 207
104	(1-3 2-2)	208 209
105	(1- 1 2-2)	210 211
106	(3-3 2-2)	212 213
107	(3- 1 2-2)	214 215
108	(1-3 2 2)	216 217
109	(1- 1 2 2)	218 219
110	(³ -3 2 2)	220 221
111	(3- 1 2 2)	222 223
112	(1 1-2-2)	224 225
113	(1 3-2-2)	226 227
114	(3 1-2-2)	228 229
115	(3 3-2-2)	230 231
116	(1 1-2 2)	232 233
117	(1 3-2 2)	234 235
118	(3 1-2 2)	236 237
119	(3 3-2 2)	238 239
120	(1 1 2-2)	240 241

por suas amostras colhidas a cada 1/2W segundos, como mostra a figura (1:3). O valor 2W é a taxa mínima de amostragem, conhecida como a taxa de Nyquist [32].

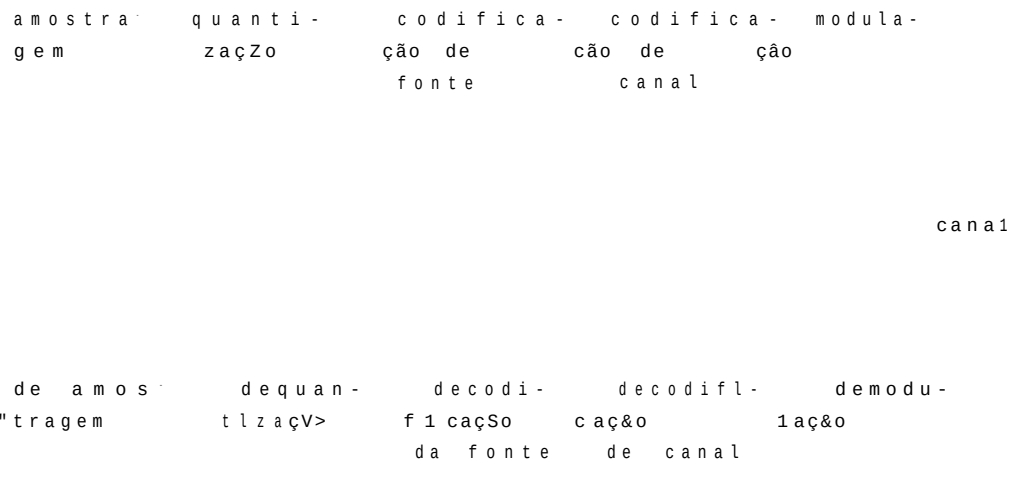


Figura (1:2). Sistema de comunicação.

121	(1 3 2-2)	242 243
122	(3 1 2-2)	244 245
123	(3 3 2-2)	246 247
124	(1 1 2 2)	248 249
125	(1 3 2 2)	250 251
126	(3 1 2 2)	252 253
127	(3 3 2 2)	254 255
128	(-3-3-3-3)	0 0
129	(-3-3-1-1)	0 0
130	(-3-1-3-1)	0 0
131	(-3-1-1-3)	0 0
132	(-1-3-3-1)	0 0
133	(-1-3-1-3)	0 0
134	(-1-1-3-3)	0 0
135	(-1-1-1-1)	0 0
136	(-3-3-3 1)	0 0
137	.(-3-3-1 3)	0 0
138	(-3-1-3 3)	0 0
139	(-3-1-1 1)	0 0
140	(-1-3-3 3)	0 0
141	(-1-3-1 1)	0 0
142	(-1-1-3 1)	0 0
144	(-3-3 1-3)	0 0
145	(-3-3 3-1)	0 0
146	(-3-1 1-1)	0 0
147	(-3-1 3-3)	0 0
148	(-1-3 1-1)	0 0
149	(-1-3 3-3)	0 0
150	(-1-1 1-3)	0 0
151	(-1-1 3-1)	0 0
152	(-3-3 1 1)	0 0
153	(-3-3 3 3)	0 0
154	(-3-1 1 3)	0 0
155	(-3-1 3 1)	0 0
156	(-1-3 1 3)	0 0
157	(-1-3 3 1)	0 0
158	(-1-1 1 1)	0 0
159	(-1-1 3 3)	0 0
160	(-3 1-3-3)	0 0
161	(-3 1-1-1)	0 0
162	(-3 3-3-1)	0 0
163	(-3 3-1-3)	0 0
164	(-1 1-3-1)	0 0
165	(-1 1-1-3)	0 0
166	(-1 3-3-3)	0 0
167	(-1 3-1-1)	0 0
168	(-3 1-3 1)	0 0
169	(-3 1-1 3)	0 0
170	(-3 3-3 3)	0 0
171	(-3 3-1 1)	0 0
172	(-1 1-3 3)	0 0
173	(-1 1-1 1)	0 0

174	(-1 3-3 1)	0 0
175	(-1 3-1 3)	0 0
176	(-3 1 1-3)	0 0
177	("3 1 3-1)	0 0
178	(-3 3 1-1)	0 0
179	("3 3 3-3)	0 0
180	(-1 1 1-1)	0 0
181	(-1 1 3-3)	0 0
182	(-1 3 1-3)	0 0
183	(-1 3 3-1)	0 0
184	("3 1 1 1)	0 0
185	(-3 1 3 3)	0 0
186	("3 3 1 3)	0 0
187	("3 3 3 1)	0 0
188	(-1 1 1 3)	0 0
189	(-1 1 3 1)	0 0
190	'(-1 3 1 1)	0 0
191	(-1 3 3 3)	0 0
192	(1-3-3-3)	0 0
193	(1-3-1-1)	0 0
194	(1-1-3-1)	0 0
195	(1-1-1-3)	0 0
196	(3-3-3-1)	0 0
197	(3-3-1-3)	0 0
198	(3-1-3-3)	0 0
199	(3-1-1-1)	0 0
200	(1-3-3 1)	0 0
201	(1-3-1 3)	0 0
202	(1-1-3 3)	0 0
203	(1-1-1 1)	0 0
204	(3-3-3 3)	0 0
205	(3-3-1 1)	0 0
206	(3-1-3 1)	0 0
207	(3-1-1 3)	0 0
208	(1-3 1-3)	0 0
209	(1-3 3-1)	0 0
210	(1-1 1-1)	0 0
211	(1-1 3-3)	0 0
212	(3-3 1-1)	0 0
213	(3-3 3-3)	0 0
214	(3-1 1-3)	0 0
215	(3-1 3-1)	0 0
216	(1-3 1 1)	0 0
217	(1-3 3 3)	0 0
218	(1-1 1 3)	0 0
219	(1-1 3 1)	0 0
220	(3-3 1 3)	0 0
221	(³-3 3 1)	0 0
222	(3-1 1 1)	0 0
223	(3-1 3 3)	0 0
224	(1 1-3-3)	0 0

225	(1 1- 1-1)	0 0
226	(1 3- 3-D	0 0
227	(1 3- 1-3)	0 0
228	(3 1- 3-D	0 0
229	(3 1- 1-3)	0 0
230	(3 3- 3-3)	0 0
231	(3 3- 1-1)	0 0
232	(1 1- 3 D	0 0
233	(1 1- 1 3)	0 0
234	(1 3- 3 3)	0 0
235	(1 3- 1 1)	0 0
236	(3 1- 3 3)	0 0
237	(3 1- 1 1)	0 0
238	(3 3- 3 1)	0 0
239	(3 3- 1 3)	0 0
241	(1 1 3-D	0 0
242	(1 3 1-1)	0 0
243	(1 3 3-3)	0 0
244	(3 1 1-1)	0 0
245	(3 1 3-3)	0 0
246	(3 3 1-3)	0 0
247	(3 3 3-1)	0 0
248	(1 1 1 1)	0 0
249	(1 1 3 3)	0 0
250	(1 3 1 3)	0 0
251	(1 3 3 D	0 0
252	(3 1 1 3)	0 0
253	(3 1 3 1)	0 0
254	(3 3 1 D	0 0
255	(3 3 3 3)	0 0

r- APÊNDICE B

SIMULAÇÃO DE DESEMPENHO

Para comparar o desempenho de cada um cios algoritmos apresentados no capítulo 4 cora aqueles de decodificação por máxima verossimilhança, os pontos do retículo A_n devem ser transmitidos através de um canal aditivo gaussiano. Neste canal cada ponto $P = (p_1, \ldots, p_n)$ e A_n é perturbado por um vetor de ruído $S = (s_1, \ldots, s_n)$, gerando um ponto novo $X = (x_1, \ldots, x_n)$ e IR^n, onde

$$X = P + S = (p_j + s_t, \ldots, p_n + s_j . \tag{B:l}$$

Nos canais gaussianos, cada coordenada do vetor S tem distribuição normal de média 0 e variância a^2, onde

$$F(s) = \frac{s}{r \cdot i} \exp -x^2/2a^2 \, dx. \tag{B:2}$$

Denotamos esta distribuição como s ** N(0, ***a***). Obtemos uma distribuição gaussiana normalizada se cr²= 1. A geração de variáveis aleatórias seguindo esta distribuição normalizada pode ser obtida utilizando-se os métodos em [26] e [27]. 0 programa em B :1 é para encontrar um valor aleatório desta distribuição. A tabela (13:1) exibe a probabilidade de encontrar qualquer valor ^ ***s^.***

Para se obter uma distribuição s- N(0, cr²), o valor x_j obtido pela distribuição s^- N(0, 1), deve ser substituído pelo valor *y*-ᵐ ᵃ ˣ j • -Isto é uma conseqüência do fato de que se y é uma variável aleatória com distribuição normal n (0, cr²), então a variável aleatória a + by têm distribuição normal n (a+b/i, b²<7²) [28]. Nos canais gaussianos aditivos, o valor de cr² representa a potência média do ruído do canal [29], resultando em uma relação sinal/ruído

$$(B:3) \qquad A_N = \frac{P_{<A>}}{no^{-2}} - ,$$

onde P(A) é a energia média dos pontos utilizados do retículo.

Na saída do canal, o ponto recebido é decodificado usando-se o algoritmo, o qual pretendemos comparar o desempenho com outro de decodificação por máxima verossimilhança. Os dois pontos obtidos serão comparados com aquele que foi transmitido. Para cada ponto, contamos como um erro de decodificação se houver diferença em, pelo menos, uma coordenada (probabilidade de erro por ponto

n-dimensional).

Para cada valor da relação sinal/ruído, um número suficientemente grande de pontos deve ser transmitido. A estim tiva é feita por simulação Monte Carlo ,utilizando-se a freqüência relativa da ocorrência de erros na saída de cada decodificador [29].

A utilização da versão fraca da lei dos grandes números, de acordo com o teorema de BERNOULLI [28], assegura que os resultados obtidos convergem para os valores reais das probabilidades.

Notamos que o número de pontos a serem transmitidos deve ser aumentado à medida que a relação sinal/ruído cresce. De acordo com a equação (B:3), tal aumento da relação acarreta uma diminuição no número de erros provocados pelo canal, e conseqüentemente, a interpretação freqüentista utilizada para o cálculo das probabilidades de erro poderá não mais ser válida. Portanto, o número de pontos que devem ser transmitidos para se atingir uma probabilidade de erro q com espalhamento relativo K é [29]

(B:4) $$n \quad \frac{1 - q}{K^2 q}$$

Em particular, se q « 1,

$$(B:5) \qquad n = \frac{1}{K^2 q}.$$

Por exemplo, se a probabilidade de erro é da ordem de 10^{-3} (conhecimento *a priori*) e é desejado um espalhamento de 10%, então a simulação deve ser realizada para n da ordem de 10^5 pontos transmitidos.

B . l - PROGRAMA PARA GERAR UMA DISTRIBUIÇÃO GAUSSIANA NORMALIZADA

Um programa em linguagem C para gerar um número € R com distribuição gaussiana.

```
main
{
int  xdum;
      int set=0,xrand;
      float vvl,m,w2,trand,vl,v2,r,fac,gset,gasdev;
      if(set==0){
      loop:
      xrand=-xdum*rand();
      trand=rand3(xrand);
      vl=2.0*trand-l.0;
      xrand=-xdum*rand();
      trand=rand3(xrand);
      v2=2.0*trand-l.0;
      r=vl*vl+v2*v2;
      if(r>=l)goto loop;
      fac=sqrt(-2*log(r)/r);
      gset=vl*fac;
      gasdev=v2*fac;
      set=l;
```

```
      >
      else{
      gasdev=gset;
      set=0;}
      return gasdev;
>
```

Esta função serve para gerar um número entre - l e i com distribuição uniforme,

```
float rand3(dum)
      int dum;
{
      long int big=1000000000, seed=16180033, z=0, a[55], ff=0, k,
      next, nextp, j ;
      int  i,kk,ii;
      float fac,trand;
      fac=l.0/big;
      if(dum<0 ||ff==0){
      ff=l;
      j=seed-abs(dum);
      j=j%big;
      a[55]=j;
      k=l;
      for(i=l;i<55;i++){
      ii=(21*i)%55;
```

```
a[ii]=k;
k=j-k;
if(k<z)k=k+big;
j-a[ii];
}
for(kk=1;kk<5;kk++)
for(i=l;i<56;i++){
a[i]=a[i]-a[1+(i+30)%55] ;
if(a[i]<z)a[i]=a[i]+big;
 >
next=0;
nextp=31;
dum=l;
}
next=next+l;
if(next==56)next=l  ;
nextp=nextp+l;
if(nextp==56)nextp=l;
j =a[next]-a[nextp];
if(j<z)j=j+big;
a[next]=j;
trand=  j*fac;
return  trand;
```

1:2:2- QUANTIZAÇXO:

As saídas do processo de amostragem são números reais devido a natureza da função f(t), a qual pode representar, por exemplo, a voz do ser humano. Como as comunicações digitais trabalham com números inteiros (i.e. alfabetos discretos- como por exemplo, o caso do sistema binário), estas amostras devem ser restritas a um destes valores descritos, ou seja quantizados. Um quantizador é um sistema que converte um número real em um número inteiro ou valor discreto fixo de um conjunto de valores.

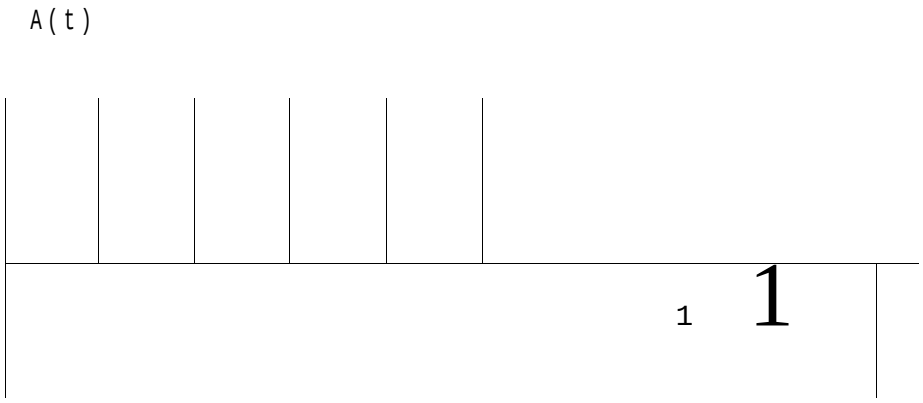


Figura (1:3). Processo de amostragem.

Tabela (B:l). Distribuição gaussiana obtida pelo programa no apêndice B:l, onde

$$F(x) = \exp^{-XV2} \quad dx$$

$$-J\ 27T$$

X	F(X)	X	F(X)
-4.00	0.0000	0.00	0.5003
-3.90	0.0000	0.10	0.5403
-3.80	0.0001	0.20	0.5797
-3.70	0.0001	0.30	0.6187
-3.60	0.0002	0.40	0.6539
-3.50	0.0003	0.50	0.6867
-3.40	0.0005	0.60	0.7406
-3.30	0.0005	0.70	0.7721
-3.20	0.0007	0.80	0.8009
-3.10	0.0010	0.90	0.8279
-3.00	0.0016	1.00	0.8513
-2.90	0.0023	1.10	0.8729
-2.80	0.0032	1.20	0.8933
-2.70	0.0040	1.30	0.9106
-2.60	0.0054	1.40	0.9257
-2.50	0.0066	1.50	0.9385
-2.40	0.0090	1.60	0.9490
-2.30	0.0113	1.70	0.9597
-2.20	0.0148	1.80	0.9674
-2.10	0.0187	1.90	0.9742
-2.00	0.0228	2.00	0.9799
-1.90	0.0287	2.10	0.9843
-1.80	0.0361	2.20	0.9877
-1.70	0.0466	2.30	0.9906
-1.60	0.0574	2.40	0.9929
-1.50	0.0698	2.50	0.9946
-1.40	0.0839	2.60	0.9958
-1.30	0.1013	2.70	0.9969
-1.20	0.1203	2.80	0.9979
-1.10	0.1416	2.90	0.9986
-1.00	0.1644	3.00	0.9989
-0.90	0.1898	3.10	0.9994
-0.80	0.2173	3.20	0.9995
-0.70	0.2475	3.30	0.9997
-0.60	0.2795	3.40	0.9998
-0.50	0.3128	3.50	0.9998
-0.40	0.3471	3.60	0.9998
-0.30	0.3840	3.70	0.9999
-0.20	0.4200	3.80	0.9999
-0.10	0.4606	3.90	0.9999

APÊNDICE C

A POTENCIA MEDIA DO RETÍCULO

PROPOSIÇÃO:

A potência média dos pontos em uma região limitada do retículo

$$A_n = 2^g Z^n + 2^{g-1} C_{g-1} + \cdots + 2^1 C_1 + C_0, \quad (1)$$

onde C_i é um código binário, e $C_i \subset C_{i+1}$, é igual à potência média dos pontos na mesma região do retículo não codificado Z^n.

PROVA:

Suponha que o retículo A_n foi construído utilizando-se uma constelação de 2^{2m} pontos, com coordenadas da forma

$$Z = \pm 1, \pm 3, \ldots\ldots, \pm(2^{2m}-1).$$

O retículo Z^n pode ser escrito na mesma forma da equação (1), l seja

(2) $$Z^n = 2^g Z\% - 2^{g-1} C_{g-l} + \_\_\_\_ + 2^1 C_1 * C_0,$$

onde C_i é o código universal (n,n,l). Seja

(3) $$P(p^1,..., p") = 2^g Z^n + 2^{g-1} B_t + \_\_\_\_ + 2^1 B_a + B_Q,$$

onde B e C e B = $(b^1.....b^n)$. um ponto arbitrário neste retículo
A potência média destes pontos é dada por

(4) $$IIPII \text{ » } IPII + I|P^2I + \ldots + |P"$$

(5) $$llp'1 = (2'Z' + 2'-' b' \quad 2^1 bj + b^1,)^2$$

$$= (2'z^1)^2 + (2'-V_1)^2 + \ldots + (2^1 b;)^2 + (b;)^2 +$$

$$2(2'z')b'_1 + \ldots + 2(2^9 z')bJ + 2(2^9 z')b^ +$$

$$2.2'-'b' \; b^1 + \ldots + 2-2'-V \; b' +$$
$$g-l \quad g-2 \qquad\qquad g-l\ 0$$

$$+ 2«2bV.$$
$$i\ o$$

Considerando todas as palavras do código universal, o número de l's é igual ao número de -l's para cada coordenada das palavras código. Portanto,

(6) $b_i = 0.$

Deste modo, a potência média da coordenada P^L é

(?) $|P'i = (2 \cdot zY_+(2' - v_1)^2_+ \ldots\ldots_+(2^1 {}_b;)^2_+(b;)^2.$

Conseqüentemente,

(8) $||P|| = (2^g Z^M)^2_+(2^6 - {}^1B_{g1})^2 + \ldots\ldots - r(2^1 B_J)^2 + (B_q)^2.$

A potência média dos pontos do retículo A^ pode ser dada pela equação (4), exceto que os vetores são palavras dos códigos C na equação (1). Nos códigos lineares, o número de l's é igual ao número de -l's para cada coordenada de todos as palavras código. Portanto

(9) $b_1 =$

Deste modo, a potência média dos pontos P é

(10) $$|PÍ = (2Y)^{2}_{+}(2^{H}B_{H})^{Z}_{+}.....{}_{t}(2^{1}{}_{B1})^{2}f(B_{0})^{2}.$$

Concluimos, então, que a potência média dos dois retículos é a mesma.

Q.E.D

REFERENCIAS BIBLIOGRÁFICAS

[1] C.E. SHANNON, "Communication in Presence of noise", *Proc. IRE,* 37, pp. 10-21, January 1948

[2] G.D. FORNEY, JR., R.G. GALLAGER, G.R. LANG, F.M. LONGSTAFF and S. U. QURESHI, "Efficient Modulation for Band-limited Channels", *IEEE Selected Areas in Communications,* vol. SAC-2, no.5, pp. 632-645, September 1984.

[3] N.J.A. SLOANE, "The Packing of Spheres", *Scientific American* , 250, pp. 116-125, Jan. 1984.

[4] F.F.E. OWEN, *"PCM and Digital Transmission Systems",* Mc Graw-Hill, 1982

[5] J.H. CONWAY and N.J.A. SLOANE, *"Sphere Packings, Lattices and Groups",* New York: Springer-Verlag, 1988

[6] H.M. de OLIVEIRA and G. BATTAIL, "On Generalized Constellations and Opportunistic Secondary Channel", *Ann Télécomm.* 47, no.5-6, pp. 202-213, 1992

[7] W.Godoy. JÚNIOR, *"Esquemas de Modulação Codificada com Códigos de Bloco",* Curitiba, CEFET-PR, 1991

[8] G. UNGERBOECK, "Channel Coding with Multilevel/phase

Signals", ***IEEE Trans. Inform. Theory***, vol 1T-28, no.l, pp. 55-67, January 1982

[9] A.R. CALDERBANK and N.J.A. SLOANE, "New Trellis Codes Based on Lattices and Cosets", ***IEEE Trans. Info. Theory***, IT 33, no.2, pp. 177-195, March 1987

[10] N.P. SECORD and R. DE BUDA, "A Two-Stage Sequential Demodulator for the Gosset Lattice", ***IEEE Selected Areas in Communication***, vol. SAC-7, no.6, pp. 974-981, August. 1989

[11] G.R. LANG and F.M. LONGSTAFF, "A Leech Lattice MODEM", ***IEEE Selected Areas in Communication"***, vol. SAC-7, no. 6, pp. 968-973, August. 1989

[12] J. LEECH, "Notes on Sphere Packings", ***Canad. J. Math.***, 19, pp. 251-267, 1967

[13] J.H.CONWAY and N.J.A.SLOANE, " Voronoi Regions of Lattices, Second Moment of Polytopes and Quantization", ***IEEE Trans. Inform. Theory***, vol IT-28, no.2, pp. 211-226, March 1982.

[14] H.M. de OLIVEIRA, W. ZHANG and G. BATTAIL, "On Performance of Lattice Codes on Gaussian Channels", ***Ann Telecomm.*** 47, no.7-8, pp. 293-305, 1992

[15] G.D. FORNEY, JR., "Coset Codes- part I : Introduction and Geometrical Classification", ***IEEE Trans. Inform. Theory***, vol IT-34, no.5, pp. 1123-1151, September 1988

[16] J.H. CONWAY and N.J.A.SLOANE, "Fast Quantizing and Decoding Algorithms for Lattice Quantizers and Codes", *IEEE Trans. Inform. Theory,* vol IT-28, no.2, pp. 227-231, March 1982.

[17] N.J.A. SLOANE, "Tables of Sphere Packings and Spherical Codes", *IEEE Trans. Inform. Theory,* vol IT-27, no.3, pp. 227-231, May 1981.

[18] J. LEECH and N.J.A SLOANE, "Sphere Packing and Error-Correcting Codes",, *Canad. J. Math.*, 23, pp. 718-745, 1971

[19] E.S. BARNES and N.J.A. SLOANE, "New Lattice Packings of Spheres", *Can. J. Math.,* 35, no.l, pp. 117-130, 1983

[20] J.H. CONWAY and N.J.A. SLOANE, " Soft Decoding Techniques for Codes and Lattices, Including the Golay Code and the Leech Lattice", *IEEE Trans. Inform. Theory,* vol IT-32, no.l, pp. 41-50, January 1988.

[21] G.D. FORNEY, JR., "Coset Codes- part II : Binary Lattices and Related Codes", *IEEE Trans. Inform. Theory,* vol IT-34, no.5, pp. 1152-1187, September 1988

[22] G.D. FORNEY,JR., "A Bounded-distance Decoding Algorithm for the Leech Lattice", *IEEE Trans. Inform. Theory,* vol IT-35, no.4, pp. 906-909, July/Aug 1989

[23] Y. BE'ERY, B. SHAHAR and J. SNYDERS, "Fast Decoding of Leech Lattice", *IEEE Selected Areas in Communication,*

vol. SÀC-7, no.6, pp. 959-967, August. 1989

[24] F.J. MacWILLIAMS and N.J.A. SLOANE, *"The Theory of Error-correcting codes"*, Amsterdam, 1986

[25] A.M. MICHELSON, *"Error-Control Techniques for Digital Communication"*, New York, John Wiley, 1985.

[26] S.A. TEUKOLSKY, B.N.P. FLANNERY and W.T. VETTERLING, *"Numerical Recipes , The Art of Scientific Computing"*, pp.202-203, Cambridge University press, 1986

[27] D. E. KNUTH, " *The Art of Computer Programming Seminumerical Algorithms*", 2nd volume, Adison Wesley, in: Computer Science and Information Processing, Mass. 1973

[28] P.G. HOEL, S.C. PORT and C.J. STONE, *"Introduction to Probability"*, Boston, Houghton Mifflin Company, 1971

[29] H.M. DE OLIVEIRA, *"Técnicas de decisão suave"*, Tese de Mestrado, Universidade Federal de Pernambuco, Recife, 1983

[30] M.A.O. DA COSTA E SILVA and R. PALAZZO JR., " A Bonded-Distance Algorithm for Lattices Obtained from a Generalized Code Formula", to be published.

[31] A.J. VITERBI and J.K. O'MURA, *"Principles of Digital Communication and Coding"*, NY, McGraw-Hill, 1979

[32] H. TAUB and B.L. SCHILING, *"Principles of Communication Systems"*, Tokyo, McGraw-Hill Koyakusha, 1971

[33] H.M. de OLIVEIRA, "Systèmes *de Modulation Codée pour la Transmission Numérique* à *Débit Elevé",* Thèse de Doctorat de l'Ecole Nationale Supérieure des Télécommunications, TELECOM Paris 92 E 001, Paris, 1992

[34] A.R. CALDERBANK, "Binary Covering Codes and High Speed data Transmission", *Mathematical Intelligence,* Eurocode, 90, Inter. Symp. on Coding and Application, Proceedings, Springer Verlag, G. Cohen and P. Charpin Eds, 1991, PP 320-336.

[35] T.M. THOMPSON, "From *Error-correcting Codes Through Sphere Packings to Simple Groups",* The Mathematical Association of America, USA 1983

[36] A.J. JERRI, "The Shannon Sampling Theorem-Its Various Extensions and Applications: A Tutorial Review", *Proceeding of IEEE,* vol 65, no. 11, pp. 1565-1596, November 1977

Como mostra a figura (1:4), o processo de quantização é realizado através da divisão do eixo de amplitude das amostras em N segmentos de amplitude AV -no caso de quantização uniforme-, cada K segmentos representa um valor discreto Y= K * AV o qual pode ser transmitido. Após a aquiesição da amostra , o quantizador escolhe o valor mais próximo. Vamos representar este processo como uma função discreta y = p(x). Em um intervalo de n amostras $(x_1, x_2, ..., x_n)$ a saída do quantizador será (Yj» $Y_1, t \cdot \cdot \cdot \cdot t\ Y_n$) •

valor da amostra.	2.8 4.2 5.2 3.2 1.8 0.2 -0.8 -2.2 -3.8	
nível de amostra-gem mais próximo.	3.0 4.0 5.0 3.0 2.0 0.0 -1.0 -2.0 -4.0	

Figura (1:4). Quantização unidimensional.

Ao contrário do processo de amostragem, o processo de quantização introduz um erro irreversível que modifica o sinal original. A magnitude deste erro é medida pela diferença entre o valor ***x^*** e $y_i = p(x)_\#$ ou seja, a distância Euclidiana em um espaço de uma dimensão;

(1:2) $e(x_i)^2 = [x - p(X_t))^2$.

A função efx^) é uma variável aleatória, devido ao fato de ***x^*** o ser também. Se ***x^*** tem uma distribuição uniforme, efx^ também tem distribuição uniforme no intervalo [-AV/2, AV/2] com uma média [4];

(1:3) $E(e(x)) = (AV)^2/12 = 0.0833\ AV.$

Este valor é denotado como o erro de quantização $\mathbf{N_q}$, o qual pode ser reduzido pela diminuição do valor AV, ou seja, pelo aumento do número de níveis.

A quantização pode ser realizada em duas dimensões ou mais, bem como ao longo de um eixo. Imagine que o plano seja dividido em regiões, não necessariamente congruentes, e imagine que em cada região um ponto seja marcado, como mostra a figura (1:5). Tal arranjo de pontos e regiões pode funcionar como um quantizador em duas dimensões. A entrada do quantizador é um par de números reais (x_1, x_2) que indicam um ponto x no plano, enquanto a saída é um ponto quantizado pré-escolhido que cai na mesma região, ou seja,

o ponto mais próximo entre todos os pontos. No caso do exemplo na figura (1:5) é o conjunto **p »P₂«. . . .»Pₙ** • Portanto qualquer ponto no plano pode ser substituído pelo ponto mais próximo do conjunto.

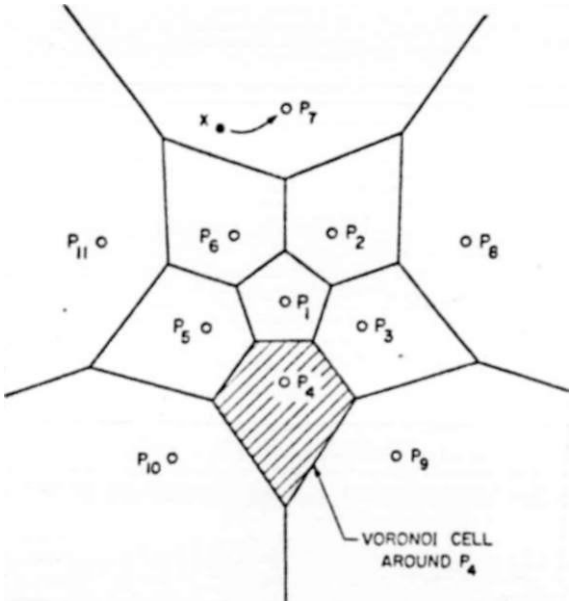


Figura (1:5). Quantizador em duas dimensões. Qualquer ponto no plano pode ser reduzido a um dos pontos Pi, **P2**,.. ...,Pi1.

Como na quantização em uma dimensão, este processo introduz erro, o qual depende da maneira de se escolher as regiões. Se o plano for dividido em quadrados iguais e os pontos do quantizador forem colocados no centro de cada quadro, o erro de quantização ainda é equivalente a $(AV)^2/12 = 0.0833AV^2$, onde AV é o comprimento do quadro [3]. Fejes Tóth, entre outros, (apud [5]), mostrou que se o plano for dividido em hexagonos com a mesma área do quadro anterior, o erro de quantização será diminuído a

(1:4) $N_q = 5\ AV^2/36vT\sim = 0.0802...AV^2.$

Zador (apud [5]) mostrou que é possível diminuir a média de erro por dimensão através do uso de quantizadores em altas dimensões. Infelizmente este resultado não é construtivo, ou seja, não exibe qual a maneira mais eficiente de quantizar em n dimensões.

A quantização é uma maneira de cobrir todo o espaço em n dimensões com áreas idênticas. .Cada área tem um centro, o qual representa o ponto quantizado. Isto ' equivale à colocação de esferas idênticas no espaço, de tal maneira que cada ponto contido neste espaço pertença, pelo menos, a uma esfera. O ponto quantizado é o centro de esfera mais próximo, e é claro que procuramos diminuir o número de esferas. Esta questão é um problema conhecido na matemática como "o problema de esferas intercaladas". A resposta deste problema é conhecida para apenas uma, duas e três dimensões . São listadas algumas soluções para outras dimensões [5] mas, não foi provado que essas correspondem aos melhores arranjos. Uma maneira de se encontrar uma embalagem de esferas intercaladas é tomar uma embalagem ótima de esferas não intercaladas em n dimensões e inflar cada esfera até que preencham o espaço vazio existente entre elas [3]. A figura (1:6) mostra esta operação em duas dimensões, onde os círculos ficaram hexagonais ao final da inflação. Esta operação não dá, necessariamente, a melhor solução ao problema, mas os melhores quantizadores conhecidos são aqueles

obtidos através deste processo usando-se as melhores embalagens de esferas [5]. Por isso, estamos interessados em estudar as embalagens de esferas não intercaladas.

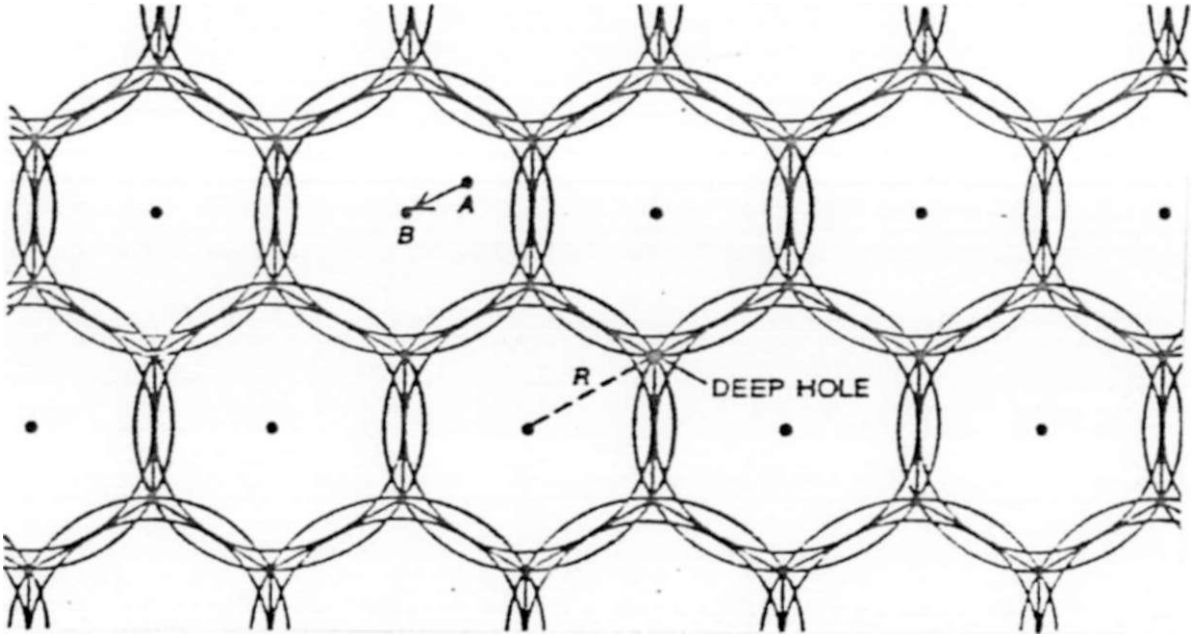


Figura (1:6). Um dos métodos de construir um quantizador que minimiza o erro da quantização em n dimensões é escolher uma embalagem ótima e inflar as esferas até que preencham todo o espaço. A figura mostra este processo em duas dimensões [2].

1:2:3- CODIFICAÇÃO DE FONTE

As amostras na saída do quantizador serão substituídas por uma seqüência de **n** letras de um alfabeto de tamanho **k.** No caso binário, as letras são 0 e 1. Se, por exemplo, os níveis de quantizaçáo são 256, caso seja realizada em uma dimensão, precisaremos de 256 seqüências binárias de 8 letras para representá-los.

1:2:4- CODIFICAÇÃO DE CANAL

Para um transmissão eficiente, o código de fonte que representa os valores das amostras deve ser codificado por outro código, o qual é referido como "o código de canal". Esta codificação envolve a adição de dígitos de redundância à seqüência da fonte, a fim de descobrir ou corrigir erros. Este aumento de dígitos pode resultar em conseqüências: Na primeira, a taxa de transmissão de informação diminui; na segunda, ocorre um aumento no atraso do transmissor e do receptor, e finalmente um aumento na complexidade do sistema de codificação e decodificação.

Um código de comprimento **n** que contenha 2 palavras código e com uma distância de Hamming mínima **d** entre elas é denotado como (r.,**k,d**). Em um código prático e eficiente, o valor de **n** é pequeno, o de **k** é grande (para aumentar a taxa de transmissão), e **d** é também grande (para corrigir muitos erros). Um dos problemas

relacionados com os códigos corretores de erro é o seguinte: dado n e d encontre A(n,d), o qual corresponde ao número máximo de palavras código no código (n,k,d). Em geral, este problema não é resolvido. De qualquer modo, cotas inferiores e superiores para A(n,d) foram encontradas e a construção de muitos códigos é conhecida [5].

Existem ligações fortes entre códigos e embalagem de esferas e veremos que boas embalagens podem ser construídas através destes códigos [5].

1:2:5- MODULAÇÃO

A função do modulador é unir a saída do codificador ao canal de transmissão. O modulador aceita símbolos binários ou seqüências M-árias (seqüências de $\log_2 M$ bits binários) e produz formas de ondas adequadas ao meio físico de transmissão, que é sempre analógico. Em muitos sistemas de comunicações digitais, onde a codificação deve ser aplicada, os métodos de modulação e demodulação são difíceis ou impossíveis de serem modificados. Em outros casos, a técnica de modulação é fixa mas, mudanças na demodulação são possíveis [31]. Ainda em outras aplicações é possível projetar a modulação e a demodulação juntamente com as técnicas de codificação (modulação codificada).

Na modulação binária, o modulador, simplesmente, converte o dígito binário, 0 ou 1, em formas de ondas, digamos

s (t) ou $s_{q}(t)$, respectivamente, com duração T_{g}. Para a modulação M-ária, os M símbolos possíveis e codificados são convertidos em um grupo correspondente de formas de ondas: $s_{0}(t)$, s ^ t) ,.....e s (t) .
M- 1 '

Entre uma grande diversidade de técnicas de modulação, existem três sistemas convencionais de modulação binária:

1) - Chaveamento por deslocamento da amplitude (PAM), onde os dois sinais têm as seguintes formas :

(l:5:a) $^{s}j(t) = A \sin wt$, e

(l:5:b) $^{s}{}_{0}\Lambda^{t}\Lambda = {}''^{A \sin wt}>$

2)- Chaveamento por deslocamento da fase (PSK), onde os dois sinais têm as seguintes formas :

(l:6:a) $»_{t}(t) = \sin wt$, e

(l:6:b) $^{s}{}_{0}\Lambda^{t}\Lambda = \sin (^{wt+18}°);$ e

3)- Chaveamento por deslocamento da freqüência (FSK), onde os dois sinais têm as seguintes formas;

(l:7:a) s^t) = sin **Zirf t**, e

(l:7:b) $^{s}{}_{0}\Lambda^{t}) \sim {}^{\sin} 2wf\ t.$

Como notado, os dois sinais nos três sistemas são funções de dois valores de amplitude, fase, ou freqüência, respectivamente. Enquanto que na modulação M-ária os M sinais são funções de M valores de amplitude, fase, ou freqüência.

O motivo de usar-se a modulação M-ária é que nos canais limitados em banda, a taxa de transmissão bits/segundo é baixa no sistema de modulação binária e a única solução é transmitir mais bits por Hz. A existência de mais de dois níveis de fase, freqüência ou amplitude aumenta a probabilidade de erro, como mostra a figura (1:7) no caso de modulação PSK. Tal prejuízo pode ser evitado ou controlado através dos sistemas de codificação de canal anterior.

Existe um sistema que combina os dois sistemas PSK e PAM [2] que é chamado de modulação de amplitudes em quadratura denotado como **QAM** . Um sistema **QAM** pode ser usado para gerar qualquer *"standared linear double-sideband modulated signal"*, o que inclui todos os tipos de modulação geralmente usados [2]. A figura (1:8) mostra um modulador **QAM** canónico.

No sistema QAM, os sinais são transmitidos através de um canal bidimensional, tal como na figura (1:9), em pares (x ,y). Tais pares são chamados "pontos de sinalização" ou "símbolos", os quais jazem em um plano de duas dimensões. Os pontos são transmitidos em taxa fixa de F símbolos/segundo, que é a banda nominal do canal. Como indica o modelo, as duas coordenadas do ponto são transmitidas independemente sobre o canal bidimensional e

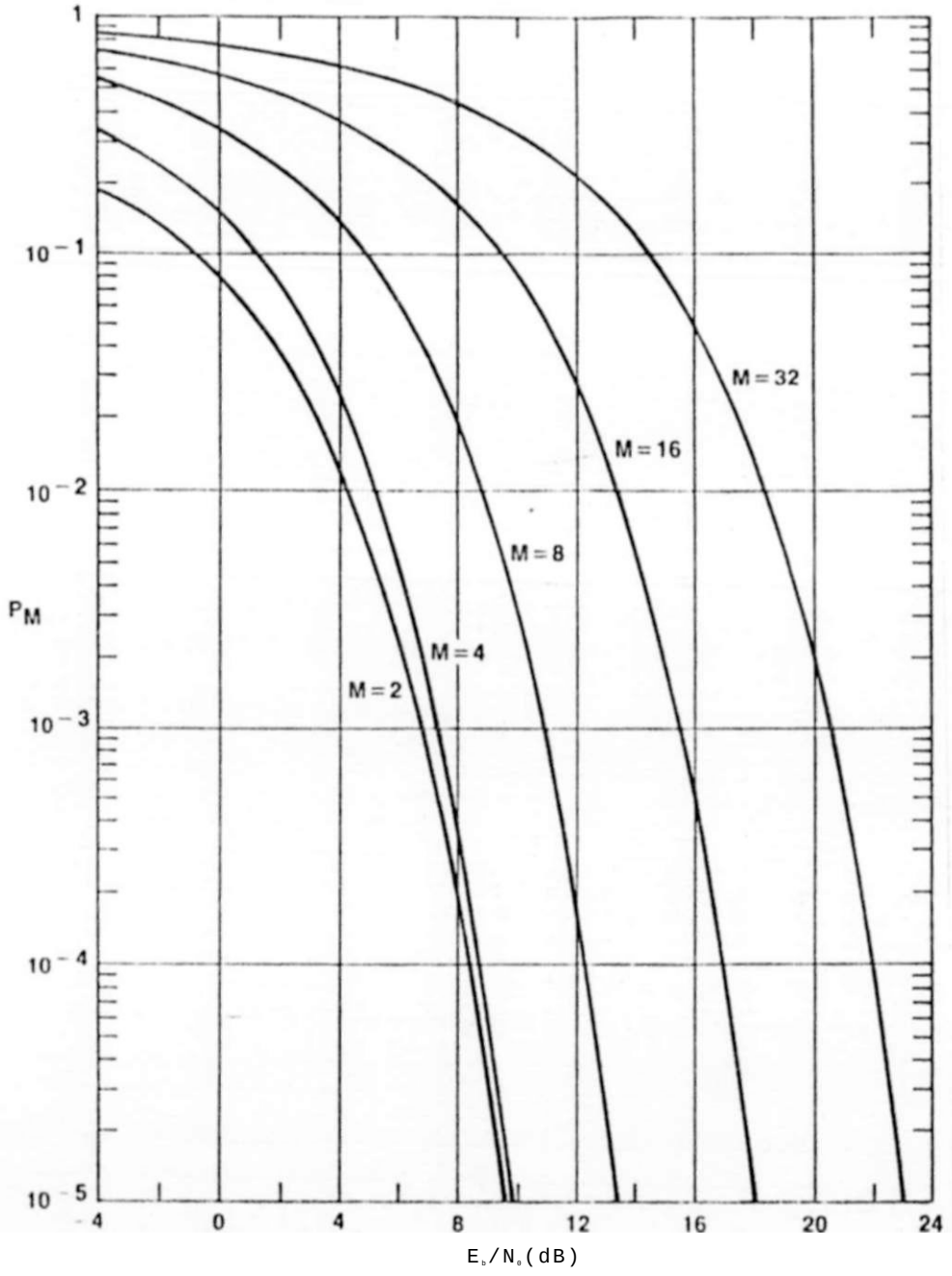


Figura (1:7) Probabilidade de erro de símbolo para modulação M—ária PSK [4] ,

são perturbadas por variáveis de ruído Gaussiano $(n_x, n_y)_i$, cada uma com variância N e média zero.

Para se transmitir m bits por símbolo (ponto) em QAM, o plano bidimensional deve possuir 2^m pontos. Tal arranjo de pontos é chamado constelação. Existem vários e diferentes métodos de se arranjar estes pontos na constelação [2] [6] e [7]. Estas variantes procuram diminuir a potência média dos pontos e

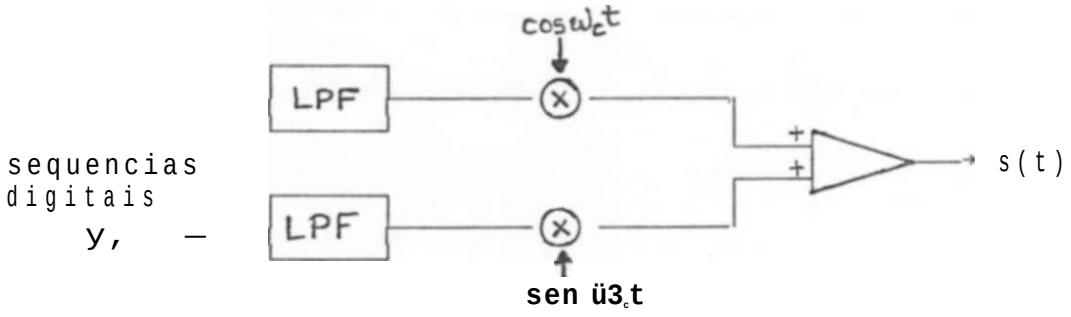


Figura (1:8). Modulador QAM cónico

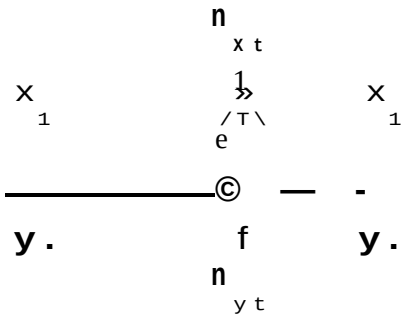


Figura (1:9) Canal bidimensional

simplificar o mapeamento entre os dígitos binários e os pontos. A constelação mais simples é a que se chama constelação retangular (figura (1:10)) desenvolvida por Campopiano e Glazer (apud [2]), onde para um número m inteiro e par de bits/símbolo, as constelações devem possuir 2^m pontos extraídos de uma grade retangular com coordenadas inteiras e ímpares, convencionalmente ±1, ±3, ±5,. . . . . , $\pm(2^{n/2}-1)$ [2].

1.3 - MODULAÇÃO CODIFICADA

Tratando-se o processo de modulação separadamente do processo de codificação, são introduzidas duas conseqüências indesejáveis:

1- a adição de dígitos de redundância no processo de codificação reduz a taxa efetiva de informações por Hz; e

2- a decisão abrupta de amplitude ou de fase feita no demodulador antes da decodificação causa uma perda irreversível de informações , (a qual equivale a cerca de 2 dB na relação sinal/ruído (SNR) [8]) .

n = 2 n = 3

n = 4 n = 5

Figura (1:10) constelações retangulares de sinais [2].

Uma alternativa para evitar-se tais conseqüências [7] é usar um conjunto maior de sinais no sistema de modulação. Neste caso, a redundância necessária ao processo de decodificação é fornecida pelo aumento do número de símbolos codificados. Essa solução implica, necessariamente, no uso de modulação não binária. Tal alternativa se chama "modulação codificada".

Apesar de existirem alguns trabalhos anteriores [7] [33] a área de modulação codificada foi sistematizada com o trabalho de Ungerboeck [8]. O autor mostrou o que teoricamente se

poderia atingir em termos de ganho de codificação sem sacrificar a taxa de informação transmitida, e apresentou sistemas práticos para a busca de novos e melhores códigos para esses sistemas. Ungerboeck conclui em seu trabalho que :

1)- ganhos de codificação podem ser obtidos utilizando-se alfabetos redundantes sem perdas na taxa de transmissão;

2)- a maior parte do ganho pode ser atingida expandindo-se a constelação por um fator de apenas duas vezes.

De acordo com Ungerboeck., tais ganhos podem ser realizados ou por códigos de bloco ou por códigos de treliça. Por sua potencialidade, os últimos foram adotados nos primeiros sistemas de Ungerboeck e a principal parte da maioria dos sistemas construídos posteriormente.

O princípio da modulação codificada, que foi estabelecido por Ungerboeck, é o que se chama de *"Mapeamento por partição de conjuntos"*, no qual um alfabeto de sinais é dividido em subconjuntos de maneira que a distância Euclidiana aumenta progressivamente a medida em que se prossegue com a partição. Dependendo deste princípio, vários métodos de modulação codificada foram construídos. Contudo, a confirmação de Shannon sobre a existência de uma embalagem de esferas em um espaço de alta dimensão desencadeou a construção de vínculos entre o estudo de embalagens e a teoria da codificação. A busca sobre a utilização de embalagens mais densas como métodos de modulação codificada teve

seu sucesso com o trabalho de Forney et *al*. **[2]** em 1984, no qual sistemas de modulação codificada em bloco e em treliça foram construídos utilizando-se o princípio de Ungerboeck e usando-se os retículos mais densos nas dimensões 4, 8, 16, e 24-a partir dos quais foram obtidos ganhos de codificação em torno de 1.5, 3.0, 4.5, 6.0 dB, respectivamente **[2]**. A mesma idéia foi descoberta separadamente por Calderbank e Sloane em 1987 [9], utilizando códigos de treliça. Existem outros sistemas especias que utilizam retículos em dimensões definidas, .como aquele de N. Secord e de Buda **[10]**, empregando o retículo de Gosset em 8 dimensões, e aquele de R. Lang e M. Longstaff [11], utilizando o retículo de Leech **[12]** em 24 dimensões.

A utilização de retículos mais densos em altas dimensões tem a finalidade de aproximar-se do sistema ideal de Shannon, que garante a possibilidade de transmitir com potência cerca de 9 dB menor do que àquela necessária em PAM e com uma probabilidade de erro muito pequena no caso de ruído Gaussiano [2]. O sistema prático e mais complexo, utilizando retículos, que já foi construído, é baseado no retículo de Leech em 24 dimensões o qual oferece 6 dB de ganho em potência [2]. Do ponto de vista de complexidade de decodificação, torna-se virtualmente impossível construir um sistema para oferecer mais ganho. Portanto, os estudos atuais estão freqüentemente interessados em simplificar os métodos de implementação dos sistemas já existentes, ou construí-los de outra maneira.

A implementação de um sistema deve incluir métodos para facilitar três processos :

1)- mapear as seqüências binárias nos pontos lo 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 segundo processo é tratado separadamente dos demais. Neste trabalho, desenvolveremos um sistema de codificação e decodificação que inclui a simplificação da complexidade dos três processos juntos. Este sistema pode ter semelhança com outros sistemas, mas as diferenças que existem são básicas.

Será considerado neste trabalho apenas o caso de modulação codificada em bloco.

r CAPÍTULO 2

EMBALAGEM DE ESFERAS E RETÍCULOS

2:1) INTRODUÇÃO

Existem três problemas ligados a embalagem de esferas em um espaço n-dimensional:

1- Qual é o número máximo de esferas idênticas que podem ser embaladas em uma dada região do espaço?

2- Qual é o número máximo de esferas idênticas que podem tocar a superfície de uma esfera do mesmo tamanho?

3- Qual é o número mínimo de esferas idênticas que podem cobrir todo o espaço de tal maneira que qualquer ponto se encontre pelo menos em uma esfera?

A primeira pergunta é conhecida como o "problema de embalagem de esferas", a segunda é conhecida como o "problema do número de contato ou número de Newton" **[5]**, e a última como o "problema de cobertura de espaço".

No capítulo 1, justificamos a importância de se responder estas perguntas, observando que as duas primeiras são

intimamente ligadas á codificação de mensagens transmitidas através de canais gaussianos ruidosos, enquanto a terceira é ligada á quantização ou decodificação de mensagens contínuas. Realmente, não estamos preocupados com o número de esferas como resposta a qualquer uma destas perguntas, mas com a maneira de se obter tais respostas. Portanto, os dois primeiros problemas estão da mesma classe e podem ser tratados conjuntamente, enquanto o terceiro é independente. Vale salientar que uma resposta ótima para a primeira pergunta fornece uma boa resposta para a terceira pergunta, mas não necessariamente ótima [3]. Portanto vamos estudar somente as embalagens densas de esferas que não sejam intercaladas.

2:2) REPRESENTAÇÃO MATEMÁTICA DE HIPER-ESFERAS

Um ponto X em um espaço de n dimensões IR^n é uma série de n números,

(2:1) $$X = (x_1, x_2, x_3, \ldots, x_n).$$

Uma hiperesfera em IR^n de raio p com centro no ponto U $(u_1, u_2, u_3, \ldots, u_n)$ corresponde a uma superfície que inclui todos os pontos $X = (x_1, x_2, x_3, \ldots, x_n)$ que satisfazem a relação

(2:2) $$(x_1 - u_1)^2 + (x_2 - u_2)^2 + \ldots + (x_n - u_n)^2 = p^2.$$

Uma embalagem de esferas em (R^n pode ser definida, então, pela especificação dos centros das esferas e seu raio.

Existem dois tipos de embalagens, dependendo da maneira de se especificar os centros das esferas embaladas. Primeiro: *embalagens não reticuladas,* nas quais não existe uma relação linear entre os centros das esferas. Tais embalagens são difíceis de serem estudadas, e portanto não estamos interessados nelas. O segundo tipo é aquele no qual existe uma relação linear entre os centros das esferas, . a qual se chama *embalagens reticuladas (retículos).*

2:3) EMBALAGENS RETICULADAS

Uma embalagem é chamada retículo se, sempre que existam duas esferas, uma que tenha centro no ponto X = ($X_1, X_2, X_3 \ldots X_n$) e outra que tenha centro no ponto Y = ($y_1, y_2, y_3 \ldots y_n$), então existem outras esferas na embalagem, as quais têm seus centros em todos os pontos da forma Z = ($z_1, z_2, z_3 \ldots z_n$), onde $z = a x + b y$, e a e b são números inteiros [3]. Portanto um retículo em R^n é totalmente definido por n pontos ou vetores linearmente independentes e não nulos, os quais se chamam de base do retículo[1]. Existem muitas maneiras de se

[1] De fato, a dimensão do retículo pode ser diferente da dimensão do espaço em que ele está definido [17].

escolher a base, mas esta escolha não altera a construção do retículo.

2:4) DEFINIÇÕES

2:4:1- Matriz geradora.

Supondo que os vetores

$$V_1 = (v_{11}, v_{12}, \ldots, v_{1n})$$

$$V_2 = (v_{21}, v_{22}, v_{23}, \ldots, v_{2n})$$

(2:3)

$$V_n = (v_{n1}, v_{n2}, v_{n3}, \ldots, v_{nn})$$

constituam a base do retículo A_n (0 índice indica a dimensão do retículo), a matriz

(2:4)

$$M = \begin{matrix} v_{11} & v_{12} & v_{13} & \ldots & v_{1n} \\ v_{21} & v_{22} & v_{23} & \ldots & v_{2n} \\ v_{n1} & v_{n2} & v_{n3} & \ldots & v_{nn} \end{matrix}$$

é chamada de matriz geradora do retículo, e os pontos de são todos os pontos da forma **K***M, onde **k = (k ,k ,k , ... »$_n$)** é um vetor arbitrário com componentes inteiras .

2:4:2- Cardinalidade do retículo em uma região R.

Definimos a cardinalidade do retículo A_n ,denotado por $|A_n|$ como o número de seus pontos existentes em uma dada região limitada R *Q* IR.

2:4:3- Potência média do retículo.

A potência de um ponto X do retículo, denotada como I |X| |², é a soma dos quadrados de seus componentes X_j, i . e . ,

(2:5) $$\mathbf{I\,IXI\,I^2 = x_1^2 + x_2^2 + \_\_\_\_ + x_n^2 .}$$

A potência média em uma região limitada de | A^ | pontos, X^e A^, do retículo A , denotada como P(A), é

$$P(A) - Z \frac{l\,l^{X\,j}H^2}{i\text{-}i \quad |A_n|}$$

onde L = |A | .

2:4:4- A região fundamental

A região fundamental de um retículo A é uma região de R^n que contém um e somente um ponto deste retículo, a qual, se for repetida, cobre todo o espaço [34].

Se V V é uma base do retículo A, então O paralelotopo que consiste dos pontos

$$H_1 V_1 + \ldots\ldots + U_n V_n \quad (O \le u_1 < 1)$$

é um exemplo de região fundamental.

A região de Voronoi [13] é um outro exemplo de região fundamental.

2:4:5- O volume fundamental.

Existem muitas maneiras de escolher a região fundemental, mas o volume desta região é unicamente determinado pelo retículo [34]. Este volume é chamado de volume fundamental $V(A_n)$ do retículo e é dado por [14]:

(2:7) $$V(A_n) = \det^{1/2} A,$$

onde $$A := M M^T,$$

sendo M a matriz geradora, onde T denota transposição, e := denota

igual por definição.

2:4:6- Número de contato.

O número dos pontos do retículo mais próximos a um ponto central que mantêm a mesma distância entre si e aquele ponto, é conhecido como "número de contato", ou "coeficiente de erro" [15].

2:4:7- Equivalência de retículos

Se dois retículos A e B diferem somente por uma rotação ou mudança de escala, diz-se que estes dois retículos são equivalentes [16], e denota-se por

(2:8) $A = B$.

2:4:8- Retículos duais.

Seja A um retículo definido pela base $\{v_1, v_2, \ldots, v_n\}$, e que contém todos os pontos $X = u_1 v_1 + u_2 v_2 + \ldots + u_n v_n$, onde $u_1, u_2, \ldots, u_n$ são números inteiros arbitrários. O retículo dual contém todos os pontos Y no espaço R^n tal que

$$X.Y = x_1 y_1 + x_2 y_2 + \ldots + x_n y_n$$

é um número inteiro para cada X e A_n. A maioria dos retículos conhecidos contêm seus duais **[16].**

2:5) EXEMPLOS DE RETÍCULOS.

Nós vamos estudar alguns retículos nas dimensões mais importantes no mundo prático das comunicações digitais. Como vamos observar, não há uma definição única nem uma forma única de construção de um retículo A_n.

2:5:1) DIMENSÃO N = 1.

Nesta dimensão o problema é trivial, e existe somente uma embalagem. Um centro de esfera é representado por uma coordenada e a esfera representa um segmento linear cujo comprimento é igual ao seu diâmetro (figura (2:1)). Cada segmento neste retículo toca outros dois, portanto o número de contato é 2. Este retículo é denotado como Z, o qual inclui todos os números inteiros. Não há nenhuma outra embalagem nesta dimensão.

Em qualquer outra dimensão, o retículo A^, que inclui todos os pontos que sejam de qualquer combinação de números inteiros Z = {...,-1,0,1,...}, é denotado por

C2:9D $Z \rightarrow \{x_1, x_2, x_3, \ldots, x_n : x_i \in Z\}$.

Tal retículo é chamado de retículo cúbico n-dimensional, e seu número de contato é T=2n [5J.

0 1 3 4 5

ESFERA

Figura (2:1). Retículo unidimensional Z.

2:5:2) DIMENSÃO N = 2.

Além do retículo Z_2 (figura (2:2)) existe outro retículo mais denso nesta dimensão (figura (2:3)) o qual é chamado de retículo hexagonal. Uma matriz geradora deste retículo pode ser a seguinte [5]

(2:10) $$M = \begin{matrix} 1 & 0 \\ 1/2 & JT/2 \end{matrix}$$ " I

O número de contato é ***z*** = 6, enquanto para Z_m é **t** = 4 [5]

2:5:3) DIMENSÃO N = 3.

Além do retículo Z_3 existe outro mais denso chamado de retículo cúbico de face centrada ***(centered cubic lattice),***

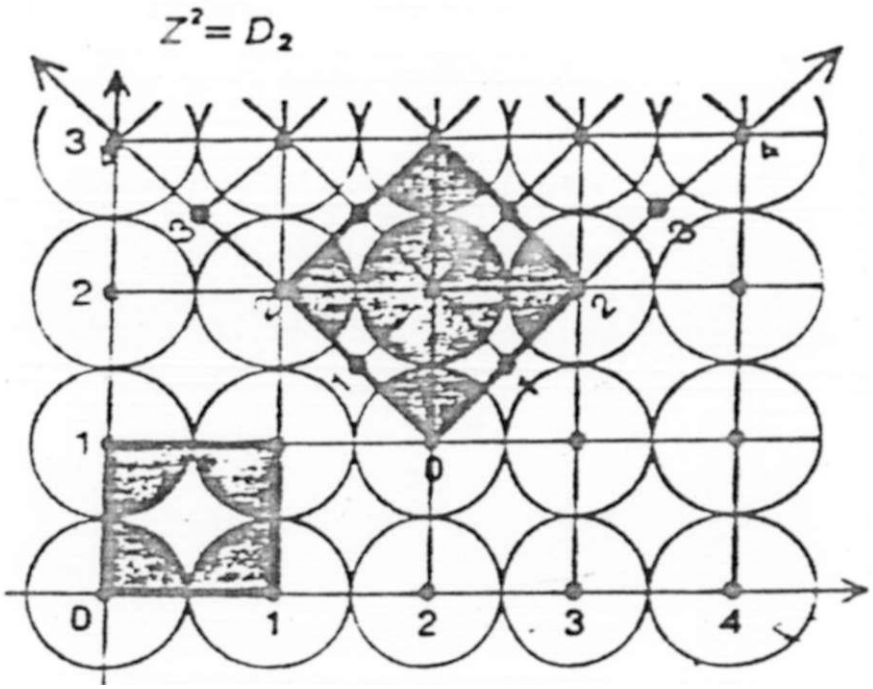


Figura (2:2) Embalagem retangular Z^2 onde os centros das esferas são todas as combinações dos números inteiros. Esta embalagem ocupa 0,7854 do plano [2]

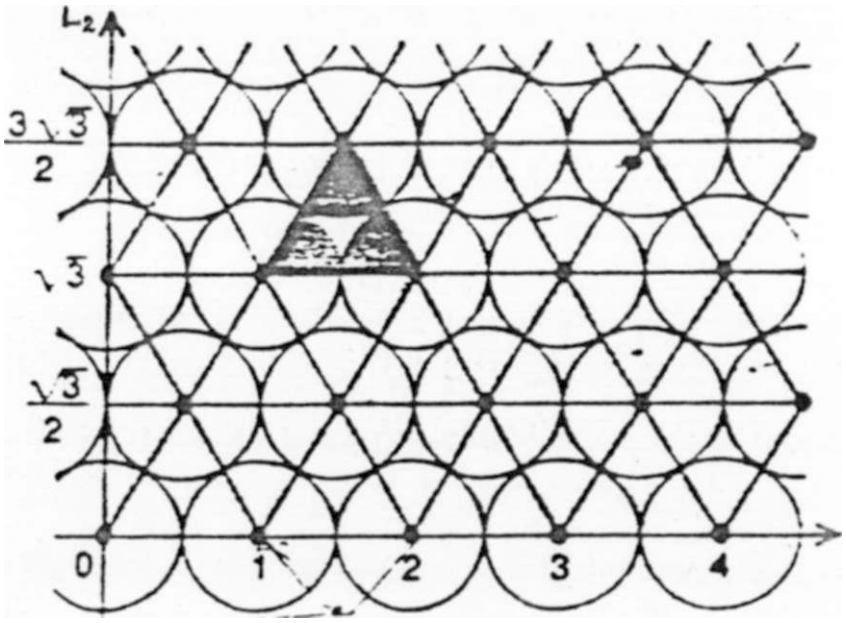


Figura (2:3), Embalagem hexagonal em duas dimensões, onde os centros de cada três esferas formam um triângulo. Esta embalagem ocupa 0, 9069 do plano [2].

denotado como D_3, que contém todos os pontos cujas coordenadas são números inteiros e cuja soma é um número par. Uma matriz geradora para D_3 é a seguinte:

(2:11)
$$M = \begin{matrix} -1 & -1 & 0 \\ 1 & -1 & 0 \\ 0 & 1 & -1 \end{matrix}$$

O número de contato é x = 12 [5].

Esta definição de D_3 pode ser generalizada para dimensões n £ 3 gerando os retículos

(2:12)
$$D_n = \{ (x_1.x_2 ....,x_n) \in Z^n ; x_1 + x_2 + ... + x_n = 0 \pmod 2 \}.$$

O número de contato de D_n é r = 2n (n-1) [5]

2:5:4) DIMENSÃO N=4

O retículo mais denso nesta dimensão é D_4, definido na equação (2:13), o qual também é a embalagem mais densa. Uma matriz geradora dele pode ser [5]

(2:13)
$$M = \begin{matrix} 2 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{matrix}$$

a qual garante que a soma das coordenadas de qualquer ponto do retículo é par. O número de contato é **T** = 24. Veremos como é possível construir este retículo usando o método de Forney et *al.* [2], baseado na idéia de partição de conjuntos estabelecida por Ungerboek [8].

2:5:5) DIMENSÃO N = 8.

Além dos retículos Z_8 e D_8 há outro mais denso descoberto por Gosset em 1900 [5] denotado como E_8 o qual é uma embalagem de duas cópias de D_8. A definição usual deste retículo [5] é

(2:14) $$E_8 = \{(x_1, \ldots, x_8): \text{ todo } x_i \in Z \text{ ou } x_i \in Z+1/2, \text{ tal que } \pounds x_i = 0 \pmod 2\}.$$

O número de contato é x = 240. Uma matriz geradora de E_8 pode ser [5]

$$\begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 1/2 & 1/2 & 1/2 & 1/2 & 1/2 & 1/2 & 1/2 & 1/2 \end{bmatrix}$$

Outra construção baseada na partição de conjuntos [2] será dada.

2:5:6) DIMENSÃO N = 16.

Nesta dimensão existem, entre outros, os retículos Z^{16}. D_{16}^{+}, H_{16} e A_{16}. O retículo H_{16} tem a mesma construção de E_8 mas em um espaço de 16 dimensões. O retículo mais denso neste espaço é A_{16} que tem inúmeras construções, uma delas é obtida aplicando-se a construção código B (veja § 2:6), que resulta na matriz geradora [17]

(2:16) $M = 1/2$

$$\begin{matrix}
4&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
2&2&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
2&0&2&0&0&0&0&0&0&0&0&0&0&0&0&0\\
2&0&0&2&0&0&0&0&0&0&0&0&0&0&0&0\\
2&0&0&0&2&0&0&0&0&0&0&0&0&0&0&0\\
2&0&0&0&0&2&0&0&0&0&0&0&0&0&0&0\\
2&0&0&0&0&0&2&0&0&0&0&0&0&0&0&0\\
2&0&0&0&0&0&0&2&0&0&0&0&0&0&0&0\\
2&0&0&0&0&0&0&0&2&0&0&0&0&0&0&0\\
2&0&0&0&0&0&0&0&0&2&0&0&0&0&0&0\\
2&0&0&0&0&0&0&0&0&0&2&0&0&0&0&0\\
1&1&1&1&0&1&0&1&1&0&0&1&0&0&0&0\\
0&1&1&1&1&0&1&0&1&1&0&0&1&0&0&0\\
0&0&1&1&1&1&0&1&0&1&1&0&0&1&0&0\\
0&0&0&1&1&1&1&0&1&0&1&1&0&0&1&0\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1
\end{matrix}$$

O número de contato é $z = 4320$. Este retículo pode ser construído através do retículo A_{16}, usando-se o método de laminação

estabelecido por Leech e Sloane [3], [5],

2:5:7) DIMENSÃO N = 24.

Sem dúvida, o retículo mais importante nesta dimensão e em toda a teoria dos retículos é A_{24}, o qual foi descoberto pelo matemático John Leech [12] e é conhecido como o retículo de Leech. Existem vários métodos para se construir [5], um dos quais é baseado no código de Golay [35], dando a matriz geradora que está na figura (2:4) [17]. 0 número de contato deste retículo é x = 196560 [3].

2:6) CONSTRUÇÃO CÓDIGO DE RETÍCULOS

J. Leech e N. J. A. Sloane [18] estabeleceram os primeiros vínculos entre embalagem 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. Este tipo de construção é de grande importância nos métodos de decodif icação.

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

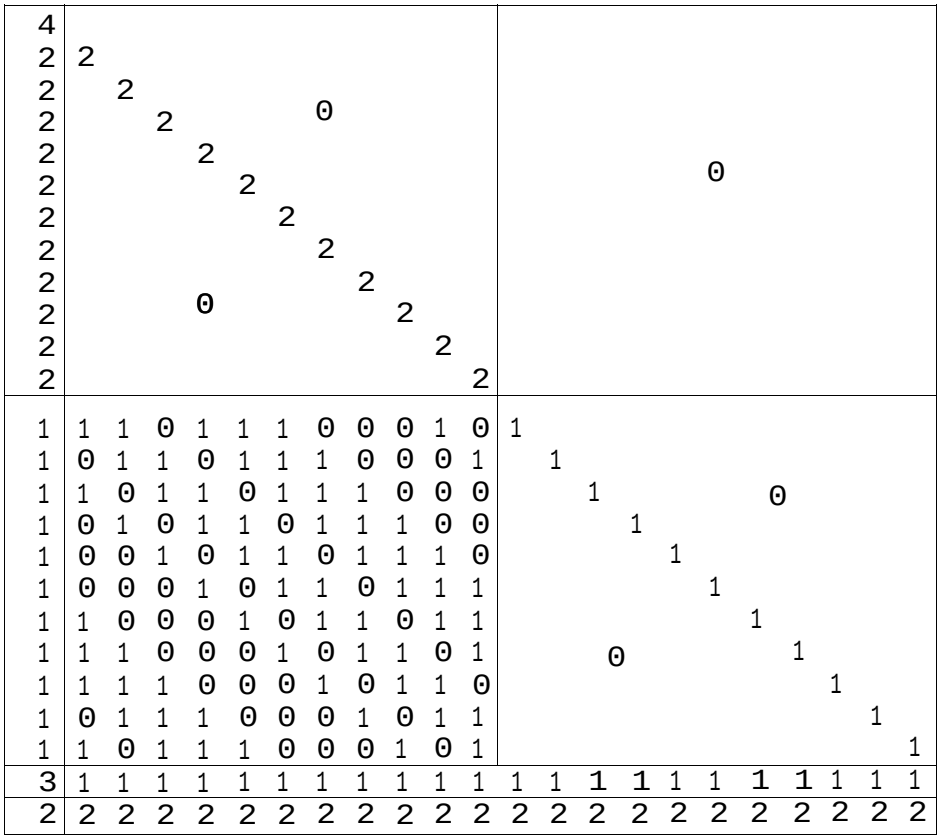


Fig. (2:4) Uma matriz geradora do retículo A_{24} - A matriz 11 x 1 no lado esquerdo inferior é tirada da matriz geradora do código de Golay (apud [17]).

generalização de A e B e é efetiva nas dimensões n = 2 [5], onde m = 1,2,....,k, e finalmente a construção D é efetiva em algumas dimensões como 36 e 64 [19].

2:6:1) CONSTRUÇÃO A.

Suponha que C é um código binário (n,m,d). A construção seguinte especifica um conjunto de pontos que forma uma embalagem de esferas em R^n :

Construção A. $X = (X , X .. . . . , X)$ é um ponto na embalagem se e somente se X for congruente (mod 2) com uma palavra código em C [18].

EXEMPLOS:

1- 0 retículo Z^n

Qualquer número inteiro pode ser escrito sob a forma:

$z = 2k+c$, onde $c = 0$ ou 1 e $k = 0, \pm1, \pm2, \ldots$

Como Z^n inclui todas as combinações de números inteiros, então sua fórmula código é

(2:17) $Z^n = 2Z^n + (n,n,l)$,

onde "+" denota soma de vetores n-uplas e Cn, n, l) é o código universal 120). Claro que esta fórmula satisfaz as condiçSes da construção À.

2- O retículo D_n.

Como este retículo inclui todos os pontos e Z^n cujas coordenadas somam-se a um número par, a fórmula código

(2:18) $D_n = 2Z^n + (n, n-l, 2)$

satisfaz esta condição, e ao mesmo tempo satisfaz as condições da construção A.

A tabela (2:1) mostra as fórmulas código que satisfazem a construção A para os retículos nas dimensões 4, 8, 16, 24 e 32 [15].

2:6:2) CONSTRUÇÃO B.

Seja C um código binário (n,m,d) comapropriedade de que cada palavra tenha peso par, então a construção seguinte especifica o conjunto de pontos que formam uma embalagem em IR"

A construção B. X = ($x_j, X_2, . . . , x_j$ é um ponto no retículo se e somente se X for congruente (mod 2) com uma palavra do código e £ ***x^*** for congruente (mod 4) com 0 [18].

Em outras palavras, os retículos que podem ser

representados pela fórmula

(2:19) $A = 4Z^{n} + 2C_{1} + C_{0}$,

onde C_q é um subcódigo de **C** , são incluídos na construção **B [15]. A** tabela **(2:2)** mostra a fórmula código de alguns destes retículos nas dimensões **16, 24** e **32 [15].**

Tabela **(2:1)** Fórmulas código de alguns retículos construídos usando-se a construção **A. (R-M : REED - MÜLLER)**

A	fórmula código	nota sobre os códigos
D_4	$2Z^4 + (4,3,2)$	código de um dígito de paridade
D_8	$2Z^8 + (8,7,2)$	código de um dígito de paridade
E_8	$2Z^8 + (8,4,4)$	código de **R-M** de primeira ordem
D_{16}	$2Z^{16} + (16,15,2)$	código de um dígito de paridade
H_{16}	$2Z^{16} + (16,11,4)$	código de **R-M** de segunda ordem
O_{24}	$2Z^{24} + (24,23,2)$	código de um dígito de paridade
X_{24}	$2Z^{24} + (24,18,4)$	
D_{32}	$2Z^{32} + (32,31,2)$	código de um dígito de paridade
H_{32}	$2Z^{32} + (32,26,4)$	código de **R-M** de segunda ordem

onde é uma palavra do código C^. Representamos isto pela fórmula código [15]

$$A_n = 2^k Z^n + 2^{k-1} C_{k-1} + \ldots\ldots + 2^{k''^2} C_{k-2} + 2C_1 + C_0 .$$

Tal construção é conhecida como construção D. No caso em que os códigos acima não são lineares ou C. não está incluído em C^, a construção é conhecida como construção C. As embalagens geradas neste caso são, geralmente, não reticuladas

2:7) RETÍCULOS E CODIFICAÇÃO.

Na introdução deste trabalho, mostramos o desenvolvimento histórico das ligações entre a teoria de retículos e a teoria de codificação, e mostramos os ganhos de codificação obtidos no uso de retículos como códigos.

A fim de utilizar os retículos como códigos de canal, eles devem ser construídos de maneira a facilitar dois processos principais:

1- um sistema simples de mapeamento entre as seqüências binárias e os pontos do retículo, ou vice versa; e

2- um processo simples, e o mais rápido possível, de decodificação. O decodificador é um sistema $<p$ (x) que encontra o ponto $\mathbf{y} = (y_1, y_2 \ldots\ldots, y_n)$ do retículo A_n mais próximo no sentido

2:6:3) CONSTRUÇÃO CÓDIGO GENERALIZADA

Suponha que $C^{\wedge}$, $_{2}$, ... /$^{c}{}_{0}$ são códigos binários com a condição de que C *Q* C . Então o retículo A inclui todos os pontos X, tal que

$$X = 2^{k}Z^{n} + 2^{k-1}c_{k-l} + \ldots\ldots + 2^{k-2}c_{k-2} + 2c_{1} + C_{0},$$

Tabela(2:2). Fórmulas código de alguns retículos construídos pela construção B. (R-M : REED - MÜLLER)

A	fórmula código	nota sobre os códigos
	$4Z^{16}+2(16,15,2)+(16,5,8)$	(16,5,8) é um código de R-M de primeira ordem.
H $_{24}$	$4Z^{24}+2(24,23,2)+(24,12,8)$	(24,12,8) é o código de Golay
A $_{24}$	$4Z^{24}+2(24,18,4)+(24,6,16)$	
H $_{32}$	$4Z^{32}+2(32,31,2)+(32,16,8)$	(32,16,8) é um código de R-M de segunda ordem.
A $_{32}$	$4Z^{32}+2(32,26,4)+(32,6,16)$	(32,26,4) e (32,6,16) são códigos de R-M de 3ª e Iª ordem, respectivamente

euclidiano de um ponto $x = (x_1, x_2, \ldots\ldots, x_n)$ e R^n.

Como notamos no capítulo 1, vários sistemas foram construídos nesta área considerando os retículos como códigos de bloco ou códigos de treliça. Forney et al. [2] apontaram que todos os sistemas conhecidos naquele tempo, incluindo os mais importantes códigos de retículos, podem ser gerados pelos mesmos elementos básicos:

1- um codificador convencional binário, de bloco ou de treliça, que opera com um determinado número de bits de dados para gerar um número maior de bits codificados;

2- estes bits codificados escolhem um dos subconjuntos de uma constelação particionada de sinais; e

3- um número adicional de bits não codificados escolhem um ponto individual do subconjunto escolhido.

A partição de uma constelação de n dimensões em subconjuntos é um sistema estabelecido por Ungerboeck [8], o qual a chamou "mapeamento por partição de conjuntos". Cnsidera-se inicialmente uma constelação bidimensional de pontos tirados de uma grade retangular de duas dimensões; tal constelação pode ser dividida em dois subconjuntos designando pontos alternativos a cada subconjunto, i.e., de acordo com o modelo da figura (2:5)
Os dois subconjuntos resultantes (A e B) tem as seguintes propriedades :

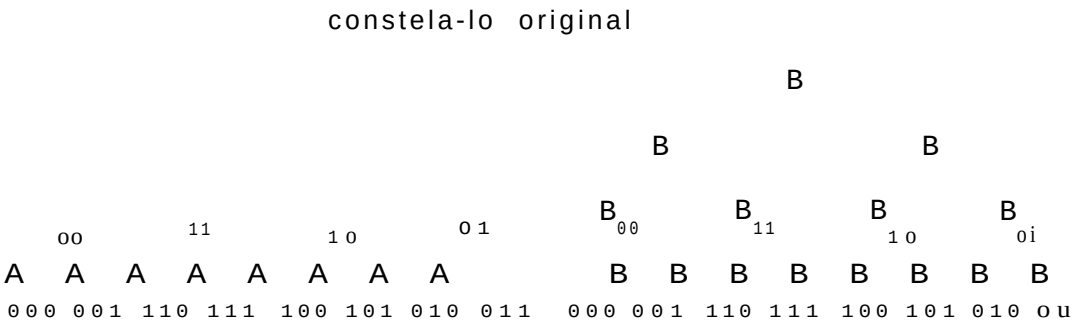
a) os pontos em cada subconjunto formam uma grade retangular (rotacionada 45° em relação à grade original).

A B A B A B
B A B A B A
A B A B A B
B A B A B A

Figura (2:5), Partição de uma constelação bidimensional em dois subconjuntos.

b) 0 quadrado da distância mínima entre os pontos de cada subconjunto é o dobro da distância mínima ao quadrado d^2 entre os pontos da constelação original.

Além disso, por causa da primeira propriedade, a partição da grade infinita pode ser repetida para se obter 4, 8, 16,.... subconjuntos com propriedades similares, e em particular com distância mínima ao quadrado entre cada subconjunto de 4, 8 ,16,... vezes *á*\ A figura (2:6) mostra uma constelação retangular de 16 pontos dividida em dois subconjuntos de 8 pontos (a), 4 subconjuntos de 4 pontos (b), 8 subconjuntos de 2 pontos (c), e 16 subconjuntos de 1 ponto (d) [2]. A atribuição de índices é feita de acordo com a árvore:



Devemos ressaltar que esta nomeação de índices facilita o cálculo da distância mínima ao quadrado entre os pontos no segundo e terceiro nível de partição, onde esta distância entre dois pontos do subconjunto A ou B é $2d^2$ vezes a distância de Hamming entre seus índices: e.g., $d(A^{\wedge}.A^{\wedge}) = 4d_{\varrho}$.

Estes subconjuntos podem ser usados para implementar um sistema de codificação relativamente simples e efetivo, ilustrado na figura (2:7), empregando os elementos básicos acima. 0 ganho de codificação é determinado pelas propriedades da distância mínima dos subconjuntos combinados com aquelas dos códigos binários, apesar do tamanho da constelação. Tal esquema pode ser usado para se construir retículos usando-se códigos de bloco nas dimensões 4, 8, 16, e 24 como mostrado a seguir.

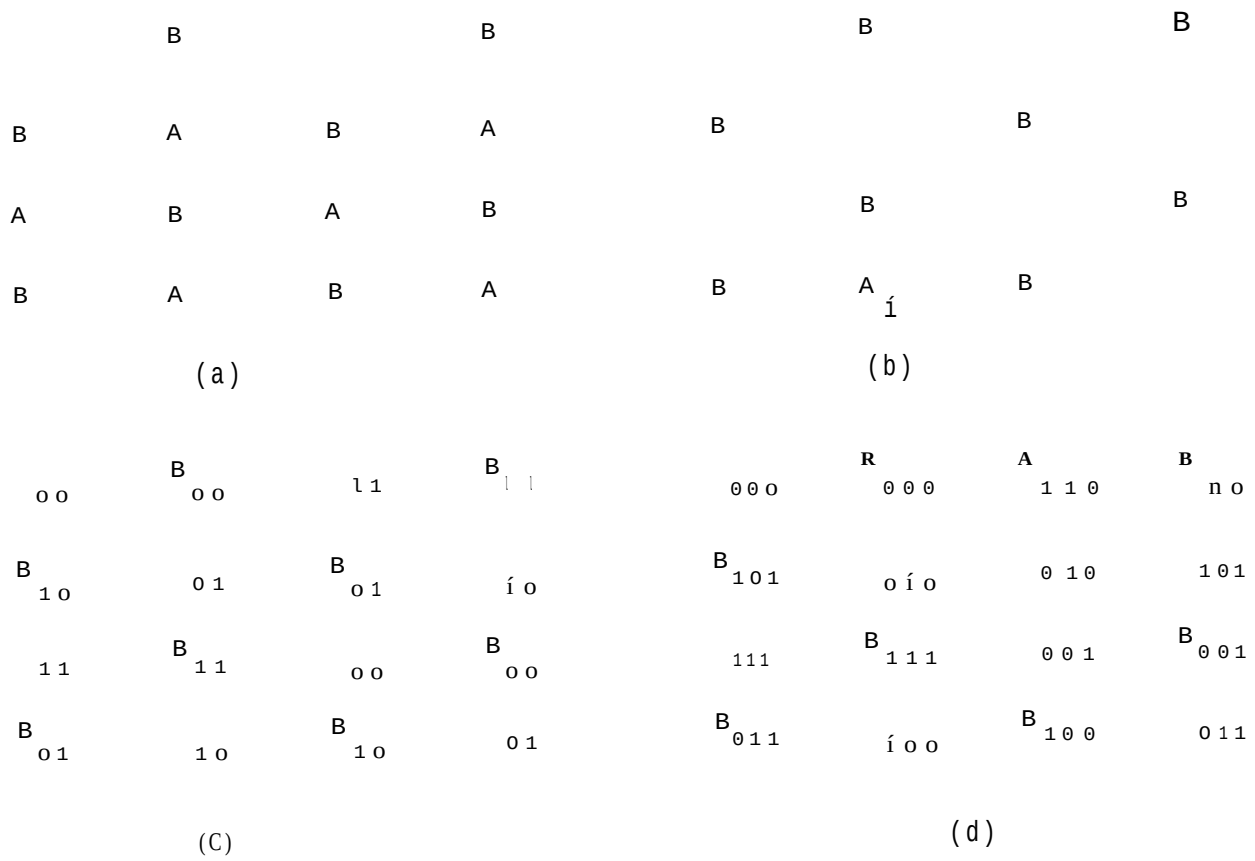


Figura (2:6) Constelação de 16 pontos particionada quatro vezes

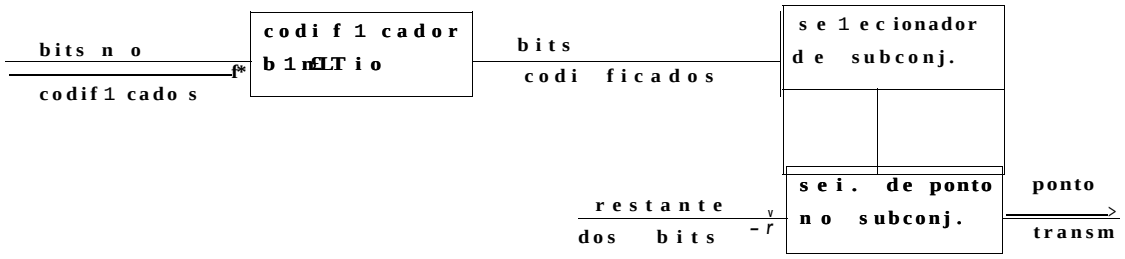


Figura. (2:7) Esquema de codificação

2:7:1) O RETÍCULO D_n

D_n contém todas as seqüências de n/2 pontos da constelação, com a condição de que 2k pontos, onde k = 0,1,2,...n/4, sejam do subconjunto B. Isto é razoável, sabendo-se que D deve conter metade dos pontos do retículo Z^n. Por exemplo D_4 inclui todas as seqüências de dois pontos bidimensionais nas quais os dois pontos são do mesmo subconjunto, i.e., seqüências da forma (A,A) ou (B,B).

2:7:2) O RETÍCULO E_8 .

E_8 consiste de todas as seqüências de quatro pontos bidimensionais nas quais todos os pontos são ou do subconjunto A ou do subconjunto B e ainda, nas quais os índices dos pontos (em b) formam uma palavra do código linear (4,3,2) de um digito de paridade.

2:7:3) O RETÍCULO A_{16} .

A_{16} consiste de todas as seqüências de oito pontos bidimensionais, nas quais todos os pontos são ou do subconjunto A ou do subconjunto B e ainda, nas quais os índices dos 8 pontos (em c) formam uma palavra do código (16,11,4), extensão do código de Hamming (15,11,3).

2:7:4) O RETÍCULO A_{24}.

Este retículo consiste de todas as seqüências de 12 pontos bidimensionais nas quais todos os pontos são ou do subconjunto A ou do subconjunto B, e os 24 (i , j) índices (em d) formam uma palavra do código de Golay estendido (24,12,8), e mais, os terceiros índices formam uma palavra do código (12,11,2) caso a seqüência seja do subconjunto A; e no caso da seqüência ser do subconjunto B, formam uma palavra do código (12,11,2) com paridade impar, i.e., £ k« 1 (mod 2).

2:^:5)-PARÂMETROS DOS RETÍCULOS CONSTRUÍDOS PELO MÉTODO FORNEY.

1) Cardinalidade.

Definimos a cardinalidade do retículo |A I como o número de seus pontos existentes em uma região IR. Como nós usamos uma constelação finita, podemos definir $|A_n|$ como o número dos pontos do retículo que podem ser obtidos usando-se todas as combinações possíveis de seus pontos. Esta cardinalidade é uma função do número dos pontos da constelação e do código usado para construir A_n.

Para o retículo não codificado Z_n, a cardinalidade é o número de todas as combinações de n/2 pontos da constelação,

portanto,

(2:20) $$|Z^n| = |S|^{n/2},$$

onde |S| é o número de pontos da constelação bidimensional.

O retículo D_n inclui metade dos pontos do retículo Z^n portanto sua cardinalidade é

(2:21) $$|D_J = |S|^{n/2}/2.$$

O retículo E_8 é construído pela divisão da constelação bidimensional em 4 subconjuntos, cada um tendo |S|/4, e pelo código (4,3,2). Considerando os dois tipos de seqüências A e B, o número total de pontos é

(2:22) $$|E_J = 2^3 \cdot (|S|/4)^4 * 2.$$

Da mesma maneira a cardinalidade do retículo A_{tr} e do retículo A_{24} são

(2:23) $$|A_{16}| = 2^{11} \cdot (|S|/8)^8 * 2.$$

e

(2:24) $$|A_{2}J = 2^{12} \cdot (|S|/8)^{12},$$

respectivamente.

2) Taxa de transmissão *R.*

A taxa de transmissão é definida como o número de bits de informações por dimensão que podemos transmitir usando um retículo A_n, sendo dada por [14]

(2:25) $$R(A_n) = -\log_2 |A^\wedge \setminus . \text{ bits/dim.}$$

Usando as relações anteriores sobre a cardinalidade e o fato de que na prática o número dos pontos da constelação é uma potência de 4, i.e., 4^m, onde m = 1,2,..., a taxa de transmissão dos retículos Z_n, D_4. E_8. A_{16} e A_{24} é

(2:26) $$K(Z^n) = -\wedge \cdot \log_2 |S|^{n/2} = m \text{ bits/dim,}$$

(2:27) $$K(D_4) = \wedge \cdot \frac{1}{2}\log_2(|S|^{n/2}/2) = m - \frac{1}{2} \text{ bits/dim,}$$

(2:28) $$K(E_8) = -\log_2(2^3 \cdot (|C|/4)^4 * 2) = m - \frac{1}{4} \text{ bits/dim,}$$

(2:29) $$K(A_{16}) = \frac{1}{16}\log_2(2^{11} \cdot (|S|/8)^8 * 2) = m - \frac{5}{16} \text{ bits/dim,}$$

(2:30) $$K(A_{24}) = \frac{1}{24}\log_2(2^{12} \cdot (|S|/8)^{12}) = m - 1 \text{ bits/dim.}$$

3) Potência média do retículo, $P(A_n)$.

Devido a simetria da constelação bidimensional, a potência média de um ponto do retículo é a mesma do Z_n **(veja** a prova no apêndice C), portanto

(2:31) $P(A_n) = n$,

onde p(S) é a potência média da constelação bidimensional S.

4) Distância mínima.

A distância mínima ao quadrado do retículo (A^) é a menor distância Euclidiana entre dois pontos neste retículo, a qual pode ser calculada da maneira seguinte:

a) O retículo D_4.

Como observamos A distância ao quadrado entre dois pontos A e B é no mínimo d_Q, conseqüentemente a distância mínima ao quadrado entre duas seqüências (A,A) e (B,B) é $2d^2$. Por outro lado, o quadrado da distância entre dois pontos do mesmo subconjunto é $2d^2$, dando a distância mínima ao quadrado entre duas seqüências de pontos do mesmo subconjunto igual a $2d^2$. Concluímos que,

<2:32> «ÍJV= 2 d 0 -

b) Os retículos E , A e A
8 16 24

Nestes retículos, os índices i ou (i , j) dos pontos que compõem a seqüência formam uma palavra dos códigos (4,3,2), (16,11,4) e (24,12,8) respectivamente. Usando a relação entre a distância de Hamming dos índices e a distância nos subconjuntos, $d^2 £ 2d_H d^2_Q$, duas seqüências do mesmo tipo mas, com índices i ou (i , j) diferentes, diferenciam-se, no mínimo, em $4d^2$, $8d^2$ e $16d^2$ respectivamente. Em ***A^,*** duas seqüências de pontos do mesmo subconjunto (A ou B) e do mesmo (i , j) índices, mas k índices diferentes, se diferenciam, no mínimo, por $8d^2$ em, pelo menos, dois símbolos [2] . Concluímos que

<2:33> d L < E 8 > = " o '

(2:34> d | ' A , 6 > = " o ' 6

<2:35> D ! . „ < A 2 J " 16 d o -
mIn 24 O

Estes valores podem ser confirmados se lembrarmos que para a

construção A dos retículos, i.e., A^= $2Z^n$+ C_Q, a distância mínima ao quadrado de A_n é

(2:36) $d^2_{min} = \min [4, d_h(C)] * d^2_u$

e, para a construção B, i.e., A^= $2Z^n$+ *2C^*+ C_Q,

(2:37) <„- minlie.Ad^CJ.d^C^] * d*,

[15].

5) Ganho assintótico de codificação.

O Ganho fundamental de codificação do retículo ?(A_n) é definido por dois parâmetros geométricos fundamentais: a distância mínima ao quadrado do retículo e o volume fundamental $V(A_n)$,

(2:38) $r(A_n) = d^(A_n)/V(A_n)^{2/"}$,

[14] e [15]. Em [22], o volume fundamental é dado em termos da redundância do retículo,

(2:39) $^V(^{A_n} > = 2^{r(An)} * d^2$,

onde para a construção A, onde A= $2Z^n$+ C(n,k),

(2:40) $\mathbf{r}(A) = n-k,$

e para a construção B, onde $A^{\wedge} = 2Z^{n} + 2C(n,k.) + C_{q}(n,j-k)$,

(2:41) $\mathbf{r}(A_n) = 2n-j$

[22], Escrevemos o ganho em termos da redundância,

(2:42) $$a_r(A) - 2^{-2r(A)/n} \; \min_{n} \; O$$

A redundância dos retículos D_4, E_8, A_{16} e A_{24} é 1, 4, 12 e 24 respectivamente. Usando estes valores e os valores da distância mínima ao quadrado, o ganho de codificação destes retículos, referido a uma constelação n-dimensional Z^{n}, é 1.5, 3.01, 4.5, 6.01 dB respectivamente, confirmando os valores em [2].

A tabela (2:3) mostra os parâmetros anteriores para os retículos mais densos nas dimensões 4, 8, 16, e 24 e o retículo v

2:7:6) CONSTRUÇÃO CÓDIGO DE RETÍCULOS OBTIDOS PELO MÉTODO DE FORNEY

Nos casos práticos, as coordenadas da constelação bidimensional usada para transmitir m bits/dimensão não codificados

têm, convencionalmente, um dos valores +±1, ±3, ±5,. . . . ±(2 -1) [2] A potência média de t a l constelação é

(2:43) P(C)= 2(4ʳᵃ-l)/3.

Tabela (2:3) Parâmetros dos retículos construídos usando-se uma constelação QAM quadrada de 4ᵐ pontos.

	taxa de transmisao blts/dim					distancia mínima ao quadrado	ganho de codifi-cação dB
m	1	2	3	4	5		
	1	2	3	4	5	2	0
D_4	0.75	1.75	2.75	3.75	4.75	4	1.5
E_8	0.5	1.5	2.5	3.5	4.5	8	3.01
A_{16}	0.25	1.25	2.25	3.25	4.25	16	4.5
A_{24}	0	1	2	3	4	32	6.01

Os pontos desta constelação (fig (2:8)) podem ser escritos sob a forma:

(2:44) p = 2Z²+ c,

onde c € C = {(1,1), (-1,-1), (-1,1), (!,-!)> « Z =*i*±1* ±3> ±S. . . . ±(2"-l)>. Podemos observar que os pontos A e os pontos B obtidos a partir da primeira partição têm a seguinte forma :

(2:45) $P_A = 2Z + c_1$, e

(2:46) $P_B = 2Z + c_2$,

onde c_1 e C_1= {(1,-1), (-1,1)}, e c e C = { (1,1), (-1,-1)}

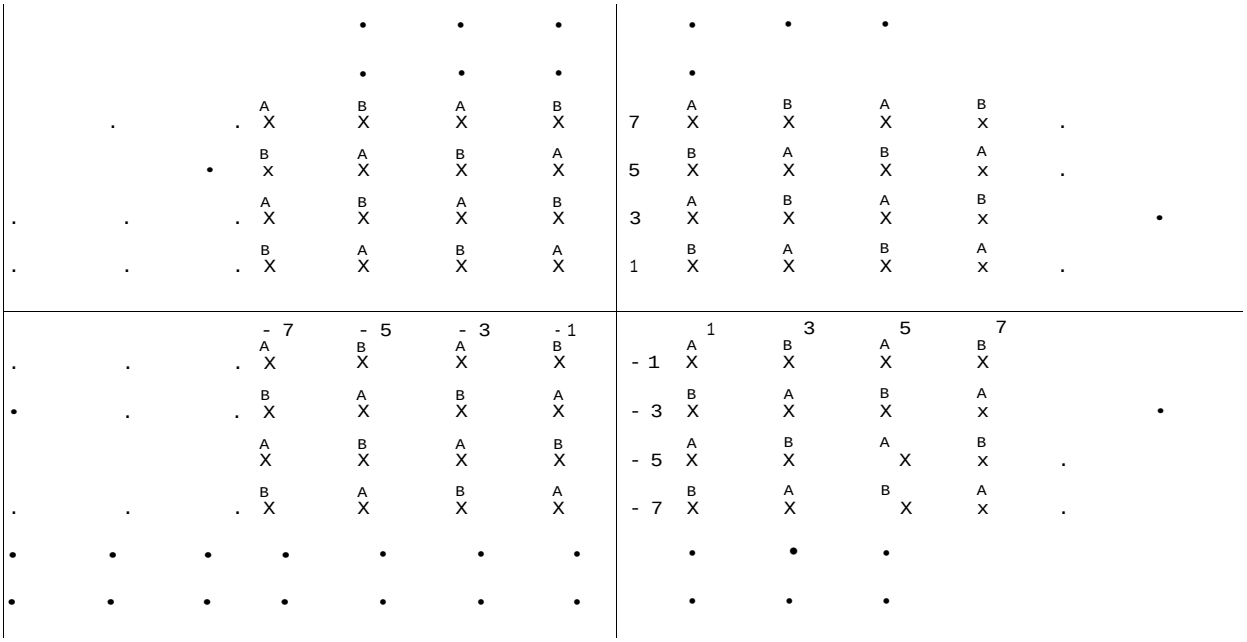


Figura (2:8). Partição de uma constelação em dois subconjuntos (A e B).

Portanto, C_j e formam dois códigos de um dígito de paridade (2,1,2), tendo as palavras de C paridade impar. Como o retíeul D_4 contém todas as seqüências (A,A) ou (B,B), temos, então, oito tipos de seqüências sob a forma :

(2:47) $D = 2Z + c$,

onde c_4 e código C_4, o qual tem as seqüências binárias apresentadas na tabela (2:4). Estas seqüências formam o código (4,3,2), confirmando a construção código.

Concluímos que os retículos D_4, E_g, A_{jfi} e A_{24} construídos usando-se a constelação bidimensional podem ser obtidos aplicando-se a construção código, substituindo 1 por -1 e 0 por 1 nas palavras dos códigos binários.

No próximo capítulo vamos estudar os métodos de codificação e decodificação e desenvolver um novo método útil para este tipo de construção (construção por partição de conjuntos).

Tabela(2:4). Palavras do código C_4 (equação (2:47)).

(A,A)	(B,B)
(1,-1,1,-1)	(1,1,-1,-1)
(-1,1,-1,1)	(-1,-1,1,1)
(-1,1,1,-1)	*(1, 1,1,1)*
(1,-1,-1,1)	(-1,-1,-1,-1)

r CAPITULO 3

INTRODUÇÃO AOS MÉTODOS DE CODIFICAÇÃO E DECODIFICAÇÃO PARA RETÍCULOS

Nos canais ruidosos.e limitados em banda passante, os retículos são usados como métodos de modulação codificada para transmitir seqüências binárias com a finalidade de aumentar a taxa de transmissão. Nestes canais, os pontos transmitidos serão perturbados por um vetor de ruído aleatório multidimensional, o qual deslocará os pontos ao espaço R^n. Um sistema de codificação e decodificação pode ser visto como um conjunto de algoritmos que realiza três funções:

1) a função de codificação (mapeamento) £ (x)

Esta função mapeia uma seqüência binária da fonte b = $(b_1, \ldots, b_k)$ em um único ponto y = $(y_1, \ldots, y_n)$ do retículo A_n, i.e.,

(3:1) $y = Ç(b)$,

2) a função de decodificação $<_p(x)$.

Como o ruído desloca o ponto transmitido para um ponto do espaço lR^n, esta função encontra o ponto $y = (y^*...,y)$ mais próximo do retículo a um ponto $x = (x....,X)$ do espaço R^n, ou seja,

(3:3) $y = <_p(x)$.

3) a função de mapeamento invertida (demapeamento)

$X(x)$

Esta função mapeia um ponto $y = tY§»»»iY)$ ã° retículo A_n na seqüência binária $b = (b_1,.....,b_k)$ correspondente,

(3:2) $b = A(y)$.

Esta função deve ser o inverso da função de mapeamento,

$A(y) = ç"^1(x)$.

O método trivial para a primeira e a terceira função é construir uma tabela de correspondência entre os pontos do retículo e as seqüências binárias *(look up table)*. Cada vez que se precisa transmitir uma seqüência binária, uma busca na tabela

fornece o ponto correspondente. O contrário será feito no caso de se demapear um ponto à seqüência binária adequada. Para a segunda função, a distância Euclidiana entre cada ponto no retículo e o ponto x é calculada, e escolhemos como ponto decodificado aquele que minimiza esta distância.

Tais métodos são úteis nos casos em que o número de pontos utilizados do retículo é pequeno. Caso contrário, o sistema de transmissão torna-se lento e requer muita memória para armazenar a tabela de correspondência. Portanto, a busca por bons métodos de codificação e decodificação é uma busca para diminuir a sua complexidade (aumentar sua velocidade), diminuir o volume da memória, e facilitar a sua implementação, levando-se em conta a finalidade do uso do retículo.

Mencionamos no capítulo 2 que a solução para os três problemas referidos, depende do método de construção do retículo. Uma boa construção é aquela que facilita a solução dos três problemas conjuntamente. Existem métodos gerais que consideram os retículos como uma questão matemática. Tais métodos dedicam-se em resolver a função de decodificação sem nenhuma preocupação com as demais, enquanto nos sistemas que utilizam os retículos como métodos de modulação codificada, a preocupação do projetista é dar uma solução global para as três funções e procurar uma construção eficiente para esta finalidade.

Vimos anteriormente, que existem duas estratégias de utilização dos retículos em modulação codificada: a primeira

considera-os como códigos de treliça, e a outra considera-os como códigos de bloco. Mencionamos que este trabalho é dedicado ao segundo método, portanto vamos estudar e desenvolver, se for possível, algoritmos de codificação e decodificação.

3:1) MÉTODOS GERAIS DE DECODIFICAÇAO

Muitos trabalhos tem sido realizados após o artigo pioneiro de J.H. Conway e N.J. Sloane [16] os quais propuseram alqoritmos de decodificação dos retículos D_n, E_6, E_7, E_8 e seus duais. A construção destes algoritmos depende da definição primitiva destes retículos, por exemplo,

(3:4) $$D_n = \{ (x_1, \ldots, x_j, x_r \in Z, [x \text{ é par}\},$$

(3:5) $$E_8 = \{ D_8 \cup (1/2, \ldots, 1/2) + D_8 \}.$$

Tais algoritmos podem ser utilizados no processo de quantização, pois os autores não apresentaram um método de codificação no intuito de utilizar estes retículos como códigos.

Dependendo da construção código dos retículos, J.H. Conway e N.J. Sloane [20] apresentaram um algoritmo de decodificação dos retículos obtidos pela construção A, onde A é um retículo desta construção se e somente se:

(3:6) $A_n = 2Z^n + C_O(n,k,d)$.

Este algoritmo é baseado na divisão do vetor recebido em duas partes: a primeira corresponde à parte não codificada $2Z^n$, e a outra à parte codificada $C_Q(n,k,d)$. A decodificação é realizada da maneira seguinte:

Algoritmo (3:1)

1) dado X = **(x ,. . . . .,x$_n$)** e R^n , primeiro reduzimos todos **x**$_i$ ao domínio -1 ***** ^{x}j$^{<3}$ P^{ela} subtração de um vetor 4Z^n.
2) suponha que S denota o conjunto dos índices i para os quais 1 s **x<** 3. Para i e S, troque ***x^*** por 2 ~^{x}j •
3) como **x** agora está no cubo -1 **x<** 1 (i • 1,...,!_), aplique o decodificador de C a X, obtendo um vetor Y= (y$_1$.• • • •$_n$y 1, y$_i$.= ±l.
4) Para cada i € ***S,*** troque por 2 - y^. Portanto Y + 4Z^n é o ponto de A_D mais próximo ao vetor original X.

Para retículos obtidos através da aplicação da construção B, onde A_n é um retículo se e somente se $A_n = 2Z^n + C_1$, o problema é mais complexo, pois existem dois códigos

dependentes que devem ser tratados. Considerando A^= **2Z**ⁿ**+** C* C_0 como união de k classes laterais do retículo Q_n **= 2Z**ⁿ**+** (n,n-l)+ 0 **[15]** e **[21],** Forney **[22]** colocou um algoritmo para decodificá-lo i seguinte maneira:

Algoritmo **(3:2)**

1) dado X = (x_1,. . . . . ,x_{-n}) e IRⁿ, aplique o algoritmo anterior para decodificar x ao ponto $\mathbf{X}_q$ mais próximo no retículo A^= **2Z**ⁿ **+ C**$_q$**;**

2) confira a paridade dos n componentes **2Z** em $\mathbf{X}_q$, se for par, aceite $\mathbf{X}_q$ como ponto em fi^= **2Z**ⁿ**+** (n,n-l). Se não for, então mude a coordenada x_{qi} de $\mathbf{X}_q$ por **±2** de tal maneira que a mudança aumente a distância ao quadrado $(x_{Oi} - \hat{x}_i)^2$ o mínimo possível;

3) para todas as classes laterais j = **1** até k**-1,** aplique as etapas **1** e **2** aos vetores X⁺ ª j ' obtendo k**-1** pontos decodificados X^.

4) entre todos X_0, X_1 X_{k-1}, escolha o mais próximo ao vetor X como o ponto decodificado.

Claro que esta decodificação não é por máxima verossimilhança, mas tem o mesmo expoente de erro. Tal decodificação foi chamada pelo autor como "decodificação da

distância cotada". Nós vamos usar este princípio para propor um algoritmo especifico para retículos utilizados nos sistemas de modulação codificada.

3:2) MÉTODOS ESPECIAIS DE CODIFICAÇÃO E DECODIFICAÇÃO

Como os retículos podem ser usados como códigos de bloco para modulação codificada, existem algoritmos especiais para codificação e decodificação. 0 método principal de se utilizar os retículos como códigos de bloco é o método de Forney, o qual é baseado no conceito de Ungerboeck. *"partição de conjuntos".* Forney *et al*. [2] apresentou em linhas gerais como seria a codificação e decodificação dos retículos D_t,, E_8" A_{16}" e A_{24} construídos usando-se seu método. Nós vamos discutir estes algoritmos com a finalidade de utilizá-los em comparações futuras.

3:2:1) O RETÍCULO D_4.

D_4 contém todas as seqüências de dois pontos de uma constelação de 2^{2m} pontos divididos em dois grupos, A e B, com a condição de que os dois pontos pertençam ao mesmo grupo (veja capitulo 2). Nesta constelação dividida, cada ponto precisa de 2m dígitos binários para ser identificado, um deles identifica o grupo ao qual o ponto pertence, A ou B, e os últimos 2m-l identificam um ponto particular neste grupo. Isto implica em uma tabela de

correspondência entre os dígitos binários e os pontos de cada grupo tal como a tabela (3:1).

Em cada ponto é capaz de transmitir 4m-l dígitos binários. Portanto, e como mostra a figura (3:1), uma seqüência binária b = (b_1,. . . ., b_{4m-l}) identifica um ponto em D_4 da maneira seguinte:

Algoritmo (3:3)

1) o dígito b^ escolhe entre uma seqüência de tipo (A,A) ou de tipo (B,B). Portanto este dígito será codificado pelo código C = {(1,1),(0,0)} em (b,b); e
2) dividimos os 4m dígitos obtidos após (1) em duas seqüências binárias,

   *>.= ($b_{í}$ -_____ e
   *> = (b_2 ..... ,b_{4m}).

   Cada uma destas seqüências seleciona um ponto na constelação. Claro que os dois pontos pertencem ao mesmo grupo.

Para demapear um ponto de em uma seqüência binária, fazemos o contrário. Cada par de coordenadas do ponto identifica sua representação binária usando a tabela de correspondência. Associamos um dígito no início da seqüência ao

tipo do ponto, (A,A) ou (B,B).

Tabela (3:1). Tabela de correspondência entre sequências e pontos de uma constelação de 16 pontos divididos em dois grupos A/B. Por exemplo, a sequência binária (1000) corresponde ao ponto (-3,3), ponto A.

sequencia bi-narla de 3 bits	pontos do grupo A (1)	pontos do grupo B (0)
000	-3, 3	-1, 3
001	1, 3	3, 3
010	-1, 1	-3, 1
011	3, 1	1, 1
100	-3,-1	-1,-1
101	1,-1	3,-1
110	-1,-3	"3,-3
111	3,-3	1,-3

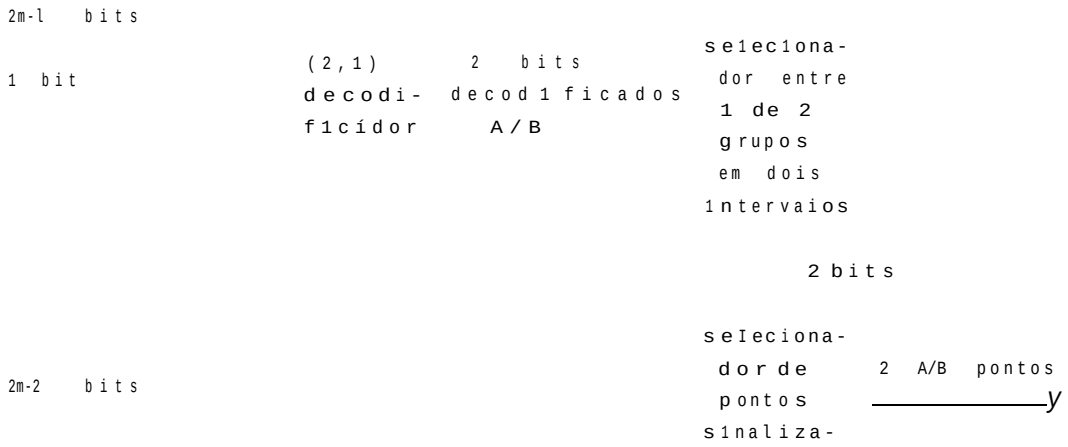


Figura (3:1). Codificador de D_4.

Decodificar um vetor $X = (x, x, x, x) \in RE$ em um ponto mais próximo em D_4 pode ser realizado da seguinte maneira:

Algoritmo (3:4)

1) divida o vetor X em dois vetores bidimensionais $X = (x, x)$ e X
   $1 \quad 1 \quad l \quad r \quad 2' \quad 2$
2) encontre o ponto do grupo A e o ponto do grupo B mais próximos de cada um destes vetores da constelação bidimensional, formando duas seqüências de pontos (A,A) e (B,B); e
3) A seqüência mais próxima a X é o ponto decodificado.

3:2:2) O retículo E_8.

Para construir o retículo E_8 dividimos os pontos da constelação bidimensional em dois grupos, **A** e **B,** em seguida dividimos cada grupo em dois subgrupos, $\mathbf{A}_q$, $A^{\wedge}$, $\mathbf{B}_q$, e B_j. Cada um destes subgrupos tem 2^{2m-2} pontos, portanto precisa-se de 2m dígitos binários para a identificação de cada ponto , um deles indica o tipo de grupo ao qual este ponto pertence, A ou **B,** um outro refere-se ao índice deste ponto , e os demais ?.m-2 dígitos escolhem um ponto no grupo, como mostra a tabela (3:2).

O retículo E_g contém todas as seqüências que têm pontos bidimensionais do mesmo grupo, A ou **B,** e cujos índices formam uma palavra do código (4,3,2). Portanto, como mostra a figura (3:2), uma seqüência binária b = (b_1,...., $^{\wedge}_{8m4}$) seleciona um ponto em E_g da maneira seguinte:

Algoritmo (3:5)

1) o primeiro dígito b_1 seleciona entre uma seqüência de pontos do grupo **A** ou do grupo **B,** Ou seja, este dígito será codificado pelo código (4,1,4) em (b_1, b_2, b_3, b_4);
2) os três dígitos (b_2, b_3, b_4) serão codificados pelo código (4, 3, 2) em (b_2, b_3, b_4, $k = b_2 + b_3 + b_4$).

3) dividimos os 8m dígitos que possuímos em 4 seqüências binárias de 2m dígitos cada,

$$b_1 = (b_1, b_2, \ldots, b_{2m}),$$
$$b_2 = (b_{2m+1}, \ldots, b_{4m})$$
$$b_3 = (b_{4m+1}, \ldots, b_{6m})$$
$$b_4 = (b_{6m+1}, \ldots, b_{8m}),$$

4) cada uma destas seqüências identifica um ponto na constelação usando-se a tabela de correspondência, . onde o primeiro dígito identifica o tipo do ponto e o segundo identifica o índice.

Tabela (3:2). Tabela de correspondência entre sequências e pontos de um constelação de 16 pontos divididos em quatro grupos A_0, , B_0 e B_1. Por exemplo, a sequência binária (0110) corresponde ao ponto (1,3), ponto A_0.

sequencia b1-nár1a de 3 bits	pontos do grupo A (1) |		pontos do gru• po B (0)	
	0 (0)	Al (1)	BO (0)	BI (1)
00	-3, 3	-1, 1	-1, 3	-3, 1
01	1, 3	3, 1	3, 3	1, 1
10	-3,-1	-1,-3	-1,-1	-3,-3
11	1,-1	3,-3	3,-1	1,-3

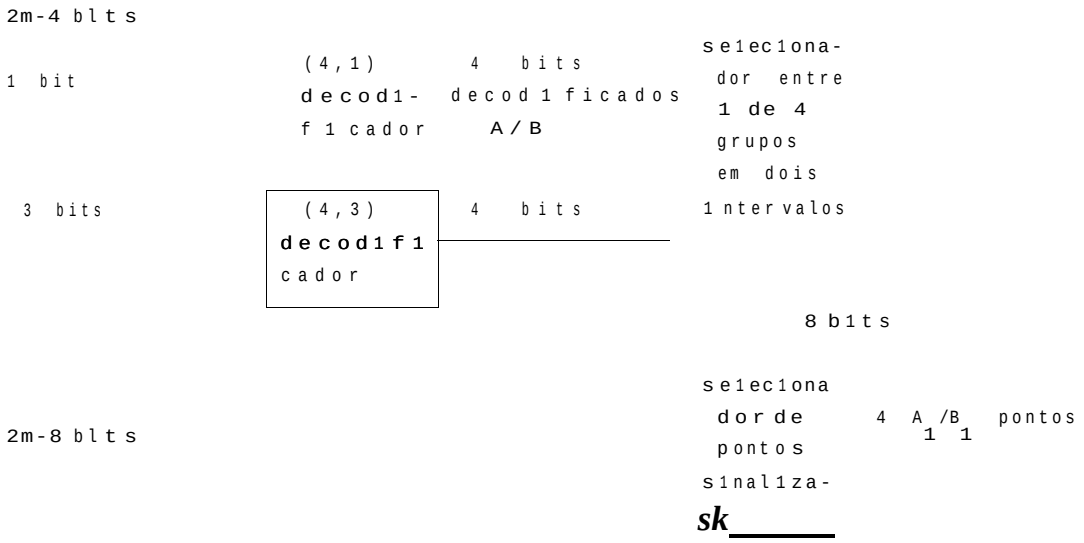


Figura (3:2). Codificador de E_8.

Para demapear um ponto de E em uma seqüência binária, cada par do ponto seleciona sua representação binária usando-se a tabela de correspondência. Da seqüência binária obtida, eliminamos os dígitos de redundância dos códigos (4,1,4) e (4,3,2).

Para decodificar um vetor $X = (x_1, \ldots, x_8)$ em um ponto no retículo, deve-se realizar o processo mostrado a seguir:

Algoritmo (3:6)

1) divida o vetor X em quatro vetores bidimensionais $X = (x_1, x_2)$, $X = (x_3, x_4)$, $x = (x_5, x_6)$ e $x = (x_7, x_8)$;

2) encontre os pontos dos grupos A e B mais

próximos de cada um destes vetores, formando duas seqüências de pontos, (A,A,A,A) e (B,B,B,B);

3) para cada seqüência, confira se os índices dos pontos formam uma palavra do código (4,3,2). Se não o formam, troque o ponto menos confiável para o ponto mais próximo do mesmo grupo;

4) a seqüência mais próxima a X é o ponto decodificado.

.

Métodos similares e mais complexos podem ser aplicados para codificar e decodificar o retículo A , dependendo dos códigos (8,1,8) e (16,11,4), e o retículo *A^,* dependendo dos códigos (12,1,12), (12,11,2), e (24,12,8). O processo de decodificação nestes dois casos utiliza métodos de decodificação suave explorando as características dos códigos (16,11) e (24,12). Este processo, ao contrário daquele do retículo E , torna-se complexo para sua implementação. Tal complexidade foi simplificada para o retículo *A^* por Be'ery et *al* [23], que propuseramum um algoritmo rápido para decodificar o código C24, 12) .

Devido a importância do retículo 24-dimensional de Leech , muitos algoritmos foram propostos para decodificá-lo, tais como: o de J.H. Conway e N.J. Sloane [20]. baseado no fato que *A^* contém como sub-retículo; o de Forney [22], o qual considerou A_{24} como duas classes laterais de H_{24}'; o de Gorden R. Lang e Frederick M. Longstaff [11], os quais propuseram um sistema

completo de codificação e decodificação deste retículo, baseado na construção código apresentada por Forney [21]. Este último, foi adotado em um MODEM 19.200 bits/seg da MOTOROLA [11].

No próximo capítulo, desenvolveremos um algoritmo de codificação e decodificação baseado na construção de Forney e utilizando a construção código dos retículos.

Devido a importância dos códigos binários na decodificação *ò\os* retículos, apresentamos neste texto métodos de decodificação suave de alguns tipos, de códigos. Tais métodos podem ser encontrados, também, em [20]. Notamos que a "decodificação" nestes algoritmos significa "encontrar a palavra código mais próxima do vetor recebido no sentido Euclidiano".

3:3) MÉTODOS DE DECODIFICAÇÃO SUAVE PARA CÓDIGOS BINÁRIOS.

Suponha que tf é um código binário (n,k). É conveniente escrever as palavras códigos como vetores de +l's e -l's. Observe-se que, se

(3:7) W = U + V em notação de 0 e 1,

então

(3:8) W = U * V em notação de +1 e -1.

A notação de +1 e -1 permite trocar o cálculo da distância pelo

cálculo da produto escalar. Se x e R^n e u € tf, então

(3:9) $dist^2(U,X) = (U-X)\ (U-X)$

$= X\ \ x - 2X.U + U\ \ u$

$= X\ \ X - 2X.U + n.$

Portanto, encontrar a palavra código mais próxima é equivalente a encontrar a palavra código (em notação +1 e -1) a qual tem o maior produto escalar com x.

Os métodos de decodificação seguintes usam este fato para encontrar a palavra código mais próxima de um vetor x.

3:3:1) COMPARAÇÃO DIRETA

Para qualquer código (n,k), calculamos o produto escalar do vetor recebido com cada palavra do código e escolhemos como palavra decodificada aquela que tem o maior produto escalar. Tal método é longo e útil somente para códigos de dimensão pequena.

3:3:2) CÓDIGOS DE REED - MULLER DE 1* ORDEM

Um código R-M, com parâmetros (2°, k = m+1), pode ser decodificado usando-se a transformada rápida de Hadamard (a qual é chamada *"Green machine")* [24].

3:3:3) CÓDIGO UNIVERSAL

É o código que contém todos os vetores binários de comprimento n. Para codificar $X = (x_n, \ldots, x_1)$, trocamos x_i por $sgn(x_i)$, onde

(3:10)
$$sgn(x) = +1, \quad \text{se } x \geq 0,$$
$$= -1, \quad \text{se } x < 0.$$

Existem códigos ou subcódigos que podem ser tratados como código universal. Especificamente, qualquer código (n,k) gerado por um matriz da forma

(3:11)
$$\begin{matrix} a & b & \ldots & b & b \\ b & a & \ldots & b & b \\ \vdots & \vdots & & \vdots & \vdots \\ b & b & \ldots & a & b \\ b & b & \ldots & b & a \end{matrix}$$

onde $a = 1^{n/k}$ (n/k vezes), e $b = 0^{(n/k)}$ (n/k vezes), pode ser decodificado da seguinte maneira . Dado $X = (x_n, \ldots, x_1)$ decodificámo-lo como

(3:12)
$$U = u_1^{(n/k)} u_m^{(n/k)} \ldots u_i^{(n/k)}$$

onde $u^{(n/k)}$ significa repetição de u n/k vezes, e

(3:13) $U = \mathrm{sgn}\,(x_{1} + x_{(l-l)n/k+l} + x_{(l-l)n/k+2} + \ldots + x_{in/k})$.

Ademais o produto escalar máximo é

(3:14) $U.X = \sum_{i=1}^{k} (x_{(l-l)n/k+l} + x_{(l-l)n/k+2} + \ldots + x_{in/k})I$.

EXEMPLO

Considere o código (8,4) gerado pela matriz

11	00	00	00
00	11	00	00
00	00	11	00
00	00	00	11

a qual pode ser representada pela matriz;

1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

onde cada dígito nesta matriz representa dois dígitos da matriz original. Tal código é um código universal. Dado um vetor X = (x ,. . . . .,x), decodificamo-locomo

U = a a b b c c d d ,

onde

a= sgn (x ^ x_2),

b= sgn (x_3+ x j ,

c= sgn(x_5+x_6),

d= sgn(x_7+x_8).

3:3:4) CÓDIGOS DE UM DÍGITO DE PARIDADE (n, k = n-1)

Tais códigos contêm todas as seqüências binárias de comprimento n e de peso par (número de l's é par). Para decodificar X = (x . ,x_n), trocamos x por sgn(x_i). Se o peso for par, paramos. Se for ímpar, trocamos o sinal de $x_{(}$ de menor magnitude.

Como no caso do código universal, existem códigos ou subcódigos que podem ser tratados como códigos de um dígito de paridade. Qualquer código (n, k = n-1) gerado por um matriz da forma,

(3:15)

$$\begin{matrix} b & b & b \\ a & b & b \\ \cdot & \cdot & \cdot \\ b & a & b \\ b & b & a \end{matrix}$$

onde a = $l^{n/k}$ (n/k vezes), e b = $o^{(n/k)}$ (n/k vezes) pode ser decodificado da seguinte maneira; dado X = (x_1 ,. . . . . ,x_n), decodificámo-lo como

(3:16) $U = (u_1^{(n/k)}, u_2^{(n/k)}, \ldots, u_k^{(n/k)})$

onde $u_j^{(n/k)}$ significa repetição de u_j n/k vezes, e

(3:17) $u_j = \operatorname{sgn}(s_j)$,

onde

(3:18) $s = (x_1 + x_{(1+1)n/k} + \ldots + x_{(k-1)n/k+2} + x_{in/k})$.

Caso o número de u 's negativos for ímpar, trocamos o sinal de para o qual $|s_j|$ seja mínimo, digamos u_i. O produto escalar máxima

(3=19) $$O.X = \sum_{i=1} |(x_{(i-1)n/k+1} + x_{(i-1)n/k+2} + \ldots + x_{in/k})|$$

se o número de U *B negativos for par, caso contrário teremos

(3=2°) $$U . X = \sum_{i=1} |(x_{(i-1)n/k+1} + \ldots + x_{in/k})| - 2|s_j|.$$

3:3:5) SUPERCÓDIGOS

Se *a* é um dos códigos anteriores, e tf contém *a* como subcódigo de pequeno índice, então tf pode ser decodificado facilmente. Escrevendo tf em termos de *a*

(3:21) $$tf = \mathbf{UJ}_{=}\dot{O}(\mathbf{P}^{(J)+} \ll),$$

onde t := |tf|/|a| é o índice de a em tf, e $p^{(j)}$ (j = 0,....,t-l) são os representantes das classes laterais de *a* em tf. tf é chamado supercódigo de a. A equação (3:21) é aplicada aos vetores com notação **0** e **1,** enquanto para vetores de notação **+1** e **-1,** temos

(3:22) $$tf = \mathbf{U}_{j=0}^{t-i}\mathbf{P}^{(j)} * \ll \bullet$$

Encontrar o produto escalar máximo de x com os vetores em $p^{ij)}$* *a.* é equivalente a encontrar o produto escalar máxima de $p^{(j)}$* x com os vetores em *a.* Então podemos decodificar x como a seguir.

calcule

(3:23) $$y"> := p'^{j)} * x,$$

e use o decodificador de *a* para encontrar $u^{(j)}$, o qual é o vetor mais próximo em a a $y^{(j>}$. Calcule também o produto escalar

(3:24) $ip^{(j)} = u^{(j)}.y^{(j)}$'

Depois de fazer isso para j = 0,1,....t-1, o máximo $ip^{(j)}$ é o produto escalar final, e o $u^{(j)}$ correspondente será usado para produzir a saída do decodificador, $p^{(j)}* u^{(j)}$.

r CAPÍTULO 4

NOVOS ALGORITMOS DE CODIFICAÇÃO E DECODIFICAÇAO PARA RETÍCULOS

4:1) INTRODUÇÃO.

Na decodificação por máxima verossimilhança, compara-se o ponto recebido com todos os pontos do retículo[1], usando-se a distância Euclidiana como métrica. A complexidade deste processo foi diminuída por algoritmos mais simples, como aqueles que mencionamos no capítulo anterior. A maioria destes algoritmos trata os retículos ilimitados (sentido estrito) no espaço infinito, enquanto os demais têm complexidade relacionada com o número de pontos da constelação bidimensional, usada para construir estes

[1]NOTA- Daqui por diante a palavra retículo faz referência a um "retículo limitado", i.e., uma constelação multidimensional extraída de pontos do retículo contidos em uma região limitada. Por simplicidade, consideramos o método de FORNEY [2] para construir os retículos (c.f. capítulo 2). Nesta construção, o ponto (0,0,0,..,0) é deslocado ao ponto (1,1,1,...,!).

retículos. Ademais, nenhum destes algoritmos dedica-se a reduzir a complexidade do problema de mapeamento e demapeamento entre as seqüências binárias e os pontos do retículo. Pretendemos desenvolver um algoritmo que trate os dois problemas simultaneamente.

Voltando a idéia primária da decodificação por máxima verossimilhança (decodificação exaustiva), a comparação é feita com todos os pontos do retículo, de modo que uma questão conveniente aparece: "por que não dividir o retículo em dois grupos com número de pontos idênticos e comparar o ponto recebido com o centro de massa de cada grupo (vamos denotá-lo como o centro de gravidade) ? ". Com esta ação, eliminamos metade dos pontos do retículo no processo de comparação. Continuaremos o mesmo processo com os pontos restantes, eliminando metade deles, e assim por diante, até chegarmos a um ponto que clamamos provavelmente ser o ponto mais próximo do vetor recebido.

As etapas envolvidas neste processo são as seguintes:

1 - encontrar o centro de gravidade do retículo;

2 - passar um hiperplano $L = a_1 x_1 + a_2 x_2 + \ldots + a_n x_n$ através deste centro, o qual divide o retículo em dois grupos idênticos. Para simplificar (explorando a simetria da constelação que usamos na construção do retículo), este hiperplano pode ser paralelo a qualquer eixo do espaço R^n, i.e. $L =$

x_i, i=l,2,...,n; o que garante que nenhum ponto do retículo cairá nele[2]. A escolha destes planos tem a vantagem de que a localização do ponto pode ser realizada através da comparação de **sinais** e valores das coordenadas dos pontos recebidos;

3 - em seguida encontramos o centro de gravidade de cada grupo, e escolhemos o grupo para o qual seu centro é o mais próximo ao vetor recebido. Da mesma maneira, este grupo escolhido será dividido em dois subgrupos menores ; e

4 - continuar o processo até chegar no final a um grupo que contenha um único ponto, o qual será, provavelmente, o ponto permitido mais próximo ao vetor recebido.

Realizamos este processo para dividir um subespaço do retículo D4 e decodificar um vetor com ruído. A tabela no apêndice A. mostra esta divisão, o centro de cada grupo e os subgrupos após a divisão.

EXEMPLO 4:1

Queremos decodificar o vetor com ruído R = (1.25,-2.13,3.1,2.8)

SOLUÇÃO:

Partindo do centro (0 0 0 0),a decodificação segue o seguinte caminho:

2 ! Rigorosamente, não tratamos com retículos, mas com uma classe lateral (coset) do retículo.

(0 0 0 0)	(2 0 0 0)	(2 2 0 0)	(2-2 2 0)
	(-2 0 0 0)	(2-2 0 0)	(2-2-20)
	(2-2 2 2)	(1-2 2 2)	(1-1 2 2)
	(2-2 2-2)	(3-2 2 2)	(1-3 2 2)
	- (1-3 1 1)	(1-3 3 3)	
	- (1-3 3 3)		

Acontece, porém, que em alguns casos, uma coordenada do vetor recebido cai em uma região crítica ao redor do plano da divisão, com a possibilidade que este ponto esteja mais próximo de um ponto do subgrupo eliminado do que todos os pontos do grupo escolhido. Isto é uma conseqüência da redundância do retículo. Observe o seguinte exemplo:

EXEMPLO 4:2

Considere o retículo D_4 e o vetor recebido R = (-0.1,-3.2,3.2,2.8). Este vetor será decodificado em Dl = (-1,-3,3,1), com distância Euclidiana igual a 2,03 de R, enquanto a distância Euclidiana entre o vetor recebido e o ponto (1-3 3 3) é igual 1,33, o qual deveria ser o ponto decodificado.

Tal problema pode ser resolvido explorando a construção código do retículo.

Os centros de cada subgrupo podem ser calculados computacionalmente e armazenados em uma memória para serem usados na decodificação dos vetores recebidos. Mas esta armazenagem consome muita memória e o processo de decodificação torna-se lento para os retículos com número muito grande de pontos, o que implica em um número bastante grande de centros a serem usados na decodificação. Aproveitando a simetria do retículo podemos encontrar os centros diretamente a partir do vetor recebido. Para simplicidade, vamos considerar uma região limitada do retículo não codificado Z^n, onde I = ±1, ±3,. . . . ., ±(2^n-l). E como os retículos codificados são obtidos atravéz da partição dos pontos da constelação I^2 em subgrupos idênticos, portanto os mesmos resultados podem ser generalizados para incluir tais retículos.

LEMA (4:1)- O centro de gravidade do retículo Z^n construído usando uma constelação bidimensional de 2^{2n+1} pontos com coordenadas sob a forma; {±1, ±3, ±5,. . . . . ., ±(2^n-l)} é o ponto (0,0,0,0,0,. . . . . ., 0).

PROVA: Como os pontos do retículo Z^n são todas as combinações das coordenadas Z, portanto o centro de Z^n é

$$\bar{c} = \langle c', \ldots, c' J \rangle$$

onde c*, i = *1,2...*,n, é o centro de Z. É fácil verificar que

$$c_i = \frac{-1 + 1 - 3 + 3 + \ldots - (2^r - 1) + (2^r - 1)}{2^n} \qquad (4:1)$$

$$c_i = 0.$$

Portanto, o centro de gravidade do retículo Z^n construído através de uma constelação bidimensional desta forma é o ponto (0,0,0,0 . . . . . 0).

Q.E.D.

Após encontrar o centro de gravidade do retículo devemos passar um hiperplano que o divide em dois subgrupos idênticos e, como notamos, este hiperplano pode ser um dos eixos do espaço IR^n. Para simplificar, podemos escolher estes eixos em ordem e ciclicamente. Por exemplo, em um espaço n-dimensional, $(x_1, x_2, \ldots, x_n)$, começamos com o hiperplano $L = x_1$ e em seguida $L = x_2$ até $L =$ e depois voltamos a $L = x_1$, e assim por diante.

DEFINIÇÃO 4:!- Vamos denotar as n primeiras etapas da divisão como a primeira fase, e a n+l-ésima até a 2n-ésima etapa como a segunda fase, e em geral denotamos a kn-ésima até a

(k+1)n-ésima etapa da divisão como a k+l-ésima fase.

Como o centro do retículo é (0,0,0,0....,0), é suficiente examinar o sinal da primeira coordenada dos pontos do retículo para localizá-los no seu grupo correto na primeira etapa da primeira fase da divisão.

Após dividir o retículo em dois grupos idênticos, encontraremos o centro de cada um e-medimos a distância Euclidiana entre o vetor recebido e cada centro. Em seguida, escolhemos o grupo do centro mais próximo para ser submetido à próxima etapa e, como indicamos, é suficiente examinar o sinal da primeira coordenada do ponto para localizá-lo no grupo do centro mais próximo. É fácil observar que os dois centros são da forma $(C_1, 0,0,0...,0)$ e $(-c_1,0,0,0....,0)$, por causa da simetria da constelação.

Na segunda etapa da primeira fase, devemos deslocar o novo centro escolhido , digamos $(c_1,0,0,...,0)$, à origem (0,0,0...,0), e em seguida deslocar o vetor recebido e todos os pontos do grupo, mantendo as distâncias entre eles, ou seja, o ponto $(x_1,x_2,x_3,...,x_n)$ será deslocado ao ponto $(x_1 - c_1, x_2, x_3, ..., x_n)$. Como este deslocamento não afeta os eixos x_2 até x_n, podemos então continuar com a divisão e a comparação, sem qualquer preocupação de realizar tal deslocamento, usando estes eixos como planos de divisão até o fim da primeira fase.

Mas qual é a forma das coordenadas dos centros nesta fase? Como a divisão nesta fase é relacionada com o sinal das coordenadas dos pontos, então, ela representa a divisão das coordenadas da constelação, o que implica serem as coordenadas do centro $[c_1, c_2, c_3, \ldots, c_n, 0, 0, \ldots, 0)$ da forma:

$$c = \pm \frac{1+3+5+\ldots+2^m-1}{2^{m-1}}$$

Como:

$$1+3+5+\ldots+(2n-3)+(2n-1)=n^2,$$

fazemos

$$2^m - 1 = 2n - 1$$

$$n = 2$$

Então:

$$c_i = \pm \frac{2^{(m-1)}}{i_2 - i} = \pm 2^{m-1}$$

É fácil concluir que os centros no final da primeira fase são da forma: $(\pm 2^{m-1}, \pm 2^{m-1}, \ldots, \pm 2^{m-1})$.

Na segunda fase da divisão e da decodificação, o ponto recebido e todos os pontos do grupo escolhido, como o grupo cujo centro é o mais próximo do vetor recebido, devem ser deslocados em relação ao novo centro $(c_1, c_2, \ldots, c_n)$, $c = \pm 2^m$, ou seja, o ponto $(y_1, y_2, y_3 \ldots, y_n)$ deve ser deslocado ao ponto $(y_1 - c_1, y_2 - c_2 \ldots y_n - c_n)$. Em seguida, passamos um hiperplano no