

UNIVERSIDADE ESTADUAL DE CAMPINAS  
FACULDADE DE ENGENHARIA ELÉTRICA  
DEPARTAMENTO DE COMUNICAÇÕES

*Este exemplar corresponde à redação final da Tese defendida por Celso de Almeida e aprovada pela Comissão Julgadora em 11 de maio de 1990.*

*Reginaldo Palazzo Júnior*

# MODULAÇÃO-CODIFICADA GENERALIZADA VIA EQUAÇÃO DE DIOFANTO

Autor: CELSO DE ALMEIDA *ntem*  
Orientador: PROF DR. REGINALDO PALAZZO JR. *tem Júnior*

Tese apresentada à Faculdade de Engenharia Elétrica da Universidade Estadual de Campinas - Unicamp, como parte dos requisitos exigidos para a obtenção do título de DOUTOR EM ENGENHARIA ELÉTRICA.

MAIO - 1990

## RESUMO

O estudo da teoria da modulação-codificada limitara-se até então ao caso binário e a alguns casos ternários e quaternários esporádicos. Neste trabalho, é conseguida uma generalização completa desta teoria para qualquer número de elementos do corpo: para as constelações uni e bi-dimensionais, PAM e QAM. Códigos ótimos são obtidos e seus desempenhos calculados.

## ABSTRACT

The study of trellis code modulation schemes was limited until now to the binary case and to some sporadic ternary and quaternary cases. In this work a complete generalization of this theory is achieved for any number of elements of a field, for constellations with one and two dimensions: PAM and QAM. Optimum codes are determined and their corresponding performances are evaluated.

## AGRADECIMENTOS

Agradeço em especial, ao Prof. Dr. Palazzo pela oportunidade e pelas valiosas sugestões e discussões.

Agradeço também, por terem aceito o convite de participar da Banca Examinadora os Profs. Drs. Hélio Waldman e Dalton Soares Arantes do DECOM/FEE UNICAMP, e aos Profs. Drs. Weiler A. Finamore do CETUC/PUC-RJ e Waldemar Cardoso da Rocha Jr. da UFPE.

# Índice

<b>1</b>	<b>INTRODUÇÃO</b>	<b>8</b>
1.1	HISTÓRICO . . . . .	8
1.2	RESUMO DO TRABALHO . . . . .	8
<b>2</b>	<b>FUNDAMENTOS DA MODULAÇÃO-CODIFICADA BINÁRIA</b>	<b>10</b>
2.1	INTRODUÇÃO . . . . .	10
2.2	CODIFICADORES CONVOLUCIONAIS . . . . .	11
2.3	PROPRIEDADES DOS CÓDIGOS CONVOLUCIONAIS . . . . .	14
2.4	REVISÃO DE RETICULADOS . . . . .	15
2.4.1	Definições . . . . .	15
2.4.2	Propriedades Geométricas . . . . .	16
2.5	MODULAÇÕES DE AMPLITUDE DE PULSO: PAM . . . . .	18
2.5.1	Introdução . . . . .	18
2.5.2	Codificador com um elemento de memória . . . . .	20
2.6	MODULAÇÕES DE AMPLITUDE EM QUADRATURA DE FASE: QAM . . . . .	23
2.6.1	Introdução . . . . .	23
2.6.2	Codificador com um elemento de memória . . . . .	24
<b>3</b>	<b>GENERALIZAÇÃO DE CÓDIGOS PARA MODULAÇÕES DO TIPO PAM</b>	<b>28</b>
3.1	INTRODUÇÃO . . . . .	28
3.2	PARTIÇÃO DE CONSTELAÇÕES UNI-DIMENSIONAIS . . . . .	28
3.3	CÓDIGOS ÓTIMOS PARA O RETICULADO UNI-DIMENSIONAL $Z$ . . . . .	29
3.3.1	Códigos de treliça lineares com um elemento de memória . . . . .	29
3.3.2	Códigos de treliça lineares com duas memórias . . . . .	40
3.3.3	Códigos de treliça lineares com tres memórias . . . . .	45
3.3.4	Códigos de treliça lineares com quatro memórias . . . . .	45
3.3.5	Códigos de treliça lineares com cinco memórias . . . . .	46
3.4	CAPACIDADE DE CANAL . . . . .	46

<b>4</b>	<b>SOLUÇÃO DA PARTIÇÃO DE CONJUNTOS: FORMAS QUADRÁTICAS BINÁRIAS</b>	<b>51</b>
4.1	INTRODUÇÃO	51
4.2	MODULAÇÕES QAM: RETICULADO $Z^2$	52
4.3	MODULAÇÕES QAM: RETICULADO $A_2$	61
4.4	MODULAÇÕES PAM	67
4.5	APLICAÇÃO EM TELEFONIA CELULAR	67
<b>5</b>	<b>GENERALIZAÇÃO DA MODULAÇÃO-CODIFICADA QAM: RETICULADOS <math>Z^2</math> E <math>A_2</math></b>	<b>69</b>
5.1	INTRODUÇÃO	69
5.2	CÓDIGOS ÓTIMOS PARA O RETICULADO $Z^2$	70
5.2.1	Introdução	70
5.2.2	Códigos de treliça lineares com um elemento de memória	70
5.2.3	Códigos de treliça lineares com duas memórias	79
5.3	CAPACIDADE DE CANAL PARA CONSTELAÇÕES QUE USAM O RETICULADO $Z^2$	83
5.4	CÓDIGOS ÓTIMOS PARA O RETICULADO $A_2$	86
5.4.1	Códigos de treliça lineares com um elemento de memória	86
5.5	CAPACIDADE DE CANAL PARA CONSTELAÇÕES QUE USAM O RETICULADO $A_2$	92
<b>6</b>	<b>COMENTÁRIOS FINAIS E CONCLUSÕES</b>	<b>97</b>
6.1	PARTIÇÕES DE CONSTELAÇÕES MULTI-DIMENSIONAIS	97
6.2	PROBLEMAS DEIXADOS EM ABERTO	98
6.3	CONCLUSÕES	99
<b>A</b>		<b>100</b>
A.1	REGRA GENERALIZADA DOS MINTERMOS	100
<b>B</b>		<b>103</b>
B.1	PROGRAMAS PRINCIPAIS	103
B.1.1	Programa para cálculo da distância livre	103
B.1.2	Programa que determina os melhores movimentos no reticulado $Z^2$	106
B.1.3	Programa para cálculo da capacidade de canal	107

# Lista das Figuras

2.1	Esquema da modulação-codificada. . . . .	11
2.2	Codificador convolucional (3, 2, 1). . . . .	13
2.3	Treliça correspondente ao exemplo do codificador (3, 2, 1) . . . . .	14
2.4	Reticulado $Z^2$ . . . . .	16
2.5	Reticulado $A_2$ . . . . .	17
2.6	Reticulados quaternários com mesma distância mínima, mas com multiplicidade diferentes. . . . .	18
2.7	Esquema de um sistema convencional de transmissão e de uma constelação 4-PAM. . . . .	19
2.8	Partição de conjuntos e constelação 8-PAM. . . . .	20
2.9	Esquema do codificador com 1 memória e treliça com os elementos da constelação alocados aos ramos. O tri-bit alocado a cada ramo representa, respectivamente, os elementos $y_i^2$ , $y_i^1$ e $y_i^0$ . . . . .	21
2.10	Estrutura do codificador de 1 memória e mapeador 8-PAM. . . . .	22
2.11	Esquema de um sistema convencional de transmissão e de uma constelação 4-QAM. . . . .	23
2.12	Partição de conjuntos e constelação 8-QAM. . . . .	25
2.13	Esquema do codificador com 1 memória e treliça com os elementos da constelação alocados aos ramos correspondente à constelação 8-QAM. . . . .	25
2.14	Estrutura do codificador de 1 memória e mapeador 8-QAM. . . . .	26
3.1	Esquema da modulação-codificada generalizada. . . . .	29
3.2	Esquema da partição de conjuntos generalizada para constelações PAM. . . . .	30
3.3	Código convolucional geral com 1 elem. de memória. . . . .	30
3.4	Codificador com duas entradas e constelação de pontos. . . . .	31
3.5	Partição, constelação e codificador binários $k = 1$ . . . . .	32
3.6	Treliça ótima do codificador binário com uma memória. . . . .	33
3.7	Codificador ótimo binário. . . . .	34
3.8	Partição de conjuntos e constelação de pontos para o caso ternário. . . . .	35
3.9	Treliça ótima do codificador ternário de uma memória. . . . .	36
3.10	Código ternário ótimo. . . . .	37
3.11	Exemplo de código não-linear para $Q = 6$ . reticulado $Z$ . . . . .	40
3.12	Código convolucional geral com duas memórias. . . . .	41

3.13	Treliça e código binário ótimo com 2 memórias. . . . .	42
3.14	Treliça ótima do codificador ternário de duas memórias. . . . .	43
3.15	Codificador ternário ótimo. . . . .	44
3.16	Estrutura do codificador binário com 3 elementos de memória. . . . .	45
3.17	Capacidade de canal para constelações binárias . . . . .	48
3.18	Capacidade de canal para constelações ternárias . . . . .	48
3.19	Capacidade de canal para constelações quaternárias . . . . .	49
4.1	Partição correspondente à solução da equação quadrática. . . . .	54
4.2	Célula Básica para $Q = 5$ . . . . .	55
4.3	Exemplo de partição de uma constelação de 64 pontos com $Q = 8$ elementos. . . . .	58
4.4	Exemplo dos 2 níveis de partição de uma constelação de 64 pontos com $Q = 4$ elementos. . . . .	59
4.5	Exemplo dos 5 níveis de partição de uma constelação de 64 pontos com $Q = 2$ elementos. . . . .	60
4.6	Exemplo das 3 partições de uma constelação de 1296 pontos com $Q = 6$ elementos. . . . .	60
4.7	Partição correspondente à solução da equação quadrática. . . . .	62
4.8	Célula Básica Inclinada para $Q = 7$ no reticulado $A_2$ . . . . .	62
4.9	Exemplo de 2 níveis de partição de uma constelação de 64 pontos com $Q = 4$ elementos. . . . .	63
4.10	Exemplo dos 3 níveis de partição de constelação de 1296 pontos com $Q = 6$ . . . . .	66
4.11	Partição de uma constelação PAM genérica . . . . .	67
4.12	Exemplo de alocação de frequências em telefonia celular. . . . .	68
5.1	Partição para o caso binário. . . . .	71
5.2	Constelação de sinais para o caso binário. . . . .	71
5.3	Treliça ótima do codificador binário com uma memória. . . . .	72
5.4	Codificador binário ótimo. . . . .	73
5.5	Partição e constelação de pontos para o caso ternário. . . . .	74
5.6	Treliça ótima do codificador ternário de uma memória. . . . .	75
5.7	Código ternário ótimo. . . . .	76
5.8	Exemplo de código não-linear para $Q = 4$ , reticulado $Z^2$ . . . . .	77
5.9	Treliça ótima do codificador binário de duas memórias. . . . .	79
5.10	Codificador binário ótimo com 2 memórias. . . . .	80
5.11	Constelação e treliça ternária ótima para codificador com duas memórias. . . . .	81
5.12	Codificador ternário ótimo com duas memórias. . . . .	82
5.13	Capacidade de canal para constelações binárias, $Z^2$ . . . . .	83
5.14	Capacidade de canal para constelações ternárias, $Z^2$ . . . . .	84
5.15	Capacidade de canal para constelações quaternárias, $Z^2$ . . . . .	84
5.16	Partição de conjuntos e constelação ternária. . . . .	87
5.17	Treliça ternária ótima do codificador de uma memória, $A_2$ . . . . .	87
5.18	Codificador ternário ótimo. . . . .	88

5.19	Partição, e constelação para o caso quaternário. . . . .	90
5.20	Treliça quaternária para codificador de uma memória. . . . .	90
5.21	Código quaternário ótimo. . . . .	91
5.22	Capacidade de canal para constelações binárias, $A_2$ . . . . .	94
5.23	Capacidade de canal para constelações ternárias, $A_2$ . . . . .	94
5.24	Capacidade de canal para constelações quaternárias, $A_2$ . . . . .	95

# Lista das Tabelas

3.1	A distância livre, o ganho e a matriz geradora dos códigos para uma memória. . . . .	38
3.2	A distância livre, o ganho e a matriz geradora dos códigos de duas memórias. . . . .	44
3.3	A distância livre, o ganho e a matriz geradora dos códigos de 3 memórias. . . . .	45
3.4	A distância livre, o ganho e a matriz geradora dos códigos ótimos de 4 memórias. . . . .	46
3.5	A distância livre, o ganho e a matriz geradora dos códigos ótimos de 5 memórias. . . . .	46
3.6	Comportamento do ganho assintótico máximo para o reticulado $Z$ . . . . .	49
4.1	Movimentos que apresentam maior distância mínima para o reticulado $Z^2$ . . . . .	57
4.2	Movimentos que apresentam maior distância mínima para o reticulado $A_2$ . . . . .	65
5.1	A distância livre, o ganho, a matriz de transformação ortogonal e a matriz geradora de um código ótimo com uma memória para o reticulado $Z^2$ . . . . .	78
5.2	A distância livre, o ganho, a matriz de transformação ortogonal e a matriz geradora de um código ótimo com duas memórias para o reticulado $Z^2$ . . . . .	82
5.3	Comportamento do ganho máximo para constelações QAM, reticulado $Z^2$ . . . . .	85
5.4	A distância livre, o ganho, a matriz de transformação ortogonal e a matriz geradora de um código ótimo com uma memória para o reticulado $A_2$ . . . . .	93
5.5	Comportamento do ganho assintótico máximo para o reticulado $A_2$ . . . . .	95
6.1	Evolução da $d_{min}^2$ e do número de pontos para partições no reticulado $Z^2$ para $Q = 2$ . . . . .	97
6.2	Evolução da $d_{min}^2$ e do número de pontos para partições no reticulado $Z^4$ para $Q = 2$ . . . . .	98
A.1	Tabela verdade para o caso binário. . . . .	101

A.2 Tabela verdade para o caso ternário. . . . .	101
--	-----

# Capítulo 1

## INTRODUÇÃO

### 1.1 HISTÓRICO

Até o final da década de 70, a codificação era usada independentemente da modulação, quando então começaram a surgir os primeiros artigos, os quais mostravam que se poderia obter ganhos maiores de desempenho, caso se concatenasse o projeto da codificação ao da modulação.

Somente em 1982 é que Ungerboeck [1] propôs um esquema de projeto da codificação em combinação com a modulação para o caso binário, esquema esse que é baseado no mapeamento por partição de conjuntos, conceito este explicado adiante.

Este trabalho generaliza o trabalho realizado por Ungerboeck no esquema de partição de conjuntos para corpos  $Q$  *ários*. Em suma, o esquema de partição proposto por Ungerboeck passa a ser um caso particular deste esquema mais geral.

### 1.2 RESUMO DO TRABALHO

O capítulo 2 oferece uma rápida revisão sobre códigos convolucionais, reticulados e sobre os princípios básicos da modulação-codificada binária para constelações PAM e QAM utilizando o reticulado  $Z^2$ .

O capítulo 3 trata da generalização da modulação-codificada para constelações uni-dimensionais (PAM), quando se obtém os ganhos em relação a esquemas convencionais de transmissão para corpos  $Q$  *ários*.

O capítulo 4 propõe modos de partição baseado na teoria dos números, mais especificamente, nas formas quadráticas binárias. Dado um reticulado e uma certa constelação inscrita neste reticulado, é fornecido o particionamento de conjuntos ótimo que garante um máximo na menor distância entre elementos de um mesmo sub-conjunto.

O capítulo 5 trata da generalização da modulação-codificada para constelações bi-dimensionais (QAM) nos reticulados  $Z^2$  e  $A_2$ , onde se obtém os ganhos em relação a esquemas convencionais de transmissão para corpos  $Q$  *ários*.

No capítulo 6 são apresentadas as conclusões finais, além de breves comentários sobre problemas deixados em aberto para pesquisa posterior.

Os principais programas usados para o desenvolvimento da tese são fornecidos no apêndice, tais como os programas para obtenção dos códigos ótimos usando modulações PAM e QAM, além do programa que fornece o melhor particionamento do reticulado  $Z^2$  para um corpo  $Q$  *arbitrário*.

Como contribuição, gostaríamos de enfatizar, que o capítulo 4 é original, assim como todos códigos não-binários obtidos nos capítulos 3 e 5.

## Capítulo 2

# FUNDAMENTOS DA MODULAÇÃO-CODIFICADA BINÁRIA

### 2.1 INTRODUÇÃO

Ungerboeck [1] em seu célebre artigo de 1982, mostrou que é possível construir sistemas de comunicações com ganhos adicionais de até 6 dB em relação ruído (SNR), em relação a esquemas convencionais de transmissão, sem uso de bandas de passagem maiores, ou sacrifício da taxa de informação, utilizando-se de um esquema novo de codificação e modulação conjugados, que a partir de agora denominaremos de modulação-codificada.

Este novo esquema de modulação-codificada representado na figura 2.1 é composto de um codificador de taxa  $\frac{k}{k+1}$  responsável pela expansão da constelação de  $2^k$  para  $2^{k+1}$  elementos para o caso binário, e de um mapeador que se utiliza de um processo que a seguir denominaremos como mapeamento por partição de conjuntos. Este mapeamento tem como objetivo básico prover um aumento na distância Euclidiana entre sinais a serem transmitidos através do canal.

Nossa atenção neste trabalho será voltada exclusivamente a códigos do tipo convolucional, devido a seu tratamento algébrico e sua implementação mais simples, mesmo sabendo que é possível contruir códigos de bloco com mesmo desempenho.

O decodificador envolvido no processo de decodificação será sempre de máxima verossimilhança (maximum likelihood), e utilizar-se-á de decodificação do tipo suave (soft decision decoding). Como se sabe de [1] [2] [3], o uso de decodificação suave ao invés da decodificação abrupta (hard decision decoding) proporciona ganhos em SNR de até 2 dB, em canais com ruído Gaussiano aditivo.

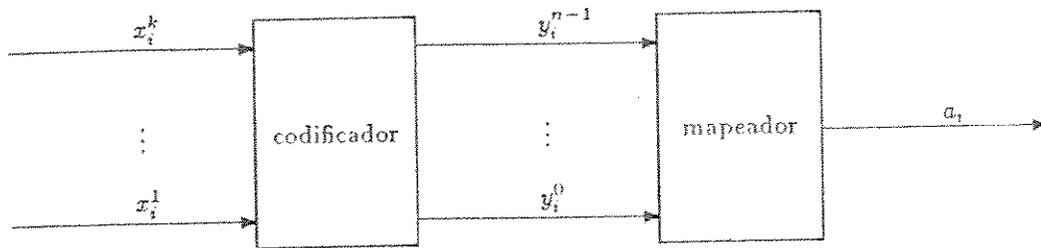


Figura 2.1: Esquema da modulação-codificada.

## 2.2 CODIFICADORES CONVOLUCIONAIS

Um codificador convolucional  $(n, k, m)$  pode ser implementado como um circuito sequencial linear de  $k$  entradas,  $n$  saídas, e  $m$  memórias, onde  $n > k$ . A razão  $R = k/n$  denomina-se taxa do código. As  $n$  saídas, em um dado instante de tempo, não dependem somente das  $k$  entradas neste mesmo instante, como também nos  $m$  instantes anteriores. À medida que se aumenta  $m$ , a imunidade em relação ao ruído também aumenta, diminuindo a taxa de erro.

A sequência de saída  $y$  em um código convolucional se relaciona à sequência  $x$  de entrada através de:

$$y = xG \quad (2.1)$$

onde  $G$  é denominada matriz geradora do código, e as sequências de saída e entrada são definidas como

$$y = [y_0^0, y_0^1, \dots, y_0^{n-1}, y_1^0, \dots, y_1^{n-1}, \dots] \text{ e } x = [x_0^1, x_0^2, \dots, x_0^k, x_1^1, \dots, x_1^k, \dots]$$

onde  $y_i^j$  representa a sequência na  $j$ ésima saída no  $i$ ésimo instante de tempo, onde  $1 \leq j \leq n$ . Esse codificador recebe esse nome, porque realiza a convolução entre a sequência de entrada e a sequência correspondente à matriz geradora do código, equivalentemente, com o conteúdo das  $m$  memórias.

Para um codificador genérico  $(n, k, m)$ , a matriz geradora semi-infinita cor-

respondente é dada por:

$$G = \begin{vmatrix} G_0 & G_1 & G_2 & \cdots & G_m & & \\ & G_0 & G_1 & \cdots & G_{m-1} & G_m & \\ & & G_0 & \cdots & G_{m-2} & G_{m-1} & G_m \\ & & & \vdots & \cdots & & \vdots \end{vmatrix} \quad (2.2)$$

onde cada  $G_i$  é uma sub-matriz de  $k \times n$  elementos representada por:

$$G_i = \begin{vmatrix} g_{1i}^1 & g_{1i}^2 & \cdots & g_{1i}^n \\ g_{2i}^1 & g_{2i}^2 & \cdots & g_{2i}^n \\ \vdots & \vdots & \cdots & \vdots \\ g_{ki}^1 & g_{ki}^2 & \cdots & g_{ki}^n \end{vmatrix} \quad (2.3)$$

onde o termo genérico  $g_{li}^j$  está associado à  $l$ ésima entrada, à  $j$ ésima saída, e ao  $i$ ésimo instante de tempo, onde  $1 \leq l \leq k$  e  $1 \leq j \leq n$ .

Seja  $m_j$  o número de memórias associada à entrada  $x_j$ . Define-se  $m$  como sendo a ordem de memória do codificador :

$$m = \max(m_j) \quad 1 \leq j \leq k \quad (2.4)$$

O número total de memórias utilizadas por esse codificador pode ser escrito como:

$$m_t = \sum_{j=1}^k m_j \quad (2.5)$$

onde o número de estados associados a um codificador binário é  $2^{m_t}$ .

Como cada bit na entrada do codificador pode durar até um máximo de  $m+1$  intervalos de tempo, e em cada intervalo de tempo pode afetar até  $n$  saídas, define-se o comprimento de restrição (constraint length) como:

$$n_A = n(m+1) \quad (2.6)$$

Uma das medidas de desempenho mais importante de um código convolucional é a sua distância livre, definida como a menor distância entre sequências  $y'$  e  $y''$ , sendo uma de referência e a outra correspondente à saída do canal tal que iniciem em um mesmo estado  $s_1$  e retornem a um mesmo estado  $s_2$ , onde  $s_2$  não precisa ser igual a  $s_1$ .

Matematicamente, isto é traduzido por:

$$d_{free} = \min \left[ \sum_n d(y'_n, y''_n) \right] \quad x'_n \neq x''_n \quad (2.7)$$

Quando do uso da codificação binária desvinculada da modulação, isto é, em canais binários simétricos, a distância usada é a de Hamming, enquanto que no caso da modulação-codificada, onde o canal é do tipo aditivo e gaussiano, a

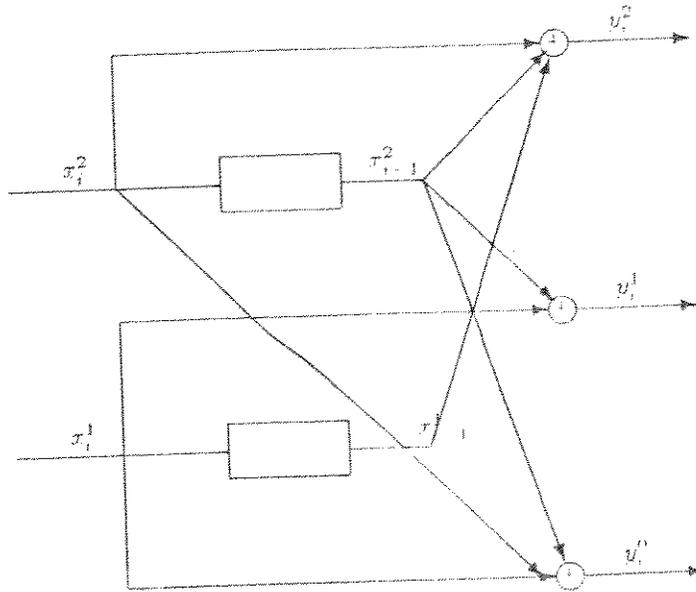


Figura 2.2: Codificador convolucional (3, 2, 1).

distância usada é a Euclideana, que nada mais é que a distância geométrica entre pontos de uma certa constelação. Neste caso:

$$d_{free}^2 = \min \left[ \sum_{j=1}^N d_E^2(a_j^1, a_j^2) \mid x_j^1 \neq x_j^2 \right] \quad (2.8)$$

onde  $N$  representa o número de transições na treliça, para que se obtenha a distância livre quadrática do código, e  $a_j$  são símbolos associados ao canal.

A evolução dos estados acompanha a seguinte lei de formação:

$$\theta = (s_i^1, s_{i-1}^1, \dots, s_{i-m_1+1}^1, s_i^2, s_{i-1}^2, \dots) \quad (2.9)$$

com  $s_i^j$  representando o estado da  $i$ ésima memória correspondente à  $j$ ésima entrada.

**Exemplo 2.1** Seja o codificador convolucional (3, 2, 1) da figura 2.2. Por inspeção tiramos que:

$$G_0 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

$$G_1 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

Calcula-se facilmente que a taxa do código  $R = 2/3$ , o comprimento de restrição  $n_A = 6$ , e que o número total de memórias do código  $m_1 = 2$ . A treliça que se encontra na figura 2.3, foi gerada a partir do codificador da figura 2.2. A distância livre (Hamming) deste código vale  $d_{free} = 2 + 1 = 3$  (2 para se sair de

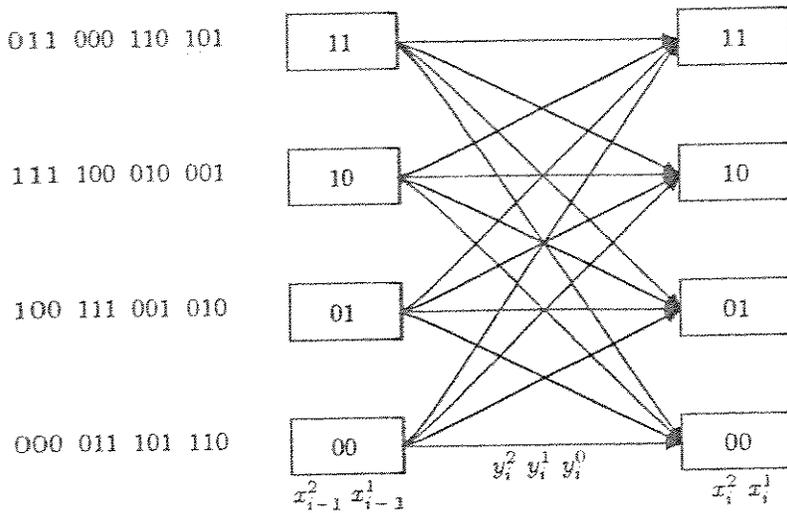


Figura 2.3: Treliça correspondente ao exemplo do codificador (3, 2, 1)

qualquer estado, e 1 para se voltar ao mesmo estado), onde a distância usada foi a de Hamming.

Para maiores detalhes na estrutura de codificadores convolucionais, ou a respeito das diversas técnicas de decodificação, ver [3].

## 2.3 PROPRIEDADES DOS CÓDIGOS CONVOLUCIONAIS

**Definição 2.1 (Linearidade)** Um código convolucional que tem a função de transferência  $y = f(x)$  é dito linear [4] se e somente se existir uma operação denotada soma, entre os símbolos do canal ( $a$ ), sob a qual o alfabeto do canal forma um grupo Abelian e  $f$  faz um mapeamento um a um entre os elementos  $y$  e  $a$ .

**Definição 2.2 (Superlinearidade)** Todo código convolucional é dito superlinear [4] se é possível associar com cada elemento  $a$  transmitido através do canal, um número real  $w(a)$  denominado peso do símbolo, tal que:

$$d_E^2(a_i, a_j) = w(a_i + a_j) \quad (2.10)$$

**Teorema 2.1** Um código linear  $f$  é dito superlinear se e somente se, a seguinte condição for satisfeita:

$$a_i + a_j = a_k + a_l \quad \Leftrightarrow \quad d(a_i, a_j) = d(a_k, a_l) \quad (2.11)$$

**Teorema 2.2** Códigos superlineares apresentam a propriedade da uniformidade de erro (UEP), ou seja, o conjunto das distâncias de toda sequência do codificador é

o mesmo, independentemente da sequência de referência usada.

*Prova:* Seja  $(a_i)$  uma sequência de referência e  $(a_j)$  uma outra sequência genérica. Seja  $(a_k) = (a_i) + (a_j)$ . Usando as equações 2.10 e 2.11 temos que:

$$\begin{aligned} d &= d_E^2(a_i, a_j) \\ &= w(a_i + a_j) = w(a_k) \\ &= w(0 + a_k) \\ &= d_E^2(0, a_k) \end{aligned}$$

CQD

A prova do teorema 2.2 nos mostra que dada uma sequência de referência, todas as outras sequências que distam  $d$  dela, podem ser colocadas em correspondência uma a uma com uma terceira sequência que tem a mesma distância  $d$  da sequência nula. Portanto, a sequência de base pode ser qualquer uma, inclusive a sequência nula, quando estivermos percorrendo pela treliça para a determinação da distância livre. Além disso, o teorema 2.2 é necessário, mas não suficiente, para que códigos superlineares apresentem UEP.

## 2.4 REVISÃO DE RETICULADOS

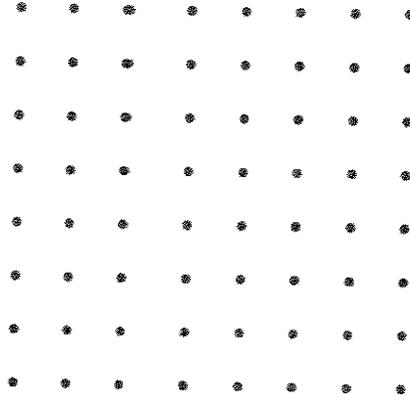
### 2.4.1 Definições

Um reticulado (lattice) real [8]  $\Lambda$  é definido como um conjunto de pontos (ou vetores) em um espaço real Euclídeano  $N$ -dimensional, que forma um grupo com o uso da adição vetorial, ou seja, a soma ou diferença de quaisquer dois pontos de  $\Lambda$  pertence a  $\Lambda$ . Pelo fato de ser um grupo,  $\Lambda$  obviamente inclui o elemento nulo, assim como para cada elemento  $\lambda$  existe o seu inverso  $-\lambda$ . Também pelo fato de ser um grupo, o estudo de reticulados, induz ao estudo de sub-reticulados (sublattices), assim como ao estudo de partições.

Os reticulados podem ser obtidos através das seguintes propriedades:

1. Escalonamento: Se  $r$  é um número real, então  $r\Lambda$  é o reticulado que consiste da multiplicação de todo vetor  $\lambda$  de  $\Lambda$  por  $r$ .
2. Transformação Ortogonal: Se  $T$  é uma matriz de transformação ortogonal para o espaço  $N$ -dimensional, então  $T\Lambda$  é um novo reticulado consistindo dos produtos  $T\lambda$  de todos os vetores  $\lambda$  de  $\Lambda$ .
3. Produto Cartesiano: O produto cartesiano de  $\Lambda$  consigo próprio  $M$  vezes, fornece um reticulado  $MN$ -dimensional denotado por  $\Lambda^M$ , que é o conjunto das  $MN$ -uplas  $(\lambda_1, \lambda_2, \dots, \lambda_M)$ , onde cada  $\lambda_j$  pertence a  $\Lambda$ .

**Exemplo 2.2**  $Z$  representa o reticulado uni-dimensional consistindo somente de inteiros, enquanto que  $Z^2$  obtido pelo produto cartesiano de  $Z$  por  $Z$ , representa o conjunto de pontos do plano, conforme a figura 2.4.

Figura 2.4: Reticulado  $Z^2$ .

**Exemplo 2.3** Como exemplo de transformação ortogonal, suponha a seguinte:

$$T = \begin{bmatrix} 1 & \frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} \end{bmatrix} \quad (2.12)$$

Esta transformação produz o reticulado da figura 2.5 à partir do reticulado  $Z^2$ , que é conhecido como  $A_2$  e que será objeto de estudo neste trabalho.

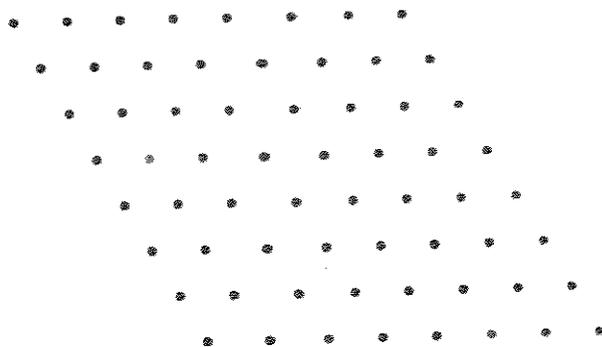
### 2.4.2 Propriedades Geométricas

Uma das propriedades mais importantes de reticulados é a distância mínima ( $d_{min}$ ) entre pontos. Outra propriedade importante é o volume do reticulado. Forney [8] define o ganho fundamental de codificação como:

$$\gamma(\Lambda) = \frac{d_{min}^2}{V(\Lambda)^{2/N}} \quad (2.13)$$

onde  $N$  é o número de dimensões do reticulado, e  $V(\Lambda)^{2/N}$  tem dimensão de área.

Quanto maior  $\gamma$ , maior é a possibilidade de que para um dado reticulado, se encontre códigos com grandes ganhos. Neste trabalho estamos interessados em reticulados e sub-reticulados no máximo bi-dimensionais. Os sub-reticulados são obtidos através de transformações ortogonais, as quais pouco alteram o ganho fundamental de codificação para constelações planares, de tal modo que nosso interesse vai se restringir somente ao uso da distância mínima do reticulado.

Figura 2.5: Reticulado  $A_2$ .

Outro parâmetro, porém de menor importância, é o número de pontos vizinhos que distam  $d_{min}^2$  de um dado ponto, que também é conhecido como multiplicidade do reticulado (sub-reticulado). Para que apresente um desempenho melhor, é evidente, que um dado reticulado deve possuir a menor multiplicidade possível. Caso dois reticulados apresentem a mesma distância mínima, a multiplicidade pode servir como elemento de desempate.

**Exemplo 2.4** Como exemplo, suponha dois tipos de sub-reticulado para o caso quaternário dado pelas seguintes matrizes de transformação ortogonal:

$$T_1 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \quad (2.14)$$

$$T_2 = \begin{bmatrix} 2 & 1 \\ 0 & 2 \end{bmatrix} \quad (2.15)$$

Observe que a distância quadrática mínima para ambos os casos é de 4 (pois a menor distância entre pontos do reticulado é unitária e supondo-se que estas células expandam-se em todas as direções). Por outro lado, a multiplicidade de  $T_1$  é igual a 4, enquanto a de  $T_2$  vale 2, conforme se observa na figura 2.6. Portanto a segunda transformação pode proporcionar maiores ganhos.



Figura 2.6: Reticulados quaternários com mesma distância mínima, mas com multiplicidades diferentes.

## 2.5 MODULAÇÃO DE AMPLITUDE DE PULSO: PAM

### 2.5.1 Introdução

Vamos analisar um exemplo que servirá não só para mostrar a efetividade do esquema proposto por Ungerboeck [1], como para que com o domínio da mecânica do assunto, possamos expandir e generalizar a teoria da modulação-codificada nos próximos capítulos.

Suponha que desejamos transmitir um trem de bits sem codificação através de uma constelação de sinais do tipo 4-PAM, conforme mostra a figura 2.7. A energia média de uma constelação PAM de  $M$  símbolos vale:

$$\mathcal{E} = \frac{\Delta^2}{12}(M^2 - 1) \quad (2.16)$$

onde  $\Delta$  é a menor distância entre pontos da constelação, e a probabilidade de erro de símbolo, para o caso PAM é dada por:

$$P_{simb} = 2 \frac{M-1}{M} Q\left(\sqrt{\frac{SNR \Delta^2}{4\mathcal{E}}}\right) \quad (2.17)$$

onde  $SNR$  é a relação sinal-ruído:

$$SNR = \frac{2\mathcal{E}}{N_0}$$

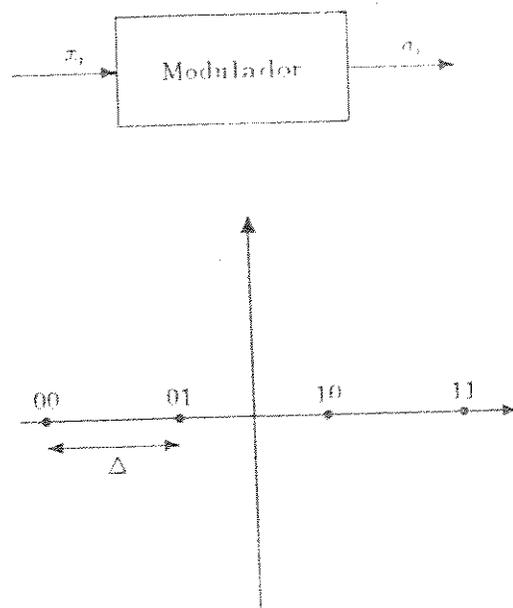


Figura 2.7: Esquema de um sistema convencional de transmissão e de uma constelação 4-PAM.

$\frac{N_0}{2}$  é a densidade espectral bilateral do ruído e  $Q(i)$  é a função de erro dada por:

$$Q(i) \triangleq \frac{1}{\sqrt{2\pi}} \int_i^{\infty} e^{-x^2/2} dx$$

Para o caso 4-PAM a energia média e o desempenho de um sistema de transmissão sem codificação são dados por:

$$\mathcal{E}_{4PAM} = \frac{5}{4} \Delta^2$$

$$F_{simb} = \frac{3}{2} Q\left(\sqrt{\frac{SNR}{5}}\right)$$

Vamos analisar agora, que vantagem em termos de desempenho o esquema de modulação-codificada tem a oferecer quando comparado com o sistema não codificado. Vamos iniciar pela expansão da constelação e fazer depois o mapeamento por partição de conjuntos.

Vamos expandir de 4-PAM para 8-PAM (como veremos na secção 3.4, uma expansão maior proporciona um ganho praticamente igual ao caso da expansão de  $k$  bits/símbolo para  $k + 1$  bits/símbolo) e fazer a partição de conjuntos conforme a figura 2.8. A finalidade da partição de conjuntos é dar nomes aos pontos da constelação de modo a conseguir um máximo na distância entre pontos pertencentes a um mesmo sub-conjunto, como por exemplo, os elementos 000 e 100 que distam entre si de 4 e pertencem ao sub-conjunto  $C_n$ . Como veremos adiante, isso vai auxiliar na maximização da distância livre do código.

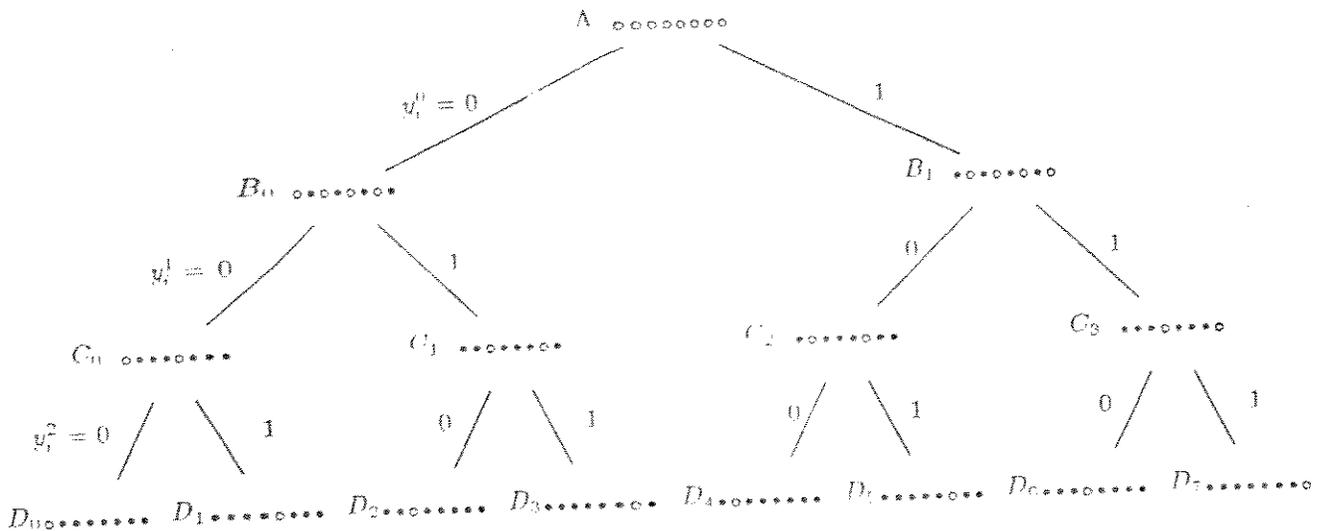


Figura 2.8: Partição de conjuntos e constelação 8-PAM.

As regras de alocação dos símbolos aos ramos da treliça foram propostas por Ungerboeck [1] e são rerepresentadas a seguir:

- Ramos que saem de um determinado estado recebem palavras do sub-conjunto  $B_0$  ou  $B_1$  (vide figura 2.8).
- Ramos que chegam a um determinado estado recebem palavras do sub-conjunto  $B_0$  ou  $B_1$ .<sup>1</sup>
- Ramos em paralelo com outros ramos recebem palavras do sub-conjunto  $C_0$  ou  $C_1$  ou  $C_2$  ou ainda  $C_3$ .

### 2.5.2 Codificador com um elemento de memória

Vamos agora, projetar o codificador para tirar proveito do esquema de partição da figura 2.8. Vamos iniciar com codificadores de apenas uma memória.

A expansão de 4-PAM para 8-PAM nos diz que necessitamos usar um codificador de parâmetros (3,2,1), conforme nos mostra a figura 2.9. Nesta figura encontra-se também um dos vários modos possíveis de se alocar os elementos da constelação expandida aos ramos da treliça.

Observe que aos ramos que saem do estado 0, alocou-se os símbolos pertencentes ao sub-conjunto  $B_0$  e aos ramos que saem do estado 1, pontos de  $B_1$ . Os ramos que chegam a um estado, neste caso em particular, não conseguem satis-

<sup>1</sup>No caso de códigos de uma memória, a segunda regra não é verdadeira.

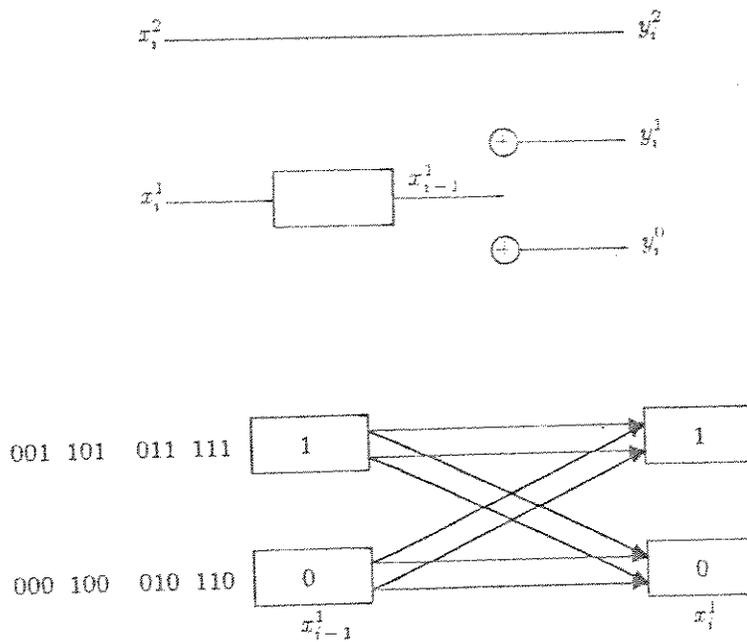


Figura 2.9: Esquema do codificador com 1 memória e treliça com os elementos da constelação alocados aos ramos. O tri-bit alocado a cada ramo representa, respectivamente, os elementos  $y_i^2$ ,  $y_i^1$  e  $y_i^0$ .

fazer a segunda regra de alocação. Aos ramos em paralelo alocou-se elementos dos sub-conjuntos  $C_i$ .

Da treliça da figura 2.9 determina-se que o  $d_{free}^2 = (4+1)\Delta^2 = 5\Delta^2$ , onde  $4\Delta^2$  é a distância quadrática para se sair de um certo estado e  $\Delta^2$  é a distância para se voltar ao mesmo estado. Além disso desta mesma treliça é possível determinarmos como deveremos realizar as conexões do codificador da figura 2.9, para que se consiga obter esta distância livre. Isto é conseguido através da regra de mintermos, muito usada em circuitos lógicos (ver Apêndice A.1):

$$\begin{aligned} y_i^0 &= x_{i-1}^1(x_i^1 + 1) + x_{i-1}^1 x_i^1 = x_{i-1}^1 \\ y_i^1 &= (x_{i-1}^1 + 1)x_i^1 + x_{i-1}^1 x_i^1 = x_i^1 \\ y_i^2 &= x_i^2 \end{aligned}$$

onde  $y_i^0$  e  $y_i^1$  representam as saídas e  $x_i^1$  e  $x_i^2$  são as entradas do codificador. A matriz que representa este codificador é  $G = [1\ 0\ 0\ 1]$

O codificador que apresenta a treliça da figura 2.9 junto do mapeador da figura 2.8, encontram-se na figura 2.10, onde constituem o esquema completo de modulação-codificada. A comparação que se faz aqui, é sobre a viabilidade em se transmitir 2 bits/símbolo conforme a figura 2.7, isto é, 4-PAM sem codificação, ou de acordo com a figura 2.10, 8-PAM com codificação, mas transmitindo-se os mesmos 2 bits/símbolo.

Aparentemente não se deveria ter muita vantagem em se transmitir 2 bits/símb

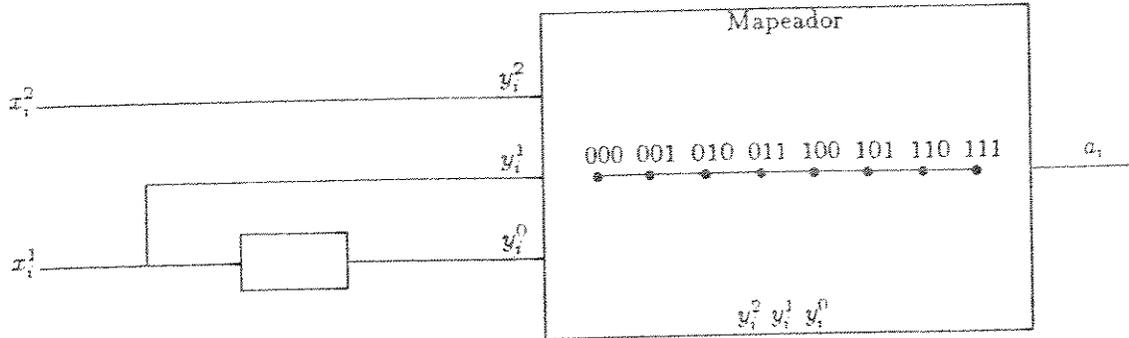


Figura 2.10: Estrutura do codificador de 1 memória e mapeador 8-PAM.

usando 8-PAM, pois a energia média desta constelação é  $\mathcal{E}_{8PAM} = \frac{21}{4}\Delta^2$ , enquanto que  $\mathcal{E}_{4PAM} = \frac{5}{4}\Delta^2$ . A desvantagem é de  $\frac{21}{5}$  vezes ou aproximadamente 6,23 dB.

Entretanto o processo de decodificação permite recuperar esta desvantagem e ainda obter um ganho em relação a sistemas sem codificação. Com o uso de decodificação suave e de máxima verossimilhança, a probabilidade de erro de evento para altas relações sinal-ruído é dada por [1]:

$$P_{evento} \simeq N(d_{free}) Q\left(\sqrt{\frac{d_{free}^2 SNR}{4\mathcal{E}}}\right) \quad (2.18)$$

onde  $N(\cdot)$  é a multiplicidade média de erro de evento.

Comparando-se a equação 2.18 com a equação 2.17, tira-se o ganho de relação sinal-ruído, isto é, quantos dB a menos podemos ter na relação sinal-ruído do esquema de modulação-codificada comparado com a relação sinal-ruído do esquema convencional e ainda termos a mesma taxa de erro. Este ganho é dado por:

$$\mathcal{G} = \frac{SNR_{ncod}}{SNR_{cod}} = \frac{d_{free}^2 \mathcal{E}_{ncod}}{\Delta^2 \mathcal{E}_{cod}} \quad (2.19)$$

Para o nosso caso em particular,  $\mathcal{E}_{8PAM}/\mathcal{E}_{4PAM} = \frac{21}{5}$  e  $d_{free}^2 = 5\Delta^2$  fazendo com que o ganho de codificação seja  $\mathcal{G} = \frac{25}{21}$ , ou seja, 0,76 dB. Para que esse ganho seja conseguido, é necessário que a distância entre ramos paralelos não seja inferior à distância livre do código, pois antes de se executar a decodificação suave ao longo da treliça, é feita uma decodificação abrupta entre as transições paralelas, quando se faz a opção pelo símbolo mais provável. Observe que a distância quadrática entre

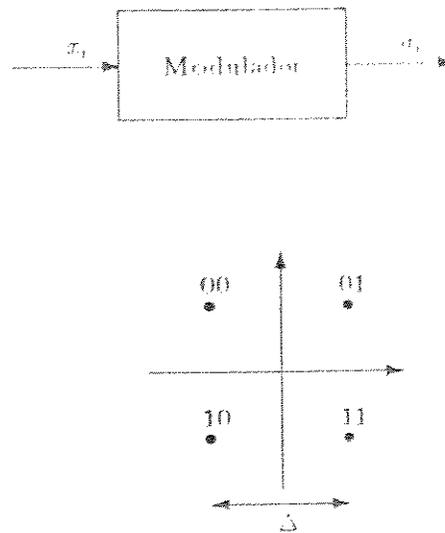


Figura 2.11: Esquema de um sistema convencional de transmissão e de uma constelação 4-QAM.

transições paralelas. é a distância quadrática entre elementos dos sub-conjuntos  $C_i$ , sendo igual a  $(4\Delta)^2 = 16\Delta^2$ .

## 2.6 MODULAÇÃO DE AMPLITUDE EM QUADRATURA DE FASE: QAM

### 2.6.1 Introdução

Suponha que desejamos transmitir um trem de bits sem codificação usando uma constelação de sinais do tipo 4-QAM, conforme mostra a figura 2.11. A energia média de uma constelação QAM de  $M$  símbolos (é exata quando  $M = 2^n$  e  $n$  é par) vale:

$$\mathcal{E} = \frac{\Delta^2}{6} (M - 1) \quad (2.20)$$

onde  $\Delta$  é a menor distância entre pontos da constelação. e a probabilidade de erro de símbolo, para o caso QAM é dada na forma exata também quando  $M = 2^n$  e  $n$  é par por:

$$P_{\text{simb}} = 2p - p^2 \quad (2.21)$$

$$p = 2\left(\frac{\sqrt{M} - 1}{\sqrt{M}}\right) Q\left(\sqrt{\frac{SNR \Delta^2}{2\mathcal{E}}}\right) \quad (2.22)$$

onde  $SNR$  é a relação sinal-ruído que vale,

$$SNR = \frac{\mathcal{E}}{N_0}$$

e  $\frac{N_0}{2}$  é a densidade espectral bilateral do ruído.

Para grandes  $SNR$ , independente da paridade de  $n$ , a probabilidade de erro de símbolo pode ser aproximada por:

$$P_{simb} \simeq 4 Q\left(\sqrt{\frac{SNR \Delta^2}{2\mathcal{E}}}\right) \quad (2.23)$$

Para o caso 4-QAM a energia média e o desempenho de um sistema de transmissão sem codificação são dados respectivamente por:

$$\mathcal{E}_{4QAM} = \frac{\Delta^2}{2}$$

$$P_{simb} \simeq 2 Q(\sqrt{SNR})$$

Novamente iremos analisar, que vantagem em termos de desempenho o esquema de modulação-codificada tem a oferecer quando comparado com aquele sem codificação. Como no caso anterior, iremos iniciar esta análise pela expansão da constelação seguida do mapeamento por partição de conjuntos.

Vamos expandir de 4-QAM para 8-QAM (como veremos na secção 5.3, uma expansão maior proporciona um ganho praticamente igual ao caso da expansão de  $k$  bits/símbolo para  $k + 1$  bits/símbolo) e fazer a partição de conjuntos conforme a figura 2.12. A finalidade da partição de conjuntos é conseguir um máximo na distância entre pontos pertencentes a um mesmo sub-conjunto, como por exemplo, os elementos 000 e 100 que distam entre si de 2 e pertencem ao sub-conjunto  $C_0$ . Como veremos adiante, este procedimento auxiliará na maximização na distância livre do código.

### 2.6.2 Codificador com um elemento de memória

Vamos agora, projetar o codificador para tirar proveito do esquema de partição da figura 2.12. Suponha que o codificador tenha apenas uma memória. A treliça gerada encontra-se na figura 2.13.

Observe que aos ramos que saem do estado 0, alocou-se os símbolos pertencentes ao sub-conjunto  $B_0$  e aos ramos que saem do estado 1, pontos de  $B_1$ . Os ramos que chegam a um estado, neste caso em particular, não conseguem satisfazer a segunda regra de alocação.

Da treliça da figura 2.13 determina-se que o  $d_{free}^2 = (2 + 1)\Delta^2 = 3\Delta^2$ , onde  $2\Delta^2$  é a distância quadrática para se sair de um certo estado e  $\Delta^2$  é a distância para se voltar ao mesmo estado. Além disso é possível determinarmos como deveremos realizar as conexões do codificador da figura 2.13, para que se consiga obter esta

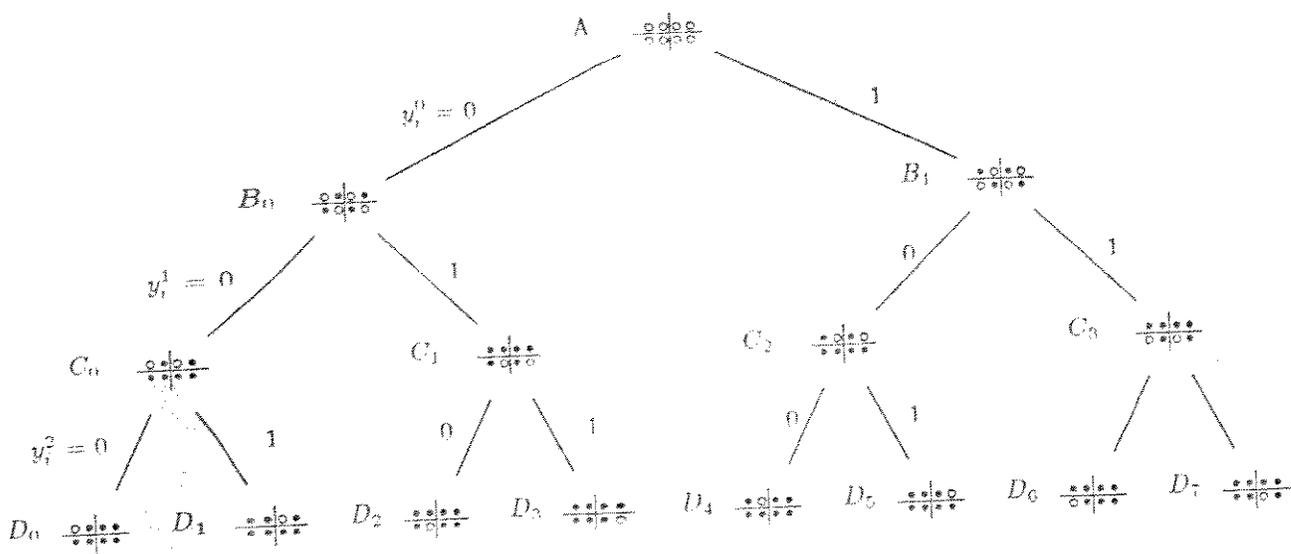


Figura 2.12: Partição de conjuntos e constelação 8-QAM.

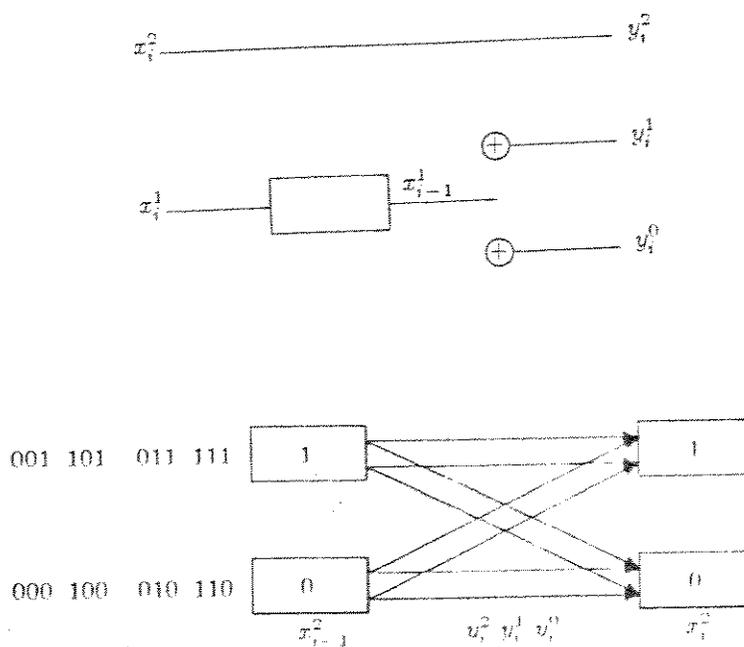


Figura 2.13: Esquema de codificador com 1 memória e treliça com os elementos de constelação alocados nos ramos correspondente à constelação 8-QAM.

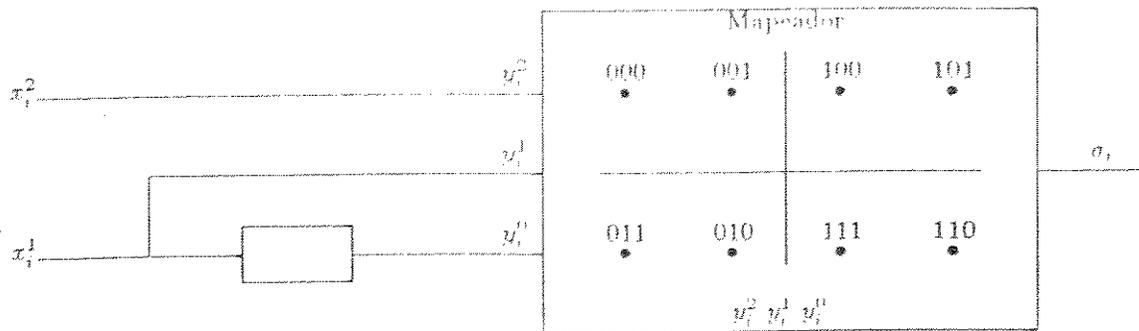


Figura 2.14: Estrutura do codificador de 1 memória e mapeador 8-QAM.

distância livre. Isto é possível através da aplicação da regra de mintermos:

$$\begin{aligned} y_i^0 &= x_{i-1}^1(x_i^1 + 1) + x_{i-1}^1 x_i^1 = x_{i-1}^1 \\ y_i^1 &= (x_{i-1}^1 + 1)x_i^1 + x_{i-1}^1 x_i^1 = x_i^1 \\ y_i^2 &= x_i^2 \end{aligned}$$

onde  $y_i^0$  e  $y_i^1$  representam as saídas e  $x_i^1$  e  $x_i^2$  são as entradas do codificador. A matriz que representa este codificador é  $G = [1 \ 0 \ 0 \ 1]$

O codificador que apresenta a treliça da figura 2.13 junto do mapeador da figura 2.12, encontram-se na figura 2.14, onde constituem o esquema completo de modulação-codificada. A comparação que se faz aqui, é sobre a viabilidade em se transmitir 2 bits/símbolo conforme a figura 2.11, isto é, 4-QAM sem codificação, ou de acordo com a figura 2.14, 8-QAM com codificação, mas transmitindo-se os mesmos 2 bits/símbolo.

Aparentemente não deveria existir vantagem em se transmitir 2 bits/símbolo usando 8-QAM, pois a energia média desta constelação é  $\mathcal{E}_{8QAM} = \frac{3}{2} \Delta^2$ , enquanto que  $\mathcal{E}_{4QAM} = \frac{1}{2} \Delta^2$ . A desvantagem é de 3 vezes ou aproximadamente 4,77 dB.

Entretanto, o processo de decodificação permite recuperar esta desvantagem e ainda obter um ganho em relação ao sistema não-codificado. Com o uso de decodificação suave e de máxima verossimilhança, a probabilidade de erro de evento para altas relações sinal-ruído é dada por:

$$P_{evento} \approx N(d_{free}) Q\left(\sqrt{\frac{d_{free}^2}{2\mathcal{E}} SNR}\right) \quad (2.24)$$

onde  $N(\cdot)$  é multiplicidade média de erro de evento.

Comparando-se a equação 2.24 com a equação 2.23, tira-se o ganho de relação sinal-ruído:

$$\mathcal{G} = \frac{SNR_{ncod}}{SNR_{cod}} = \frac{d_{free}^2 \mathcal{E}_{ncod}}{\Delta^2 \mathcal{E}_{cod}}$$

que é idêntica a equação 2.19. Portanto o ganho de relação sinal-ruído é o mesmo, independente do uso de constelações uni ou bi-dimensionais. O ganho depende apenas da relação de energias, e da distância livre quadrática do código.

Para o nosso caso em particular,  $\mathcal{E}_{8QAM}/\mathcal{E}_{4QAM} = 3$  e  $d_{free}^2 = 3\Delta^2$  fazendo com que neste caso o ganho de codificação seja  $\mathcal{G} = 1$ , ou seja, 0,00 dB. Observe que a distância quadrática entre transições paralelas, que é a distância quadrática entre sub-conjuntos  $C_i$ , vale  $(2\Delta)^2 = 4\Delta^2$ .

Usando códigos com até 10 memórias, Ungerboeck [1] conseguiu ganhos da ordem de até 6 dB. Nos próximos capítulos estudaremos codificadores mais complexos, que além de usar mais que uma memória, trabalham com símbolos  $Q$  ariais.

## Capítulo 3

# GENERALIZAÇÃO DE CÓDIGOS PARA MODULAÇÃO DO TIPO PAM

### 3.1 INTRODUÇÃO

No capítulo anterior analisamos um único caso de modulação codificada binária em que a modulação usada era do tipo PAM, e o codificador binário e de apenas uma memória ( $m = 1$ ).

Este capítulo pretende generalizar o trabalho de Ungerboeck [1], obtendo esquemas de modulações-codificadas  $Q$  *ários*, isto é, encontrar o codificador  $Q$  *ário* inicialmente com uma memória, depois com duas, e assim sucessivamente, e o esquema de partição de conjuntos associado. Como veremos o ganho obtido em relação a esquemas não codificados, será o maior possível, quando  $Q$  for primo, pois quando  $Q$  não é primo, a operação multiplicação  $\text{mod}(Q)$  não gera todos os elementos do corpo. Neste caso, em geral, os códigos lineares obtidos apresentarão um desempenho inferior aos códigos obtidos pelo primo mais próximo. Existem códigos não lineares superiores aos lineares nos casos em que  $Q$  não é primo, entretanto a procura destes códigos é inviável a partir de  $Q = 10$  devido à grande quantidade de códigos possíveis, mesmo com o uso de apenas uma memória.

### 3.2 PARTIÇÃO DE CONSTELAÇÕES UNI-DIMENSIONAIS

A figura 3.1 nos mostra o esquema completo da modulação-codificada generalizada, onde os códigos usados obviamente serão  $Q$  *ários*.

Na próxima secção estaremos analisando sucintamente os códigos  $Q$  *ários*. Neste momento vamos generalizar o esquema de partição para constelações de uma dimensão de uma maneira intuitiva (ao contrário das constelações bi-dimensionais). No capítulo 4 o esquema de partição de conjuntos é analisado com mais profundidade, e lá o particionamento uni-dimensional é visto como sendo um caso particular de uma concepção mais ampla.

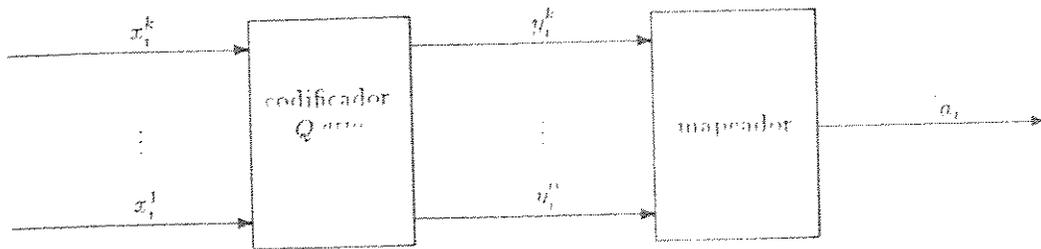


Figura 3.1: Esquema da modulação-codificada generalizada.

Dada uma constelação de  $Q^{k+1}$  pontos, a melhor maneira de se realizar o particionamento consiste em selecionar os pontos pertencentes a um mesmo sub-conjunto distantes de  $Q\Delta$  entre si, ou seja, é o mesmo que caminhar pela constelação com um passo de  $Q$ , conforme mostra a figura 3.2. Para um codificador com  $k$  entradas, e  $k - \bar{k}$  entradas não-codificadas, a partição apresentará  $Q^{k+1}$  pontos no total com  $Q^{k+1}$  sub-conjuntos e  $Q^{k-\bar{k}}$  elementos por sub-conjunto.

### 3.3 CÓDIGOS ÓTIMOS PARA O RETICULADO UNIDIMENSIONAL $Z$

#### 3.3.1 Códigos de treliça lineares com um elemento de memória

##### Introdução

A representação para este tipo de código está na figura 3.3, onde  $y_i^0$  e  $y_i^1$  são saídas que dependem da entrada  $x_i^1$ , além disso  $y_i^0$  e  $y_i^1$  são funções do tipo:

$$y_i^1 = k_1 x_i^1 + k_3 x_{i-1}^1$$

$$y_i^0 = k_2 x_i^1 + k_4 x_{i-1}^1$$

onde  $y_i^0$  e  $y_i^1$  dependem única e exclusivamente de  $x_{i-1}^1$  e  $x_i^1$  e as constantes  $k_i$  são inteiros entre 0 e  $Q - 1$ . Segundo a notação de códigos adotada na literatura, diz-se que este código tem matriz geradora  $G = [k_1 \ k_2 \ k_3 \ k_4]$ .

Nosso interesse é trabalhar com códigos de razão  $R = \frac{k}{k+1}$  onde  $k \geq 2$ , que pressupõe a utilização de pelo menos uma entrada não codificada, implicando no

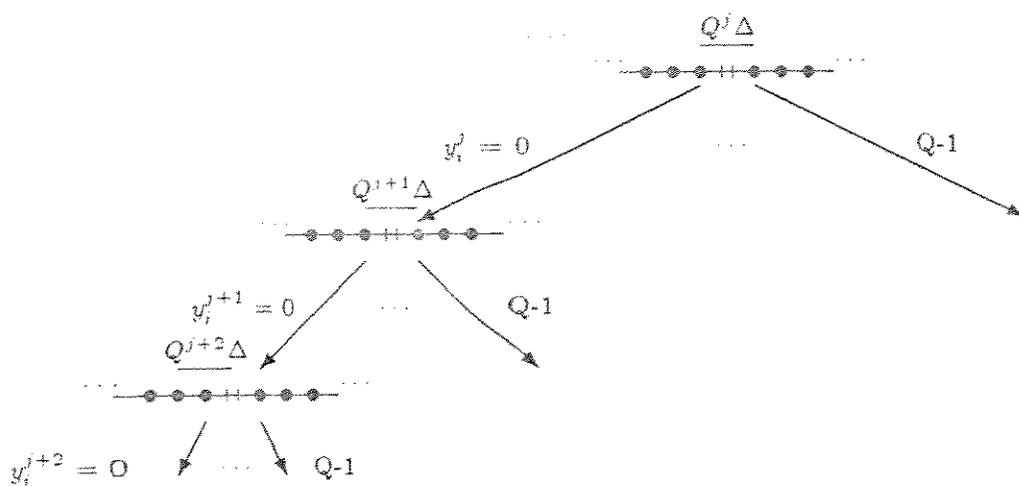


Figura 3.2: Esquema da partição de conjuntos generalizada para constelações PAM.

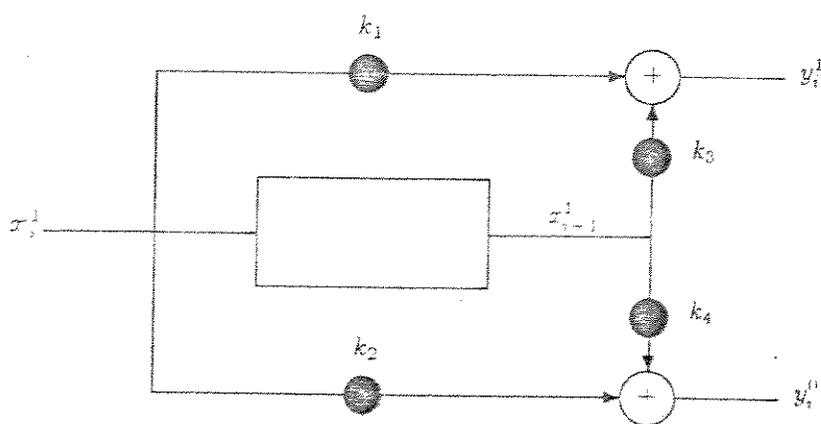


Figura 3.3: Código convolucional geral com 1 elemento de memória

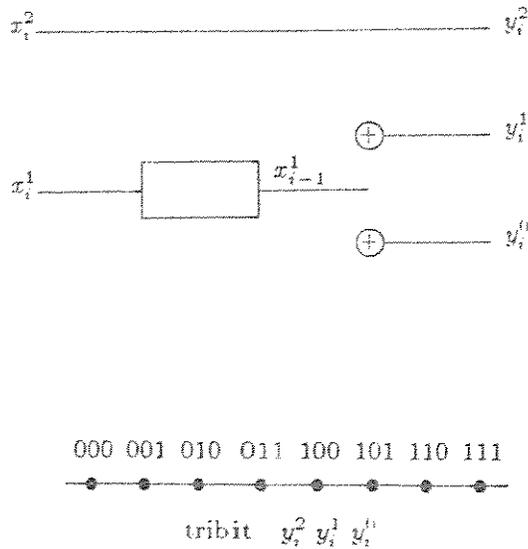


Figura 3.4: Codificador com duas entradas e constelação de pontos.

uso de mais de uma entrada. A utilização de entradas não codificadas, acarretará em transições paralelas dentro da treliça, que além de limitar a distância livre em  $Q^4 \Delta^2$  (que é a distância entre elementos do sub-grupo 00), vai proporcionar uma distribuição de pontos mais uniforme e poderá apresentar portanto, uma distância livre menor que no caso em que não há transições paralelas ( $k = 1$ ). Esta questão relativa a uma distribuição mais uniforme vai ficar mais clara no exemplo a seguir. Constelações PAM dão origem a códigos que não apresentam a propriedade da uniformidade de erro (exceto para o caso binário), tal que para a obtenção da distância livre na treliça, não podemos usar a sequência de referência nula.

**Exemplo 3.1** Na figura 2.8, é apresentado o esquema de partição, e na figura 3.4 a constelação com os elementos já rotulados e o codificador binário com 2 entradas ( $k = 2$ ). Para este caso temos que  $k = 2$  e  $k - \bar{k} = 1$ ; portanto haverá  $2^{k+1} = 8$  elementos e  $2^{k-\bar{k}} = 4$  sub-conjuntos com  $2^{\bar{k}} = 2$  elementos cada. Observe que caso tivéssemos somente uma entrada ( $k = 1$ ) e nenhuma entrada sem codificação ( $k - \bar{k} = 0 \Rightarrow \bar{k} = 1$ ) teríamos 4 sub-conjuntos com 1 elemento cada, conforme a figura 3.5.

Veja que neste caso a distância entre os elementos 00 e 11 vale  $3\Delta$  enquanto que no caso anterior a distância entre elementos do sub-conjunto 00 e 11 é de apenas  $\Delta$ , proporcionando o que se chamou anteriormente de uma distribuição de pontos mais uniforme. Portanto a partir de agora vamos nos preocupar com códigos em que  $k = \bar{k} = 1$ , deixando os códigos em que  $k - \bar{k} = 0$  para uma outra oportunidade. Como está implícito o uso de entradas não codificadas, a partir de agora vamos nos preocupar somente com o codificador em si, esquecendo que

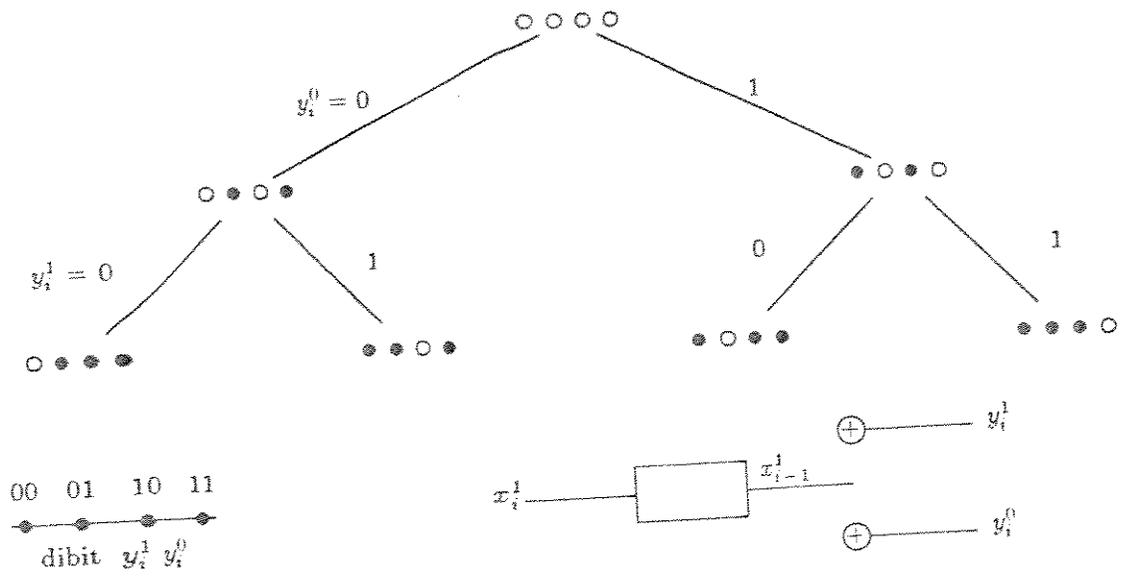


Figura 3.5: Partição, constelação e codificador binários  $k = 1$ .

existem transições paralelas na treliça para o cálculo da distância livre. Quando mencionarmos o elemento 00 por exemplo, implicitamente estaremos dizendo que o sub-conjunto 00 que agrupa os elementos 000 e 100, e assim por diante. Após a obtenção da distância livre, deve-se calcular sempre a distância entre transições paralelas, para se garantir que o desempenho não seja limitado por esta última.

Vamos a seguir calcular as distâncias livres máximas para cada esquema de modulação  $Q$ -ária, e descobrir qual codificador atende esta máxima distância livre.

### Códigos Binários

A partição de conjunto, a constelação de pontos, e o codificador encontram-se na figura 3.5. e a treliça na figura 3.6.

Os símbolos associados aos ramos que saem de um estado, devem ser escolhidos tais que a distância entre si seja a maior possível. Os elementos que satisfazem esta condição são os elementos 00 e 10 ou 01 e 11, pontos que distam quadraticamente entre si de  $2^2 \Delta^2 = 4\Delta^2$ . Os símbolos associados aos ramos que chegam a um estado são pontos distantes entre si de  $\Delta^2$  (por exclusão), tais como os pontos 00 e 01 ou 00 e 11 ou ainda 10 e 01 ou 10 e 11.

Uma dentre as várias possíveis alocações ótimas dos símbolos aos ramos da treliça está na figura 3.6, donde se tira que o  $d_{free}^2 = (4 + 1)\Delta^2 = 5\Delta^2$ , e o código que satisfaz esta alocação pode ser obtido através da regra de minitermos: (ver Apêndice)

$$y_i^1 = x_i^1(x_{i-1}^1 + 1) + (x_i^1 + 1)x_{i-1}^1$$

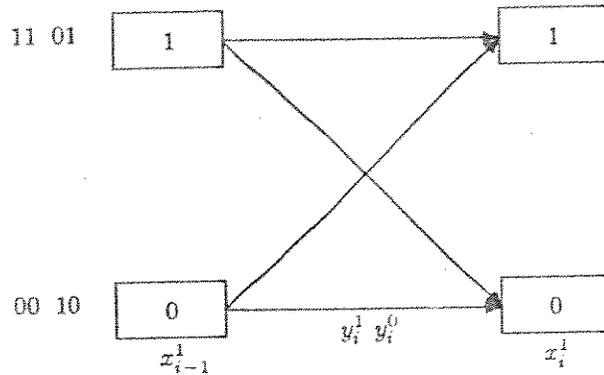


Figura 3.6: Treliça ótima do codificador binário com uma memória.

$$\begin{aligned}
 &= x_i^1 + x_{i-1}^1 \\
 y_i^0 &= (x_i^1 + 1)x_{i-1}^1 + x_i^1 x_{i-1}^1 \\
 &= x_{i-1}^1
 \end{aligned}$$

onde os símbolos associados aos ramos são representados na figura 3.6 por  $y_i^1$  e  $y_{i-1}^1$ , e estes são calculados através da soma de produtos de  $x_{i-1}^1$  e  $x_i^1$ , onde  $x_i^1$  é o bit de entrada presente e  $x_{i-1}^1$  é o bit futuro. A soma usada é a soma módulo 2, portanto quando se tiver que negar uma variável basta substituir esta negação por  $\bar{x} = x + 1$  e proceder a multiplicação convencional seguida da adição módulo 2.

O codificador que executa a função calculada anteriormente encontra-se na figura 3.7, e o gerador que descreve este código é [1 0 1 1].

Como mencionado anteriormente a ocorrência de entradas não codificadas, implicará em transições paralelas aos ramos da treliça. Neste caso a distância entre transições paralelas é de  $2^4 \Delta^2 = 16 \Delta^2$ .

O aumento no número de entradas não codificadas implicará num aumento da constelação. A energia média de um esquema de modulação PAM com  $M$  possíveis símbolos é dada por: (vide equação 2.16)

$$\mathcal{E} = \frac{\Delta^2}{12} (M^2 - 1) \tag{3.1}$$

O ganho de codificação vale (vide equação 2.19):

$$\mathcal{G} = \frac{d_{f_{res}}^2}{\Delta^2} \frac{\mathcal{E}_{ncod}}{\mathcal{E}_{cod}} \tag{3.2}$$

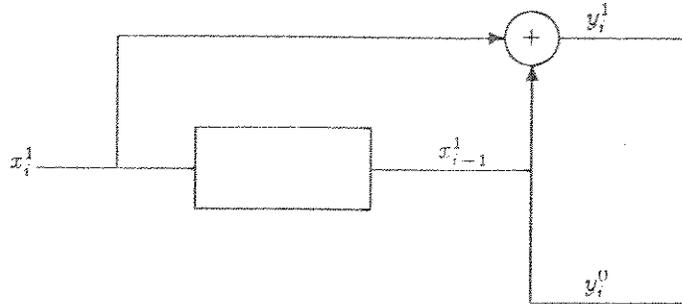


Figura 3.7: Codificador ótimo binário.

onde:

$$\mathcal{E}_{cod} = \frac{\Delta^2}{12} [(Q^{k+1})^2 - 1]$$

$$\mathcal{E}_{ncod} = \frac{\Delta^2}{12} [(Q^k)^2 - 1]$$

pois os códigos usados são sempre de razão  $\frac{k}{k+1}$

O ganho de codificação varia em função do número de entradas; à medida que  $k$  aumenta, as constelações tendem a ficar cada vez maiores e a razão entre energias tende a  $Q^2$ , fazendo com que o ganho tenda a  $\mathcal{G} = \frac{d_{f,ncod}^2}{Q^2 \Delta^2}$ . Este comportamento pode ser visto abaixo:

- Para  $k = 2$  temos que  $\mathcal{E}_{ncod} = \frac{4^2-1}{12} \Delta^2 = \frac{5}{4} \Delta^2$  enquanto que  $\mathcal{E}_{cod} = \frac{8^2-1}{12} \Delta^2 = \frac{21}{4} \Delta^2$  e  $\mathcal{G} = 5 \frac{5}{21}$  ou ainda  $\mathcal{G} = 0,76 \text{ dB}$ .
- Para  $k = 3$  temos que  $\mathcal{E}_{ncod} = \frac{21}{4} \Delta^2$  enquanto que  $\mathcal{E}_{cod} = \frac{85}{4} \Delta^2$  e  $\mathcal{G} = 5 \frac{21}{85} \Rightarrow 0,92 \text{ dB}$ .
- Assintoticamente quando  $k \rightarrow \infty$   $\frac{\mathcal{E}_{cod}}{\mathcal{E}_{ncod}} = 4$  então  $\mathcal{G} = 0,97 \text{ dB}$ .

Portanto à medida que o número de entradas não codificadas cresce, a constelação também cresce, provocando um aumento no ganho de codificação, devido ao comportamento assintótico envolvido com a razão de energias codificada e não codificada. Portanto, independente do número de entradas não codificadas ( $k - k \geq 1$ ) o código obtido é ótimo e apresenta um  $d_{f,ncod}^2$  constante para todos os valores de  $k$  ( $k \geq 2$ ).

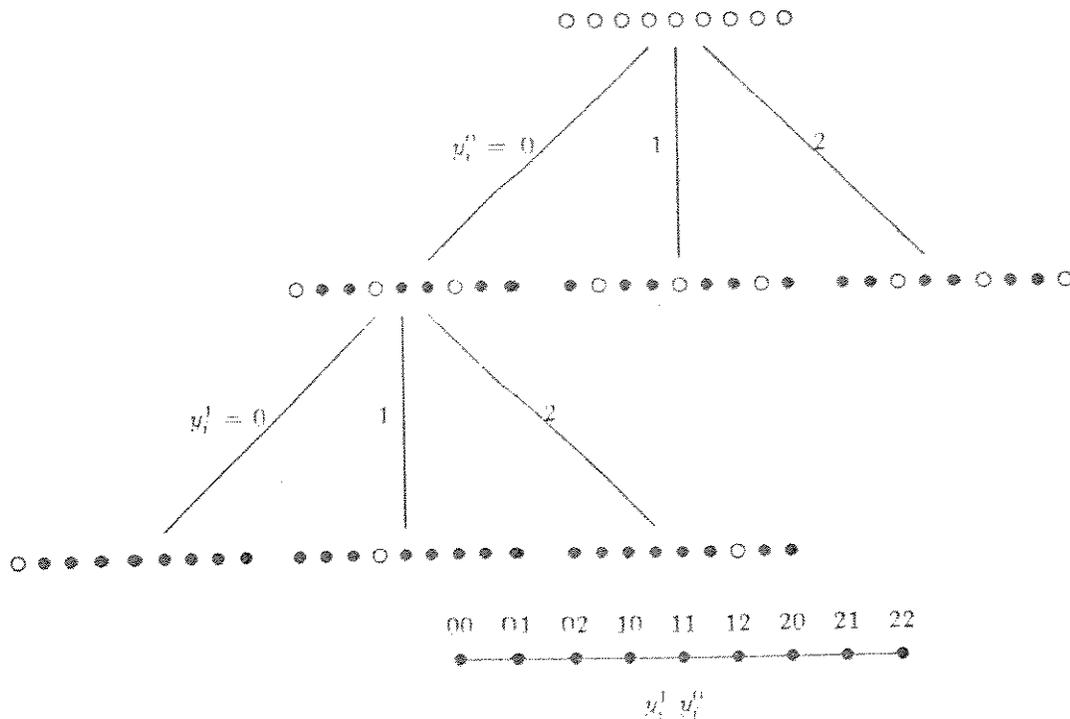


Figura 3.8: Partição de conjuntos e constelação de pontos para o caso ternário.

### Códigos Ternários

Para o caso ternário, a partição de conjuntos e a constelação de pontos, encontram-se na figura 3.8.

Os pontos da constelação que devem ser alocados aos ramos que saem de um nó, devem distar entre si a maior distância possível, que é o caso dos pontos 00, 10, 20 ou 01, 11, 21 ou ainda 02, 12, 22. Os pontos da constelação que devem ser alocados aos ramos que chegam a um nó, também devem distar entre si o máximo possível, como é o caso dos pontos 00, 21, 12, ou 01, 22, 10, ou ainda 02, 20, 11.

Associando estes pontos aos ramos da treliça, conforme podemos ver na figura 3.9, vamos determinar qual o código que satisfaz este  $d_{free}^2$  máximo e quanto ele vale. Os pontos na horizontal correspondem a ramos que saem, enquanto que os pontos em uma vertical estão associados a ramos que chegam a um nó. Ressaltando mais uma vez, observe que pontos em qualquer horizontal estão distantes entre si de pelo menos  $Q^2 \Delta^2 = 9 \Delta^2$ , enquanto que os pontos pertencentes a qualquer vertical estão distantes entre si de pelo menos  $(Q-1)^2 \Delta^2 = 4 \Delta^2$ , portanto o  $d_{free}^2 = (9+4) \Delta^2 = 13 \Delta^2$ . Observe que as colunas ou linhas são entidades independentes, implicando que um rearranjo na ordem das colunas ou linhas nos fornecerá outros códigos, que neste caso em particular ( $Q = 3$ ), apresentarão mesmo desempenho que o código acima citado.

Ao se expandir de uma constelação de  $3^k$  pontos para uma outra de  $3^{k+1}$  pontos, a energia da constelação aumenta de  $3^2$  vezes, de tal modo que o ganho apresente o seguinte comportamento assintótico :

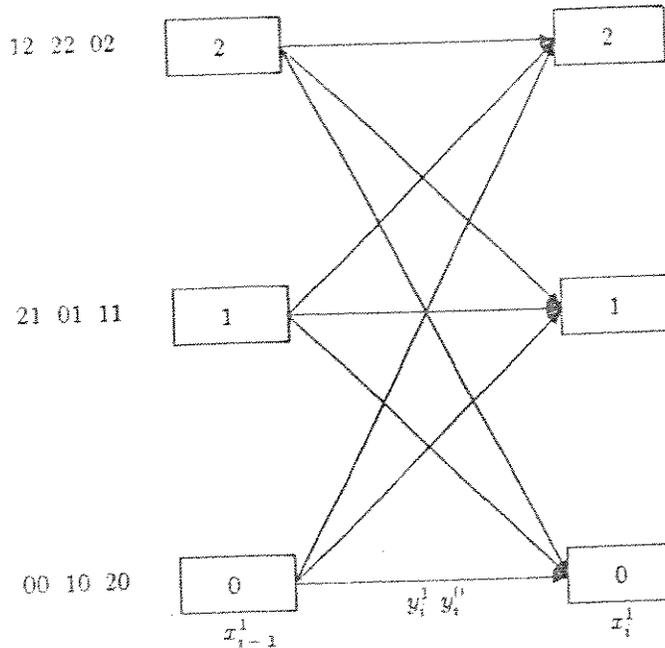


Figura 3.9: Treliça ótima do codificador ternário de uma memória.

- Para  $k = 2$  temos que  $\mathcal{E}_{ncod} = \frac{20}{3} \Delta^2$  enquanto que  $\mathcal{E}_{cod} = \frac{182}{3} \Delta^2$  fazendo com que  $\mathcal{G} = 13 \frac{20}{182} \Rightarrow \mathcal{G} = 1,55 \text{ dB}$ .
- Para  $k = 3$  temos que  $\mathcal{E}_{ncod} = \frac{182}{3} \Delta^2$  enquanto que  $\mathcal{E}_{cod} = \frac{1640}{3} \Delta^2$  fazendo com que  $\mathcal{G} = 13 \frac{182}{1640} \Rightarrow \mathcal{G} = 1,59 \text{ dB}$ .
- Para  $k = \infty$  temos que  $\mathcal{G} = 10 \log(\frac{13}{9}) = 1,60 \text{ dB}$ .

Um dos códigos (figura 3.9) com o qual se consegue o desempenho acima mencionado, pode ser obtido através do método dos mintermos generalizados (ver o Apêndice):

$$y_i^1 = (x_{i-1}^1 + 1)(x_{i-1}^1 + 2)x_i^1(x_i^1 + 1) + 2(x_{i-1}^1 + 1)(x_{i-1}^1 + 2)x_i^1(x_i^1 + 2) - 2x_{i-1}^1(x_{i-1}^1 + 1)(x_i^1 + 1)(x_i^1 + 2) + x_{i-1}^1(x_{i-1}^1 + 1)x_i^1(x_i^1 + 2) + x_{i-1}^1(x_{i-1}^1 + 2)(x_i^1 + 1)(x_i^1 + 2) + 2x_{i-1}^1(x_{i-1}^1 + 2)x_i^1(x_i^1 + 1)$$

$$y_i^1 = x_i^1 + 2x_{i-1}^1$$

$$y_i^0 = x_{i-1}^1(x_{i-1}^1 + 1)(x_i^1 + 1)(x_i^1 + 2) + x_{i-1}^1(x_{i-1}^1 + 1)x_i^1(x_i^1 + 1) + x_{i-1}^1(x_{i-1}^1 + 1)x_i^1(x_i^1 + 2) + 2x_{i-1}^1(x_{i-1}^1 + 2)(x_i^1 + 1)(x_i^1 + 2) + 2x_{i-1}^1(x_{i-1}^1 + 2)x_i^1(x_i^1 + 1) + 2x_{i-1}^1(x_{i-1}^1 + 2)x_i^1(x_i^1 + 2)$$

$$y_i^0 = x_{i-1}^1$$

O código desejado encontra-se na figura 3.10, e a matriz geradora que descreve este código é  $[1 \ 0 \ 2 \ 1]$ . A distância entre transições paralelas é igual a

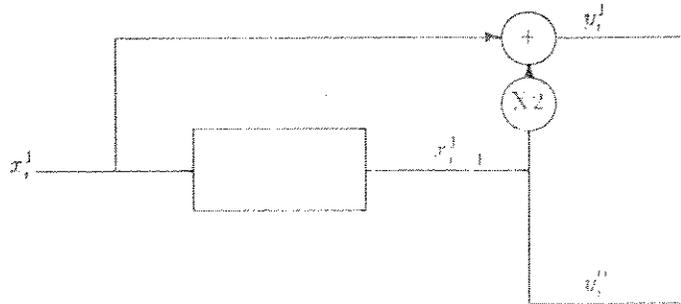


Figura 3.10: Código ternário ótimo.

$3^4\Delta^2$ . Como veremos, para o caso PAM, as transições paralelas só restringem o ganho obtido, quando estivermos trabalhando com códigos binários com mais de 8 memórias; portanto para  $Q \neq 2$  não precisamos nos preocupar mais com as transições paralelas.

#### Códigos $Q$ -ários

A aplicação sistemática do procedimento anterior para um  $Q$  genérico, nos conduz à tabela 3.1, obtida por meios computacionais, que fornece o  $d_{free}^2$ , o ganho em dB, e um dos códigos ótimos que atinge este desempenho.

Observe também que conforme  $Q$  aumenta, o número de intervalos de tempo para se retornar ao mesmo estado para se obter a mínima distância livre também aumenta, e o código ótimo quando  $Q$  é primo é sempre do tipo  $[1 \ 0 \ \frac{Q-1}{2} \ \frac{Q+1}{2}]$ .

Da tabela 3.1 pode-se inferir que o  $d_{free}^2$  assim como o número de intervalos de tempo para  $Q$  primo valem:

$$\frac{d_{free}^2}{\Delta^2} = Q^2 + S \quad (3.3)$$

onde

$$S = \sum_{i=1}^N a_i^2 \quad (3.4)$$

onde  $a_N = 2$  ou  $a_N = 1$

$$a_1^2 = (Q - 1)^2 \quad (3.5)$$

	Q=2	Q=3	Q=4	Q=5
$\frac{d_{free}^2}{\Delta^2}$	4+1=5	9+4= 13	16+9=25	25+16+4=45
$\mathcal{G}$	0,97 dB	1,60 dB	1,94 dB	2,55 dB
Matriz	[1 0 1 1]	[1 0 1 2]	[1 0 1 3]	[1 0 2 3]
	Q=6	Q=7	Q=8	Q=9
$\frac{d_{free}^2}{\Delta^2}$	36+25=61	49+36+9+4=98	64+49+9=122	81+64+16+4=165
$\mathcal{G}$	2,29 dB	3,01 dB	2,80 dB	3,09 dB
Matriz	[1 0 1 5]	[1 0 3 4]	[1 0 3 5]	[1 0 4 5]
	Q=10	Q=11	Q=12	Q=13
$\frac{d_{free}^2}{\Delta^2}$	100+81+9= 190	121+100+25+9+4= 259	290	169+144+36+9+4= 362
$\mathcal{G}$	2,79 dB	3,31 dB	3,04 dB	3,31 dB
Matriz	[1 0 3 7]	[1 0 5 6]	[1 0 5 7]	[1 0 6 7]
	Q=14	Q=15	Q=16	Q=17
$\frac{d_{free}^2}{\Delta^2}$	399	490	530	289+256+64+16+4=629
$\mathcal{G}$	3,09 dB	3,38 dB	3,16 dB	3,38 dB
Matriz	[1 0 3 11]	[1 0 7 8]	[1 0 7 9]	[1 0 8 9]
	Q=18	Q=19	Q=23	Q=41
$\frac{d_{free}^2}{\Delta^2}$	682	361+324+81+25+9+4+1=805	529+484+121+36+9+4=1183	1681+1600+400+100+25+9+4=3819
$\mathcal{G}$	3,23 dB	3,48 dB	3,50 dB	3,56 dB
Matriz	[1 0 7 11]	[1 0 9 10]	[1 0 11 12]	[1 0 20 21]

Tabela 3.1: A distância livre, o ganho e a matriz geradora dos códigos para uma memória.

$$a_i = \text{MAINT}\left(\frac{a_{i-1}}{2}\right) \text{ para } i > 1 \quad (3.6)$$

onde a função  $\text{MAINT}(k + \delta) = k + 1$

MAINT: maior inteiro

$k$ : inteiro qualquer

$\delta$ : número real tal que  $0 < \delta \leq 1$

O número de intervalos de tempo é obtido por:

$$N = N + 1 \quad (3.7)$$

**Exemplo 3.2** Por exemplo se  $Q=19$

$$\begin{aligned} \frac{d_{free}^2}{\Delta^2} &= 19^2 + 18^2 + 9^2 + 5^2 + 3^2 + 2^2 + 1^2 \\ N &= 7 \end{aligned}$$

Vamos calcular a seguir o ganho assintótico para  $Q \rightarrow \infty$  e  $k \rightarrow \infty$ . Usando as equações 3.3, 3.4, 3.6. e 3.6 podemos escrever que:

$$\frac{d_{free}^2}{\Delta^2} = Q^2 + (Q-1)^2 + \left(\frac{Q-1}{2}\right)^2 + \left(\frac{Q+\alpha}{4}\right)^2 + \left(\frac{Q+\beta}{8}\right)^2 + \dots \quad (3.8)$$

onde  $\alpha$  e  $\beta$  são inteiros quaisquer.

Vamos calcular o ganho assintótico:

$$\begin{aligned} \mathcal{G} = \lim_{Q \rightarrow \infty} \lim_{k \rightarrow \infty} \frac{d_{free}^2}{(Q\Delta)^2} &\Rightarrow \mathcal{G} = \lim_{Q \rightarrow \infty} \frac{2Q^2 + \frac{Q^2}{4} + \frac{Q^2}{16} + \frac{Q^2}{64} + \dots}{Q^2} \\ \text{ou seja : } \mathcal{G} &= \frac{7}{3} = 3,68 \text{ dB} \end{aligned}$$

Portanto o ganho máximo que se consegue com modulações PAM com uma memória é de 3,68 dB.

Para  $Q$  não primo não se consegue códigos lineares tão bons quanto os obtidos para  $Q$  primo, pois a operação multiplicação  $\text{mod}(Q)$  por um elemento do corpo não gera todos os elementos, como é o caso de  $Q = 4$ , no qual a multiplicação por 2 gera somente os elementos 0 e o próprio 2. Assim sendo, quando  $Q$  não for primo, a solução alternativa é a busca de códigos não-lineares.

### Códigos não-lineares

Como mencionado anteriormente, para  $Q$  não primo, é possível encontrarmos códigos não-lineares melhores que os lineares, o que não pode ocorrer quando  $Q$  é primo.

**Exemplo 3.3** Como exemplo suponha o caso em que  $Q = 6$ . A melhor alocação das palavras aos ramos da treliça, encontra-se na figura 3.11.

Constata que neste caso conseguiu-se um  $d_{free}^2 = 65\Delta^2$ , portanto maior que no caso linear, quando se conseguiu somente  $61\Delta^2$ , acarretando em um aumento no ganho assintótico de 2,29 dB para 2,57 dB.

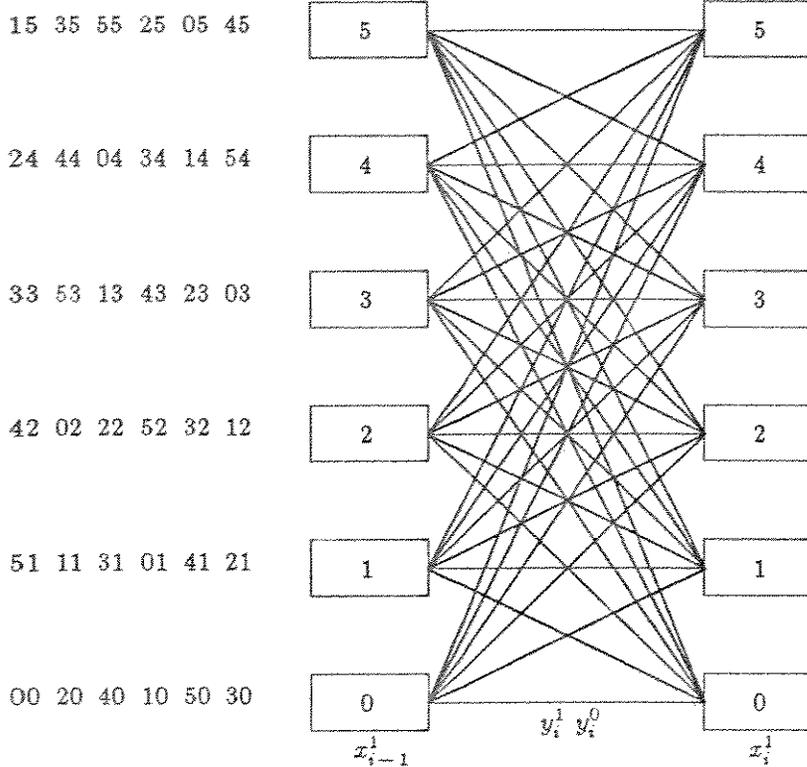


Figura 3.11: Exemplo de código não-linear para  $Q = 6$ , reticulado  $Z$ .

A procura através do computador de códigos não-lineares para  $Q > 10$ , torna-se mais e mais difícil, pois o número de códigos cresce com  $Q!$ .

### 3.3.2 Códigos de treliça lineares com dois elementos de memória

#### Introdução

A representação para este tipo de código encontra-se na figura 3.12, onde  $y_i^0$  e  $y_i^1$  são funções do tipo:

$$\begin{aligned} y_i^1 &= k_1 x_i^1 + k_3 x_{i-1}^1 + k_5 x_{i-2}^1 \\ y_i^0 &= k_2 x_i^1 + k_4 x_{i-1}^1 + k_6 x_{i-2}^1 \end{aligned}$$

onde as constantes  $k_i$  são inteiros entre 0 e  $Q - 1$ . A matriz geradora para este tipo de código é  $G = [k_1 \ k_2 \ k_3 \ k_4 \ k_5 \ k_6]$ .

Usou-se dois elementos de memória em cascata ao invés de ambos em entradas diferentes, pois o  $d_{free}^2$  ainda é bem menor que a distância envolvida entre transições paralelas da treliça ( $Q^4 \Delta^2$ ), e a colocação de dois elementos de memória em série permitirá obter um  $d_{free}^2$  maior.

O raciocínio utilizado anteriormente, implicando na utilização de pelo menos duas entradas permanece válido aqui, assim como a partição de conjuntos permanece a mesma. Neste caso a treliça, obviamente, sofre alteração passando a ter  $Q^2$  estados diferentes.

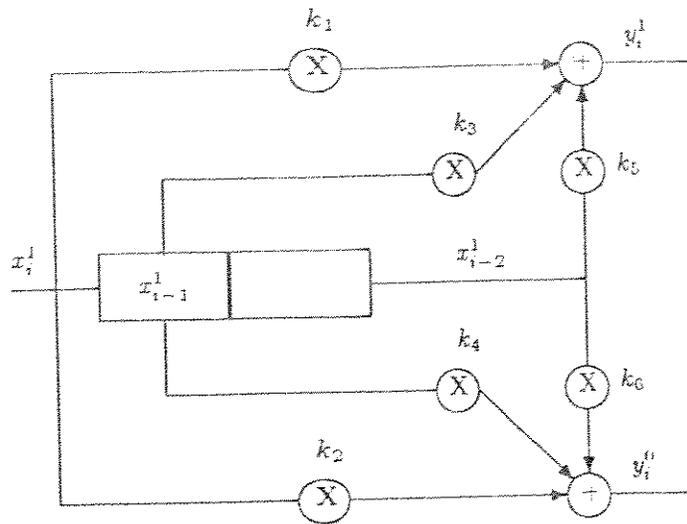


Figura 3.12: Código convolucional geral com dois elementos de memória.

Códigos Binários

Para  $Q = 2$  uma dentre as muitas possíveis alocações ótimas pode ser vista na figura 3.13, da qual tira-se que  $d_{frec}^2 = (4 + 1 + 4)\Delta^2 = 9\Delta^2$  e o código que satisfaz esta alocação é obtida através da regra de mintermos:

$$\begin{aligned}
 y_i^1 &= (x_{i-2}^1 + 1)(x_{i-1}^1 + 1)x_i^1 + (x_{i-2}^1 + 1)x_{i-1}^1(x_i^1 + 1) + x_{i-2}^1(x_{i-1}^1 + 1)(x_i^1 + 1) \\
 &\quad + x_{i-2}^1 x_{i-1}^1 x_i^1 \\
 &= x_{i-2}^1 + x_{i-1}^1 + x_i^1
 \end{aligned}$$

$$\begin{aligned}
 y_i^0 &= (x_{i-2}^1 + 1)x_{i-1}^1(x_i^1 + 1) + (x_{i-2}^1 + 1)x_{i-1}^1 x_i^1 + x_{i-2}^1 x_{i-1}^1 (x_i^1 + 1) + x_{i-2}^1 x_{i-1}^1 x_i^1 \\
 &= x_{i-1}^1
 \end{aligned}$$

A matriz geradora para este código é  $G = [1\ 0\ 1\ 1\ 1\ 0]$  e o codificador que descreve este código pode ser visto na figura 3.13.

Códigos Ternários

A treliça para o caso ternário pode ser vista na figura 3.14. O  $d_{frec}^2$  vale  $(9 + 1 + 1 + 9)\Delta^2 = 20\Delta^2 \ll Q^4\Delta^2 = 81\Delta^2$  e o código obtido pelo método dos mintermos generalizados é obtido a seguir:

$$\begin{aligned}
 y_i^1 &= 2(r^2 + r)(s^2 + 2)(t^2 + 2) + (r^2 + 2r)(s^2 + 2)(t^2 + 2) \\
 &\quad + (r^2 + 2)(s^2 + s)(t^2 + 2) + 2(r^2 + 2r)(s^2 + s)(t^2 + 2)
 \end{aligned}$$

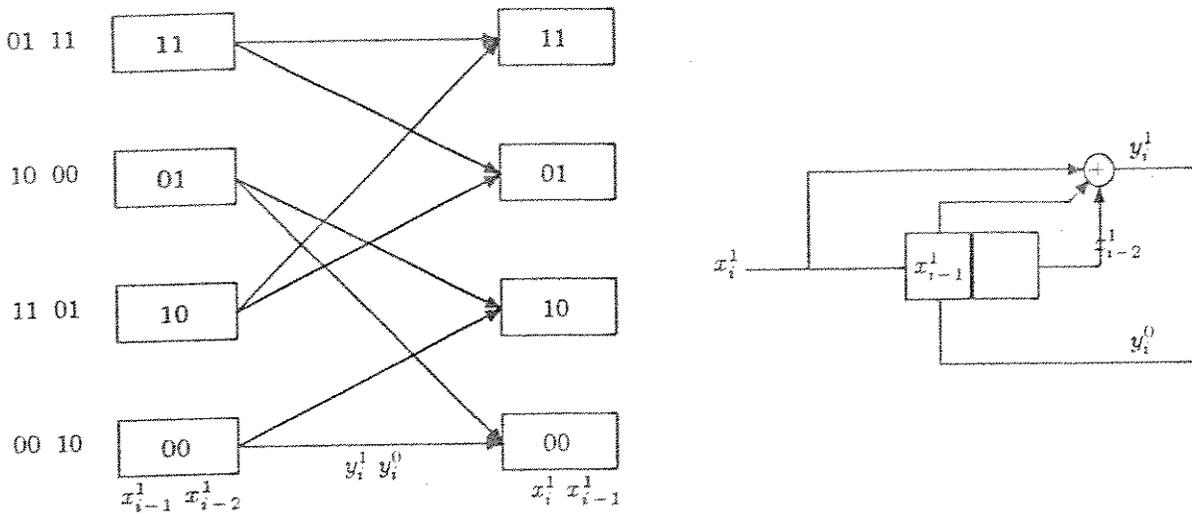


Figura 3.13: Treliça e código binário ótimo com dois elementos de memória.

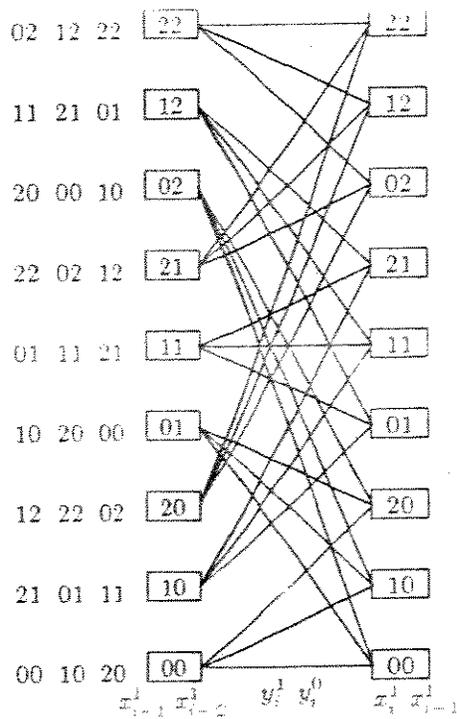


Figura 3.14: Treliça ótima do codificador ternário de dois elementos de memória.

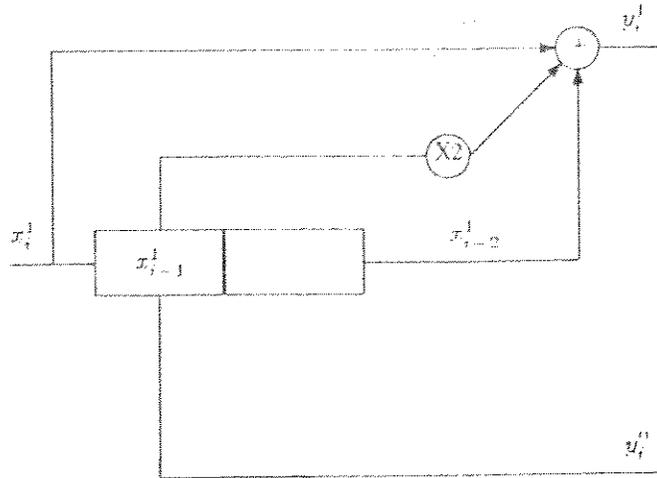


Figura 3.15: Codificador ternário ótimo.

$$\begin{aligned}
 &+ 2(r^2 + 2)(s^2 + 2s)(t^2 + 2) + (r^2 + r)(s^2 + 2s)(t^2 + 2) \\
 &+ 2(r^2 + 2)(s^2 + 2)(t^2 + t) + (r^2 + r)(s^2 + 2)(t^2 + t) \\
 &+ 2(r^2 + r)(s^2 + s)(t^2 + t) + (r^2 + 2r)(s^2 + s)(t^2 + t) \\
 &+ (r^2 + 2)(s^2 + 2s)(t^2 + t) + 2(r^2 + 2r)(s^2 + 2s)(t^2 + t) \\
 &+ (r^2 + 2)(s^2 + 2)(t^2 + 2t) + 2(r^2 + 2r)(s^2 + 2)(t^2 + 2t) \\
 &+ 2(r^2 + 2)(s^2 + s)(t^2 + 2t) + (r^2 + 2r)(s^2 + s)(t^2 + 2t) \\
 &+ 2(r^2 + r)(s^2 + 2s)(t^2 + 2t) + (r^2 + 2r)(s^2 + 2s)(t^2 + 2t) \\
 &= (2r + s)(t^2 + 2) + (2 + 2r + s)(t^2 + t) + (1 + 2r + s)(t^2 + 2t) \\
 &= x_i^1 + 2x_{i-1}^1 + x_{i-2}^1
 \end{aligned}$$

onde por facilidade de notação usamos  $r = x_i^1$ ,  $s = x_{i-1}^1$  e  $t = x_{i-2}^1$ .

A saída  $y_i^0$  pode ser determinada facilmente por inspeção e conclui-se que:

$$y_i^0 = x_{i-1}^1$$

O codificador que realiza esta função encontra-se na figura 3.15, e a matriz geradora deste código é  $G = [1 \ 0 \ 2 \ 1 \ 1 \ 0]$ .

#### Códigos Q-ários

A partir de  $Q = 3$ , a procura de códigos ótimos foi realizada por meios computacionais.

	Q=2	Q=3	Q=5
$\frac{d_{free}^2}{\Delta^2}$	4+1+4= 9	9+1+1+9= 20	25+4+1+1+4+25= 60
$\mathcal{G}$	3,52 dB	3,47 dB	3,80 dB
Matriz	$\begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 1 & 2 & 1 & 0 \\ 1 & 0 & 1 & 2 & 2 & 0 \\ 1 & 0 & 2 & 1 & 1 & 0 \\ 1 & 0 & 2 & 1 & 2 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 2 & 2 & 4 & 0 \\ 1 & 0 & 3 & 3 & 4 & 0 \end{bmatrix}$
	Q=7	Q=9	Q=11
$\frac{d_{free}^2}{\Delta^2}$	122	216	328
$\mathcal{G}$	3,96 dB	4,26 dB	4,33 dB
Matriz	$\begin{bmatrix} 1 & 0 & 1 & 5 & 4 & 0 \\ 1 & 0 & 2 & 3 & 2 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 1 & 6 & 2 & 0 \\ 1 & 0 & 4 & 6 & 5 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 5 & 7 & 3 & 0 \end{bmatrix}$

Tabela 3.2: A distância livre, o ganho e a matriz geradora dos códigos de dois elementos de memória.

A tabela 3.2 fornece para alguns valores de  $Q$ , o valor do  $d_{free}^2$ , o ganho ótimo e alguns dos inúmeros códigos ótimos, obtidos por meios computacionais.

Passando de um para dois elementos de memória ganha-se assintoticamente não mais que 1 dB. Ao contrário de códigos com uma memória, aqui parece não haver uma lei de formação conforme  $Q$  aumenta, o que dificulta a obtenção de códigos ótimos à medida em que o número de elementos aumenta, devido à procura de códigos ótimos ser exaustiva (existem  $Q^6$  códigos possíveis), e devido ao fato destes códigos não apresentarem a propriedade da uniformidade de erro. Neste caso, para que se obtenha a distância livre, vários caminhos pela treliça devem ser percorridos.

### 3.3.3 Códigos de treliça lineares com tres elementos de memória

#### Introdução

A representação para este tipo de código encontra-se na figura 3.16, onde  $y_i^0$  e  $y_i^1$  são funções do tipo:

$$\begin{aligned} y_i^1 &= k_1 x_i^1 + k_3 x_{i-1}^1 + k_5 x_{i-2}^1 + k_7 x_{i-3}^1 \\ y_i^0 &= k_2 x_i^1 + k_4 x_{i-1}^1 + k_6 x_{i-2}^1 + k_8 x_{i-3}^1 \end{aligned}$$

A matriz geradora para este tipo de código é  $G = [k_1 \ k_2 \ k_3 \ k_4 \ k_5 \ k_6 \ k_7 \ k_8]$ . Usou-se 3 elementos de memória em cascata ao invés de dividi-los em entradas diferentes, pelo mesmo motivo já mencionado. A treliça, neste caso, passa a ter  $Q^3$  estados diferentes.

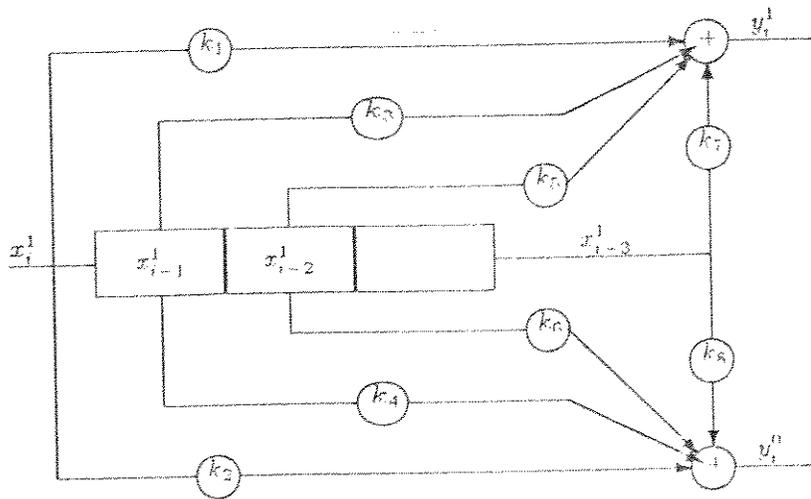


Figura 3.16: Estrutura do codificador binário com 3 elementos de memória.

	Q=2	Q=3	Q=5
$d_{free}^2$	10	22	68
$\frac{d_{free}^2}{\Delta^2}$	3,98 dB	3,88 dB	4,35 dB
Matriz	$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 2 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 2 & 2 & 0 \\ 1 & 0 & 0 & 0 & 2 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 2 & 1 & 2 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 1 & 2 & 0 & 3 & 0 \\ 1 & 0 & 0 & 1 & 3 & 2 & 1 & 0 \\ 1 & 0 & 0 & 1 & 3 & 2 & 4 & 0 \\ 1 & 0 & 2 & 1 & 2 & 0 & & \end{bmatrix}$

Tabela 3.3: A distância livre, o ganho e a matriz geradora dos códigos de tres elementos de memória.

### Códigos $Q$ ários

Obteve-se por meios computacionais a tabela 3.3, que fornece o  $d_{free}^2$ , o ganho ótimo e alguns dos inúmeros códigos ótimos em função do valor de  $Q$ .

### 3.3.4 Códigos de treliça lineares com quatro elementos de memória

#### Códigos $Q$ ários

Obteve-se por meios computacionais a tabela 3.4, que fornece o  $d_{free}^2$ , o ganho ótimo e alguns dos inúmeros códigos ótimos em função de  $Q$ .

	Q=2	Q=3
$\frac{d_{free}^2}{\Delta^2}$	11	24
G	4,39 dB	4,26 dB
Matriz	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 2 & 2 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$

Tabela 3.4: A distância livre, o ganho e a matriz geradora dos códigos ótimos de quatro elementos de memória.

	Q=2	Q=3
$\frac{d_{free}^2}{\Delta^2}$	13	28
G	5,12 dB	4,93 dB
Matriz	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 2 & 2 & 1 & 1 & 0 & 2 & 0 \end{bmatrix}$

Tabela 3.5: A distância livre, o ganho e a matriz geradora dos códigos ótimos de cinco elementos de memória.

### 3.3.5 Códigos de treliça lineares com cinco elementos de memória

Códigos Q-ários

Obteve-se por meios computacionais a tabela 3.5, que fornece o  $d_{free}^2$ , o ganho ótimo e alguns dos inúmeros códigos ótimos em função de Q.

## 3.4 CAPACIDADE DE CANAL

Como mencionado anteriormente, a decodificação do esquema de modulação conjugada à codificação pressupõe o uso de decodificação por decisão do tipo suave (soft decision decoding).

Para cálculo da capacidade de canal neste caso, deve-se usar a expressão da informação mútua média de um canal sem memória com entrada discreta e com saída contínua, que é dada por [2]:

$$I = \sum_{j=0}^{M-1} \int_{-\infty}^{\infty} p(z|a^j) P(a^j) \log_2 \left\{ \frac{p(z|a^j)}{p(z)} \right\} dz \quad (3.9)$$

A capacidade de canal é definida como a informação transferida entre entrada e saída do mesmo canal, maximizada pelas probabilidades de ocorrência dos símbolos  $a_i$ . Quando os símbolos transmitidos são equiprováveis, a capacidade de canal é numericamente igual a informação mútua. A equação 3.9 aliada à identidade

$$p(z) = \sum_{j=0}^{M-1} p(z|a_j) P(a_j) = \frac{1}{M} \sum_{j=0}^{M-1} p(z|a_j)$$

faz com que obtenhamos

$$C = \frac{1}{M} \sum_{j=0}^{M-1} \int_{-\infty}^{\infty} p(z|a^j) \cdot \log_2 \left\{ \frac{p(z|a^j)}{\sum_{i=0}^{M-1} \frac{1}{M} p(z|a^i)} \right\} dz \quad (3.10)$$

onde  $P(i)$  denota a probabilidade de ocorrência do símbolo  $a_i$ ,  $p(z|a_i)$  representa a função densidade de probabilidade no caso de se ter transmitido o símbolo  $a_i$  e  $M = Q^k$  representa o número de sinais discretos transmitidos. Desdobrando a equação anterior temos:

$$C = \log_2 M - \frac{1}{M} \sum_{j=0}^{M-1} \int_{-\infty}^{\infty} \log_2 \left\{ \sum_{i=0}^{M-1} \frac{1}{M} p(z|a^i) \right\} \cdot p(z|a^j) dz \quad (3.11)$$

Caso usemos o ruído como sendo gaussiano branco e aditivo (AWGN), a função densidade de probabilidade pode ser escrita como:

$$p(z|a^i) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left[-\frac{|z - a^i|^2}{2\sigma^2}\right] \quad (3.12)$$

A relação sinal-ruído é definida como:

$$SNR = \frac{2\mathcal{E}}{N_0} \quad (3.13)$$

Usando a equação 3.11 podemos obter as curvas de capacidade de canal para constelações  $Q$  variáveis do tipo PAM, que se encontram nas figuras 3.17, 3.18 e 3.19.

Observe que estas figuras apresentam a relação sinal-ruído para uma taxa de erro de  $10^{-5}$  obtida através da equação 3.13. Suponha que a transmissão normal é feita com o uso de uma constelação de  $Q^k$  pontos. Usando-se um codificador que expanda a constelação para  $Q^{k+1}$ , tem-se que o ganho de codificação máximo é obtido através da diferença entre  $SNRs$ ; a primeira correspondente à curva de capacidade de  $\log_2 Q^k$  bits/período no ponto de taxa de erro igual a  $10^{-5}$ ; a segunda correspondente à junção entre a curva de capacidade de  $\log_2 Q^{k+1}$  bits/período com a reta paralela ao eixo de  $SNR$  com valor de capacidade igual a  $\log_2 Q$ . Em outras palavras, deseja-se saber com quantos  $dB$  a menos de relação sinal-ruído consegue-se transmitir uma constelação de  $Q^{k+1}$  pontos através de uma taxa de  $\log_2 Q^k$  bits/per., contra uma constelação de  $Q^k$  pontos usando-se a mesma taxa de bits.

A tabela 3.6 apresenta os máximos ganhos em função de  $Q$ . Observe que à medida em que o número  $k$  de entradas do codificador aumenta, temos que o ganho diminui até se estabilizar por volta de 6,5dB. À medida em que  $Q$  aumenta (para um número de entradas  $k = 1$ ) o ganho do codificador diminui levemente estabilizando por volta de 7db. O que se conclui é que o máximo ganho disponível diminui gradativamente à medida em que se aumenta  $Q$ .

No entanto se a comparação for feita para uma mesma quantidade de informação, então o que se verifica é que o ganho aumenta levemente, conforme pode ser confirmado pela diferença de ganho entre as expansões de 4-PAM para

CAPÍTULO 3. GENERALIZAÇÃO DE CÓDIGOS PARA MODULAÇÕES DO TIPO PAM 48

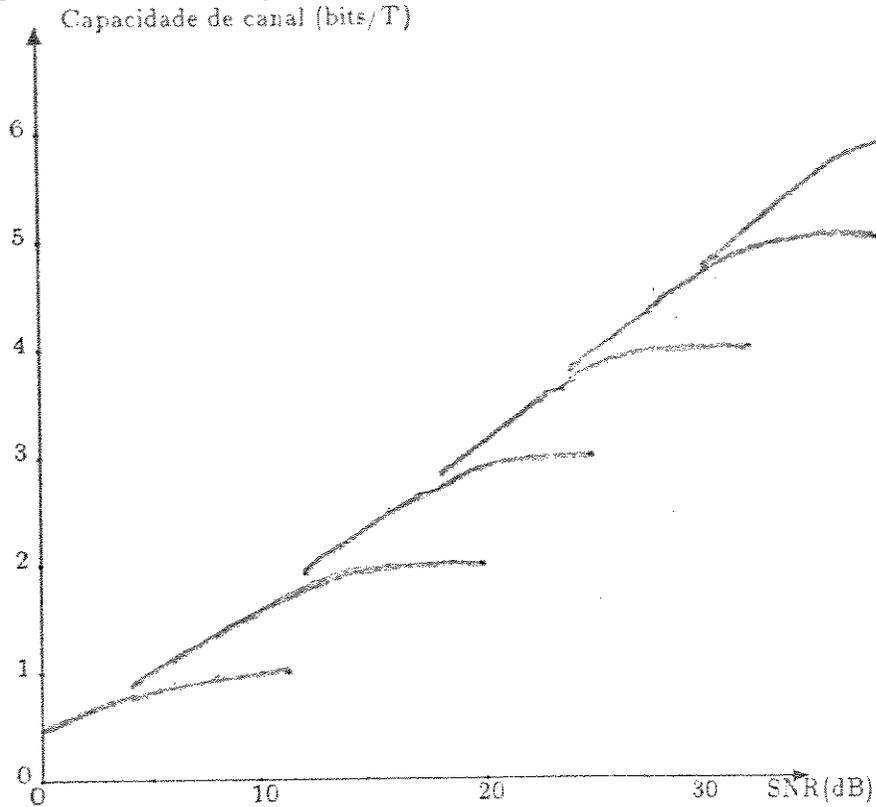


Figura 3.17: Capacidade de canal para constelações binárias

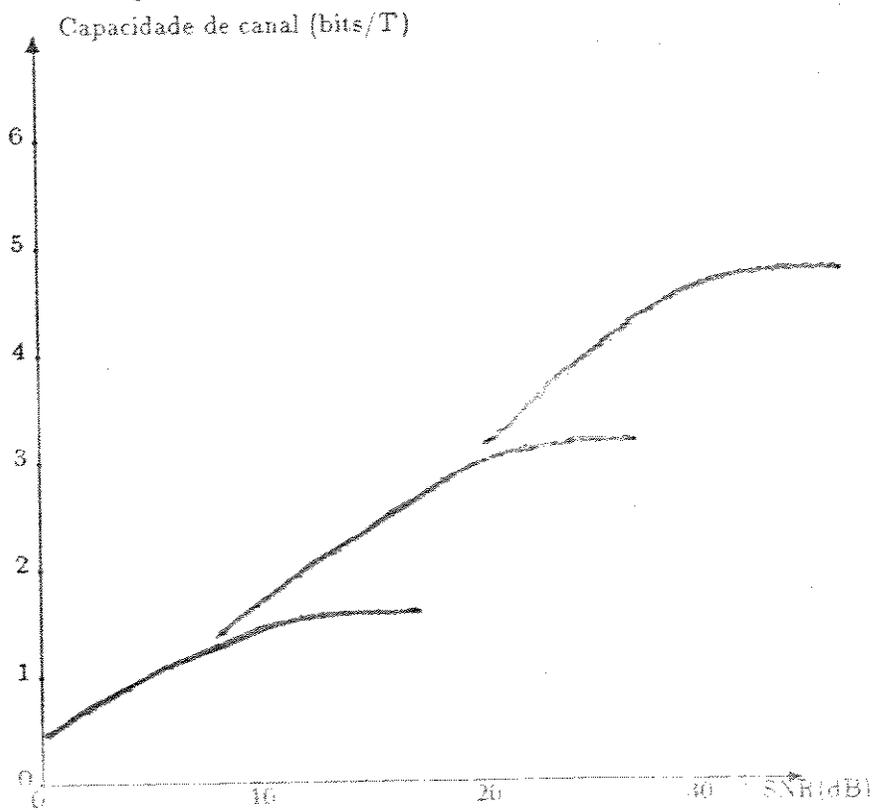


Figura 3.18: Capacidade de canal para constelações ternárias

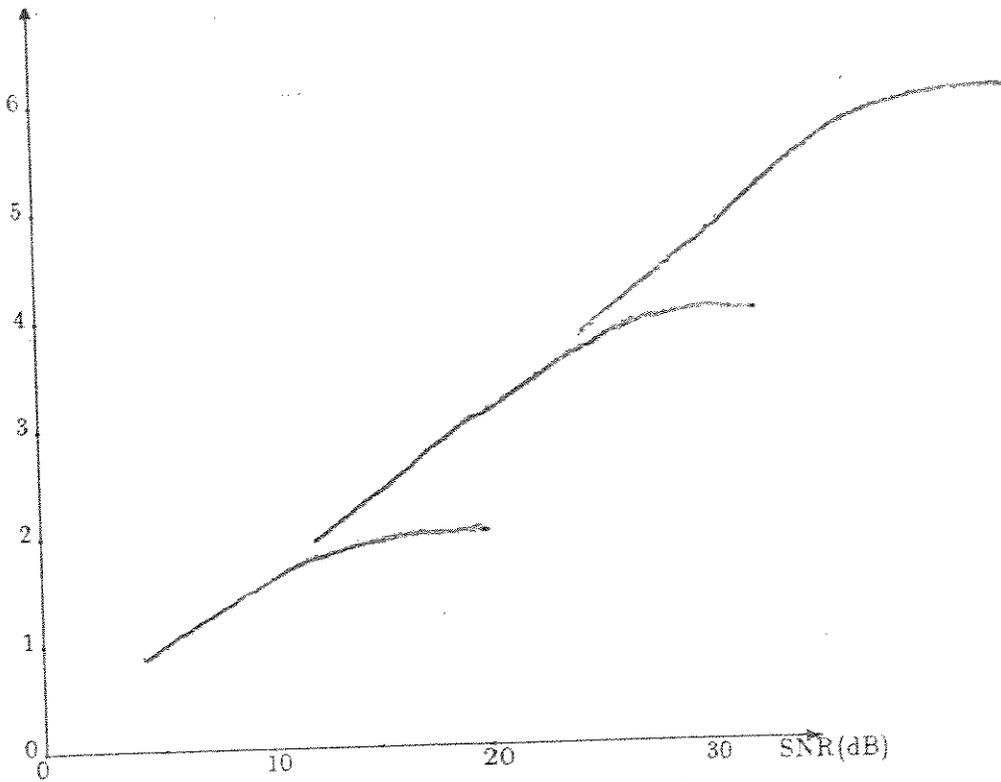


Figura 3.19: Capacidade de canal para constelações quaternárias

Expansão da Constelação	Ganho
2-PAM → 4-PAM	$12,9 - 5,0 = 7,9\text{dB}$
4-PAM → 8-PAM	$19,8 - 12,7 = 7,1\text{dB}$
8-PAM → 16-PAM	$26,1 - 19,2 = 6,9\text{dB}$
16-PAM → 32-PAM	$32,2 - 25,4 = 6,8\text{dB}$
32-PAM → 64-PAM	$38,2 - 31,5 = 6,7\text{dB}$
3-PAM → 9-PAM	$17,1 - 9,6 = 7,5\text{dB}$
9-PAM → 27-PAM	$27,1 - 20,2 = 6,9\text{dB}$
4-PAM → 16-PAM	$19,8 - 12,5 = 7,3\text{dB}$
16-PAM → 64-PAM	$32,2 - 25,4 = 6,8\text{dB}$
5-PAM → 25-PAM	$21,9 - 14,7 = 7,2\text{dB}$
6-PAM → 36-PAM	$23,5 - 16,3 = 7,2\text{dB}$
7-PAM → 49-PAM	$24,9 - 17,9 = 7,0\text{dB}$
8-PAM → 64-PAM	$26,1 - 19,1 = 7,0\text{dB}$

Tabela 3.6: Comportamento do ganho assintótico máximo para o retiniano Z.

8-PAM e de 4-PAM para 16-PAM, quando se obtém 0,2dB. Observe que neste caso trabalhar com base quaternária ao invés de binária, é equivalente a trabalharmos com códigos de razão  $\frac{k}{k+2}$ .

Para o caso PAM a grande vantagem em utilizar códigos  $Q$  *ários* está no fato de se conseguir praticamente todo o ganho disponível com o uso de apenas uma ou duas memórias.

## Capítulo 4

# SOLUÇÃO DA PARTIÇÃO DE CONJUNTOS: FORMAS QUADRÁTICAS BINÁRIAS

### 4.1 INTRODUÇÃO

Desde a proposta por Ungerboeck [1] da combinação das operações de codificação e modulação resultando em esquemas de modulação-codificada eficientes para canais gaussianos aditivos e limitados em faixa, uma quantidade substancial de pesquisa tem sido realizada.

Fundamental para esta forma combinada foi a introdução da técnica de particionamento de conjuntos, isto é, particionamento em sub-conjuntos do conjunto de sinais básicos de modulação. Entretanto esta técnica foi somente utilizada para o caso de partições binárias, isto é, pertencentes ao corpo de Galois com dois elementos  $GF(2)$ .

Vários são os pesquisadores que introduziram e/ou utilizaram de estruturas algébricas na proposição de esquemas eficientes de modulação-codificada. Por exemplo: Calderbank e Sloane [6], propuseram um procedimento alternativo fundamentado em particionamento de reticulados via classes laterais (cosets) dos sub-reticulados através da generalização do codificador onde foram considerados alguns casos esporádicos de partições binárias, ternárias e quaternárias; Forney [8] estendeu a proposta de Calderbank e Sloane para particionamentos de grupos em sub-grupos dando origem ao particionamento em cadeia.

Neste capítulo, estaremos basicamente interessados em introduzir os conceitos de representação de inteiros por formas quadráticas, isto é, equações de Diofanto, como uma proposta de solução alternativa ou generalizada do conceito geral de particionamento de conjunto segundo Ungerboeck. Através destes conceitos, estaremos propiciando um entendimento abrangente do conceito de particionamento de conjunto segundo Ungerboeck, bem como uma maneira sistemática de realizá-lo para os reticulados em uma e duas dimensões (ver [9]).

Uma equação polinomial com coeficientes inteiros é chamada de Diofanto se

as soluções são somente inteiros. Nosso interesse, neste momento, está voltado para o caso especial em que o grau de equação de Diofanto é igual a dois, e para o caso em que a forma quadrática é binária, ou seja, ela apresenta apenas duas variáveis. Dessa maneira, o problema básico passa a ser caracterizado por:

- Dada a forma quadrática binária  $A$  e  $Q$  inteiro, determine se a equação de Diofanto tem solução,

$$A(x, y) = ax^2 + bxy + cy^2 = Q^{2/D} \quad (4.1)$$

com  $D$  representando o número de dimensões ( $D = 1$  ou  $D = 2$ ).

Como veremos, quando  $x$  e  $y$  forem solução de 4.1, então o movimento  $(x, y)$  ao longo da constelação fornecerá uma partição de conjuntos que é ótima, isto é, a distância mínima entre os elementos de um mesmo sub-conjunto é máxima para a ordem do corpo  $Q$  em consideração.

À forma quadrática pode-se em geral, associar o tipo de modulação desejado. Assim, igualando-se a forma quadrática à ordem do corpo (para constelações uni-dimensionais associa-se o quadrado da ordem do corpo), estabelece-se a equação de Diofanto. Dentre o conjunto de soluções desta equação, encontra-se aquela que fornecerá o movimento a ser utilizado na partição de conjuntos básico de sinais de modulação.

Como exemplo de formas quadráticas binárias associadas a reticulados temos:

1. Em 4.1, se  $a = b = 0$  e  $c = 1$  ou  $a = 1$  e  $b = c = 0$  então a forma quadrática está associada ao reticulado  $Z$ ;
2. Em 4.1, se  $a = c = 1$  e  $b = 0$ , então a forma quadrática binária está associada ao reticulado  $Z^2$ ;
3. Em 4.1, se  $a = b = c = 1$ , então a forma quadrática binária está associada ao reticulado  $A_2$  ou hexagonal.

Neste trabalho, o nosso interesse está voltado para reticulados uni-dimensionais  $Z$  usado em modulações PAM e bi-dimensionais, reticulados  $Z^2$  e  $A_2$ , usados em modulação QAM.

## 4.2 MODULAÇÃO QAM: RETICULADO $Z^2$

Antes de entrarmos no mérito das formas quadráticas, vamos definir alguns conceitos que nos serão úteis a seguir.

**Definição 4.1 (Célula Básica)** *Uma Célula Básica de ordem  $Q$  é um arranjo de pontos de  $Q$  linhas e  $Q$  colunas, no caso de um reticulado  $Z^2$ .*

**Definição 4.2 (Quadrado Latino)** *Um Quadrado Latino de ordem  $Q$  é uma Célula Básica de mesma ordem, onde um certo símbolo ocorre  $Q$  vezes, mas não duas vezes na mesma linha ou coluna.*

$$A(x, y) = x^2 + y^2 = Q \quad (4.2)$$

Associando à forma quadrática binária o número de elementos do corpo, tem-se o conjunto de soluções  $(x, y)$ , onde o par  $(x, y)$  assume valores inteiros, e representam o movimento para o particionamento da constelação. Entretanto, a geração deste conjunto de soluções por substituição direta dos valores de  $x$  e  $y$  é um tanto quanto primitiva.

Uma alternativa, matematicamente elegante e precisa de se resolver este problema de representação de inteiros por formas quadráticas binárias foi proposta por Fermat.

Desse modo, todas as possíveis soluções de 4.2 são garantidas através dos seguintes teoremas.

**Teorema 4.1 (Genus)**  $A(x, y) = Q$  pode ser resolvido para  $x$  e  $y$  inteiros e com  $Q$  primo se e somente se,  $Q \equiv 1 \pmod{4}$  ou  $Q = 2$ .

**Teorema 4.2 (Composição)**  $A(x, y)A(x', y') = A(xx' - yy', xy' + x'y)$

A demonstração do teorema 4.1, por ser razoavelmente elaborada e por não contribuir significativamente para o desenvolvimento da tese, não será apresentada.

Com relação ao teorema 4.2, sua demonstração pode ser feita a um nível elementar de verificação de ambos os membros da igualdade ou através da caracterização do par  $(x, y)$  com os números gaussianos, os quais são especificados por

$$G = \{x + iy \mid x, y \in Z\}$$

onde  $i = \sqrt{-1}$  e  $A(x, y)$  passa a ser a norma do número gaussiano. Assim, o teorema da Composição passa a ser interpretado como: o produto das normas de dois números gaussianos é igual a norma do produto destes mesmos dois números.

Dessa forma, dado um inteiro qualquer  $Q$ , através da fatoração de  $Q$  em primos, que atendam ao teorema 4.1, podemos aplicar 4.2, para cada primo e em seguida utilizar o teorema da Composição na determinação de todas as soluções possíveis de 4.2.

**Exemplo 4.1** Como exemplo, considere  $Q = 10$ . Como resultado da fatoração temos 2 e 5. Logo

$$A(x_1, y_1) = x_1^2 + y_1^2 = 2 \quad (4.3)$$

e

$$A(x_2, y_2) = x_2^2 + y_2^2 = 5 \quad (4.4)$$

Usando o teorema 4.1 uma das possíveis soluções de 4.3 e 4.4 são

$$(x_1, y_1) = (1, 1) \quad \text{e} \quad (x_2, y_2) = (2, 1)$$

Como consequência, uma possível solução para

$$A(x, y) = x^2 + y^2 = 10$$

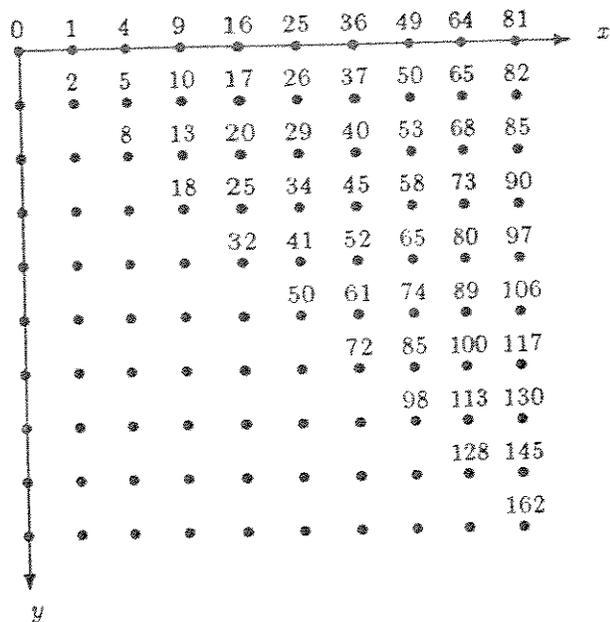


Figura 4.1: Partição correspondente à solução da equação quadrática.

será dada pela aplicação do teorema da Composição, isto é,

$$x = x_1x_2 - y_1y_2 = 1 \times 2 - 1 \times 1 = 1$$

$$y = x_1y_2 + x_2y_1 = 1 \times 1 + 2 \times 1 = 3$$

$$\Rightarrow (x, y) = (1, 3)$$

Como resultado do procedimento da substituição direta de inteiros em 4.2, todas as soluções para  $Q < 100$  da forma quadrática binária associada ao reticulado  $Z^2$  estão na figura 4.1.

Portanto a partição de conjuntos para constelações  $Q$  variáveis em que  $Q$  é solução de  $A(x, y) = Q$ , consiste em obter pontos movendo pelo reticulado segundo o passo  $(\pm x, \pm y)$ , ou  $(\pm y, \pm x)$  conforme ilustra o próximo exemplo.

**Exemplo 4.2** Suponha que queiramos trabalhar com  $Q = 5$  elementos. A Célula Básica neste caso tem  $Q^2 = 25$  elementos. Uma das soluções possíveis conforme pode-se ver na figura 4.1 é o par  $(2, 1)$ . Isto nos diz que deveremos escolher os pontos da Célula Básica segundo o movimento  $(2, 1)$ , isto é, 2 na direção  $x$  e 1 na direção  $y$ , ou vice-versa. Este movimento nos lembra do movimento do cavalo no jogo de xadrez, e podemos ver que a Célula correspondente à figura 4.2 forma um Quadrado Latino. A matriz de transformação ortogonal para este caso vale

$$\begin{vmatrix} 2 & 1 \\ 1 & 3 \end{vmatrix}$$

Para os casos em que:

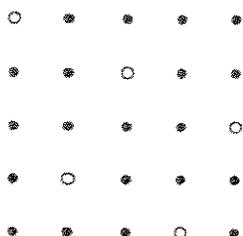


Figura 4.2: Célula Básica para  $Q = 5$ .

- $x = y$  (exceto  $Q = 2$ )
- $x = 0$  ou  $y = 0$

será apresentada mais à frente (ver tabela 4.1) um outro particionamento que apresentará a mesma distância mínima, porém com menor número de vizinhos. Todos as soluções que não satisfazem uma destas duas condições mencionadas anteriormente formam Quadrado Latino.

A tabela 4.1, obtida por meios computacionais, nos mostra quais os melhores movimentos (representados pela matriz de transformação ortogonal) dentro da Célula, de modo a maximizarmos a distância entre elementos do mesmo sub-conjunto, para todo  $Q$ , inclusive para os casos em que não há um Quadrado Latino, ou mesmo quando ele não é solução da equação 4.2.

A construção de constelações com mais de  $Q^2$  pontos, isto é, maior que a Célula Básica, é obtida da junção de Células lado a lado, de modo a se perfazer o total de pontos desejado. Para que a energia média seja minimizada, um rearranjo de pontos se faz necessário, de tal modo que a região ocupada pela constelação se aproxime de um círculo, mas no entanto preservando a posição original de um ponto dentro de uma Célula Básica.

**Exemplo 4.3** *Este exemplo serve para ilustrar a efetividade do esquema de partição  $Q$  arie. Suponho uma constelação de 64 pontos*

*Para  $Q = 8$  a partição de conjuntos encontra-se na figura 4.8. Neste caso temos 8 sub-conjuntos, onde a distância quadrática mínima dentro do sub-conjunto*

$Q$	$d_{min}^2$	matriz transf. ortog.	
2	2	1 1 -1 1	
3	2	2 1 1 2	
4	4	2 0 1 2	
5	5	2 1 1 3	
6	5	2 0 1 3	
7	5	2 1 1 4	3 2 1 3
8	8	3 1 1 3	
9	9	3 0 1 3	
10	10	3 2 1 4	
11	10	3 1 1 4	4 1 1 3
12	13	3 0 2 4	
13	13	5 2 1 3	
14	13	4 2 1 4	
15	17	4 1 1 4	
16	16	4 0 1 4	4 0 2 4
17	17	4 3 1 5	
18	18	5 2 1 4	7 3 1 3
19	17	4 1 1 5	5 1 1 4
20	20	8 4 1 3	

Q	$d_{min}^2$	matriz transf. ortog.			
21	18	6 3 1 4	8 3 1 3		
22	20	5 3 1 5	6 2 1 4	9 5 1 3	
23	25	5 2 1 5	9 4 1 3		
24	26	5 1 1 3			
25	25	5 0 1 5	7 3 1 4	10 5 1 3	
26	26	5 4 1 6			
27	26	5 3 1 6	11 6 1 3		
28	29	6 2 1 5			
29	29	12 7 1 3			
30	34	5 0 3 6			
31	29	12 5 1 3	13 8 1 3		
32	32	7 3 1 5	9 4 1 4		
33	34	6 3 1 6			
34	37	6 2 1 6			
35	37	6 1 1 6			
36	36	6 0 1 6			
37	37	6 5 1 7			
38	40	16 10 1 3			

Tabela 4.1: Movimentos que apresentam maior distância mínima para o reticulado  $Z^3$ .

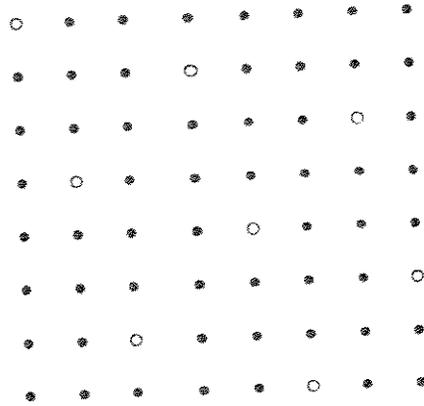


Figura 4.3: Exemplo de partição de uma constelação de 64 pontos com  $Q = 8$  elementos.

é igual a 8. A matriz de transformação ortogonal vale:  $\begin{vmatrix} 3 & 1 \\ 1 & 3 \end{vmatrix}$ .

Para o caso em que  $Q = 4$ , teremos uma Célula Básica de 16 pontos, onde a distância quadrática mínima é de 4 e o número de sub-conjuntos para a primeira partição é de 4. Como a constelação é formada de 4 Células, teremos um total de 16 sub-conjuntos. A matriz de transformação ortogonal neste caso vale:  $\begin{vmatrix} 2 & 0 \\ 1 & 2 \end{vmatrix}$ .

Na segunda partição, o número de sub-conjuntos total cai para 4, e a distância quadrática mínima sobe a 16. As duas partições encontram-se na figura 4.4. A matriz de transformação ortogonal vale:  $\begin{vmatrix} 4 & 0 \\ 2 & 4 \end{vmatrix}$ .

Para o caso em que  $Q = 2$ , teremos uma Célula Básica de 4 pontos, onde a distância quadrática mínima é de 2 e o número de sub-conjuntos para a primeira partição é de 2. Como a constelação é formada de 16 Células, teremos um total de 32 sub-conjuntos. A matriz de transformação ortogonal vale:  $\begin{vmatrix} 1 & -1 \\ 1 & 1 \end{vmatrix}$ .

Na segunda partição, o número de sub-conjuntos total cai para 16, e a distância quadrática mínima sobe a 4. A matriz correspondente vale:  $\begin{vmatrix} 2 & 0 \\ 0 & 2 \end{vmatrix}$ . A terceira partição apresenta 8 sub-conjuntos com distância quadrática mínima de 8. A matriz de transformação ortogonal vale:  $\begin{vmatrix} 2 & -2 \\ 2 & 2 \end{vmatrix}$ . A quarta partição apresenta 4 sub-conjuntos com distância quadrática mínima de 16. A matriz de transformação

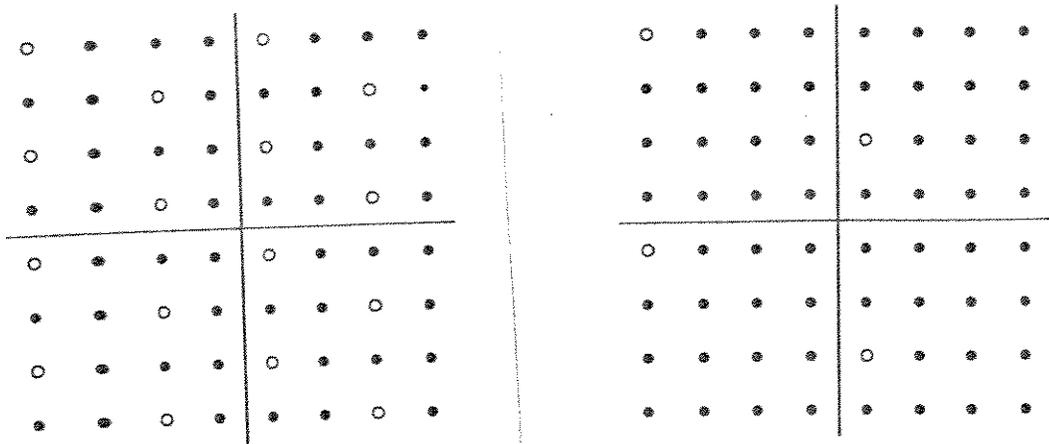


Figura 4.4: Exemplo dos 2 níveis de partição de uma constelação de 64 pontos com  $Q = 4$  elementos.

ortogonal vale:  $\begin{vmatrix} 4 & 0 \\ 0 & 4 \end{vmatrix}$ . A quinta partição apresenta 2 sub-conjuntos com distância mínima de 32. Estas 5 partições encontram-se na figura 4.5. A matriz correspondente neste caso vale:  $\begin{vmatrix} 4 & -4 \\ 4 & 4 \end{vmatrix}$ .

**Exemplo 4.4** Vamos considerar um exemplo de partição em que  $Q$  não é solução da equação 4.2. Suponha  $Q = 6$  e uma constelação de  $6^4 = 1296$  pontos. O melhor movimento para se fazer a primeira partição segundo a tabela 4.1 é  $\begin{vmatrix} 2 & 0 \\ 1 & 3 \end{vmatrix}$ . Neste caso o número de sub-conjuntos é de 216 e a distância quadrática mínima é de 5. A segunda partição apresenta 36 sub-conjuntos com distância quadrática mínima de 36, onde a matriz correspondente a este reticulado é  $\begin{vmatrix} 6 & 0 \\ 3 & 6 \end{vmatrix}$ . A terceira partição tem 6 sub-conjuntos com distância quadrática mínima de 180, e a matriz de transformação ortogonal vale:  $\begin{vmatrix} 12 & 0 \\ 6 & 18 \end{vmatrix}$ . Estas partições estão presentes na figura 4.6.

A conclusão que se tira é que se  $Q$  for solução da forma quadrática binária então o particionamento de uma constelação de  $Q^{k+1}$  pontos apresentará:

- A primeira partição nos fornece  $Q^k$  sub-conjuntos com uma distância quadrática mínima de  $Q$ .

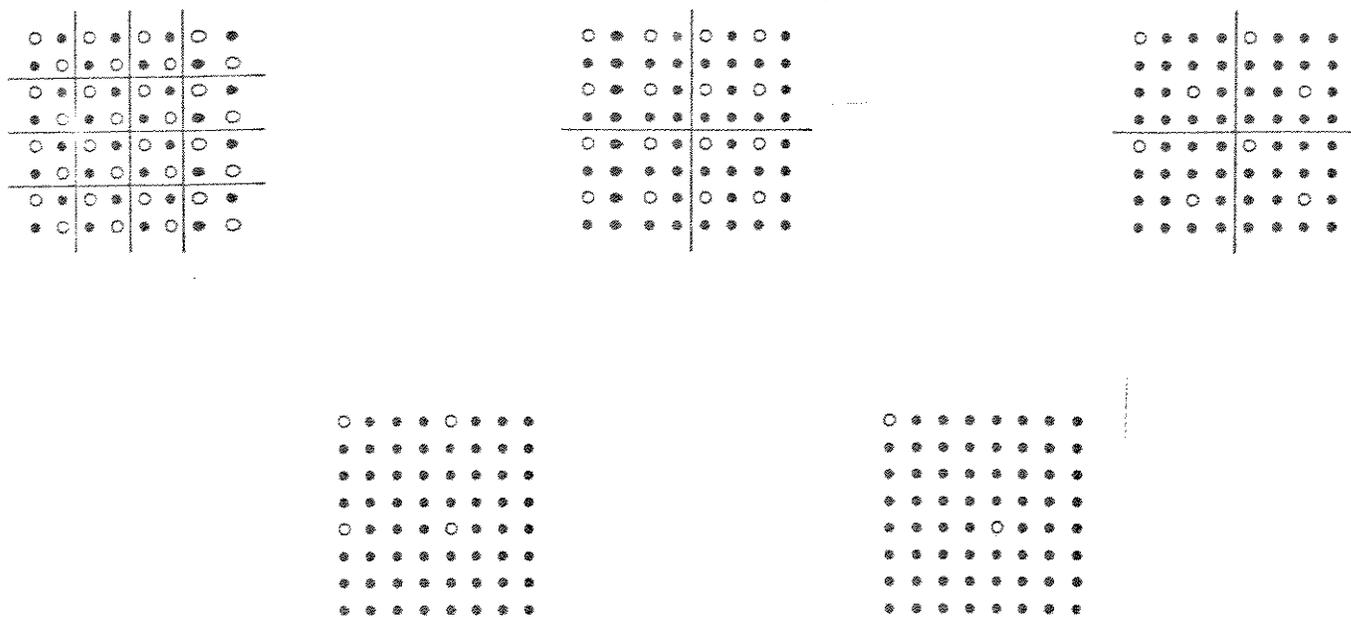


Figura 4.5: Exemplo dos 5 níveis de partição de uma constelação de 64 pontos com  $Q = 2$  elementos.

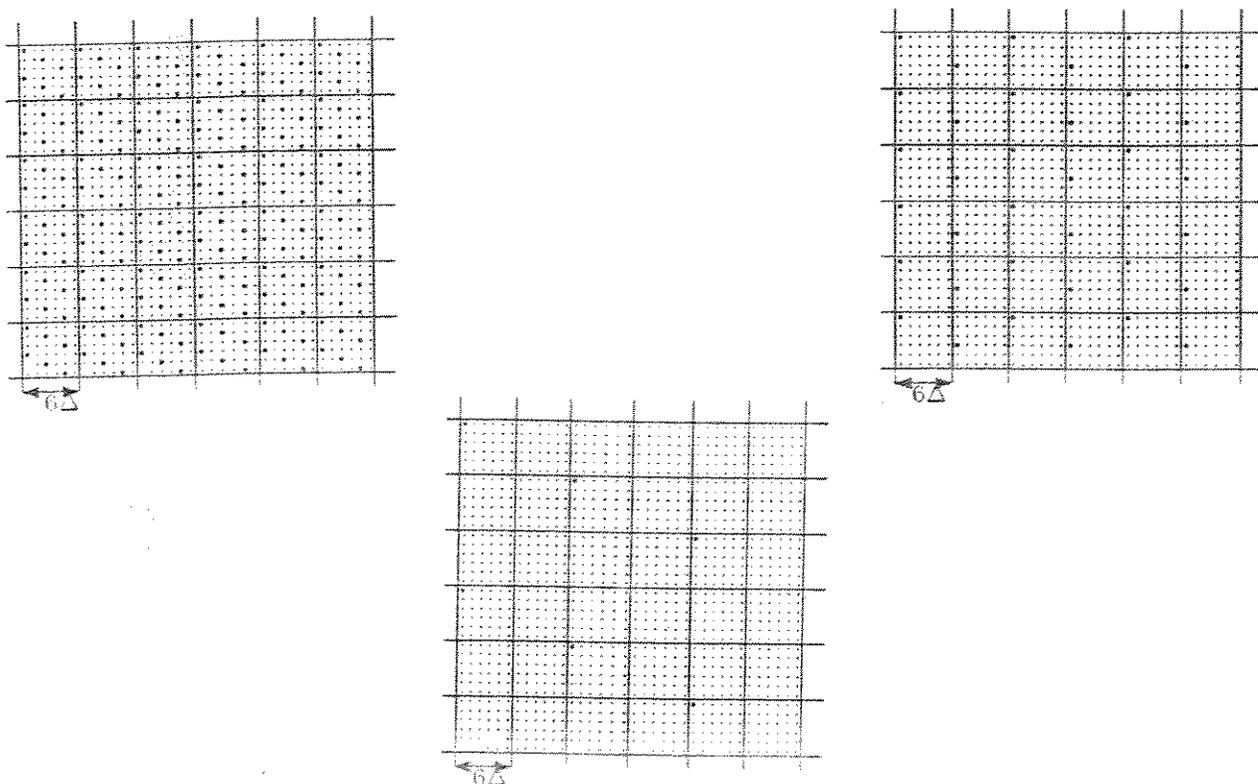


Figura 4.6: Exemplo das 3 partições de uma constelação de 1296 pontos com  $Q = 6$  elementos

- A segunda partição nos fornece  $Q^{k-1}$  sub-conjuntos com uma distância quadrática mínima de  $Q^2$ .
- A  $i$ -ésima partição nos fornece  $Q^{k+1-i}$  sub-conjuntos com uma distância quadrática mínima de  $Q^i$ .

Quando  $Q$  não é solução da forma quadrática binária então uma constelação de  $Q^{k+1}$  pontos apresenta:

- A primeira partição nos fornece  $Q^k$  sub-conjuntos com uma distância quadrática mínima de  $d_{min}^2$ , onde  $d_{min}^2$  é a distância quadrática mínima dentro da Célula Básica, obtida da tabela 4.1
- A segunda partição nos fornece  $Q^{k-1}$  sub-conjuntos com uma distância quadrática mínima de  $Q^2$ .
- A  $i$ -ésima partição, nos fornece  $Q^{k+1-i}$  sub-conjuntos com uma distância quadrática mínima de  $d_{min}^2 Q^{i-1}$  quando  $i$  é ímpar, ou de  $Q^i$  para  $i$  par.

### 4.3 MODULAÇÃO QAM: RETICULADO $A_2$

Antes de entrarmos no mérito das formas quadráticas, definiremos alguns conceitos que nos serão úteis.

**Definição 4.3 (Célula Básica Inclinada)** *Uma Célula Básica Inclinada de ordem  $Q$  é um arranjo de pontos de  $Q$  linhas e  $Q$  colunas inclinadas, no caso do reticulado  $A_2$ .*

A equação 4.5 fornece as soluções para o reticulado  $A_2$

$$A(x, y) = x^2 + y^2 + xy \tag{4.5}$$

Todas as soluções possíveis de  $A(x, y) = Q$  são obtidas através dos seguintes teoremas:

**Teorema 4.3**  *$A(x, y) = Q$  pode ser resolvido para  $x$  e  $y$  inteiros para  $Q$  primo se e somente se.  $Q = 1 \pmod{6}$  ou  $Q = 3$ .*

**Teorema 4.4**  $A(x, y)A(x', y') = A(xx' - yy', xy' + x'y + yy')$

As soluções destes teoremas encontram-se na figura 4.7. A partição para uma constelação  $Q$ -ária, quando  $Q$  é solução da equação 4.5, consiste em se movimentar de  $(x, y)$  através do reticulado conforme ilustra o próximo exemplo.

**Exemplo 4.5** *Suponha que queiramos trabalhar com  $Q = 7$  elementos. A Célula Básica Inclinada neste caso tem  $Q^2 = 49$  elementos. Uma das soluções possíveis conforme pode-se ver na figura 4.7 é o par  $(2, 1)$ . Isto nos diz que deveremos escolher os pontos da Célula Básica Inclinada segundo o movimento  $(2, 1)$ , isto é, 2 na direção  $x$  e 1 na direção  $y$ , ou vice-versa. A matriz de transformação*

*ortogonal correspondente vale:*

$$\begin{pmatrix} 2 & 1 \\ 1 & 4 \end{pmatrix}$$

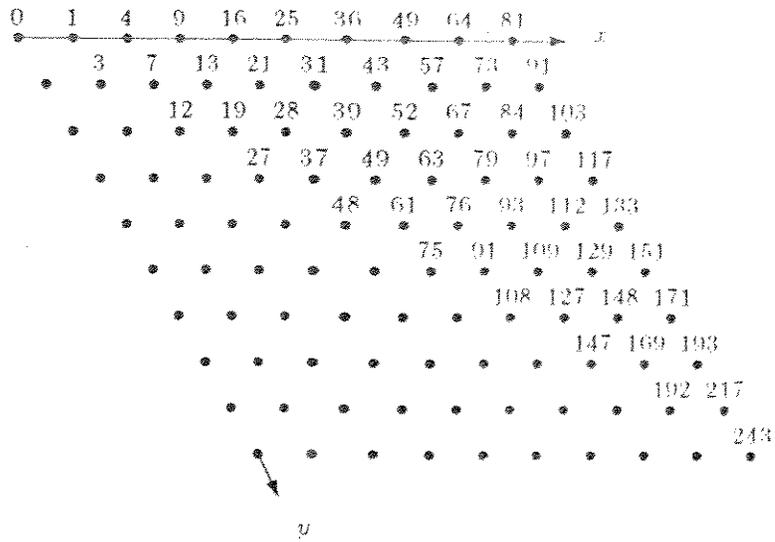


Figura 4.7: Partição correspondente à solução da equação quadrática.

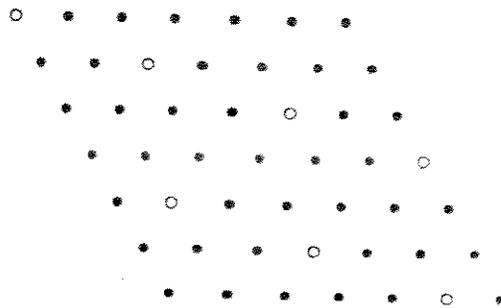


Figura 4.8: Célula Básica Inclinada para  $Q = 7$  no reticulado  $A_2$ .

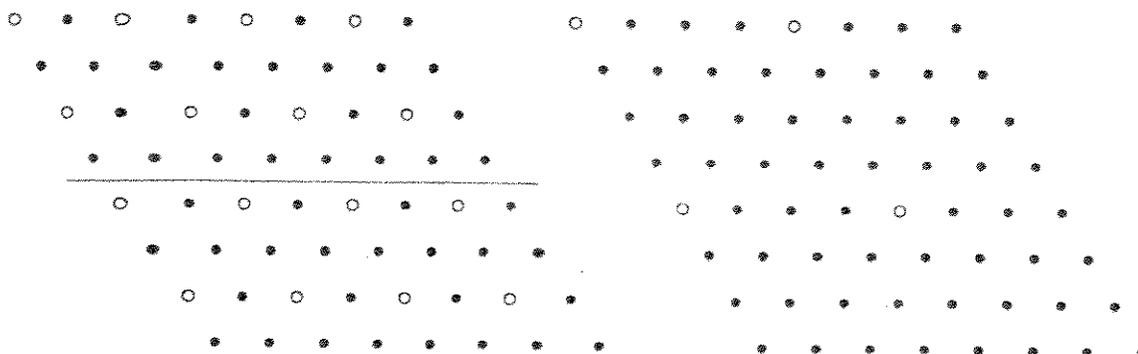


Figura 4.9: Exemplo de 2 níveis de partição de uma constelação de 64 pontos com  $Q = 4$  elementos.

Aqui, ao contrário do reticulado  $Z^2$ , a solução da forma quadrática nos diz sempre como deveremos nos mover dentro da Célula. A tabela 4.2 nos mostra quais os melhores movimentos dentro da Célula, de modo a maximizarmos a distância entre elementos do mesmo sub-conjunto, para todo  $Q$ , mesmo quando ele não é solução da equação 4.5.

A construção de constelações com mais de  $Q^2$  pontos, isto é, maior que a Célula Básica, é obtida da junção de Células lado a lado, de modo a se perfazer o total de pontos desejado. Para que a energia média seja minimizada, um rearranjo de pontos se faz necessário, de tal modo que a constelação se aproxime de um círculo, preservando a posição original de um ponto dentro de uma Célula Básica.

**Exemplo 4.6** Este exemplo serve para ilustrar a efetividade do esquema de partição  $Q$ -ário. Suponha uma constelação de 64 pontos. Para o caso em que  $Q = 4$ , teremos uma Célula Básica de 16 pontos, onde a distância quadrática mínima é de 4 e o número de sub-conjuntos para a primeira partição é de 4. Como a constelação é formada de 4 Células, teremos um total de 16 sub-conjuntos. A matriz usada é  $\begin{vmatrix} 2 & 0 \\ 0 & 2 \end{vmatrix}$ . Na segunda partição, o número de sub-conjuntos total cai para 4, e a distância quadrática mínima sobe a 16, usando-se a matriz  $\begin{vmatrix} 4 & 0 \\ 0 & 4 \end{vmatrix}$ . As duas partições encontram-se na figura 4.9.

**Exemplo 4.7** Consideraremos agora um exemplo de partição em que  $Q$  não é solução da equação 4.5. Suponha  $Q = 6$  e uma constelação de  $6^4 = 1296$  pontos.

CAPÍTULO 4. SOLUÇÃO DA PARTIÇÃO DE CONJUNTOS: FORMAS QUADRÁTICAS BINÁRIAS<sup>64</sup>

Q	$d_{min}^2$	matriz de transformação ortogonal			
2	1	1	-1		
		1	1		
3	3	1	-1		
		1	2		
4	4	2	0		
		0	2		
5	3	2	1		
		1	3		
6	4	2	0		
		1	3		
7	7	2	1		
		1	4		
8	7	2	0		
		1	4		
9	9	3	0		
		0	3		
10	7	2	0	3	2
		1	5	1	4
11	7	2	1	3	1
		1	6	1	4
				4	1
				1	3
12	12	2	-4		
		2	2		
13	13	3	2		
		1	5		
14	12	3	1	5	1
		1	5	1	3
15	13	3	0		
		1	5		
16	16	4	0		
		0	4		
17	13	3	1	4	3
		1	6	1	5
				6	1
				1	3
18	13	3	0		
		1	6		
19	19	7	2		
		1	3		
20	16	4	0	5	0
		1	5	1	4

Q	$d_{\min}^c$	matriz de transformação ortogonal			
21	21	4 3 1 6			
22	19	4 2 1 6	8 2 1 3		
23	19	4 1 1 6	6 1 1 4	9 4 1 3	
24	21	4 0 1 6			
25	25	5 0 0 5			
26	21	4 0 1 6	5 4 1 6		
27	27	3 -6 3 3			
28	21	4 0 1 7	11 5 1 3		
29	21	4 3 1 8	5 1 1 6	6 1 1 5	
30	27	8 2 1 4			
31	31	5 4 1 7			
32	28	5 3 1 7	13 7 1 3		
37	37	10 3 1 4			
41	37	11 3 1 4	15 4 1 3	17 10 1 3	

Tabela 4.2: Movimentos que apresentam maior distância mínima para o reticulado  $A_2$ .

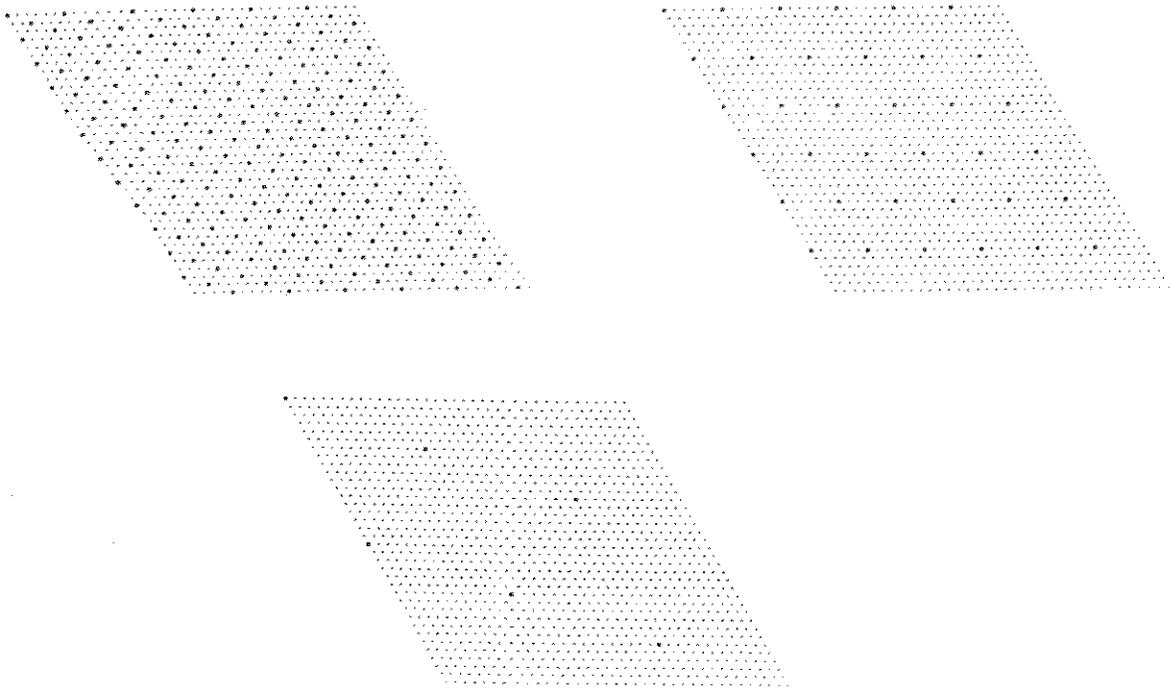


Figura 4.10: Exemplo dos 3 níveis de partição de constelação de 1296 pontos com  $Q = 6$ .

O melhor movimento para se fazer a primeira partição segundo a tabela 4.2 é:  $\begin{vmatrix} 2 & 0 \\ 1 & 3 \end{vmatrix}$ . Neste caso o número de sub-conjuntos é de 216 e a distância quadrática mínima é de 4. A segunda partição apresenta 36 sub-conjuntos com distância quadrática mínima de 36. A melhor partição é obtida usando  $\begin{vmatrix} 6 & 0 \\ 3 & 6 \end{vmatrix}$ . A terceira partição tem 6 sub-conjuntos com distância quadrática mínima de 144 usando  $\begin{vmatrix} 12 & 0 \\ 6 & 18 \end{vmatrix}$ . Estas partições estão presentes na figura 4.10

A conclusão que se tira é que se  $Q$  for solução da forma quadrática binária então uma constelação de  $Q^{k+1}$  pontos apresenta:

- A primeira partição nos fornece  $Q^k$  sub-conjuntos com uma distância quadrática mínima de  $Q$ .
- A segunda partição nos fornece  $Q^{k-1}$  sub-conjuntos com uma distância quadrática mínima de  $Q^2$ .
- A  $i$ ésima partição nos fornece  $Q^{k+1-i}$  sub-conjuntos com uma distância quadrática mínima de  $Q^i$ .

E se  $Q$  não for solução da forma quadrática binária então uma constelação de  $Q^{k+1}$  pontos apresenta:

- A primeira partição nos fornece  $Q^k$  sub-conjuntos com uma distância quadrática mínima de  $d_{min}^2$ , onde  $d_{min}^2$  é a distância mínima dentro da Célula Básica,

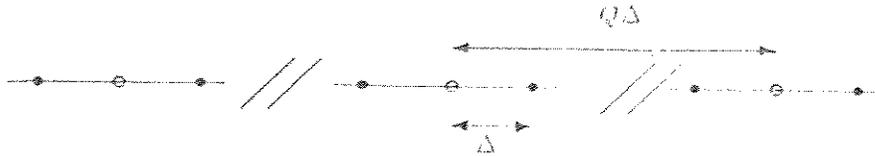


Figura 4.11: Partição de uma constelação PAM genérica

obtida da tabela 4.2.

- A segunda partição nos fornece  $Q^k$  sub-conjuntos com uma distância quadrática mínima de  $Q^2$ .
- A  $i$ ésima partição, nos fornece  $Q^{k+1-i}$  sub-conjuntos com uma distância quadrática mínima de  $d_{min}^2 Q^{i-1}$  quando  $i$  é ímpar, ou de  $Q^i$  para  $i$  par.

#### 4.4 MODULAÇÃO PAM

Fazendo  $a = 1$  e  $b = c = 0$  na equação 4.2 e igualando a  $Q^2$ , temos que

$$x = Q$$

isto é, para fazer a partição, basta andarmos na constelação de  $Q$  unidades conforme a figura 4.11, como foi feito no capítulo 3 de maneira intuitiva.

Como já foi dito no capítulo anterior, o particionamento para sinais PAM é um caso particular de uma concepção mais geral.

#### 4.5 APLICAÇÃO EM TELEFONIA CELULAR

As secções que generalizam o problema da partição no plano, mais especificamente nos reticulados  $Z^2$  e  $A_2$ , tem aplicação no problema de alocação de frequências da telefonia móvel (ver [5]).

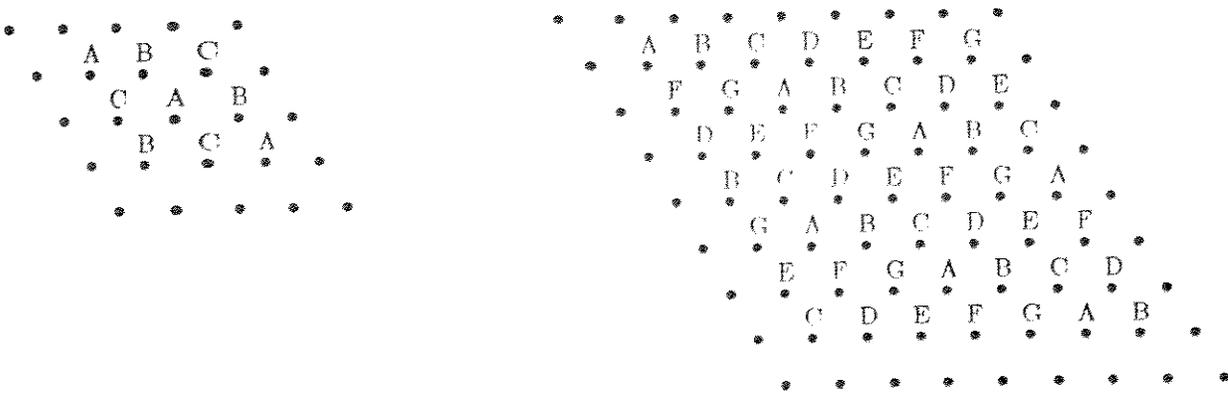


Figura 4.12: Exemplo de alocação de frequências em telefonia celular.

Suponha que o mapa de centrais em um sistema de telefonia celular obedeça a um reticulado  $A_2$ . As soluções da tabela 4.2 nos mostram como alocar as centrais, para que uma dada separação entre elas seja satisfeita.

Suponha inicialmente o seguinte problema: Queremos ter centrais separadas de no mínimo  $\sqrt{3}\Delta$ , onde  $\Delta$  é o menor espaçamento no reticulado  $A_2$ . A solução está em consultar a tabela 4.2 e verificar que a solução consiste em usarmos 3 frequências diferentes (5 frequências diferentes também apresentariam a mesma distância). Caso desejemos uma distância de  $\sqrt{7}\Delta$ , então deveremos usar 7 frequências diferentes (vide figura 4.12), e assim por diante.

Caso a questão seja a seguinte: dada um número de frequências diferentes, qual a menor distância a se conseguir entre duas centrais de mesma frequência? A resposta está em consultar a tabela 4.2 para o reticulado  $A_2$ . Se quisermos usar 13 frequências diferentes, então a menor distância será de  $\sqrt{13}\Delta$  e o melhor

movimento é  $\begin{vmatrix} 3 & 2 \\ 1 & 5 \end{vmatrix}$ .

## Capítulo 5

# GENERALIZAÇÃO DA MODULAÇÃO-CODIFICADA QAM: RETICULADOS $Z^2$ E $A_2$

### 5.1 INTRODUÇÃO

No capítulo 2 analisamos um único caso de modulação- codificada binária usando codificador de taxa  $R = \frac{2}{3}$  e uma memória, seguida de uma comparação com o caso não codificado. Esta análise comparativa teve como objetivo primeiro a introdução do procedimento a ser utilizado.

Este capítulo pretende generalizar o trabalho de Ungerboeck [1], obtendo esquemas de modulações por codificação em treliça  $Q$  *arias*, variando o número de memórias, baseado nos esquemas de partição propostos no capítulo 4 (ver [10]).

Gostaríamos de salientar que o desempenho de esquemas de modulação por codificação em treliça baseados no reticulado  $A_2$  para o caso binário não foi analisado por [1], uma vez que a partição não propicia um aumento da distância intra-conjunto, conforme pode ser constatado na tabela 4.2 para  $Q = 2$ .

Como veremos o ganho obtido em relação a esquemas de não codificação, será o maior possível, quando  $Q$  for primo, pois neste caso as operações de adição e multiplicação  $mod(Q)$  resultam em um corpo de Galois com  $Q$  elementos, o que não ocorre quando  $Q$  não é primo. Em particular a operação multiplicação  $mod(Q)$ , neste caso, não gera todos os elementos. Sendo assim, o conjunto com  $Q$  elementos munido das operações adição e multiplicação  $mod(Q)$ , para  $Q$  não primo, não forma um corpo. Para a obtenção de um corpo, as operações adição e multiplicação deverão seguir uma regra especificada através da montagem de tabela padrão.

Assim sendo, em geral, os códigos lineares obtidos para  $Q$  não-primo serão inferiores, em termos de desempenho, aos códigos obtidos pelo primo mais próximo. Entretanto, existem códigos não-lineares que apresentam melhor desempenho que os lineares, para  $Q$  não primo, porém a procura destes códigos começa a ser inviável a partir de  $Q = 10$  devido à grande quantidade de códigos possíveis, mesmo com

o uso de apenas uma memória.

## 5.2 CÓDIGOS ÓTIMOS PARA O RETICULADO $Z^2$

### 5.2.1 Introdução

Para o caso  $Z^2$  consideraremos códigos de treliça lineares com uma memória, onde iremos determinar os códigos ótimos via regra generalizada de mintermos para alguns valores de  $Q$  e via busca exaustiva para os demais casos, bem como calcular o seu desempenho. Analisaremos um caso de código de treliça não-linear, para  $Q$  não primo, de modo a enfatizar a sua existência, calculando o seu desempenho. Consideraremos em seguida os códigos de treliça lineares com dois elementos de memória.

### 5.2.2 Códigos de treliça lineares com um elemento de memória

#### Introdução

O esquema da modulação-codificada usado é o mesmo da figura 3.1. A teoria do particionamento de conjuntos usada é aquela descrita no capítulo 4, bem como o codificador generalizado é o da figura 3.3.

Como no capítulo 3 nosso interesse é trabalhar com códigos de razão  $R = \frac{k}{k+1}$  onde  $k \geq 2$ , que pressupõe a utilização de pelo menos uma entrada não codificada.

A utilização de entradas não codificadas, acarretará em transições paralelas dentro da treliça, que além de limitar a distância livre em  $Q^2 \Delta^2$ , vai proporcionar uma distribuição de pontos mais uniforme e deverá apresentar portanto, uma distância livre menor que no caso em que não há transições paralelas ( $k = 1$ ).

Chamando o número total de entradas de  $k$ , o número de entradas não codificadas de  $k - \bar{k}$ , e o número de saídas de  $k + 1$ , a partição generalizada apresentará  $Q^{k+1}$  elementos no total, com  $Q^{\bar{k}+1}$  sub-conjuntos com  $Q^{k-\bar{k}}$  elementos por subconjunto.

Na determinação do codificador convolucional ótimo para uma dada taxa  $R = \frac{k}{k+1}$  e com um elemento de memória, estaremos usando como critério de otimalidade o quadrado da distância livre Euclidiana. Constelações QAM dão origem a códigos que apresentam a propriedade da uniformidade de erro, tal que para a obtenção da distância livre podemos usar a sequência nula como sequência de referência.

Assim sendo, a partir de agora vamos considerar os casos  $Q$  ímpios.

#### Códigos Binários

A partição de conjuntos encontra-se na figura 5.1, e a constelação de pontos na figura 5.2.

Os símbolos associados aos ramos que saem de um estado, devem ser escolhidos tais que a distância entre si seja a maior possível. Os elementos que

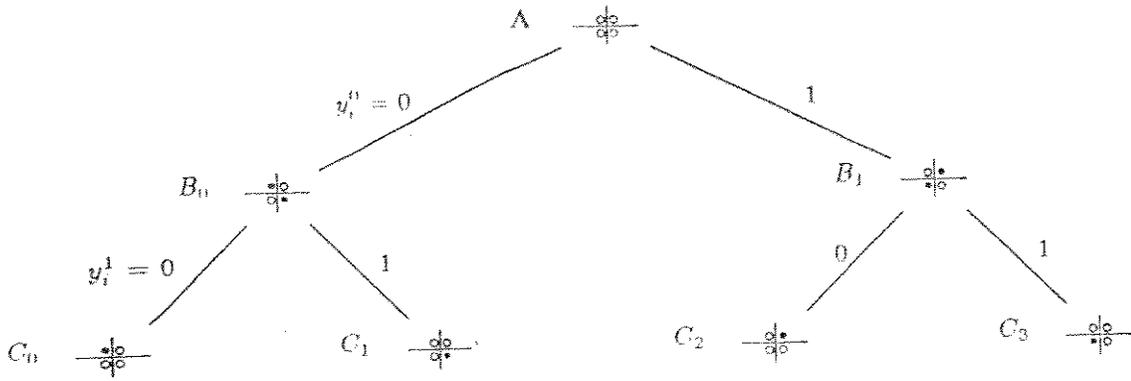


Figura 5.1: Partição para o caso binário.

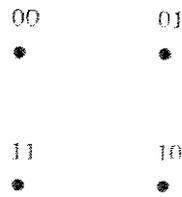


Figura 5.2: Constelação de sinais para o caso binário.

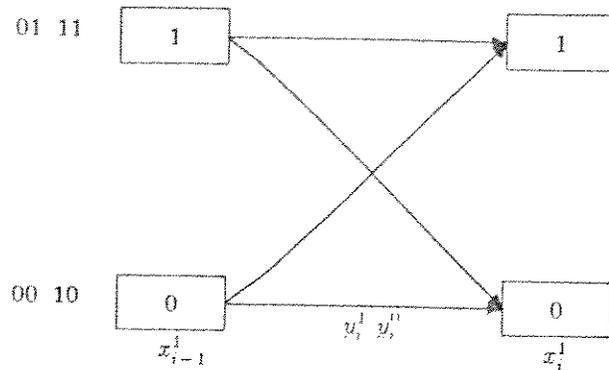


Figura 5.3: Treliça ótima do codificador binário com uma memória.

satisfazem esta condição são os elementos 00 e 10 ou 01 e 11, pontos que distam entre si quadraticamente de  $2\Delta^2$ . Os símbolos associados aos ramos que chegam a um estado são pontos distantes entre si de  $\Delta^2$ , tais como os pontos 00 e 01 ou 00 e 11 ou ainda 10 e 01 ou 10 e 11.

Uma das várias possíveis alocações dos símbolos aos ramos da treliça está na figura 5.3, donde se tira que o  $d_{free}^2 = (2 + 1)\Delta^2 = 3\Delta^2$ , e o código que satisfaz esta alocação pode ser obtido através da regra de mintermos:

$$\begin{aligned} y_i^1 &= (x_{i-1}^1 + 1)x_i^1 + x_{i-1}^1 x_i^1 \\ &= x_i^1 \\ y_i^0 &= x_{i-1}^1(x_i^1 + 1) + x_{i-1}^1 x_i^1 \\ &= x_{i-1}^1 \end{aligned}$$

onde o díbit é representado na figura 5.2 por  $y_i^1$  e  $y_i^0$ , e estes são calculados através da soma de produtos de  $x_{i-1}^1$  e  $x_i^1$  onde  $x_i^1$  é o bit de entrada presente e  $x_{i-1}^1$  é o bit futuro. O codificador que executa a função calculada anteriormente encontra-se na figura 5.4, e o gerador que descreve este código é  $[1\ 0\ 0\ 1]$ .

Como já foi dito anteriormente a ocorrência de entradas não codificadas, implicará em transições paralelas aos ramos da treliça. A distância entre transições paralelas é de  $2^2\Delta^2 = 4\Delta^2$  que é maior que a distância livre obtida neste caso.

A energia média de um esquema de modulação QAM com  $M$  possíveis símbolos é:

$$\xi = \frac{\Delta^2}{6} (M - 1) \quad (5.1)$$

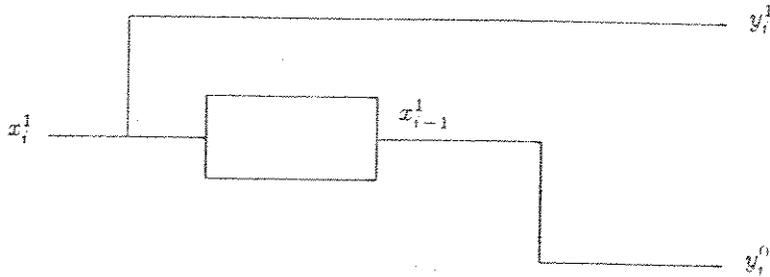


Figura 5.4: Codificador binário ótimo.

Repetindo a equação 2.19, temos que o ganho de codificação vale:

$$\mathcal{G} = \frac{d_{free}^2 \mathcal{E}_{ncod}}{\Delta^2 \mathcal{E}_{cod}} \quad (5.2)$$

onde:

$$\mathcal{E}_{cod} = \frac{\Delta^2}{6} (Q^{k+1} - 1)$$

$$\mathcal{E}_{ncod} = \frac{\Delta^2}{6} (Q^k - 1)$$

pois os códigos usados são sempre de razão  $k/(k+1)$ . Na verdade a equação 5.1 somente é exata quando  $M = Q^k$  e  $k$  for par. No entanto à medida que as constelações tornam se maiores, a aproximação tende à exatidão. No entanto, quando a constelação é de tamanho reduzido, a sua energia deve ser calculada pela força bruta.

O ganho de codificação não varia muito com o número de entradas do codificador. No limite quando  $k \rightarrow \infty$ , a razão de energias tende a  $Q$ , fazendo com que o ganho tenda a:

$$\mathcal{G} = \frac{d_{free}^2}{Q \Delta^2} \quad (5.3)$$

Este comportamento pode ser visto no exemplo abaixo:

- Para  $k = 2$  temos que  $\mathcal{E}_{ncod} = \frac{4-1}{6} \Delta^2 = \frac{1}{2} \Delta^2$  enquanto que  $\mathcal{E}_{cod} = \frac{3}{2} \Delta^2$  e  $\mathcal{G} = 3 \frac{1}{2}$  ou ainda  $\mathcal{G} = 0,00dB$ .

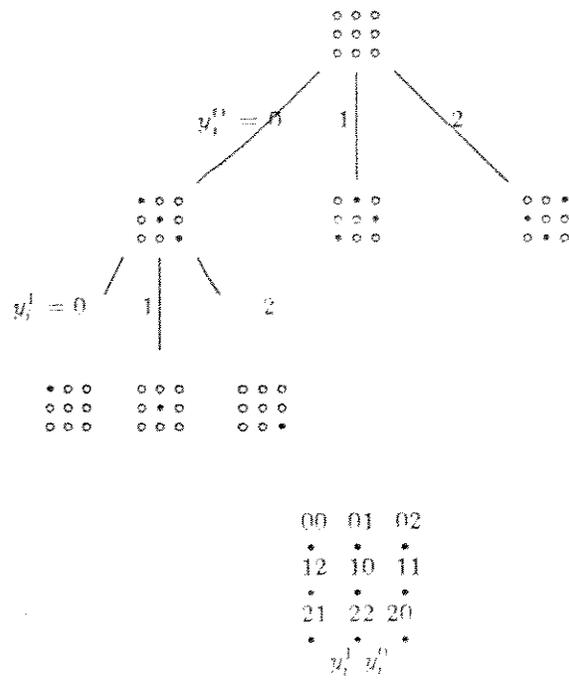


Figura 5.5: Partição e constelação de pontos para o caso ternário.

- Para  $k = 3$  temos que  $\mathcal{E}_{ncod} = \frac{3}{2} \Delta^2$  enquanto que  $\mathcal{E}_{cod} = \frac{5}{2} \Delta^2$  e  $\mathcal{G} = 3\frac{3}{5} \Rightarrow 2,55dB$ .
- Assintoticamente quando  $k \rightarrow \infty$   $\frac{\mathcal{E}_{cod}}{\mathcal{E}_{ncod}} = 2$  então  $\mathcal{G} = 1,76dB$ .

Este comportamento oscilatório deve-se ao fato de que conforme se aumenta o tamanho das constelações, consegue-se melhor ou pior empacotamento das mesmas.

Portanto à medida que o número de entradas não-codificadas cresce, a constelação também cresce, provocando uma estabilização no ganho de codificação, devido ao comportamento assintótico envolvido com a razão de energias codificada e não codificada. Portanto, independente do número de entradas não codificadas ( $k - \bar{k} \geq 1$ ) o código obtido é ótimo e apresenta um  $d_{free}^2$  igual para todos os valores de  $k$  ( $k \geq 2$ ).

### Códigos Ternários

Para o caso ternário, a partição de conjuntos, e a constelação de pontos encontram-se na figura 5.5.

Os pontos da constelação que devem ser alocados aos ramos que saem de um nó, devem distar entre si a maior distância possível, que é o caso dos pontos 00, 10, 20 ou 01, 11, 21 ou ainda 02, 12, 22, ou seja, usando a matriz de transformação ortogonal  $\begin{bmatrix} 1 & -1 \\ 1 & 2 \end{bmatrix}$  da tabela 4.1. Os pontos da constelação que devem ser alocados aos ramos que chegam a um nó, também devem distar entre

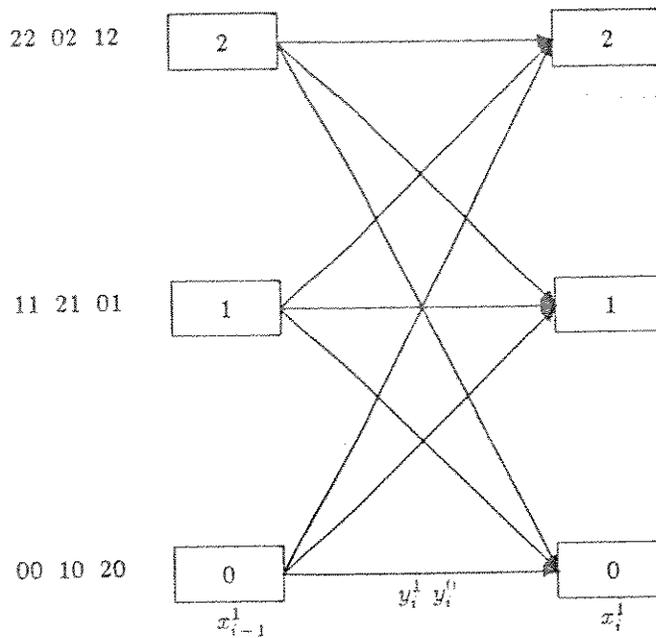


Figura 5.6: Treliça ótima do codificador ternário de uma memória.

si o máximo possível como é o caso dos pontos 00, 11, 22, ou 01, 12, 20, ou ainda 02, 10, 21.

Associando estes pontos aos ramos da treliça, conforme podemos ver na figura 5.6, vamos determinar qual o código que satisfaz este  $d_{free}^2$  máximo e quanto ele vale.

Os pontos na horizontal correspondem a ramos que saem, enquanto que os pontos em uma vertical estão associados a ramos que chegam a um nó. Ressaltando mais uma vez, observe que pontos em qualquer horizontal estão distantes entre si de  $2\Delta^2$ , enquanto que os pontos pertencentes a qualquer vertical são distantes entre si de também de  $2\Delta^2$ , portanto o  $d_{free}^2 = (2 + 2)\Delta^2 = 4\Delta^2$ . Observe que as colunas são entidades independentes, implicando que um rearranjo na ordem das colunas nos fornecerá outros códigos, que neste caso em particular ( $Q = 3$ ), apresentarão mesmo desempenho que o código acima citado. A distância entre transições paralelas vale  $9\Delta^2$ .

À medida que as constelações vão se tornando cada vez maiores, a razão entre energias tende a  $Q = 3$ , tal que o comportamento assintótico pode ser visto abaixo:

- Para  $k = 2$  temos que  $\mathcal{E}_{ncod} = \frac{4\Delta^2}{3}$  enquanto que  $\mathcal{E}_{cod} = \frac{118\Delta^2}{27}$  fazendo com que  $\mathcal{G} = 4 \frac{18}{59} \Rightarrow \mathcal{G} = 0,86dB$ .
- Para  $k = 3$  temos que  $\mathcal{E}_{ncod} = \frac{118\Delta^2}{27}$  enquanto que  $\mathcal{E}_{cod} = \frac{49\Delta^2}{3}$  fazendo com que  $\mathcal{G} = 4 \frac{59}{180} \Rightarrow \mathcal{G} = 1,18dB$ .
- Para  $k \rightarrow \infty$  temos que  $\mathcal{G} = 10 \log\left(\frac{4}{3}\right) = 1,25dB$ .

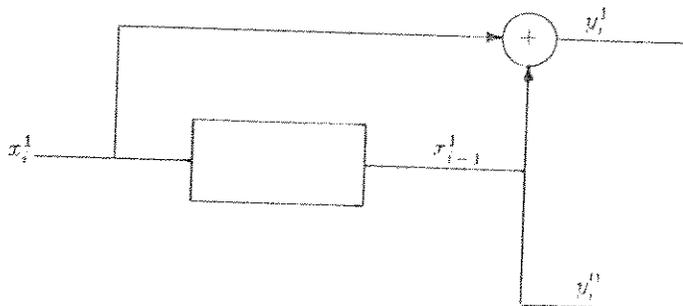


Figura 5.7: Código ternário ótimo.

Um dos códigos (figura 5.6) com o qual se consegue o desempenho acima mencionado, pode ser obtido através do método dos mintermos generalizados:

$$\begin{aligned} y_i^1 &= (r^2 + r)(s^2 + 2) + 2(r^2 + 2r)(s^2 + 2) + (r^2 + 2)(s^2 + s) \\ &+ 2(r^2 + r)(s^2 + s) + 2(r^2 + 2)(s^2 + 2s) + (r^2 + 2r)(s^2 + 2s) \\ &= x_i^1 + x_{i-1}^1 \end{aligned}$$

$$\begin{aligned} y_i^0 &= (r^2 + 2)(s^2 + s) + (r^2 + r)(s^2 + s) + (r^2 + 2r)(s^2 + s) \\ &+ 2(r^2 + 2)(s^2 + 2s) + 2(r^2 + r)(s^2 + 2s) + 2(r^2 + 2r)(s^2 + 2s) \\ &= x_{i-1}^1 \end{aligned}$$

onde por simplicidade de notação usou-se  $r = x_i^1$  e  $s = x_{i-1}^1$ . O código desejado encontra-se na figura 5.7, e a matriz geradora que descreve este código é  $[1 \ 0 \ 1 \ 1]$ .

Como veremos, à medida em que  $Q$  aumenta, a diferença entre o valor da distância livre e a distância entre transições paralelas torna-se cada vez maior, até o momento em que mesmo para um número elevado de memórias, a distância das transições paralelas não se tornarão mais um fator limitante.

#### Códigos $Q$ "111".

Repetindo o raciocínio e procedimento anteriores para o caso de um  $Q$  genérico, nos conduz à tabela 5.1. Nesta tabela constam o  $d_{free}^2$ , o ganho assintótico em dB,

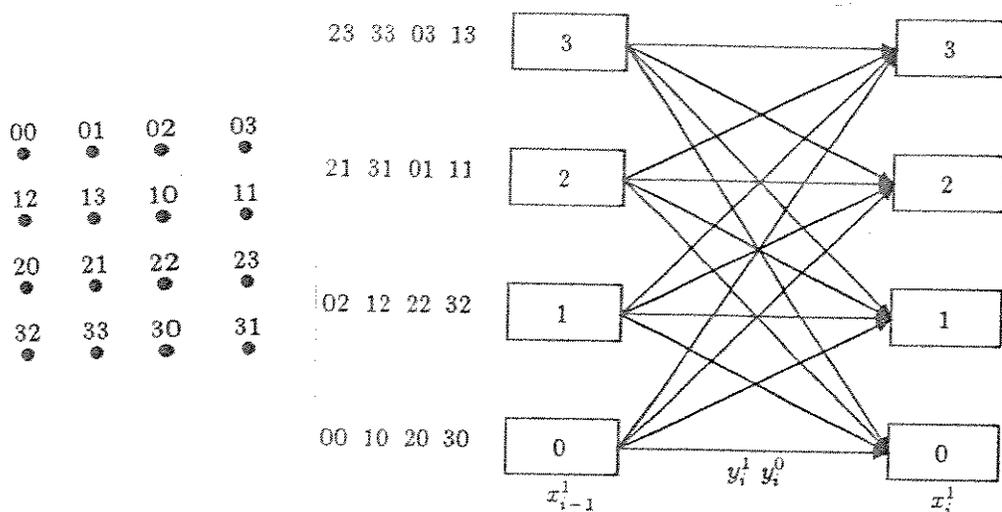


Figura 5.8: Exemplo de código não-linear para  $Q = 4$ , reticulado  $Z^2$ .

a matriz de transformação ortogonal utilizada e um dos códigos ótimos que atinge este desempenho.

Observe que neste caso é difícil encontrar uma lei de formação para o ganho assintótico à medida em que  $Q$  aumenta. Entretanto podemos inferir que o ganho assintótico para  $Q \rightarrow \infty$  e  $k \rightarrow \infty$  tenda aproximadamente a 4,50 dB.

### Códigos não-lineares

Como mencionado anteriormente para  $Q$  não primo, é possível encontrarmos códigos não-lineares melhores que os lineares. Vale ressaltar que pelo fato de não dispormos de um modo sistemático para a obtenção de tais códigos, vamos mostrar a existência de tais códigos através de um exemplo.

**Exemplo 5.1** Como exemplo suponha o caso em que  $Q = 4$ . Neste caso a melhor partição é conseguida com a matriz de transformação ortogonal  $\begin{vmatrix} 2 & 0 \\ 1 & 2 \end{vmatrix}$ . A melhor alocação das palavras aos ramos da treliça, encontra-se na figura 5.8.

Constata-se que neste caso conseguiu-se um  $d_{free}^2 = 9\Delta^2$ , portanto maior que no caso linear, quando se conseguiu somente  $8\Delta^2$ , acarretando em um aumento no ganho assintótico de 3,01 dB para 3,52 dB.

A referência [6] menciona a melhor distância livre para esta partição como sendo de  $8\Delta^2$ . Esta asserção somente é válida para os códigos lineares como fica demonstrado pelo caso não linear acima.

	Q=2	Q=3	Q=4	Q=5
$\frac{d_{free}^2}{\Delta^2}$	3	4	8	10
$\mathcal{G}$	1,76 dB	1,25 dB	3,01 dB	3,01 dB
Mat. Transf. Ort.	$\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & -1 \\ 1 & 2 \end{bmatrix}$	$\begin{bmatrix} 2 & 0 \\ 1 & 2 \end{bmatrix}$	$\begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix}$
Mat. Codif.	[1 0 0 1]	[1 0 1 1]	[1 0 2 1]	[1 0 1 1]
	Q=6	Q=7	Q=8	Q=9
$\frac{d_{free}^2}{\Delta^2}$	11	12	16	18
$\mathcal{G}$	2,63 dB	2,34 dB	3,01 dB	3,01 dB
Mat. Transf. Ort.	$\begin{bmatrix} 2 & 0 \\ 1 & 3 \end{bmatrix}$	$\begin{bmatrix} 2 & 1 \\ 1 & 4 \end{bmatrix}$	$\begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix}$	$\begin{bmatrix} 3 & 0 \\ 1 & 3 \end{bmatrix}$
Mat. Codif.	[1 0 1 1]	[1 0 1 1]	[1 0 1 2]	[1 0 1 1]
	Q=10	Q=11	Q=12	Q=13
$\frac{d_{free}^2}{\Delta^2}$	23	25	24	31
$\mathcal{G}$	3,62 dB	3,57 dB	3,01 dB	3,77 dB
Mat. Transf. Ort.	$\begin{bmatrix} 3 & 2 \\ 1 & 4 \end{bmatrix}$	$\begin{bmatrix} 3 & 1 \\ 1 & 4 \end{bmatrix}$	$\begin{bmatrix} 3 & 0 \\ 2 & 4 \end{bmatrix}$	$\begin{bmatrix} 5 & 2 \\ 1 & 3 \end{bmatrix}$
Mat. Codif.	[1 0 1 1]	[1 0 4 4]	[1 0 3 4]	[1 0 2 6]
	Q=14	Q=15	Q=16	Q=17
$\frac{d_{free}^2}{\Delta^2}$	33	36	38	45
$\mathcal{G}$	3,72 dB	3,80 dB	3,76 dB	4,23 dB
Mat. Transf. Ort.	$\begin{bmatrix} 4 & 2 \\ 1 & 4 \end{bmatrix}$	$\begin{bmatrix} 4 & 1 \\ 1 & 4 \end{bmatrix}$	$\begin{bmatrix} 4 & 0 \\ 1 & 4 \end{bmatrix}$	$\begin{bmatrix} 4 & 3 \\ 1 & 5 \end{bmatrix}$
Mat. Codif.	[1 0 3 11]	[1 0 4 8]	[1 0 1 6]	[1 0 3 10]
	Q=18	Q=19	Q=20	Q=21
$\frac{d_{free}^2}{\Delta^2}$	47	51	51	54
$\mathcal{G}$	4,17 dB	4,29 dB	4,06 dB	4,10 dB
Mat. Transf. Ort.	$\begin{bmatrix} 5 & 2 \\ 1 & 4 \end{bmatrix}$	$\begin{bmatrix} 4 & 1 \\ 1 & 5 \end{bmatrix}$	$\begin{bmatrix} 8 & 4 \\ 1 & 3 \end{bmatrix}$	$\begin{bmatrix} 6 & 3 \\ 1 & 4 \end{bmatrix}$
Mat. Codif.	[1 0 5 13]	[1 0 5 5]	[1 0 3 1]	[1 0 10 20]

Tabela 5.1: A distância livre, o ganho, a matriz de transformação ortogonal e a matriz geradora de um código ótimo com uma memória para o reticulado  $Z^2$ .

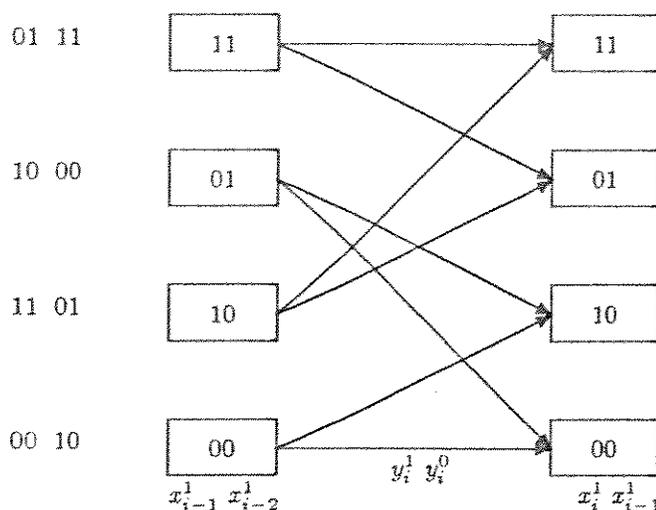


Figura 5.9: Treliça ótima do codificador binário de dois elementos de memória.

### 5.2.3 Códigos de treliça lineares com dois elementos de memória

#### Introdução

A representação esquemática para este tipo de código encontra-se na figura 3.12.

Usou-se 2 elementos de memória em cascata ao invés de ambos em entradas diferentes, pois o  $d_{free}^2$  ainda é menor (exceto para o caso binário que como veremos as transições paralelas limitarão o desempenho) que a distância envolvida entre transições paralelas da treliça ( $Q^2 \Delta^2$ ). Como consequência a colocação de 2 elementos de memória em série permitirá obter um  $d_{free}^2$  maior.

As hipóteses utilizadas anteriormente continuam válidas, ou seja, utilização de pelo menos 2 entradas como a partição de conjuntos permanece a mesma. Neste caso a treliça passa a ter  $Q^2$  estados diferentes.

#### Códigos Binários

Para  $Q = 2$  uma das muitas possíveis alocações ótimas pode ser vista na figura 5.9. Correspondente a esta alocação ótima podemos concluir que a distância livre vale  $d_{free}^2 = (2 + 1 + 2)\Delta^2 = 5\Delta^2$ , e o código que satisfaz esta alocação é obtida através da regra de mintermos:

$$\begin{aligned}
 y_i^1 &= x_i^1(x_{i-1}^1 + 1)(x_{i-2}^1 + 1) + (x_i^1 + 1)x_{i-1}^1(x_{i-2}^1 + 1) + (x_i^1 + 1)(x_{i-1}^1 + 1)x_{i-2}^1 \\
 &+ x_i^1 x_{i-1}^1 x_{i-2}^1 \\
 &= x_i^1 + x_{i-1}^1 + x_{i-2}^1
 \end{aligned}$$

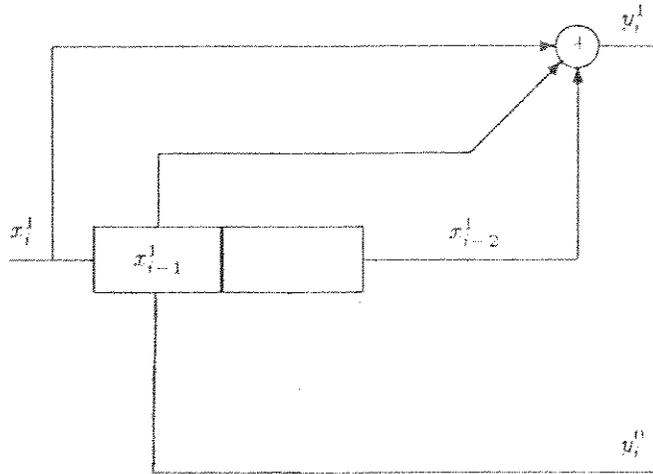


Figura 5.10: Codificador binário ótimo com 2 memórias.

$$\begin{aligned}
 y_i^0 &= (x_i^1 + 1)x_{i-1}^1(x_{i-2}^1 + 1) + x_i^1 x_{i-1}^1(x_{i-2}^1 + 1) + (x_i^1 + 1)x_{i-1}^1 x_{i-2}^1 \\
 &+ x_i^1 x_{i-1}^1 x_{i-2}^1 \\
 &= x_{i-1}^1
 \end{aligned}$$

Entretanto a distância intra-conjunto vale  $Q^2\Delta^2 = 4\Delta^2$ , que para o caso binário começa a limitar a distância livre. Para que essa limitação não ocorra com 3 memórias, se faz necessário a distribuição das memórias entre as demais entradas.

A matriz geradora para este código é dada por  $[1\ 0\ 1\ 1\ 1\ 0]$  e o codificador que descreve este código pode ser visto na figura 5.10.

### Códigos Ternários

A constelação e a treliça para o caso ternário podem ser vistas na figura 5.11. O  $d_{free}^2$  vale  $6\Delta^2 < Q^2\Delta^2 = 9\Delta^2$  e o código obtido pelo método dos mintermos generalizados é descrito a seguir:

$$\begin{aligned}
 y_i^1 &= 2(r^2 + r)(s^2 + 2)(t^2 + 2) + (r^2 + 2r)(s^2 + 2)(t^2 + 2) \\
 &+ 2(r^2 + 2)(s^2 + s)(t^2 + 2) + (r^2 + r)(s^2 + s)(t^2 + t) \\
 &+ (r^2 + 2)(s^2 + 2s)(t^2 + 2) + 2(r^2 + 2r)(s^2 + 2s)(t^2 + 2) \\
 &+ 2(r^2 + 2)(s^2 + 2)(t^2 + t) + (r^2 + r)(s^2 + 2)(t^2 + t) \\
 &+ (r^2 + 2)(s^2 + s)(t^2 + t) + (2(r^2 + 2r)(s^2 + s)(t^2 + t) \\
 &+ 2(r^2 + r)(s^2 + 2s)(t^2 + t) + (r^2 + 2r)((s^2 + 2s)(t^2 + t)
 \end{aligned}$$

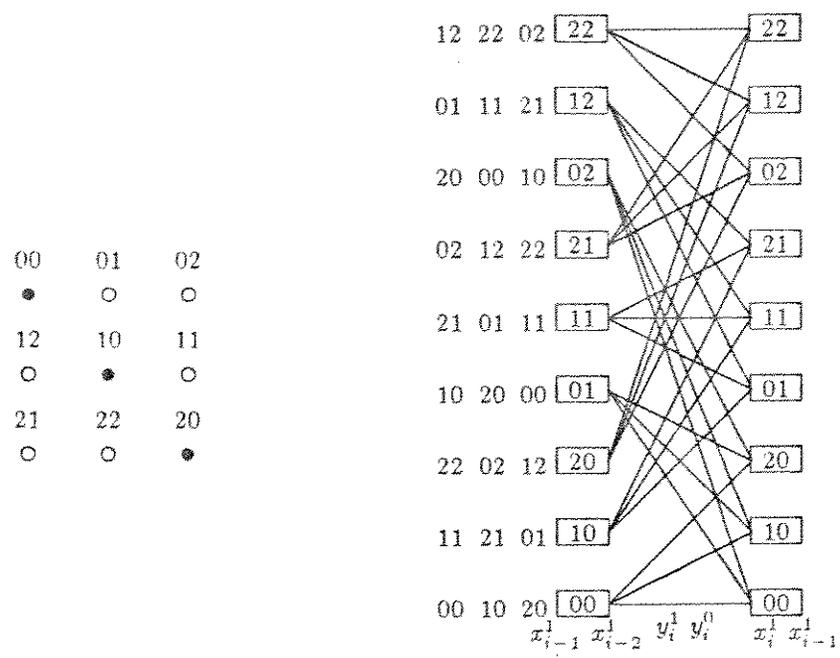


Figura 5.11: Constelação e treliça ternária ótima para codificador com duas memórias.

$$\begin{aligned}
 &+ (r^2 + 2)(s^2 + 2)(t^2 + 2t) + 2(r^2 + 2r)(s^2 + 2)(t^2 + 2t) \\
 &+ 2(r^2 + r)(s^2 + s)(t^2 + 2t) + (r^2 + 2r)(s^2 + s)(t^2 + 2t) \\
 &+ 2(r^2 + 2)(s^2 + 2s)(t^2 + 2t) + (r^2 + r)(s^2 + 2s)(t^2 + 2t)
 \end{aligned}$$

$$y_i^1 = x_i^1 + x_{i-1}^1 + x_{i-2}^1$$

onde por simplicidade de notação usou-se  $r = x_i^1$ ,  $s = x_{i-1}^1$ , e  $t = x_{i-2}^1$ .

A saída  $y_i^0$  pode ser determinada facilmente por inspeção e conclui-se que:

$$y_i^0 = x_{i-1}^1$$

O codificador que realiza esta função encontra-se na figura 5.12, e a matriz geradora deste código é dada por  $[1 \ 0 \ 1 \ 1 \ 1 \ 0]$ .

Códigos  $Q^{n \times 1}$

Repetindo o raciocínio para  $Q > 3$  obteve-se por meios computacionais a tabela 5.2, que fornece o valor do  $d_{free}^2$ , o ganho assintótico ótimo, a matriz de transformação usada e um dos inúmeros códigos ótimos.

Observando-se a tabela 5.2 podemos inferir que ao passar de 1 para 2 elementos de memória deve-se ganhar assintoticamente pouco mais que 1 dB.

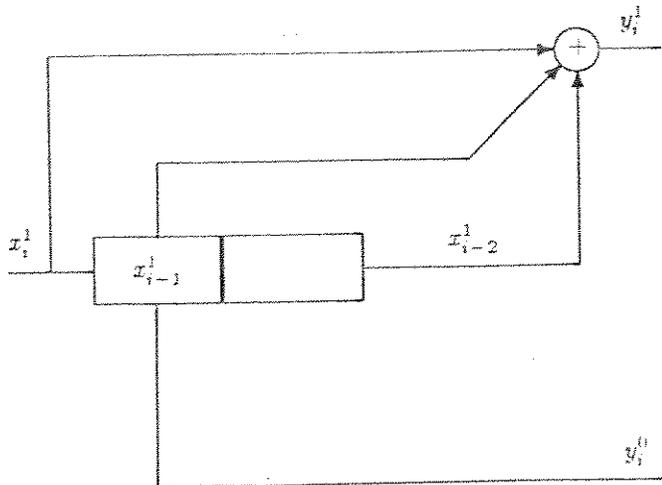


Figura 5.12: Codificador ternário ótimo com duas memórias.

	Q=2	Q=3	Q=4
$\frac{d_{free}^2}{\Delta^2}$	4	6	12
$\mathcal{G}$	3,01 dB	3,01 dB	4,77 dB
Mat. Transf. Ort.	$\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & -1 \\ 1 & 2 \end{bmatrix}$	$\begin{bmatrix} 2 & 0 \\ 1 & 2 \end{bmatrix}$
Mat. Codif.	[1 0 0 1 1 0]	[1 0 1 1 1 0]	[1 0 1 1 1 0]
	Q=5	Q=6	Q=7
$\frac{d_{free}^2}{\Delta^2}$	15	16	19
$\mathcal{G}$	4,77 dB	4,26 dB	4,34 dB
Mat. Transf. Ort.	$\begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix}$	$\begin{bmatrix} 2 & 0 \\ 1 & 3 \end{bmatrix}$	$\begin{bmatrix} 2 & 1 \\ 1 & 4 \end{bmatrix}$
Mat. Codif.	[1 0 1 1 4 0]	[1 0 0 3 2 1]	[1 0 0 1 3 3]
	Q=8	Q=9	
$\frac{d_{free}^2}{\Delta^2}$	27	29	
$\mathcal{G}$	5,28 dB	5,08 dB	
Mat. Transf. Ort.	$\begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix}$	$\begin{bmatrix} 3 & 0 \\ 1 & 3 \end{bmatrix}$	
Mat. Codif.	[1 0 4 0 1 7]	[1 0 2 5 3 5]	

Tabela 5.2: A distância livre, o ganho, a matriz de transformação ortogonal e a matriz geradora de um código ótimo com duas memórias para o reticulado  $Z^2$ .

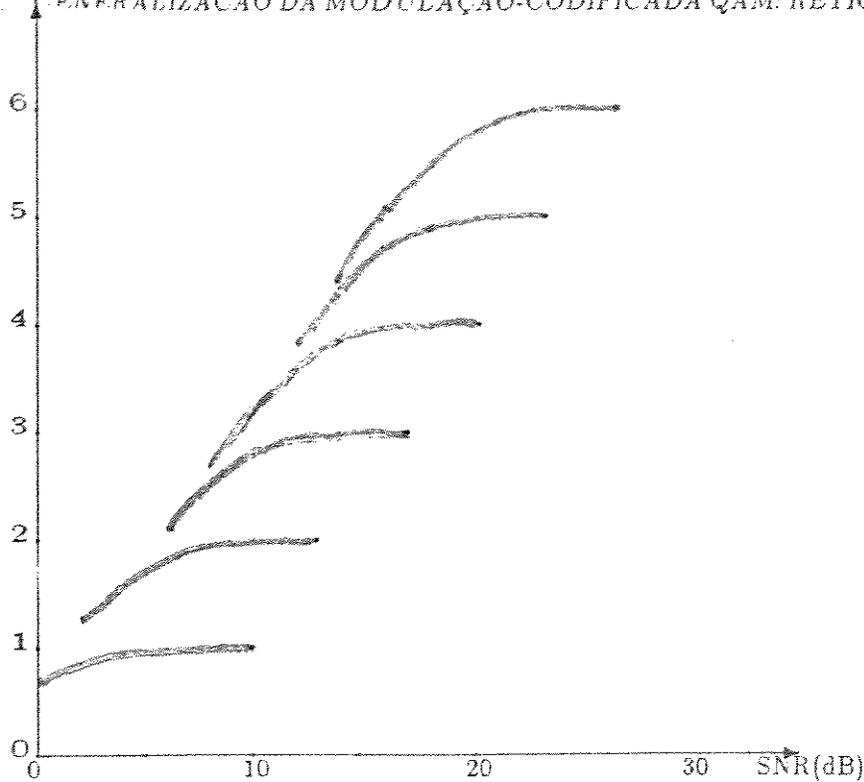


Figura 5.13: Capacidade de canal para constelações binárias,  $Z^2$ .

### 5.3 CAPACIDADE DE CANAL PARA CONSTELAÇÕES QUE USAM O RETICULADO $Z^2$

Para se atingir a máxima eficiência, a decodificação do esquema de modulação conjugada à codificação pressupõe o uso de decodificação por decisão suave (soft decision decoding).

Para o cálculo da capacidade de canal em questão, deve-se usar a expressão da capacidade de um canal sem memória, conforme a expressão 3.11.

$$C = \log_2 M - \frac{1}{M} \sum_{j=0}^{M-1} \int_{-\infty}^{\infty} \log_2 \left\{ \sum_{i=0}^{M-1} \frac{1}{M} p(z|a^i) \right\} \cdot p(z|a^j) dz \quad (5.4)$$

A relação sinal-ruído é definida como:

$$SNR = \frac{\mathcal{E}}{N_0} \quad (5.5)$$

onde  $\frac{N_0}{2}$  é a densidade espectral bilateral do ruído.

Usando a equação 5.4 podemos traçar as curvas de capacidade de canal para constelações  $Q$  *arias* do tipo QAM, que se encontram nas figuras 5.13, 5.14 e 5.15. Observe que estas figuras apresentam a relação sinal-ruído para uma taxa de erro de  $10^{-5}$  obtida através da equação 5.6. Esta equação é obtida através do uso das equações 5.5, 5.1 e 2.23, e sabendo que  $Q(4,4) = 10^{-5}$ :

$$SNR(dB) \simeq 10 \log[6,5(M - 1)] \quad (5.6)$$

CAPÍTULO 5. GENERALIZAÇÃO DA MODULAÇÃO-CODIFICADA QAM: RETICULADOS  $Z^2$  E  $A_2^{84}$

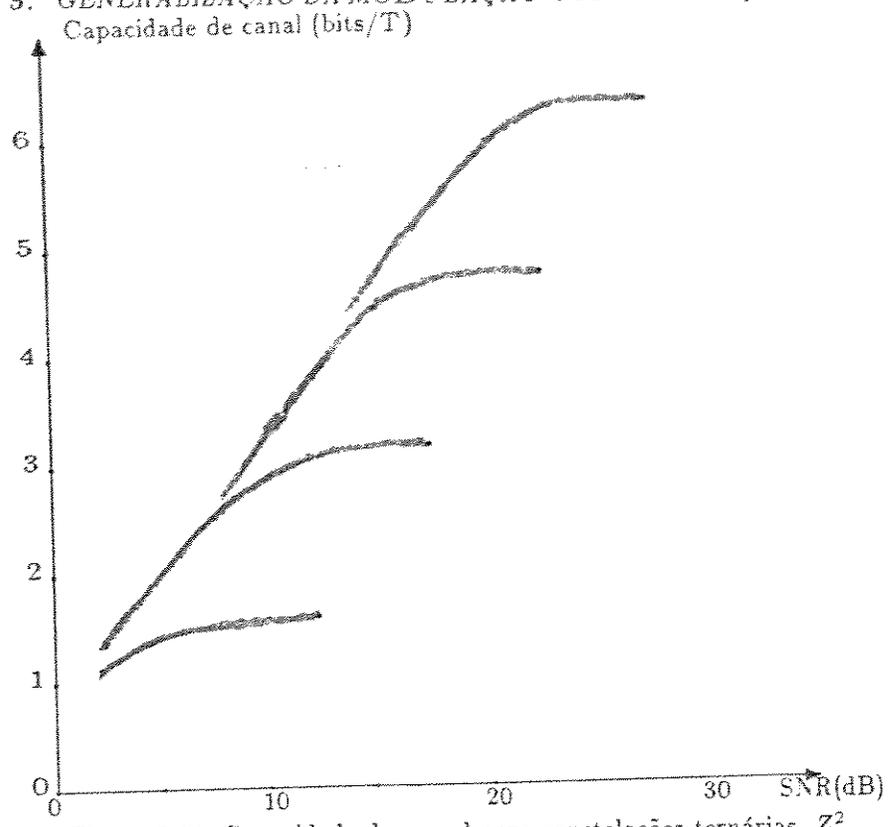


Figura 5.14: Capacidade de canal para constelações ternárias,  $Z^2$ .

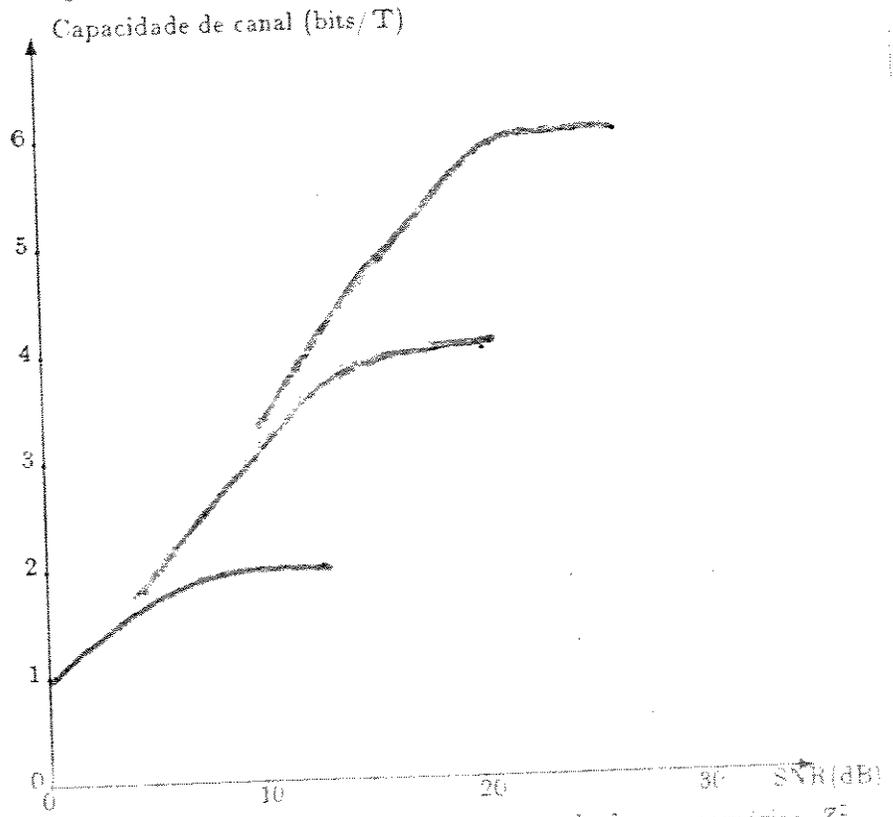


Figura 5.15: Capacidade de canal para constelações quaternárias,  $Z^2$ .

Expansão da Constelação	Ganho
2-QAM $\rightarrow$ 4-QAM	$9,9 - 0,3 = 9,6\text{dB}$
4-QAM $\rightarrow$ 8-QAM	$12,9 - 6,1 = 6,8\text{dB}$
8-QAM $\rightarrow$ 16-QAM	$17,9 - 9,3 = 8,6\text{dB}$
16-QAM $\rightarrow$ 32-QAM	$20,1 - 12,6 = 7,5\text{dB}$
32-QAM $\rightarrow$ 64-QAM	$23,2 - 16,2 = 7,0\text{dB}$
3-QAM $\rightarrow$ 9-QAM	$12,4 - 3,3 = 9,1\text{dB}$
9-QAM $\rightarrow$ 27-QAM	$17,3 - 9,7 = 7,6\text{dB}$
27-QAM $\rightarrow$ 81-QAM	$22,5 - 15,1 = 7,4\text{dB}$
4-QAM $\rightarrow$ 16-QAM	$12,9 - 5,0 = 7,9\text{dB}$
16-QAM $\rightarrow$ 64-QAM	$20,1 - 12,6 = 7,5\text{dB}$

Tabela 5.3: Comportamento do ganho máximo para constelações QAM, reticulado  $Z^2$ .

Suponha que a transmissão normal é feita com o uso de uma constelação de  $Q^k$  pontos. Usando-se um codificador que expanda a constelação para  $Q^{k+1}$ , tem-se que o ganho de codificação máximo é obtido através da diferença entre SNRs; a primeira correspondente à curva de capacidade de  $\log_2 Q^k$  bits/período no ponto de taxa de erro igual a  $10^{-5}$ ; a segunda correspondente à junção entre a curva de capacidade de  $\log_2 Q^{k+1}$  bits/período com a reta paralela ao eixo de SNR com valor de capacidade igual a  $\log_2 Q^k$ . Equivalentemente, deseja-se saber com quantos dB a menos de relação sinal-ruído consegue-se transmitir uma constelação de  $Q^{k+1}$  pontos através de uma taxa de informação de  $k \log_2 Q$  bits, contra uma constelação de  $Q^k$  pontos para a mesma taxa de bits.

A tabela 5.3 apresenta os máximos ganhos em função de  $Q$ . Observe que à medida em que o número de entradas do codificador  $k$  aumenta, temos que o ganho diminui de maneira oscilatória (devido ao melhor ou pior empacotamento que se consegue) até se estabilizar. Por outro lado, à medida em que  $Q$  aumenta (para um mesmo número de entradas  $k$ ) o ganho do codificador diminui gradativamente até se estabilizar.

Da mesma tabela podemos investigar o aumento de ganho caso se use a taxa de informação como padrão de comparação. Neste caso, enquanto a expansão de 4-QAM a 8-QAM proporciona um ganho de 6,8dB, a expansão de 4-QAM a 16-QAM proporciona 7,9dB, ou seja, mais de 1,0dB. Observe que neste caso, existe uma vantagem palpável em se utilizar base quaternária, principalmente quando  $k$  é pequeno, pois neste caso evita-se trabalhar com constelações em que  $k$  é ímpar o que certamente leva a constelações de empacotamento pouco denso

## 5.4 CÓDIGOS ÓTIMOS PARA O RETICULADO $A_2$

### 5.4.1 Códigos de treliça lineares com um elemento de memória

#### Introdução

Nosso interesse aqui também é trabalhar com códigos de razão  $R = \frac{k}{k+1}$  onde  $k \geq 2$ , que pressupõe a utilização de pelo menos uma entrada não codificada. A utilização de entradas não codificadas, acarretará em transições paralelas dentro da treliça, que além de limitar a distância livre em  $Q^2\Delta^2$ , vai proporcionar uma distribuição de pontos mais uniforme e poderá apresentar portanto, uma distância livre menor que no caso em que não há transições paralelas ( $k = 1$ ).

Chamando o número total de entradas de  $k$ , o número de entradas não codificadas de  $k - \tilde{k}$ , e o número de saídas de  $k + 1$ , a partição generalizada apresentará  $Q^{k+1}$  elementos no total, com  $Q^{\tilde{k}+1}$  sub-conjuntos e  $Q^{k-\tilde{k}}$  elementos por subconjunto.

Vamos a seguir calcular as distâncias livres máximas para cada esquema de modulação  $Q$  *aria*, e descobrir qual codificador atende esta máxima distância livre.

Como mencionado anteriormente, o reticulado  $A_2$  não oferece um bom particionamento para o caso binário, pelo fato de apresentar um maior número de pontos vizinhos que no caso do reticulado  $Z^2$ , conforme pode ser observado na tabela 4.2.

#### Códigos Ternários

A partição de conjuntos, e a constelação de pontos encontram-se na figura 5.16.

Os símbolos associados aos ramos que saem de um estado, devem ser escolhidos tais que a distância entre si seja a maior possível. Os elementos que satisfazem esta condição são os elementos 00, 10 e 20 ou 01, 11 e 21 ou 02, 12 e 22 pontos que distam entre si de  $3\Delta^2$ . Os símbolos associados aos ramos que chegam a um estado são pontos distantes entre si de  $\Delta^2$ , tais como os pontos 00, 11 e 22.

Uma das várias possíveis alocações ótimas dos símbolos aos ramos da treliça está na figura 5.17, da qual se tira que o  $d_{free}^2 = (3 + 1)\Delta^2 = 4\Delta^2$ , e o código que satisfaz esta alocação pode ser obtido através da regra de mintermos:

$$\begin{aligned} y_i^1 &= (r^2 + 2)(s^2 + s) + (r^2 + r)(s^2 + s) + (r^2 + 2r)(s^2 + s) \\ &+ 2(r^2 + 2)(s^2 + 2s) + 2(r^2 + r)(s^2 + 2s) + 2(r^2 + 2r)(s^2 + 2s) \\ &= x_{i-1}^1 \end{aligned}$$

$$\begin{aligned} y_i^0 &= (r^2 + r)(s^2 + 2) + 2(r^2 + 2r)(s^2 + 2) + (r^2 + 2)(s^2 + s) \\ &+ 2(r^2 + r)(s^2 + s) + 2(r^2 + r)(s^2 + 2s) + (r^2 + 2r)(s^2 + 2s) \\ &= x_i^1 + x_{i-1}^1 \end{aligned}$$

onde o primeiro bit de cada dibit é representado na figura 5.16 por  $y_i^1$ , e o segundo por  $y_i^0$ . Ambos podem ser escritos como soma de produtos de  $x_{i-1}^1$  e  $x_i^1$ , onde  $x_i^1$  é

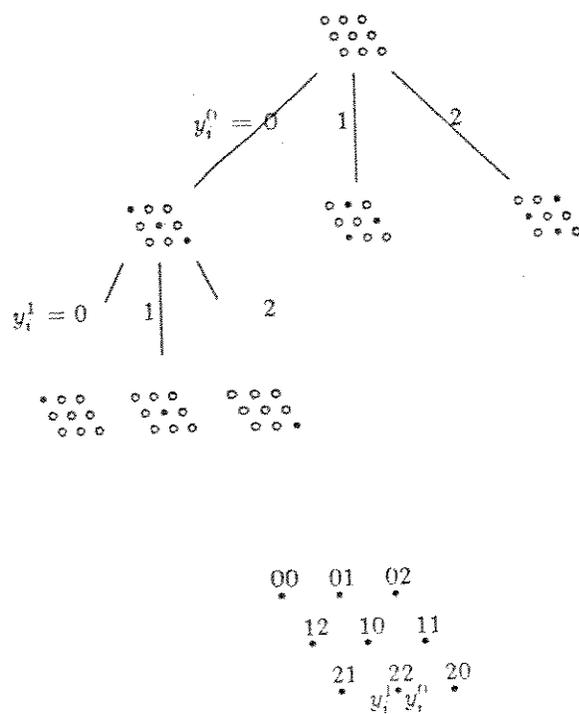


Figura 5.16: Partição de conjuntos e constelação ternária.

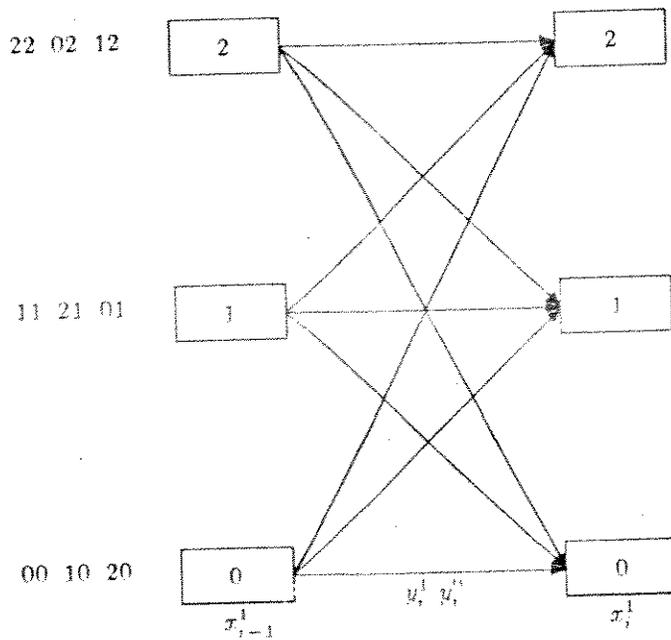


Figura 5.17: Treliça ternária ótima de codificador de uma memória,  $A_2$ .

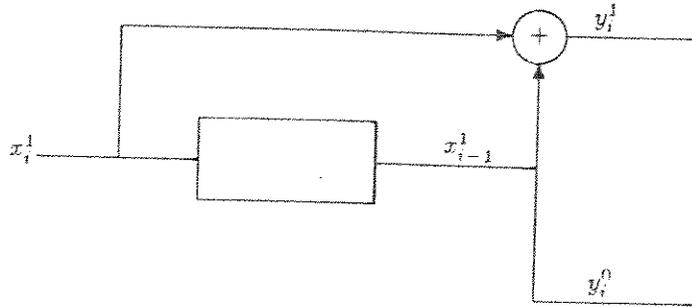


Figura 5.18: Codificador ternário ótimo.

o bit de entrada presente e  $x_{i-1}^1$  é o bit futuro. O codificador que executa a função calculada anteriormente encontra-se na figura 5.18, e o gerador que descreve este código é [1 0 1 1].

Como já foi dito anteriormente a ocorrência de entradas não codificadas, implicará em transições paralelas aos ramos da treliça. A distância entre transições paralelas é de  $3^2 \Delta^2 = 9\Delta^2$ , que neste caso é maior que a distância livre.

O aumento no número de entradas não codificadas implicará num aumento da constelação. A energia média de um esquema de modulação QAM com  $M$  possíveis símbolos é:

$$\mathcal{E} = \frac{\Delta^2}{6}(M - 1) \quad (5.7)$$

O ganho de codificação vale:

$$\mathcal{G} = \frac{d_{free}^2 \mathcal{E}_{ncod}}{\Delta^2 \mathcal{E}_{cod}} \quad (5.8)$$

onde:

$$\mathcal{E}_{cod} = \frac{\Delta^2}{6}(Q^{k+1} - 1)$$

$$\mathcal{E}_{ncod} = \frac{\Delta^2}{6}(Q^k - 1)$$

pois os códigos usados são sempre de razão  $k/(k + 1)$ . A expressão 5.7 somente é exata quando  $M = Q^k$  para  $k$  par, assim sendo, quando isto não ocorrer a energia deverá ser calculada através da força bruta.

O ganho de codificação não varia muito em função do número de entradas do codificador. Com o aumento da constelação a razão entre energias tende a  $Q$ , fazendo com que o ganho tenda a  $\mathcal{G} = \frac{d_{frec}^2}{Q\Delta^2}$ . Este comportamento pode ser visto no exemplo abaixo:

- Para  $k = 2$  temos que  $\mathcal{E}_{ncod} = \frac{9-1}{6}\Delta^2 = \frac{4}{3}\Delta^2$  enquanto que  $\mathcal{E}_{cod} = 5\Delta^2$  e  $\mathcal{G} = 4\frac{4}{15}$  ou ainda  $\mathcal{G} = 0,28dB$ .
- Para  $k = 3$  temos que  $\mathcal{E}_{ncod} = 5\Delta^2$  enquanto que  $\mathcal{E}_{cod} = \frac{40}{3}\Delta^2$  e  $\mathcal{G} = 4\frac{3}{8}\Delta^2 \Rightarrow 1,76dB$ .
- Assintoticamente quando  $k \rightarrow \infty$   $\frac{\mathcal{E}_{cod}}{\mathcal{E}_{ncod}} = 3$  então  $\mathcal{G} = 1,25dB$ .

Este comportamento oscilatório deve-se ao fato de que conforme se aumenta o tamanho das constelações, consegue-se melhor ou pior empacotamento das mesmas.

Portanto à medida que o número de entradas não codificadas cresce, a constelação também cresce, provocando uma estabilização no ganho de codificação, devido ao comportamento assintótico envolvido com a razão de energias codificada e não codificada. Portanto, independente do número de entradas não codificadas ( $k - \tilde{k} \geq 1$ ) o código obtido é ótimo e apresenta um  $d_{frec}^2$  constante para todos os valores de  $k$  ( $k \geq 2$ ).

#### Códigos Quaternários

Para o caso quaternário, a partição de conjuntos, e a constelação de pontos encontram-se na figura 5.19. Por uma perversidade da natureza o uso da matriz de transformação ortogonal  $\begin{vmatrix} 2 & 0 \\ 0 & 2 \end{vmatrix}$  não produz códigos, lineares ou não-lineares,

com mesmo desempenho que a matriz usada na partição da figura 5.19  $\begin{vmatrix} 1 & -2 \\ 1 & 2 \end{vmatrix}$  que tem uma distância quadrática mínima de 3.

Os pontos da constelação que devem ser alocados aos ramos que saem de um nó, devem distar entre si a maior distância possível, que é o caso dos pontos 00, 10, 20, 30, ou 01, 11, 21, 31, ou 02, 12, 22, 32, ou ainda 03, 13, 23, 33. Os pontos da constelação que devem ser alocados aos ramos que chegam a um nó, também devem distar entre si o mais longe possível, como é o caso dos pontos 00, 11, 22, 33, ou 01, 12, 23, 30, ou ainda 02, 13, 20, 31, ou ainda 03, 10, 21, 32.

Associando estes pontos aos ramos da treliça, conforme podemos ver na figura 5.20, vamos determinar qual o código que satisfaz este  $d_{frec}^2$  máximo e quanto ele vale. Os pontos na horizontal correspondem a ramos que saem, enquanto que os pontos em uma vertical estão associados a ramos que chegam a um nó. Ressaltando mais uma vez, observe que pontos em qualquer horizontal estão distantes entre si de  $3\Delta^2$ , enquanto que os pontos pertencentes a qualquer vertical estão distantes entre si de  $3\Delta^2$ , portanto o  $d_{frec}^2 = (3 + 3)\Delta^2 = 6\Delta^2$ . Observe que as colunas são entidades independentes, implicando que um rearranjo na ordem das colunas

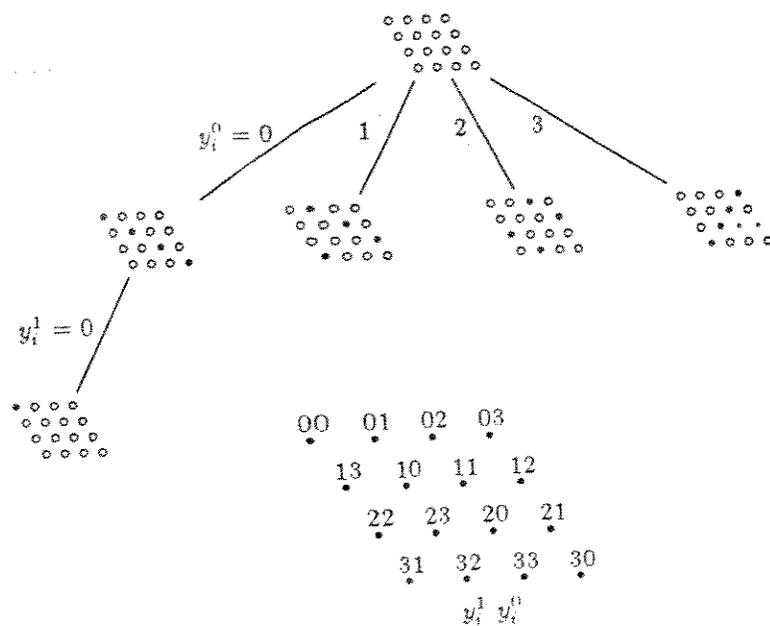


Figura 5.19: Partição, e constelação para o caso quaternário.

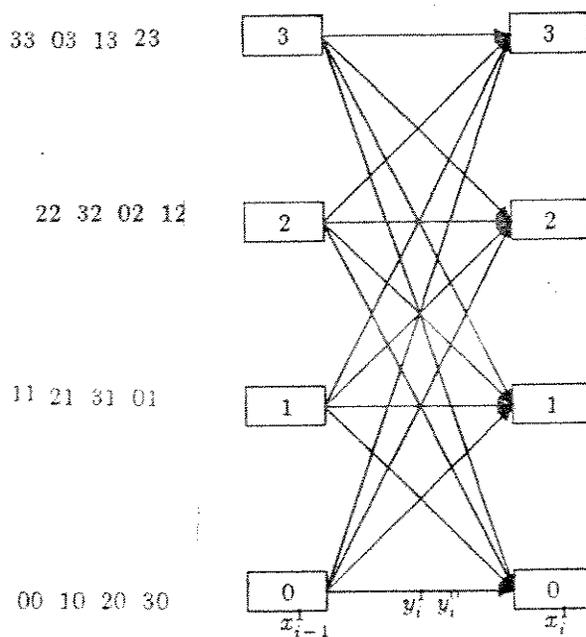


Figura 5.20: Treliça quaternária para codificador de uma memória

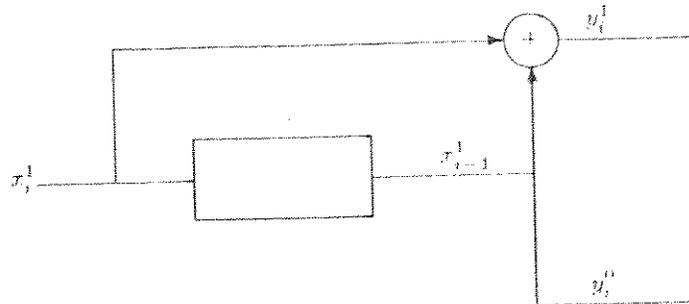


Figura 5.21: Código quaternário ótimo.

nos fornecerá outros códigos, que neste caso em particular ( $Q = 4$ ), apresentarão mesmo desempenho que o código acima citado.

À medida que as constelações vão se tornando cada vez maiores, a razão entre energias tende a  $Q = 4$ , tal que o comportamento assintótico pode ser visto abaixo:

- Para  $k = 2$  temos que  $\mathcal{E}_{ncod} = \frac{5}{2} \Delta^2$  enquanto que  $\mathcal{E}_{cod} = \frac{21}{2} \Delta^2$  fazendo com que  $\mathcal{G} = 6 \frac{5}{21} \Delta^2 \Rightarrow \mathcal{G} = 1,55dB$ .
- Para  $k = 3$  temos que  $\mathcal{E}_{ncod} = \frac{21}{2} \Delta^2$  enquanto que  $\mathcal{E}_{cod} = \frac{85}{2} \Delta^2$  fazendo com que  $\mathcal{G} = 6 \frac{21}{85} \Delta^2 \Rightarrow \mathcal{G} = 1,70dB$ .
- Para  $k \rightarrow \infty$  temos que  $\mathcal{G} = 10 \log\left(\frac{6}{4}\right) = 1,76dB$ .

Um dos códigos (figura 5.20) com o qual se consegue o desempenho acima mencionado é:

$$y_i^1 = x_i^1 + x_{i-1}^1$$

$$y_i^0 = x_{i-1}^1$$

O código desejado encontra-se na figura 5.21. e a matriz geradora que descreve este código é  $[1 \ 0 \ 1 \ 1]$ . A distância entre transições paralelas é igual a  $4^2 \Delta^2$ .

Códigos  $Q$  *arias*

Repetindo todo este raciocínio que foi aplicado atrás, se aplicado a um  $Q$  genérico, nos conduz à tabela 5.4, que fornece o  $d_{free}^2$ , o ganho em dB, e um dos códigos ótimos que atinge este desempenho.

Observe que neste caso também é difícil encontrar uma lei de formação à medida em que  $Q$  aumenta.

É de se supor que o ganho assintótico para  $Q \rightarrow \infty$  e  $k \rightarrow \infty$  tenda aos 4 dB.

## 5.5 CAPACIDADE DE CANAL PARA CONSTELAÇÕES QUE USAM O RETICULADO $A_2$

A decodificação do esquema de modulação conjugada à codificação pressupõe o uso de decodificação por decisão do tipo suave (soft decision decoding).

Para cálculo da capacidade de canal neste caso, deve-se usar a expressão da capacidade de um canal sem memória.

$$C = \log_2 M - \frac{1}{M} \sum_{j=0}^{M-1} \int_{-\infty}^{\infty} \log_2 \left\{ \sum_{i=0}^{M-1} \frac{1}{M} p(z|a^i) \right\} \cdot p(z|a^j) dz \quad (5.9)$$

A relação sinal-ruído é definida como:

$$SNR = \frac{\mathcal{E}}{N_0} \quad (5.10)$$

onde  $\frac{N_0}{2}$  é a densidade espectral bilateral do ruído.

Usando a equação 5.9 podemos tirar as curvas de capacidade de canal para constelações  $Q$  *arias* do tipo QAM, que se encontram nas figuras 5.5, 5.5 e 5.5.

Observe que estas figuras apresentam a relação sinal-ruído para uma taxa de erro de  $10^{-5}$  obtida através da equação 5.6. Suponha que a transmissão normal é feita com o uso de uma constelação de  $Q^k$  pontos. Usando-se um codificador que expanda a constelação para  $Q^{k+1}$ , tem-se que o ganho de codificação máximo é obtido através da diferença entre  $SNRs$ ; a primeira correspondente à curva de capacidade de  $\log_2 Q^k$  bits/período no ponto de taxa de erro igual a  $10^{-5}$ ; a segunda correspondente à junção entre a curva de capacidade de  $\log_2 Q^{k+1}$  bits/período com a reta paralela ao eixo de  $SNR$  com valor de capacidade igual a  $\log_2 Q$ . Trocando em miúdos, deseja-se saber com quantos dB a menos de relação sinal-ruído consegue-se transmitir uma constelação de  $Q^{k+1}$  pontos através de uma taxa de  $k \log_2 Q$  bits, contra uma constelação de  $Q^k$  pontos usando-se a mesma taxa de bits.

A tabela 5.5 apresenta os máximos ganhos em função de  $Q$ . Observe que à medida em que o número de entradas do codificador  $k$  aumenta, temos que o ganho diminui até se estabilizar por volta de 6,5dB. À medida em que  $Q$  aumenta (para um mesmo número de entradas  $k$ ) o ganho do codificador diminui gradativamente até se estabilizar.

	Q=2	Q=3	Q=4	Q=5
$\frac{d_{free}^2}{\Delta^2}$	2	4	6	7
$\mathcal{G}$	0,00 dB	1,25 dB	1,76 dB	1,46 dB
Mat. Transf. Ort.	$\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & -1 \\ 1 & 2 \end{bmatrix}$	$\begin{bmatrix} 1 & -2 \\ 1 & 2 \end{bmatrix}$	$\begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix}$
Mat. Codif.	[1 0 0 1]	[1 0 1 1]	[1 0 1 1]	[1 0 1 1]
	Q=6	Q=7	Q=8	Q=9
$\frac{d_{free}^2}{\Delta^2}$	8	14	14	18
$\mathcal{G}$	1,25 dB	3,01 dB	2,43 dB	3,01 dB
Mat. Transf. Ort.	$\begin{bmatrix} 2 & 0 \\ 1 & 3 \end{bmatrix}$	$\begin{bmatrix} 2 & 1 \\ 1 & 4 \end{bmatrix}$	$\begin{bmatrix} 2 & 0 \\ 1 & 4 \end{bmatrix}$	$\begin{bmatrix} 2 & 1 \\ 1 & 5 \end{bmatrix}$
Mat. Codif.	[1 0 1 1]	[1 0 1 2]	[1 0 1 1]	[1 0 3 4]
	Q=10	Q=11	Q=12	Q=13
$\frac{d_{free}^2}{\Delta^2}$	20	21	25	27
$\mathcal{G}$	3,01 dB	2,81 dB	3,19 dB	3,17 dB
Mat. Transf. Ort.	$\begin{bmatrix} 2 & 0 \\ 1 & 5 \end{bmatrix}$	$\begin{bmatrix} 2 & 1 \\ 1 & 6 \end{bmatrix}$	$\begin{bmatrix} 3 & 0 \\ 1 & 4 \end{bmatrix}$	$\begin{bmatrix} 3 & 2 \\ 1 & 5 \end{bmatrix}$
Mat. Codif.	[1 0 3 5]	[1 0 4 4]	[1 0 2 2]	[1 0 2 12]
	Q=14	Q=15	Q=16	Q=17
$\frac{d_{free}^2}{\Delta^2}$	28	33	33	39
$\mathcal{G}$	3,01 dB	3,42 dB	3,14 dB	3,61 dB
Mat. Transf. Ort.	$\begin{bmatrix} 3 & 1 \\ 1 & 5 \end{bmatrix}$	$\begin{bmatrix} 3 & 0 \\ 1 & 5 \end{bmatrix}$	$\begin{bmatrix} 3 & 2 \\ 1 & 6 \end{bmatrix}$	$\begin{bmatrix} 3 & 1 \\ 1 & 6 \end{bmatrix}$
Mat. Codif.	[1 0 1 2]	[1 0 4 4]	[1 0 7 5]	[1 0 7 7]
	Q=18	Q=19	Q=20	Q=21
$\frac{d_{free}^2}{\Delta^2}$	38	45	48	48
$\mathcal{G}$	3,25 dB	3,74 dB	3,80 dB	3,59 dB
Mat. Transf. Ort.	$\begin{bmatrix} 3 & 0 \\ 1 & 6 \end{bmatrix}$	$\begin{bmatrix} 7 & 2 \\ 1 & 3 \end{bmatrix}$	$\begin{bmatrix} 4 & 0 \\ 1 & 5 \end{bmatrix}$	$\begin{bmatrix} 4 & 3 \\ 1 & 6 \end{bmatrix}$
Mat. Codif.	[1 0 7 7]	[1 0 2 8]	[1 0 5 3]	[1 0 2 3]

Tabela 5.4: A distância livre, o ganho, a matriz de transformação ortogonal e a matriz geradora de um código ótimo com uma memória para o reticulado  $A_2$ .

CAPÍTULO 5. GENERALIZAÇÃO DA MODULAÇÃO-CODIFICADA QAM: RETICULADOS  $Z^2$  E  $A_2$  94

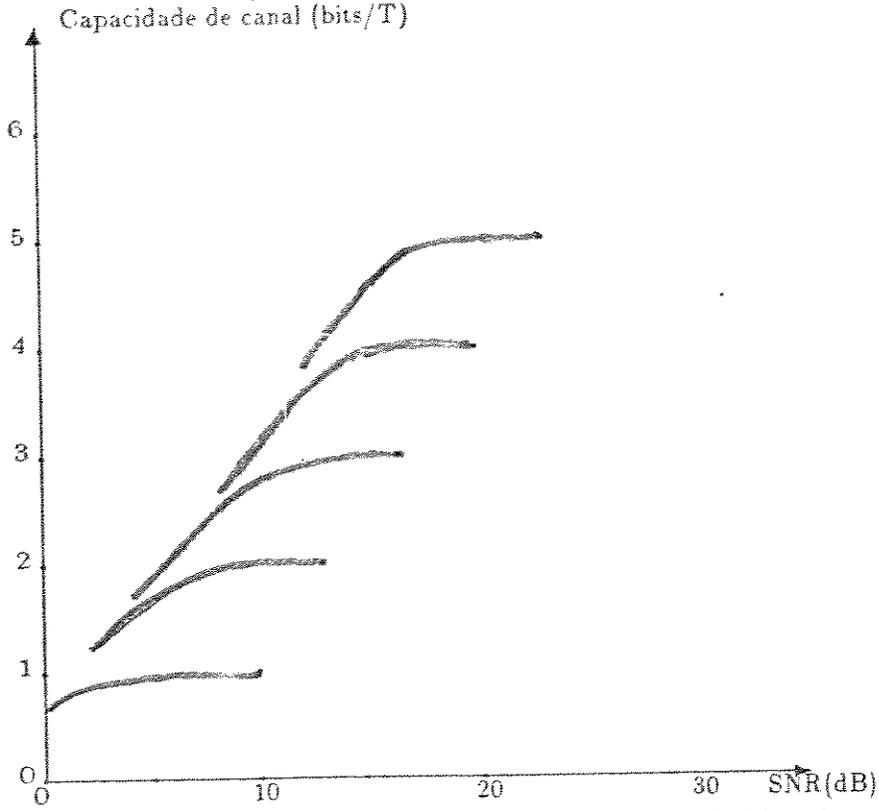


Figura 5.22: Capacidade de canal para constelações binárias,  $A_2$ .

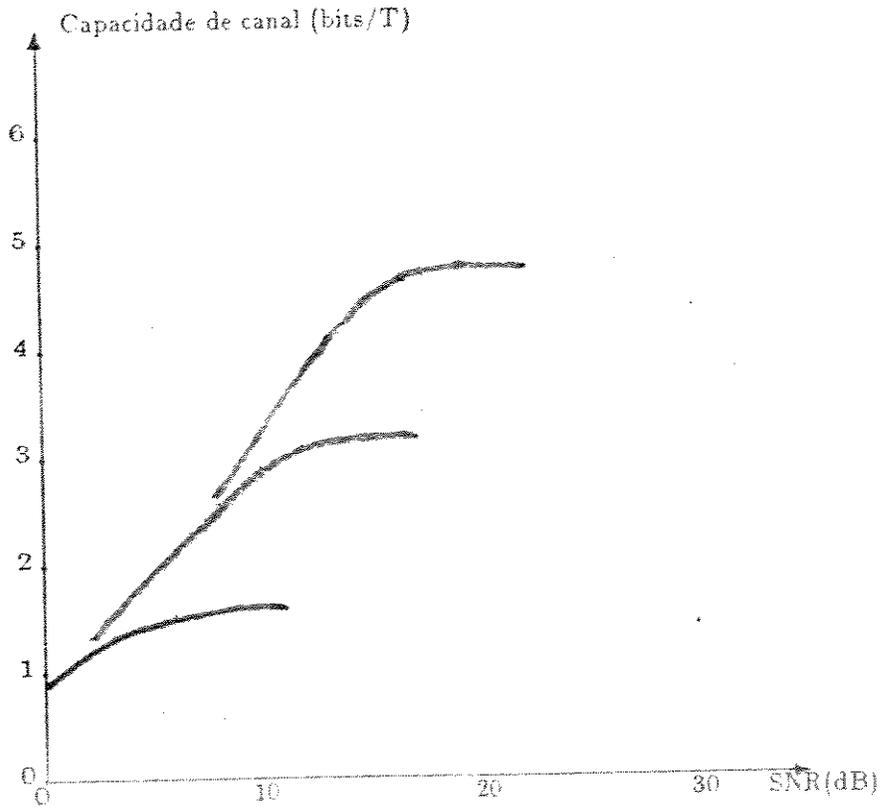


Figura 5.23: Capacidade de canal para constelações ternárias,  $A_2$ .

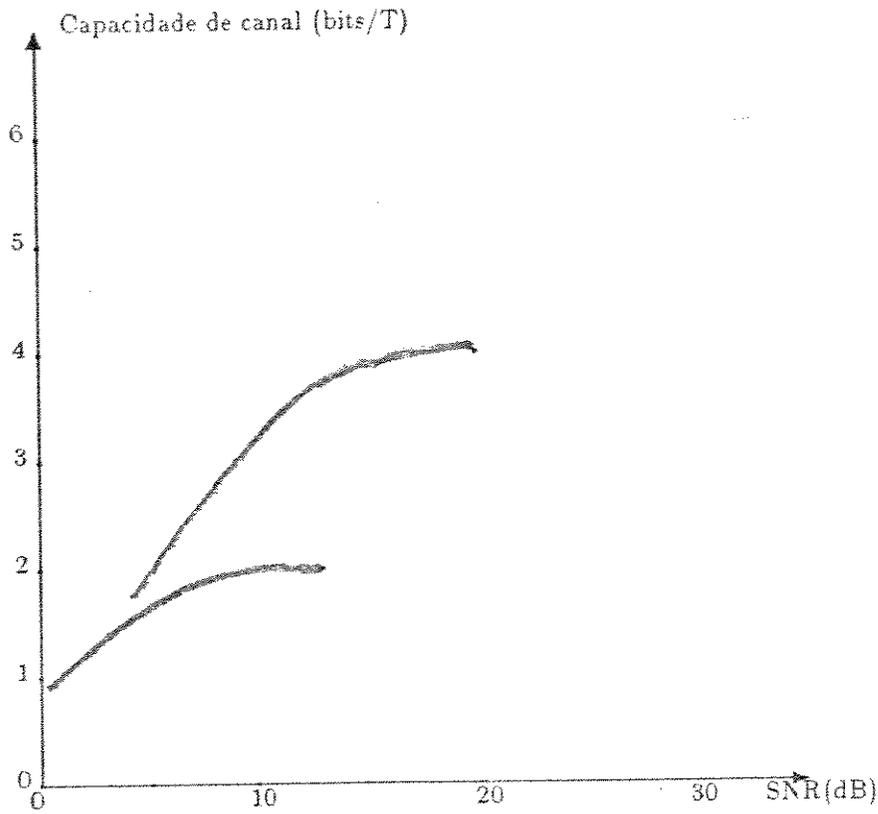


Figura 5.24: Capacidade de canal para constelações quaternárias,  $A_2$ .

Expansão da Constelação	Ganho
2-QAM $\rightarrow$ 4-QAM	$9,9 - 0,5 = 9,4\text{dB}$
4-QAM $\rightarrow$ 8-QAM	$12,9 - 5,5 = 7,4\text{dB}$
8-QAM $\rightarrow$ 16-QAM	$16,4 - 9,3 = 7,1\text{dB}$
16-QAM $\rightarrow$ 32-QAM	$19,6 - 12,8 = 6,8\text{dB}$
3-QAM $\rightarrow$ 9-QAM	$11,2 - 3,3 = 7,9\text{dB}$
9-QAM $\rightarrow$ 27-QAM	$17,0 - 9,8 = 7,2\text{dB}$
4-QAM $\rightarrow$ 16-QAM	$12,9 - 5,1 = 7,8\text{dB}$

Tabela 5.5: Comportamento do ganho assintótico máximo para o reticulado  $A_2$ .

Da mesma tabela podemos investigar o aumento de ganho caso se use códigos de razão  $\frac{k}{k+2}$ . Enquanto a expansão de 4-QAM a 8-QAM proporciona um ganho de 7,4dB, a expansão de 4-QAM a 16-QAM proporciona 7,8dB. Portanto, o uso de constelações quaternárias apresentam um ganho de 0,4db em relação ao uso de constelações binárias.

## Capítulo 6

# COMENTÁRIOS FINAIS E CONCLUSÕES

### 6.1 PARTIÇÕES DE CONSTELAÇÕES MULTI-DIMENSIONAIS

O objetivo neste secção não é fornecer a solução exata sobre o modo de partição de constelações multi-dimensionais, mas sim deixar uma proposta de como isso seria feito em 4 dimensões.

Para o reticulado  $Z^2$  definiu-se no capítulo 4 a Célula Básica como sendo um arranjo composto de  $Q^2$  elementos. Neste caso é mais conveniente defini-lo como sendo um arranjo de  $Q^4$  pontos.

Para o caso binário já exaustivamente estudado de 2 dimensões, o reticulado é o  $Z^2$ , a Célula Básica consiste de  $Q^2 = 4$  pontos separados de uma distância  $d_{min}^2 = 1$  formando um quadrado. Suponha que as coordenadas desses 4 pontos sejam  $(0, 0)$ ,  $(0, 1)$ ,  $(1, 0)$ ,  $(1, 1)$ . Neste caso a tabela 6.1 nos mostra como a partição vai evoluindo e como as distâncias mínimas vão crescendo para o caso binário, onde  $N$  denota o número de pontos.

Para o caso binário de 4 dimensões, o reticulado é o  $Z^4$ , a Célula Básica consiste de  $Q^4 = 16$  pontos separados de uma distância  $d_{min}^2 = 1$  formando um hiper-cubo quadri-dimensional. Suponha que as coordenadas desses 16 pontos sejam  $(0, 0, 0, 0)$ ,  $\dots$ ,  $(1, 1, 1, 1)$ . Neste caso a tabela 6.2 nos mostra como a partição vai evoluindo e como as distâncias mínimas vão crescendo.

$N = 4$	$N = 2$	$N = 1$
$d_{min}^2 = 1$	$d_{min}^2 = 2$	$d_{min}^2 = 4$
$(0, 0)$	$(0, 0)$	$(0, 0)$
$(0, 1)$	$(1, 1)$	
$(1, 0)$		
$(1, 1)$		

Tabela 6.1: Evolução da  $d_{min}^2$  e do número de pontos para partições no reticulado  $Z^2$  para  $Q = 2$ .

$N = 16$	$N = 8$	$N = 4$	$N = 2$
$d_{min}^2 = 1$	$d_{min}^2 = 2$	$d_{min}^2 = 2$	$d_{min}^2 = 4$
(0, 0, 0, 0)	(0, 0, 0, 0)	(0, 0, 0, 0)	(0, 0, 0, 0)
(0, 0, 0, 1)	(0, 0, 1, 1)	(0, 0, 1, 1)	(1, 1, 1, 1)
(0, 0, 1, 0)	(0, 1, 0, 1)	(1, 1, 0, 0)	
(0, 0, 1, 1)	(0, 1, 1, 0)	(1, 1, 1, 1)	
(0, 1, 0, 0)	(1, 0, 0, 1)		
(0, 1, 0, 1)	(1, 0, 1, 0)		
(0, 1, 1, 0)	(1, 1, 0, 0)		
(0, 1, 1, 1)	(1, 1, 1, 1)		
(1, 0, 0, 0)			
(1, 0, 0, 1)			
(1, 0, 1, 0)			
(1, 0, 1, 1)			
(1, 1, 0, 0)			
(1, 1, 0, 1)			
(1, 1, 1, 0)			
(1, 1, 1, 1)			

Tabela 6.2: Evolução da  $d_{min}^2$  e do número de pontos para partições no reticulado  $Z^4$  para  $Q = 2$ .

A primeira partição a partir do  $Z^4$  dá origem a um reticulado de 8 pontos, esse reticulado recebe o nome de  $D_4$ , ou reticulado de Schläfli. A próxima partição dá origem a um novo sub-reticulado em que o número de pontos cai à metade, porém sem aumento na distância mínima, fazendo com que ele apresente uma densidade menor que a do reticulado  $D_4$ . Para que o reticulado apresente um aumento na distância mínima, deve-se particionar novamente, conforme indica a tabela 6.2. Observe que em quatro dimensões apareceu uma novidade que não existia em duas dimensões: a ocorrência de partições sem aumento da distância mínima.

## 6.2 PROBLEMAS DEIXADOS EM ABERTO

Como problemas deixados em aberto para posterior estudo temos:

- Obtenção de códigos não-lineares de maneira sistemática
- Obtenção dos códigos não-lineares para mais de uma memória
- Obtenção de códigos e ganhos para constelações do tipo M-PSK
- Cálculo da capacidade de canal para constelações do tipo M-PSK
- Obtenção de códigos para os casos em que  $Q$  não é primo, mas é potência de primo, usando operações que forneçam um corpo
- Obtenção de códigos e os respectivos ganhos para constelações multi-dimensionais

### 6.3 CONCLUSÕES

Os resultados novos deste trabalho estão presentes nos capítulos 3, 4, 5.

No capítulo 3 é feito um particionamento intuitivo de constelações PAM  $Q$  *arias*, particionamento este que é fundamentado com a leitura do capítulo seguinte. Ainda no capítulo 3 são obtidos os códigos  $Q$  *arias* ótimos e seus respectivos desempenhos através das suas distâncias livres para o número de memórias variando de um a cinco. Mostrou-se que mesmo para  $Q$  tendendo a infinito, para um determinado número de memórias, o ganho de codificação atende a um comportamento assintótico. A grande vantagem de tais códigos sobre os binários, é que, enquanto estes últimos precisam de várias memórias para que se tenha um ganho apreciável, os códigos  $Q$  *arias* necessitam de apenas uma memória para se conseguir praticamente todo o ganho. Mostrou-se também que para alguns  $Q$ 's, em particular, existem códigos não-lineares melhores que os lineares (uma linearização destes códigos seria possível às custas de uma nova rotulação dos pontos). No final do capítulo foram traçadas curvas de capacidade de canal, e o que se verificou é que o ganho máximo para uma mesma quantidade de informação não é maior que 0,2dB.

No capítulo 4 o conceito de particionamento de conjuntos baseado na equação de Diofanto é introduzido. À partir desta equação, mostra-se que o particionamento do reticulado  $Z^2$  para o caso binário, é apenas um caso particular de uma concepção muito mais ampla. A tabela 4.1, obtida por meios computacionais, fornece um particionamento mais geral ainda, incluindo casos em que não se tem solução da equação de Diofanto, e mesmo, casos em que há solução da equação, mas com a vantagem de que o novo particionamento apresenta um menor número de pontos vizinhos. Além do reticulado  $Z^2$ , estudou-se também o particionamento para o reticulado  $A_2$  e para o reticulado  $Z$ . Além disso, é mostrado que todos estes reticulados são um caso particular de uma equação mais geral, deixando nos antever a possibilidade do uso de tais equações no estudo de reticulados multi-dimensionais.

O capítulo 5 é similar ao capítulo 3, com a diferença que aqui o enfoque é dirigido para constelações QAM, isto é, na obtenção dos códigos ótimos e seus respectivos desempenhos para os reticulados  $Z^2$  e  $A_2$ . Mostrou-se também que o ganho de codificação atende a um comportamento assintótico, à medida que se eleva o número de elementos do corpo. Também neste caso, existem códigos não-lineares com desempenho superior aos lineares, para alguns  $Q$ 's em particular. Curvas de capacidade de canal foram traçadas, e o que se verificou é que o ganho máximo para uma mesma quantidade de informação pode ser da ordem de 1,0dB para o caso em que as constelações são pequenas.

Como consequência deste trabalho, foram apresentados [9] e [10] no 1990 IEEE International Symposium on Information Theory, em Janeiro de 1990.

# Apêndice A

## A.1 REGRA GENERALIZADA DOS MINTERMOS

Como se sabe a regra dos mintermos é muito usada em circuitos lógicos combinacionais binários, quando à partir da tabela verdade, ou seja, de um resultado conhecido, quer se determinar qual a função que realiza tal tabela.

O objetivo aqui é generalizar esta regra para  $Q > 2$ , desde que  $Q$  seja primo. O nosso interesse é usar a regra dos mintermos na determinação da matriz geradora de códigos convolucionais, que nada mais são que circuitos lógicos combinatório-sequenciais. No caso  $Q$  primo esta regra consiste em escrever a saída como soma de produtos de mintermos  $Q$  ários, onde cada mintermo está associado a uma entrada e pode ser expresso por:

$$\text{mint}(x = x_a) = [(Q - 1)!]^{-1} \frac{\prod_{i=0}^{Q-1} (x + i)}{x - x_a} \quad (\text{A.1})$$

Observe que da maneira como um mintermo foi definido, ele só é diferente de 0 para  $x = x_a$ . Portanto se quisermos escrever uma função de saída  $y$  em função das entradas  $r$  e  $s$ , de tal modo que quando  $r = r_a$  e  $s = s_a \Rightarrow y = y_a$ , então

$$y = y_a \cdot \text{mint}(r = r_a) \cdot \text{mint}(s = s_a)$$

**Exemplo A.1** Para o caso binário, usando a equação A.1 podemos ter os seguintes mintermos:

$$\begin{aligned} \text{mint}(x = 0) &= x + 1 \\ \text{mint}(x = 1) &= x \end{aligned}$$

Portanto se necessitarmos satisfazer a tabela verdade A.1, que corresponde à alocação de elementos da constelação à treliça da figura 3.6. Neste caso então as funções de saída  $y_i^1$  e  $y_i^0$  podem ser escritas como:

$$\begin{aligned} y_i^1 &= x_i^1(x_{i-1}^1 + 1) + (x_i^1 + 1)x_{i-1}^1 \\ y_i^0 &= (x_i^1 + 1)x_{i-1}^1 + x_i^1 x_{i-1}^1 \end{aligned}$$

$x_{i-1}^1$	$x_i^1$	$y_i^1$	$y_i^0$
0	0	0	0
0	1	1	0
1	0	1	1
1	1	0	1

Tabela A.1: Tabela verdade para o caso binário.

$x_{i-1}^1$	$x_i^1$	$y_i^1$	$y_i^0$
0	0	0	0
0	1	1	0
0	2	2	0
1	0	2	1
1	1	0	1
1	2	1	1
2	0	1	2
2	1	2	2
2	2	0	2

Tabela A.2: Tabela verdade para o caso ternário.

que simplificando com o uso das operações  $\text{mod}(2)$  convencionais chega-se a:

$$\begin{aligned} y_i^1 &= x_i^1 + x_{i-1}^1 \\ y_i^0 &= x_{i-1}^1 \end{aligned}$$

**Exemplo A.2** Para o caso ternário, usando a equação A.1, podemos ter os seguintes mintermos:

$$\begin{aligned} \text{mint}(x = 0) &= 2x^2 + 1 \\ \text{mint}(x = 1) &= 2x^2 + 2x \\ \text{mint}(x = 2) &= 2x^2 + x \end{aligned}$$

Portanto se necessitarmos satisfazer a tabela verdade A.2, que corresponde à alocação de elementos da constelação à treliça da figura 9.9. Neste caso então as funções de saída  $y_i^1$  e  $y_i^0$  podem ser escritas como:

$$\begin{aligned} y_i^1 &= (x_{i-1}^1 + 1)(x_{i-1}^1 + 2)x_i^1(x_i^1 + 1) + 2(x_{i-1}^1 + 1)(x_{i-1}^1 + 2)x_i^1(x_i^1 + 2) + \\ &2x_{i-1}^1(x_{i-1}^1 + 1)(x_i^1 + 1)(x_i^1 + 2) + x_{i-1}^1(x_{i-1}^1 + 1)x_i^1(x_i^1 + 2) + \\ &x_{i-1}^1(x_{i-1}^1 + 2)(x_i^1 + 1)(x_i^1 + 2) + 2x_{i-1}^1(x_{i-1}^1 + 2)x_i^1(x_i^1 + 1) \end{aligned}$$

$$\begin{aligned} y_i^0 &= x_{i-1}^1(x_{i-1}^1 + 1)(x_i^1 + 1)(x_i^1 + 2) + x_{i-1}^1(x_{i-1}^1 + 1)x_i^1(x_i^1 + 1) + \\ &x_{i-1}^1(x_{i-1}^1 + 1)x_i^1(x_i^1 + 2) + 2x_{i-1}^1(x_{i-1}^1 + 2)(x_i^1 + 1)(x_i^1 + 2) + \\ &2x_{i-1}^1(x_{i-1}^1 + 2)x_i^1(x_i^1 + 1) + 2x_{i-1}^1(x_{i-1}^1 + 2)x_i^1(x_i^1 + 2) \end{aligned}$$

que simplificando com as operações  $\text{mod}(3)$  convencionais chega-se a,

$$\begin{aligned} y_i^1 &= x_i^1 + 2x_{i-1}^1 \\ y_i^0 &= x_{i-1}^1 \end{aligned}$$

**Exemplo A.3** Para o caso quinário, usando a equação A.1 temos os seguintes mintermos:

$$\begin{aligned} \text{mint}(x = 0) &= 4x^4 + 1 \\ \text{mint}(x = 1) &= 4x^4 + 4x^3 + 4x^2 + 4x \\ \text{mint}(x = 2) &= 4x^4 + 3x^3 + x^2 + 2x \\ \text{mint}(x = 3) &= 4x^4 + 2x^3 + x^2 + 3x \\ \text{mint}(x = 4) &= 4x^4 + 2x^3 + x^2 + 3x \end{aligned}$$

# Apêndice B

## B.1 PROGRAMAS PRINCIPAIS

### B.1.1 Programa para cálculo da distância livre

Este programa tem por objetivo o cálculo da distância livre usando códigos convolucionais de uma memória usando constelações PAM  $Q$  *arias*.

Dados de entrada:

- Semente do gerador pseudo-aleatório (preferencialmente número ímpar de 7 dígitos)
- Número de elementos do campo
- Comprimento da palavra
- Distância livre alvo
- Código matricial

Dados de saída:

- Valor da distância livre encontrada
- Código que apresenta a distância livre

```

INTEGER D, R, S, T, Q, DF, AA, AB, AE, AG, SX, SY, Z, PA, PB, Y1, Y2, Y6, Y
INTEGER X1, X2, X6, X7, U1, U2, U3, D1, P, CW
DIMENSION D(400), R(400), S(400), T(400)
TYPE *, 'SEMENTE DO GERADOR PSEUDO ALEATORIO (IMPAR 7 DIGITOS
ACCEPT 99, IRD
11 TYPE *, 'NUMERO DE ELEMENTOS DO CAMPO'
ACCEPT 99, Q
N=1
M=1
INUT=0
TYPE *, 'COMPRIMENTO DA PALAVRA'
ACCEPT 99, L
TYPE *, 'DFREE ALVO'
ACCEPT 99, DF
DO 1 I=1,M
1 N=N*Q
TYPE *, 'CODIGO MATRICIAL '
ACCEPT 99 AA, AE, AB, AG
IA=AA
IB=AB
IE=AE
IG=AG
10 TYPE *, 'CODIF '
TYPE 99, IA, IE, IB, IG
CW=0
IF (IE.EQ.0 .AND. IG.EQ.0) GOTO 13
IF (IA.EQ.0 .AND. IB.EQ.0) GOTO 13
12 RND=RAN(IRD)
SX=INT(RND*Q)
SY=0
Z=1
DO 2 I=1,N
2 R(I)=50000
R(SY+1)=0
14 PA=SX+IA+SY*IB
PB=SX+IE+SY*IG
PA=PA-Q*INT(PA/Q)
PB=PB-Q*INT(PB/Q)
DO 3 IY=0,Q-1
Y1=IY*IB
Y2=IY*IG
Y6=Y1-Q*INT(Y1/Q)
Y7=Y2-Q*INT(Y2/Q)
DO 3 IX=0,Q-1

```

```

X1=IX*IA
X2=IX*IE
X6=X1-Q*INT(X1/Q)
X7=X2-Q*INT(X2/Q)
U1=X6+Y6-Q*INT((X6+Y6)/Q)
U2=X7+Y7-Q*INT((X7+Y7)/Q)
U3=IX+IY*Q
D1=ABS((U1-PA)*Q+U2-PB)
IF (D1.GT.INT(Q*Q/2)) D1=Q+Q-D1
3 D(U3+1)=D1+D1
  D(SY*Q+SX+1)=50000
  DO 5 P=0,Q-1
    DO 5 K=P*N,(P+1)*N-1
      J=K-N*INT(K/N)
      II=INT(K/Q)
      IF (P .NE. 0) GOTO 7
      S(J+1)=R(II+1)+D(K+1)
      GOTO 5
7   T(J+1)=R(II+1)+D(K+1)
   IF (T(J+1) .LT. S(J+1)) S(J+1)=T(J+1)
5   CONTINUE
   DO 8 I=1,N
8   R(I)=S(I)
   IF (Z .NE. 1) TYPE 99, R(SX+1)
   IF (Z .EQ. 1) TYPE 99, INUT
   Z=Z+1
   IF (Z .EQ. 2) GOTO 30
   IF (R(SX+1) .LT. DF) GOTO 9
   IF (Z .LE. L) GOTO 30
9   IF (R(SX+1) .GE. DF) GOTO 19
13  IG=IG+1
   IF (IG .LE. Q-1) GOTO 10
   IG=0
   IB=IB+1
   IF (IB .LE. Q-1) GOTO 10
   IB=0
   IE=IE+1
   IF (IE .LE. Q-1) GOTO 10
   IE=0
   IA=IA+1
   IF (IA .LE. 1) GOTO 10
   GOTO 11
19  CW=CW+1
   IF (CW .EQ. 10) GOTO 20

```

```

GOTO 12
20  TYPE *,'GRANDE DFREE'
    TYPE 99, NDF
    TYPE *,'CODIGO COM DFREE GRANDE'
    TYPE 99, IA, IE, IB, IG
    TYPE *,'NOVO DFREE ALVO'
    ACCEPT 99, DF
    IF (DF .EQ. 0) GOTO 11
    TYPE *,'MESMO CODIGO [1] CONTINUA SEQUENCIALMENTE [2]'
    ACCEPT 99, ISTR1
    IF (ISTR1-1) 10,10,13
30  SY= SX
    RND=RAN(IRD)
    SX=INT(Q*RND)
    IF (Z .EQ. 3 .AND. CW .EQ. 1) NDF=R(SY+1)
    IF (NDF .GT. R(SY+1)) NDF=R(SY+1)
    R(SY+1)=50000
    GOTO 14
99  FORMAT(7I7)
    END

```

### B.1.2 Programa que determina os melhores movimentos no reticulado $Z^2$

Este programa tem por objetivo a obtenção dos melhores movimentos em constelações no reticulado  $Z^2$ . Por melhores entenda-se os movimentos que apresentam a maior distância possível entre elementos do mesmo sub-conjunto.

Dados de entrada:

- Número de elementos do campo
- Distância quadrática alvo

Dados de saída:

- Valor da distância quadrática
- Movimento que apresenta uma distância igual ou superior à distância alvo

```

INTEGER A,B,D,Q,PX,PY
2  TYPE *,'NUMERO DE ELEMENTOS DO CAMPO'
   ACCEPT 99, Q
   TYPE *,'DISTANCIA QUADRATICA ALVO'
   ACCEPT 99, D
   DO 3 PY=1, INT(Q/2)
   DO 3 PX=PY, INT(Q/2)
   B=Q*Q
   DO 1 I=1,Q-1
   IX=PX*I
   IY=PY*I
   IX=IX-Q*INT(IX/Q)
   IY=IY-Q*INT(IY/Q)
   IF (IX .GT. INT(Q/2)) IX=Q-IX
   IF (IY .GT. INT(Q/2)) IY=Q-IY
   A=IX*IX+IY*IY
   IF (A .LT. B) B=A
1  CONTINUE
   IF (B .LT. D) GOTO 3
   TYPE *,'DISTANCIA OTIMA '
   TYPE 99, B
   TYPE *,'PASSO OTIMO '
   TYPE 99, PX, PY
   TYPE *,'NOVA DISTANCIA OTIMA'
   ACCEPT 99, D
3  CONTINUE
   GOTO 2
99  FORMAT(4I4)
   END

```

### B.1.3 Programa para cálculo da capacidade de canal

Este programa tem por objetivo o cálculo da capacidade de canal de constelações uni ou bi-dimensionais:

Dados de entrada:

- Número de elementos do campo
- Número de bits por símbolo
- Número de dimensões
- Par de coordenadas no caso de constelações irregulares
- Relação sinal-ruído

Dados de saída:

- Energia média da constelação
- Capacidade de canal

```

INTEGER Q,DM
REAL N, NQP, IR, II
DIMENSION AR(1000), AI(1000), IAR(1000), IAI(1000)
PI=4.*ATAN(1.)
1  TYPE *, 'NUMERO DE ELEMENTOS DO CAMPO '
   ACCEPT 99, Q
   TYPE *, 'NUMERO DE BITS POR SIMBOLO '
   ACCEPT 99, M
   N=Q**M
   TYPE *, 'NUMERO DE DIMENSOES '
   ACCEPT 99, DM
   EM=(N*N-1.)/12.
   IF(DM .EQ. 2) EM=0.
   AMR=0.
   AMI=0.
   DO 2 IA=1,N
   AR(IA)=(2*IA-1-N)/2
   AI(IA)=0.
   IF(DM .NE. 2) GOTO 2
   TYPE *, 'COORDENADAS REAL E IMAG '
   ACCEPT 99, IAR(IA), IAI(IA)
   AR(IA)=IAR(IA)
   AI(IA)=IAI(IA)
   EM=EM+(AR(IA)*AR(IA)+AI(IA)*AI(IA))/N
   AMR=AMR+AR(IA)
   AMI=AMI+AI(IA)
2  CONTINUE
   EM=EM-(AMR*AMR+AMI*AMI)/N/N
   TYPE *, 'ENERGIA MEDIA DA CONSTELACAO '
   TYPE 98, EM
5  TYPE *, 'INPUT S/N (DB) '
   ACCEPT 99, IRD
   IF(IRD .EQ. 99) GOTO 1
   RD=IRD
   RL=10.*(RD/10.)
   VA=EM/DM/RL
   SI=SQRT(VA)
   DE=SI/2
   C=0.
   DO 4 IA=1,N
   DO 4 II=(DM-1)*(-1-3*SI),(DM-1)*(1+3*SI).DE
   DO 4 IR=-1-3*SI,1+3*SI,DE
   PP=-1.*(IR*IR+II*II)
   PP=EXP(PP/2./VA)/((2.*PI*VA)**(DM/2.))

```

```
QP=0.  
DO 6 IB=1,N  
NQP=(AR(IA)+IR-AR(IB))**2+(AI(IA)+II-AI(IB))**2-IR*IR-II*II  
NQP=EXP(-NQP/2./VA)  
6 QP=QP+NQP  
4 C=C+PP*ALOG(QP)/ALOG(2.)  
C=-C*(DE**DM)/N+ALOG(N)/ALOG(2.)  
TYPE *,'CAPACIDADE DE CANAL '  
TYPE 98, C  
GOTO 5  
99 FORMAT(7I5)  
98 FORMAT(F9.3)  
END
```

# Bibliografia

- [1] Ungerboeck G. *Channel coding with multilevel/phase signals - IEEE Trans. on Inform. Theory - vol IT-28, pp 55-67, 1982*
- [2] Proakis, J. G. - *Digital Communications - Mc Graw Hill - 1983*
- [3] Lin, Shu & Costello Jr., Daniel J. - *Error Control Coding - Prentice Hall - 1983*
- [4] Benedetto S., Marsan M. A., Albertengo G., Giachin E. - *Combined Coding and Modulation: Theory and Applications - IEEE Trans. on Inform. Theory - vol IT-34, pp 223-236, 1988*
- [5] Denes J., Keedwell A. D. - *Frequency allocation for a mobile radio technique system - IEEE Trans. on Comm. - vol COM-36, pp 765,767, June 1988*
- [6] Calderbank A. R., Sloane N. J. A. - *New trellis codes based on lattices and cosets - IEEE Trans. on Inform. Theory - vol IT-33, pp 117-125, 1987*
- [7] Cohn H. - *Advanced Number Theory - Dover Publications Inc., New York, 1962*
- [8] Forney G. D. - *Coset codes I: Introduction and geometrical classification - IEEE Trans. on Inform. Theory - vol IT-34, pp 1123-1187, Sept 1988*
- [9] de Almeida C., Palazzo Jr. R. - *Binary quadratic form: a solution to the set partitioning over  $GF(q)$  - Trabalho apresentado no 1990 IEEE International Symposium on Information Theory, San Diego*
- [10] de Almeida C., Palazzo Jr. R. - *New trellis codes over  $GF(q)$  for one and two dimensional lattices - Trabalho apresentado no 1990 IEEE International Symposium on Information Theory, San Diego*