

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA
DEPARTAMENTO DE COMUNICAÇÕES

DETERMINAÇÃO E ANÁLISE DE ESTRUTURAS
RETICULARES PARA MODULAÇÃO POR CODIFICAÇÃO
EM TRELIÇA EM 2 E 4 DIMENSÕES

ORLANDO GEORGES CUNHA GONÇALVES

ORIENTADOR: Prof. Dr. REGINALDO PALAZZO JÚNIOR

61.07.166/BC
Tese apresentada à Faculdade de Engenharia
Elétrica da Universidade Estadual de Campinas
-UNICAMP, como parte dos requisitos exigidos
para obtenção do título de MESTRE EM
ENGENHARIA ELÉTRICA.

abril de 1990

Este exemplar corresponde à redação final da tese
defendida por Orlando Georges Cunha
Gonçalves e aprovada pela Comissão
Julgadora em 14/05/1990.

Reginaldo Palazzo Júnior
Orientador

aos meus pais Antônio e Celeste
aos meus irmãos Fernando e Fatinha

BANCA

Presidente:	Prof. Dr. Reginaldo Palazzo Júnior	(FEE)
	Prof. Dr. Rege R. Scarabucci	(FEE)
	Prof. Dr. João Frederico da Costa Meyer	(IMECC)
Suplentes:	Prof. Dr. Dalton Soares Arantes	(FEE)
	Prof. Dr. Edson Moschim	(FEE)

Agradecimentos

Agradeço ao Prof. Palazzo pela intensa dedicação, paciência, estímulo e pela confiança em mim depositada;

Aos professores da Faculdade de Engenharia Elétrica com os quais realizei cursos que foram de grande valia para realização deste trabalho. Em particular, agradeço aos Profs. Dalton e Borelli pelos excelentes cursos de Codificação ministrados, os quais despertaram em mim, um interesse maior por esta área;

Ao "Prof. Joni", pelo estímulo dado no início da minha formação acadêmica.

Agradeço aos elementos da banca pela atenção, sugestões e participação na leitura final deste trabalho;

Agradeço especialmente ao Stamatios e ao Nishi da "Publish" pelo suporte computacional, pela amizade e pelo equipamento utilizado na edição deste trabalho e à Lucinha pelo capricho dedicado aos desenhos.

Aos meus pais pelo carinho, e dedicação em toda a minha vida e por terem me proporcionado condições de estudar. Agradeço o apoio e amizade dos meus irmãos;

Aos meus amigos e colegas, sem os quais não teria a menor graça.

Este trabalho teve o suporte financeiro da CAPES-Coordenação de Amparo à Pesquisa e Ensino Superior, e do Convênio UNICAMP-TELEBRÁS.

Í N D I C E

RESUMO

12

ABSTRACT

14

INTRODUÇÃO

16

CAPÍTULO 1

ESTUDO DO PROCESSO DE CODIFICAÇÃO DE CANAL UTILIZANDO A
TÉCNICA DE MODULAÇÃO POR CODIFICAÇÃO EM TRELIÇA

I - INTRODUÇÃO 20

I.1 - Motivação 20

I.2 - Evolução do Sistema Convencional para o Sistema

TCM 22

II - DESCRIÇÃO FUNCIONAL DOS BLOCOS DE UM SISTEMA	
DE COMUNICAÇÕES CONVENCIONAL	26
II.1 - Modulador/Demodulador	26
II.2 - Codificador de Canal	28
II.2.1 - Condições Gerais de Desempenho e	
Estratégias	30
II.3 - Codificador Convolutacional	32
II.4 - Decodificador de Máxima Verossimilhança:	
Algoritmo de Viterbi	36
II.4.1 - Processo Recursivo	41
II.4.2 - Evento de Erro	42
III -TCM - MODULAÇÃO POR CODIFICAÇÃO EM TRELIÇA	43
III.1 - Introdução	43
III.2 - O Processo de Codificação	45
III.3 - Mapeamento por Partição de Conjunto	47
III.4 - Regras de Codificação	48
III.5 - Desempenho dos Códigos de Treliça	58
IV - CONCLUSÕES	62

.....

C A P Í T U L O 2

ANÁLISE DE ESTRUTURAS RETICULARES PARA MODULAÇÃO POR
CODIFICAÇÃO EM TRELIÇA EM 2 DIMENSÕES

I - INTRODUÇÃO	64
II - ESTRUTURAS RETICULARES	66
II.1 - Empacotamento Esférico e seus parâmetros	68
II.2 - Reticulado Quadrado ($\mathbb{Z}^2$)	71
II.2.1 - Constelações Propostas em $\mathbb{Z}^2$	73
II.3 - Reticulado Hexagonal (A_2)	78
II.3.1 - Constelações Propostas em A_2	80
II.4 - Partição de Conjunto de Sinais	86
II.4.1 - Regras Para Partição de Conjuntos	90
II.4.2 - Partição de Constelações tipo $\mathbb{Z}^2$	91
II.4.3 - Partição de Constelações tipo A_2	98
III - CONCLUSÕES	111

.....

CAPÍTULO 3

DETERMINAÇÃO E ANÁLISE DE CODIFICADORES TCM ÓTIMOS PARA
RETICULADOS EM 2 DIMENSÕES

I - INTRODUÇÃO	113
II - ANÁLISE ALGÉBRICA	114
III - CONSIDERAÇÕES SOBRE A PROCURA EFETUADA [30]	127
III.1 - Representação por Vetores Coeficientes	130
IV - ANÁLISE DOS RESULTADOS	150
IV.1 - TAXA $\frac{2}{3}$	150
IV.2 - TAXA $\frac{3}{4}$	153
IV.3 - TAXA $\frac{4}{5}$	154

.....

C A P Í T U L O 4

ANÁLISE DE ESTRUTURAS RETICULARES PARA MODULAÇÃO POR
CODIFICAÇÃO EM TRELIÇA EM 4 DIMENSÕES

I - INTRODUÇÃO	159
II - ESTRUTURAS RETICULARES EM 4-D	160
II.1 - $\mathbb{Z}^4$	160
II.2 - $\mathbb{A}_4$	164
II.3 - $\mathbb{D}_4$	166
II.4 - $\mathbb{A}_2 \times \mathbb{A}_2$	172
III - TCM MULTIDIMENSIONAL	175
III.1 - TCM EM 4 DIMENSÕES	177
III.1.1 - PARTIÇÃO DE CONJUNTO DE $\mathbb{Z}^4$	178
III.1.2 - PARTIÇÃO DE CONJUNTO DE $\mathbb{A}_2 \times \mathbb{A}_2$	189
IV - CONCLUSÕES	198

.....

CAPÍTULO 5

CONCLUSÕES _____ 202

.....

APÊNDICE 1

CODIFICADORES EQUIVALENTES

RETICULADOS Z^2 e A_2

TAXAS $\frac{2}{3}, \frac{3}{4}, \frac{4}{5}$

.....

APÊNDICE 2

PROGRAMAS E RESULTADOS

(d) - PROGRAMA READA

(u) - PROGRAMA DELTA0A

(uu) - PROGRAMA READC

(w) - PROGRAMA DELTA0C

.....

REFERÊNCIAS BIBLIOGRÁFICAS

_____ 242

R E S U M O

Os códigos convolucionais assim como os códigos de treliça empregados no sistema TCM (Modulação por Codificação em Treliça) ainda não apresentam uma estrutura matemática bem definida, apesar do contínuo esforço neste sentido. Este trabalho procura, através do posicionamento dos sinais no espaço de sinais, caracterizar determinadas estruturas reticulares, que pelas propriedades que apresentam, nos permitem inferir sobre os conjuntos de sinais mais indicados a serem utilizados no sistema TCM e da mesma forma quantificar o desempenho alcançado quando da utilização dos mesmos.

Em primeiro lugar, tratamos de apresentar o estudo do processo de codificação de canal, utilizando a técnica TCM (forma combinada de codificação e modulação) para transmissão digital sobre canais de faixa limitada.

Em segundo lugar, foi feito um estudo de conjuntos de sinais gerados a partir de duas estruturas reticulares bi-dimensionais (reticulado Quadrado Z^2 , e reticulado Hexagonal A_2), e o comportamento destes tipos de sinais quando do emprego no sistema TCM.

Em seguida foi feita uma procura por codificadores convolucionais — não catastróficos, mínimos e que geram códigos TCM ótimos — utilizando conjuntos de sinais gerados em cada um dos reticulados apresentados. Estes resultados são tabelados e em seguida é feita uma análise comparativa de desempenho entre os esquemas TCM para várias taxas de codificação.

No quarto Capítulo são introduzidas algumas estruturas reticulares em 4 dimensões (Z^4 , A_4 , D_4 e $A_2 \times A_2$) que servirão para projeção de códigos TCM 4-D. É também descrita a Partição de Conjunto de sinais definidos em Z^4 e $A_2 \times A_2$.

A partir dos resultados obtidos e de dois novos conceitos ($FLDA$ e GCA_{max}) permitimo-nos comparar as estruturas estudadas e direccionar o sentido de futuras pesquisas nesta área.

A B S T R A C T

As it happens with Convolutional Codes, Trellis Codes Modulation (TCMD) still lacks an algebraic structure in spite of the efforts made by many researchers. This work, aims at characterizing lattices structures with the purpose of specifying proper locations of the signal points in a constellation such that the best performance may be achieved by a communications system using this TCM scheme.

First of all, it is presented as an introductory material, the channel coding process when using TCM schemes in digital transmission over band limited channels.

Second of all, it is presented an analysis of the signal sets taken from bi-dimensional lattices structures (lattices $\mathbb{Z}^2$ and A_2) for use in TCM schemes.

As a consequence, search for non catastrophic convolutional codes leading to good TCM schemes under the proposed lattices is realized. These results are tabulated and a performance analysis of some TCM schemes is considered.

Some lattice structures in 4-dimensional space ($\mathbb{Z}^4$, A_4 , D_4 , $A_2 \times A_2$) which will be used as possible candidates for TCM schemes is introduced. Moreover, it is described a set partitioning rule for the $\mathbb{Z}^4$ and $A_2 \times A_2$ lattice structures.

From the achieved results and the introduction of two measuring parameters it was possible to compare the lattice structures and to point out prospective research in this area.

INTRODUÇÃO

Nos sistemas de comunicação digital via cabo telefônico, rádio ou satélite, tanto a potência de transmissão, quanto a largura de faixa espectral dos canais são limitadas. Assim, para atender a uma crescente demanda de serviços de comunicação e satisfazer às exigências de confiabilidade e segurança da informação (imunidade quanto à ação de ruídos e outras interferências), são necessárias técnicas de transmissão com eficiência cada vez maior.

Neste trabalho, abordamos o problema da codificação de canal (aumentar a confiabilidade da transmissão sem a eventual expansão de faixa) através da técnica TCM — Modulação por Codificação em Treliça.

No sistema TCM o codificador e o modulador são tratados como uma entidade única.

Os códigos projetados por esta técnica levam em conta a máxima distância Euclideana entre as seqüências de sinais transmitida/estimada, e a razão para isto, se deve ao fato de que a redundância necessária para o processo de codificação deva vir da expansão do conjunto de sinais, evitando desta forma uma expansão de faixa. Desta forma, quanto maior esta distância melhor o desempenho do sistema.

Esta técnica, tira proveito de um mapeamento específico da saída de um codificador convolucional otimizado em termos de distância Euclideana, em uma constelação de sinais oriunda do espaço Euclideano de sinais.

Interessa-nos portanto, encontrar constelações de sinais que apresentem a maior distância possível entre os seus sinais.

Para abordar este problema, optamos por caracterizar o espaço Euclideano de sinais de forma que este seja gerado por estruturas reticulares, que definem o posicionamento dos pontos de sinais entre si, de maneira regular e exata.

Desta forma, nos é permitido quantificar as constelações de sinais através da distância entre os mesmos, obedecendo sempre a uma quantidade restritiva de energia média de transmissão.

Neste trabalho, procura-se encontrar, tanto para as estruturas geradas em 2 quanto em 4 dimensões, o maior distanciamento possível entre os sinais, seja por características inerentes às estruturas, seja pelo método de particioná-las em subconjuntos de distâncias crescentes para posterior codificação. Desta forma, obtem-se ganhos substanciais quando comparado aos métodos convencionais de modulação multi-nível não codificados.

C A P Í T U L O 1

ANÁLISE DO PROCESSO DE CODIFICAÇÃO DE CANAL
UTILIZANDO A
TÉCNICA DE MODULAÇÃO POR CODIFICAÇÃO EM TRELIÇA

I - INTRODUÇÃO

O objetivo deste Capítulo é de apresentar o estudo do processo de codificação de canal, utilizando-se a técnica proposta por Ungerböeck [5], que combina o processo de codificação e o de modulação, a serem utilizados na transmissão de sinais, como uma entidade única. Esta técnica denominada Modulação por Codificação em Treliça (Trellis Coded Modulation - TCMD, reúne em torno de si as propriedades de eficiência em potência e em faixa, características fundamentais para seu emprego.

I.1- MOTIVAÇÃO

A técnica TCM vem sendo objeto de estudo nos últimos anos por inúmeros pesquisadores tanto de Universidades bem como de renomados centros de pesquisa. Mais recentemente, o estudo desta técnica vem sendo desenvolvida nos departamentos de Telemática e Comunicações da FEE-UNICAMP.

A modulação por codificação em treliça tem provado ser uma técnica recompensadora tanto nos aspectos teóricos quanto nos

aspectos práticos, pois a mesma pode proporcionar ganhos assintóticos de até 6 dB na relação sinal/ruído, quando comparado com os sistemas convencionais de modulação, obtendo desta forma desempenhos próximos daquele estabelecido por Shannon para a capacidade de transmissão de informação.

A principal característica inerente a esta técnica é a redução da taxa de erro de transmissão, isto é, para uma mesma relação sinal/ruído, um modem TCM, operando com altas taxas, fornece uma significativa vazão (throughput) por reduzir a taxa de erros de transmissão várias ordens de magnitude. Por exemplo, um modem TCM pode requerer retransmissão somente uma vez a cada 10.000 blocos ao passo que um modem convencional requer retransmissão a cada 10 blocos [19].

Dadas as características acima, o emprego da técnica TCM passa a assumir um papel relevante, uma vez que o acesso às estruturas de redes é concebido visando o atendimento de uma multitude de serviços de interesse do usuário e que as condições impostas quanto à qualidade destes mesmos serviços são bastantes restritivas.

I.2 - EVOLUÇÃO DO SISTEMA CONVENCIONAL PARA O SISTEMA TCM

Nos sistemas clássicos de comunicação digital as funções desempenhadas pelo modulador e codificador (ver Fig.1.2.1), são consideradas independentemente.

Nos sistemas de modulação M-ária convencionais, durante cada intervalo de T seg de duração de um símbolo, o modulador mapeia m dígitos binários em um dos $M = 2^m$ possíveis sinais enquanto que o demodulador recupera os m correspondentes dígitos binários fazendo independentes decisões através do sinal recebido.

Codificadores e decodificadores convencionais utilizam para correção de erros, códigos com símbolos binários ou M-ários, transmitidos em um canal discreto. Para um código de taxa $\frac{k}{n} < 1$, $(n - k)$ símbolos redundantes são adicionados a cada k símbolos de informação com o objetivo de detectar ou corrigir possíveis erros oriundos de interferências do meio de transmissão. Para canais discretos o processo de máxima verossimilhança conduz ao critério de otimalidade conhecido como distância de Hamming (número de símbolos em que duas seqüências codificadas diferem) que é uma medida apropriada de distância para a decodificação. A distância mínima de Hamming, tanto para códigos lineares como para não lineares, garante que o decodificador pode corrigir pelo

menos $[(dH_{\min} - 1)/2]$ padrões de erro. Portanto, quanto maior esta distância, melhor o código.

Para superar as limitações do sistema de modulação, quando este opera com baixos valores de relação sinal/ruído, deve-se combiná-lo com um processo de codificação. Esta forma combinada pode assumir as seguintes situações: 1- Codificação/Modulação independentes e 2- Codificação/Modulação dependentes. Como o processo de codificação adiciona bits redundantes aos bits de informação, ocorrerá um decréscimo na taxa de informação, porém a habilidade de correção de erros justifica esta redução de taxa.

Para compensar esta redução de taxa existem duas alternativas:

(i) Manter T fixo e aumentar o número de bits com um conseqüente aumento do número de sinais (forma combinada independente).

(ii) Manter fixos T e o número de bits porém aumentando o número de sinais (forma combinada dependente).

No segundo caso, existe a necessidade de utilizar-se uma modulação M -ária, porém, nos sistemas clássicos onde a modulação e a codificação são tratadas de maneira independente, os resultados esperados dependem fortemente dos códigos sendo utilizados. A justificativa para tal fato advém do seguinte argumento:

Uma vez que se aumente o número de bits e conseqüentemente a cardinalidade do conjunto de sinais, porém, mantendo a energia média constante implica que estes sinais estão mais próximos uns dos outros (em termos de distância Euclideana) levando à ocorrência de erros mais freqüentes. Assim, códigos com capacidade de correção de erros elevada devem ser utilizados para mitigar o efeito da redução da distância Euclideana.

Uma tentativa de solução deste problema seria então o emprego de decisão suave no processo de decodificação da seqüência dos sinais. Neste caso, os erros mais freqüentes ocorrem entre pares de seqüências, transmitida-estimada, apresentando distâncias Euclidianas próximas. Mesmo assim, teríamos que considerar o seguinte fato:

O mapeamento da saída de um codificador convolucional otimizado em termos de distância de Hamming, para uma modulação *M-ária* não implica em distâncias Euclidianas otimizadas. Isto sem dúvida foi um dos motivos que dificultaram a determinação de códigos ótimos para modulação multinível.

Foi tentando solucionar este problema que Ungerböeck [5] propôs a técnica TCM. A motivação para tal proposta adveio de um trabalho que utilizava o algoritmo de Viterbi para detecção de sinais em presença de interferência intersimbólica. Os resultados decorrentes desta investigação apontavam para o fato da importância de se utilizar diretamente a distância Euclideana entre seqüências de sinais ao invés da distância de Hamming como um passo final.

A razão pela qual os códigos deveriam ser projetados com a máxima distância Euclideana, ao invés da distância de Hamming, é que a redundância necessária para o processo de codificação deveria vir da expansão do conjunto de sinais, evitando desta forma uma expansão de faixa.

Como decorrência desta proposição, resultaram esquemas simples de treliça com quatro estados apresentando "imunidade" ao ruído Gaussiano aditivo quando comparado com a modulação não codificada convencional. Esquemas mais complexos proporcionam ganhos assintóticos de codificação de 6 dB ou mais, sem expansão de faixa ou redução da taxa de informação efetiva (requerida nos esquemas de correção de erro tradicionais).

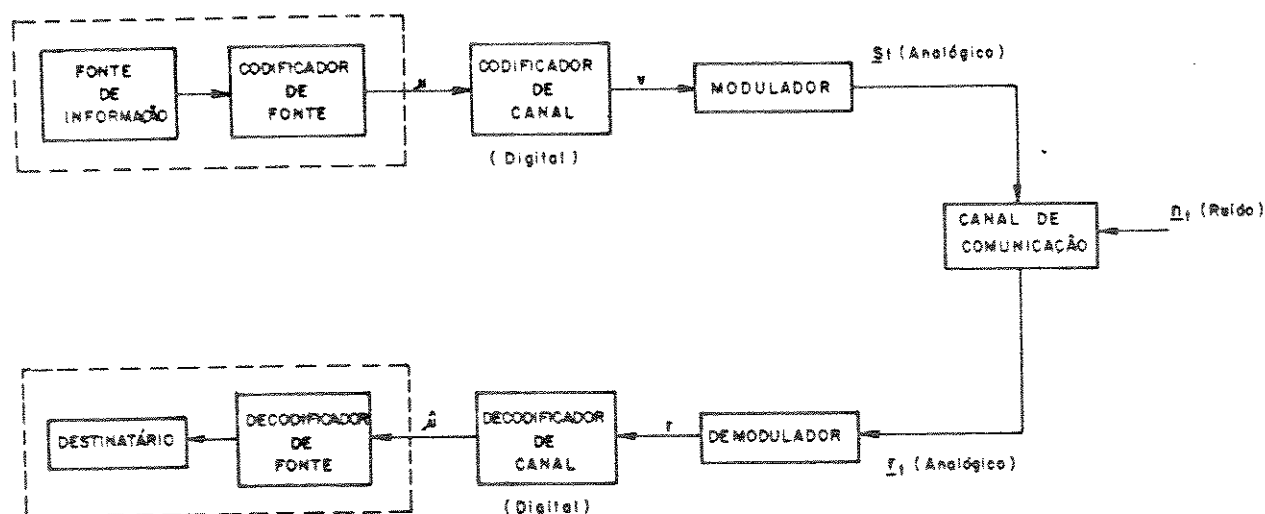


Fig. 1.2.1 — Modelo de Sistema de Comunicação

II - DESCRIÇÃO FUNCIONAL DOS BLOCOS DE UM SISTEMA DE COMUNICAÇÕES

CONVENCIONAL

Com o objetivo de facilitar o entendimento do material contido na seção III é que descreveremos com certo detalhe alguns dos blocos que compõem um sistema de comunicações convencional.

II.1- MODULADOR/DEMULADOR:

O modulador aceita um fluxo de bits como entrada e converte-os em uma forma-de-onda apropriada para transmissão no canal de comunicação. Dessa forma, os parâmetros importantes do modulador são as formas-de-onda utilizadas, a duração das mesmas, o nível de potência e a largura de faixa utilizada.

A forma apropriada para transmissão pela qual o modulador está encarregado visa proporcionar uma troca eficiente da relação sinal/ruído com faixa, ou seja proporcionar uma certa imunidade ao ruído. Assim, embora a idéia de se aumentar as quantidades de potência do sinal e largura de faixa para combater os efeitos do

Capítulo 1

ruído seja um método óbvio, estes parâmetros não podem ser aumentados indefinidamente devido às limitações do canal e do equipamento.

Por outro lado, o número de formas-de-onda distintas que o modulador pode gerar aumenta exponencialmente com o número de bits que se quer transmitir, acarretando um aumento na complexidade do receptor. Assim sendo, não é possível aumentar indiscriminadamente a quantidade de bits a ser transmitida.

Para uma dada característica modulador-canal-demodulador a taxa de erro de bit média teórica pode ser determinada. Entretanto, taxas de erro de bit inferiores podem ser obtidas pela inserção de códigos corretores de erros.

II.2 - CODIFICADOR DE CANAL

Na análise de desempenho de sistemas de comunicações, via probabilidade de erro, consideram-se como variáveis a relação sinal/ruído na entrada do receptor e a taxa de dados. Na prática, entretanto, um razoável número de variáveis é considerado. Por exemplo: 1) a potência máxima do sinal e a largura de faixa são restritos a valores fixados através de regulamentações governamentais ou de companhias privadas; 2) em certas aplicações existe a restrição do nível de potência do ruído permitido; 3) para a minimização dos custos e da complexidade dos equipamentos existe a restrição quanto ao tipo de sinais a serem utilizados bem como à sua quantidade.

Com todas estas restrições, não é freqüente chegar-se a um esquema de modulação que leve a uma probabilidade de erro aceitável, para uma dada aplicação. Em face a este problema, uma alternativa prática para reduzir a probabilidade de erro é a utilização de códigos corretores de erro.

A codificação digital de canal é um método prático de se realizar alta taxa de transmissão com segurança e eficiência através da utilização de um conjunto relativamente pequeno de sinais para a transmissão. Desta forma a tarefa do demodulador fica bastante facilitada visto que terá que distinguir entre um

número pequeno de formas-de-onda previamente conhecidas.

O controle de erros é realizado pelas operações de codificação do canal, que consiste de uma forma sistemática de adicionar bits adicionais à saída do codificador de fonte. Uma vez que os bits redundantes não contêm informação, eles possibilitam ao receptor detectar e/ou corrigir possíveis erros introduzidos pelo meio de transmissão.

Existem duas classes de códigos que podem ser utilizados para realizar a operação de codificação de canal. A primeira classe, chamada de Códigos de Bloco, é tal que o codificador utiliza blocos de k bits de informação da fonte e adiciona r bits de controle de erro. O número de bits de controle de erro adicionados dependerá do valor de k e da capacidade de controle de erro desejada. A segunda classe, que está diretamente relacionada a este trabalho, é a dos Códigos Convolucionais, onde a informação é codificada continuamente intercalando bits de informação e bits de controle de erro.

Ambos os métodos requerem armazenamento e velocidades de processamento de dados relativamente altas tanto no codificador como no decodificador. Embora estes requisitos tenham sido fator limitante há algum tempo atrás, hoje a disponibilidade de memórias RAM e micro-processadores dedicados com altas velocidades de processamento a preços razoáveis viabilizam sistemas de comunicação bastante sofisticados.

II.2.1 - CONDIÇÕES GERAIS DE DESEMPENHO E ESTRATÉGIAS

A Figura II.2.1, apresenta um sistema de comunicação digital. Este sistema será utilizado para transmissão de dados $\langle b_k \rangle$ de uma fonte binária sobre um canal ruidoso com capacidade C , a uma taxa $R = r_b$ bits/seg. Devido ao ruído do canal, a sequência de bits $\langle \hat{b}_k \rangle$ estimada no receptor pode diferir ocasionalmente da sequência transmitida $\langle b_k \rangle$. É desejado que a probabilidade de erro $P(\hat{b}_k \neq b_k)$ seja menor que algum valor pré-fixado p . Vejamos então como o processo de codificação pode auxiliar para que $P(\hat{b} \neq b) < p$.

O codificador e o decodificador de canal são blocos funcionais no sistema que interagem no sentido de reduzir a probabilidade de erro global. O codificador divide as mensagens de entrada em blocos de k bits e substitui cada bloco de k bits por uma palavra código de comprimento n através da adição de $(n-k)$ bits de paridade. O decodificador opera sobre a palavra recebida, de tal forma a reproduzir os k bits da mensagem original. O projeto do codificador e do decodificador consiste então em selecionar regras para gerar palavras-código correspondentes às mensagens e extrair das versões recebidas das palavras-código, as mensagens.

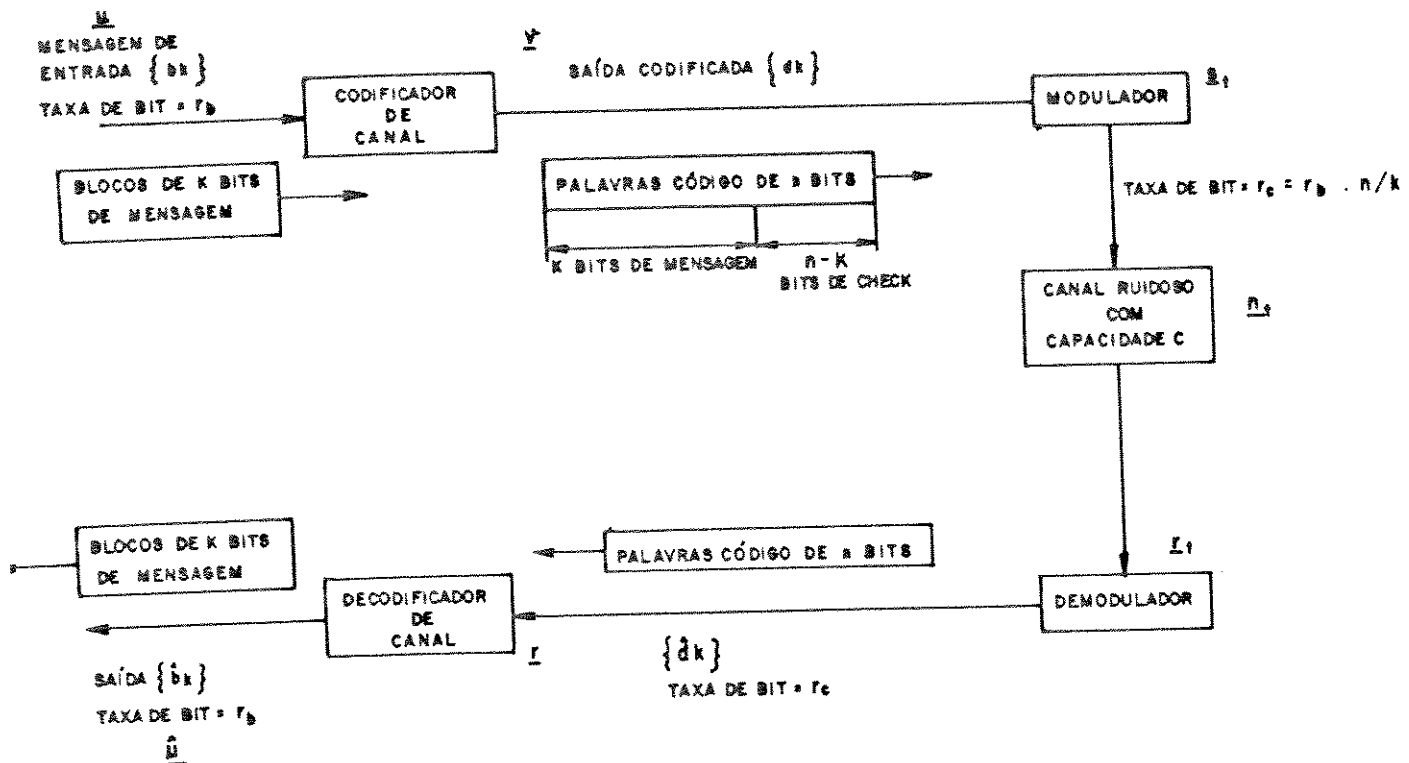


Fig. II.2.1.1 - Codificação de canal. Probabilidade de erro de bit na mensagem é $P_b = P \{ \hat{b}_k \neq b_k \}$ e $r_c < C$

Como podemos ver na Figura II.2.1.1, uma vez que o decodificador recebe somente símbolos discretos, o número de símbolos em que duas seqüências codificadas diferem implica em uma medida adequada de distância para a decodificação. Esta medida é denominada distância de Hamming. A menor destas distâncias é dita distância mínima de Hamming.

O projeto do codificador deve seguir no sentido de obter códigos com a maior distância mínima de Hamming possível com o compromisso de não acrescentar um número muito grande de bits de paridade.

II.3 - CODIFICADOR CONVOLUCIONAL

Em um código convolucional $[20], [21], [22], [23]$, um bloco de n dígitos codificados, gerados pelo codificador em uma unidade de tempo, não depende somente do bloco de k dígitos de mensagem desta unidade de tempo, mas também dos m blocos precedentes, isto é, o código tem memória. Um código convolucional (n, k, m) , onde k é o tamanho do bloco, n é o tamanho da palavra-código ramo e m é o número de memórias, pode ser implementado por um circuito seqüencial linear com k entradas, n saídas e m memórias. Usualmente os valores de n e k são pequenos, com $k < n$. Para um valor fixo da relação sinal/ruído, quanto maior o valor de m menor é a probabilidade de erro. Quando m cresce também cresce exponencialmente a complexidade de decodificação. No caso de se utilizar o processo de estimação de seqüências pela máxima verossimilhança (MLSE), ou seja, o algoritmo de Viterbi, o valor de m deve estar em torno de 10.

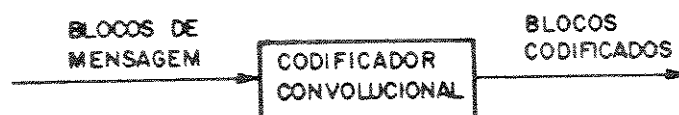


FIG. II 3.1. - Codificador convolucional

Todo codificador convolucional pode ser implementado utilizando-se m registros de deslocamento com alimentação unidirecional, n somadores módulo-2 (portas ou-exclusivo) e um multiplexador para serialização das saídas do codificador.

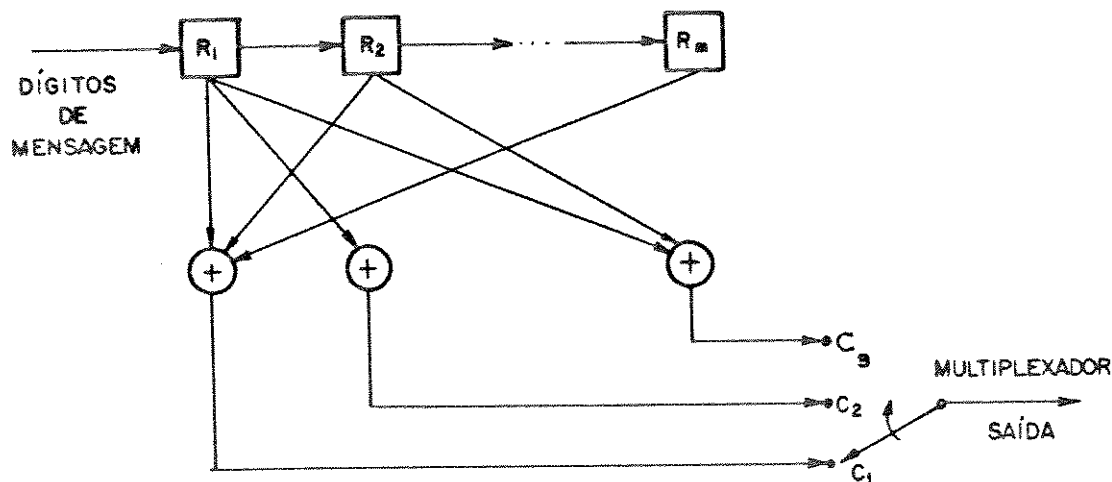


FIG. II.3.2 - Exemplo de um codificador convolucional para $K=1$, $n=3$ e m genérico

A operação de codificação, (ver Figura II.3.2) segue o seguinte procedimento:

Assume-se que os registradores de deslocamento estão vazios inicialmente. O primeiro bit da entrada de dados alimenta R_1 . Durante este intervalo de bit, o multiplexador amostra as saídas dos somadores módulo-2, c_1, c_2, c_3 . Assim um simples bit de dados produz $n = 3$ bits de saída. O próximo bit de mensagem na

Capítulo 1

seqüência de entrada alimenta R_1 , enquanto o conteúdo de R_1 é transferido para R_2 e o multiplexador novamente amostra as saídas dos três somadores. O processo se repete até que o último bit de dados seja transportado para R_m e em seguida m zeros são introduzidos seqüencialmente.

Deve-se enfatizar que o codificador convolucional opera nos dados de forma contínua, assim um codificador convolucional exige pouco hardware para armazenamento de dados. Outro ponto importante é que cada bit de dados influencia $m \times n$ dígitos, onde m é o número de registros de deslocamentos e n é o número de saídas do codificador.

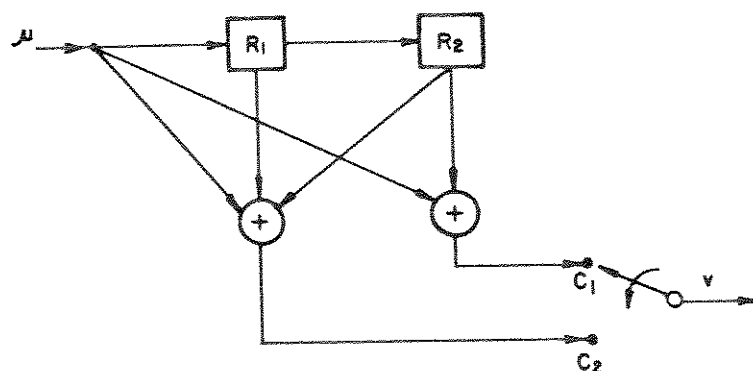


FIG. II.33. - Exemplo de um codificador convolucional
 $(n, k, m) = (2, 1, 2)$

Como um codificador convolucional é um circuito seqüencial, este pode ser representado por um diagrama de estados (veja Figuras II.3.3 e II.3.4). O estado de um codificador é definido pelo conteúdo dos registros de deslocamento. Em um código (n,k,m) com $k > 1$, cada bloco de k entradas, causa uma transição para um dos 2^{km} estados. Portanto, existem 2^k ramos partindo de cada estado onde cada um destes ramos corresponde a diferentes blocos de dados. Assumindo-se que o codificador esteja inicialmente no estado S_0 (registros vazios), uma palavra código está associada a um caminho através do diagrama de estados determinada pela seqüência de dados. O diagrama de estados facilita a elaboração de um outro diagrama, o de treliça, que é útil para a visualização do processo de decodificação, como veremos a seguir.

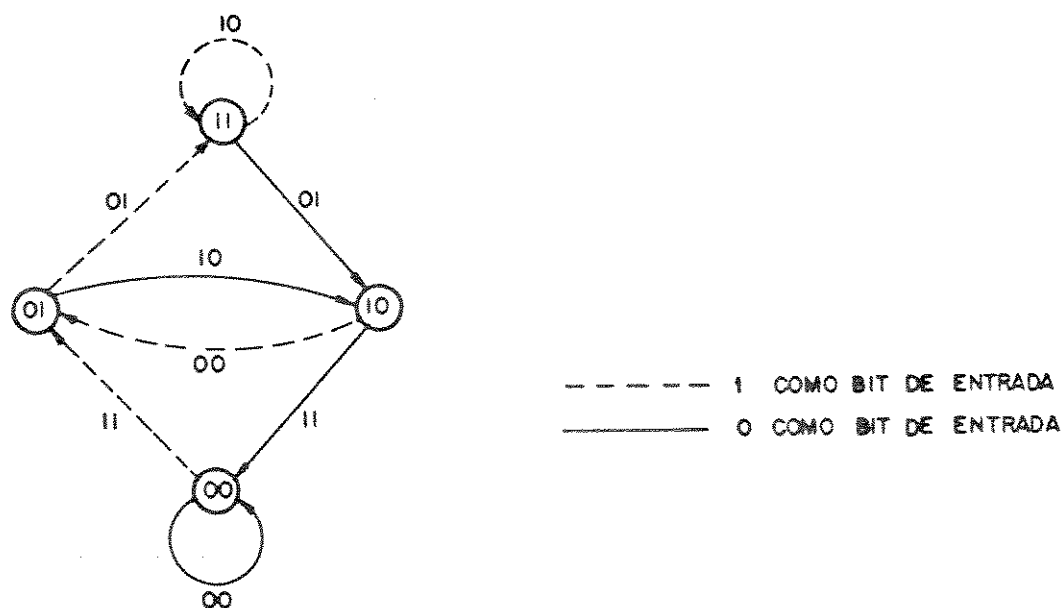


FIG.II.3.4 - Diagrama de estados para o codificador convolucional da Fig II.3.3

II.4 - DECODIFICADOR DE MÁXIMA VEROSSIMILHANÇA: ALGORITMO DE VITERBI

Um dos métodos de decodificação para códigos convolucionais é aquele que emprega o Algoritmo de Viterbi (A.V.). Este algoritmo consiste basicamente na realização da estimação de seqüências através do critério de máxima verossimilhança resultando portanto na menor probabilidade de erro de bit [24].

Sucintamente, o algoritmo de Viterbi opera da seguinte forma: seja r_i a seqüência recebida e s_j a seqüência transmitida. O critério de máxima verossimilhança consiste em escolher uma seqüência s_k a partir de r_i tal que:

$$P[r_i/s_k] > P[r_i/s_j] \quad \text{para todo } j \neq k \quad (\text{II.4.1})$$

onde $P[./.]$ é a probabilidade condicional. O objetivo do algoritmo de Viterbi é então maximizar o processo de escolha de s_k .

Para um melhor entendimento do algoritmo de Viterbi é conveniente utilizar a representação do diagrama de estados evoluindo no tempo, ou seja, o diagrama de treliça, como consequência de ser uma máquina de estados finito, (Fig. II.4.1).

Capítulo 1

Neste diagrama, os nós representam os estados (conteúdo dos registradores de deslocamento do codificador convolucional) e os ramos da treliça (transição de estados) contêm a informação codificada. Iremos representar por uma linha pontilhada a informação " 1 " e por uma linha sólida a informação " 0 ". Suponha que queiramos transmitir a sequência " 1 1 0 1 0 ", utilizando o codificador da Fig. II.3.a. A sequência codificada pode ser observada através do diagrama de treliça (Fig. II.4.1).

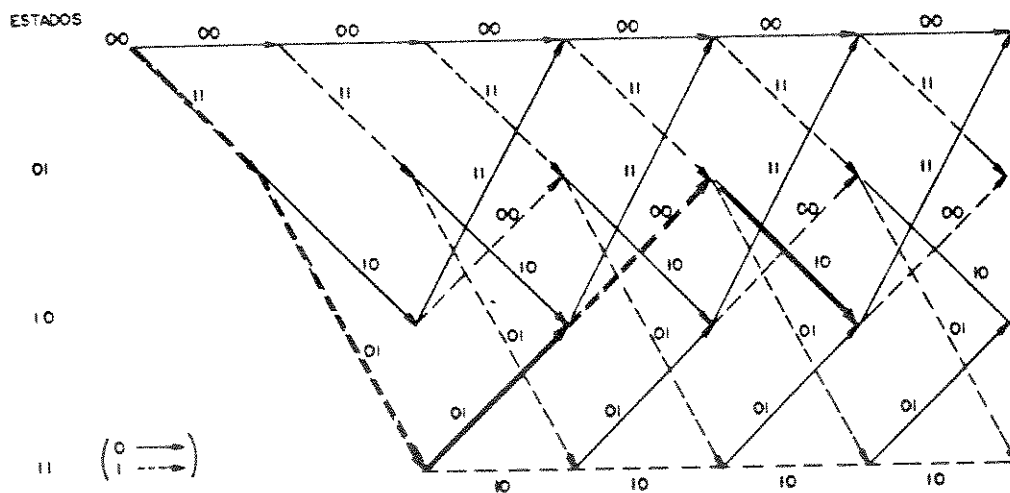


FIG II.4.1 - Diagrama de treliça para o codificador (2,1,2) da figura II.3.3.
As palavras-ramo codificadas correspondentes à sequência de informação "11010" estão destacadas no diagrama.

Suponha que o processo de codificação se inicia a partir do estado 00. O primeiro dígito de informação a ser codificado é o bit " 1 " (linha pontilhada). Este dígito de informação alimenta o circuito codificador de tal forma que o novo estado (conteúdo dos registros) passa a ser 01, e a resposta do circuito a esta entrada é " 11 " (palavra-ramo codificada, rotulada sobre as transições no diagrama). A linha correspondente ao próximo dígito de informação da sequência resulta na palavra-ramo codificada, " 01 ", e assim sucessivamente. Portanto, " 11 01 01 00 10 ... " é a sequência codificada correspondente a sequência de informação "1 1 0 1 0 ...".

Devido a restrições de implementação, o algoritmo de Viterbi é economicamente viável se o codificador convolucional contiver até 10 unidades de memória.

O algoritmo, quando aplicado à sequência recebida r de um canal DMC (Canal Discreto Sem Memória), busca uma sequência s através da treliça com a maior métrica (isto é, a sequência de máxima verossimilhança). O algoritmo processa a sequência r de uma maneira iterativa. A cada passo no processo de decodificação o A.V. compara as métricas acumuladas de todos os caminhos entrando em cada estado e retém aquelas que apresentem a maior métrica acumulada em cada estado. Estes caminhos são chamados sobreviventes.

Para um canal DMC a métrica utilizada no A.V. resulta em

$$d_{HK}(r_i, s_j) < d_{HK}(r_i, s_k) \text{ para todo } j \neq k \quad (\text{II.4.1})$$

onde $d_{HK}(\dots)$ é a distância de Hamming, entre duas quaisquer seqüências de mesmo comprimento.

Para facilitar o entendimento, a Figura II.4.2 apresenta um exemplo de utilização do algoritmo de Viterbi para o diagrama de treliça da Fig. II.4.1, corrigindo o padrão de erro "1 1 0 0 0".

Suponha que uma seqüência de zeros tenha sido transmitida, isto é, " 00 00 00 00 ... " e " 11 00 00 00 ... " é a correspondente seqüência recebida. O A.V. recursivamente realiza a decodificação passo-a-passo calculando os sobreviventes em cada nó, através da comparação das distâncias de Hamming acumuladas, entre as palavras-ramo recebidas e todas as palavras-ramo do código geradas no receptor. Neste exemplo, depois de 9 passos todos os caminhos sobreviventes iniciaram a divergência do caminho correto a partir do estado zero. Dessa forma, os dois erros são corrigidos e o bloco original é enviado ao destinatário.

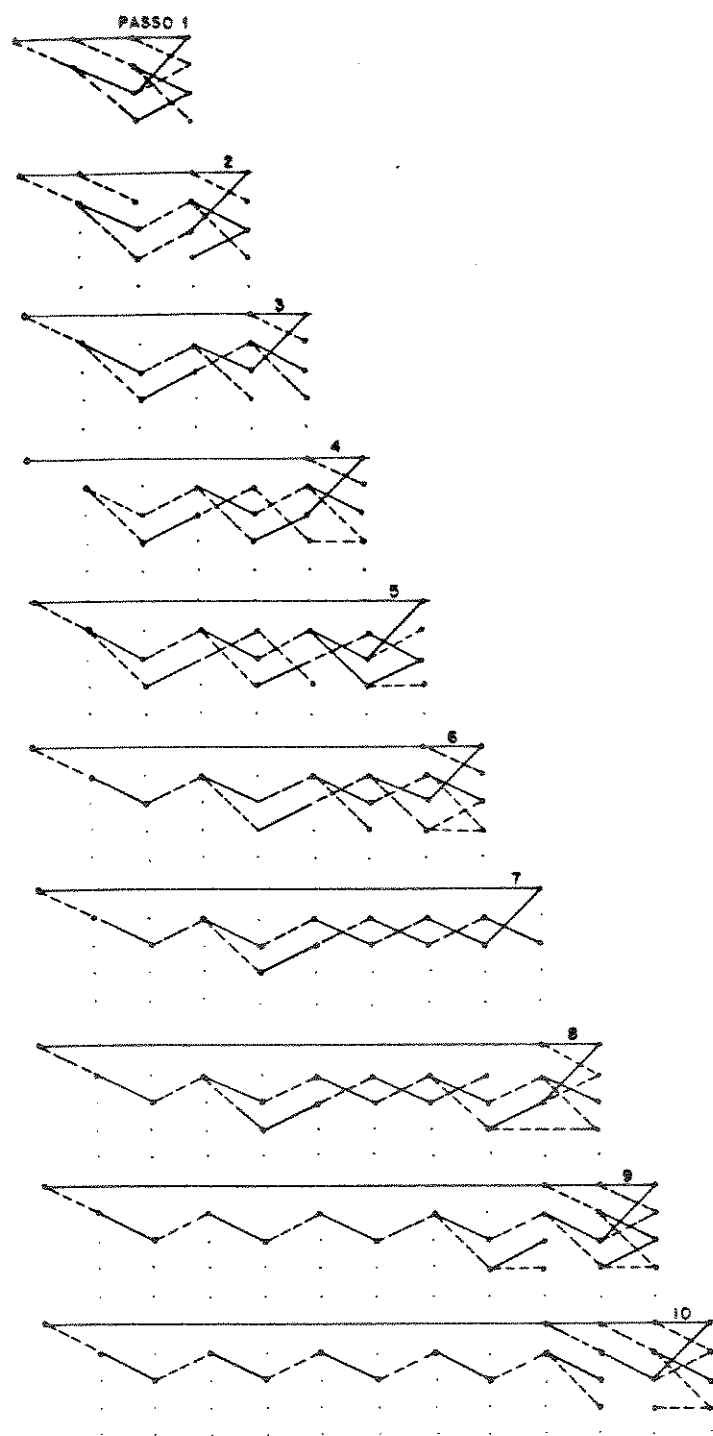


FIG. II. 4. 2 - Algoritmo de Viterbi.
 Processo de decodificação para o
 codificador convolucional da FIG. II. 3. 3

II.4.1 - PROCESSO RECURSIVO

- (i) - Seleciona um caminho sobrevivente, que chega a um estado com a menor distância de Hamming no instante jt ;
- (ii) - No próximo instante, $(j+1)t$ aumenta de um ramo todos sobreviventes anteriores;
- (iii) - Compara as distâncias de Hamming que chegam a cada um dos novos estados e retém a menor distância. Acumula estes valores em registradores;
- (iv) - Vai para (i), ou termina a decodificação.

Se as distâncias de Hamming forem iguais entre duas comparações, escolhe-se aleatoriamente qualquer uma das duas.

II.4.2 - EVENTO DE ERRO

A seqüência recebida r pode diferir da seqüência transmitida s durante um intervalo de tempo, devido a interferências introduzidas pelo meio de transmissão. Quando isto acontece define-se este evento como evento de erro.

Considere a Figura II.4.2.1, seja s a seqüência enviada e r a correspondente seqüência decodificada durante um intervalo de tempo T , então estas seqüências poderão divergir e reencontrar várias vezes. Define-se como evento de erro simples ao intervalo de tempo em que s e r divergem com somente um retorno.

Define-se como evento de múltiplos erros ao intervalo de tempo em que s e r divergiram com pelo menos dois retornos. Dessa forma, quando nos referirmos à probabilidade de evento de erro simples, estará explícito que esta probabilidade está condicionada a um determinado instante j_t da treliça onde as seqüências divergiram com somente um retorno.

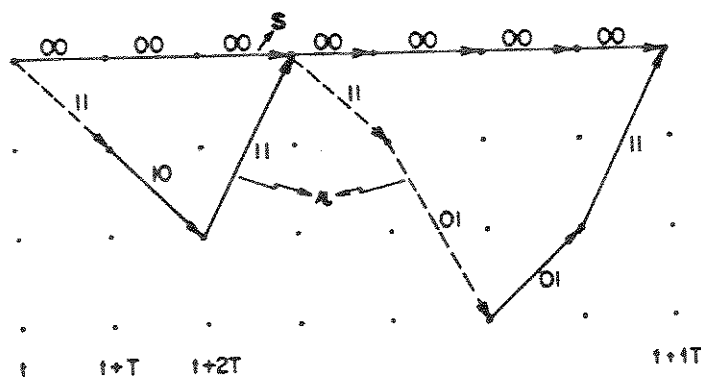


FIG. II. 4. 2. 1 - Eventos de erro. s é a seqüência transmitida e r é a seqüência decodificada.

III - TCM - MODULAÇÃO POR CODIFICAÇÃO EM TRELIÇA

III.1 - INTRODUÇÃO

Ungerböeck [3] mostrou que se aumentarmos o tamanho do alfabeto de modulação mantendo a taxa de transmissão de bits de informação inalterada podemos diminuir o consumo de potência sem modificarmos o desempenho do sistema.

Consideremos a codificação convolucional e a modulação no espaço de sinais como uma entidade única (Figura III.1.1). Denominemos os códigos gerados por esta entidade de Códigos de Treliça (a literatura também classifica estes códigos como Códigos de Alfabeto Redundante ou Códigos de Ungerböeck).

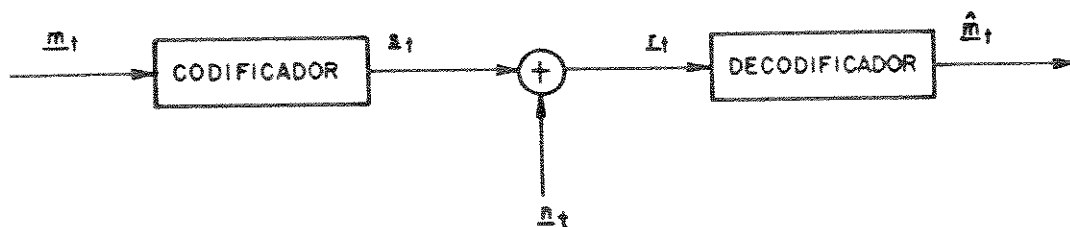


FIG. III.1 - Sistema TCM

Os Códigos de Treliça oferecem a atrativa possibilidade de obtenção de melhorias no desempenho, comparativamente à modulação não-codificada, sem a expansão da largura de faixa espectral, que normalmente acontece no processo de codificação [25]. Isto se baseia no fato de que, para relações sinal/ruído suficientemente altas, se o número de sinais é aumentado, mantendo-se a taxa de transmissão de informação constante, é possível obter-se um decréscimo significativo na probabilidade de erro de detecção dos sinais.

Como exemplo, se quisermos transmitir 4 bits de informação por intervalo T segundos de duração do símbolo de canal, a uma taxa de erro de símbolo de 10^{-5} , e usarmos um alfabeto com 32 símbolos (ao invés de 16), podemos diminuir o consumo de potência em aproximadamente 6 dB. O mesmo pode-se afirmar para 5 e 6 bits de informação, se usarmos alfabetos de 64 e 128 símbolos respectivamente e supondo-se um esforço total de decodificação. Este fato pode ser facilmente constatado através das curvas do parâmetro R_o versus E_b/N_o [22,26,27].

III.2 - O PROCESSO DE CODIFICAÇÃO

Uma maneira conveniente de caracterizar o processo de codificação consiste em visualizá-lo como um codificador convolucional binário seguido de um mapeamento das saídas do codificador, num conjunto de símbolos de canal de multiamplitude e/ou multifase (Fig. III.2.1). Como pode ser observado nesta figura, alguns bits de informação não são codificados, apenas selecionam sinais resultantes do processo de indexação a ser descrito como Partição de Conjunto. A expansão do conjunto de sinais está diretamente relacionada ao processo de codificação através da associação da redundância de sinais ao subconjunto terminal do espaço de sinais, mantendo assim inalterada a taxa de transmissão de bits de informação.

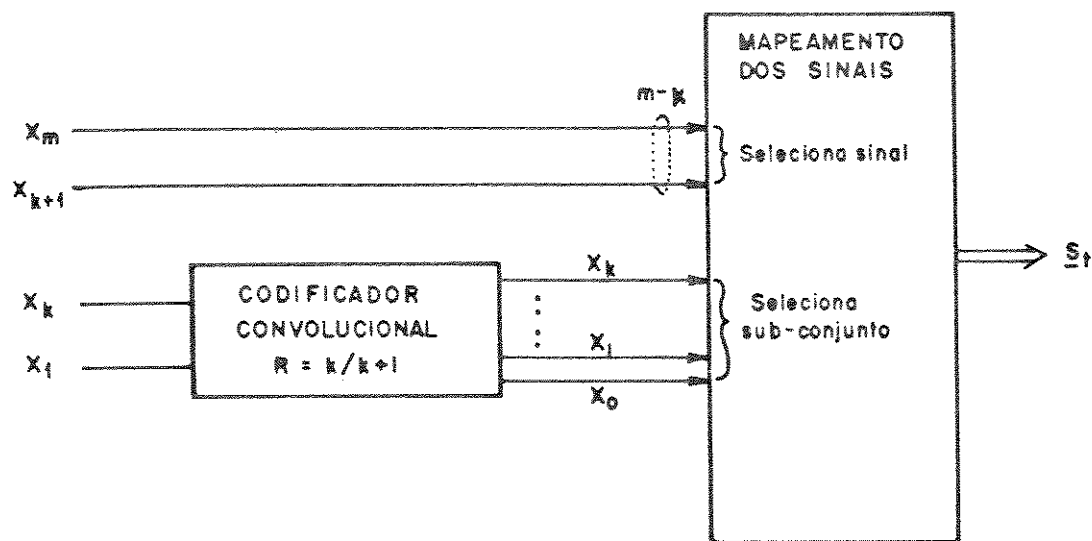


FIG. III. 2.1 - Estrutura do codificador

Através de um codificador de taxa $R = k/n$, k bits de informação, por intervalo de modulação T seg, são codificados em n bits (assume-se que $n=k+1$). Através de um mapeamento conveniente, os 2^n possíveis elementos da saída do codificador serão associados aos 2^n sinais do conjunto de sinais a serem transmitidos. Este mapeamento conveniente é denominado Partição de Conjunto.

III.3 - MAPEAMENTO POR PARTIÇÃO DE CONJUNTO

Este mapeamento é obtido como resultado de sucessivas partições da constelação de sinais em subconjuntos com crescentes distâncias mínimas ($\Delta_0 \leq \Delta_1 \leq \Delta_2 \leq \dots \leq \Delta_{n-1}$), entre os sinais que os compõem (Fig. III.3.1). Para o Mapeamento por Partição de Conjunto feito nesta figura, não existe apenas uma forma de subdivisão dos conjuntos, e também não é única a forma de numeração dos sinais. Permutando-se determinados subconjuntos, outros mapeamentos podem ser obtidos, com o mesmo padrão de aumento das distâncias mínimas dos subconjuntos.

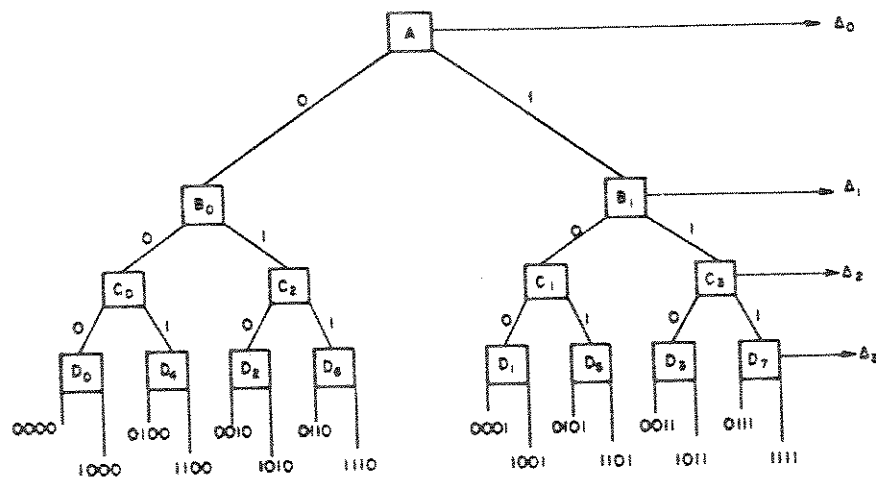


FIG. III.3.1 - Partição de uma constelação de $M=16$ sinais $\Delta_0 < \Delta_1 < \Delta_2 < \Delta_3$

III.4 - REGRAS DE CODIFICAÇÃO

Ungerböeck [5], de maneira heurística chegou a um determinado conjunto de regras que são comuns a todos os melhores códigos encontrados.

Após selecionar uma determinada estrutura de treliça, o passo seguinte consiste em relacionar os 2^n sinais possíveis da constelação, às transições da treliça, de modo a obter a máxima distância mínima Euclideana para o código, obedecendo então às seguintes regras:

- (i) - Todos os sinais da constelação devem aparecer com a mesma frequência caracterizando uma estrutura simétrica;
- (ii) - Os subconjuntos associados às transições paralelas na treliça (transições entre nós consecutivos, que começam e terminam juntas, ocasionadas por ligações diretas no circuito codificador) devem ter a maior distância mínima possível entre os sinais que os compõem;

(iii)- Os subconjuntos associados a um mesmo estado origem ou a um mesmo estado terminal (transições adjacentes no diagrama de treliça) devem ter a maior distância mínima possível entre os sinais que os compõem, sendo estas distâncias imediatamente inferiores às distâncias das transições paralelas obtidas em (ii).

Explicitadas as regras de codificação passaremos a seguir a um exemplo de aplicação, para constelações de 8 e 32 sinais e treliças de 4 e 8 estados respectivamente.

Exemplo III.4.1 -

Suponha que se deseja transmitir 2 bits de informação por intervalo de modulação. Para tal, necessitamos dispor de 4 pontos no espaço de sinal que representam estes 2 bits. Estes bits alimentarão um codificador convolucional (Fig. III.4.2) de taxa $R=2/3$. A Tabela III.4.1 detalha a relação entrada/estado inicial/estado final/saída do codificador da Fig. III.4.2.

Entrada X2 X1	Estado inicial	----->	Estado final	Saída Y2 Y1 Y0
0 0	0 0	----->	0 0	0 0 0
1 0	0 0	----->	0 0	1 0 0
0 1	0 0	----->	1 0	0 1 0
1 1	0 0	----->	1 0	1 1 0
0 0	1 0	----->	0 1	0 0 1
1 0	1 0	----->	0 1	1 0 1
0 1	1 0	----->	1 1	0 1 1
1 1	1 0	----->	1 1	1 1 1
0 0	0 1	----->	0 0	0 1 0
1 0	0 1	----->	0 0	1 1 0
0 1	0 1	----->	1 0	0 0 0
1 1	0 1	----->	1 0	1 0 0
0 0	1 1	----->	0 1	0 1 1
1 0	1 1	----->	0 1	1 1 1
0 1	1 1	----->	1 1	0 0 1
1 1	1 1	----->	1 1	1 0 1

TABELA III. 4. 1

Para este conjunto de sinais pode ser feito o mapeamento da saída do codificador em uma constelação 8-PSK obedecendo as regras de partição de conjunto como mostrado na Figura III.4.3.

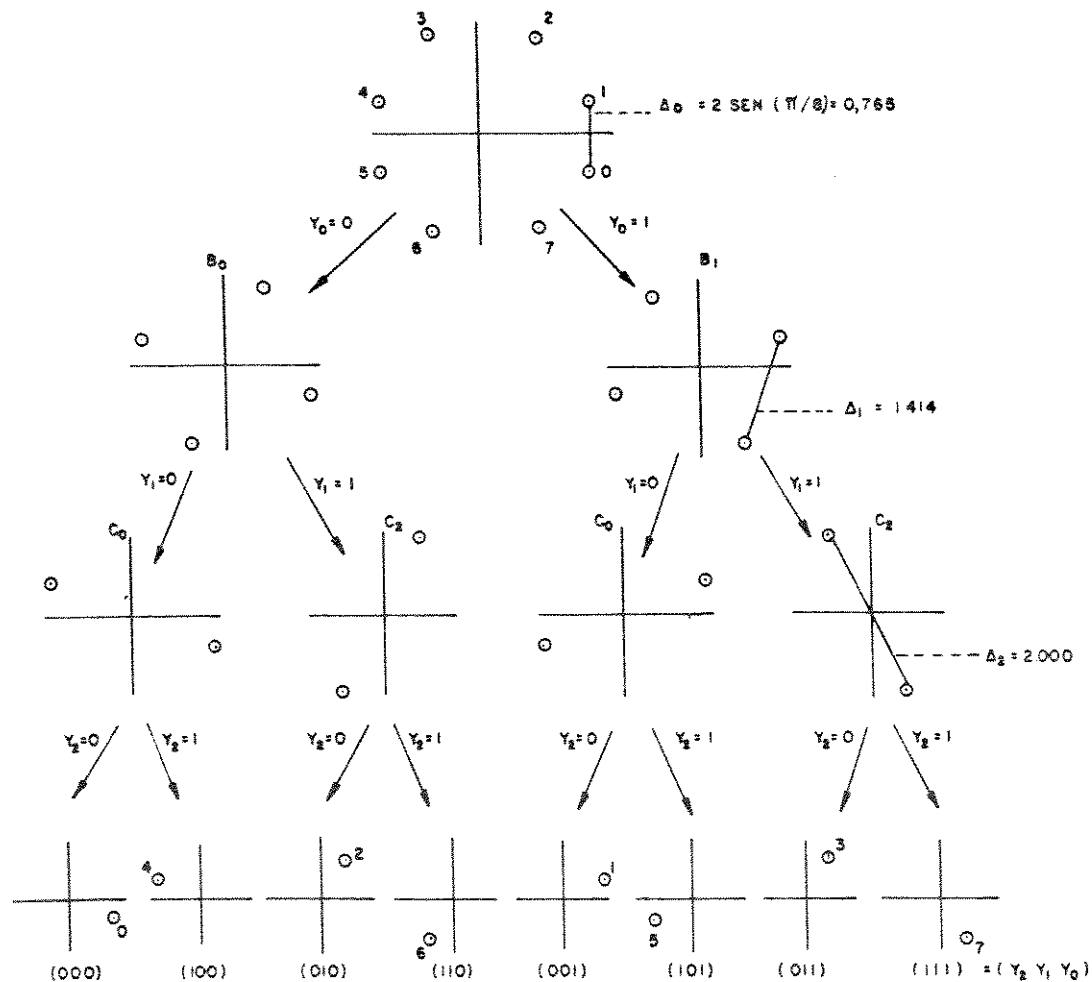


FIG. III.4.1 - Partição dos sinais 8-PSK em subconjuntos com distâncias mínimas crescentes ($\Delta_1 < \Delta_2 < \Delta_3$). $E\{|a^2 n|\} = 1$ (supõe-se a energia média dos sinais igual a 1)

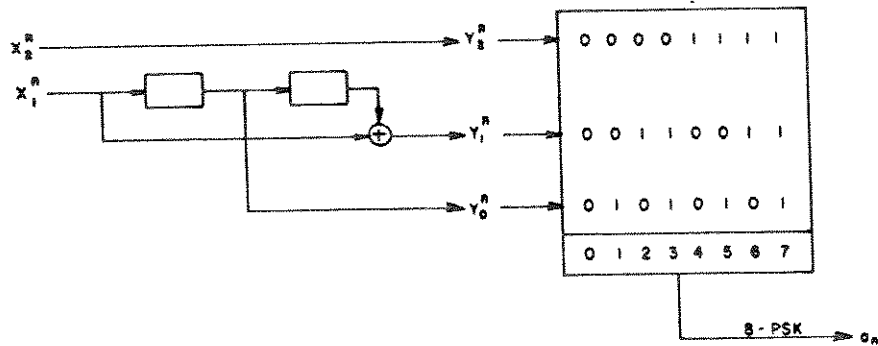


FIG III. 4.2 - Codificador convolucional associado à constelação 8-PSK com respectiva indexação dos sinais.

Veremos a seguir o diagrama de treliça associado ao circuito codificador (Fig. III.4.2).

Aos subconjuntos de sinais C_i serão associadas as transições paralelas no diagrama de treliça (Fig. III.4.3), pois estes sinais têm, entre si, a maior distância Euclideana possível.

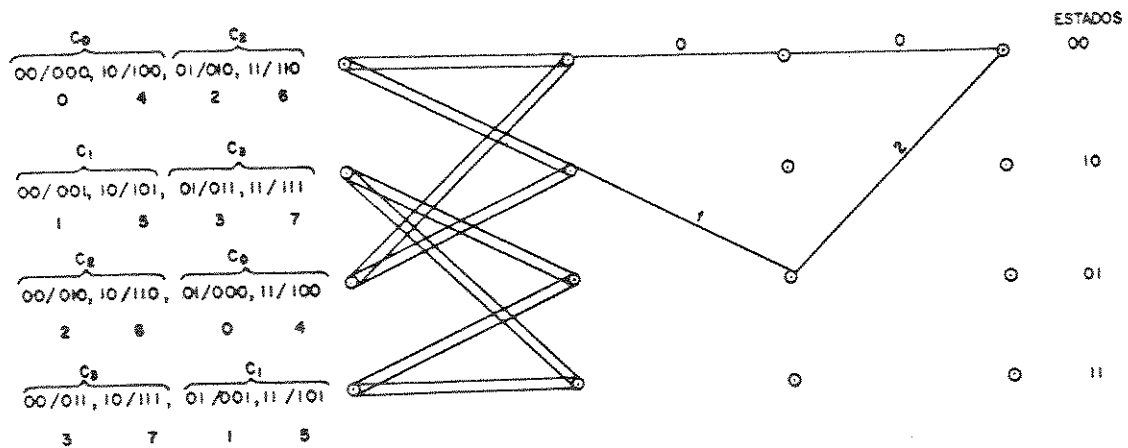


FIG III. 4.3 - Diagrama de treliça apresentando as regras de codificação aplicadas à partição do conjunto de sinais da FIG III. 4.2

Para relacionarmos os sistemas 8-PSK codificado e 4-PSK não-codificado, o sistema não-codificado pode ser imaginado como um sistema "codificado" em uma treliça de 1 único estado (Fig. III.4.4 e Fig. III.4.5).

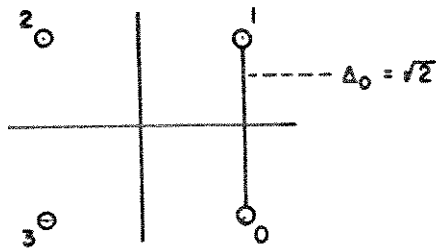


FIG. III.4.4 - Constelação 4-PSK

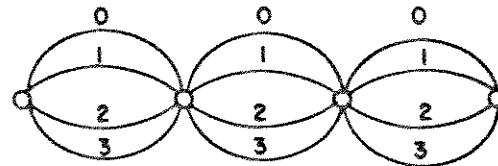


FIG. III.4.5 - Diagrama de treliça relativo ao sistema 4-PSK

As 4 transições no diagrama de estado permitem que qualquer sequência possa ser transmitida. A menor distância entre os sinais 4-PSK é $\Delta_0 = \sqrt{2}$ e pode ser identificada como a distância mínima desta constelação 4-PSK.

Observando-se agora o sistema 8-PSK da Figura III.4.1, quaisquer 2 caminhos na treliça (Figura III.4.2), que divergem de um estado em que se originam, e retornam depois de mais de um intervalo de tempo (transições adjacentes) tem a distância entre si maior que a distância Euclideana entre os sinais assinalados às transições paralelas ($\Delta_2 = 2.000$, Fig. III.4.1). Portanto a distância mínima para o sistema 8-PSK codificado em

uma treliça de 4 estados, segundo a partição de conjuntos da Figura III.4.1 é $d_{\min} = 2.000$. Isto pode ser verificado fazendo-se simples observações no diagrama de treliça da Figura III.4.2.

Suponha que foi transmitida a sequência toda nula e 2-1-2 a correspondente sequência decodificada. Calculemos a distância Euclideana entre duas sequências:

Para o primeiro intervalo de tempo a distância entre o sinal 2 e o sinal 0, (observar a Figura III.4.1) é determinada ao percorrermos em sentido contrário ao da criação da árvore que originou a partição de conjunto dos sinais 8-PSK. Isto é, parte-se dos subconjuntos terminais respectivos aos sinais 2 e 0 e localiza-se o nó intermediário que os originou. A distância entre os sinais neste conjunto fornecerá a distância $d_E(2,0)$. No caso, o subconjunto é o B_0 ($\Delta_1^2 = 1.999$). Para o segundo intervalo de tempo segue-se o mesmo procedimento obtendo, entre os sinais 1 e 0 a distância $\Delta_0^2 = 0.585$. No terceiro intervalo de tempo T a distância entre 2 e 0 é novamente $\Delta_1^2 = 1.999$. Somando-se estas distâncias, obtem-se $\Delta_1^2 + \Delta_0^2 + \Delta_1^2 = 4.583$. Como a distância Euclideana de qualquer transição adjacente é maior do que a das transições paralelas ($\Delta_2^2 = 4$), conclui-se que o $d_{\min}$ deste código é 2.

Exemplo III.4.2 - Constelação modulada em amplitude/fase
 32- Cross (Reticulado tipo Z_2) associada a uma estrutura de
 treliça de 8 estados.

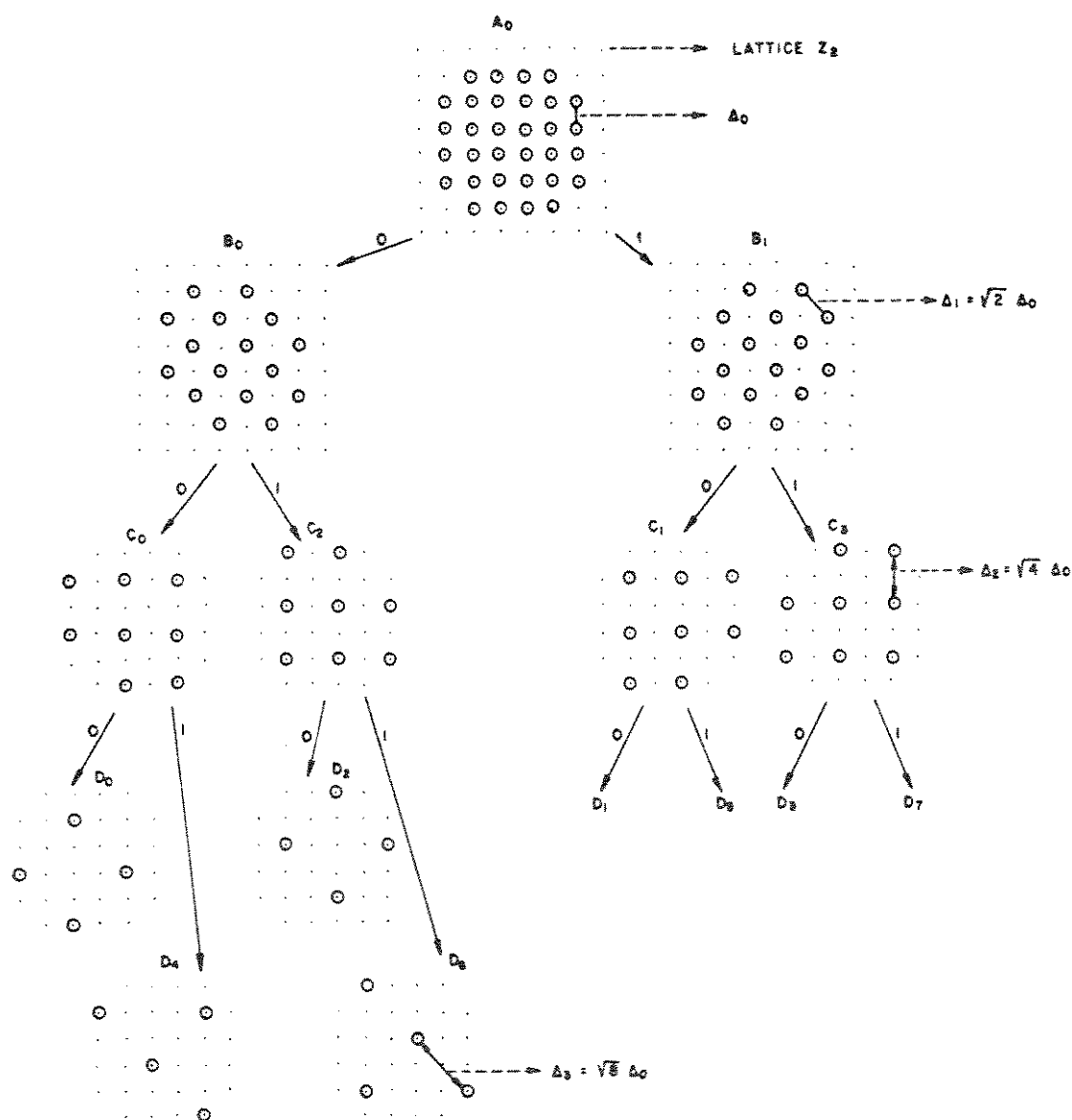


FIG. III. 4.6 - Partição de conjunto para a constelação "32-Cross" -
 Reticulado tipo Z_2

Aplicando-se a partição de conjunto da Figura III.4.6 a uma estrutura de treliça de 8 estados obtemos o seguinte diagrama:

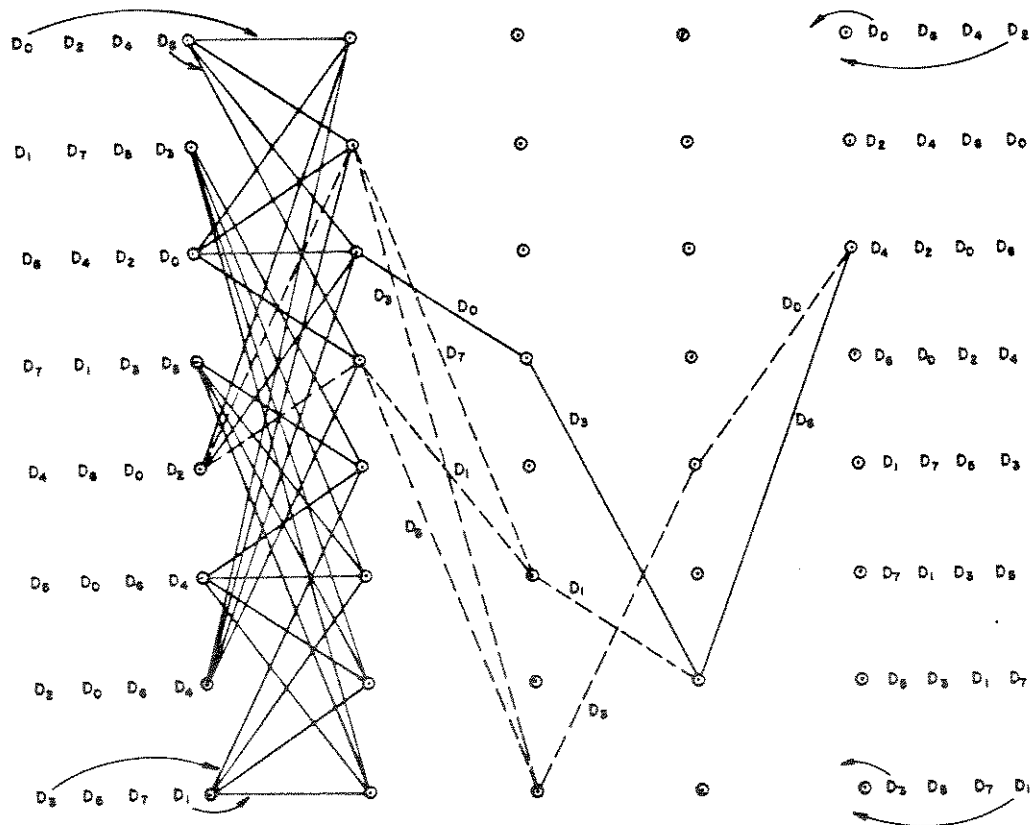


Fig. III.4.7 - Treliça de 8 estados para modulação de amplitude e fase com conjunto de sinais "32 - Cross". $d_{\min} = \sqrt{5} \Delta_0$.

Para uma sequência particular — $D_0 - D_0 - D_3 - D_6$ — a Fig. III.4.7 ilustra 4 caminhos de erro com distância mínima $\sqrt{5}\Delta_0$. Todos emergem de um mesmo estado e retornam depois de três ou quatro transições. Cada traço representa 4 transições paralelas referentes a cada subconjunto D_i da Figura III.4.6, composto de sinais com a maior distância Euclideana entre si. Cada uma das transições adjacentes assinaladas no diagrama de treliça, tem distância Euclideana em relação à sequência padrão menor do que cada conjunto D_i individualmente.

III.5 - DESEMPENHO DOS CÓDIGOS DE TRELIÇA

Para o cálculo do desempenho destes códigos, iremos assumir que o meio de transmissão é constituído de amostras de um Processo Gaussiano Branco Aditivo (AWGN) limitado em faixa, isento de interferência intersimbólica com um perfeito sincronismo da frequência e da fase da portadora, e que a energia média é unitária (para sinais equiprováveis).

A distância Euclideana entre duas seqüências é definida como

$$dE(x,y) = \sum_l dE(x_l, y_l) \quad (\text{III.5.1})$$

onde $dE(\dots)$ é a distância Euclideana ou Geométrica dos pontos no espaço dos sinais correspondentes aos bytes x_l, y_l , para quaisquer duas seqüências codificadas de mesmo comprimento.

Define-se como $d_{\min}$ a menor das distâncias entre todas as possíveis seqüências de símbolo de modulação, isto é:

$$d_{\min} = \min dE(x,y) = \min_{x \neq y} \left(\sum dE(x_l, y_l) \right) \quad (\text{III.5.2})$$

Por outro lado, para alta relação sinal/ruído a probabilidade de evento de erro aproxima-se assintoticamente do limitante inferior:

$$P_e \geq N(d_{\min}) * Q(d_{\min}/\sigma) \quad (\text{III.5.2})$$

onde $N(d_{\min})$ é o número médio de eventos de erro com distância mínima Euclideana (Eq. III.5.2), σ^2 é a variância do ruído e $Q(\cdot)$ é a função de probabilidade de erro Gaussiana:

$$Q(\xi) = \frac{1}{\sqrt{2\pi}} \int_{\xi}^{\infty} \exp(-y^2/2) dy \quad (\text{III.5.3})$$

Note que $Q(\xi)$ é monotonicamente decrescente, então para minimizar-se P_e deve-se maximizar $d_{\min}$. Dessa forma, o desempenho dos códigos de treliça baseia-se no critério de máximo $d_{\min}$. De outra forma, poderíamos utilizar o critério de mínima probabilidade de erro de bit, via função de transferência [2a], porém, a complexidade computacional crescendo exponencialmente, inviabiliza uma análise mais detalhada. Portanto os códigos devem ser projetados com o sentido de se obter a máxima distância mínima Euclideana entre as seqüências dos sinais modulados.

A análise de desempenho entre os sistemas de comunicações na forma combinada modulação/codificação e não-codificados é feita através do Ganho Assintótico de Codificação (GCA).

Este ganho é definido como:

$$GCA = 10 * \log_{10} \left\{ \frac{\left[\frac{d_{min}^2}{\bar{\xi}} \right]}{\left[\frac{d_{ref}^2}{\bar{\xi}} \right]} \right\} \quad (III.5.4)$$

onde d_{min} é definido pela Eq. III.5.2 e d_{ref} é a distância mínima entre os sinais de um sistema não-codificado, utilizado como referência, e $\bar{\xi} = \bar{\xi} \left\{ |a_n^2| \right\}$ é a Energia Média Quadrática de cada constelação de sinais.

Observe o diagrama de treliça abaixo em que a mensagem transmitida é a achurada. Um código é considerado ótimo quando o correspondente valor de d_{min} é o maior possível, isto é, qualquer transição que divirja da sequência transmitida e retorne depois de um ou mais intervalos de tempo deve apresentar um valor para a distância Euclideana com relação à sequência transmitida, a maior possível, porém, menor que a distância entre as transições paralelas.

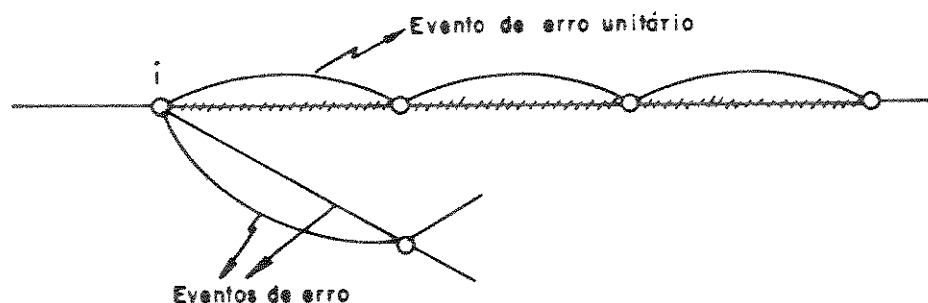


FIG. III. 4.8 - Eventos de erro.

O evento de erro unitário, associado à transição paralela, ocorrerá menos freqüentemente do que um evento de erro simples se e somente se a correspondente distância Euclideana for maior que $d_{\min}$ do código. Caso contrário, o evento de erro unitário limitará o desempenho do sistema. Portanto, esta é a razão pela qual devem ser associados às transições paralelas os subconjuntos de sinais com a maior distância Euclideana entre os mesmos. Este procedimento, juntamente com as demais regras de codificação, garante um Ganho Assintótico de Codificação superior àqueles dos sistemas não-codificados.

IV - CONCLUSÕES

Neste Capítulo procuramos abordar os aspectos básicos da técnica TCM (forma combinada de codificação e modulação) para transmissão digital sobre canais de faixa limitada. A razão fundamental para este estudo, vem do fato desta técnica permitir obter substanciais ganhos assintóticos de codificação, quando comparada com as técnicas de modulação multinível não-codificada, sem a eventual expansão de faixa.

A motivação para este estudo decorre do fato de que, com a proliferação de redes de comunicação digital interconectadas e a troca de informação entre computadores, gerou uma demanda por equipamentos de comunicação sofisticados operando com altas taxas de transmissão, alta vazão e com alta confiabilidade.

Assim sendo, estes requisitos podem ser supridos através da técnica TCM e uma nova fase de geração de modems será iniciada.

A seguir faremos um estudo comparativo entre Estruturas Reticulares que comportam os sinais a serem transmitidos, com o objetivo de selecionar as mais propícias para o sistema TCM.

C A P Í T U L O 2

ANÁLISE DE ESTRUTURAS RETICULARES PARA
MODULAÇÃO POR CODIFICAÇÃO EM TRELIÇA
EM 2 DIMENSÕES

I - INTRODUÇÃO

Neste Capítulo abordaremos o desempenho do sistema TCM (Modulação por Codificação em Treliça) em 2 dimensões onde dois tipos de estruturas para a configuração dos sinais a serem transmitidos serão apresentados.

Uma das estruturas a ser considerada será o reticulado Quadrado (Z^2) enquanto que a outra será a Hexagonal (A_2). A estrutura reticular deve ser vista como uma forma de representação dos sinais no espaço dos sinais, de tal forma que estes tenham entre si um posicionamento regular definido pelos parâmetros que geram este reticulado.

Uma análise comparativa de desempenho entre os esquemas TCM para várias taxas de codificação utilizando conjuntos de sinais gerados em cada um dos reticulados será apresentada no capítulo seguinte. O objetivo será então, comparar o desempenho do sistema de comunicações utilizando a técnica TCM quando levado em consideração as constelações de sinais propostas, tanto do tipo Z^2 como do tipo A_2 , sob o critério de minimizar a energia de transmissão destes sinais, porém, com a maior distância Euclídeana entre si. Este compromisso entre máximo distanciamento entre os sinais e mínima energia de transmissão deu origem à idéia da utilização do reticulado A por ser o mais denso em R^2 [2a].

Algumas das constelações de sinais comumente utilizadas e que são provenientes do reticulado $\mathbb{Z}^2$ foram propostas por Campopiano e Glazer [29]. Portanto, é sob esta consideração que será feita a Partição de Conjunto para estas constelações. Uma subsequente realização de procura por códigos ótimos para algumas taxas de codificação será feita no Capítulo 3, utilizando-se o algoritmo de procura de códigos ótimos [30]. O estudo das constelações propostas para o reticulado Hexagonal (A_2) segue o mesmo desenvolvimento.

As diversas constelações propostas e as já existentes serão comparadas entre si através do critério de distancia entre os sinais que as compõem, pela energia de transmissão dos mesmos, e pela capacidade de desmembramento dos sinais no espaço, segundo o método do Partição de Conjunto.

Os códigos gerados em cada uma das estruturas reticulares serão comparados mediante a mesma taxa de codificação.

Os parâmetros de desempenho dos códigos gerados sobre cada constelação serão o Δ_i , $d_{\min}$, $\Delta_{\min}(\mu)$, $\Delta(\mu+1)$, GCA, parâmetros que também serão caracterizados no Capítulo seguinte.

II - ESTRUTURAS RETICULARES

Um reticulado é uma estrutura do espaço Euclideano n -dimensional obtida ao se fazer a combinação linear de uma base linearmente independente (L.I.) com um conjunto de elementos com coordenadas inteiras [33].

O conjunto de todos os pontos x da forma

$$x = \mu_1 a_1 + \mu_2 a_2 + \dots + \mu_n a_n, \quad (\text{II.1})$$

onde $\mu_1, \dots, \mu_n$ são inteiros, é chamado de reticulado A com base $a_1, \dots, a_n$.

Como $a_1, \dots, a_n$ são L.I., o vetor x no formato de (II.1) com $\mu_1, \dots, \mu_n$ inteiros é único para a base específica e portanto caracteriza exatamente seu posicionamento.

A partir deste fato, iremos construir um código de treliça que mapeia uma sequência de símbolos de um alfabeto de entrada específico em uma sequência de pontos de saída obtidos de um certo reticulado A contido no espaço Euclideano n -dimensional $\mathbb{R}^n$. O conjunto de todos os possíveis pontos de saída formam a constelação de sinais que é um subconjunto de A .

Cada ponto gerado em um destes reticulados pode ser caracterizado como o centro de uma esfera de raio ρ . Estas esferas estão compactadas de modo a não se interseccionarem [34].

Vistas sob o aspecto da demodulação de sinais estas esferas seriam as regiões de decisão abrupta de cada sinal transmitido.

O objetivo deste trabalho é exatamente verificar e comparar o desempenho dos códigos gerados a partir de constelações oriundas dos reticulados propostos, observando-se a princípio que os códigos TCM mapeados em constelações de sinais tipo A_2 conduzem-nos a inferir um desempenho superior quando comparado com as constelações de sinais tipo Z^2 , pois aqueles sinais são os mais densamente compactados em $\mathbb{R}^2$.

II.1 - EMPACOTAMENTO ESFÉRICO E SEUS PARÂMETROS

Informalmente, um empacotamento esférico em $\mathbb{R}^n$ é um arranjo de infinitas esferas não sobrepostas de mesmo tamanho.

Um empacotamento esférico Λ em $\mathbb{R}^n$ de raio ρ [?] consiste de uma sequência infinita de pontos $x^{(1)}, x^{(2)}, \dots$ no $\mathbb{R}^n$ (os centros das esferas) tal que

$$\text{dist} \left(x^{(i)}, x^{(j)} \right)^2 = \left\| x^{(i)} - x^{(j)} \right\|^2 \geq 4\rho^2 \quad \text{para } i \neq j \quad (\text{II.1.1})$$

então, se esferas de raio ρ são desenhadas em volta dos centros $x^{(1)}, x^{(2)}, \dots$, estas esferas não se sobrepõem e $x^{(i)}$ são os pontos gerados de um reticulado.

A dimensão d de um empacotamento reticulado Λ é o máximo número de centros (pontos $x^{(1)}, x^{(2)}, \dots$ em $\mathbb{R}^n$) Linearmente Independentes em Λ . Desde que $\Lambda \subseteq \mathbb{R}^n$, $d \leq n$.

Sejam $x^{(1)} = (x_1^{(1)}, \dots, x_n^{(1)})$, $\dots$, $x^{(d)} = (x_1^{(d)}, \dots, x_n^{(d)}) \in \Lambda$

os centros Linearmente Independentes que geram o reticulado.

A matriz $d \times n$ $M = \begin{bmatrix} x_j^{(i)} \end{bmatrix}$, $1 \leq i \leq d$, $1 \leq j \leq n$, é a matriz geradora de Λ e o reticulado consiste de todas as combinações inteiras das linhas de M .

$$\text{O determinante de } \Lambda \text{ é } \det \Lambda = \left(\det M \times M^t \right)^{1/2} \quad (\text{II.1.2})$$

Quando o reticulado tem a mesma dimensão do espaço em que se situa, isto é, $d = n$, M é uma matriz quadrada e

$$\det \Lambda = \left| \det M \right| \quad (\text{II.1.3})$$

O número de contato $\tau(x)$ de uma esfera centrada em $x^{(i)}$ é o número de esferas que tocam esta esfera. O valor máximo de $\tau(x)$ para $x^{(i)} \in \Lambda$ é denotado por $\tau_{\max}$. Para um empacotamento reticulado $\tau(x) = \tau_{\max} = \tau$, independentemente da escolha de $x^{(i)}$.

A região de Voronoi [31] é a região de todos os pontos do espaço em volta do centro das esferas que estão mais próximos do centro de uma esfera do que de qualquer outra,

$$V(x^{(i)}) = \left\{ y \in \mathbb{R}^n \mid \left\| y - x^{(i)} \right\| \leq \left\| y - x^{(j)} \right\| \text{ para todo } j \neq i \right\} \quad (\text{II.1.4})$$

A densidade de um empacotamento esférico [7] é a fração do espaço $\mathbb{R}^n$ que é coberto pelas esferas. Para um empacotamento esférico de raio ρ e dimensão $d = n$, a densidade é dada pela fórmula

$$\Delta = \frac{V_n \rho^n}{\det \Lambda} \quad (\text{II.1.5})$$

onde

$$V_n = \frac{\pi^{n/2}}{\Gamma((n/2)+1)} \text{ é o volume de uma esfera unitária.}$$

A densidade do empacotamento esférico, quantifica o arranjo dos elementos que compõem um determinado reticulado. Isto é

relevante na medida em que quanto mais denso for o empacotamento, ou da mesma forma, dada uma região delimitada no reticulado quanto menos espaço for "desperdiçado" na compactação dos sinais respeitando o distanciamento padrão dos mesmos (2ρ), melhor será o aproveitamento deste reticulado como base para projeção de constelações de sinais empregados na transmissão digital.

Em uma aproximação assintótica, reticulados mais densos contêm um contingente maior de sinais, isto é, os pontos estão mais próximos entre si resultando em constelações mais econômicas em termos de energia de transmissão. Visto de outra forma, para um mesmo número de pontos, e uma energia fixa de transmissão, constelações geradas em reticulados mais densos acarretam em maior distanciamento entre os sinais que as compõem, favorecendo diretamente no processo de decodificação dos mesmos.

Como foi visto anteriormente, quanto maior o $d_{\min}$ de um código corretor de erros, melhor desempenho apresentará este código. No caso do sistema TCM o $d_{\min}$ do código está diretamente relacionado com a distância Euclideana entre os sinais da constelação utilizada no mapeamento da saída do codificador convolucional. A maximização de $d_{\min}$ é uma maneira de melhorarmos o sistema de comunicação, portanto, gerar codificadores a partir de constelações oriundas de reticulados mais densos pode levar a algum ganho assintótico de codificação. Esta afirmação só pode ser comprovada mediante o estudo do comportamento das constelações propostas, sob o ponto de vista de eficiência em termos da Partição de Conjunto, como veremos mais adiante.

A seguir será feito o estudo do reticulado Quadrado (Z^2) e Hexagonal (A_2) e calculados os seus parâmetros.

II.2 - RETICULADO QUADRADO ($\mathbb{Z}^2$)

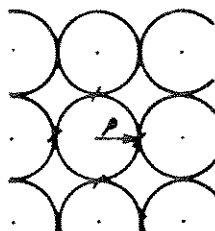
A estrutura reticular mais simples do $\mathbb{R}^n$ é o reticulado $\mathbb{Z}^n$, que consiste de todos os pontos do $\mathbb{R}^n$ com coordenadas inteiras. $\mathbb{Z}^2$ é a projeção de $\mathbb{Z}^n$ em $\mathbb{R}^2$ e é denominado Reticulado Quadrado.

Iremos construir um código de treliça que mapeia uma sequência de símbolos de um alfabeto de entrada específico em uma sequência de pontos de saída obtidos de um certo reticulado $\mathbb{Z}^2$ (Fig. I) em um espaço Euclideano de 2 dimensões, $\mathbb{R}^2$. O conjunto de todos os possíveis pontos de saída formam as constelações de sinais que são subconjuntos de $\mathbb{Z}^2$ (Figs. I.4-a, I.4-b, I.4i, I.4ii).

Para tanto definiremos o reticulado $\mathbb{Z}^2$ como uma estrutura no espaço bi-dimensional gerado pelos vetores $(1,0)$ e $(0,1)$ e que consiste de todos os pontos em $\mathbb{R}^2$ com coordenadas inteiras. Este reticulado tem dimensão $d = 2$, matriz Geradora $M = I_2$, ou seja, $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ e o determinante que a caracteriza é $\det A = 1$.

A Figura I apresenta o reticulado $\mathbb{Z}^2$ centrado no ponto-lacuna $(\frac{1}{2}, \frac{1}{2})$. Todos os pontos gerados apresentam as mesmas propriedades e como a distância entre quaisquer dois pontos consecutivos é $\Delta_0 = 1$, o raio para cada centro gerado será $\rho = 1/2$.

O número de contato $\tau(x)$ de uma esfera centrada em $x^{(i)}$, onde $x^{(i)}$ é qualquer ponto gerado em $\mathbb{Z}^2$, é o número de esferas vizinhas (circunferências em $\mathbb{R}^2$) que tocam $x^{(i)}$, portanto para $\mathbb{Z}^2$ $\tau(x) = 4$, conforme ilustração abaixo.



A densidade do empacotamento esférico $\mathbb{Z}^2$ é dada por

$$\Delta = \frac{\pi^{2/2} (1/2)^2}{\Gamma((2/2)+1)1} = \frac{\pi}{4 \Gamma(2)} = 0.7853$$

A área da Região de Voronoi $\gamma(x^{(i)})$ (ver Fig. 1) para qualquer ponto do reticulado $\mathbb{Z}^2$ é $\gamma(x^{(i)}) = \Delta_0^2$ onde $\Delta_0 = 2\rho$ é a distância entre dois pontos consecutivos.

II.2.1 - CONSTELAÇÕES PROPOSTAS EM Z^2

Uma constelação de sinais tipo Z^2 é um conjunto de pontos que reúne todas as possibilidades de representar uma determinada sequência de bits de saída de um codificador convolucional em um espaço Euclidiano bidimensional Z^2 .

Para enviar n bits/símbolo é necessária uma constelação de 2^n pontos. As Figuras I.4-a, I.4-b, I.4, I.4ii ilustram as constelações propostas para conjuntos de 8, 8, 16, 32 pontos respectivamente. Cada constelação em Z^2 está centrada no ponto-lacuna $(\frac{1}{2}, \frac{1}{2})$ e os pontos foram distribuídos simetricamente em relação aos eixos coordenados de tal forma a sintetizar a menor quantidade de energia em torno da origem.

A disposição dos pontos de sinais na constelação possibilita uma maior ou menor imunidade ao ruído, uma vez que esta disposição influencia diretamente no processo de demodulação.

O cálculo da energia de cada ponto é baseado na distância Euclidiana de cada ponto à origem, tomando-se como medida padrão Δ_0 que é a distância entre quaisquer dois pontos consecutivos no reticulado.

A energia média total é a média da somatória das energias parciais

$$\bar{\varepsilon} \left\{ \left| a_n^2 \right| \right\} = \frac{1}{m} \sum_{i=1}^m \|a_i\|^2 \quad (\text{II.2.1})$$

Estas distâncias parciais são facilmente obtidas devido à simetria e regularidade dos pontos em Z^2 e são todas obtidas em função de Δ_0 .

A energia de pico é dada por

$$\varepsilon_p = \max_i \|a_i\|^2 \quad (\text{II.2.2})$$

Uma outra forma de caracterizar-se os pontos de uma determinada constelação, é supor que a energia média total é unitária e, desta forma, cada constelação terá uma distância específica Δ_0 , entre os sinais.

As seguintes constelações de sinais I.4-a, I.4, I.4ii, em Z^2 foram apresentadas por Campopiano e Glazer em [29]. Neste trabalho analisaremos como as diferentes constelações se comportam quando empregadas em sistemas de Modulação por Codificação em Treliça.

Para o conjunto de 8 sinais foram propostas duas constelações alternativas, I.4-a conhecida como "8 QASK" e I.4-b. Conforme cálculo efetuado a energia média de I.4-a $\left(\frac{3}{2} \Delta_0^2 \right)$ é menor do que a de I.4-b $\left(\frac{5}{2} \Delta_0^2 \right)$, mas mantivemos mesmo assim esta constelação, pois esta configuração de sinais, conforme veremos, sugere dois tipos de partição de conjunto e portanto merece uma atenção à parte.

I - Retículo tipo $\mathbb{Z}^2$

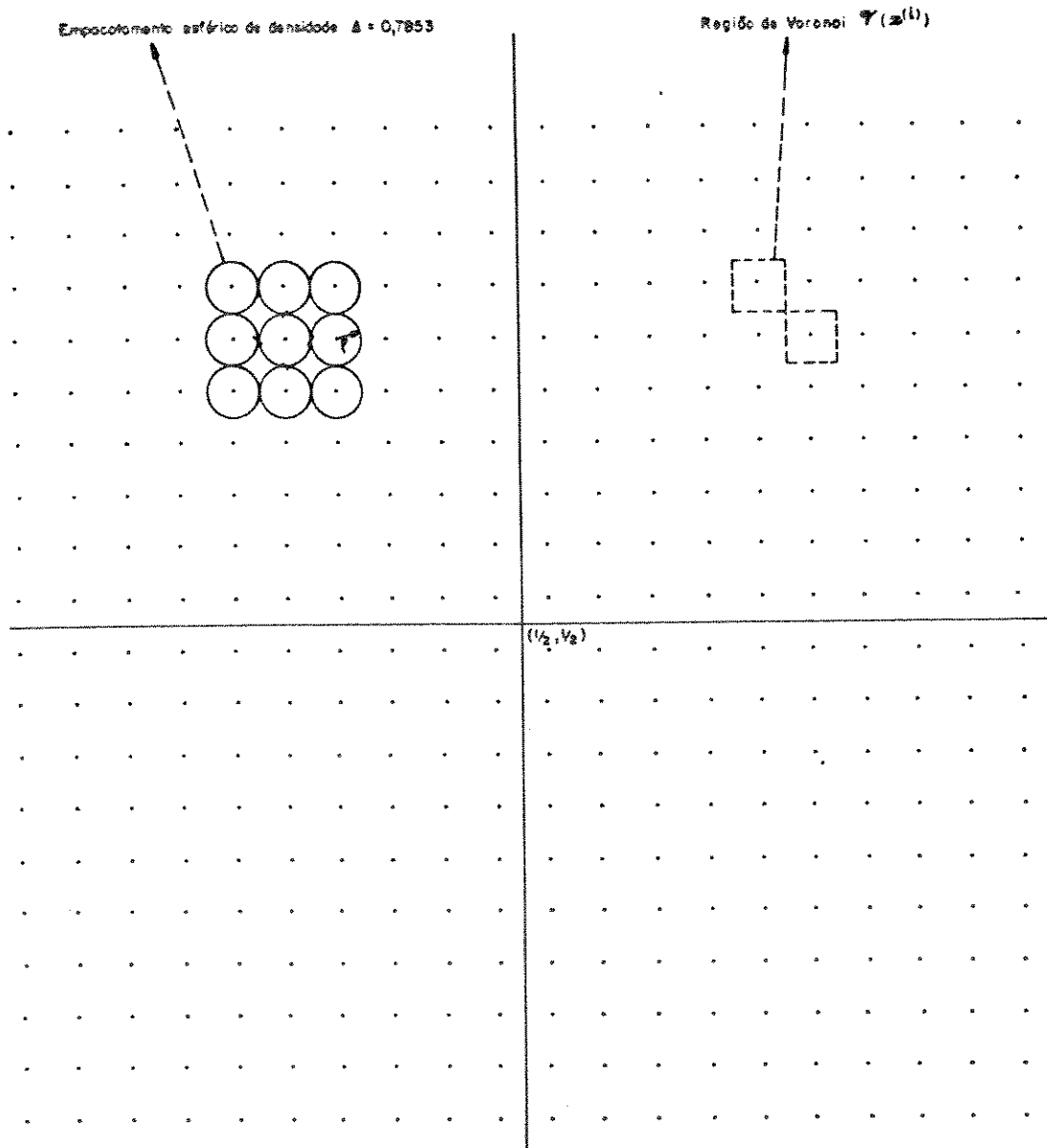


FIGURA 1

Vetores geradores $[1,0], [0,1]$

$$M = I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$d = 2$$

$$\det = 1$$

$$\rho = 1/2$$

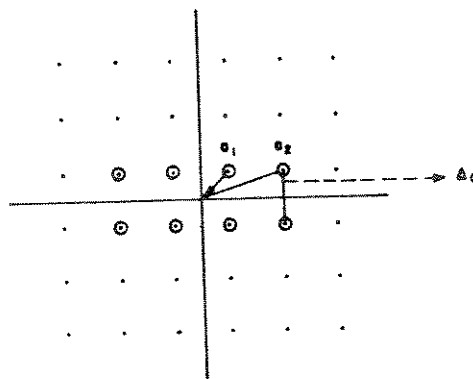
Centrada no ponto-lacuna $(1/2, 1/2)$

$$\tau(\mathbb{Z}) = 4$$

$$\Delta = 0,7853$$

$$V(x^{(1)}) = \Delta_0^2$$

Constelações em $\mathbb{Z}^2$
 I. (L) $m = 8$ sinais
 Configuração (a)
 Centrado em $(1/2, 1/2)$



$$\frac{1}{8} \{ |a_n^2| \} = \frac{3}{2} \Delta_0^2 = 1$$

$$\Delta_0 = 0.816$$

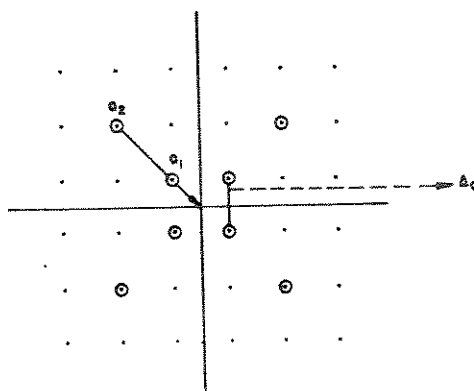
$$a_1^2 = \frac{\Delta_0^2}{2}$$

$$a_n^2 = \frac{5\Delta_0^2}{2}$$

$$\frac{1}{8} \{ |a_n^2| \} = \frac{1}{8} \left[4 \left(\frac{\Delta_0^2}{2} \right) + 4 \left(\frac{5\Delta_0^2}{2} \right) \right]$$

Fig. I. L-a

I. (L) Configuração (b)
 Centrado em $(1/2, 1/2)$



$$\frac{1}{8} \{ |a_n^2| \} = \frac{5}{2} \Delta_0^2 = 1$$

$$\Delta_0 = 0.632$$

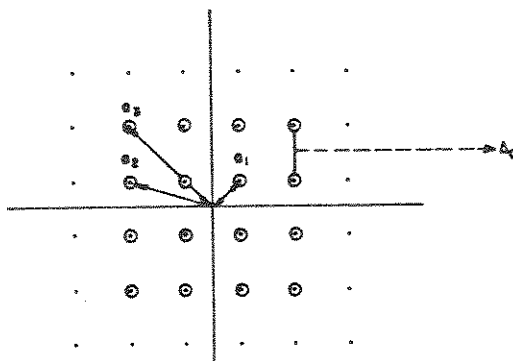
$$a_1^2 = \frac{\Delta_0^2}{2}$$

$$a_n^2 = \frac{9}{2} \Delta_0^2$$

$$\frac{1}{8} \{ |a_n^2| \} = \frac{1}{8} \left[4 \left(\frac{\Delta_0^2}{2} \right) + 4 \left(\frac{9\Delta_0^2}{2} \right) \right]$$

Fig. I. L-b

I. (L) $m = 16$ sinais
 Centrado em $(1/2, 1/2)$



$$\frac{1}{16} \{ |a_n^2| \} = \frac{5}{2} \Delta_0^2 = 1$$

$$\Delta_0 = 0.632$$

$$a_1^2 = \frac{\Delta_0^2}{2}$$

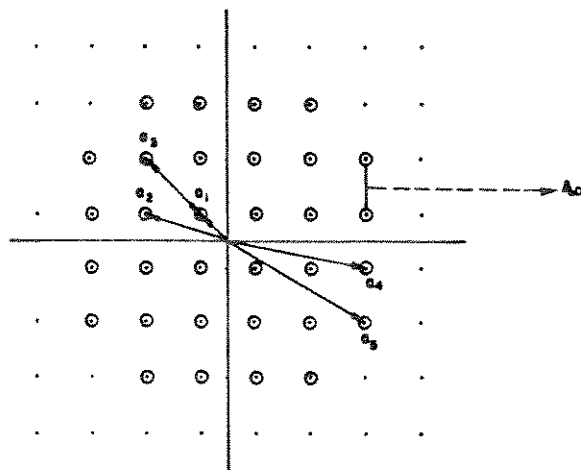
$$a_2^2 = \frac{5\Delta_0^2}{2}$$

$$a_3^2 = \frac{9\Delta_0^2}{2}$$

$$\frac{1}{16} \{ |a_n^2| \} = \frac{1}{16} \left[4 \left(\frac{\Delta_0^2}{2} \right) + 8 \left(\frac{5\Delta_0^2}{2} \right) + 4 \left(\frac{9\Delta_0^2}{2} \right) \right]$$

Fig. I. LL

I. (44) $m=32$ slots



$$\frac{1}{32} \left[|A_n|^2 \right] = \frac{4!}{8} \Delta_0^2 = 1$$

$$\Delta_0 = 0.441$$

$$A_1^2 = \frac{\Delta_0^2}{2}$$

$$A_2^2 = \frac{5}{2} \Delta_0^2$$

$$A_3^2 = \frac{9}{2} \Delta_0^2$$

$$A_4^2 = \frac{27}{4} \Delta_0^2$$

$$A_5^2 = \frac{17}{2} \Delta_0^2$$

$$\frac{1}{32} \left[|A_n|^2 \right] = \frac{1}{32} \left[4 \left(\frac{\Delta_0^2}{2} \right) + 8 \left(\frac{5}{2} \Delta_0^2 \right) + 4 \left(\frac{9}{2} \Delta_0^2 \right) + 8 \left(\frac{27}{4} \Delta_0^2 \right) + 8 \left(\frac{17}{2} \Delta_0^2 \right) \right]$$

FIGURA I. 44i

II.3 - RETICULADO HEXAGONAL (A_2)

Para $n \geq 1$, A_n é o reticulado n -dimensional consistindo dos pontos $(x_0, \dots, x_n)$ em $\mathbb{Z}^{n+1}$ com $\sum x_i = 0$. Assim A_n é uma estrutura de n dimensões descrita por coordenadas $(n+1)$ -dimensionais [7].

Para $n \geq 1$

$$A_n = \left\{ x \in \mathbb{Z}^{n+1} \mid \sum_{j=1}^{n+1} x_j = 0 \right\} \quad (\text{II.3.1})$$

Desse modo, A_2 é o reticulado Hexagonal no espaço Euclideano bi-dimensional, descrito pelas coordenadas inteiras tri-dimensionais (x_0, x_1, x_2) com $x_0 + x_1 + x_2 = 0$.

As μ -coordenadas (μ_1, μ_2) bi-dimensionais podem ser obtidas das x -coordenadas por

$$(\mu_1, \mu_2) = (x_0, x_1, x_2) \times \frac{1}{2} \times M^t \quad (\text{II.3.2})$$

onde

$$M = \begin{pmatrix} 1 & 0 & -1 \\ \frac{\sqrt{3}}{2} & -\frac{2\sqrt{3}}{3} & \frac{\sqrt{3}}{3} \end{pmatrix}$$

Por exemplo os pontos $(0,0)$; $(1,0)$; $(\frac{1}{2}, \frac{\sqrt{3}}{2})$; $(-\frac{1}{2}, \frac{\sqrt{3}}{2})$ têm x -coordenadas $(0,0,0)$; $(1,0,-1)$; $(1,-1,0)$; $(0,-1,1)$ respectivamente.

O reticulado A_2 também pode ser gerado pelos vetores $(1,0)$, e $(-\frac{1}{2}, \frac{\sqrt{3}}{2})$, assim

$$d = 2,$$

$$M = \begin{pmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \end{pmatrix},$$

$$\det = \frac{\sqrt{3}}{2}$$

$$\text{O mapeamento } (\gamma_1, \gamma_2) \in \mathbb{Z}^2 \Leftrightarrow \left(\gamma_1 - \left(\frac{1}{2}\right)\gamma_2, \frac{\sqrt{3}}{2}\gamma_2 \right) \in A_2$$

nos permite rotular os pontos de A_2 com pares de inteiros.

A forma mais simples do reticulado A_2 é um triângulo equilátero, e a região de Voronoi de um ponto pertencente a este reticulado é um hexágono de área $\frac{\sqrt{3}}{2} \Delta_0^2$ onde Δ_0 é a distância entre quaisquer dois pontos consecutivos.

A densidade do empacotamento esférico $\Delta = \frac{\pi\sqrt{3}}{6} = 0.9069$ é a maior obtida no $\mathbb{R}^2$ (veja [32, p.11]). Todos os pontos estão à distância de 2ρ onde ρ é o raio das "esferas" centradas em $x^{(i)}$ e que se justapõem, dando a esta estrutura um empacotamento 15% mais denso do que o de $\mathbb{Z}^2$ ($\Delta_H = 1.1548 \Delta_Q$).

O número de contato de cada "esfera" centrada em $x^{(i)}$ é $\tau(x) = 6$, isto é cada "esfera" tem contato com outras 6 (observar Fig.II).

II.3.1 - CONSTELAÇÕES PROPOSTAS EM A_2

Uma constelação de sinais tipo A_2 é um conjunto de 2^n sinais de norma mínima em torno da origem do reticulado A_2 , destinados a representar todas as possíveis saídas de um codificador operando a uma determinada taxa k/n .

As Figuras II.i-a, II.i-b, II.ii-a, II.ii-b, II.ii-c, II.iii-a, II.iii-b, II.iii-c, ilustram as constelações propostas para conjuntos de 8, 16 e 32 sinais.

As constelações variam quanto à centralização dos eixos coordenados na intenção de procurar configurações de energia mínima em torno da origem. Como a disposição dos pontos no reticulado é tão importante quanto a energia que acarretam sob o ponto de vista da Partição de Conjunto, não nos restringiremos somente às constelações com energia mínima e manteremos constelações menos eficientes mas com variada disposição de sinais.

Veremos adiante que um critério para escolha de constelações eficientes, no caso do sistema TCM, está na distância Euclideana intraconjunto de pontos no último nível da Partição de Conjunto da Constelação, nível referente ao conjunto de sinais indexados às transições paralelas na treliça. Via de regra quanto maior a menor distância neste nível da Partição, melhor o desempenho do sistema para os códigos procurados.

A regularidade dos pontos no reticulado, assim como a "disposição equilátera" dos mesmos compensou a falta de simetria de algumas constelações, tornando possível obter todas as energias em função de Δ_0 (distância entre quaisquer dois pontos consecutivos).

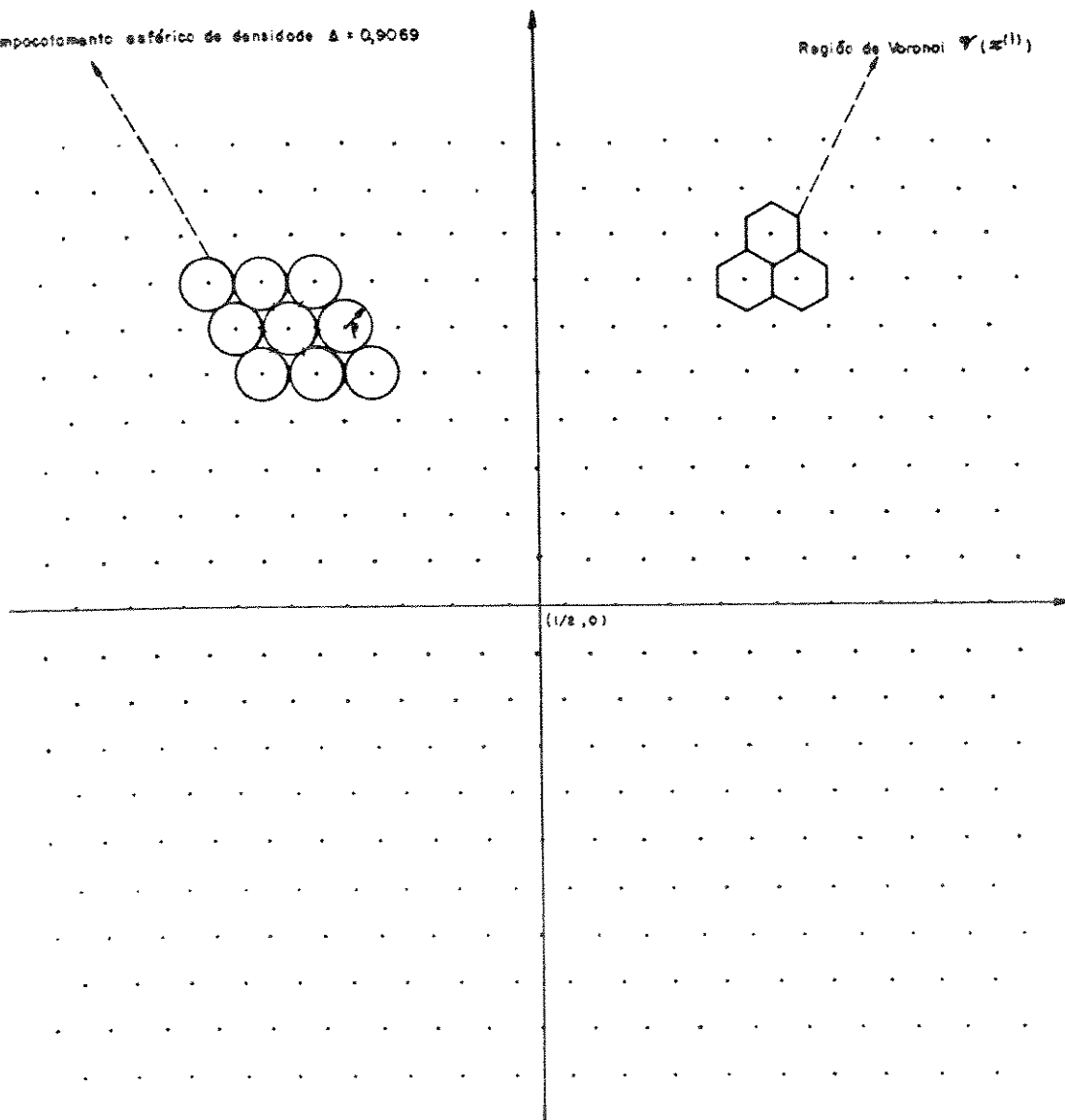
O algoritmo que utilizaremos para a procura de códigos ótimos utiliza-se da distância Euclideana entre os sinais em cada constelação gerada no reticulado. Portanto, ao igualarmos a energia média total a 1, obtemos uma distância Δ_0 para cada constelação utilizada.

O centro da constelação II.4-b foi obtido de [6] uma vez que esta é a melhor disposição para 16 sinais contidos em $\mathbb{R}^2$, $\left\{ \left| a_n^2 \right| \right\} = \frac{35}{16} \Delta_0^2 = 2.1875 \Delta_0^2$ para a centralização de 16 sinais ao redor de $(-\frac{1}{4}, 0)$ no reticulado A_2 . Para a constelação II.4-b foi adotada a origem no ponto-lacuna $(\frac{1}{2}, \frac{\sqrt{3}}{2})$, proporcionando $\left\{ \left| a_n^2 \right| \right\} = \frac{13}{12} \Delta_0^2$. Esta constelação aparece sem ponto origem em [28]. As demais constelações foram propostas como contribuição deste trabalho.

A constelação II.4-c aparece neste trabalho como alternativa à II.4.a, apesar desta constelação configurar 32 pontos com a menor energia em 2 dimensões. A constelação alternativa é mais eficiente em termos da Partição de Conjunto.

II - Reticulado tipo A_2

Empacotamento esférico de densidade $\Delta = 0,9069$



Região de Voronoi $\mathcal{V}(z^{(1)})$

FIGURA II

Vetores geradores $[1, 0], [-1/2, \sqrt{3}/2]$

$$M = \begin{bmatrix} 1 & 0 \\ -1/2 & \sqrt{3}/2 \end{bmatrix}$$

$$d = 2$$

$$\det = \sqrt{3}/2$$

$$\rho = 1/2$$

Centrada no ponto-lacuna $(1/2, 0)$

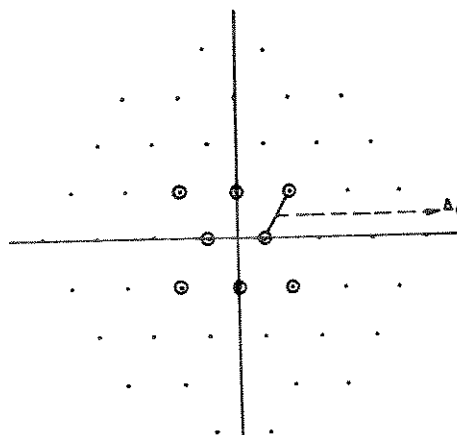
$$\tau(z) = 6$$

$$\Delta = 0,9069$$

$$\mathcal{V}(z^{(1)}) = \frac{\sqrt{3}}{2} \Delta_0^2$$

Constelações em A_2
 II. (4) m = 8 sinais

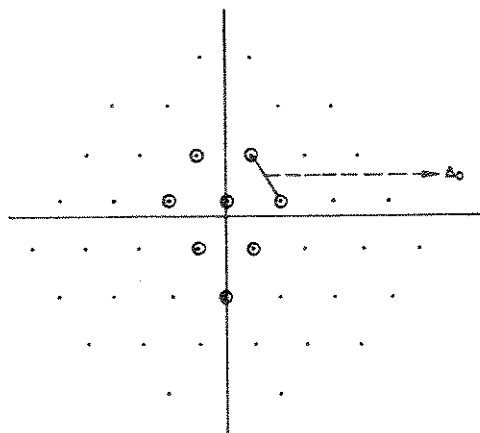
Configuração (a)
 Centrado em $(1/2, 0)$



$$\frac{1}{8} \{ |a_n|^2 \} = \frac{9}{8} A_0^2 = 1$$

$$A_0 = 0.942$$

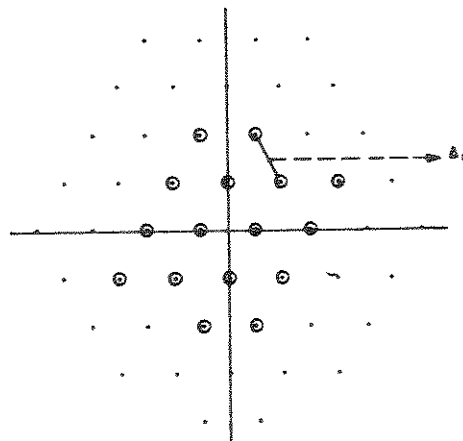
III. (4) Configuração (b)
 Centrado em $(1/2, \frac{\sqrt{3}}{3})$



$$\frac{1}{8} \{ |a_n|^2 \} = \frac{13}{8} A_0^2 = 1$$

$$A_0 = 0.960$$

IV. (16) m = 16 sinais
 Configuração (a)
 Centrado em $(1/2, 0)$



$$\frac{1}{16} \{ |a_n|^2 \} = \frac{36}{16} A_0^2 = 1$$

$$A_0 = 0.666$$

II. (44) $m = 16$ sinais
 Configuração (b)
 Centrada em $(-1/4, 0)$

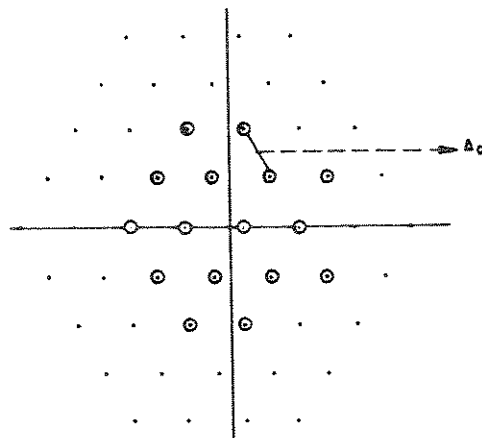


Fig. II. 44-b

$$\frac{1}{16} \{ |a_n|^2 \} = \frac{35}{16} \Delta_0^2 = 1$$

$$\Delta_0 = 0,676$$

II. (44) $m = 16$ sinais
 Configuração (c)
 Centrada em $(0,0)$

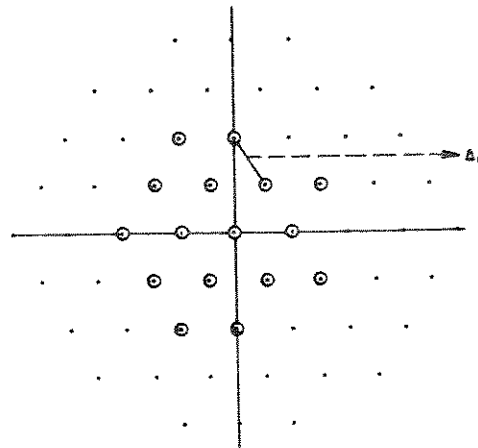
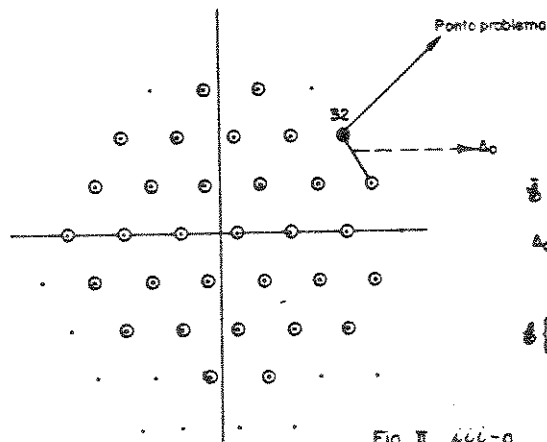


Fig. II. 44-c

$$\frac{1}{16} \{ |a_n|^2 \} = \frac{36}{16} \Delta_0^2 = 1$$

$$\Delta_0 = 0,666$$

II. (44) $m = 32$ sinais
 Configuração (a)
 Centrada em $(1/8, 0)$



$$\frac{1}{32} \{ |a_n|^2 \} = \frac{283}{64} \Delta_0^2 = 1$$

$$\Delta_0 = 0,475$$

$$\frac{1}{32} \{ |a_{32}|^2 \} = \frac{1}{32} \left(\frac{129}{16} \Delta_0^2 \right) = 0,25 \Delta_0^2$$

Fig. II. 444-a

II. (444) $m = 32$ sinais
 Configuração (b)
 Centrada em $(1/2, 0)$

$$\frac{1}{32} \{ |a_n^2| \} = \frac{71}{16} \Delta_0^2 = 1$$

$$\Delta_0 = 0.474$$

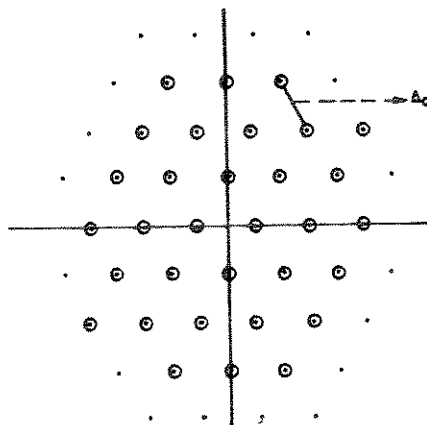
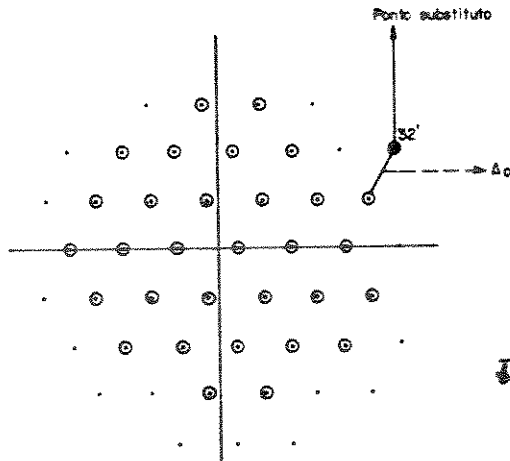


FIG. II. iii-b

$$\frac{1}{32} \{ |a_n^2| \} = \frac{1}{32} \left\{ 2 \left(\frac{\Delta_0^2}{4} + \frac{9}{4} \Delta_0^2 + \frac{3}{4} \Delta_0^2 + \frac{27}{4} \Delta_0^2 + \frac{57}{4} \Delta_0^2 + \frac{25}{4} \Delta_0^2 \right) + 4 \left(\frac{7}{4} \Delta_0^2 + \frac{19}{4} \Delta_0^2 + \frac{13}{4} \Delta_0^2 + \frac{21}{4} \Delta_0^2 + \frac{31}{4} \Delta_0^2 \right) \right\}$$

II. (444) $m = 32$ sinais
 Configuração (c)
 = II. (444) (a) modificada
 Centrada em $(-\frac{1}{4}, 0)$



$$\frac{1}{32} \{ |a_n^2| \} = \frac{147}{32} \Delta_0^2 = 1$$

$$\Delta_0 = 0.466$$

$$\frac{1}{32} \{ |a_{32}^2| \} = \frac{1}{32} \left(\frac{217}{16} \Delta_0^2 \right) = 0.423 \Delta_0^2$$

FIG. II. iii-c

$$\frac{1}{32} \{ |a_n^2| \} = \frac{1}{32} \left\{ \frac{\Delta_0^2}{16} + \frac{25\Delta_0^2}{16} + \frac{91\Delta_0^2}{16} + \frac{9\Delta_0^2}{16} + \frac{49\Delta_0^2}{16} + \frac{121\Delta_0^2}{16} + \frac{129\Delta_0^2}{16} + \frac{217\Delta_0^2}{16} + 2 \left(\frac{109\Delta_0^2}{16} + \frac{13\Delta_0^2}{16} + \frac{93\Delta_0^2}{16} + \frac{37\Delta_0^2}{16} + \right. \right.$$

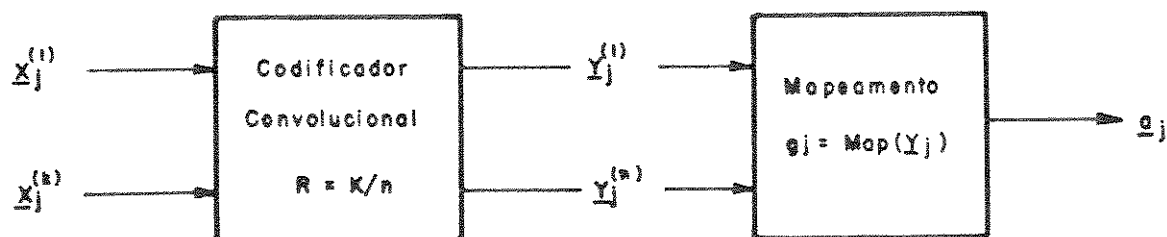
$$\left. \left. + \frac{21\Delta_0^2}{16} + \frac{61\Delta_0^2}{16} + \frac{133\Delta_0^2}{16} + \frac{97\Delta_0^2}{16} + \frac{57\Delta_0^2}{16} + \frac{49\Delta_0^2}{16} + \frac{73\Delta_0^2}{16} + \frac{117\Delta_0^2}{16} \right) \right\}$$

II.4 - PARTIÇÃO DE CONJUNTO DE SINAIS

A Partição de Conjunto é um processo obtido como resultado de sucessivas partições da constelação de sinais em subconjuntos com distâncias Euclidianas mínimas crescentes ($\Delta_0 \leq \Delta_1 \leq \dots \leq \Delta_{n-1}$).

Conforme visto no Capítulo 1, a estrutura básica do codificador TCM consiste de um bloco associado ao codificador convolucional e de outro associado ao arranjo de pontos no espaço de sinais.

Através de um mapeamento apropriado das saídas do codificador convolucional aos pontos de sinais de uma constelação contida em um determinado reticulado, obtém-se a forma combinada de codificação e modulação.



Cada palavra-código depende da informação x_j e das m mensagens anteriores armazenadas nos registros do codificador.

Considera-se que o codificador convolucional é binário de taxa $R = k/n$ e que este tem suas saídas mapeadas aos 2^n pontos de sinais. O mapeamento $a_j = Map(x_j)$ ocorre a cada T segundos, tempo de duração de cada k dígitos de informação. Neste trabalho consideramos $n = k+1$. Como o número de sinais é 2^n então temos 2^{k+1} sinais na constelação. Portanto, indiretamente relaciona-se 2^k sinais na entrada com 2×2^k sinais na constelação. Desta forma, a redundância introduzida pelo processo de codificação está relacionada com a expansão do conjunto de sinais.

Cada transição de estado do Codificador Convolucional é representado no diagrama de treliça por um ramo na treliça. Cada ramo na treliça representa uma palavra-código-ramo do código TCM. O mapeamento a que nos referimos consiste em associar-se apropriadamente cada palavra-código-ramo com um subconjunto terminal resultante da partição do conjunto de sinais. Este mapeamento é resultado de uma associação biunívoca entre as palavras-código-ramo e os pontos da constelação feito de maneira a garantir uma máxima distância Euclideana intraconjuntos da partição.

O processo de Partição de Conjunto aliado às Regras de Codificação (mapeamento) [Cap.1, sec.III.4], aparecem como solução para o impasse criado por expandir-se o número de pontos na constelação ao mesmo tempo que se mantêm inalterados as condições do ruído e energia média dos sinais recebidos.

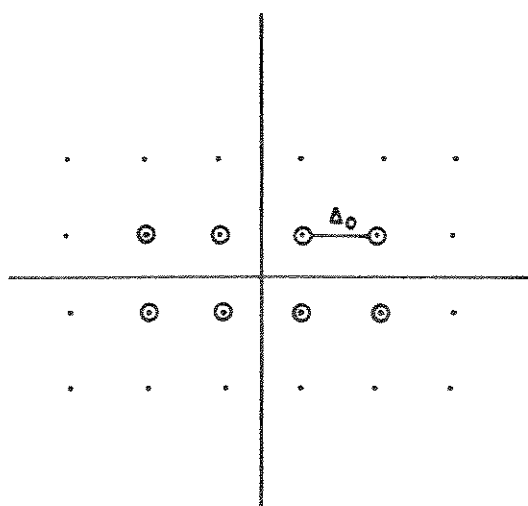
Vamos supor como exemplo a transmissão do conjunto de 8 sinais (I.4-a) obtidos de Z^2 com $\bar{E}\{|a_8^2|\} = \frac{8}{2} \Delta_0^2 = 1$ então a distância Euclideana entre quaisquer 2 pontos consecutivos desta configuração vale $\Delta_0 = 0.816$. Uma constelação com o dobro de sinais (I.4b) oriundos do mesmo reticulado, transmitidos à mesma energia média $\bar{E}\{|a_{16}^2|\} = \frac{40}{16} \Delta_0^2 = 1$, possui uma distância $\Delta_0 = 0.632$ entre os sinais que a compõem, isto é, estão bem mais próximos entre si.

Por outro lado sabe-se que o decodificador ao receber o sinal $r = a_j + n$, onde n é o ruído Gaussiano branco aditivo, pode decidir por a_k diferente de a_j . Portanto, quanto mais próximo estiver a_k de a_j (dois sinais consecutivos na constelação) em termos de distância Euclideana, maior a probabilidade de decisão errônea. Como as condições de ruído por hipótese são as mesmas, ao expandir-se a constelação temos como consequência um aumento na ocorrência de erros, se compararmos com a transmissão dos sinais da constelação não expandida.

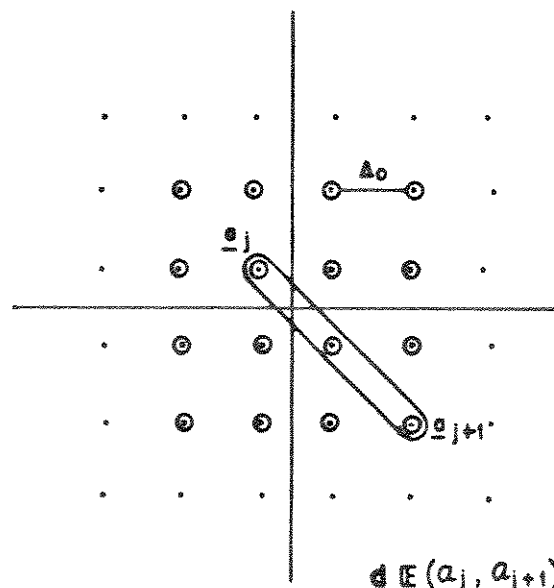
A codificação TCM soluciona este problema. A idéia está em associar ao mesmo tempo a saída y_j a um sinal a_j bem como associar as possíveis saídas y_{j+1} , mais prováveis de ocorrerem após y_j , aos sinais a_{j+1} , onde a_{j+1} são os pontos de sinais que guardam uma razoável distância Euclideana em relação à a_j . A Partição de Conjunto possibilita escolher os sinais a_{j+1} mais distantes de a_j . Ao realizar o mapeamento, o codificador convolucional fica especificado através das regras de partição e codificação.

A ilustração a seguir apresenta a constelação original de 8 sinais I.4-a e a correspondente constelação expandida I.4. Observe como pode-se escolher a_{j+1} mantendo-se a_j fixo de tal maneira que a distância $dE(a_j, a_{j+1})$ seja maior que a distância mínima Δ_0 da constelação original.

Este processo procura maximizar a distância Euclídeana entre dois pontos associados a y_j e y_{j+1} , isto é, maximizar $dE(a_j, a_{j+1})$ e desta forma garantir que a probabilidade de erro na decodificação dos sinais na constelação expandida seja minimizada. A escolha de a_{j+1} é possível para cada a_j da constelação.



Constelação original I.4



Constelação expandida I.4

O processo de escolha de a_{j+1} é obtido a partir das seguintes regras de Partição de Conjunto:

II.4.1 - REGRAS PARA PARTIÇÃO DE CONJUNTOS

- (1) - O conjunto de sinais deve ser dividido sucessivamente em dois subconjuntos de tal maneira que as distâncias Euclidianas sejam no mínimo maiores ou iguais do que as do conjunto que lhes deu origem;
- (2) - A partição é repetida sistematicamente até obter-se 2^k subconjuntos terminais onde k é o número de saídas codificadas.

É importante frisar que não existe apenas uma forma única de subdivisão de conjuntos. O processo estabelece uma relação de ordem crescente entre as distâncias à cada subdivisão. Portanto a distância mínima entre dois subconjuntos é a distância mínima dos sinais do conjunto origem.

II.4.2 - PARTIÇÃO DE CONSTELAÇÕES TIPO Z^2

As Figuras ($Z^2.8-a$, $Z^2.8-b$, $Z^2.8-b'$, $Z^2.16$, $Z^2.32$) apresentam os conjuntos de 8,16,32 pontos tipo Z^2 particionados de acordo com as regras mencionadas acima.

Devido à simetria e regularidade dos sinais neste reticulado a partição ocorre quase que sistematicamente para estes conjuntos de pontos, porém, não de forma única. É importante ressaltar que a estratégia de subdivisão procura levar aos conjuntos terminais, a maior distância possível entre os pontos que os compõem. Portanto a primeira composição encontrada pode não ser a mais eficiente.

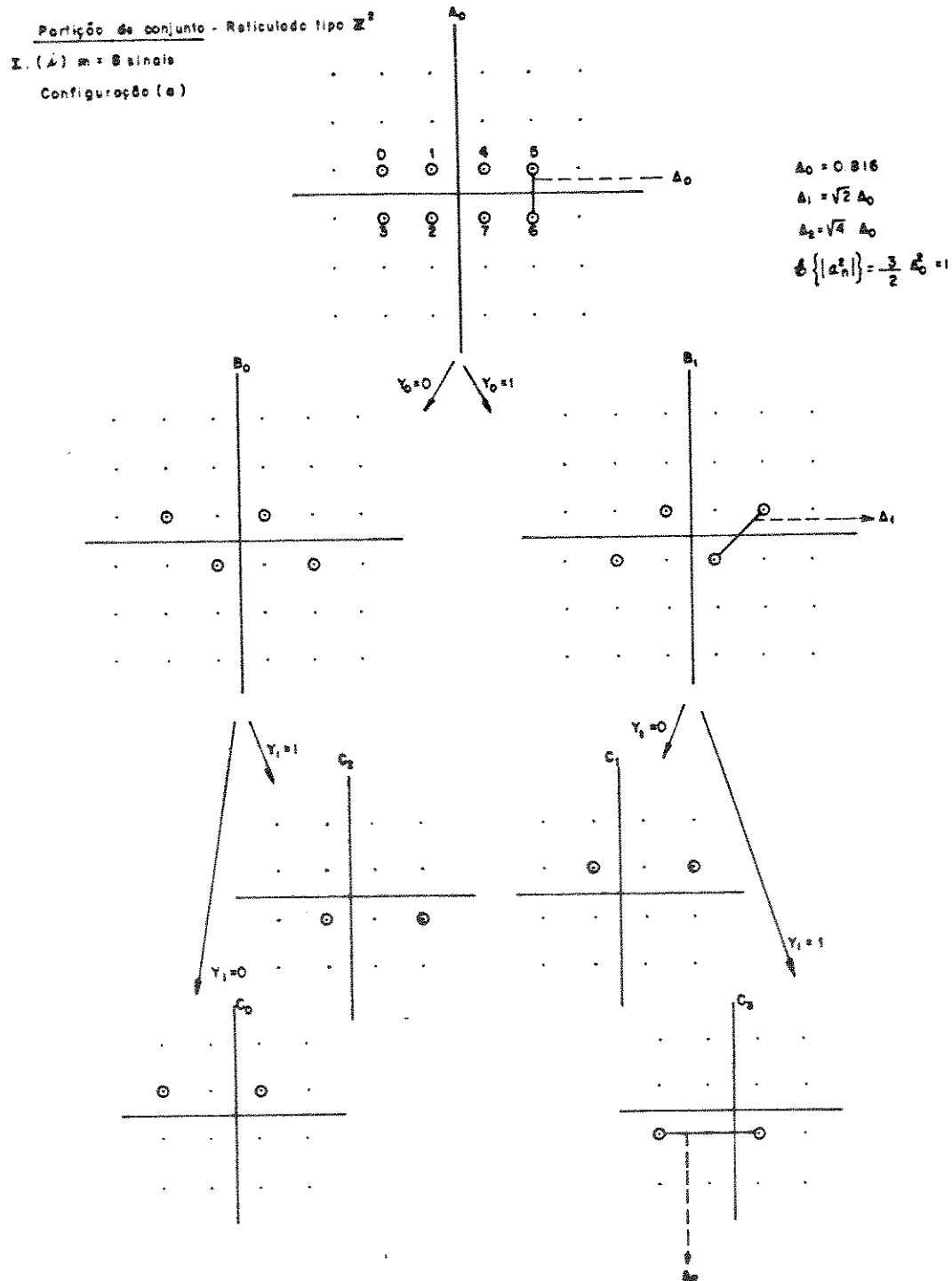
Um exemplo que caracteriza este fato é mostrado nas duas partições encontradas para a constelação I.4-b. Apesar das duas composições apresentarem o mesmo Δ_0 e Δ_1 , diferem quanto ao Δ_2 . A estratégia da divisão dos sinais é diferente (observe as figuras, $Z^2.8-b$ e $Z^2.8-b'$). Obtém-se para a composição b — $\Delta_2 = \sqrt{8} \Delta_0$, e para a composição b' — $\Delta_2 = \sqrt{5} \Delta_0$. Dessa forma, espera-se um desempenho superior para a composição b de sinais. Isto deverá ser confirmado, por ocasião da procura por codificadores ótimos.

PARTIÇÃO DE CONJUNTO DE SINAIS TIPO Z^2

sinais $\rightarrow$ $\frac{d\mathbf{f}}{dt}$	8-a	8-b	8-b'	16	32
Δ_0	0.816	0.632	0.632	0.632	0.441
Δ_1	1.153	0.893	0.893	0.893	0.623
Δ_2	1.632	1.787	1.418	1.264	0.882
Δ_3	----	----	----	1.787	1.247
Δ_4	----	----	----	----	1.247
$\overline{\mathcal{S}}\left\{ a_n^2 \right\}$	$1.5\Delta_0^2$	$2.5\Delta_0^2$	$2.5\Delta_0^2$	$2.5\Delta_0^2$	$5.125\Delta_0^2$

TABELA I

$$\overline{\mathcal{S}}\left\{|a_n^2|\right\} = 1$$



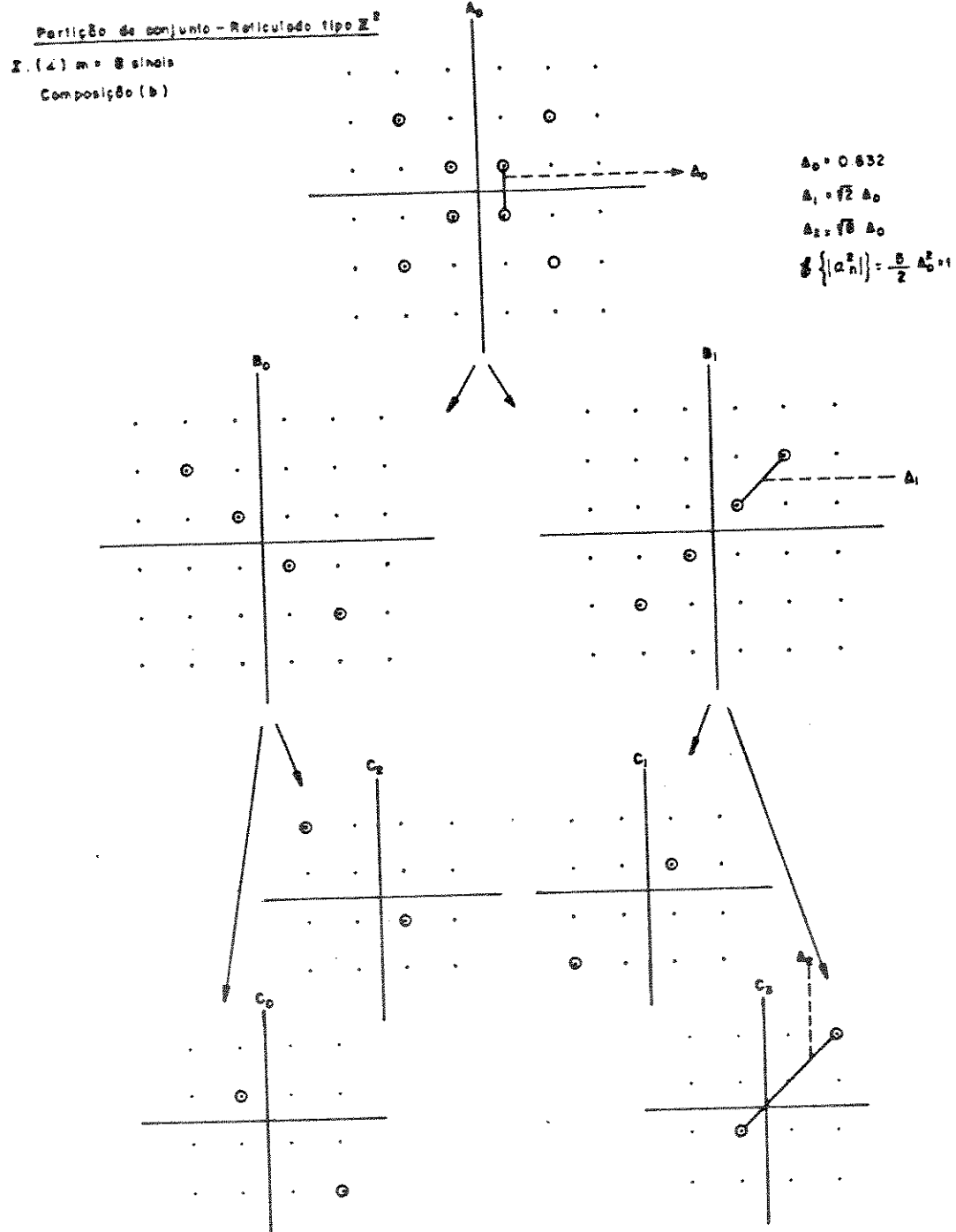


FIGURA $\mathbb{Z}^2$. 6 - b

Partição de conjunto - Retículo tipo $\mathbb{Z}^2$

I. (Δ) $m = 8$ sinais

Composição (b')

* Partição alternativa

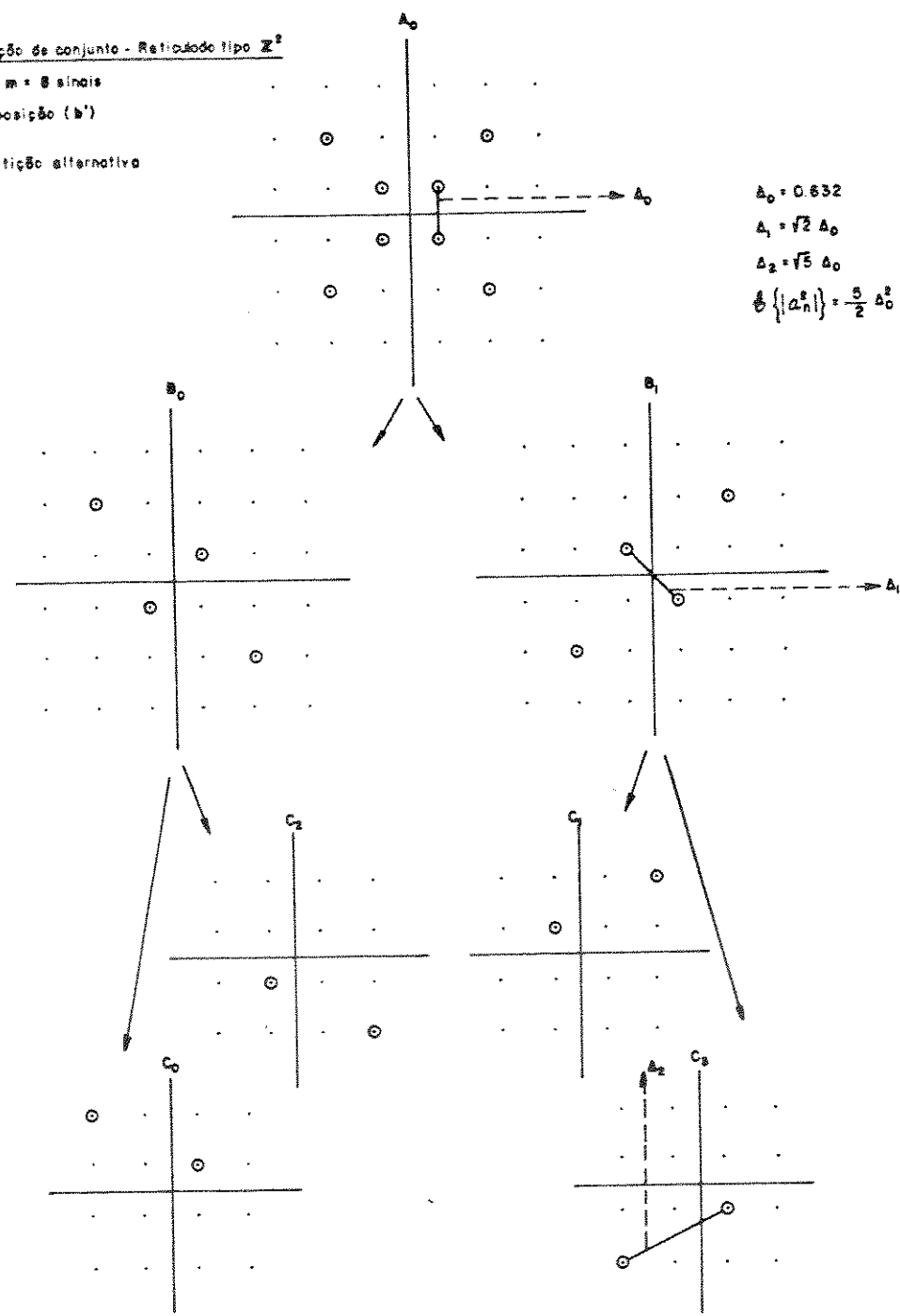
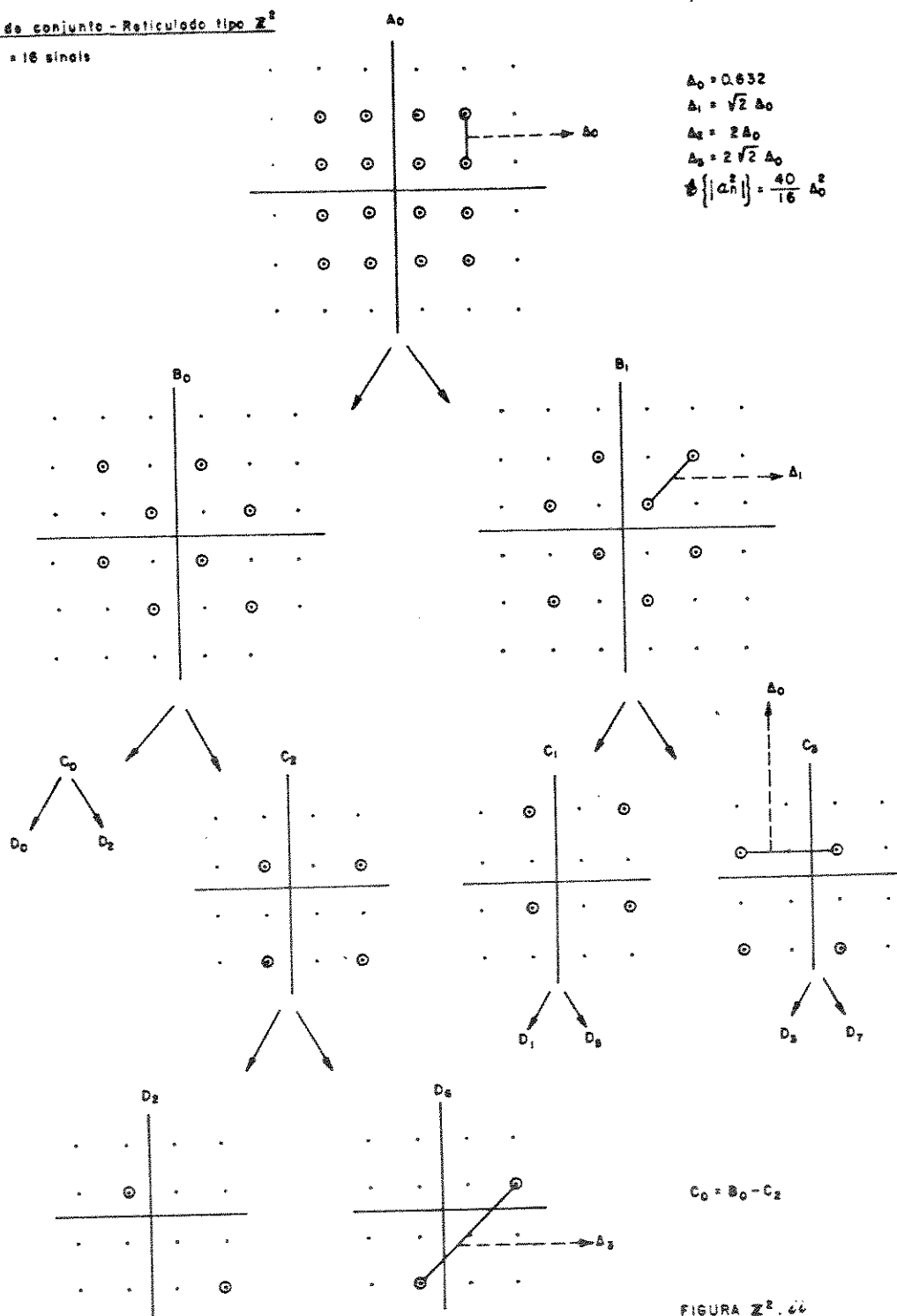


FIGURA $\mathbb{Z}^2, \Delta - b'$

Partição do conjunto - Reticulado tipo Z^2
 $\Sigma(\Delta_i) - m = 16$ sinais



Partição do conjunto - Reticulado tipo $\mathbb{Z}^2$
 $\mathbb{Z}(\Delta \Delta \Delta) m = 32$ sinais

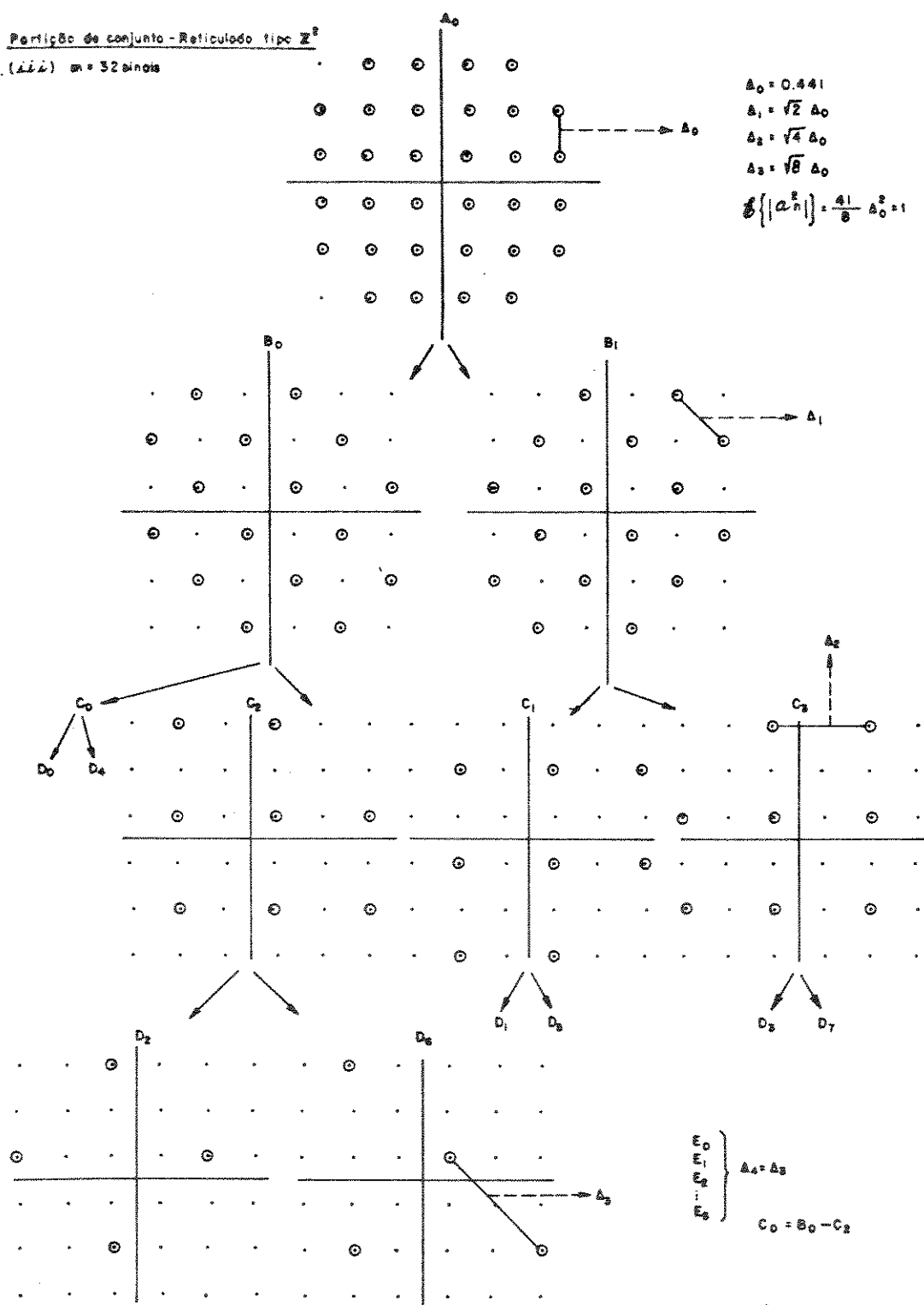


FIGURA $\mathbb{Z}^2$. 200

II.4.3. - PARTIÇÃO DE CONSTELAÇÕES TIPO A_2

As Figuras $A_2.8-a$, $A_2.8-b$, $A_2.16-a$, $A_2.16-b$, $A_2.16-c$, $A_2.32-a$, $A_2.32-b$, $A_2.32-c$, apresentam a partição para os conjuntos de 8, 16 e 32 pontos tipo A_2 .

A partição de pontos neste reticulado não ocorre de maneira sistemática como no reticulado Z^2 . Nem todas as constelações apresentam simetria, devido ao posicionamento dos eixos coordenados de forma a minimizar a energia de transmissão, não obstante a este fato as regras de partição foram obedecidas.

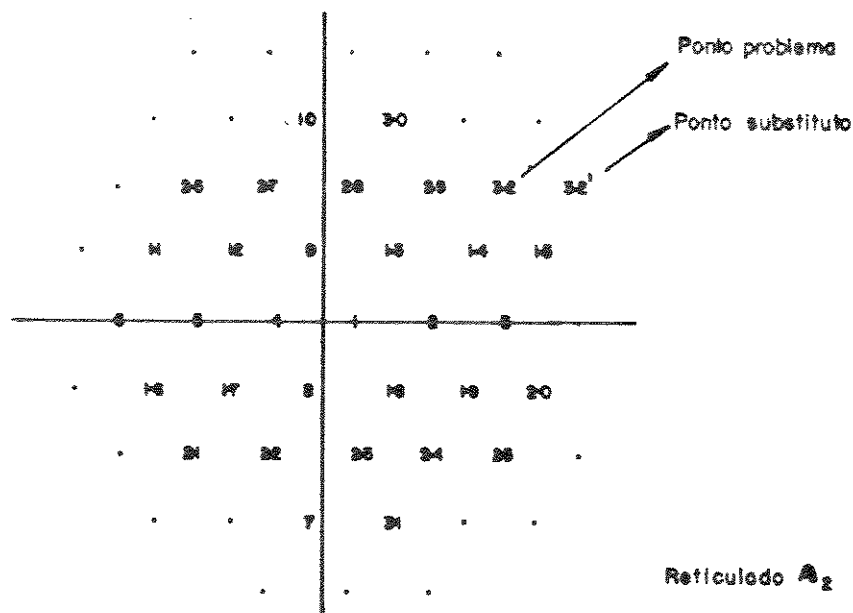
O comportamento dos pontos neste reticulado sugere várias estratégias de subdivisão dos conjuntos — escolhe-se a de maior distância intraconjunto no último nível da partição — mesmo assim observa-se que a distância mínima (Δ_i) algumas vezes repete-se em subdivisões consecutivas, como pode ser observado nas figuras.

No caso da constelação II.iii-a não foi encontrada uma partição de conjunto satisfatória para este conjunto de pontos, isto é, à medida em que ocorrem as subdivisões, os subconjuntos obtidos mantêm a mesma distância Δ_0 que caracteriza a distância inicial entre os pontos da constelação. Ao analisarmos a partição (Fig. $A_2.32-a$), observa-se um ponto que designamos de "ponto problema" (sinal α_{32}). Este sinal impossibilita, devido ao seu posicionamento, qualquer aumento de distância intraconjunto até o

terceiro nível da árvore, o que faz desta constelação apesar de "ótima" em termos de quantidade mínima de energia para 32 sinais em $\mathbb{R}^2$, ineficiente ao se tratar da partição de conjunto dos seus pontos.

Como alternativa, foi feito o estudo da mudança de posição deste sinal versus o desempenho relativo de energia, com o objetivo de obter-se uma partição coerente e uma constelação com energia média ainda satisfatória.

Encontrou-se então a constelação II.iii-c que tem como base II.iii-a (apenas mudando a posição de a_{32} para a_{32}') onerada em apenas $0.172\Delta_0^2$ de energia em relação à constelação "ótima" original.



$$\overline{\{ |a_n^2| \}} = \frac{1}{32} \left\{ a_1^2 + a_2^2 + a_3^2 + a_4^2 + a_5^2 + a_6^2 + 2 \left[a_7^2 + a_8^2 + a_{11}^2 + a_{12}^2 + a_{13}^2 + a_{14}^2 + a_{15}^2 + a_{21}^2 + a_{22}^2 + a_{23}^2 + a_{24}^2 + a_{30}^2 + a_{32}^2 \right] \right\}$$

$$\overline{\{ |a_n^2| \}} = \frac{1}{32} \left\{ \frac{\Delta_0^2}{16} + \frac{25\Delta_0^2}{16} + \frac{81\Delta_0^2}{16} + \frac{9\Delta_0^2}{16} + \frac{49\Delta_0^2}{16} + \frac{121\Delta_0^2}{16} + 2 \left[\frac{109\Delta_0^2}{16} + \frac{13\Delta_0^2}{16} + \frac{93\Delta_0^2}{16} + \frac{37\Delta_0^2}{16} + \frac{21\Delta_0^2}{16} + \frac{61\Delta_0^2}{16} + \frac{133\Delta_0^2}{16} + \frac{97\Delta_0^2}{16} + \frac{57\Delta_0^2}{16} + \frac{49\Delta_0^2}{16} + \frac{73\Delta_0^2}{16} + \frac{117\Delta_0^2}{16} + \frac{129\Delta_0^2}{16} \right] \right\}$$

$$\text{Energia Média Total de II. iii-a} = \overline{\{ |a_n^2| \}} = \frac{283\Delta_0^2}{64} = 4.421\Delta_0^2 = 1$$

$$\rightarrow \Delta_0 = 0.475$$

$$\overline{\{ |a_{32}^2| \}} = \frac{1}{32} \left(\frac{129\Delta_0^2}{16} \right) = 0.251\Delta_0^2$$

$$\text{Energia Média Total de II. iii-c} = \overline{\{ |a_n^2| \}} = \frac{147\Delta_0^2}{32} = 4.593\Delta_0^2 = 1$$

$$\rightarrow \Delta_0 = 0.466$$

$$\overline{\{ |a_{32'}^2| \}} = \frac{1}{32} \left(\frac{217\Delta_0^2}{16} \right) = 0.423\Delta_0^2$$

$$\overline{\{ |a_{32'}^2| \}} - \overline{\{ |a_{32}^2| \}} = 0.172\Delta_0^2$$

Como pode ser observado nos cálculos anteriores o valor Δ_0 de II.4-c é menor do que o de II.4-a, porém, como também pode ser observado na Figura A₂.32-c, a partição de II.4-c obtém já no segundo nível da árvore uma distância intraconjunto de $2\Delta_0 = 0.932$ ao passo que em A₂.32-a, ainda é mantido o mesmo $\Delta_0 = 0.475$ inicial para este mesmo nível. II.4-c é mais eficiente portanto que II.4-a. Entretanto a constelação II.4-b surge em meio à procura entre configurações de 32 sinais deste reticulado como a mais promissora, pois tem a mesma evolução do crescimento entre distâncias intraconjunto ($\Delta_0, \Delta_1 = \Delta_0, \Delta_2 = 2\Delta_0, \Delta_3 = 2\Delta_0, \Delta_4 = 2\sqrt{3}\Delta_0$) que II.4-c e ainda uma menor energia média ($\bar{x} \left\{ |a_n^2| \right\} = 4.437\Delta_0^2$). Devido a estes fatores a constelação II.4-b foi escolhida neste reticulado como base para procura de codificadores ótimos de taxa $\frac{4}{5}$.

As constelações II.4-a e II.4-c apesar de serem diferentes quanto ao posicionamento de seus sinais, possuem tanto a mesma energia média total quanto as mesmas distâncias intraconjuntos Δ_i (observar Figuras A₂.16-a, A₂.16-c e Tabela II). Portanto, a procura por codificadores só se faz necessária tomando como base apenas uma das constelações.

PARTIÇÃO DE CONJUNTO DE SINAIS TIPO A_2

SINAIS →	8-a	8-b	16-a	16-b	16-c	32-a	32-b	32-c
$d \in$								
A_0	0.943	0.961	0.666	0.676	0.666	0.475	0.474	0.466
A_1	0.943	0.961	0.666	0.676	0.666	0.475	0.474	0.466
A_2	1.632	1.664	1.333	1.352	1.333	0.475	0.949	0.933
A_3	—	—	1.333	1.352	1.333	0.950	0.949	0.933
A_4	—	—	—	—	—	1.256	1.644	1.616
$\sum_{i=0}^4 a_i ^2$	$1.125 A_0^2$	$1.083 A_0^2$	$2.250 A_0^2$	$2.187 A_0^2$	$2.250 A_0^2$	$4.421 A_0^2$	$4.437 A_0^2$	$4.593 A_0^2$

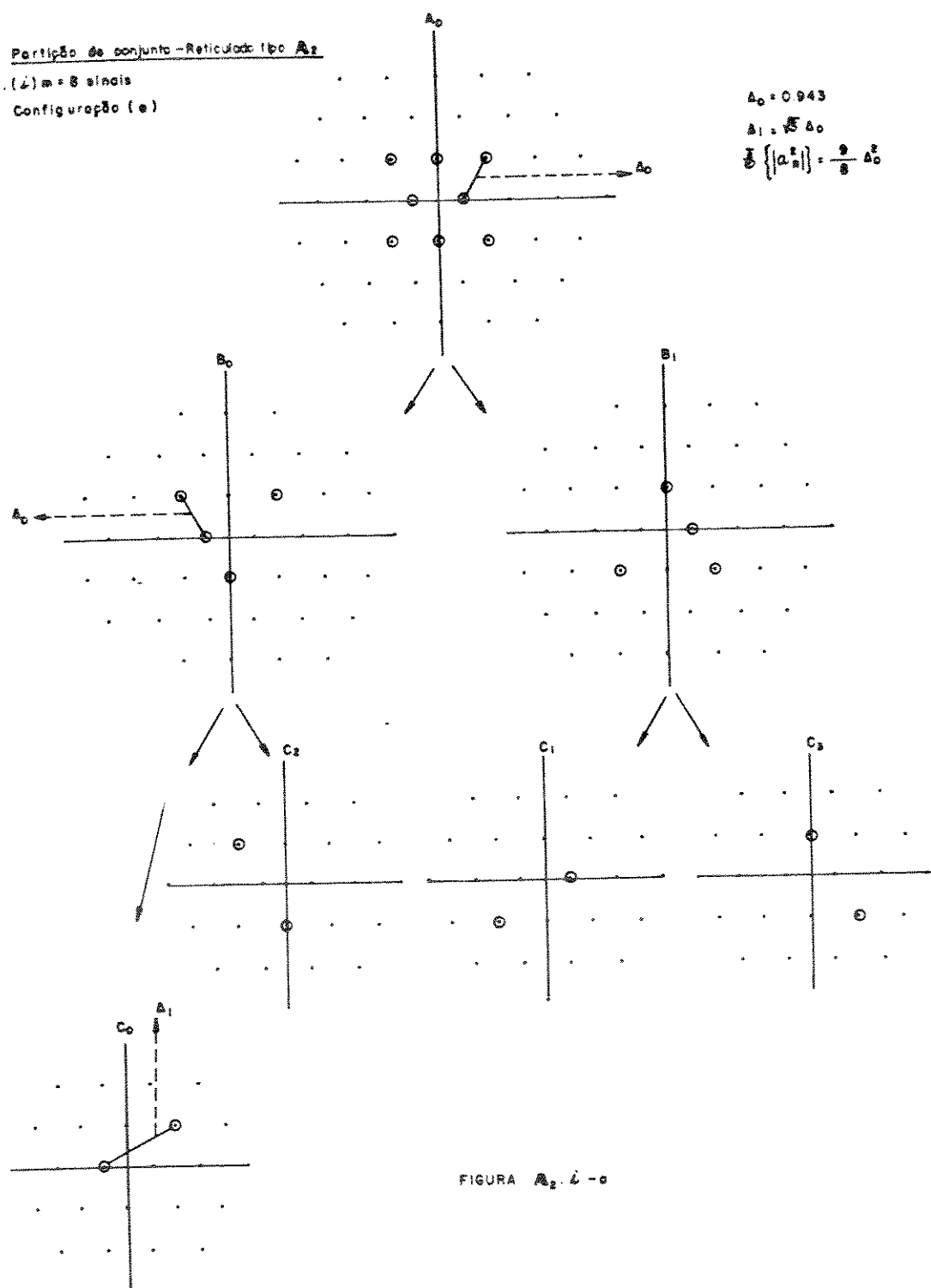
$\sum_{i=0}^4 |a_i|^2 = 1$

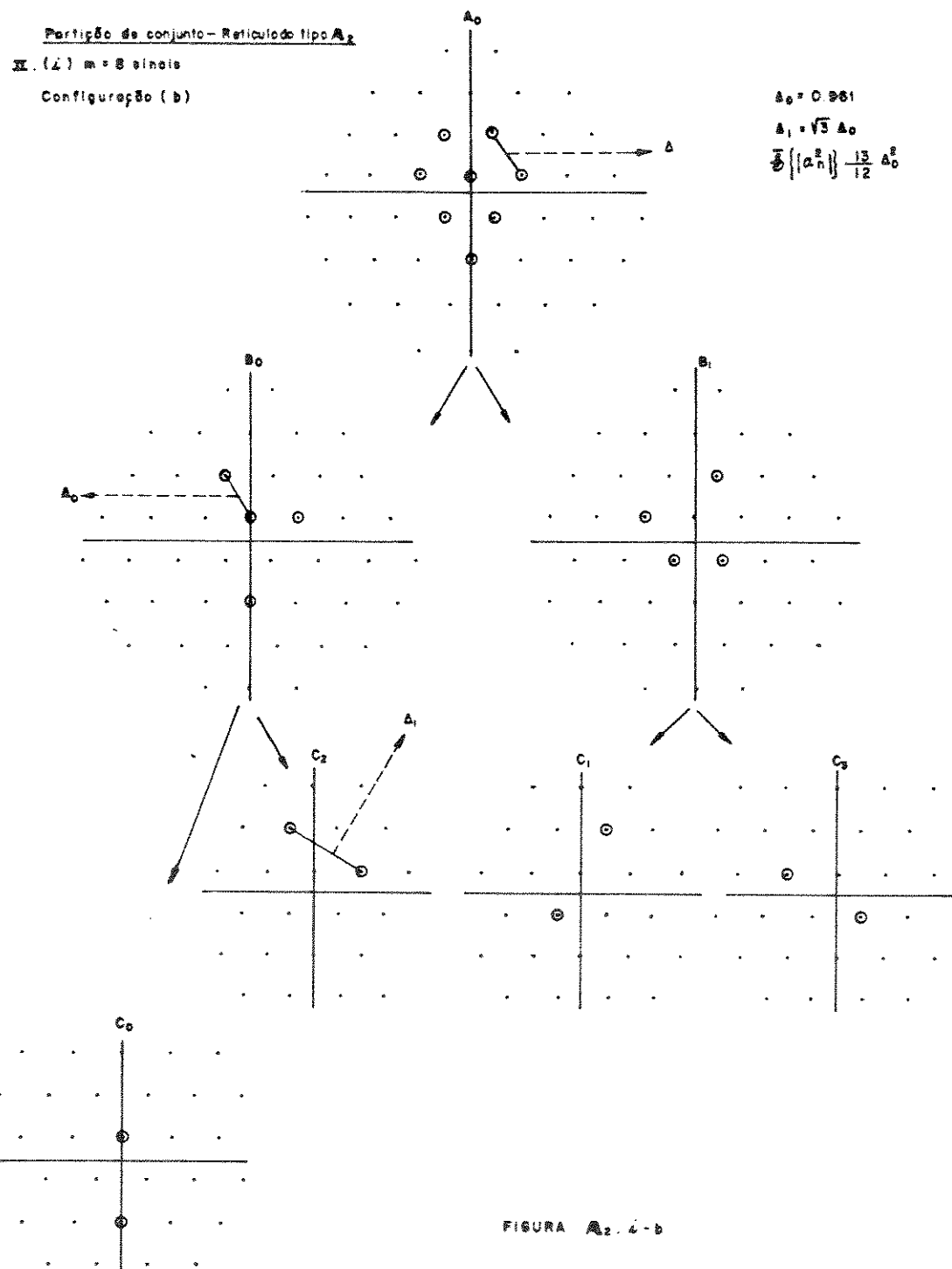
TABELA II

Partição de conjunto - Reticulado tipo A_2

II. (2) $m = 8$ sinais

Configuração (a)

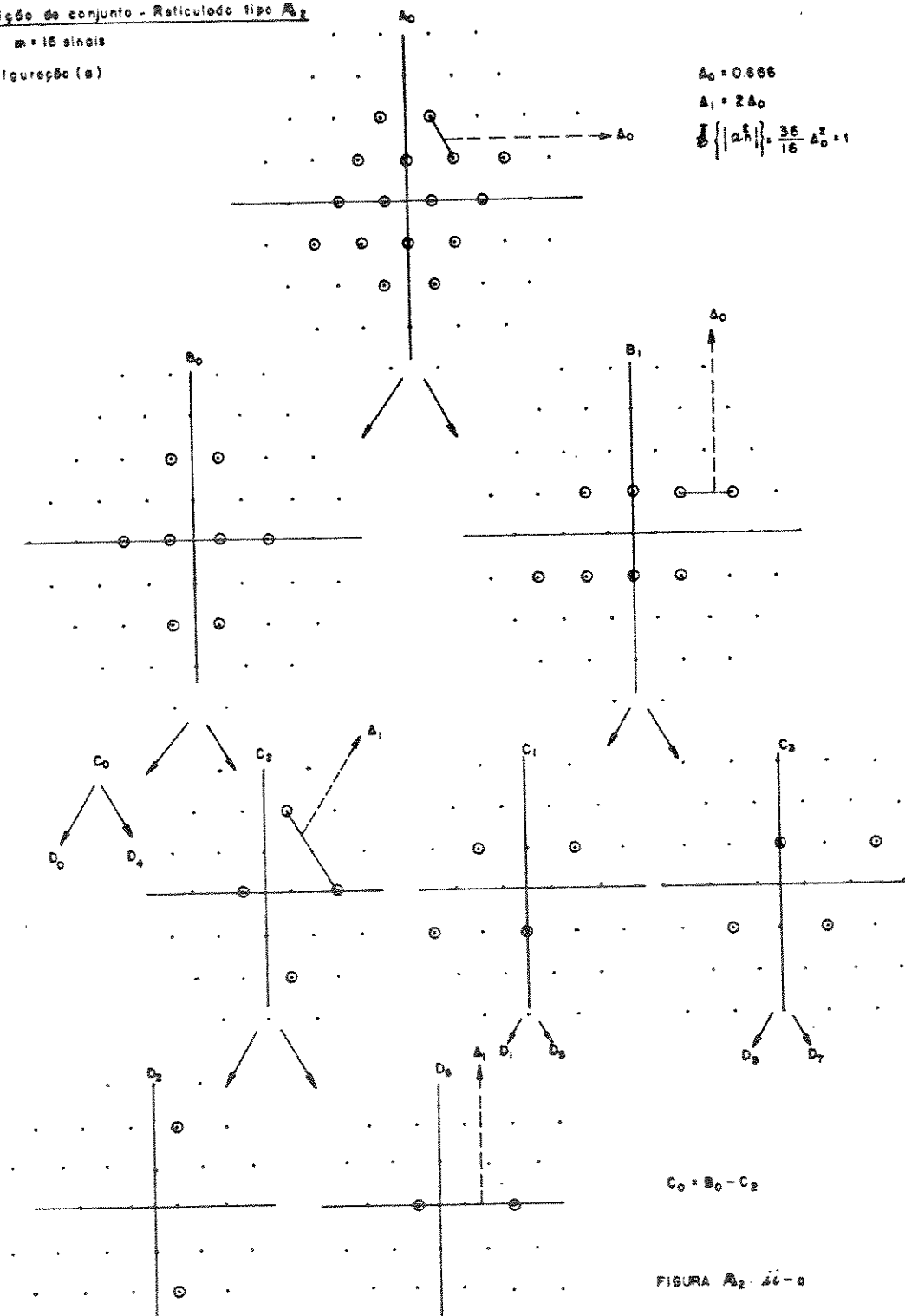




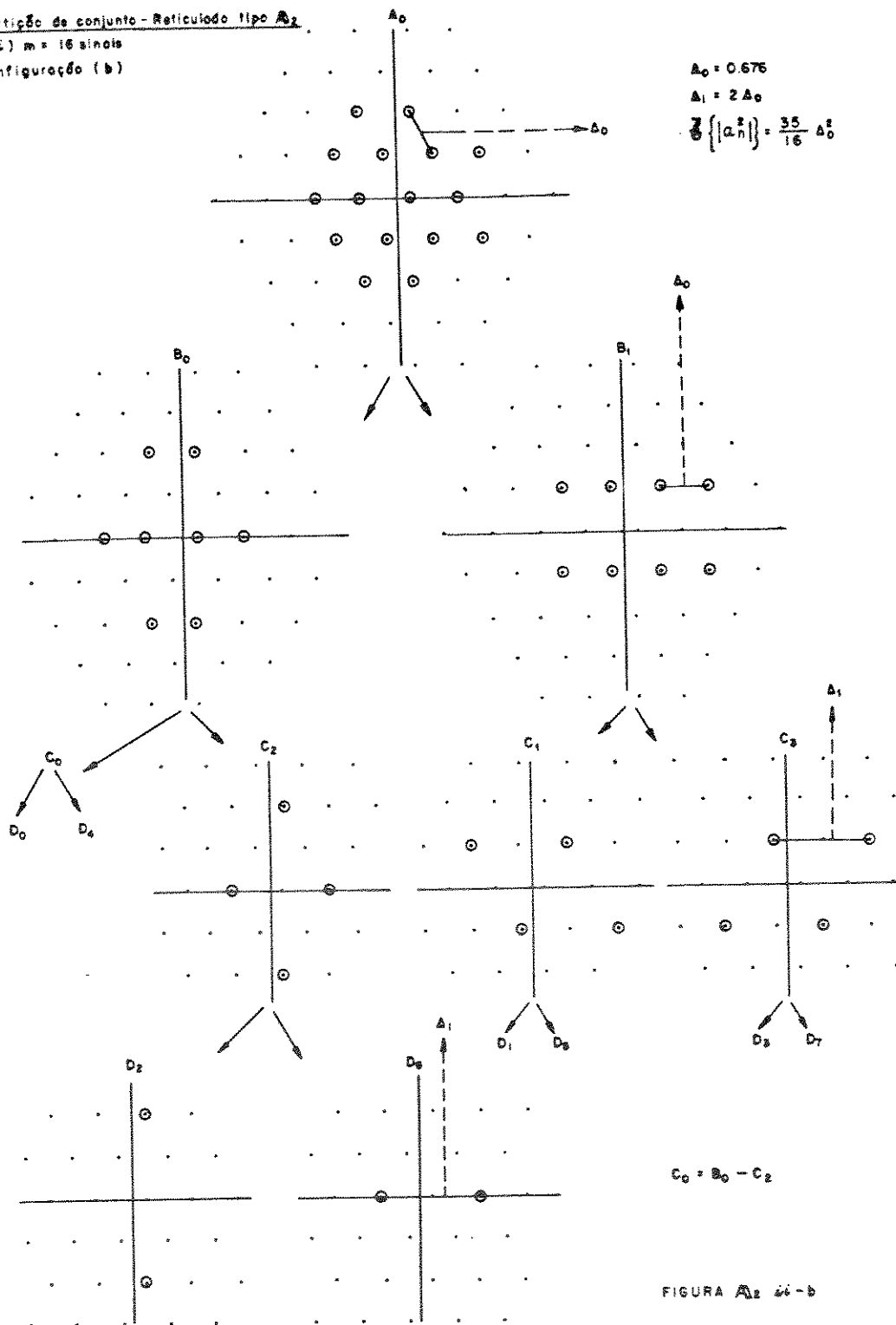
Partição de conjunto - Reticulado tipo A_2

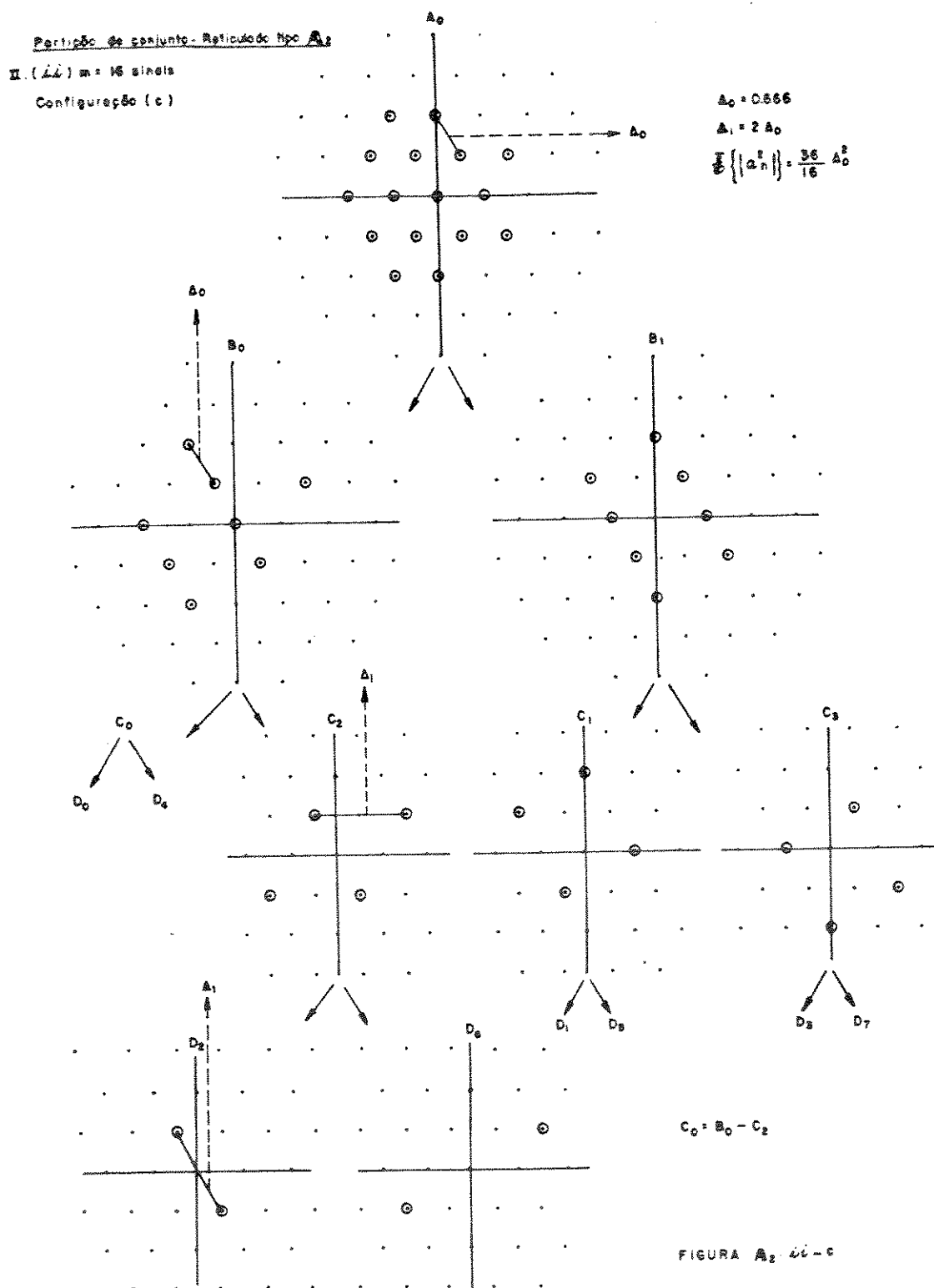
II. (24) $m = 16$ sinais

Configuração (a)



Partição de conjunto - Reticulado tipo A_2
 II. (44) $m = 16$ sinais
 Configuração (b)





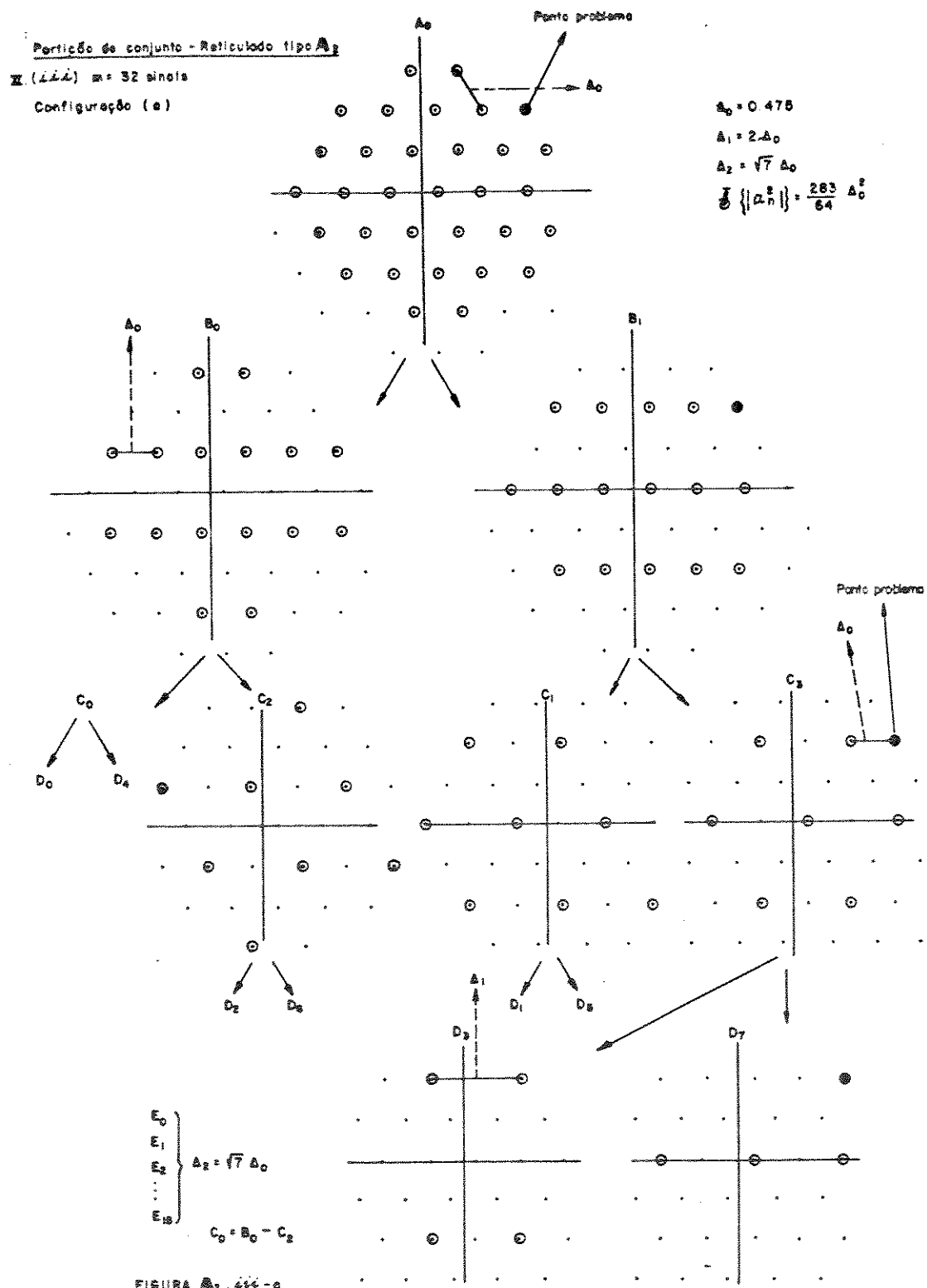
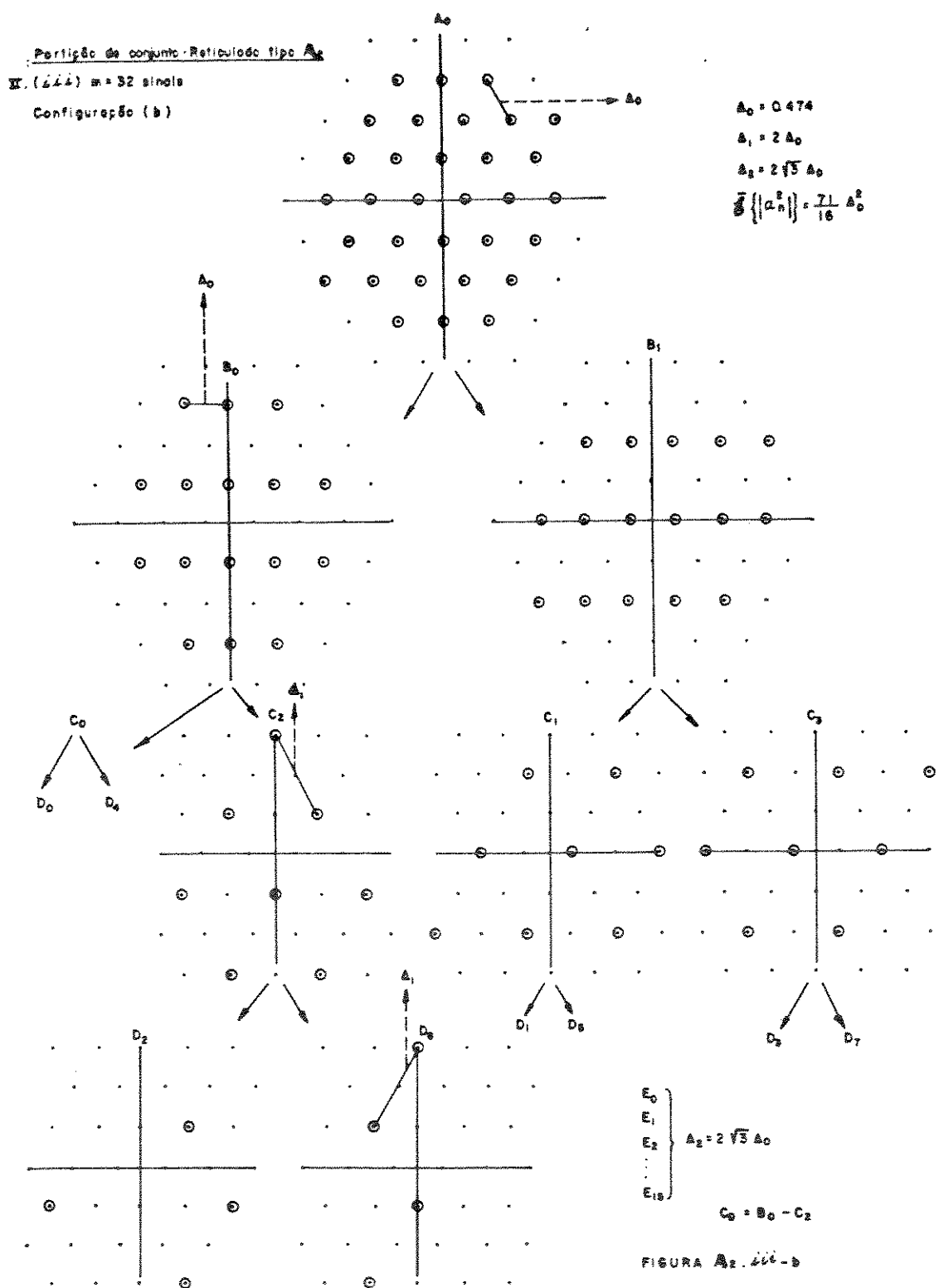
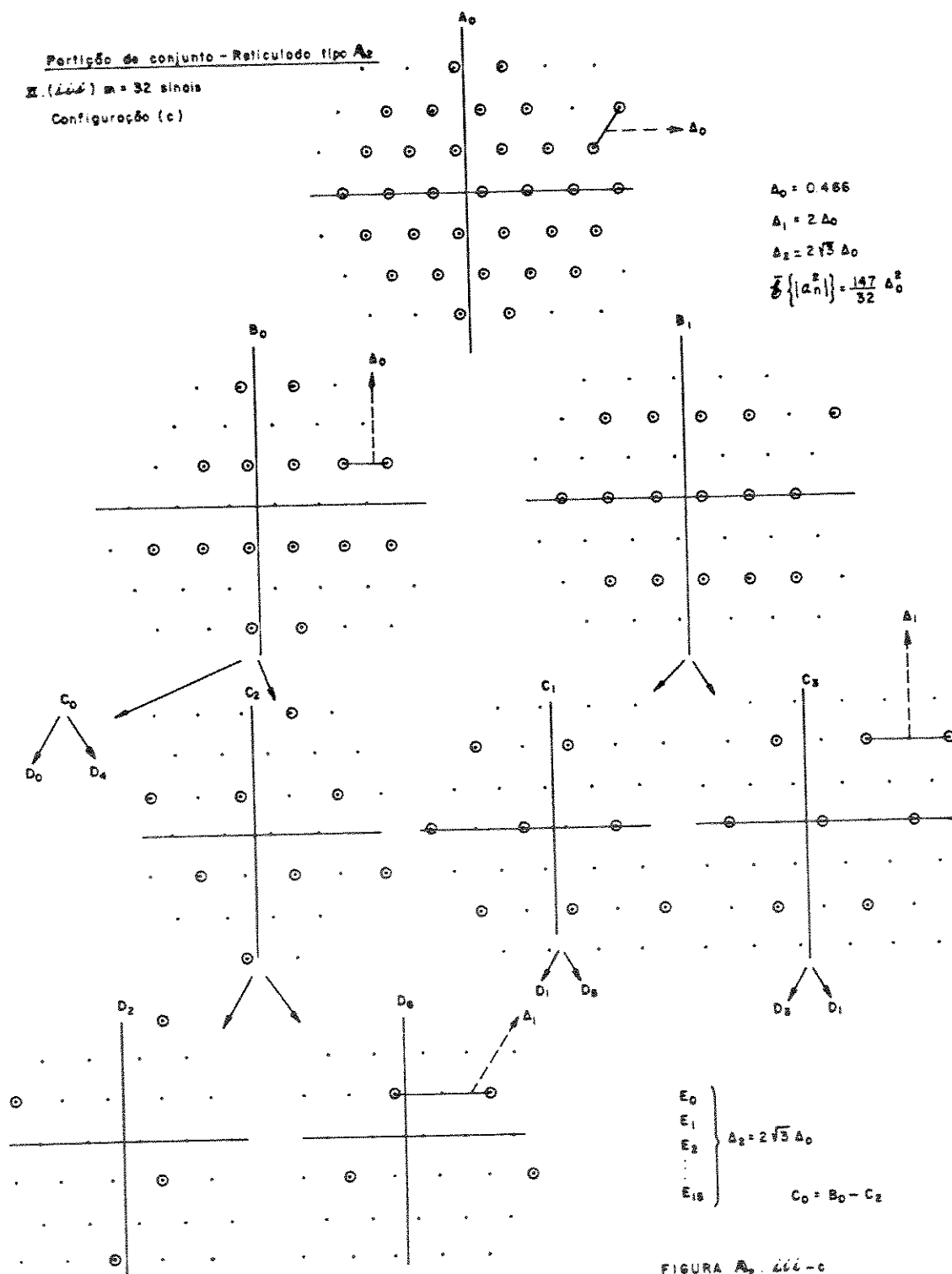


FIGURA A2.444-a





III - CONCLUSÕES

Neste Capítulo procuramos realizar o aspecto prático do estudo abordado no Capítulo 1. Para tanto caracterizaram-se geometricamente duas estruturas reticulares em 2 dimensões (A_2 e Z^2), propondo-se algumas constelações de sinais e realizando-se a partição de conjunto das mesmas e desta forma, obtendo parâmetros suficientes para a procura por codificadores TCM ótimos a ser realizada no Capítulo 3.

A seguir, será feito um estudo dos codificadores convolucionais sob o ponto de vista algébrico com o objetivo de classificação dos mesmos quando da procura a ser realizada.

CAPÍTULO 3

DETERMINAÇÃO E ANÁLISE DE CODIFICADORES TCM

ÓTIMOS PARA RETICULADOS EM 2 DIMENSÕES

I - INTRODUÇÃO

Uma vez definidos o reticulado e a constelação com o respectivo particionamento, basta escolher uma estrutura de treliça e um mapeamento apropriado dos sinais para ter-se um código TCM.

Neste trabalho, ao invés de apenas escolher-se uma dada estrutura de treliça, que apenas definiria um único codificador, será feita uma procura por codificadores convolucionais — não catastróficos, mínimos e que geram códigos TCM ótimos — em uma determinada classe simples à cada taxa de codificação. Para tanto neste capítulo iremos definir o que vem a ser estes codificadores.

II - ANÁLISE ALGÉBRICA

Nos capítulos anteriores foi feita uma análise do ponto de vista estrutural dos codificadores convolucionais, cabe aqui uma análise algébrica dos mesmos que venha auxiliar o entendimento e especificar os códigos encontrados.

Definição - II.1 [35]: Um codificador convolucional G_{kn} com k entradas e n saídas, sobre um corpo finito $\mathbb{F}(q)$, é uma máquina sequencial de estados finitos, linear e realizável, a qual estava no estado zero num passado longínquo.

Este trabalho restringe-se a codificadores e códigos convolucionais binários. As operações realizadas ocorrem no corpo de Galois de dois elementos $\mathbb{F}(2)$.

Seja uma sequência de informação de entrada representada por uma máquina autônoma de estados finitos linear e causal:

$$x = \{ \underline{x}^0, \underline{x}^1, \underline{x}^2, \dots \}$$

onde $x^i = [x_i^{(0)}, x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(k)}]$. Então,

$$x = \{ x_1^0 x_2^0 \dots x_k^0, x_1^1 x_2^1 \dots x_k^1, x_1^2 x_2^2 \dots x_k^2, \dots \}$$

onde $x_i^{(p)}$ representa a p -ésima posição de blocos das k - entradas, com $1 \leq p \leq k$ no instante de tempo i .

A transformada-D da sequência de informação é expressa por:

$$\underline{X}(D) = [X_1(D) \ X_2(D) \dots \ X_k(D)]$$

onde para cada $i = 1, 2, \dots, k$

$$X_i(D) = X_i^0 + X_i^1 D + X_i^2 D^2 + \dots$$

Como por definição os codificadores são realizáveis, a sequência de saída, ou sequência transmitida ou ainda palavra-código é da forma:

$$\underline{y} = \{ \underline{y}^0, \underline{y}^1, \underline{y}^2, \dots \}$$

onde $\underline{y}^j = [y_j^{(0)}, y_j^{(1)}, y_j^{(2)}, \dots, y_j^{(n)}]$. Então,

$$\underline{y} = \left\{ \begin{matrix} y_1^0 & y_2^0 & \dots & y_n^0, & y_1^1 & y_2^1 & \dots & y_n^1, & y_1^2 & y_2^2 & \dots & y_n^2, & \dots \end{matrix} \right\}$$

onde $y_j^{(q)}$ representa a q -ésima posição dos blocos das n saídas, com $1 \leq q \leq n$, no instante j .

A transformada-D da da sequência transmitida é expressa por

$$\underline{Y}(D) = [Y_1(D) \ Y_2(D) \ \dots \ Y_n(D)]$$

onde para cada $j = 1, 2, \dots, n$

$$Y_j(D) = Y_j^0 + Y_j^1 D + Y_j^2 D^2 + \dots$$

definindo assim o processo de codificação como [30,35]

$$\underline{Y}(D) = \underline{X}(D) G(D)_{kn} \quad (II.4)$$

onde $G(D)_{kn}$ é a matriz geradora do tipo linear e causal, representando por esta forma um codificador convolucional contendo polinômios na variável "D" correspondendo a um codificador de estrutura sem realimentação.

Neste caso a matriz $G(D)_{kn}$ é do tipo:

$$G(D)_{kn} = \begin{bmatrix} G_{11}(D) & G_{12}(D) & \dots & G_{1n}(D) \\ G_{21}(D) & G_{22}(D) & \dots & G_{2n}(D) \\ \vdots & \vdots & & \vdots \\ G_{k1}(D) & G_{k2}(D) & \dots & G_{kn}(D) \end{bmatrix}$$

$$e \quad G_{ij}(D) = g_{ij}^0 + g_{ij}^1 D + \dots + g_{ij}^{m_{ij}} D^{m_{ij}}$$

é o polinômio gerador da saída "j" relativa à entrada "i" sendo de grau m_{ij} para $i = 1 \dots k$ e $j = 1 \dots n$.

Uma outra maneira de se obter a palavra-código é através da convolução (portanto a nomenclatura para estes codificadores) da sequência x de entrada e as sequências geradoras g_j , onde

$$g_j = [g_j^{(0)}, g_j^{(1)}, \dots, g_j^{(n)}]$$

e $g_j^{(q)}$ representa a sequência geradora de saída q com $1 \leq j \leq m$.

A sequência geradora $g = [g_1, g_2, \dots, g_m]$ pode ser obtida da

máquina sequencial que define o codificador, uma vez que se alimenta um e somente um registro com o dígito "1" e os demais com "0". Isto consiste na aplicação de um impulso na entrada de um sistema linear para obtenção da função de transferência do mesmo. Portanto, matematicamente tem-se como resposta $Y_L^{(j)}$ do sistema a uma excitação x [21]

$$Y_L^{(j)} = \sum_{i=0}^m X_{L-i} \delta_i^{(j)} = X_L \delta_0^{(j)} + X_{L-1} \delta_1^{(j)} + \dots + X_{L-m} \delta_m^{(j)} \quad (\text{II.2})$$

para $j=1, \dots, n$ e $X_{L-i} = 0$ para todo $L < i$.

A palavra-código é então a sequência obtida através da multiplexação das sequências de saída.

Ordenando-se as sequências geradas, obtemos a matriz geradora G do código convolucional $\mathcal{C}(n, k, m)$

$$G = \begin{bmatrix} G_0 & G_1 & G_2 & \dots & G_m \\ & G_0 & G_1 & \dots & G_{m-1} & G_m \\ & & G_0 & \dots & G_{m-2} & G_{m-1} & G_m \\ & & & \ddots & & & \ddots \end{bmatrix}$$

onde G_L é uma submatriz $k \times n$ da forma

$$G_L = \begin{bmatrix} \delta_{1,L}^{(1)} & \delta_{1,L}^{(2)} & \dots & \delta_{1,L}^{(n)} \\ \delta_{2,L}^{(1)} & \delta_{2,L}^{(2)} & \dots & \delta_{2,L}^{(n)} \\ \vdots & \vdots & \ddots & \vdots \\ \delta_{k,L}^{(1)} & \delta_{k,L}^{(2)} & \dots & \delta_{k,L}^{(n)} \end{bmatrix}$$

As áreas em branco em G são preenchidas por "zeros", cada conjunto de k linhas de G é idêntico ao anterior mas deslocado n colunas para a direita. Dada uma sequência de entrada de comprimento finito ($L \times k$), G é uma matriz semi-infinita com $k(m + L)$ colunas. Desde que uma palavra-código é uma combinação linear das linhas da matriz geradora G , um código convolucional (n, k, m) é um código linear obtido por

$$y = x \cdot G \quad (II.2)$$

Seguem-se então as definições:

Def. - II.2: O código convolucional $\mathcal{C}(n, k, m)$, gerado por um codificador convolucional G_{kn} , é o conjunto de todas as sequências de saída y produzidas por todas as sequências de entrada x à uma taxa $R = \frac{k}{n}$, [95].

Def. - II.3: Dois codificadores são equivalentes se têm o mesmo conjunto de sequências de saída, sobre o conjunto de todas as sequências de entrada [95].

Def. - II.4: Um codificador G_{kn} é dito estar na forma sistemática se k dos n dígitos de saída são idênticos aos k dígitos de entrada. Este codificador sistemático é dito canônico quando pode ser associado a uma matriz $G(D)_{kn}$ da forma:

$$\begin{bmatrix} G_{11}(D) & G_{12}(D) & \dots & G_{1,n-k}(D) & 1 & 0 & \dots & 0 \\ G_{21}(D) & G_{22}(D) & \dots & G_{2,n-k}(D) & 0 & 1 & \dots & 0 \\ \vdots & \vdots & & \vdots & & \ddots & & \\ G_{k1}(D) & G_{k2}(D) & \dots & G_{k,n-k}(D) & 0 & 0 & \dots & 1 \end{bmatrix} = [G'(D) : I_k]$$

onde I_k é uma matriz identidade de ordem k e, para $j \leq n-k$, $G_{ij}(D)$ pode ser qualquer polinômio de grau m_{ij} , tal que cada saída $Y_j(D) \neq X_i(D)$ [30,35,37].

Todo codificador é equivalente a um codificador sistemático [28,30], o qual forma uma classe canônica, no sentido de ser realizado com o menor número possível de elementos de memória entre todos os codificadores equivalentes.

Todo codificador é equivalente a um codificador mínimo [28,30]. Um codificador mínimo tem sua representação esquemática (circuito seqüencial) sem realimentação e com o menor número de elementos de memória possível entre todos os seus codificadores equivalentes. Os codificadores mínimos levam às seqüências de erro decodificadas mais curtas por evento de erro.

Os codificadores mínimos serão a base dos melhores códigos encontrados neste trabalho, para cada classe de codificadores, segundo o comprimento total de memória.

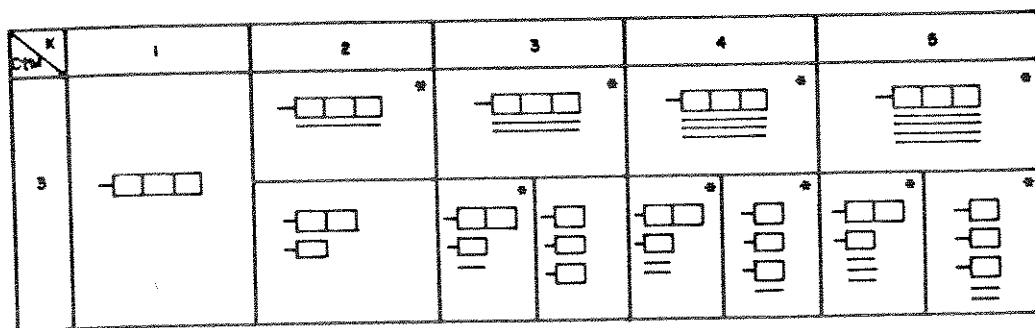
Def. - 11.5: Um código convolucional é dito catastrófico quando um número finito de erros introduzidos pelo meio pode resultar em um número infinito de erros de decodificação — o Algoritmo de procura utilizado [30] trata da eliminação destes códigos no momento da seleção dos melhores codificadores.

Os codificadores convolucionais podem ser separados em classes simples $\{G_{kn}: m_1, m_2, \dots, m_k\}$ de acordo com a estrutura de memória que apresentam. Se apresentam a mesma estrutura de memória e portanto a mesma treliça definem uma classe simples [30]. O conjunto total de classes $\{G_{kn}: M\}$ é definido pelo conjunto de todas as classes com comprimento total de memória

igual a
$$M = \sum_{i=1}^k m_i \quad (II.4)$$

Em outras palavras, para cada comprimento total de memória M , as classes simples são os codificadores formados por disposições diferentes dos registros de memória.

A figura abaixo exemplifica as classes simples para o Comprimento Total de Memória igual a 3



* Presença de transições parciais no diagrama de treliça
Fig.II.1 - Classes simples de codificadores

Conforme foi definido no Capítulo 1, podemos redefinir $d_{\min}$ para um código convolucional binário como

$$d_{\min} = \min_{y \neq y'} dH(y, y') \quad (II.5)$$

onde
$$dH(y, y') = \sum_{t=1}^{Ly} dH(\underline{y}^t, \underline{y}'^t)$$

é a distância de Hamming (número de bits não-coincidentes) entre duas palavras-código em $\mathcal{C}$, ou caminhos no diagrama de treliça (L_y é o comprimento de caminhos não-nulos y associado a cada byte $\underline{y}$ em y).

A distância mínima também pode ser definida em termos da distância Euclideana entre a sequência de sinais modulados

$$d_{\min} = \min_{y \neq y'} dE \left[a(y), a(y') \right] \quad (\text{II.6})$$

$$\text{onde } dE \left[a(y), a(y') \right] = \left\{ \sum_{t=1}^{L_y} dE^2 \left[a(\underline{y}^t), a(\underline{y}'^t) \right] \right\}^{\frac{1}{2}}$$

é a distância Euclideana entre as sequências $a(y)$ e $a(y')$ de sinais modulados; $dE \left[a(\underline{y}^t), a(\underline{y}'^t) \right]$ representa a distância Euclideana entre os símbolos de canal correspondente aos bytes $\underline{y}^t$ e $\underline{y}'^t$ reciprocamente para quaisquer duas sequências codificadas não-coincidentes e de mesmo comprimento $L_y = L_{y'}$.

No caso do sistema TCM, a distância mínima para os códigos obtidos é a menor distância Euclideana entre as transições paralelas e as transições adjacentes no diagrama de treliça.

$$d_{\min}^2 = \min \left[\Delta^2(\mu+1), \Delta_{\min}^2(\mu) \right] \quad (\text{II.7})$$

onde $\Delta^2(\mu+1)$ é a menor distância Euclideana no último nível da partição de conjunto (distância relativa às transições paralelas) e $\Delta_{\min}^2(\mu)$ denota a distância Euclideana mínima, relativa às

transições adjacentes no diagrama de treliça, e μ é o número de entradas codificadas, isto é, entradas que possuem registros de memória no circuito codificador associado, $1 \leq \mu \leq m$.

O peso Euclideano de um caminho y de comprimento L_y é definido como [5]

$$W_E(y, L_y) = \left\{ \sum_{t=1}^{L_y} W_E^2(\underline{y}^t) \right\}^{\frac{1}{2}} = \left\{ \sum_{t=1}^{L_y} \Delta_q^2(\underline{y}^t) \right\}^{\frac{1}{2}} \quad (\text{II.8})$$

onde $q(\underline{y}^t)$ é o número de zeros seguidos, a partir do bit de mais baixa ordem, no byte codificado não-nulo $\underline{y}^t$ do caminho y . $W_E(\underline{y}^t) = \Delta_q(\underline{y}^t)$, representa o peso Euclideano do byte codificado não-nulo $\underline{y}^t$ de y , o qual é dado pela distância mínima do subconjunto, cujo índice é igual a $q(\underline{y}^t)$. Por definição $W_E(\underline{0}) = 0$.

O peso Euclideano do byte codificado $\underline{y}^t$ é um limitante inferior da menor distância relacionada a um padrão de erro igual a $\underline{y}^t$. Ungerböeck [5] obteve um limitante inferior para a distância mínima Euclideana, em termos dos pesos Euclidianos W_E de tal forma que

$$d_{\min} \geq \Delta_{\min} = \min_{y \neq 0} W_E(y, L_y) \quad (\text{II.9})$$

Def. - II.8 Um codificador G_{kn} com comprimento total de memória igual a M , é um codificador ótimo, se o código a ele relacionado possui o máximo $\Delta_{\min}$ obtido entre os codificadores do conjunto total de classes $\{G_{kn}, M\}$ ao qual pertence [30].

Def. - II.7: Um código $\mathcal{C}$ é dito Linear [38] se existe uma operação chamada "soma" em relação à qual o alfabeto de saída, ou de canal $\langle a \rangle$ forma um grupo Abelianiano com relação a esta operação. O código implica um homomorfismo do alfabeto de entrada $\langle \underline{X} \rangle$ no alfabeto de canal $\langle a \rangle$; definindo a propriedade

$$a(\underline{X}_i + \underline{X}_j) = a(\underline{X}_i) + a(\underline{X}_j) \quad (\text{II.10})$$

Teorema - II.1: [38] Dados n , k , m , o conjunto de todos os códigos lineares de taxa $R = \frac{k}{n}$ e m memórias pode ser obtido através da operação em cascata de um codificador convolucional (k, n, m) , com um mapeamento biunívoco do alfabeto de saída $\langle \underline{Y} \rangle$ ao alfabeto de canal $\langle a \rangle$.

Conforme a definição II.7 e teorema II.1, se o codificador for um sistema linear, dando origem a um código linear, este código forma um grupo Abelianiano com respeito à operação soma no corpo de *Galois* com 2 elementos, e cuja propriedade de fechamento garante que o conjunto de distância de todas as seqüências do código a uma dada seqüência de referência (geralmente utiliza-se como referência a seqüência toda-nula) é sempre o mesmo, a despeito de qualquer referência.

Este fato permite o cálculo da distância mínima do código convolucional quando uma seqüência qualquer é fixada. Esta propriedade facilita o processo de decodificação, pois a comparação entre os pares de seqüências permite determinar a menor distância do conjunto de todas as distâncias obtidas, a partir desta seqüência fixada, com uma menor complexidade

computacional. De outra forma, seria necessário calcular-se o conjunto de distâncias entre cada sequência transmitida a todas as demais sequências possíveis na treliça, onerando em demasia a complexidade e o tempo de processamento do algoritmo de decodificação.

Para o sistema TCM, Marsan et al [38] definiram a classe de códigos denominada Superlinear. Estes códigos possuem a propriedade pela qual apenas uma sequência é suficiente como referência para cálculo de $d_{\min}$ do código TCM de maneira análoga ao processo utilizado nos códigos convolucionais.

Def. - II.8: [38] Um código $\mathcal{C}$ é dito Superlinear se:

(I) - é linear segundo a def. II.7

(II) - é possível associar à cada símbolo de canal $\langle a \rangle$ um número real positivo $W(a)$, denominado peso de "a" tal que:

$$dE(a_i, a_j) = W(a_i + a_j) \quad (\text{II.11})$$

Define-se como matriz adição do alfabeto de canal $\langle a \rangle$ aquela cuja posição (i, j) corresponde ao símbolo que representa a soma " $a_i + a_j$ " e, a matriz distância é definida como sendo aquela cuja posição (i, j) corresponde ao valor da distância entre os símbolos a_i e a_j .

Teo. - II.2: [38] Um código $\mathcal{C}$ é dito Superlinear se e somente se sinais de saída iguais na matriz de adição do alfabeto de modulação $\langle a \rangle$ correspondem a distâncias iguais nas posições respectivas da matriz distância, ou seja:

$$a_i + a_j = a_r + a_l \longrightarrow dE(a_i, a_j) = dE(a_r, a_l)$$

A partir da definição II.8 e do teorema II.2 torna-se possível fazer o cálculo de $d_{\min}$, ou analisar o desempenho dos códigos superlineares tomando-se como referência apenas uma sequência específica. Em outras palavras, reduz-se substancialmente o esforço computacional caso a procura por códigos ótimos seja realizada apenas para códigos superlineares.

Ao analisar-se, porém, as partições das constelações propostas no Capítulo 2, observa-se que nem todas possuem uma estrutura simétrica (Figs. $A_2.u-a$, $A_2.u-b$, $A_2.u-c$, $A_2.iii-a$), e como a métrica utilizada é a Euclideana, a não-superlinearidade ocorre devido ao fato de que os subconjuntos provenientes do ramo esquerdo da partição, são diferentes em termos de distância Euclideana dos provenientes do ramo direito, no mesmo nível da partição. Desta forma o código resultante é do tipo linear, porém a constelação onde este é mapeado não caracteriza uma partição que garanta as condições de superlinearidade, e portanto não satisfaz o Teorema II.2. As matrizes distância e adição possuem, por este fato, para sinais de saída iguais, distâncias diferentes em alguma posição (i, j) .

O algoritmo [90] utilizado supera este obstáculo utilizando no lugar das distâncias Euclidianas reais entre os sinais, o peso Euclideano W_E (Eq. II.8). Verifica-se então que a partir da sua utilização, os códigos TCM passam a satisfazer o Teorema II.2, ou seja, podem ser analisados como se fossem superlineares. A diferença é que no cômputo da distância mínima do código, será

obtido o valor estipulado pelo limitante inferior ($\Delta_{\min}$) em detrimento do valor real da distância mínima.

A superlinearidade é obtida, pelo fato de que, por definição, o peso Euclideano, $W_E(y, Ly)$ (Eq. II.8), limita a distância mínima intra-conjunto a cada nível da partição pela menor distância entre dois sinais daquele nível.

III - CONSIDERAÇÕES SOBRE A PROCURA EFETUADA

Como o codificador TCM possui um codificador convolucional em sua estrutura, este gera, então, um código de treliça, não apresentando o mesmo tratamento algébrico que ocorre com os códigos de bloco. Desta forma a determinação de códigos TCM ótimos costuma ser realizada através de procura sobre o universo destes códigos como foi feito em [30 e 37].

Por questões de linearidade e conseqüente simplificação computacional, para efeito de procura, o algoritmo proposto em [30] utiliza o critério de máximo $\Delta_{\min}$ (Eq. II.9), que é um limitante inferior da distância Euclideana mínima do código, quando não, igual a $d_{\min}$ (Eq. II.7).

A definição de código TCM ótimo está associada então a uma taxa de codificação $R = \frac{k}{n}$ e a um dado valor de comprimento total de memória cm , uma vez que estes fatores definem a complexidade de decodificação. Portanto, será definido como ótimo o código que possuir o máximo $\Delta_{\min}$ entre todos os códigos cujos codificadores pertençam ao conjunto total de classes. Conseqüentemente, o algoritmo utilizado realiza a procura para todas as classes simples de um conjunto total de classes para depois efetuar a comparação final dos melhores códigos de cada uma destas classes.

Como as transições paralelas não caracterizam uma codificação pois resultam dos bits não codificados, o algoritmo

[30] realiza a busca considerando somente os bits que serão codificados. Isto resulta em codificadores com taxas inferiores a aquelas originais. O valor $\Delta_{\min}(\mu)$ encontrado, onde μ é o número de entradas com registros de memória, deve ser comparado a $\Delta(\mu+1)$ (distância relativa às transições paralelas). O $d_{\min}$ para os códigos TCM será o menor destes dois valores.

Exemplo III.1 : Suponha a procura por codificadores ótimos a uma taxa $R = \frac{k}{n} = \frac{4}{5}$, 4 memórias, 4 transições paralelas ou seja 2 ligações diretas no circuito codificador mapeado em um conjunto de 32 sinais tipo A_2 — Constelação II.4-b, com o particionamento como mostrado na Fig. $A_2.4-b$ e cujos valores Δ_i são:

$$\Delta_0 = 0.475$$

$$\Delta_1 = 0.475$$

$$\Delta_2 = 0.950$$

$$\Delta_3 = 0.950$$

$$\Delta_4 = 1.644$$

A figura esquemática para o circuito codificador é do tipo

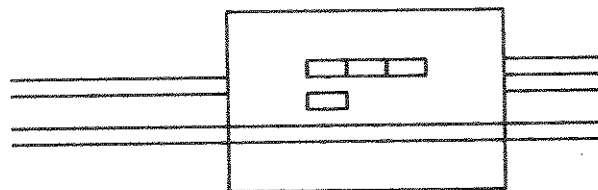


Fig. III.1: Circuito com 2 entradas codificadas, 2 entradas não codificadas, 4 memórias, e 5 saídas.

Como o codificador apresenta 2 entradas não codificadas e 2 entradas codificadas ($\mu=2$), temos que a correspondente treliça apresentará 4 transições paralelas entre estados e a procura é realizada então à taxa de $2/3$, ao invés de $4/5$, dentre os codificadores desta classe simples. Como resultado, tem-se os melhores codificadores desta classe para o respectivo valor $\Delta_{\min}(\mu)$. A notação adotada para representação destes codificadores é feita através de vetores coeficientes.

III.1 - REPRESENTAÇÃO POR VETORES COEFICIENTES [20]

Neste trabalho, a procura por códigos ótimos efetuada, restringe-se à estrutura de memória da forma: $m_1 \geq m_2 \geq m_3 \geq m_4 \geq m_5 \geq 0$. Desta forma, são eliminadas representações redundantes de um mesmo código, no sentido de suas matrizes geradoras serem iguais a menos de permutações de linhas. As taxas de codificação estabelecidas variam de $\frac{2}{3}$ a $\frac{4}{5}$. O número de transições paralelas variam a cada taxa de codificação chegando a 8 transições paralelas para a taxa $\frac{4}{5}$.

Os codificadores encontrados G_{kn} que foram representados em II.1 por uma matriz geradora polinomial, também podem ser representados de maneira única por um conjunto de n vetores, $\{ \underline{vc}(j), j = 1, 2, \dots, n \}$, que caracterizam uma classe simples $\{ G_{kn}: m_1, m_2, \dots, m_k \}$ de codificadores sem realimentação, definidos através de sua estrutura de memória.

Portanto, cada vetor $\underline{VC}(j)$, com $j=1, 2, \dots, n$ corresponde a um arranjo especial do vetor de todos os $k+m$ coeficientes da matriz geradora, relativos à saída " j " ou seja, os coeficientes dos polinômios da coluna " j " em $G(D)_{kn}$.

$\underline{VC}(j)$ é definido como

$$\underline{VC}(j) = \left[\begin{array}{c} \left(\begin{array}{c} m_1 \\ \sigma_{1j} \end{array} \right) \left(\begin{array}{c} m_1^{-1} \\ \sigma_{1j} \end{array} \right) \dots \left(\begin{array}{c} m_2^{+1} \\ \sigma_{1j} \end{array} \right) \left(\begin{array}{cc} m_2 & m_2 \\ \sigma_{1j} & \sigma_{2j} \end{array} \right) \\ \left(\begin{array}{cc} m_2^{-1} & m_2^{-1} \\ \sigma_{1j} & \sigma_{2j} \end{array} \right) \dots \left(\begin{array}{ccc} m_{k+1} & m_{k+1} & m_{k+1} \\ \sigma_{1j} & \sigma_{2j} & \dots \sigma_{k-1,j} \end{array} \right) \\ \left(\begin{array}{ccc} m_k & m_k & m_k \\ \sigma_{1j} & \sigma_{2j} & \dots \sigma_{kj} \end{array} \right) \left(\begin{array}{ccc} m_k^{-1} & m_k^{-1} & m_k^{-1} \\ \sigma_{1j} & \sigma_{2j} & \dots \sigma_{kj} \end{array} \right) \dots \\ \left(\begin{array}{ccc} m_1 & m_1 & m_1 \\ \sigma_{1j} & \sigma_{2j} & \dots \sigma_{kj} \end{array} \right) \left(\begin{array}{ccc} m_0 & m_0 & m_0 \\ \sigma_{1j} & \sigma_{2j} & \dots \sigma_{kj} \end{array} \right) \end{array} \right] \quad (\text{III.1.1})$$

onde σ_{ij}^p é igual a "1" (ou "0") no caso de haver (ou não) conexão entre a saída "j" ($j=1,2,\dots,n$) do codificador e a entrada "i" ($p=0$), ou um estágio ($p=1,2,\dots,m_i$) do registrador de deslocamento da entrada "i" ($i=1,2,\dots,k$).

Daremos um exemplo de como pode ser obtido o circuito de um codificador convolucional que esteja representado por um conjunto de vetores coeficientes.

Suponha a procura efetuada no exemplo III.1:

O valor encontrado para $\Delta_{\min}(\mu)$ é 0.950

A distância relativa às transições paralelas é $\Delta(\mu+1)=0.950$

$\Delta_{\min}(\mu)=\Delta(\mu+1)$, portanto o $d_{\min}$ é 0.950 e os melhores codificadores para esta classe simples com este valor são:

{VC(1),VC(2),VC(3)}

{24,14,33}

{24,6,35}

{24,14,37}

{24,6,41}

Cada um destes codificadores possui o maior valor de $d_{\min}$ dentro da classe em que foi feita a procura e tem como características ser superlinear, não-catastrófico e mínimo. Sua estrutura pode ser obtida da seguinte maneira.

Tomemos por exemplo o codificador {24,14,33}, sabemos que sua figura esquemática é da forma da Fig.III.1. As ligações entre os elementos de memória e os somadores módulo-2 obedecem o procedimento a seguir.

Os codificadores são representados por seqüências de n números inteiros $\{vc(1), \dots, vc(n)\}$ correspondentes às n saídas do codificador. Cada número inteiro associado a $vc(j)$ corresponde à maneira como estão ligadas as entradas e memórias com as saídas. Os números inteiros são convertidos em binários e indicam com o bit "1" o ponto de ligação com a saída correspondente e com o bit "0" a não ligação com esta saída. Para mais de dois pontos ligados a uma mesma saída, esta passa a ser a soma módulo-2 dos

Capítulo 3

sinais provenientes das ligações. A ordenação das entradas e memórias segue o seguinte critério — de cima para baixo, da direita para esquerda em ordem crescente.

Verifica-se que, para codificadores de uma mesma classe simples $\{ G_{kn:m_1,m_2,\dots,m_k} \}$, a sua representação por vetores coeficientes possuirá o mesmo número de vetores (n) e as mesmas posições por vetor (em número de $k+m$). Os codificadores diferirão entre si pela designação de "0" ou "1", para estas posições [30].

A síntese destes procedimentos originam a tabela e circuito seguintes:

	1	2	3	4	5	6		$\leftrightarrow$	δ_{1j}^2	δ_{2j}^2	δ_{1j}^1	δ_{2j}^1	δ_{1j}^0	δ_{2j}^0
VC(1)	0	1	1	0	0	0	24							
VC(2)	0	0	1	1	1	0	14							
VC(3)	1	0	0	0	0	1	33							

TABELA III. 1. 1

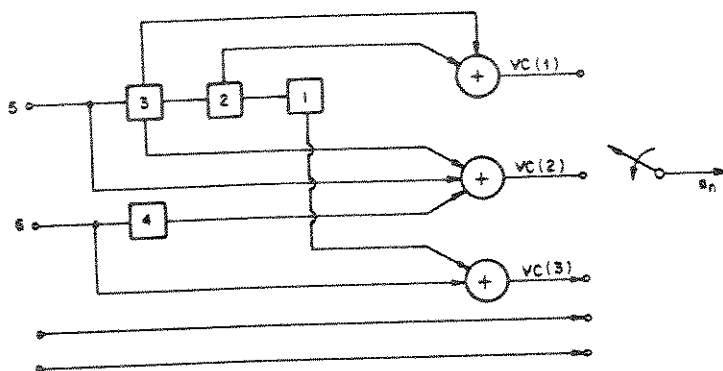


FIG. 1. 1

A figura III.1.1 apresenta o circuito esquemático para o codificador convolucional $\{ G_{45:3,1,0,0} \}$ — codificador com 4

entradas, 5 saídas, 3 memórias ligadas à primeira entrada, 1 memória ligada à segunda entrada e nenhuma memória ligada à terceira e quarta entrada, portanto 2 ligações diretas — que é representado pelo conjunto $\{24, 14, 33\}$ de vetores coeficientes que caracteriza as ligações deste circuito.

Os melhores códigos encontrados a seguir para os reticulados A_2 e Z^2 foram obtidos mediante a utilização deste Algoritmo, segundo as suas duas versões: procura Exaustiva e procura Não-Exaustiva.

Na procura Exaustiva de códigos ótimos, o Algoritmo trata de analisar todos os codificadores de uma classe simples de um conjunto total de classes. Para tornar a procura eficiente, são eliminadas realizações equivalentes de um mesmo codificador, ou seja, aquelas que geram códigos com as mesmas propriedades de distância [30, cap. 5]. Este Algoritmo mostrou-se seguro para obtenção dos melhores códigos na procura efetuada para classes simples onde o valor de k, n e m são pequenos, porém com um tempo de processamento relativamente alto.

Com o objetivo de acelerar o tempo de procura e tornar prática a procura por códigos em classes simples de maior complexidade, isto é, valores de k, n e m impraticáveis para a procura Exaustiva, o Algoritmo de procura Não-Exaustiva trata de expurgar classes de códigos indesejáveis e faz a procura sobre subconjuntos especiais de codificadores [30, 39] de um conjunto total de classes, onde são encontrados um pequeno número de codificadores e com alta probabilidade de possuírem bons códigos.

Para termos uma noção da importância e eficiência destes Algoritmos de procura de códigos, mostraremos com um exemplo a

complexidade existente em cada cálculo de $d_{\min}$.

Para a procura dos códigos ótimos dentro da classe simples $\left\{ G_{45}: 3,1,0,0 \right\}$ mapeados em um conjunto de 32 sinais tipo A_2 — constelação II.44-b — utilizando-se o Algoritmo de procura Exaustiva no sistema VAX-UNICAMP com o tempo de processamento iterativo de 3:15 hs, foi obtido o valor de $\Delta_{\min}(\mu) = 0.950$ para esta classe simples, relativo a 16 janelas de tempo no diagrama de treliça. Obtendo desta forma o conjunto de 4 codificadores ótimos relacionados anteriormente.

De outro modo, utilizando-se o método heurístico de Ungerböeck da construção de códigos TCM, para a criação de um código pertencente à mesma classe simples, com o mesmo conjunto de sinais para o mapeamento, obtemos o diagrama de treliça Fig. III.1.2 com 16 estados associado à constelação II.44-b (vide Cap 2). Onde de cada nó emergem 4 transições de estado e cada um destes ramos da treliça representa 4 transições paralelas, alcançando o nível D_1 da partição de conjunto $A_2.32-b$ apresentada no Cap. 2. As regras de codificação estabelecem o código e por sua vez o codificador.

Para realizar o cálculo de $d_{\min}$ do código estabelecido é necessário comparar o valor de $\Delta(\mu+1)$ das transições paralelas e o valor $\Delta_{\min}(\mu)$ relativo às transições adjacentes.

O valor da distância Euclideana referente às transições paralelas é facilmente observado na partição de conjunto de 32 sinais — $A_2.32-b \rightarrow \Delta(\mu+1) = 0.950$.

O valor de $\Delta_{\min}(\mu)$ relativo às transições adjacentes é obtido da seguinte forma:

(d) - Supondo-se o código não-superlinear: deve ser obtido o menor valor do cálculo entre todas as distâncias Euclidianas dos conjuntos de sinais que emergem 2 a 2 em um nó e se encontram em algum nó adiante;

(u) - Supondo o código superlinear: estabelece-se uma sequência padrão e computa-se o menor valor obtido ao se somar, a cada intervalo de tempo, a distância Euclidiana entre os sinais da sequência padrão e o conjunto de sinais que emerge por um nó desta sequência padrão e retorna depois de pelo menos 2 intervalos de tempo.

Se o código for considerado não-superlinear este cálculo torna-se impraticável.

Supondo-se o código superlinear, o valor mais provável encontrado é $\Delta_{\min}(\mu) = 0.820$, valor relativo à distância Euclidiana entre a transição adjacente $\{D_3 - D_0 - D_3\}$ e a sequência padrão $\{D_5 - D_5 - D_5\}$.

$$dE^2(\text{Transições Paralelas}) = (2\Delta_0^2) = 4\Delta_0^2$$

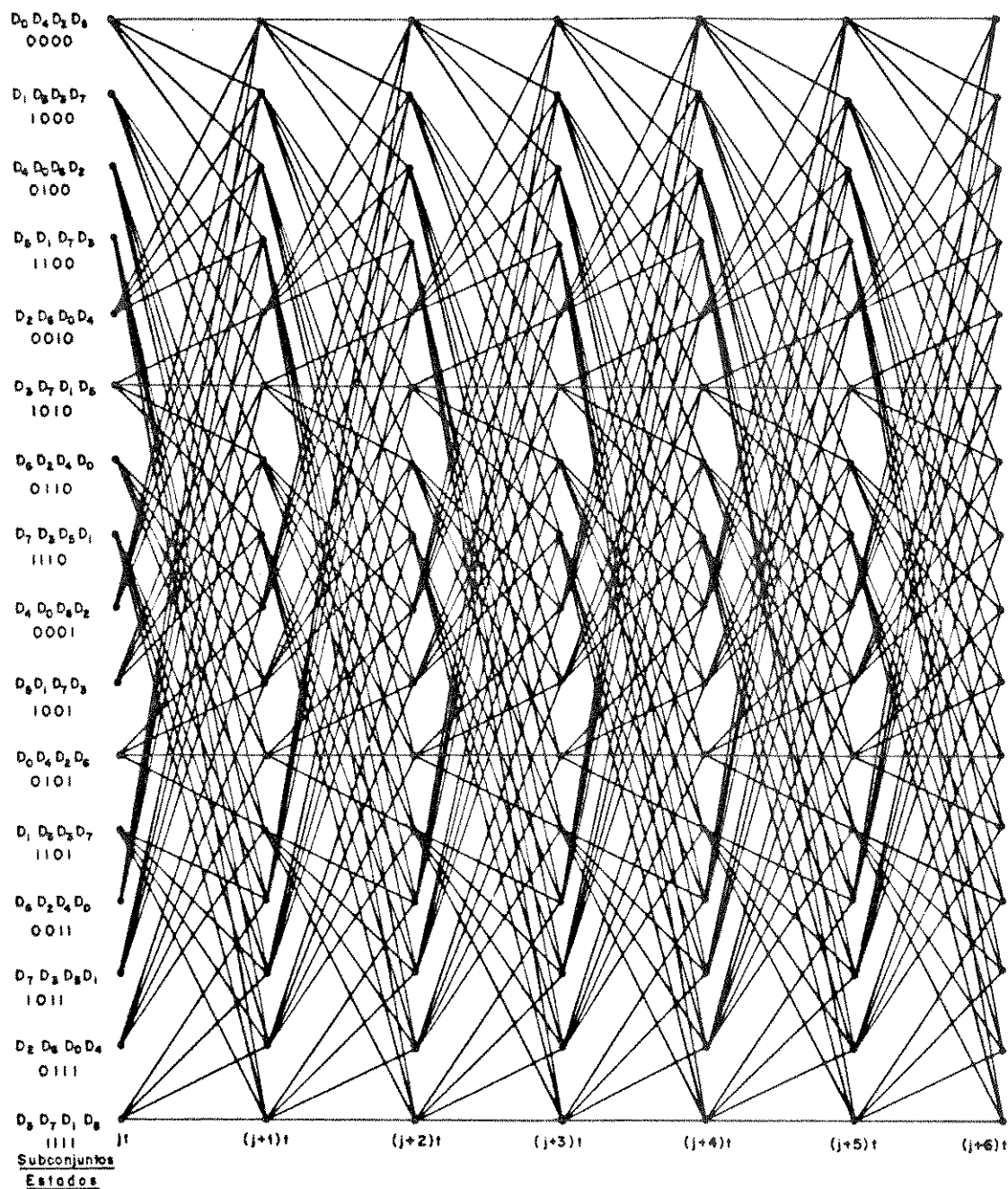
$$\begin{aligned} dE^2(\text{Transições Adjacentes}) &= dE^2(D_3, D_5) + dE^2(D_0, D_5) + dE^2(D_3, D_5) \\ &= \Delta_0^2 + \Delta_0^2 + \Delta_0^2 \end{aligned}$$

$$\text{Portanto, } d_{\min} = \sqrt{3} \Delta_0^2$$

TAXA 4/5 - DESEMPENHO EM A_2

$\{6, 4, 3, 1, 0, 0\}$ - Codificador com 4 memórias, trelço de 16 estados associado à constelação Π_{16}

Cada ramo do trelço representa 4 transições paralelas.



CÁLCULO DE $d_{\min}$

$$d^2 E \text{ (TRANSIÇÕES PARALELAS)} = (2 A_0)^2 = 4 A_0^2$$

$$d^2 E \text{ (TRANSIÇÕES ADJACENTES)} = d^2 E(D_0, D_1) + d^2 E(D_0, D_3) + d^2 E(D_1, D_3) = 3 A_0^2$$

$$\text{Portanto } d_{\min} = \sqrt{3} A_0$$

FIGURA III.1.2

Para a garantia de que este é o valor de $d_{\min}$ para o código estabelecido, em apenas 6 janelas temporais no diagrama de treliça (Fig. III.1.2), seria preciso mostrar que não existe qualquer outra sequência na treliça que comparada à sequência padrão $\{D_5 - D_5 - D_5 - D_5 - D_5 - D_5\}$ resulte em um valor menor que o encontrado, o que só é possível de se fazer por inspeção do diagrama de treliça, salvo ainda o fato do código não ser catastrófico.

Desta maneira deve ser feito o cômputo de todas as distâncias seguindo o procedimento do algoritmo de Viterbi, o que resultaria em um cálculo da ordem de $3 \times \left(\frac{1 - 4^6}{1 - 4} \right) \cong 4 \times 10^3$ operações. Porém a simples obtenção do valor $d_{\min} = 0.820 < 0.950$ (onde 0.950 é o valor obtido para os melhores códigos TCM desta classe através de procura Exaustiva) implica que este código, estabelecido heurísticamente pelas regras de codificação, não figura entre os 4 valores ótimos obtidos através do algoritmo de procura.

Portanto, o Algoritmo realiza de forma segura um número razoavelmente grande de operações para cada um dos códigos existentes dentro de cada classe simples, procurando o maior valor existente de $d_{\min}$ na classe e ainda relaciona todos os codificadores existentes com este referido valor, justificando desta forma o tempo computacional excessivo de processamento.

Apresentamos a seguir o resultado da procura efetuada dos melhores códigos TCM para as taxas $\frac{2}{3}$, $\frac{3}{4}$ e $\frac{4}{5}$ mapeados em constelações de 8, 16 e 32 sinais respectivamente oriundos dos reticulados A_2 e Z^2 .

RETICULADO Z^2 - TAXA $\frac{2}{3}$

Partição de conjunto: [Fig. 2.1-a]

$$\Delta_0 = 0.816$$

$$\Delta_1 = 1.154$$

$$\Delta_2 = 1.632$$

$$8 \left\{ |a_n^2| \right\} = \frac{3}{2} \Delta_0^2 = 1$$

CTM=2	CLASSE SIMPLES	$\langle G_{23} : 2, 0 \rangle$
	$\Delta_{\min}(\mu)$	1.825
	$\Delta(\mu+1)$	1.632
	$d_{\min} \text{ TCM}$	1.632
	GCA	1.244
	Melhores códigos [VC(1), ..., VC(n)]	[2, 5]
CTM=3	CLASSE SIMPLES	$\langle G_{23} : 3, 0 \rangle$
	$\Delta_{\min}(\mu)$	1.999
	$\Delta(\mu+1)$	1.632
	$d_{\min} \text{ TCM}$	1.632
	GCA	1.244
	Melhores códigos [VC(1), ..., VC(n)]	[10, 7] [12, 7] [4, 11] [5, 11] *1
CTM=4	CLASSE SIMPLES	$\langle G_{23} : 4, 0 \rangle$
	$\Delta_{\min}(\mu)$	2.161
	$\Delta(\mu+1)$	1.632
	$d_{\min} \text{ TCM}$	1.632
	GCA	1.244
	Melhores códigos [VC(1), ..., VC(n)]	[4, 19] [8, 19] [14, 19] [14, 23]

TABELA III.1.2:

Resultado da procura por melhores códigos TCM para a constelação de 8 sinais [1.1-a] com GCA obtido em relação à constelação 4-PSK não-codificada.
* → Apêndice 1.

RETICULADO Z^2 - TAXA $\frac{2}{3}$

Partição de conjunto: [Fig. Z^2 .i-b)

$$\Delta_0 = 0.692$$

$$\Delta_1 = 0.894$$

$$\Delta_2 = 1.788$$

$$\sum \{ |a_n^2| \} = \frac{5}{2} \Delta_0^2 = 1$$

CTM=2	CLASSE SIMPLES	$\langle G_{29} : 2, 0 \rangle$
	$\Delta_{\min}(\mu)$	1.413
	$\Delta(\mu+1)$	1.788
	$d_{\min} \text{ TCM}$	1.413
	GCA	-0.007
	Melhores Códigos [VC(1), ..., VC(N)]	[2, 5]
CTM=3	CLASSE SIMPLES	$\langle G_{29} : 3, 0 \rangle$
	$\Delta_{\min}(\mu)$	1.549
	$\Delta(\mu+1)$	1.788
	$d_{\min} \text{ TCM}$	1.549
	GCA	0.785
	Melhores Códigos [VC(1), ..., VC(N)]	[10, 7] [12, 7] [4, 11] [5, 11] * ₂
CTM=4	CLASSE SIMPLES	$\langle G_{29} : 4, 0 \rangle$
	$\Delta_{\min}(\mu)$	1.672
	$\Delta(\mu+1)$	1.788
	$d_{\min} \text{ TCM}$	1.672
	GCA	1.454
	Melhores Códigos [VC(1), ..., VC(N)]	[4, 19] [9, 19] [14, 19] [14, 29]

TABELA III.1.3:

Resultado da procura por melhores códigos TCM para a constelação de 8 sinais [1.1-b] com GCA obtido em relação à constelação 4-PSK não-codificada * → Apêndice 1

RETICULADO Z^2 - TAXA $\frac{2}{3}$

Partição de conjunto: [Fig. Z^2 .i-b']

$$\Delta_0 = 0.632$$

$$\Delta_1 = 0.894$$

$$\Delta_2 = 1.413$$

$$\sum \left\{ |a_n^2| \right\} = \frac{5}{2} \Delta_0^2 = 1$$

CTM=2	CLASSE SIMPLES	$\langle G_{23}:2,0 \rangle$
	$\Delta_{\min}(\mu)$	1.413
	$\Delta(\mu+1)$	1.413
	$d_{\min}$ TCM	1.413
	GCA	-0.007
	Melhores códigos [VC(1),...,VC(N)]	[2,5]
CTM=3	CLASSE SIMPLES	$\langle G_{23}:3,0 \rangle$
	$\Delta_{\min}(\mu)$	1.548
	$\Delta(\mu+1)$	1.413
	$d_{\min}$ TCM	1.413
	GCA	-0.007
	Melhores códigos [VC(1),...,VC(N)]	[10,7] [12,7] [4,11] [5,11] ₂
CTM=4	CLASSE SIMPLES	$\langle G_{23}:4,0 \rangle$
	$\Delta_{\min}(\mu)$	1.672
	$\Delta(\mu+1)$	1.413
	$d_{\min}$ TCM	1.413
	GCA	-0.007
	Melhores códigos [VC(1),...,VC(N)]	[4,19] [8,19] [14,19] [14,23]

* Apêndice 1

TABELA III.1.4:

Resultado da procura por melhores códigos TCM para a constelação de 8 sinais [1.i-b] e partição alternativa Z^2 .i-b', com GCA obtido em relação ao 4-PSK não-codificado.

RETICULADO A_2 - TAXA $\frac{2}{3}$

Partição de conjunto: [fig. A_2 , i-a]

$$\Delta_0 = 0.943$$

$$\Delta_1 = 0.943$$

$$\Delta_2 = 1.033$$

$$\sum \left\{ |a_n^2| \right\} = \frac{9}{8} \Delta_0^2 = 1$$

CTM=2	CLASSE SIMPLES	$\langle G_{23}; 2, \alpha \rangle$
	$\Delta_{\min}(\mu)$	1.033
	$\Delta(\mu+1)$	1.033
	$d_{\min}$ TCM	1.033
	GCA	1.249
	Melhores Códigos [VC(1), ..., VC(N)]	(6,1) (7,1) (5,2) (7,2) *4
CTM=3	CLASSE SIMPLES	$\langle G_{23}; 3, \alpha \rangle$
	$\Delta_{\min}(\mu)$	1.996
	$\Delta(\mu+1)$	1.033
	$d_{\min}$ TCM	1.033
	GCA	1.249
	Melhores Códigos [VC(1), ..., VC(N)]	(12,2) (13,2) (13,3) (14,3) *5
CTM=4	CLASSE SIMPLES	$\langle G_{23}; 4, \alpha \rangle$
	$\Delta_{\min}(\mu)$	2.109
	$\Delta(\mu+1)$	1.033
	$d_{\min}$ TCM	1.033
	GCA	1.249
	Melhores Códigos [VC(1), ..., VC(N)]	(25,5) (21,5) (22,9) (21,9) *6

TABELA III.1.5:

Resultado da procura por melhores códigos TCM para a constelação de 8 sinais [III.1-a] com GCA obtido em relação ao 4-PSK não-codificado.
* → Apêndice 1

RETICULADO A_2 - TAXA $\frac{2}{3}$

Partição de conjunto: [Fig. A_2 .i-b]

$$\Delta_0 = 0.961$$

$$\Delta_1 = 0.961$$

$$\Delta_2 = 1.004$$

$$\sum \left\{ |a_n^2| \right\} = \frac{13}{12} \Delta_0^2 = 1$$

CTM=2	CLASSE SIMPLES	$\langle G_{29}^{12,0} \rangle$
	$\Delta_{\min}(\mu)$	1.005
	$\Delta(\mu+1)$	1.004
	$d_{\min}$ TCM	1.004
	GCA	1.412
	Melhores Códigos [VC(1), ..., VC(N)]	[6,1] [7,1] [5,2] [7,2] *7
CTM=3	CLASSE SIMPLES	$\langle G_{29}^{13,0} \rangle$
	$\Delta_{\min}(\mu)$	1.922
	$\Delta(\mu+1)$	1.004
	$d_{\min}$ TCM	1.004
	GCA	1.412
	Melhores Códigos [VC(1), ..., VC(N)]	[13,2] [15,2] [13,3] [14,3] *8
CTM=4	CLASSE SIMPLES	$\langle G_{29}^{14,0} \rangle$
	$\Delta_{\min}(\mu)$	2.149
	$\Delta(\mu+1)$	1.004
	$d_{\min}$ TCM	1.004
	GCA	1.412
	Melhores Códigos [VC(1), ..., VC(N)]	[20,5] [21,5] [22,9] [21,9] *9

TABELA III.1.6:

Resultado da procura por melhores códigos TCM para a constelação de 8 sinais [11.i-b] com GCA obtido em relação ao 4-PSK não-codificado.
* → Apêndice 1

RETICULADO Z^2 - TAXA $\frac{3}{4}$

Partição de conjunto: [Fig. Z^2 .ii]

$$\Delta_0 = 0.692$$

$$\Delta_1 = 0.894$$

$$\Delta_2 = 1.265$$

$$\Delta_3 = 1.789$$

$$\sum \left\{ |a_n^2| \right\} = \frac{40}{16} \Delta_0^2 = 1$$

* → Apêndice 1

TABELA III.1.7:

Resultado da procura por melhores códigos TCM para a constelação de 16 sinais [1.ii] com GCA obtido em relação ao 8-PSK não-codificado.

CTM=2	CLASSE SIMPLES	$\langle G_{34}; 2, 0, 0 \rangle$	
	$\Delta_{\min}(\mu)$	1.413	
	$\Delta(\mu+1)$	1.265	
	$d_{\min}$ TCM	1.265	
	GCA	4.368	
	Melhores Códigos [VC(1), ..., VC(N)]	[2, 5]	
CTM=3	CLASSE SIMPLES	$\langle G_{34}; 2, 0, 0 \rangle$	$\langle G_{34}; 2, 1, 0 \rangle$
	$\Delta_{\min}(\mu)$	1.548	1.413
	$\Delta(\mu+1)$	1.265	1.789
	$d_{\min}$ TCM	1.265	1.413
	GCA	4.368	5.329
	Melhores Códigos [VC(1), ..., VC(N)]	[10, 7] [12, 7] [4, 11] [5, 11] * ₁₁	[8, 19, 5] [8, 17, 6] * ₁₁
CTM=4	CLASSE SIMPLES	$\langle G_{34}; 4, 0, 0 \rangle$	$\langle G_{34}; 3, 1, 0 \rangle$
	$\Delta_{\min}(\mu)$	1.672	1.548
	$\Delta(\mu+1)$	1.265	1.789
	$d_{\min}$ TCM	1.265	1.548
	GCA	4.368	6.122
	Melhores Códigos [VC(1), ..., VC(N)]	[4, 19] [8, 19] [14, 19] [14, 23]	[24, 25, 6] * ₁₂ [24, 41, 6]

RETICULADO A_2 - TAXA $\frac{9}{4}$

Partição de conjunto: [Fig. A_2 .44-a]

$$\Delta_0 = 0.666$$

$$\Delta_1 = 0.666$$

$$\Delta_2 = 1.333$$

$$\Delta_3 = 1.333$$

$$\sum \left\{ |a^2| \right\} = \frac{36}{4} \Delta_0^2 = 1$$

* → Apêndice 1

CTM=2	CLASSE SIMPLES	$\langle G_{00}: 2, 0, 0 \rangle$
	$\Delta_{\min}(\mu)$	1.154
	$\Delta(\mu+1)$	1.333
	$d_{\min}$ TCM	1.154
	GCA	3.570
	Melhores códigos [VC(1), ..., VC(N)]	[6, 1] [7, 1] [5, 2] [7, 2] *13

TABELA III.1.8:

Resultado da procura por melhores códigos TCM para constelação de 16 sinais [II.ii-a] com GCA obtido em relação ao 8-PSK não-codificado.

CTM=3	CLASSE SIMPLES	$\langle G_{34}: 3, 0, 0 \rangle$	$\langle G_{34}: 2, 1, 0 \rangle$
	$\Delta_{\min}(\mu)$	1.333	1.154
	$\Delta(\mu+1)$	1.333	1.333
	$d_{\min}$ TCM	1.333	1.154
	GCA	4.823	3.570
	Melhores códigos [VC(1), ..., VC(N)]	[13, 2] [15, 2] [13, 3] *14	[24, 6, 1] [30, 6, 1] *15
CTM=4	CLASSE SIMPLES	$\langle G_{34}: 4, 0, 0 \rangle$	$\langle G_{34}: 3, 1, 0 \rangle$
	$\Delta_{\min}(\mu)$	1.489	1.154
	$\Delta(\mu+1)$	1.333	1.333
	$d_{\min}$ TCM	1.333	1.154
	GCA	4.823	3.570
	Melhores códigos [VC(1), ..., VC(N)]	[25, 5] [31, 5] [22, 9] *16	[16, 14, 33] [16, 14, 35] *17

RETICULADO A_2 - TAXA $\frac{3}{4}$

Partição de conjunto: [fig. $A_2.11-b$]

$$\Delta_0 = 0.676$$

$$\Delta_1 = 0.676$$

$$\Delta_2 = 1.352$$

$$\Delta_3 = 1.352$$

$$\sum \left\{ |a_n^2| \right\} = \frac{95}{16} \Delta_0^2 = 1$$

* → Apêndice 1

TABELA III.1.9:

Resultado da procura por melhores códigos TCM para constelação de 16 sinais [11.11-b] com GCA obtido em relação ao 8-PSK não-codificado.

CTM=2	CLASSE SIMPLES	$\langle G_{34}:2,0,0 \rangle$	
	$\Delta_{\min}(\mu)$	1.171	
	$\Delta(\mu+1)$	1.352	
	$d_{\min}$ TCM	1.171	
	GCA	3.697	
	Melhores Códigos [VC(1), ..., VC(N)]	[6,1] [7,1] [5,2] *18	
CTM=3	CLASSE SIMPLES	$\langle G_{34}:3,0,0 \rangle$	$\langle G_{34}:2,1,0 \rangle$
	$\Delta_{\min}(\mu)$	1.352	1.171
	$\Delta(\mu+1)$	1.352	1.352
	$d_{\min}$ TCM	1.352	1.171
	GCA	4.946	3.697
	Melhores Códigos [VC(1), ..., VC(N)]	[13,2] [15,2] [13,3] *19	[24,6,1] [30,6,1] *20
CTM=4	CLASSE SIMPLES	$\langle G_{34}:4,0,0 \rangle$	$\langle G_{34}:3,1,0 \rangle$
	$\Delta_{\min}(\mu)$	1.512	1.171
	$\Delta(\mu+1)$	1.352	1.352
	$d_{\min}$ TCM	1.352	1.171
	GCA	4.946	3.697
	Melhores Códigos [VC(1), ..., VC(N)]	[25,5] [31,5] [22,9] *21	[16,14,33] [16,14,35] *22

RETICULADO A_2 - TAXA $\frac{3}{4}$

Partição de conjunto: [Fig. $A_2.11-c$]

$$\Delta_0 = 0.666$$

$$\Delta_1 = 0.666$$

$$\Delta_2 = 1.333$$

$$\Delta_3 = 1.333$$

$$\sum \left\{ |a_n^2| \right\} = \frac{36}{16} \quad \Delta_0^2 = 1$$

* → Apêndice 1

TABELA III.1.10:

Resultado da procura por melhores códigos TCM para constelação de 16 sinais [11.11-c] com GCA obtido em relação ao 8-PSK não-codificado.

CTM=2	CLASSE SIMPLES	$\langle G_{00}: 2, 0, 0 \rangle$	
	$\Delta_{\min}(\mu)$	1.154	
	$\Delta(\mu+1)$	1.333	
	$d_{\min}$ TCM	1.154	
	GCA	3.570	
	Melhores códigos [VC(1), ..., VC(N)]	[6, 1] [7, 1] [5, 2] *23	
CTM=3	CLASSE SIMPLES	$\langle G_{34}: 3, 0, 0 \rangle$	$\langle G_{34}: 2, 1, 0 \rangle$
	$\Delta_{\min}(\mu)$	1.333	1.154
	$\Delta(\mu+1)$	1.333	1.333
	$d_{\min}$ TCM	1.333	1.154
	GCA	4.823	3.570
	Melhores códigos [VC(1), ..., VC(N)]	[13, 2] [15, 2] [13, 3] *24	[24, 6, 1] [30, 6, 1] *25
CTM=4	CLASSE SIMPLES	$\langle G_{34}: 4, 0, 0 \rangle$	$\langle G_{34}: 3, 1, 0 \rangle$
	$\Delta_{\min}(\mu)$	1.489	1.154
	$\Delta(\mu+1)$	1.333	1.333
	$d_{\min}$ TCM	1.333	1.154
	GCA	4.823	3.570
	Melhores códigos [VC(1), ..., VC(N)]	[25, 5] [31, 5] [22, 9] *26	[16, 14, 33] [16, 14, 35] *27

RETICULADO Z^2 - TAXA $\frac{4}{5}$ Partição de conjunto: [Fig. Z^2 .iii]

$$\Delta_0 = 0.442$$

$$\Delta_1 = 0.625$$

(※ → Apêndice 1)

$$\Delta_2 = 0.884$$

$$\Delta_3 = 1.250$$

$$\Delta_4 = 1.250$$

$$\frac{1}{2} \left\{ |a_n^2| \right\} = \frac{44}{8} \Delta_0^2 = 1$$

TABELA III.1.11:

Resultado da procura por melhores códigos TCM para constelação de 32 sinais [Fig. I.iii] com GCA obtido em relação ao 16-PSK não-codificado.

CTM=3	CLASSE SIMPLES	$\langle G_{45}; 3, 0, 0, 0 \rangle$	$\langle G_{45}; 2, 1, 0, 0 \rangle$	
	$\Delta_{\min}(\mu)$	1.089	0.988	
	$\Delta(\mu+1)$	0.884	1.250	
	$d_{\min}$ TCM	0.884	0.988	
	GCA	7.107	8.079	
	Melhores Códigos [VC(1), ..., VC(N)]	[10, 7] [12, 7] [4, 11] *28	[8, 19, 5] [8, 17, 6] *29	
CTM=4	CLASSE SIMPLES	$\langle G_{45}; 4, 0, 0, 0 \rangle$	$\langle G_{45}; 3, 1, 0, 0 \rangle$	$\langle G_{45}; 2, 1, 1, 0 \rangle$
	$\Delta_{\min}(\mu)$	1.169	1.089	0.988
	$\Delta(\mu+1)$	0.884	1.250	1.250
	$d_{\min}$ TCM	0.884	1.089	0.988
	GCA	7.107	8.871	8.079
	Melhores Códigos [VC(1), ..., VC(N)]	[4, 19] [8, 19] [14, 19] *30	[24, 14, 33] [24, 6, 35] *31	[32, 20, 10, 65] *32
CTM=5	CLASSE SIMPLES	$\langle G_{45}; 5, 0, 0, 0 \rangle$	$\langle G_{45}; 4, 1, 0, 0 \rangle$	$\langle G_{45}; 3, 1, 1, 0 \rangle$
	$\Delta_{\min}(\mu)$	1.250	1.089	1.089
	$\Delta(\mu+1)$	0.884	1.250	1.250
	$d_{\min}$ TCM	0.884	1.089	1.089
	GCA	7.107	8.871	8.871
	Melhores Códigos [VC(1), ..., VC(N)]	[54, 11] [34, 18] *33	[56, 6, 65] *34	*35

RETICULADO A_2 - TAXA $\frac{4}{5}$ Partição de conjunto: [fig. A_2 . iii-b]

$$\Delta_0 = 0.475$$

$$\Delta_1 = 0.475$$

$$\Delta_2 = 0.950$$

$$\Delta_3 = 0.950$$

$$\Delta_4 = 1.643$$

(* → Apêndice 1)

$$\sum \{|a_n|\} = \frac{71}{16} \Delta_0^2 = 1$$

TABELA III.1.12:

Resultado da procura por melhores códigos TCM para constelação de 32 sinais [II.iii-b] com GCA obtido em relação ao 8-PSK não-codificado.

CTM=3	CLASSE SIMPLES	$\langle G_{45}; 3, 0, 0, 0 \rangle$	$\langle G_{45}; 2, 1, 0, 0 \rangle$	
	$\Delta_{\min}(\mu)$	0.950	0.823	
	$\Delta(\mu+1)$	0.950	0.950	
	$d_{\min}$ TCM	0.950	0.823	
	GCA	7.733	6.486	
	Melhores códigos [VC(1), ..., VC(N)]	[19, 2] [15, 2] [19, 3] *36	[24, 6, 1] [30, 6, 1] *37	
CTM=4	CLASSE SIMPLES	$\langle G_{45}; 4, 0, 0, 0 \rangle$	$\langle G_{45}; 3, 1, 0, 0 \rangle$	$\langle G_{45}; 2, 1, 1, 0 \rangle$
	$\Delta_{\min}(\mu)$	1.062	0.950	0.823
	$\Delta(\mu+1)$	0.950	0.950	1.643
	$d_{\min}$ TCM	0.950	0.950	0.823
	GCA	7.733	7.733	6.486
	Melhores códigos [VC(1), ..., VC(N)]	[26, 5] [31, 5] [22, 9] *38	[24, 14, 33] [24, 6, 35] *39	[32, 20, 10, 65] *40
CTM=5	CLASSE SIMPLES	$\langle G_{45}; 5, 0, 0, 0 \rangle$	$\langle G_{45}; 4, 1, 0, 0 \rangle$	$\langle G_{45}; 3, 1, 1, 0 \rangle$
	$\Delta_{\min}(\mu)$	1.164	1.062	0.950
	$\Delta(\mu+1)$	0.950	0.950	1.643
	$d_{\min}$ TCM	0.950	0.950	0.950
	GCA	7.733	7.733	7.733
	Melhores códigos [VC(1), ..., VC(N)]	[55, 10] [61, 10] *41	[56, 22, 65] *42	*43

IV - ANÁLISE DOS RESULTADOS

IV.1 - TAXA $\frac{2}{3}$

Os resultados da procura efetuada pelos melhores códigos TCM, para as constelações de 8 sinais, taxa $\frac{2}{3}$, não apresentam ganhos expressivos em relação ao 4-PSK não codificado tanto em A_2 quanto em Z^2 , porém servem como base para comparação entre as constelações propostas e confirmação de algumas tendências anteriormente mencionadas à respeito das constelações e dos tipos de sinais estudados.

Os códigos obtidos da procura efetuada nas classes simples de codificadores que compõem os conjuntos de 8 sinais tipo A_2 apresentam um ganho assintótico de codificação que varia de 1.249 a 1.412 dB em relação à 4-PSK não-codificado.

Os valores obtidos para conjuntos de 8 sinais tipo Z^2 apresentam um GCA que varia de -0.007 a 1.454 dB sobre 4-PSK não-codificado.

O maior valor obtido de $d_{\min}$ para esta taxa foi 1.672 com um ganho relativo de 1.454 dB para Z^2 . 8-b: $\langle G_{23}; 4,0 \rangle$ e este valor ainda poderia ser melhorado caso aumentássemos o CTM (comprimento total de memória) para 5.

Contrariando as expectativas com relação à compactação dos conjuntos de sinais, observa-se o exemplo desta constelação que tem a característica de ser a constelação de 8 sinais menos compacta das estudadas $\left[\bar{8} \left\{ |a_n^2| \right\} = 2.5 \Delta_0^2 \text{ ou } \Delta_0 = 0.692 \right]$. Apesar deste fato, apresenta o melhor desempenho obtido, uma vez que a partição escolhida para este conjunto de sinais ($Z^{2.8-b}$), resultou no maior valor alcançado para a distância dos sinais no último nível da partição ($\Delta_{\text{final}} = 1.787$). Ocorrendo o melhor desmembramento dos sinais com a partição, comparado às demais partições das constelações de 8 sinais, tanto de Z^2 como de A_2 .

O que se pode constatar logo no início da análise destes resultados é que a constelação com o menor Δ_{inicial} através do processo de partição de conjunto pôde alcançar o maior valor Δ_{final} dos conjuntos de 8 sinais e este valor (por estar diretamente ligado às transições paralelas) foi fator preponderante para o melhor resultado obtido.

Observando os resultados, o que também pode ser constatado é que $\Delta_{(\mu+1)}$ ou Δ_{final} (distância Euclideana dos sinais no último nível da partição) é um dos fatores limitantes na obtenção de melhores resultados, isto é, se pudermos aumentar o CTM (comprimento total de memória) dentro dos limites de decodificação, pode-se alcançar um valor $\Delta_{\text{min}}^{(\mu)}$ (d_{min} relativo às transições adjacentes ou em outras palavras, d_{min} do código convolucional embutido no código TCMD tão próximo ou igual a $\Delta_{(\mu+1)}$ (fator limitante); e estaremos, portanto, otimizando a procura. Lembrando que o valor escolhido para d_{min} será sempre o menor destes valores.

As constelações de 8 sinais oriundas de A_2 apresentam no último nível da partição um espaçamento entre os sinais superior a 1.414 ($d_{\min}$ 4-PSK não-codificado), isto justifica a obtenção de ganhos positivos de codificação em todas as procuras efetuadas em A_2 , fato que não ocorreu para a procura efetuada em Z^2 (GCA = -0.007 para $Z^2.8-b: \langle G_{29}:2,0 \rangle$ e $Z^2.8-b': \langle G_{29}:2,0 \rangle, \langle G_{29}:3,0 \rangle, \langle G_{29}:4,0 \rangle$) estes valores negativos de GCA caracterizam um desempenho inferior ao 4-PSK não-codificado.

IV.2 - TAXA $\frac{9}{4}$

Os ganhos de codificação obtidos para os códigos gerados com esta taxa apresentam resultados bastante significativos tanto para sinais tipo Z^2 quanto A_2 — variam de 4.368 a 6.122 dB sobre 8-PSK não-codificado para Z^2 , e de 3.570 a 4.946 dB para sinais tipo A_2 .

Conforme já havíamos mencionado a procura por códigos TCM mapeados em II.4-a e II.4-c apresentam os mesmos resultados, pois apesar destas constelações apresentarem uma configuração diferente dos sinais, possuem os mesmos valores de distância Δ_i na partição de conjuntos de seus sinais.

O melhor desempenho foi obtido para $Z^2.16:(G_{34}; 9,1,0)$ com $d_{\min}$ de 1.548 e um ganho sobre 8-PSK não-codificado de 6.122 dB.

IV.3 - TAXA $\frac{4}{5}$

Os resultados encontrados para esta taxa são os mais satisfatórios das procuras efetuadas. Os valores variam de 7.107 a 8.871 dB sobre 16-PSK não-codificado para Z^2 e de 6.486 a 7.733 dB para classe simples de códigos gerados a partir de sinais tipo A_2 .

O melhor desempenho alcançado corresponde a Z^2 . 32: $(G_{45}:3,1,0,0), (G_{45}:4,1,0,0)$ com $d_{\min} = 1.083$ e um ganho relativo à 16-PSK não-codificado de 8.871 dB.

As constelações tomadas como referência neste trabalho — 4-PSK, 8-PSK e 16-PSK não-codificado — possuem como $d_{\min}$ exatamente a distância entre dois sinais consecutivos na constelação. Para quantidades iguais de energia, a distância entre dois sinais nestas constelações obedecem a seguinte relação: $d_{16-PSK} < d_{8-PSK} < d_{4-PSK}$, portanto nestas constelações não-codificadas, a distância entre os sinais diminuem com o crescimento do número de sinais.

Este valor para distância, porém, é o denominador da equação

$$GCA = 20 \log \frac{d_{\min}}{d_{ref}} \quad [\text{Cap. I, Eq. III.5.5}], \text{ portanto, basta que}$$

$$\frac{d_{\min}}{d_{ref}} > 1 \text{ para que tenhamos ganhos positivos de codificação ou}$$

ainda que $d_{\min} > d_{ref}$. Como o valor de d_{ref} decresce com o aumento do número de sinais, caracteriza-se a tendência de

encontrar ganhos superiores para códigos obtidos do mapeamento em constelações com maior número de sinais, desde que o valor $d_{\min}$ permaneça constante ou decresça numa taxa inferior a d_{ref} , como de fato acontece para constelações de 8, 16 e 32 sinais estudadas neste trabalho, e esta tendência deve se confirmar em constelações de cardinalidade superior.

Observando os valores Δ_i na partição do Reticulado A_2 , constata-se que o particionamento de sinais nos primeiros níveis da partição de constelações originadas deste reticulado, limitam o desempenho alcançado por códigos mapeados nestas constelações. A justificativa para isto advem do fato de que a estrutura hexagonal não favorece a simetria do posicionamento dos sinais quando da partição. No entanto, para constelações superiores em número de sinais onde a partição atinge níveis maiores ou iguais a 4, outra tendência se faz evidente.

Vejamos:

Apesar de não ter sido feita a procura na classe simples $\langle G_{45}; 5, 1, 1, 0 \rangle$ com $\text{CTM} = 7$, projeta-se para esta classe simples a expectativa dos melhores resultados nesta taxa de codificação, um ganho de 10.116 dB sobre 16-PSK não-codificado para $Z^2.32$ e 9.497 dB para $A_2.32-b$. Para esta constatação basta observar os valores de $\Delta_{\min}(\mu)$ encontrados na procura efetuada em $\langle G_{45}; 5, 0, 0, 0 \rangle$ tanto de Z^2 como de A_2 — $\Delta_{\min}(\mu) = 1.250$ e $\Delta_{\min}(\mu) = 1.164$ para $Z^2.32$ e $A_2.32-b$ respectivamente — se codificarmos a 2ª e 3ª entrada com pelo menos um elemento de memória, elevando-se o CTM de 5 para $\text{CTM}=7$ atingiremos o último nível da partição (Δ_4), onde a distância entre os sinais transmitidos é a maior possível e também o fator limitante para

os ganhos nesta classe. Tanto para os códigos projetados em Z^2 como os códigos projetados em A_2 o valor de $d_{\min}$ será $\Delta(\mu)$. No caso de Z^2 o valor de $d_{\min}$ não pode ser melhorado pois $\Delta(\mu) = \Delta(\mu+1)$ (isto é, o valor de distância das transições adjacentes é igual ao valor de distância das transições paralelas). Para o caso de A_2 , $\Delta(\mu)$ é menor que o fator limitante $\Delta(\mu+1)$. Neste caso pode-se melhorar ainda mais o desempenho da procura efetuada para A_2 taxa $\frac{4}{5}$ em $A_2.32-b$, bastando para isto aumentar-se o número de memórias nas entradas codificadas, levando-se em conta um grande aumento na complexidade de decodificação.

Iremos caracterizar como *Fator Limitante de Desempenho Assintótico (FLDA)* o valor estipulado à distância Euclideana dos sinais do último nível da partição de sinais ($\Delta(\mu+1)$) e que vem a ser também o valor diretamente relacionado às transições paralelas. O limite do desempenho que poderia alcançar um código TCM é função deste valor, supondo uma capacidade infinita de decodificação, portanto, um maximizante superior para o GCA

$$GCA = 20 \log \frac{d_{\min}}{d_{\text{ref}}} \leq 20 \log \frac{\Delta(\mu+1)}{d_{\text{ref}}} = GCA_{\max}$$

Para II.iii-b, com particionamento mostrado em $A_2.32-b$ a uma taxa $\frac{4}{5}$ de codificação, supondo uma capacidade infinita de decodificação, obteve-se $\Delta(\mu+1) = 1.649$, portanto

$$GCA_{\max} = 20 \log \frac{\Delta(\mu+1)}{d_{\text{ref}}} = 20 \log \frac{1.649}{0.990} = 13.491 \text{ dB}$$

Para I.iii, com particionamento mostrado em $Z^2.32$ a uma

taxa $\frac{4}{5}$ de codificação supondo capacidade máxima de decodificação, obteve-se $\Delta(\mu+1) = 1.250$

$$GCA_{\max} = 20 \log \frac{1.250}{0.990} = 10.116 \text{ dB}$$

Os valores $FLDA$ e $GCA_{\max}$ permitem a comparação entre as diferentes configurações de sinais, gerados a partir de qualquer estrutura reticular levando-se apenas em consideração a capacidade máxima de codificação destes sinais sem levar em conta sua decodificação.

O que se pode constatar para os casos acima mencionados, sob a hipótese das constelações apresentarem um maior número de pontos de sinais (64, 128, 256, ...), com correspondentes taxas de codificação $\left[\frac{5}{6}, \frac{6}{7}, \frac{7}{8}, \dots \right]$, é que existe uma tendência de encontrarem-se valores maiores de $FLDA$ (que levem a um $GCA_{\max}$ superior) para os mapeamentos feitos em A_2 . Da mesma forma, ao admitir-se uma complexidade computacional substancial de decodificação, os valores de $FLDA$ tendem a ser superiores aos valores projetados para Z^2 .

A justificativa para esta hipótese vem da densidade do empacotamento esférico inerente a A_2 ($\Delta_H = 1.1548 \Delta_Q$ o maior em 2 dimensões), e a perspectiva de haver um melhor desmembramento dos sinais reticulados A_2 a partir do 4º nível da partição.

A procura por codificadores TCM ótimos com taxas altas, para constelações projetadas em Z^2 e A_2 , fica como sugestão para futuros trabalhos. A idéia de se encontrar constelações oriundas de estruturas reticulares mais densas persiste, e o próximo passo será estudar constelações de 4 dimensões.

CAPÍTULO 4

ANÁLISE DE ESTRUTURAS RETICULARES PARA MODULAÇÃO
POR CODIFICAÇÃO EM TRELIÇA EM 4 DIMENSÕES

I - INTRODUÇÃO

A idéia principal do sistema TCM consiste em um mapeamento específico da saída de um codificador convolucional em uma dada constelação de sinais. Conforme pode ser observado nos capítulos anteriores, a configuração dos sinais no espaço de sinais é de grande relevância na obtenção dos melhores desempenhos para este sistema. Foi com o objetivo de particularizar o posicionamento dos sinais no espaço de sinais que se procurou estudar determinadas estruturas reticulares em 2 dimensões (A_2 e Z^2) e comparou-se o desempenho relativo entre os sistemas definidos sobre as mesmas.

Neste Capítulo continuaremos com a idéia de abordar estruturas reticulares que apresentem a maior compactação possível dos sinais que representam, pois como vimos anteriormente, esta característica evidencia bons desempenhos.

Decidimos então direcionar nossas análises para dimensões superiores onde a compactação dos sinais é maior do que em 2 dimensões. Serão introduzidas, portanto, algumas estruturas reticulares em 4 dimensões, tais como a Z^4 , a A_4 , a D_4 e a $A_2 \times A_2$, que poderão servir como base para projeção de códigos TCM 4-D em trabalhos futuros. Será também descrita a partição de conjunto para esquemas TCM utilizando conjuntos de sinais definidos em Z^4 e $A_2 \times A_2$.

II - ESTRUTURAS RETICULARES EM 4-D

II.1 - Z^4

Z^4 é caracterizado como o reticulado inteiro em 4-D e consiste de todos os pontos em $\mathbb{R}^n$ com coordenadas inteiras. Z^4 também é denominado reticulado Cúbico Regular [7]. Os parâmetros que definem este reticulado são:

$d = 4$ dimensões,

$M = I_4$ ou seja $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ é a Matriz Geradora de Z^4 ,

$\det A = 1$,

$\tau(x) = 8$ (número de contatos de uma hiper-esfera centrada em $x^{(i)}$),

$$\Delta = \frac{V_n \rho^n}{\det A} = \frac{\pi^{n/2} \rho^n}{\Gamma((n/2)+1) \det A} = \frac{\pi^2}{32} = 0.3084 \text{ é a}$$

a densidade do empacotamento esférico Z^4 para um empacotamento de raio $\rho = 1/2$ e dimensão $d = 4$.

A construção de um conjunto de sinais A_4^0 tipo Z^4 pode ser feita através do produto cartesiano de um conjunto de sinais tipo Z^2 com ele mesmo, isto é, esta operação consiste de todas as concatenações dos elementos do primeiro conjunto com os elementos do segundo, isto é,

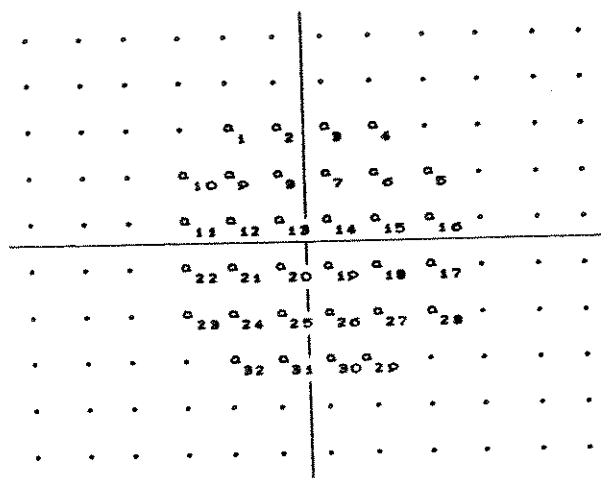
$$A_4^0 = A_0 \times A_0$$

onde A_0 é ilustrado pela Figura II.1.1

A_0 é portanto a constelação de 32 sinais tipo Z^2 bem definida no Cap.2, fig. I.iii, gerada em função de Δ_0 no ponto-lacuna $(1/2, 1/2)$ do reticulado Z^2 .

Com o objetivo de construir A_4^0 e calcular o valor de Δ_0 para uma quantidade de Energia Média Quadrática igual a 1 e A_0 definido como um conjunto de pontos bi-dimensionais, foi elaborado um programa computacional em linguagem Pascal (Apêndice 2 - READA e DELTA0A).

O Algoritmo lê o conjunto de pontos A_0 e cria A_4^0 , isto é, faz o produto cartesiano $A_0 \times A_0$. São gerados 1024 pontos tipo Z^4 em função de Δ_0 . A seguir é calculado o centro de massa deste empacotamento esférico — $\mathcal{M} = \left[\left(\sum x_i, \sum y_i, \sum z_i, \sum w_i \right) \Delta_0 / n \right]$ — e a somatória das distâncias Euclidianas quadradas de cada ponto a este $\mathcal{M}$ produz a Energia Média Quadrática deste empacotamento em função de $(\Delta_0)^2$. Igualando-se este valor a 1 obtém-se finalmente o valor de Δ_0 . Posteriormente são listados os 1024 pontos tipo Z^4 em função do valor encontrado.



CONJUNTO A_0

$a_1 \rightarrow (-1.5000\Delta_0, 2.5000\Delta_0)$	$a_{17} \rightarrow (2.5000\Delta_0, -0.5000\Delta_0)$
$a_2 \rightarrow (-0.5000\Delta_0, 2.5000\Delta_0)$	$a_{18} \rightarrow (1.5000\Delta_0, -0.5000\Delta_0)$
$a_3 \rightarrow (0.5000\Delta_0, 2.5000\Delta_0)$	$a_{19} \rightarrow (0.5000\Delta_0, -0.5000\Delta_0)$
$a_4 \rightarrow (1.5000\Delta_0, 2.5000\Delta_0)$	$a_{20} \rightarrow (-0.5000\Delta_0, -0.5000\Delta_0)$
$a_5 \rightarrow (2.5000\Delta_0, 1.5000\Delta_0)$	$a_{21} \rightarrow (-1.5000\Delta_0, -0.5000\Delta_0)$
$a_6 \rightarrow (1.5000\Delta_0, 1.5000\Delta_0)$	$a_{22} \rightarrow (-2.5000\Delta_0, -0.5000\Delta_0)$
$a_7 \rightarrow (0.5000\Delta_0, 1.5000\Delta_0)$	$a_{23} \rightarrow (-2.5000\Delta_0, -1.5000\Delta_0)$
$a_8 \rightarrow (-0.5000\Delta_0, 1.5000\Delta_0)$	$a_{24} \rightarrow (-1.5000\Delta_0, -1.5000\Delta_0)$
$a_9 \rightarrow (-1.5000\Delta_0, 1.5000\Delta_0)$	$a_{25} \rightarrow (-0.5000\Delta_0, -1.5000\Delta_0)$
$a_{10} \rightarrow (-2.5000\Delta_0, 1.5000\Delta_0)$	$a_{26} \rightarrow (0.5000\Delta_0, -1.5000\Delta_0)$
$a_{11} \rightarrow (-2.5000\Delta_0, 0.5000\Delta_0)$	$a_{27} \rightarrow (1.5000\Delta_0, -1.5000\Delta_0)$
$a_{12} \rightarrow (-1.5000\Delta_0, 0.5000\Delta_0)$	$a_{28} \rightarrow (2.5000\Delta_0, -1.5000\Delta_0)$
$a_{13} \rightarrow (-0.5000\Delta_0, 0.5000\Delta_0)$	$a_{29} \rightarrow (1.5000\Delta_0, -2.5000\Delta_0)$
$a_{14} \rightarrow (0.5000\Delta_0, 0.5000\Delta_0)$	$a_{30} \rightarrow (0.5000\Delta_0, -2.5000\Delta_0)$
$a_{15} \rightarrow (1.5000\Delta_0, 0.5000\Delta_0)$	$a_{31} \rightarrow (-0.5000\Delta_0, -2.5000\Delta_0)$
$a_{16} \rightarrow (2.5000\Delta_0, 0.5000\Delta_0)$	$a_{32} \rightarrow (-1.5000\Delta_0, -2.5000\Delta_0)$

Fig. II.1.1: Constelação de 32 sinais tipo Z^2 (Cap. 2, Fig. I.iii)

Utilizando os programas READA e DELTAOA (Apêndice 2) os valores encontrados para o centro de massa, energia média e Δ_0 para o reticulado Z^4 são respectivamente:

$$\bar{m} = (0.0000, 0.0000, 0.0000, 0.0000)$$

$$\bar{s} \left\{ |\alpha_n^2| \right\} = 10.0000 \Delta_0^2 = 1$$

$$\Delta_0 = 0.3162$$

O crescimento das distâncias intraconjunto para o reticulado Z^4 pode ser facilmente obtido e o mesmo é dado por

$$\Delta_0, \Delta_1 = \Delta_2 = \sqrt{2} \Delta_0, \Delta_3 = \Delta_4 = \sqrt{4} \Delta_0$$

portanto,

$$\Delta_1 = \Delta_2 = 0.4471$$

$$\Delta_3 = \Delta_4 = 0.6324$$

Estes resultados devem ser confirmados mais adiante quando do cálculo da distância intraconjunto no nível C da partição para o esquema TCM em consideração.

II.2 - A_4

A_2 pode ser generalizado para dimensões superiores assim como segue [7]:

Descreve-se A_n como sendo um reticulado n -dimensional em $\mathbb{R}^{n+1}$. Assim para $n \geq 2$

$$A_n = \left\{ x \in \mathbb{Z}^{n+1} \mid \sum_{j=1}^{n+1} x_j = 0 \right\}$$

Esta definição, no caso de $n = 2$, está de acordo com a definição apresentada no Cap.2, a menos de uma rotação se a versão anterior for reescalada multiplicando-se por $c = \sqrt{2}$.

Freqüentemente é necessário reescalonar-se um empacotamento esférico, substituindo-se Λ por $\Lambda' = c\Lambda = \{cx : x \in \Lambda\}$ para alguma apropriada constante $c \in \mathbb{R}$. Os parâmetros de Λ' e Λ estão relacionados entre si por

$$\rho' = c\rho$$

$$\dim \Lambda' = \dim \Lambda$$

$$M' = cM$$

$$\det \Lambda' = c^{\dim \Lambda} \cdot \det \Lambda$$

$$\Delta' = \Delta$$

$$\tau'(\omega) = \tau(\omega) \quad , \quad \tau' = \tau$$

Portanto se $c = \sqrt{2}$ então A_n tem $\dim = n$, e a matriz geradora M para este reticulado é dada por

$$M = \begin{bmatrix} 1 & -1 & 0 & \dots & 0 \\ 1 & 0 & -1 & \dots & 0 \\ & \dots & & \dots & \\ 1 & 0 & 0 & \dots & -1 \end{bmatrix}$$

com dimensão $n \times (n+1)$, $\rho = \frac{1}{\sqrt{2}}$, $\det = \sqrt{n+1}$ e o $\tau = 2n(n-1)$.

Podemos considerar A_4 como um reticulado no $\mathbb{R}^4$ restringindo nossa atenção ao hiperplano:

$$\left\{ x \in \mathbb{R}^4 \mid x_1 + \dots + x_5 = 0 \right\}.$$

Desta forma, A_4 tem densidade

$$\Delta = \frac{V_4}{2^{4/2} \sqrt{4n+1}} = \frac{\pi^{n/2}}{\Gamma((n/2)+1) 4\sqrt{5}} = 0.5172$$

II.3 - D_4

O empacotamento esférico D_n em $\mathbb{R}^n$ é obtido aplicando-se a construção $\mathcal{A}$ em um código consistindo de todas os vetores binários de peso par, e depois reescalando-se todos os centros esféricos por $1/\sqrt{2}$.

Construção $\mathcal{A}$:

Um código (n, M, d) sobre $\mathcal{GF}(q)$ é um conjunto de M vetores (palavras-código) de comprimento n com símbolos de $\mathcal{GF}(q)$ tais que a distância de Hamming entre quaisquer duas palavras-código é pelo menos d , isto é, a distância de Hamming entre dois vetores é o número de componentes onde eles diferem.

Se c é uma palavra-código e $\mathcal{A}_i(c)$ denota o número de palavras-código com distância de Hamming i de c . Então

$$\sum_i \mathcal{A}_i(c) = M.$$

Para códigos de Grupo $\mathcal{A}_i(c)$ independe de c e é denotado por $\mathcal{A}_i$.

Se a soma de todos as componentes de quaisquer duas palavras-código é uma palavra-código, e o produto de todos os componentes de quaisquer palavra-código e qualquer elemento de $\mathcal{GF}(q)$ é uma palavra-código, então o código é um código de grupo.

Para a construção $\mathcal{A}$ é necessário que $\mathcal{C}$ seja um código binário (n, M, d) . Esta construção especifica o conjunto de centros para um empacotamento esférico em $\mathbb{R}^n$, isto é,

$x = (x_1, \dots, x_n)$ é um centro se e somente se x é congruente (módulo 2) a uma palavra-código de $\mathcal{C}$.

O empacotamento de um reticulado é obtido se e somente se, $\mathcal{C}$ é um código de Grupo [9].

Para o cubo unitário centrado na origem,

$$\langle 0 \leq x_i \leq 1: 1, \dots, n \rangle,$$

os centros são exatamente as M palavras-código. Todos os outros centros são obtidos adicionando inteiros a quaisquer coordenadas de uma palavra-código. Isto corresponde a transportar o cubo unitário duas vezes em qualquer direção. Assim todos os centros podem ser obtidos repetindo o bloco construído.

Cada cópia de $2 \times 2 \times 2 \times \dots \times 2$ cubos contribui com M esferas de raio ρ , então o centro de densidade obtido da construção $\mathcal{A}$ é

$$\delta = M \rho^n 2^{-n}.$$

Se dois centros são congruentes à mesma palavra-código sua separação é de pelo menos 2. Se eles são congruentes a diferentes palavras-código então os mesmos diferem por pelo menos 1 em pelo menos d lugares então são separados em ao menos $\sqrt{d}$.

Assim pode-se considerar o raio das esferas como sendo

$$\rho = \frac{1}{2} \min(2, \sqrt{d}).$$

Portanto, se $\mathcal{C}$ é um código binário de comprimento n , o conjunto de centros $c+2x$ ($c \in \mathcal{C}$, $x \in \mathbb{Z}^n$) forma o empacotamento esférico $\mathbb{D}_n$ em $\mathbb{R}^n$. A maioria das propriedades deste empacotamento pode ser obtida diretamente do código $\mathcal{C}$.

O reticulado $\mathbb{D}_n$ em $\mathbb{R}^n$ é obtido aplicando-se a Construção $\mathcal{A}$ no código de Grupo $\mathcal{C}(n, 2^{n-1}, 2)$ consistindo de todas as palavras-código de peso par, e então reescalando-se todos os centros por $\frac{1}{\sqrt{2}}$. Para este reticulado temos os seguintes valores

$$d = n,$$

$$\mathcal{A}_2 = \frac{1}{2} n(n-1),$$

$$M = \begin{bmatrix} 2 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ 1 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 1 & 0 & 0 & \dots & 1 \end{bmatrix},$$

$$\rho = \frac{1}{2},$$

$$\det = 2^{-(n-2)/2},$$

$$\tau = 2n(n-1),$$

$$\Delta = V_n 2^{-(n+2)/2}$$

Os centros de $\mathbb{D}_n$ são vértices alternados de um reticulado Cúbico Regular, isto é, pode-se imaginar $\mathbb{D}_n$ colorindo-se alternadamente de preto e branco os pontos (reticulados inteiros $\mathbb{Z}^n$) de um tabuleiro de xadrez. $\mathbb{D}_n$ consiste então dos pontos pretos multiplicados por $1/\sqrt{2}$.

Para um estudo mais aprofundado de $\mathbb{D}_n$ recomendamos [7,8,9,18].

O reticulado $\mathbb{D}_4$ ou "Schläfli Lattice $\mathbb{D}_4$ " é o reticulado mais denso em $\mathbb{R}^4$ [7,9] e é gerado pelos vetores

$$x^{(1)} = \frac{1}{\sqrt{2}}(2, 0, 0, 0),$$

$$x^{(2)} = \frac{1}{\sqrt{2}}(1, 1, 0, 0)$$

$$x^{(3)} = \frac{1}{\sqrt{2}}(1, 0, 1, 0)$$

$$x^{(4)} = \frac{1}{\sqrt{2}}(1, 0, 0, 1).$$

cuja matriz geradora é dada por

$$M = \frac{1}{\sqrt{2}} \begin{bmatrix} 2 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix},$$

$$\text{com } \det \mathbb{D}_4 = |\det M| = \frac{1}{2}.$$

Os 24 pontos mais próximos da origem são dados por:

$$\frac{1}{\sqrt{2}}(\pm 1, \pm 1, 0, 0); \frac{1}{\sqrt{2}}(\pm 1, 0, \pm 1, 0); \frac{1}{\sqrt{2}}(\pm 1, 0, 0, \pm 1);$$

$$\frac{1}{\sqrt{2}}(0, \pm 1, \pm 1, 0); \frac{1}{\sqrt{2}}(0, \pm 1, 0, \pm 1); \frac{1}{\sqrt{2}}(0, 0, \pm 1, \pm 1)$$

O raio das esferas é $\rho = \frac{1}{2}$, o número de contatos é $\tau = 24$, e a densidade vale

$$\Delta = \frac{\pi^2}{16} = 0.61686\dots$$

Exemplo II.3.1:

Em [6] é listada uma tabela com $4^4 = 256$ vetores em ordem lexicográfica, os quais representam uma das melhores maneiras encontradas de se compactar um subconjunto de 256 pontos tipo $\mathbb{D}_4$ em $\mathbb{R}^4$, de tal maneira a apresentarem a menor energia média. Visto de forma simplificada, o que se deseja é determinar o ponto a tal que adicionado a cada vetor de $\mathbb{D}_4$ desloca a origem do subconjunto de forma a obter a energia média mínima para o mesmo. Estes pontos serão listados a seguir (Tabela III.3.1) para

$$a = (0, \frac{9}{16}, \frac{11}{32}, \frac{17}{32})$$

TABELA III.3.1

(-2, 0, 0, 1)	(-2, 0, 1, 0)	(-2, 1, 0, 0)	(-2, 1, 1, 1)
(-2, -1, -1, 0)	(-2, -1, -1, 2)	(-2, -1, 0, -1)	(-2, -1, 0, 1)
(-2, -1, 1, 0)	(-2, -1, 1, 2)	(-2, -1, 2, -1)	(-2, -1, 2, 1)
(-2, 0, -1, -1)	(-2, 0, -1, 1)	(-2, 0, 0, 0)	(-2, 0, 0, 2)
(-2, 0, 1, -1)	(-2, 0, 1, 1)	(-2, 0, 2, 0)	(-2, 0, 2, 2)
(-2, 1, -1, 0)	(-2, 1, -1, 2)	(-2, 1, 0, -1)	(-2, 1, 0, 1)
(-2, 1, 1, 0)	(-2, 1, 1, 2)	(-2, 1, 2, -1)	(-2, 1, 2, 1)
(-2, 2, -1, -1)	(-2, 2, -1, 1)	(-2, 2, 0, 0)	(-2, 2, 0, 2)
(-2, 2, 1, -1)	(-2, 2, 1, 1)	(-2, 2, 2, 0)	(-2, 2, 2, 2)
(-1, -2, -1, 0)	(-1, -2, -1, 2)	(-1, -2, 0, -1)	(-1, -2, 0, 1)
(-1, -2, 1, 0)	(-1, -2, 1, 2)	(-1, -2, 2, -1)	(-1, -2, 2, 1)
(-1, -1, -2, 0)	(-1, -1, -2, 2)	(-1, -1, -1, -1)	(-1, -1, -1, 1)
(-1, -1, -1, 3)	(-1, -1, 0, -2)	(-1, -1, 0, 0)	(-1, -1, 0, 2)
(-1, -1, 1, -1)	(-1, -1, 1, 1)	(-1, -1, 1, 3)	(-1, -1, 2, 0)
(-1, -1, 2, 2)	(-1, -1, 3, 1)	(-1, 0, -2, -1)	(-1, 0, -2, 1)
(-1, 0, -1, -2)	(-1, 0, -1, 0)	(-1, 0, -1, 2)	(-1, 0, 0, -1)
(-1, 0, 0, 1)	(-1, 0, 0, 3)	(-1, 0, 1, -2)	(-1, 0, 1, 0)
(-1, 0, 1, 2)	(-1, 0, 2, -1)	(-1, 0, 2, 1)	(-1, 0, 3, 0)
(-1, 1, -2, 0)	(-1, 1, -2, 2)	(-1, 1, -1, -1)	(-1, 1, -1, 1)
(-1, 1, -1, 3)	(-1, 1, 0, -2)	(-1, 1, 0, 0)	(-1, 1, 0, 2)
(-1, 1, 1, -1)	(-1, 1, 1, 1)	(-1, 1, 1, 3)	(-1, 1, 2, 0)
(-1, 1, 2, 2)	(-1, 1, 3, 1)	(-1, 2, -1, 0)	(-1, 2, -1, 2)
(-1, 2, 0, -1)	(-1, 2, 0, 1)	(-1, 2, 1, 0)	(-1, 2, 1, 2)
(-1, 2, 2, -1)	(-1, 2, 2, 1)	(-1, 3, 0, 0)	(-1, 3, 1, 1)
(0, -3, 0, 1)	(0, -3, 1, 0)	(0, -2, -1, -1)	(0, -2, -1, 1)
(0, -2, 0, 0)	(0, -2, 0, 2)	(0, -2, 1, -1)	(0, -2, 1, 1)
(0, -2, 2, 0)	(0, -2, 2, 2)	(0, -1, -2, -1)	(0, -1, -2, 1)
(0, -1, -1, 1)	(0, -1, -1, 0)	(0, -1, -1, 2)	(0, -1, 0, -1)
(0, -1, 0, 1)	(0, -1, 0, 3)	(0, -1, 1, -2)	(0, -1, 1, 0)
(0, -1, 1, 2)	(0, -1, 2, -1)	(0, -1, 2, 1)	(0, -1, 3, 0)
(0, 0, -3, 1)	(0, 0, -2, 0)	(0, 0, -2, 2)	(0, 0, -1, -1)
(0, 0, -1, 1)	(0, 0, -1, 3)	(0, 0, 0, -2)	(0, 0, 0, 0)
(0, 0, 0, 2)	(0, 0, 0, 4)	(0, 0, 1, -1)	(0, 0, 1, 1)
(0, 0, 1, 3)	(0, 0, 2, 0)	(0, 0, 2, 2)	(0, 0, 3, 1)
(0, 1, -2, -1)	(0, 1, -2, 1)	(0, 1, -1, -2)	(0, 1, -1, 0)
(0, 1, -1, 2)	(0, 1, 0, -1)	(0, 1, 0, 1)	(0, 1, 0, 3)
(0, 1, 1, -2)	(0, 1, 1, 0)	(0, 1, 1, 2)	(0, 1, 2, -1)
(0, 1, 2, 1)	(0, 1, 3, 0)	(0, 2, -1, -1)	(0, 2, -1, 1)
(0, 2, 0, 0)	(0, 2, 0, 2)	(0, 2, 1, -1)	(0, 2, 1, 1)
(0, 2, 2, 0)	(0, 2, 2, 2)	(0, 3, 0, 1)	(0, 3, 1, 0)
(1, -2, -1, 0)	(1, -2, -1, 2)	(1, -2, 0, -1)	(1, -2, 0, 1)
(1, -2, 1, 0)	(1, -2, 1, 2)	(1, -2, 2, -1)	(1, -2, 2, 1)
(1, -1, -2, 0)	(1, -1, -2, 2)	(1, -1, -1, -1)	(1, -1, -1, 1)
(1, -1, -1, 3)	(1, -1, 0, -2)	(1, -1, 0, 0)	(1, -1, 0, 2)
(1, -1, 1, -1)	(1, -1, 1, 1)	(1, -1, 1, 3)	(1, -1, 2, 0)
(1, -1, 2, 2)	(1, -1, 3, 1)	(1, 0, -2, -1)	(1, 0, -2, 1)
(1, 0, -1, -2)	(1, 0, -1, 0)	(1, 0, -1, 2)	(1, 0, 0, -1)
(1, 0, 0, 1)	(1, 0, 0, 3)	(1, 0, 1, -2)	(1, 0, 1, 0)
(1, 0, 1, 2)	(1, 0, 2, -1)	(1, 0, 2, 1)	(1, 0, 3, 0)
(1, 1, -2, 0)	(1, 1, -2, 2)	(1, 1, -1, -1)	(1, 1, -1, 1)
(1, 1, -1, 3)	(1, 1, 0, -2)	(1, 1, 0, 0)	(1, 1, 0, 2)
(1, 1, 1, -1)	(1, 1, 1, 1)	(1, 1, 1, 3)	(1, 1, 2, 0)
(1, 1, 2, 2)	(1, 1, 3, 1)	(1, 2, -1, 0)	(1, 2, -1, 2)
(1, 2, 0, -1)	(1, 2, 0, 1)	(1, 2, 1, 0)	(1, 2, 1, 2)
(1, 2, 2, -1)	(1, 2, 2, 1)	(1, 3, 0, 0)	(1, 3, 1, 1)
(2, -1, -1, 0)	(2, -1, -1, 2)	(2, -1, 0, -1)	(2, -1, 0, 1)
(2, -1, 1, 0)	(2, -1, 1, 2)	(2, -1, 2, -1)	(2, -1, 2, 1)
(2, 0, -1, -1)	(2, 0, -1, 1)	(2, 0, 0, 0)	(2, 0, 0, 2)
(2, 0, 1, -1)	(2, 0, 1, 1)	(2, 0, 2, 0)	(2, 0, 2, 2)
(2, 1, -1, 0)	(2, 1, -1, 2)	(2, 1, 0, -1)	(2, 1, 0, 1)
(2, 1, 1, 0)	(2, 1, 1, 2)	(2, 1, 2, -1)	(2, 1, 2, 1)
(2, 2, -1, -1)	(2, 2, -1, 1)	(2, 2, 0, 0)	(2, 2, 0, 2)
(2, 2, 1, -1)	(2, 2, 1, 1)	(2, 2, 2, 0)	(2, 2, 2, 2)
(3, 0, 0, 1)	(3, 0, 1, 0)	(3, 1, 0, 0)	(3, 1, 1, 1)

II.4 - $\underline{A_2 \times A_2}$

Este reticulado é definido como o resultado do produto cartesiano do reticulado A_2 com ele mesmo, compondo portanto uma estrutura reticular em 4 dimensões gerada a partir de um reticulado hexagonal. A idéia de se trabalhar com esta estrutura de sinais, veio de aproveitar a densidade superior que a estrutura hexagonal (A_2) apresenta em relação à estrutura quadrada (Z^2), no sentido de averiguar se este ganho se perpetua em 4-D.

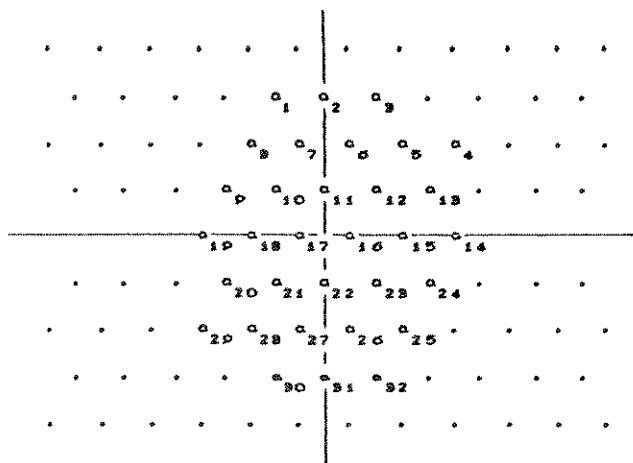
Apesar de não serem conhecidos os parâmetros que caracterizam a estrutura reticular $A_2 \times A_2$, é conhecido o processo de construção deste reticulado, e portanto utilizaremos os programas READA e DELTA0A, (Apêndice 2), para gerá-la e posteriormente obter o valor de A_0 .

Neste sentido,

$$A_4^0 = A_0 \times A_0$$

onde A_0 é ilustrado pela Figura II.4.1

A é portanto a constelação de 32 sinais tipo A_2 bem definida no Cap. 2, fig. II.iii-b, gerada em função de Δ no ponto-lacuna $(1/2, 0)$ do reticulado A_2



CONJUNTO A_0

$a_1 \rightarrow (-1.0000\Delta_0, 2.5980\Delta_0)$	$a_{17} \rightarrow (-0.5000\Delta_c, 0.0000\Delta_0)$
$a_2 \rightarrow (0.0000\Delta_0, 2.5980\Delta_0)$	$a_{18} \rightarrow (-1.5000\Delta_c, 0.0000\Delta_0)$
$a_3 \rightarrow (1.0000\Delta_0, 2.5980\Delta_0)$	$a_{19} \rightarrow (-2.5000\Delta_c, 0.0000\Delta_0)$
$a_4 \rightarrow (2.5000\Delta_0, 1.7320\Delta_0)$	$a_{20} \rightarrow (-2.0000\Delta_c, -0.8660\Delta_0)$
$a_5 \rightarrow (1.5000\Delta_0, 1.7320\Delta_0)$	$a_{21} \rightarrow (-1.0000\Delta_c, -0.8660\Delta_0)$
$a_6 \rightarrow (0.5000\Delta_0, 1.7320\Delta_0)$	$a_{22} \rightarrow (0.0000\Delta_c, -0.8660\Delta_0)$
$a_7 \rightarrow (-0.5000\Delta_0, 1.7320\Delta_0)$	$a_{23} \rightarrow (1.0000\Delta_c, -0.8660\Delta_0)$
$a_8 \rightarrow (-1.5000\Delta_0, 1.7320\Delta_0)$	$a_{24} \rightarrow (2.0000\Delta_c, -0.8660\Delta_0)$
$a_9 \rightarrow (-2.0000\Delta_0, 0.8660\Delta_0)$	$a_{25} \rightarrow (1.5000\Delta_c, -1.7320\Delta_0)$
$a_{10} \rightarrow (-1.0000\Delta_0, 0.8660\Delta_0)$	$a_{26} \rightarrow (0.5000\Delta_c, -1.7320\Delta_0)$
$a_{11} \rightarrow (0.0000\Delta_0, 0.8660\Delta_0)$	$a_{27} \rightarrow (-0.5000\Delta_c, -1.7320\Delta_0)$
$a_{12} \rightarrow (1.0000\Delta_0, 0.8660\Delta_0)$	$a_{28} \rightarrow (-1.5000\Delta_c, -1.7320\Delta_0)$
$a_{13} \rightarrow (2.0000\Delta_0, 0.8660\Delta_0)$	$a_{29} \rightarrow (-2.5000\Delta_c, -1.7320\Delta_0)$
$a_{14} \rightarrow (2.5000\Delta_0, 0.0000\Delta_0)$	$a_{30} \rightarrow (-1.0000\Delta_c, -2.5980\Delta_0)$
$a_{15} \rightarrow (1.5000\Delta_0, 0.0000\Delta_0)$	$a_{31} \rightarrow (0.0000\Delta_c, -2.5980\Delta_0)$
$a_{16} \rightarrow (0.5000\Delta_0, 0.0000\Delta_0)$	$a_{32} \rightarrow (1.0000\Delta_c, -2.5980\Delta_0)$

FIG. II.4.1: Const. de 32 sinais tipo A_2 (Cap. 2, Fig. II.iii-b)

Utilizando os programas READA e DELTAOA (Apêndice 2) os valores encontrados para centro de massa, energia média e Δ_0 para o reticulado $A_2 \times A_2$ são, respectivamente

$$\mathbb{M} = (0.0000, 0.0000, 0.0000, 0.0000)$$

$$\bar{s} \left\{ |a_n^2| \right\} = 8.87471 \Delta_0^2 = 1$$

$$\Delta_0 = 0.3357$$

III - TCM MULTI-DIMENSIONAL

Atualmente existe uma série de investigações em códigos de treliça com conjuntos de sinais definidos em mais de duas dimensões [12,13,14,15,16].

Na prática, sinais multi-dimensionais podem ser transmitidos como seqüências constituídas de sinais em uma ou duas dimensões (1-D ou 2-D). Neste Capítulo serão considerados esquemas TCM $2k$ -D ($k=2$) que podem transmitir m bits por componente de sinal bi-dimensional e portanto mk bits por sinal $2k$ -D.

Será mantido o princípio de se utilizar a redundância do conjunto de sinal duas vezes o tamanho necessário para a modulação não-codificada. Assim, esquemas TCM $2k$ -D utilizam conjuntos 2^{km+1} -ários de sinais $2k$ -D.

Os esquemas TCM $2k$ -D comparado aos esquemas TCM 2-D resultam em menor redundância de sinal por componente bi-dimensional. Nos esquemas TCM 2-D com conjuntos de sinais tipo Z^2 , seria preciso reduzir-se o espaçamento mínimo de sinal (Δ_0) pelo fator aproximado de $\sqrt{2}$ (-3 dB) para se obter a mesma potência média de sinal comparada a dos sistemas de modulação não-codificada. Esta perda em espaçamento de sinal precisa ser mais do que compensada pela codificação para se alcançar ganhos vantajosos em $d_{\min}$.

Os esquemas TCM multi-dimensionais projetados a partir de conjuntos de sinais do tipo " Z^{2k} ", por desfrutarem de uma menor

redundância de sinal, quando comparada aos esquemas TCM 2-D, contribuem para a redução do espaçamento mínimo de sinal pelo fator raiz- $2k$ de 2 (-1.5 dB para $k=2$) para se igualar aos sistemas não-codificados. Desta forma a codificação tem menos a contribuir do que no caso de TCM 2-D para obter o mesmo ganho em distância mínima.

III.1 - TCM EM 4 DIMENSÕES

Os esquemas TCM 4-D utilizam conjuntos de 2^{2m+1} sinais escolhidos de um reticulado Λ pertencente a $\mathbb{R}^4$ com distância mínima Δ_0 entre os mesmos.

Através da sequência de distâncias mínimas intra-conjuntos que procuraremos obter, poderá ser feita futuramente a procura por codificadores ótimos utilizando-se o Algoritmo estudado no Capítulo 3. Os ganhos em codificação neste caso serão válidos para grandes conjuntos de sinais que preencham o mesmo volume em espaço de sinal tal qual o espaço de sinal utilizado para modulação não-codificada. Assim, a comparação pode ser feita para a mesma potência média do sinal e para a mesma potência de pico.

A realização prática de um espaço de sinais de 4 dimensões pode ser feita pelo uso de dois campos elétricos polarizados ortogonalmente comunicando na mesma portadora de frequência. Também é possível considerar-se cada símbolo de 4 dimensões como dois símbolos consecutivos de 2 dimensões.

Com o objetivo de gerar estruturas reticulares em 4-D que sirvam como base para projeção dos conjuntos de sinais empregados no sistema TCM 4-D e da mesma forma comparar entre si estas estruturas será feito a seguir a partição de conjunto para os reticulados $\mathbb{Z}^4$ e $\Lambda_2 \times \Lambda_2$.

III.1.1 - PARTIÇÃO DE CONJUNTO DE Z^4

A figura III.1.1.1 ilustra a partição de conjunto de um conjunto A_4^0 do tipo Z^4 . A idéia geral consiste em se derivar a partição de conjunto do conjunto de sinais em 4 dimensões a partir da partição dos conjuntos componentes de dimensão inferior [5].

Neste caso, A_4^0 e seus subconjuntos ($B_4^0, B_4^1, C_4^0, C_4^1, \dots$) são caracterizados por duas componentes de conjunto de sinais tipo Z^2 — A_0 e seus subconjuntos $B_0, B_1, C_0, C_1, C_2, C_3, \dots$ — tais como introduzidos nos capítulos anteriores.

Isto nos leva a uma árvore de partição com conjuntos de sinais do tipo $Z^4 \rightarrow D_4 \rightarrow Z^4 \rightarrow D_4 \rightarrow Z^4$, etc. Com distância mínima intra-conjunto $\Delta_0, \Delta_1 = \Delta_2 = \sqrt{2} \Delta_0, \Delta_3 = \Delta_4 = \sqrt{4} \Delta_0$, etc. Serão descritos a seguir os detalhes do processo de partição:

A_4^0 é definido como sendo o produto cartesiano de A_0 com A_0 .

A substituição de A_0 por $B_0 \cup B_1$ leva a $A_4^0 = (B_0 \times B_1) \cup (B_1 \times B_0)$ e da mesma forma:

$$A_4^0 = (B_0 \times B_0) \cup (B_0 \times B_1) \cup (B_1 \times B_0) \cup (B_1 \times B_1)$$

A primeira partição divide A_4^0 em dois subconjuntos:

$$B_4^0 = [B_0 \times B_0] \cup [B_1 \times B_1] \text{ e}$$

$$B_4^1 = [B_0 \times B_1] \cup [B_1 \times B_0]$$

respectivamente.

Estes subconjuntos são do tipo D_4 , onde D_4 é o reticulado mais denso no espaço 4-D [7]. A distância mínima intra-conjunto em B_4^0 e B_4^1 vale $\sqrt{2}\Delta_0$, que é a distância mínima entre as componentes de sinais 2-D em B_0 ou B_1 , e também entre um sinal 4-D em $B_0 \times B_0$ e um outro em $B_1 \times B_1$. Na próxima partição binária, a saber, quando B_4^0 é particionado nos subconjuntos $B_0 \times B_0$ e $B_1 \times B_1$ não é alcançado nenhum acréscimo em termos de distância Euclideana. Estes subconjuntos são do tipo Z^4 , como em A_4^0 , de onde eles diferem somente em sua orientação, posição com respeito à origem, e escala. Portanto, a sua partição é conceitualmente similar à de A_4^0 . A distância mínima intra-conjunto aumenta para $\sqrt{4}\Delta_0$ quando $B_0 \times B_0$ é dividido nos subconjuntos

$$C_4^0 = [C_0 \times C_0] \cup [C_2 \times C_2] \text{ e } C_4^1 = [C_0 \times C_2] \cup [C_2 \times C_0].$$

Note que estes conjuntos formam novamente um reticulado D_4 .

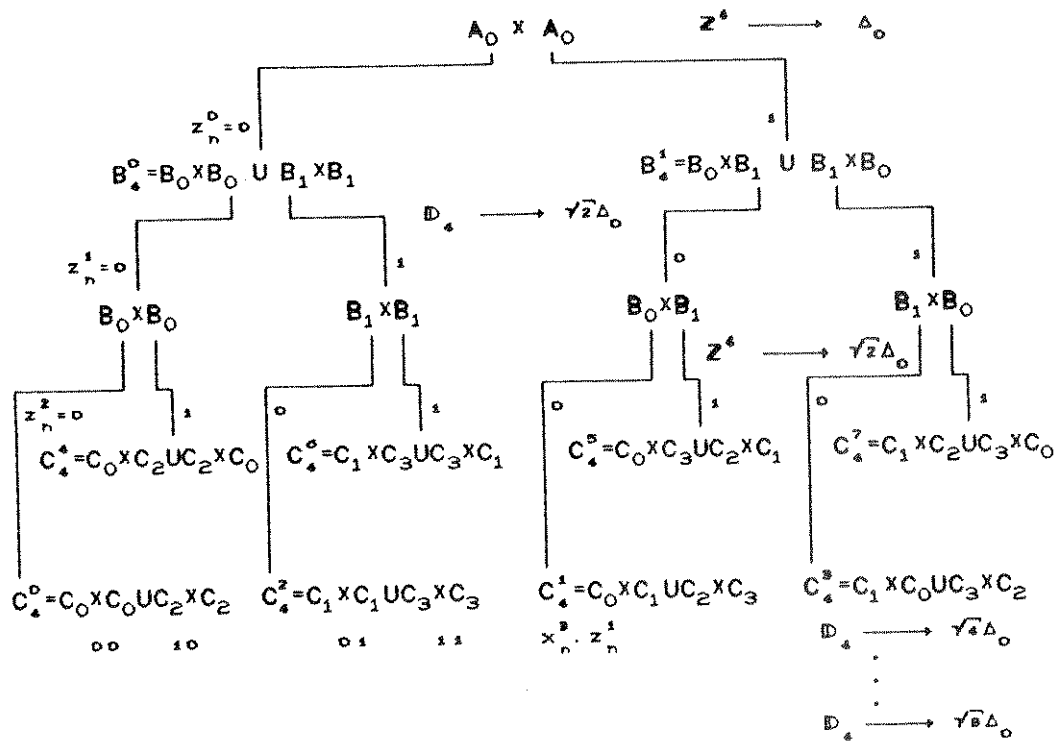


Fig. III.1.1.1: Partição de conjunto para sinais em reticulados tipo Z^4 . A operação produto cartesiano é denotada "x", e "U" denota união de conjuntos.

Partição de conjunto - Reticulado tipo $\mathbb{Z}^2$
 $\chi(\lambda\lambda\lambda) \ m = 32$ sinais

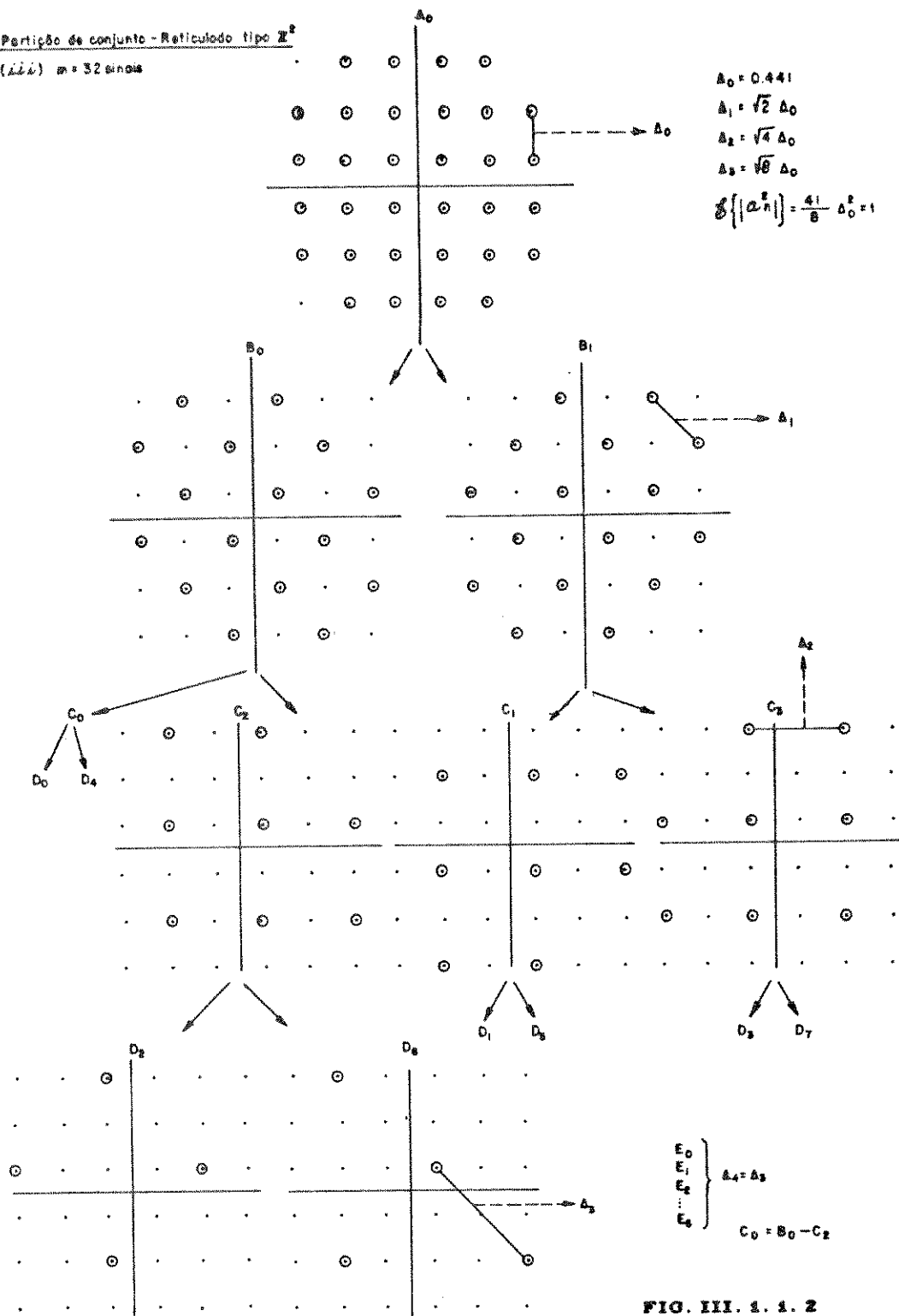


FIG. III. 1. 1. 2

Para efeito de visualização do processo, iremos criar o subconjunto C_4^7 com base nos subconjuntos C_0 , C_1 , C_2 e C_3 pertencentes à Figura III.1.1.2, bem caracterizada no Cap. 2, e calcular a distância mínima intraconjunto em função Δ_0 para este nível da partição de A_4^0 .

Para tanto, foi criado um programa em linguagem Pascal (Apêndice 2 - READC e DELTAOC). O Algoritmo gera os subconjuntos C_1 e realiza as operações de produto cartesiano e união dos conjuntos de acordo com a composição de C_4^j ($j=0...7$). Para o cálculo da distância mínima neste nível da partição é necessário que os pontos pertencentes à este subconjunto sejam projetados com coordenadas quadri-dimensionais em função de Δ_0 . Este cuidado foi tomado no momento da construção de A_4^0 e o valor utilizado para Δ_0 foi obtido em DELTAOA — $\Delta_0 = 0.3162$.

Os subconjuntos C_0 , C_1 , C_2 e C_3 de sinais tipo Z^2 são caracterizados respectivamente pelos pontos

$$\left\{ \alpha_{10}, \alpha_8, \alpha_6, \alpha_{22}, \alpha_{20}, \alpha_{18}, \alpha_{31}, \alpha_{29} \right\}, \left\{ \alpha_9, \alpha_7, \alpha_5, \alpha_{21}, \alpha_{19}, \alpha_{17}, \alpha_{32}, \alpha_{30} \right\},$$

$$\left\{ \alpha_1, \alpha_4, \alpha_{11}, \alpha_{19}, \alpha_{15}, \alpha_{29}, \alpha_{25}, \alpha_{27} \right\} \text{ e } \left\{ \alpha_2, \alpha_4, \alpha_{11}, \alpha_{19}, \alpha_{15}, \alpha_{29}, \alpha_{25}, \alpha_{27} \right\}$$

com coordenadas obtidas em função de Δ_0 e ordenados à partir da Figura II.1.1

O Algoritmo lê os subconjuntos C_1 , lista o resultado da construção de C_4^7 em função de Δ_0 anteriormente calculado, e calcula a menor das distâncias intraconjuntos para este nível da partição. Vejamos estes resultados:

Delta 0 = 3.1620000000E-01

SUBCONJUNTO C_0

1 => (-0.7905, 0.4743 >
 2 => (-0.1581, 0.4743 >
 3 => (0.4743, 0.4743 >
 4 => (-0.7905,-0.1581 >
 5 => (-0.1581,-0.1581 >
 6 => (0.4743,-0.1581 >
 7 => (-0.1581,-0.7905 >
 8 => (0.4743,-0.7905 >

SUBCONJUNTO C_1

1 => (-0.4743, 0.4743 >
 2 => (0.1581, 0.4743 >
 3 => (0.7905, 0.4743 >
 4 => (-0.4743,-0.1581 >
 5 => (0.1581,-0.1581 >
 6 => (0.7905,-0.1581 >
 7 => (-0.4743,-0.7905 >
 8 => (0.1581,-0.7905 >

SUBCONJUNTO C_2

1 => (-0.4743, 0.7905 >
 2 => (0.1581, 0.7905 >
 3 => (-0.4743, 0.1581 >
 4 => (0.1581, 0.1581 >
 5 => (0.7905, 0.1581 >
 6 => (-0.4743,-0.4743 >
 7 => (0.1581,-0.4743 >
 8 => (0.7905,-0.4743 >

SUBCONJUNTO C_3

1 => (-0.1581, 0.7905 >
 2 => (0.4743, 0.7905 >
 3 => (-0.7905, 0.1581 >
 4 => (-0.1581, 0.1581 >
 5 => (0.4743, 0.1581 >
 6 => (-0.7905,-0.4743 >
 7 => (-0.1581,-0.4743 >
 8 => (0.4743,-0.4743 >

SUBCONJUNTO C_4^7 DA PARTIÇÃO DE Z^4

- 1 $\Rightarrow$ (-0.4743, 0.4743, -0.4743, 0.7905)
- 2 $\Rightarrow$ (-0.4743, 0.4743, 0.1581, 0.7905)
- 3 $\Rightarrow$ (-0.4743, 0.4743, -0.4743, 0.1581)
- 4 $\Rightarrow$ (-0.4743, 0.4743, 0.1581, 0.1581)
- 5 $\Rightarrow$ (-0.4743, 0.4743, 0.7905, 0.1581)
- 6 $\Rightarrow$ (-0.4743, 0.4743, -0.4743, -0.4743)
- 7 $\Rightarrow$ (-0.4743, 0.4743, 0.1581, -0.4743)
- 8 $\Rightarrow$ (-0.4743, 0.4743, 0.7905, -0.4743)
- 9 $\Rightarrow$ (0.1581, 0.4743, -0.4743, 0.7905)
- 10 $\Rightarrow$ (0.1581, 0.4743, 0.1581, 0.7905)
- 11 $\Rightarrow$ (0.1581, 0.4743, -0.4743, 0.1581)
- 12 $\Rightarrow$ (0.1581, 0.4743, 0.1581, 0.1581)
- 13 $\Rightarrow$ (0.1581, 0.4743, 0.7905, 0.1581)
- 14 $\Rightarrow$ (0.1581, 0.4743, -0.4743, -0.4743)
- 15 $\Rightarrow$ (0.1581, 0.4743, 0.1581, -0.4743)
- 16 $\Rightarrow$ (0.1581, 0.4743, 0.7905, -0.4743)
- 17 $\Rightarrow$ (0.7905, 0.4743, -0.4743, 0.7905)
- 18 $\Rightarrow$ (0.7905, 0.4743, 0.1581, 0.7905)
- 19 $\Rightarrow$ (0.7905, 0.4743, -0.4743, 0.1581)
- 20 $\Rightarrow$ (0.7905, 0.4743, 0.1581, 0.1581)
- 21 $\Rightarrow$ (0.7905, 0.4743, 0.7905, 0.1581)
- 22 $\Rightarrow$ (0.7905, 0.4743, -0.4743, -0.4743)
- 23 $\Rightarrow$ (0.7905, 0.4743, 0.1581, -0.4743)
- 24 $\Rightarrow$ (0.7905, 0.4743, 0.7905, -0.4743)
- 25 $\Rightarrow$ (-0.4743, -0.1581, -0.4743, 0.7905)
- 26 $\Rightarrow$ (-0.4743, -0.1581, 0.1581, 0.7905)

27 => (-0.4743, -0.1581, -0.4743, 0.1581)
28 => (-0.4743, -0.1581, 0.1581, 0.1581)
29 => (-0.4743, -0.1581, 0.7905, 0.1581)
30 => (-0.4743, -0.1581, -0.4743, -0.4743)
31 => (-0.4743, -0.1581, 0.1581, -0.4743)
32 => (-0.4743, -0.1581, 0.7905, -0.4743)
33 => (0.1581, -0.1581, -0.4743, 0.7905)
34 => (0.1581, -0.1581, 0.1581, 0.7905)
35 => (0.1581, -0.1581, -0.4743, 0.1581)
36 => (0.1581, -0.1581, 0.1581, 0.1581)
37 => (0.1581, -0.1581, 0.7905, 0.1581)
38 => (0.1581, -0.1581, -0.4743, -0.4743)
39 => (0.1581, -0.1581, 0.1581, -0.4743)
40 => (0.1581, -0.1581, 0.7905, -0.4743)
41 => (0.7905, -0.1581, -0.4743, 0.7905)
42 => (0.7905, -0.1581, 0.1581, 0.7905)
43 => (0.7905, -0.1581, -0.4743, 0.1581)
44 => (0.7905, -0.1581, 0.1581, 0.1581)
45 => (0.7905, -0.1581, 0.7905, 0.1581)
46 => (0.7905, -0.1581, -0.4743, -0.4743)
47 => (0.7905, -0.1581, 0.1581, -0.4743)
48 => (0.7905, -0.1581, 0.7905, -0.4743)
49 => (-0.4743, -0.7905, -0.4743, 0.7905)
50 => (-0.4743, -0.7905, 0.1581, 0.7905)
51 => (-0.4743, -0.7905, -0.4743, 0.1581)
52 => (-0.4743, -0.7905, 0.1581, 0.1581)
53 => (-0.4743, -0.7905, 0.7905, 0.1581)
54 => (-0.4743, -0.7905, -0.4743, -0.4743)
55 => (-0.4743, -0.7905, 0.1581, -0.4743)

56 => (-0.4743, -0.7905, 0.7905, -0.4743)
57 => (0.1581, -0.7905, -0.4743, 0.7905)
58 => (0.1581, -0.7905, 0.1581, 0.7905)
59 => (0.1581, -0.7905, -0.4743, 0.1581)
60 => (0.1581, -0.7905, 0.1581, 0.1581)
61 => (0.1581, -0.7905, 0.7905, 0.1581)
62 => (0.1581, -0.7905, -0.4743, -0.4743)
63 => (0.1581, -0.7905, 0.1581, -0.4743)
64 => (0.1581, -0.7905, 0.7905, -0.4743)
65 => (-0.1581, 0.7905, -0.7905, 0.4743)
66 => (-0.1581, 0.7905, -0.1581, 0.4743)
67 => (-0.1581, 0.7905, 0.4743, 0.4743)
68 => (-0.1581, 0.7905, -0.7905, -0.1581)
69 => (-0.1581, 0.7905, -0.1581, -0.1581)
70 => (-0.1581, 0.7905, 0.4743, -0.1581)
71 => (-0.1581, 0.7905, -0.1581, -0.7905)
72 => (-0.1581, 0.7905, 0.4743, -0.7905)
73 => (0.4743, 0.7905, -0.7905, 0.4743)
74 => (0.4743, 0.7905, -0.1581, 0.4743)
75 => (0.4743, 0.7905, 0.4743, 0.4743)
76 => (0.4743, 0.7905, -0.7905, -0.1581)
77 => (0.4743, 0.7905, -0.1581, -0.1581)
78 => (0.4743, 0.7905, 0.4743, -0.1581)
79 => (0.4743, 0.7905, -0.1581, -0.7905)
80 => (0.4743, 0.7905, 0.4743, -0.7905)
81 => (-0.7905, 0.1581, -0.7905, 0.4743)
82 => (-0.7905, 0.1581, -0.1581, 0.4743)
83 => (-0.7905, 0.1581, 0.4743, 0.4743)
84 => (-0.7905, 0.1581, -0.7905, -0.1581)

85 => (-0.7905, 0.1581, -0.1581, -0.1581)
86 => (-0.7905, 0.1581, 0.4743, -0.1581)
87 => (-0.7905, 0.1581, -0.1581, -0.7905)
88 => (-0.7905, 0.1581, 0.4743, -0.7905)
89 => (-0.1581, 0.1581, -0.7905, 0.4743)
90 => (-0.1581, 0.1581, -0.1581, 0.4743)
91 => (-0.1581, 0.1581, 0.4743, 0.4743)
92 => (-0.1581, 0.1581, -0.7905, -0.1581)
93 => (-0.1581, 0.1581, -0.1581, -0.1581)
94 => (-0.1581, 0.1581, 0.4743, -0.1581)
95 => (-0.1581, 0.1581, -0.1581, -0.7905)
96 => (-0.1581, 0.1581, 0.4743, -0.7905)
97 => (0.4743, 0.1581, -0.7905, 0.4743)
98 => (0.4743, 0.1581, -0.1581, 0.4743)
99 => (0.4743, 0.1581, 0.4743, 0.4743)
100 => (0.4743, 0.1581, -0.7905, -0.1581)
101 => (0.4743, 0.1581, -0.1581, -0.1581)
102 => (0.4743, 0.1581, 0.4743, -0.1581)
103 => (0.4743, 0.1581, -0.1581, -0.7905)
104 => (0.4743, 0.1581, 0.4743, -0.7905)
105 => (-0.7905, -0.4743, -0.7905, 0.4743)
106 => (-0.7905, -0.4743, -0.1581, 0.4743)
107 => (-0.7905, -0.4743, 0.4743, 0.4743)
108 => (-0.7905, -0.4743, -0.7905, -0.1581)
109 => (-0.7905, -0.4743, -0.1581, -0.1581)
110 => (-0.7905, -0.4743, 0.4743, -0.1581)
111 => (-0.7905, -0.4743, -0.1581, -0.7905)
112 => (-0.7905, -0.4743, 0.4743, -0.7905)
113 => (-0.1581, -0.4743, -0.7905, 0.4743)

114 => (-0.1581, -0.4743, -0.1581, 0.4743)
115 => (-0.1581, -0.4743, 0.4743, 0.4743)
116 => (-0.1581, -0.4743, -0.7905, -0.1581)
117 => (-0.1581, -0.4743, -0.1581, -0.1581)
118 => (-0.1581, -0.4743, 0.4743, -0.1581)
119 => (-0.1581, -0.4743, -0.1581, -0.7905)
120 => (-0.1581, -0.4743, 0.4743, -0.7905)
121 => (0.4743, -0.4743, -0.7905, 0.4743)
122 => (0.4743, -0.4743, -0.1581, 0.4743)
123 => (0.4743, -0.4743, 0.4743, 0.4743)
124 => (0.4743, -0.4743, -0.7905, -0.1581)
125 => (0.4743, -0.4743, -0.1581, -0.1581)
126 => (0.4743, -0.4743, 0.4743, -0.1581)
127 => (0.4743, -0.4743, -0.1581, -0.7905)
128 => (0.4743, -0.4743, 0.4743, -0.7905)

Cálculo das distâncias de cada subconjunto

Distancia minima em C04.dat = 0.6324
Distancia minima em C14.dat = 0.6324
Distancia minima em C24.dat = 0.6324
Distancia minima em C34.dat = 0.6324
Distancia minima em C44.dat = 0.6324
Distancia minima em C54.dat = 0.6324
Distancia minima em C64.dat = 0.6324
Distancia minima em C74.dat = 0.6324

III.1.2 - PARTIÇÃO DE CONJUNTO DE $A_2 \times A_2$

A Figura III.1.1.4 pode de maneira análoga ilustrar a partição de conjunto para um conjunto de sinais do tipo $A_2 \times A_2$. Basta que para isto, o conjunto inicial da partição — A_4^0 — seja gerado a partir do produto cartesiano de um conjunto de sinais do tipo A_2 consigo próprio. A árvore de partição será gerada a partir da partição dos conjuntos componentes de dimensão inferior.

Neste caso, A_4^0 e seus subconjuntos ($B_4^0, B_4^1, C_4^0, C_4^1, \dots$) são caracterizados por duas componentes de conjunto de sinais tipo A_2 — A_0 e seus subconjuntos $B_0, B_1, C_0, C_1, C_2, C_3, \dots$ — tais como introduzidos anteriormente no Capítulo 2 e ilustrado pela Figura III.1.2.1.

Os subconjuntos C_0, C_1, C_2 e C_3 de sinais tipo A_2 são caracterizados respectivamente pelos pontos

$$\left\{ a_1, a_9, a_{17}, a_{25}, a_{33}, a_{41}, a_{49}, a_{57}, a_{65}, a_{73}, a_{81}, a_{89}, a_{97}, a_{105}, a_{113}, a_{121}, a_{129}, a_{137}, a_{145}, a_{153}, a_{161}, a_{169}, a_{177}, a_{185}, a_{193}, a_{201}, a_{209}, a_{217}, a_{225}, a_{233}, a_{241}, a_{249}, a_{257}, a_{265}, a_{273}, a_{281}, a_{289}, a_{297}, a_{305}, a_{313}, a_{321}, a_{329}, a_{337}, a_{345}, a_{353}, a_{361}, a_{369}, a_{377}, a_{385}, a_{393}, a_{401}, a_{409}, a_{417}, a_{425}, a_{433}, a_{441}, a_{449}, a_{457}, a_{465}, a_{473}, a_{481}, a_{489}, a_{497}, a_{505}, a_{513}, a_{521}, a_{529}, a_{537}, a_{545}, a_{553}, a_{561}, a_{569}, a_{577}, a_{585}, a_{593}, a_{601}, a_{609}, a_{617}, a_{625}, a_{633}, a_{641}, a_{649}, a_{657}, a_{665}, a_{673}, a_{681}, a_{689}, a_{697}, a_{705}, a_{713}, a_{721}, a_{729}, a_{737}, a_{745}, a_{753}, a_{761}, a_{769}, a_{777}, a_{785}, a_{793}, a_{801}, a_{809}, a_{817}, a_{825}, a_{833}, a_{841}, a_{849}, a_{857}, a_{865}, a_{873}, a_{881}, a_{889}, a_{897}, a_{905}, a_{913}, a_{921}, a_{929}, a_{937}, a_{945}, a_{953}, a_{961}, a_{969}, a_{977}, a_{985}, a_{993}, a_{1001}, a_{1009}, a_{1017}, a_{1025}, a_{1033}, a_{1041}, a_{1049}, a_{1057}, a_{1065}, a_{1073}, a_{1081}, a_{1089}, a_{1097}, a_{1105}, a_{1113}, a_{1121}, a_{1129}, a_{1137}, a_{1145}, a_{1153}, a_{1161}, a_{1169}, a_{1177}, a_{1185}, a_{1193}, a_{1201}, a_{1209}, a_{1217}, a_{1225}, a_{1233}, a_{1241}, a_{1249}, a_{1257}, a_{1265}, a_{1273}, a_{1281}, a_{1289}, a_{1297}, a_{1305}, a_{1313}, a_{1321}, a_{1329}, a_{1337}, a_{1345}, a_{1353}, a_{1361}, a_{1369}, a_{1377}, a_{1385}, a_{1393}, a_{1401}, a_{1409}, a_{1417}, a_{1425}, a_{1433}, a_{1441}, a_{1449}, a_{1457}, a_{1465}, a_{1473}, a_{1481}, a_{1489}, a_{1497}, a_{1505}, a_{1513}, a_{1521}, a_{1529}, a_{1537}, a_{1545}, a_{1553}, a_{1561}, a_{1569}, a_{1577}, a_{1585}, a_{1593}, a_{1601}, a_{1609}, a_{1617}, a_{1625}, a_{1633}, a_{1641}, a_{1649}, a_{1657}, a_{1665}, a_{1673}, a_{1681}, a_{1689}, a_{1697}, a_{1705}, a_{1713}, a_{1721}, a_{1729}, a_{1737}, a_{1745}, a_{1753}, a_{1761}, a_{1769}, a_{1777}, a_{1785}, a_{1793}, a_{1801}, a_{1809}, a_{1817}, a_{1825}, a_{1833}, a_{1841}, a_{1849}, a_{1857}, a_{1865}, a_{1873}, a_{1881}, a_{1889}, a_{1897}, a_{1905}, a_{1913}, a_{1921}, a_{1929}, a_{1937}, a_{1945}, a_{1953}, a_{1961}, a_{1969}, a_{1977}, a_{1985}, a_{1993}, a_{2001}, a_{2009}, a_{2017}, a_{2025}, a_{2033}, a_{2041}, a_{2049}, a_{2057}, a_{2065}, a_{2073}, a_{2081}, a_{2089}, a_{2097}, a_{2105}, a_{2113}, a_{2121}, a_{2129}, a_{2137}, a_{2145}, a_{2153}, a_{2161}, a_{2169}, a_{2177}, a_{2185}, a_{2193}, a_{2201}, a_{2209}, a_{2217}, a_{2225}, a_{2233}, a_{2241}, a_{2249}, a_{2257}, a_{2265}, a_{2273}, a_{2281}, a_{2289}, a_{2297}, a_{2305}, a_{2313}, a_{2321}, a_{2329}, a_{2337}, a_{2345}, a_{2353}, a_{2361}, a_{2369}, a_{2377}, a_{2385}, a_{2393}, a_{2401}, a_{2409}, a_{2417}, a_{2425}, a_{2433}, a_{2441}, a_{2449}, a_{2457}, a_{2465}, a_{2473}, a_{2481}, a_{2489}, a_{2497}, a_{2505}, a_{2513}, a_{2521}, a_{2529}, a_{2537}, a_{2545}, a_{2553}, a_{2561}, a_{2569}, a_{2577}, a_{2585}, a_{2593}, a_{2601}, a_{2609}, a_{2617}, a_{2625}, a_{2633}, a_{2641}, a_{2649}, a_{2657}, a_{2665}, a_{2673}, a_{2681}, a_{2689}, a_{2697}, a_{2705}, a_{2713}, a_{2721}, a_{2729}, a_{2737}, a_{2745}, a_{2753}, a_{2761}, a_{2769}, a_{2777}, a_{2785}, a_{2793}, a_{2801}, a_{2809}, a_{2817}, a_{2825}, a_{2833}, a_{2841}, a_{2849}, a_{2857}, a_{2865}, a_{2873}, a_{2881}, a_{2889}, a_{2897}, a_{2905}, a_{2913}, a_{2921}, a_{2929}, a_{2937}, a_{2945}, a_{2953}, a_{2961}, a_{2969}, a_{2977}, a_{2985}, a_{2993}, a_{3001}, a_{3009}, a_{3017}, a_{3025}, a_{3033}, a_{3041}, a_{3049}, a_{3057}, a_{3065}, a_{3073}, a_{3081}, a_{3089}, a_{3097}, a_{3105}, a_{3113}, a_{3121}, a_{3129}, a_{3137}, a_{3145}, a_{3153}, a_{3161}, a_{3169}, a_{3177}, a_{3185}, a_{3193}, a_{3201}, a_{3209}, a_{3217}, a_{3225}, a_{3233}, a_{3241}, a_{3249}, a_{3257}, a_{3265}, a_{3273}, a_{3281}, a_{3289}, a_{3297}, a_{3305}, a_{3313}, a_{3321}, a_{3329}, a_{3337}, a_{3345}, a_{3353}, a_{3361}, a_{3369}, a_{3377}, a_{3385}, a_{3393}, a_{3401}, a_{3409}, a_{3417}, a_{3425}, a_{3433}, a_{3441}, a_{3449}, a_{3457}, a_{3465}, a_{3473}, a_{3481}, a_{3489}, a_{3497}, a_{3505}, a_{3513}, a_{3521}, a_{3529}, a_{3537}, a_{3545}, a_{3553}, a_{3561}, a_{3569}, a_{3577}, a_{3585}, a_{3593}, a_{3601}, a_{3609}, a_{3617}, a_{3625}, a_{3633}, a_{3641}, a_{3649}, a_{3657}, a_{3665}, a_{3673}, a_{3681}, a_{3689}, a_{3697}, a_{3705}, a_{3713}, a_{3721}, a_{3729}, a_{3737}, a_{3745}, a_{3753}, a_{3761}, a_{3769}, a_{3777}, a_{3785}, a_{3793}, a_{3801}, a_{3809}, a_{3817}, a_{3825}, a_{3833}, a_{3841}, a_{3849}, a_{3857}, a_{3865}, a_{3873}, a_{3881}, a_{3889}, a_{3897}, a_{3905}, a_{3913}, a_{3921}, a_{3929}, a_{3937}, a_{3945}, a_{3953}, a_{3961}, a_{3969}, a_{3977}, a_{3985}, a_{3993}, a_{4001}, a_{4009}, a_{4017}, a_{4025}, a_{4033}, a_{4041}, a_{4049}, a_{4057}, a_{4065}, a_{4073}, a_{4081}, a_{4089}, a_{4097}, a_{4105}, a_{4113}, a_{4121}, a_{4129}, a_{4137}, a_{4145}, a_{4153}, a_{4161}, a_{4169}, a_{4177}, a_{4185}, a_{4193}, a_{4201}, a_{4209}, a_{4217}, a_{4225}, a_{4233}, a_{4241}, a_{4249}, a_{4257}, a_{4265}, a_{4273}, a_{4281}, a_{4289}, a_{4297}, a_{4305}, a_{4313}, a_{4321}, a_{4329}, a_{4337}, a_{4345}, a_{4353}, a_{4361}, a_{4369}, a_{4377}, a_{4385}, a_{4393}, a_{4401}, a_{4409}, a_{4417}, a_{4425}, a_{4433}, a_{4441}, a_{4449}, a_{4457}, a_{4465}, a_{4473}, a_{4481}, a_{4489}, a_{4497}, a_{4505}, a_{4513}, a_{4521}, a_{4529}, a_{4537}, a_{4545}, a_{4553}, a_{4561}, a_{4569}, a_{4577}, a_{4585}, a_{4593}, a_{4601}, a_{4609}, a_{4617}, a_{4625}, a_{4633}, a_{4641}, a_{4649}, a_{4657}, a_{4665}, a_{4673}, a_{4681}, a_{4689}, a_{4697}, a_{4705}, a_{4713}, a_{4721}, a_{4729}, a_{4737}, a_{4745}, a_{4753}, a_{4761}, a_{4769}, a_{4777}, a_{4785}, a_{4793}, a_{4801}, a_{4809}, a_{4817}, a_{4825}, a_{4833}, a_{4841}, a_{4849}, a_{4857}, a_{4865}, a_{4873}, a_{4881}, a_{4889}, a_{4897}, a_{4905}, a_{4913}, a_{4921}, a_{4929}, a_{4937}, a_{4945}, a_{4953}, a_{4961}, a_{4969}, a_{4977}, a_{4985}, a_{4993}, a_{5001}, a_{5009}, a_{5017}, a_{5025}, a_{5033}, a_{5041}, a_{5049}, a_{5057}, a_{5065}, a_{5073}, a_{5081}, a_{5089}, a_{5097}, a_{5105}, a_{5113}, a_{5121}, a_{5129}, a_{5137}, a_{5145}, a_{5153}, a_{5161}, a_{5169}, a_{5177}, a_{5185}, a_{5193}, a_{5201}, a_{5209}, a_{5217}, a_{5225}, a_{5233}, a_{5241}, a_{5249}, a_{5257}, a_{5265}, a_{5273}, a_{5281}, a_{5289}, a_{5297}, a_{5305}, a_{5313}, a_{5321}, a_{5329}, a_{5337}, a_{5345}, a_{5353}, a_{5361}, a_{5369}, a_{5377}, a_{5385}, a_{5393}, a_{5401}, a_{5409}, a_{5417}, a_{5425}, a_{5433}, a_{5441}, a_{5449}, a_{5457}, a_{5465}, a_{5473}, a_{5481}, a_{5489}, a_{5497}, a_{5505}, a_{5513}, a_{5521}, a_{5529}, a_{5537}, a_{5545}, a_{5553}, a_{5561}, a_{5569}, a_{5577}, a_{5585}, a_{5593}, a_{5601}, a_{5609}, a_{5617}, a_{5625}, a_{5633}, a_{5641}, a_{5649}, a_{5657}, a_{5665}, a_{5673}, a_{5681}, a_{5689}, a_{5697}, a_{5705}, a_{5713}, a_{5721}, a_{5729}, a_{5737}, a_{5745}, a_{5753}, a_{5761}, a_{5769}, a_{5777}, a_{5785}, a_{5793}, a_{5801}, a_{5809}, a_{5817}, a_{5825}, a_{5833}, a_{5841}, a_{5849}, a_{5857}, a_{5865}, a_{5873}, a_{5881}, a_{5889}, a_{5897}, a_{5905}, a_{5913}, a_{5921}, a_{5929}, a_{5937}, a_{5945}, a_{5953}, a_{5961}, a_{5969}, a_{5977}, a_{5985}, a_{5993}, a_{6001}, a_{6009}, a_{6017}, a_{6025}, a_{6033}, a_{6041}, a_{6049}, a_{6057}, a_{6065}, a_{6073}, a_{6081}, a_{6089}, a_{6097}, a_{6105}, a_{6113}, a_{6121}, a_{6129}, a_{6137}, a_{6145}, a_{6153}, a_{6161}, a_{6169}, a_{6177}, a_{6185}, a_{6193}, a_{6201}, a_{6209}, a_{6217}, a_{6225}, a_{6233}, a_{6241}, a_{6249}, a_{6257}, a_{6265}, a_{6273}, a_{6281}, a_{6289}, a_{6297}, a_{6305}, a_{6313}, a_{6321}, a_{6329}, a_{6337}, a_{6345}, a_{6353}, a_{6361}, a_{6369}, a_{6377}, a_{6385}, a_{6393}, a_{6401}, a_{6409}, a_{6417}, a_{6425}, a_{6433}, a_{6441}, a_{6449}, a_{6457}, a_{6465}, a_{6473}, a_{6481}, a_{6489}, a_{6497}, a_{6505}, a_{6513}, a_{6521}, a_{6529}, a_{6537}, a_{6545}, a_{6553}, a_{6561}, a_{6569}, a_{6577}, a_{6585}, a_{6593}, a_{6601}, a_{6609}, a_{6617}, a_{6625}, a_{6633}, a_{6641}, a_{6649}, a_{6657}, a_{6665}, a_{6673}, a_{6681}, a_{6689}, a_{6697}, a_{6705}, a_{6713}, a_{6721}, a_{6729}, a_{6737}, a_{6745}, a_{6753}, a_{6761}, a_{6769}, a_{6777}, a_{6785}, a_{6793}, a_{6801}, a_{6809}, a_{6817}, a_{6825}, a_{6833}, a_{6841}, a_{6849}, a_{6857}, a_{6865}, a_{6873}, a_{6881}, a_{6889}, a_{6897}, a_{6905}, a_{6913}, a_{6921}, a_{6929}, a_{6937}, a_{6945}, a_{6953}, a_{6961}, a_{6969}, a_{6977}, a_{6985}, a_{6993}, a_{7001}, a_{7009}, a_{7017}, a_{7025}, a_{7033}, a_{7041}, a_{7049}, a_{7057}, a_{7065}, a_{7073}, a_{7081}, a_{7089}, a_{7097}, a_{7105}, a_{7113}, a_{7121}, a_{7129}, a_{7137}, a_{7145}, a_{7153}, a_{7161}, a_{7169}, a_{7177}, a_{7185}, a_{7193}, a_{7201}, a_{7209}, a_{7217}, a_{7225}, a_{7233}, a_{7241}, a_{7249}, a_{7257}, a_{7265}, a_{7273}, a_{7281}, a_{7289}, a_{7297}, a_{7305}, a_{7313}, a_{7321}, a_{7329}, a_{7337}, a_{7345}, a_{7353}, a_{7361}, a_{7369}, a_{7377}, a_{7385}, a_{7393}, a_{7401}, a_{7409}, a_{7417}, a_{7425}, a_{7433}, a_{7441}, a_{7449}, a_{7457}, a_{7465}, a_{7473}, a_{7481}, a_{7489}, a_{7497}, a_{7505}, a_{7513}, a_{7521}, a_{7529}, a_{7537}, a_{7545}, a_{7553}, a_{7561}, a_{7569}, a_{7577}, a_{7585}, a_{7593}, a_{7601}, a_{7609}, a_{7617}, a_{7625}, a_{7633}, a_{7641}, a_{7649}, a_{7657}, a_{7665}, a_{7673}, a_{7681}, a_{7689}, a_{7697}, a_{7705}, a_{7713}, a_{7721}, a_{7729}, a_{7737}, a_{7745}, a_{7753}, a_{7761}, a_{7769}, a_{7777}, a_{7785}, a_{7793}, a_{7801}, a_{7809}, a_{7817}, a_{7825}, a_{7833}, a_{7841}, a_{7849}, a_{7857}, a_{7865}, a_{7873}, a_{7881}, a_{7889}, a_{7897}, a_{7905}, a_{7913}, a_{7921}, a_{7929}, a_{7937}, a_{7945}, a_{7953}, a_{7961}, a_{7969}, a_{7977}, a_{7985}, a_{7993}, a_{8001}, a_{8009}, a_{8017}, a_{8025}, a_{8033}, a_{8041}, a_{8049}, a_{8057}, a_{8065}, a_{8073}, a_{8081}, a_{8089}, a_{8097}, a_{8105}, a_{8113}, a_{8121}, a_{8129}, a_{8137}, a_{8145}, a_{8153}, a_{8161}, a_{8169}, a_{8177}, a_{8185}, a_{8193}, a_{8201}, a_{8209}, a_{8217}, a_{8225}, a_{8233}, a_{8241}, a_{8249}, a_{8257}, a_{8265}, a_{8273}, a_{8281}, a_{8289}, a_{8297}, a_{8305}, a_{8313}, a_{8321}, a_{8329}, a_{8337}, a_{8345}, a_{8353}, a_{8361}, a_{8369}, a_{8377}, a_{8385}, a_{8393}, a_{8401}, a_{8409}, a_{8417}, a_{8425}, a_{8433}, a_{8441}, a_{8449}, a_{8457}, a_{8465}, a_{8473}, a_{8481}, a_{8489}, a_{8497}, a_{8505}, a_{8513}, a_{8521}, a_{8529}, a_{8537}, a_{8545}, a_{8553}, a_{8561}, a_{8569}, a_{8577}, a_{8585}, a_{8593}, a_{8601}, a_{8609}, a_{8617}, a_{8625}, a_{8633}, a_{8641}, a_{8649}, a_{8657}, a_{8665}, a_{8673}, a_{8681}, a_{8689}, a_{8697}, a_{8705}, a_{8713}, a_{8721}, a_{8729}, a_{8737}, a_{8745}, a_{8753}, a_{8761}, a_{8769}, a_{8777}, a_{8785}, a_{8793}, a_{8801}, a_{8809}, a_{8817}, a_{8825}, a_{8833}, a_{8841}, a_{8849}, a_{8857}, a_{8865}, a_{8873}, a_{8881}, a_{8889}, a_{8897}, a_{8905}, a_{8913}, a_{8921}, a_{8929}, a_{8937}, a_{8945}, a_{8953}, a_{8961}, a_{8969}, a_{8977}, a_{8985}, a_{8993}, a_{9001}, a_{9009}, a_{9017}, a_{9025}, a_{9033}, a_{9041}, a_{9049}, a_{9057}, a_{9065}, a_{9073}, a_{9081}, a_{9089}, a_{9097}, a_{9105}, a_{9113}, a_{9121}, a_{9129}, a_{9137}, a_{9145}, a_{9153}, a_{9161}, a_{9169}, a_{9177}, a_{9185}, a_{9193}, a_{9201}, a_{9209}, a_{9217}, a_{9225}, a_{9233}, a_{9241}, a_{9249}, a_{9257}, a_{9265}, a_{9273}, a_{9281}, a_{9289}, a_{9297}, a_{9305}, a_{9313}, a_{9321}, a_{9329}, a_{9337}, a_{9345}, a_{9353}, a_{9361}, a_{9369}, a_{9377}, a_{9385}, a_{9393}, a_{9401}, a_{9409}, a_{9417}, a_{9425}, a_{9433}, a_{9441}, a_{9449}, a_{9457}, a_{9465}, a_{9473}, a_{9481}, a_{9489}, a_{9497}, a_{9505}, a_{9513}, a_{9521}, a_{9529}, a_{9537}, a_{9545}, a_{9553}, a_{9561}, a_{9569}, a_{9577}, a_{9585}, a_{9593}, a_{9601}, a_{9609}, a_{9617}, a_{9625}, a_{9633}, a_{9641}, a_{9649}, a_{9657}, a_{9665}, a_{9673}, a_{9681}, a_{9689}, a_{9697}, a_{9705}, a_{9713}, a_{9721}, a_{9729}, a_{9737}, a_{9745}, a_{9753}, a_{9761}, a_{9769}, a_{9777}, a_{9785}, a_{9793}, a_{9801}, a_{9809}, a_{9817}, a_{9825}, a_{9833}, a_{9841}, a_{9849}, a_{9857}, a_{9865}, a_{9873}, a_{9881}, a_{9889}, a_{9897}, a_{9905}, a_{9913}, a_{9921}, a_{9929}, a_{9937}, a_{9945}, a_{9953}, a_{9961}, a_{9969}, a_{9977}, a_{9985}, a_{9993}, a_{10001}, a_{10009}, a_{10017}, a_{10025}, a_{10033}, a_{10041}, a_{10049}, a_{10057}, a_{10065}, a_{10073}, a_{10081}, a_{10089}, a_{10097}, a_{10105}, a_{10113}, a_{10121}, a_{10129}, a_{10137}, a_{10145}, a_{10153}, a_{10161}, a_{10169}, a_{10177}, a_{10185}, a_{10193}, a_{10201}, a_{10209}, a_{10217}, a_{10225}, a_{10233}, a_{10241}, a_{10249}, a_{10257}, a_{10265}, a_{10273}, a_{10281}, a_{10289}, a_{10297}, a_{10305}, a_{10313}, a_{10321}, a_{10329}, a_{10337}, a_{10345}, a_{10353}, a_{10361}, a_{10369}, a_{10377}, a_{10385}, a_{10393}, a_{10401}, a_{10409}, a_{10417}, a_{10425}, a_{10433}, a_{10441}, a_{10449}, a_{10457}, a_{10465}, a_{10473}, a_{10481}, a_{10489}, a_{10497}, a_{10505}, a_{10513}, a_{10521}, a_{10529}, a_{10537}, a_{10545}, a_{10553}, a_{10561}, a_{10569}, a_{10577}, a_{10585}, a_{10593}, a_{10601}, a_{10609}, a_{10617}, a_{10625}, a_{10633}, a_{10641}, a_{10649}, a_{10657}, a_{10665}, a_{10673}, a_{10681}, a_{10689}, a_{10697}, a_{10705}, a_{10713}, a_{10721}, a_{10729}, a_{10737}, a_{10745}, a_{10753}, a_{10761}, a_{10769}, a_{10777}, a_{10785}, a_{10793}, a_{10801}, a_{10809}, a_{10817}, a_{10825}, a_{10833}, a_{10841}, a_{10849}, a_{10857}, a_{10865}, a_{10873}, a_{10881}, a_{10889}, a_{10897}, a_{10905}, a_{10913}, a_{10921}, a_{10929}, a_{10937}, a_{10945}, a_{10953}, a_{10961}, a_{10969}, a_{10977}, a_{10985}, a_{10993}, a_{11001}, a_{11009}, a_{11017}, a_{11025}, a_{11033}, a_{11041}, a_{11049}, a_{11057}, a_{11065}, a_{11073}, a_{11081}, a_{11089}, a_{11097}, a_{11105}, a_{11113}, a_{11121}, a_{11129}, a_{11137}, a_{11145}, a_{11153}, a_{11161}, a_{11169}, a_{11177}, a_{11185}, a_{11193}, a_{11201}, a_{11209}, a_{11217}, a_{11225}, a_{11233}, a_{11241}, a_{11249}, a_{11257}, a_{11265}, a_{11273}, a_{11281}, a_{11289}, a_{11297}, a_{11305}, a_{11313}, a_{11321}, a_{11329}, a_{11337}, a_{11345}, a_{11353}, a_{11361}, a_{11369}, a_{11377}, a_{11385}, a_{11393}, a_{11401}, a_{11409}, a_{11417}, a_{11425}, a_{11433}, a_{11441}, a_{11449}, a_{11457}, a_{11465}, a_{11473}, a_{11481}, a_{11489}, a_{11497}, a_{11505}, a_{11513}, a_{11521}, a_{11529}, a_{11537}, a_{11545}, a_{11553}, a_{11561}, a_{11569}, a_{11577}, a_{11585}, a_{11593}, a_{11601}, a_{11609}, a_{11617}, a_{11625}, a_{11633}, a_{11641}, a_{11649}, a_{11657}, a_{11665}, a_{11673}, a_{11681}, a_{11689}, a_{11697}, a_{11705}, a_{11713}, a_{11721}, a_{11729}, a_{11737}, a_{11745}, a_{11753}, a_{11761}, a_{11769}, a_{11777}, a_{11785}, a_{11793}, a_{11801}, a_{11809}, a_{11817}, a_{11825}, a_{11833}, a_{11841}, a_{11849}, a_{11857}, a_{11865}, a_{11873}, a_{11881}, a_{11889}, a_{11897}, a_{11905}, a_{11913}, a_{11921}, a_{11929}, a_{11937}, a_{11945}, a_{11953}, a_{11961}, a_{11969}, a_{11977}, a_{11985}, a_{11993}, a_{12001}, a_{12009}, a_{12017}, a_{12025}, a_{12033}, a_{12041}, a_{12049}, a_{12057}, a_{12065}, a_{12073}, a_{12081}, a_{12089}, a_{12097}, a_{12105}, a_{12113}, a_{12121}, a_{12129}, a_{12137}, a_{12145}, a_{12153}, a_{12161}, a_{12169}, a_{12177}, a_{12185}, a_{12193}, a_{12201}, a_{12209}, a_{12217}, a_{12225}, a_{12233}, a_{12241}, a_{12249}, a_{12257}, a_{12265}, a_{12273}, a_{12281}, a_{12289}, a_{12297}, a_{12305}, a_{12313}, a_{12321}, a_{12329}, a_{12337}, a_{12345}, a_{12353}, a_{12361}, a_{12369}, a_{12377}, a_{12385}, a_{12393}, a_{12401}, a_{12409}, a_{12417}, a_{12425}, a_{12433}, a_{12441}, a_{12449}, a_{12457}, a_{12465}, a_{12473}, a_{12481}, a_{12489}, a_{12497}, a_{12505}, a_{12513}, a_{12521}, a_{12529}, a_{12537}, a_{12545}, a_{12553}, a_{12561}, a_{12569}, a_{12577}, a_{12585}, a_{12593}, a_{12601}, a_{12609}, a_{12617}, a_{12625}, a_{12633}, a_{12641}, a_{12649}, a_{12657}, a_{12665}, a_{12673}, a_{12681}, a_{12689}, a_{12697}, a_{12705}, a_{12713}, a_{12721}, a_{12729}, a_{12737}, a_{12745}, a_{12753}, a_{12761}, a_{12769}, a_{12777}, a_{12785}, a_{12793}, a_{12801}, a_{12809}, a_{12817}, a_{12825}, a_{12833}, a_{12841}, a_{12849}, a_{12857}, a_{12865}, a_{12873}, a_{12881}, a_{12889}, a_{12897}, a_{12905}, a_{12913}, a_{12921}, a_{12929}, a_{12937}, a_{12945}, a_{12953}, a_{12961}, a_{12969}, a_{12977}, a_{12985}, a_{12993}, a_{13001}, a_{13009}, a_{13017}, a_{13025}, a_{13033}, a_{13041}, a_{13049}, a_{13057}, a_{13065}, a_{13073}, a_{13081}, a_{13089}, a_{13097}, a_{13105}, a_{13113}, a_{13121}, a_{13129}, a_{13137}, a_{13145}, a_{13153}, a_{13161}, a_{13169}, a_{13177}, a_{13185}, a_{13193}, a_{13201}, a_{13209}, a_{13217},$$

Configuração (b)

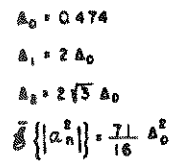


Fig. III.1.2.1

Como forma de ilustrar o processo de partição de um conjunto de sinais tipo $A_2 \times A_2$ iremos criar o subconjunto C_4^7 com base nos conjuntos C_1 ilustrados pela Fig.III.1.2.1.

C_4^7 é um dos 8 subconjuntos de A_4^0 e

$$C_4^7 = C_1 \times C_2 \cup C_3 \times C_0.$$

Para gerar este subconjunto serão utilizados os programas READC e DELTAOC (Apêndice 2) que geram C_4^7 em função do valor Δ_0 encontrado, para A_4^0 tipo $A_2 \times A_2$, para em seguida calcular a menor distância intraconjunto neste nível da partição. No caso do reticulado $A_2 \times A_2$, não é conhecido o comportamento do cescimento das distâncias intraconjuntos ao longo da partição, portanto para a obtenção dos mesmos é necessário fazer-se o cálculo das distâncias intra-conjunto à cada nível da partição. A seguir serão listados os resultados para o nível C da partição de $A_2 \times A_2$.

Delta 0 = 3.3570000000E-01

SUBCONJUNTO C_0

1 => (-0.3357, 0.8721 >
2 => (0.3357, 0.8721 >
3 => (-0.6714, 0.2907 >
4 => (0.0000, 0.2907 >
5 => (0.6714, 0.2907 >
6 => (-0.3357,-0.2907 >
7 => (0.3357,-0.2907 >
8 => (0.0000,-0.8721 >

SUBCONJUNTO C_1

1 => (-0.1678, 0.5814 >
2 => (0.5036, 0.5814 >
3 => (-0.5036, 0.0000 >
4 => (0.1678, 0.0000 >
5 => (0.8393, 0.0000 >
6 => (-0.8393,-0.5814 >
7 => (-0.1678,-0.5814 >
8 => (0.5036,-0.5814 >

SUBCONJUNTO C_2

1 => (0.0000, 0.8721 >
2 => (-0.3357, 0.2907 >
3 => (0.3357, 0.2907 >
4 => (-0.6714,-0.2907 >
5 => (0.0000,-0.2907 >
6 => (0.6714,-0.2907 >
7 => (-0.3357,-0.8721 >
8 => (0.3357,-0.8721 >

SUBCONJUNTO C_3

1 => (-0.5036, 0.5814 >
2 => (0.1678, 0.5814 >
3 => (0.8393, 0.5814 >
4 => (-0.8393, 0.0000 >
5 => (-0.1678, 0.0000 >
6 => (0.5036, 0.0000 >
7 => (-0.5036,-0.5814 >
8 => (0.1678,-0.5814 >

SUBCONJUNTO C_4^7 DA PARTIÇÃO DE $A_2 \times A_2$

- 1 $\Rightarrow$ (-0.1678, 0.5814, 0.0000, 0.8721)
- 2 $\Rightarrow$ (-0.1678, 0.5814, -0.3357, 0.2907)
- 3 $\Rightarrow$ (-0.1678, 0.5814, 0.3357, 0.2907)
- 4 $\Rightarrow$ (-0.1678, 0.5814, -0.6714, -0.2907)
- 5 $\Rightarrow$ (-0.1678, 0.5814, 0.0000, -0.2907)
- 6 $\Rightarrow$ (-0.1678, 0.5814, 0.6714, -0.2907)
- 7 $\Rightarrow$ (-0.1678, 0.5814, -0.3357, -0.8721)
- 8 $\Rightarrow$ (-0.1678, 0.5814, 0.3357, -0.8721)
- 9 $\Rightarrow$ (0.5036, 0.5814, 0.0000, 0.8721)
- 10 $\Rightarrow$ (0.5036, 0.5814, -0.3357, 0.2907)
- 11 $\Rightarrow$ (0.5036, 0.5814, 0.3357, 0.2907)
- 12 $\Rightarrow$ (0.5036, 0.5814, -0.6714, -0.2907)
- 13 $\Rightarrow$ (0.5036, 0.5814, 0.0000, -0.2907)
- 14 $\Rightarrow$ (0.5036, 0.5814, 0.6714, -0.2907)
- 15 $\Rightarrow$ (0.5036, 0.5814, -0.3357, -0.8721)
- 16 $\Rightarrow$ (0.5036, 0.5814, 0.3357, -0.8721)
- 17 $\Rightarrow$ (-0.5036, 0.0000, 0.0000, 0.8721)
- 18 $\Rightarrow$ (-0.5036, 0.0000, -0.3357, 0.2907)
- 19 $\Rightarrow$ (-0.5036, 0.0000, 0.3357, 0.2907)
- 20 $\Rightarrow$ (-0.5036, 0.0000, -0.6714, -0.2907)
- 21 $\Rightarrow$ (-0.5036, 0.0000, 0.0000, -0.2907)
- 22 $\Rightarrow$ (-0.5036, 0.0000, 0.6714, -0.2907)
- 23 $\Rightarrow$ (-0.5036, 0.0000, -0.3357, -0.8721)
- 24 $\Rightarrow$ (-0.5036, 0.0000, 0.3357, -0.8721)
- 25 $\Rightarrow$ (0.1678, 0.0000, 0.0000, 0.8721)
- 26 $\Rightarrow$ (0.1678, 0.0000, -0.3357, 0.2907)

27 => (0.1678, 0.0000, 0.3357, 0.2907)
28 => (0.1678, 0.0000, -0.6714, -0.2907)
29 => (0.1678, 0.0000, 0.0000, -0.2907)
30 => (0.1678, 0.0000, 0.6714, -0.2907)
31 => (0.1678, 0.0000, -0.3357, -0.8721)
32 => (0.1678, 0.0000, 0.3357, -0.8721)
33 => (0.8393, 0.0000, 0.0000, 0.8721)
34 => (0.8393, 0.0000, -0.3357, 0.2907)
35 => (0.8393, 0.0000, 0.3357, 0.2907)
36 => (0.8393, 0.0000, -0.6714, -0.2907)
37 => (0.8393, 0.0000, 0.0000, -0.2907)
38 => (0.8393, 0.0000, 0.6714, -0.2907)
39 => (0.8393, 0.0000, -0.3357, -0.8721)
40 => (0.8393, 0.0000, 0.3357, -0.8721)
41 => (-0.8393, -0.5814, 0.0000, 0.8721)
42 => (-0.8393, -0.5814, -0.3357, 0.2907)
43 => (-0.8393, -0.5814, 0.3357, 0.2907)
44 => (-0.8393, -0.5814, -0.6714, -0.2907)
45 => (-0.8393, -0.5814, 0.0000, -0.2907)
46 => (-0.8393, -0.5814, 0.6714, -0.2907)
47 => (-0.8393, -0.5814, -0.3357, -0.8721)
48 => (-0.8393, -0.5814, 0.3357, -0.8721)
49 => (-0.1678, -0.5814, 0.0000, 0.8721)
50 => (-0.1678, -0.5814, -0.3357, 0.2907)
51 => (-0.1678, -0.5814, 0.3357, 0.2907)
52 => (-0.1678, -0.5814, -0.6714, -0.2907)
53 => (-0.1678, -0.5814, 0.0000, -0.2907)
54 => (-0.1678, -0.5814, 0.6714, -0.2907)
55 => (-0.1678, -0.5814, -0.3357, -0.8721)

56 => (-0.1678, -0.5814, 0.3357, -0.8721)
57 => (0.5036, -0.5814, 0.0000, 0.8721)
58 => (0.5036, -0.5814, -0.3357, 0.2907)
59 => (0.5036, -0.5814, 0.3357, 0.2907)
60 => (0.5036, -0.5814, -0.6714, -0.2907)
61 => (0.5036, -0.5814, 0.0000, -0.2907)
62 => (0.5036, -0.5814, 0.6714, -0.2907)
63 => (0.5036, -0.5814, -0.3357, -0.8721)
64 => (0.5036, -0.5814, 0.3357, -0.8721)
65 => (-0.5036, 0.5814, -0.3357, 0.8721)
66 => (-0.5036, 0.5814, 0.3357, 0.8721)
67 => (-0.5036, 0.5814, -0.6714, 0.2907)
68 => (-0.5036, 0.5814, 0.0000, 0.2907)
69 => (-0.5036, 0.5814, 0.6714, 0.2907)
70 => (-0.5036, 0.5814, -0.3357, -0.2907)
71 => (-0.5036, 0.5814, 0.3357, -0.2907)
72 => (-0.5036, 0.5814, 0.0000, -0.8721)
73 => (0.1678, 0.5814, -0.3357, 0.8721)
74 => (0.1678, 0.5814, 0.3357, 0.8721)
75 => (0.1678, 0.5814, -0.6714, 0.2907)
76 => (0.1678, 0.5814, 0.0000, 0.2907)
77 => (0.1678, 0.5814, 0.6714, 0.2907)
78 => (0.1678, 0.5814, -0.3357, -0.2907)
79 => (0.1678, 0.5814, 0.3357, -0.2907)
80 => (0.1678, 0.5814, 0.0000, -0.8721)
81 => (0.8393, 0.5814, -0.3357, 0.8721)
82 => (0.8393, 0.5814, 0.3357, 0.8721)
83 => (0.8393, 0.5814, -0.6714, 0.2907)
84 => (0.8393, 0.5814, 0.0000, 0.2907)

Capítulo 4

85 => (0.8393, 0.5814, 0.6714, 0.2907)
86 => (0.8393, 0.5814, -0.3357, -0.2907)
87 => (0.8393, 0.5814, 0.3357, -0.2907)
88 => (0.8393, 0.5814, 0.0000, -0.8721)
89 => (-0.8393, 0.0000, -0.3357, 0.8721)
90 => (-0.8393, 0.0000, 0.3357, 0.8721)
91 => (-0.8393, 0.0000, -0.6714, 0.2907)
92 => (-0.8393, 0.0000, 0.0000, 0.2907)
93 => (-0.8393, 0.0000, 0.6714, 0.2907)
94 => (-0.8393, 0.0000, -0.3357, -0.2907)
95 => (-0.8393, 0.0000, 0.3357, -0.2907)
96 => (-0.8393, 0.0000, 0.0000, -0.8721)
97 => (-0.1678, 0.0000, -0.3357, 0.8721)
98 => (-0.1678, 0.0000, 0.3357, 0.8721)
99 => (-0.1678, 0.0000, -0.6714, 0.2907)
100 => (-0.1678, 0.0000, 0.0000, 0.2907)
101 => (-0.1678, 0.0000, 0.6714, 0.2907)
102 => (-0.1678, 0.0000, -0.3357, -0.2907)
103 => (-0.1678, 0.0000, 0.3357, -0.2907)
104 => (-0.1678, 0.0000, 0.0000, -0.8721)
105 => (0.5036, 0.0000, -0.3357, 0.8721)
106 => (0.5036, 0.0000, 0.3357, 0.8721)
107 => (0.5036, 0.0000, -0.6714, 0.2907)
108 => (0.5036, 0.0000, 0.0000, 0.2907)
109 => (0.5036, 0.0000, 0.6714, 0.2907)
110 => (0.5036, 0.0000, -0.3357, -0.2907)
111 => (0.5036, 0.0000, 0.3357, -0.2907)
112 => (0.5036, 0.0000, 0.0000, -0.8721)
113 => (-0.5036, -0.5814, -0.3357, 0.8721)

114 => (-0.5036, -0.5814, 0.3357, 0.8721)
115 => (-0.5036, -0.5814, -0.6714, 0.2907)
116 => (-0.5036, -0.5814, 0.0000, 0.2907)
117 => (-0.5036, -0.5814, 0.6714, 0.2907)
118 => (-0.5036, -0.5814, -0.3357, -0.2907)
119 => (-0.5036, -0.5814, 0.3357, -0.2907)
120 => (-0.5036, -0.5814, 0.0000, -0.8721)
121 => (0.1678, -0.5814, -0.3357, 0.8721)
122 => (0.1678, -0.5814, 0.3357, 0.8721)
123 => (0.1678, -0.5814, -0.6714, 0.2907)
124 => (0.1678, -0.5814, 0.0000, 0.2907)
125 => (0.1678, -0.5814, 0.6714, 0.2907)
126 => (0.1678, -0.5814, -0.3357, -0.2907)
127 => (0.1678, -0.5814, 0.3357, -0.2907)
128 => (0.1678, -0.5814, 0.0000, -0.8721)

Calculo das distancias de cada subconjunto

Distancia minima em C04.dat = 0.4748
Distancia minima em C14.dat = 0.4748
Distancia minima em C24.dat = 0.4748
Distancia minima em C34.dat = 0.4748
Distancia minima em C44.dat = 0.4748
Distancia minima em C54.dat = 0.4748
Distancia minima em C64.dat = 0.4748
Distancia minima em C74.dat = 0.4748

IV - CONCLUSÕES

Em síntese, os resultados encontrados através de cálculo computacional são mostrados na TABELA IV.1, onde Δ_c é a distância mínima intraconjunto no nível C da partição.

TABELA IV.1

	Z^4	$A_2 \times A_2$
Δ_o	0.3162	0.3357
Δ_c	0.6324	0.4748

Para haver coerência entre os resultados encontrados, os pontos reticulados foram gerados sob a condição de Energia Média Quadrática igual a 1. Assim, a comparação entre estas estruturas reticulares pode ser feita diretamente pela distância Euclideana entre os mesmos.

A distância mínima (Δ_o) no nível A da partição aponta para uma densidade maior entre os pontos gerados a partir de $A_2 \times A_2$ comparados com os pontos de Z^4 . Pois, se a energia média é igual, o número de pontos é o mesmo para os dois reticulados e os pontos de $A_2 \times A_2$ estão mais distantes entre si (vide TABELA II.1), este reticulado apresenta, então, uma melhor compactação dos

pontos quando comparada com a compactação dos pontos em Z^4 .

A julgar por este resultado, poder-se-ia supor para a constelação de sinais gerada sobre $A_2 \times A_2$ um melhor aproveitamento comparado à constelação gerada sobre Z^4 , no caso de se utilizar estas constelações como base para o mapeamento de um codificador TCM. Isto, dado que a medida de desempenho está baseada no critério do maior valor para $d_{\min}$ o qual é obtido em função de Δ_0 . Porém, como foi caracterizado no Capítulo 2, apenas este resultado é insuficiente para se fazer esta análise.

Descendo-se, portanto, ao nível C da partição o que se constata é a inversão dos resultados, isto é, a distância mínima intraconjunto em Z^4 cresce substancialmente, ao passo que a distância mínima intraconjunto (Δ_c) em $A_2 \times A_2$ cresce em valor bem inferior. Neste nível da partição, o ganho alcançado pela compactabilidade dos sinais não foi relevante. A particularidade do particionamento do conjunto de sinais e sua eficácia quanto ao desmembramento dos mesmos, está diretamente ligado à simetria de cada reticulado, e ao número de vizinhos que cada sinal comporta.

O número de vizinhos é inerente à cada estrutura reticular, em Z^4 este número é certamente inferior ao de um sinal em $A_2 \times A_2$ (este último se origina a partir de um reticulado hexagonal onde cada sinal se avizinha de 6 outros). Ocorrendo, desta forma, para a constelação de sinais tipo Z^4 , uma maior facilidade no desmembramento dos sinais com o decorrer da partição.

Os resultados acima projetam, portanto, um ganho de

$$20 \log_{10} (0.6324/0.4748) = 2.48 \text{ dB}$$

da constelação Z^4 sobre a de $A_2 \times A_2$.

Este valor pode se alterar caso desçamos a níveis mais inferiores na árvore de partição (D, E,...), onde deve ocorrer um maior desmembramento entre os sinais, porém, a complexidade aumentaria substancialmente com o aumento do número de estados do codificador.

Esquemas TCM multi-dimensionais de baixa complexidade, no geral, exibem ganhos assintóticos de codificação superiores aos sistemas de dimensão inferior [5].

A idéia de se trabalhar com estruturas reticulares de sinais em 4-D consiste em projetar os sinais o mais distantes possível uns dos outros (sujeitos à restrição de energia média e com uma estrutura de sinais padronizada). Na análise bi-dimensional efetuada no Capítulo 2, houve a preocupação de se estabelecer os mesmos critérios na elaboração das estruturas de sinais. Porém em 4-D, existe mais "espaço" em determinadas estruturas resultando em pontos um pouco mais distantes do que em 2-D. Assim, enquanto se mantém a energia média por componente constante e o mesmo nível de ruído, é menos provável que ocorra um erro de detecção em 4-D. Por outro lado, a vantagem do aumento na distância entre os pontos é parcialmente reduzida pelo fato de que há mais caminhos onde um erro pode ocorrer (quando o número de pontos em uma constelação é aumentado e aumentado também o número de vizinhos por sinal).

De qualquer forma, uma constelação é um subconjunto finito de um reticulado de infinitos pontos, e é de se esperar que, para os casos reais de sistemas de comunicações, um típico ponto x_i pertencente à constelação perderá alguns dos vizinhos que ele tinha no reticulado.

Por exemplo, no caso do reticulado D_4 , cada sinal desta estrutura está em contato com 24 outros sinais, no entanto no caso de se trabalhar com uma constelação de 256 sinais o número de vizinhos varia de 8 a 23 com valor médio de 14.25 sinais [17], diminuindo, desta forma, o número de vizinhos, o que justificaria no futuro o estudo do comportamento de estruturas como esta para o sistema TCM.

CAPÍTULO 5

CONCLUSÕES

Conclusões

Neste trabalho, nos preocupamos em estudar determinadas estruturas reticulares do espaço Euclideano n -dimensional e utilizá-las para definir o lugar no espaço dos pontos de sinais, de forma a que estes se situem de maneira regular e exata.

Desta forma, a saída de um codificador convolucional mapeada em uma constelação originada de uma destas estruturas reticulares, possui seus pontos bem definidos, e portanto, torna-se fácil e sistemático o cálculo dos parâmetros que quantificam o desempenho deste sistema — TCM — de comunicações.

A vantagem de realizar o estudo focalizando-o por este ângulo, foi o de relacionar a Teoria relativa ao Empacotamento Esférico de sinais com a Teoria de Codificação, e aproveitar toda uma gama variada de ferramentas, que nos permitiu, ao caracterizar as estruturas através da geometria dos números, tirar um número maior de informações a respeito das mesmas. Desta forma, foram propostas estruturas alternativas às convencionais para configurar os sinais no espaço, uma vez que estas apontavam por oferecer uma maior distância Euclideana entre os pontos de sinais que compunham. Feito isto, passou-se a analisar o comportamento de constelações de sinais originadas destas estruturas quando do emprego da técnica TCM.

No Capítulo 1, abordamos o problema da codificação de canal e apresentamos o estudo da técnica TCM (Modulação por Codificação em Treliça) indicando-a como forma de obter ganhos vantajosos sobre as interferências indesejadas aos sinais transmitidos, e superando com ganhos substanciais, as demais técnicas convencionais de modulação multi-nível não codificadas.

No Capítulo 2, foram definidos os conceitos a respeito do Empacotamento Esférico de sinais. Foram apresentadas duas estruturas reticulares de 2 dimensões, Z^2 e A_2 . Foram propostas constelações de 8, 16 e 32 sinais para estas estruturas, com a preocupação de apresentar a melhor configuração possível, em termos de energia média dos sinais. Foi feita a Partição de Conjunto para cada uma destas constelações, calculando-se a distância mínima intra-conjunto e a energia média de cada configuração obtida.

No Capítulo 3, foi posta em prática a teoria estudada nos Capítulos anteriores. No sentido de analisar os resultados obtidos, foi feita uma análise algébrica dos codificadores convolucionais de tal forma a auxiliar a classificação dos mesmos. Realizou-se uma procura por codificadores convolucionais — não catastróficos, mínimos, que geram códigos TCM ótimos — utilizando os conjuntos de sinais gerados em cada um dos reticulados apresentados. Estes resultados foram tabelados, assim como os codificadores equivalentes, (Apêndice 1), e em seguida foi feita uma análise comparativa de desempenho entre os esquemas TCM, para várias taxas de codificação.

Neste Capítulo 3, para análise dos resultados, além dos já existentes, foram definidos dois novos parâmetros: $FLDA$ e GCA_{max} (respectivamente, Fator Limitante de Desempenho Assintótico e Ganho máximo de Codificação Assintótica,). Estes valores permitem a comparação entre as diferentes configurações de sinais, gerados a partir de qualquer estrutura reticular, levando-se apenas em consideração a capacidade máxima de codificação destes sinais, sem levar em conta a decodificação dos mesmos. Isto nos permitiu inferir sobre quais estruturas, e a que taxas de codificação devemos concentrar esforços em trabalhos futuros, com o objetivo de encontrar sistemas TCM mais eficientes, uma vez que sejam aperfeiçoados os algoritmos de decodificação e procura por codificadores ótimos existentes.

No Capítulo 4, foram apresentadas três estruturas conhecidas de 4 dimensões (Z^4 , A_4 , D_4), e calculados os seus principais parâmetros, e proposta uma nova estrutura reticular, também de 4 dimensões ($A_2 \times A_2$) baseada no reticulado bi-dimensional hexagonal.

Através de um Algoritmo original deste trabalho, (Apêndice 2), foram construídos dois conjuntos de 1024 sinais do tipo Z^4 e $A_2 \times A_2$ e da mesma forma, calculadas a Energia Média Quadrática e a distância mínima entre estes sinais quadri-dimensionais.

Um outro Algoritmo foi proposto (Apêndice 2) para se fazer a Partição de Conjunto destas constelações. Foram listados alguns subconjuntos da partição e calculada a distância mínima intraconjunto para o nível C de cada partição. Através dos resultados pudemos comparar as estruturas e inferir sobre futuros

Conclusões

trabalhos.

Como sugestão, propomos o aperfeiçoamento dos algoritmos de procura e decodificação, elevando a sua velocidade de processamento, assim como, a capacidade de trabalhar com taxas superiores, e quantidade maior de memórias codificadas.

Propomos o estudo comparativo de constelações de 64, 128, 256, ... sinais tipo Z^2 e A_2 , à taxas de codificação superiores, acreditando encontrarem-se resultados superiores e substanciais para as constelações originadas de A_2 , dado que o desmembramento dos sinais com o decorrer da Partição de Conjunto ocorre com maior facilidade a partir do 4º nível da partição, para este reticulado.

Da mesma, forma acreditamos que o reticulado $A_2 \times A_2$, também apresente resultados promissores nos níveis inferiores da partição de conjunto.

Também propomos o estudo do comportamento, no ambiente TCM, do reticulado D_4 , dado suas promissoras características, cuidando para não se formar constelações de número elevado de sinais, para não se estender o número de vizinhos por sinal.

APÊNDICE 1

CODIFICADORES EQUIVALENTES

RETICULADOS: Z^2 e A_2

TAXAS: $\frac{2}{3}$, $\frac{3}{4}$, $\frac{4}{5}$

*1	[6, 11]
*2	[6, 11]
*3	[6, 11]
*4	[4, 9] [7, 9] [2, 5]
*5	[11, 5] [14, 5] [11, 6] [10, 7] [11, 7] [12, 7] [19, 7] [4, 11] [6, 11] [7, 11]
*6	[20, 11] [21, 11] [30, 11] [31, 11] [18, 19] [19, 19] [29, 14] [21, 15] [22, 15] [25, 15] [26, 15] [19, 19] [14, 19] [11, 21] [15, 21] [14, 29]
*7	[4, 9] [7, 9] [2, 5]
*8	[11, 5] [6, 11] [7, 11] [14, 5] [11, 6] [10, 7] [11, 7] [12, 7] [19, 7] [4, 11] [15, 11]
*9	[20, 11] [21, 11] [30, 11] [31, 11] [8, 19] [19, 19] [30, 19] [31, 19] [19, 14] [29, 14] [21, 15] [22, 15] [25, 15] [26, 15] [19, 19] [14, 19] [11, 21] [15, 21] [14, 29]
*10	[6, 11]
*11	[8, 19, 6] [8, 17, 7] [8, 22, 7] [8, 6, 19]

*12	[24,39,7] [24,38,7] [24,41,7] [24,40,7] [24,35,19] [24,38,19] [24,49,19] [24,40,19] [24,39,14] [24,49,14] [24,0,35]
-----	---

*13	[7,9] [2,5]
-----	-------------

*14	[14,9] [11,5] [14,5] [11,0] [10,7] [11,7] [12,7] [19,7] [4,11] [5,11] [0,11] [7,11]
-----	--

*15	[22,12,1] [20,12,1] [20,14,1] [20,14,1] [14,20,1] [12,22,1] [0,24,1] [20,4,9] [30,4,9] [20,0,9] [28,0,9] [22,14,9] [24,14,9] [14,22,9] [4,20,9] [0,20,9] [25,9,5] [20,9,5] [19,8,5] [27,8,5] [20,11,5] [20,14,5] [20,14,5] [8,19,5] [24,1,0] [25,1,0] [17,8,0] [19,8,0] [25,8,0] [27,8,0] [20,9,0] [19,10,0] [25,10,0] [19,11,0] [24,11,0] [8,17,0] [8,19,0] [9,19,0] [10,19,0] [11,19,0] [24,9,7] [17,10,7] [19,10,7] [20,10,7] [22,10,7] [25,10,7] [27,10,7] [28,10,7] [30,10,7] [17,11,7] [20,11,7] [22,12,7] [20,12,7] [22,14,7] [24,14,7] [8,17,7] [9,17,7] [10,17,7] [11,17,7] [10,19,7] [10,20,7] [8,22,7] [10,22,7] [12,22,7] [14,22,7] [20,4,9] [30,4,9] [20,0,9] [28,0,9] [4,20,9] [0,20,9] [24,0,11] [30,0,11] [0,24,11] [24,9,19] [27,9,19] [20,1,14] [27,1,14] [1,20,14] [8,0,19] [10,0,19] [12,0,19] [14,0,19] [0,8,19] [0,10,19] [17,11,5] [19,9,0]
-----	---

*16	[31,9] [20,11] [21,11] [30,11] [31,11] [18,19] [19,19] [30,19] [31,19] [19,14] [29,14] [21,15] [22,15] [25,15] [26,15] [19,19] [14,19] [11,21] [15,21] [14,29]
-----	---

*17	[16,14,37] [16,14,39]
-----	--------------------------

*18	[7,2] [4,3] [7,9] [2,5]
-----	-------------------------

*19	[14,13] [11,5] [14,5] [11,6] [10,7] [11,7] [12,7] [13,7] [4,11] [5,11] [6,11] [7,11]
-----	---

*20	IDEM *15
-----	----------

*21	IDEM *16
-----	----------

*22	IDEM *17
-----	----------

*23	IDEM *18
-----	----------

*24	IDEM *14
-----	----------

*25	[16,14,39] [16,14,35] [16,14,37] [16,14,39] [39,26,14] [46,22,1] [56,22,1] [22,46,1] [38,30,3] [56,30,3] [30,38,3] [51,11,5] [56,11,5] [54,14,5] [56,14,5] [49,26,5] [46,26,5] [49,26,5] [52,26,5] [26,49,5] [26,46,5] [49,9,6] [56,9,6] [49,19,6] [56,19,6] [35,24,6] [41,24,6] [49,24,6] [59,24,6] [35,26,6] [57,26,6] [24,35,6] [26,35,6] [24,41,6] [19,49,6] [9,49,6] [34,24,7] [38,24,7] [41,24,7] [46,24,7] [49,24,7] [54,24,7] [57,24,7] [62,24,7] [39,24,7] [56,25,7] [35,26,7] [36,26,7] [57,26,7] [62,26,7] [38,30,7] [56,30,7] [24,39,7] [25,36,7] [26,35,7] [26,36,7] [24,38,7] [30,38,7] [24,41,7] [24,46,7] [56,6,9] [62,6,9] [6,56,9] [49,26,12] [49,26,12]
-----	--

*25	continuação,	[26,49,12]	[46,22,19]	[56,22,19]
		[35,24,19]	[38,24,19]	[49,24,19]
		[46,24,19]	[51,24,19]	
		[54,24,19]	[59,24,17]	[62,24,19]
		[69,24,19]	[44,26,19]	
		[54,26,19]	[59,26,19]	[35,27,19]
		[56,27,19]	[26,33,19]	
		[24,35,19]	[27,35,19]	[24,38,19]
		[24,49,19]	[26,44,19]	
		[22,46,19]	[24,46,19]	[41,17,14]
		[56,17,14]	[39,24,14]	
		[49,24,14]	[51,24,14]	[57,24,14]
		[99,26,14]	[59,26,14]	
		[24,39,14]	[26,39,14]	[17,41,14]
		[24,49,14]	[38,26,15]	
		[41,26,15]	[51,26,15]	[60,26,15]
		[26,38,15]	[26,41,15]	
		[56,6,19]	[62,6,19]	[6,56,19]
		[56,1,22]	[57,1,22]	
		[1,56,22]	[56,3,29]	[59,3,29]
		[24,6,35]	[26,6,35]	
		[28,6,35]	[30,6,35]	[6,24,35]
		[6,26,35]	[22,12,39]	
		[26,12,39]	[12,22,39]	

*26	IDEM *16
*27	IDEM *17
*28	[5,11] [6,11]
*29	[8,19,16] [8,17,7] [8,22,7] [8,6,19]
*30	[14,23]
*31	[24,14,37] [24,6,41]
*32	[92,28,10,65]

*35	O RESULTADO DA PROCURA PELOS MELHORES CODIFICADORES DESTA CLASSE SIMPLES EXIGE UM TEMPO EXCESSIVAMENTE GRANDE DE CPU, PORTANTO FOI EVITADA ESTA PROCURA DADO QUE PARA O CORRESPONDENTE VALOR DE $d_{\min}$ QUALQUER CODIFICADOR DE *34 POSSUI O MESMO DESEMPENHO. O VALOR $\Delta_{\min}(\mu) = 1.083$ FOI OBTIDO ATRAVÉS DA EQUAÇÃO $\text{Min} \left(\Delta_4^2 ; \Delta_2^2 + \Delta_1^2 ; 2\Delta_1^2 + 2\Delta_0^2 \right), \text{ [38, Fig. 7]}$
-----	--

*36	[14,9] [11,5] [14,5] [11,6] [10,7] [11,7] [12,7] [19,7] [4,11] [5,11] [6,11] [7,11]
-----	---

*37	IDEM *25
-----	----------

*38	IDEM *16
-----	----------

*39	[24,14,33] [24,6,95] [24,14,37] [24,6,41]
-----	---

*40	[52,20,10,65] [32,28,10,65] [36,28,10,65] [56,28,10,65] [60,28,10,65]
-----	--

*41	[25,10] [61,10] [59,11] [62,11] [47,18] [45,19] [46,19] [61,19] [62,19] [46,21] [59,21] [41,22] [49,22] [61,22] [63,22] [41,23] [62,23] [46,25] [47,25] [54,25] [55,25] [37,27] [38,27] [61,27] [62,27] [38,29] [42,29] [47,29] [50,29] [55,29]
-----	--

*41	continuação, [59,29] [97,30] [53,30] [97,91] [41,91] [49,91] [45,91] [50,91] [52,91] [54,91] [58,91] [26,97] [27,97] [30,97] [91,97] [22,49] [31,49] [19,45] [91,45] [18,47] [20,47] [25,47] [26,47] [29,47] [10,55]
-----	---

*42	[56,22,65] [56,30,65] [56,22,67] [56,30,67] [56,14,69] [56,30,69] [56,14,71] [56,30,71] [56,6,73] [56,6,75] [56,22,77] [56,22,79] [56,6,81] [56,14,81] [56,6,83] [56,14,83]
-----	---

*43	O RESULTADO DA PROCURA PELOS MELHORES CODIFICADORES TCM DESTA CLASSE SIMPLES EXIGE UM TEMPO DE CPU EXCESSIVAMENTE ELEVADO, PORTANTO EXCLUÍMOS ESTA PROCURA, DADO QUE PARA O CORRESPONDENTE VALOR DE d_{min} ENCONTRADO, QUALQUER UM DOS CODIFICADORES DE *36, *38, *41, *42 ALCANÇAM O MESMO DESEMPENHO. O VALOR $\Delta_{min}(\mu) = 0.950$ FOI OBTIDO ATRAVÉS DA EQUAÇÃO $\text{Min} \left[\Delta_4^2 ; \Delta_2^2 + \Delta_1^2 ; 2\Delta_1^2 + 2\Delta_0^2 \right], [38, \text{Fig. 7}]$
-----	--

APÊNDICE 2

PROGRAMAS E RESULTADOS:

- (i) - Programa READA
 - (ii) - Programa DELTAOA
 - (iii) - Programa READC
 - (iv) - Programa DELTAOC
-

```
program READA;
```

```
uses
```

```
    CRT;
```

```
const
```

```
    max = 8;
```

```
type
```

```
    vetor = array [1..max] of real;
```

```
    XY = record  
        X, Y : Real;  
    end;
```

```
    XYZW = record  
        X, Y, Z, W : Real;  
    end;
```

```
var
```

```
    i, j : Integer;
```

```
    dmin : real;
```

```
    cmx, cmx, cmz, cmw : real;
```

```
    tipoxy1, tipoxy2 : xy;
```

```
    tipoxyzw : xyzw;
```

```
    x, y : vetor;
```

```
    ( arquivo dos resultados )  
    arqres : text;
```

```
    ( arquivo de pontos bidimensionais )  
    A0 : file of XY;
```

```
    ( arquivos de pontos 4-dimensionais )  
    A0xA0 : file of XYZW;
```

```
procedure WRITE_ARQ2;
```

```
var
```

```

    tipoxy : xy;
    i, j : integer;

begin

    reset ( A0 );

    clrscr;
    write ( 'Conjunto A0' );
    writeln;
    writeln;
    write ( arqres, 'Conjunto A0' );
    writeln ( arqres );
    writeln ( arqres );
    for i := 1 to 4 do
    begin
        for j := 1 to 8 do
        begin
            seek ( A0, (8*(i-1))+j-1 );
            read ( A0, tipoxy );
            with tipoxy do
            begin
                write ( (8*(i-1))+j, ' => ( ', x:6:4, ', ', y:6:4,
                        ' )' );

                write ( arqres, (8*(i-1))+j, ' => ( ', x:6:4,
                        ', ', y:6:4, ' )' );

            end;
            writeln;
            writeln ( arqres );
        end;
    end;

    close ( A0 );

    writeln ( arqres );
    writeln ( arqres );

    readln;

end;

procedure READ_ARQ2;

var

    tipoXY : XY;
    certo : char;
    i, j : integer;

begin

    assign ( A0, 'A0.dat' );
    rewrite ( A0 );

```

```

for i := 1 to 4 do
begin

    certo := 'N';

    while ( certo = 'N' ) do
    begin

        clrscr;
        write ( 'Leitura A0' );
        writeln;
        writeln;

        for j := 1 to 8 do
        begin
            write ( 'x', (8*(i-1))+j, ' = ' );
            readln ( x[j] );
            write ( 'y', (8*(i-1))+j, ' = ' );
            readln ( y[j] );
            writeln;
        end;
        writeln ( 'Esta certo (s/n) ? ' );
        readln ( certo );
        certo := upcase ( certo );

        if ( certo <> 'N' ) then
        begin
            for j := 1 to 8 do
            begin
                tipoxy.x := x[j];
                tipoxy.y := y[j];
                seek ( A0, (8*(i-1))+j-1 );
                write ( A0, tipoxy );
            end;
        end;
    end;
end;

close ( A0 );

WRITE_ARQ2;

end;

begin

    assign ( arqres, 'arquivo1.dat' );
    rewrite ( arqres );

    READ_ARQ2;

    close ( arqres );
end.

```

```
program DELTA0A;
```

```
uses
```

```
    CRT;
```

```
const
```

```
    max = 8;
```

```
type
```

```
    vetor = array [1..max] of real;
```

```
    XY = record  
        X, Y : Real;  
    end;
```

```
    XYZW = record  
        X, Y, Z, W : Real;  
    end;
```

```
var
```

```
    i, j : Integer;
```

```
    dmin, d0 : real;
```

```
    cmx, cmx, cmz, cmw : real;
```

```
    tipoxy1, tipoxy2 : xy;
```

```
    tipoxyzw : xyzw;
```

```
    x, y : vetor;
```

```
    { arquivo dos resultados }  
    arqres : text;
```

```
    { arquivo de pontos bidimensionais }  
    A0 : file of XY;
```

```
    { arquivos de pontos 4-dimensionais }  
    A0xA0 : file of XYZW;
```

```
procedure WRITE_ARQ4;
```

```
var
```

```

tipoxyzw : xyzw;
i : integer;

begin

  clrscr;
  write ( 'Conjunto A0xA0' );
  writeln;
  writeln;

  write ( arqres, 'Conjunto A0xA0' );
  writeln ( arqres );
  writeln ( arqres );

  reset ( A0xA0 );

  for i := 1 to 1024 do
  begin
    seek ( A0xA0, i-1 );
    read ( A0xA0, tipoxyzw );
    with tipoxyzw do
    begin
      write ( i, ' => ( ', x:6:4, ', ', y:6:4, ', ', z:6:4,
              ', ', w:6:4, ' )' );

      write ( arqres, i, ' => ( ', x:6:4, ', ', y:6:4, ', ', z:6:4,
              ', ', w:6:4, ' )' );

    end;
    writeln;
    writeln ( arqres );
  end;

  close ( A0xA0 );

  writeln ( arqres );
  writeln ( arqres );

  readln;

end;

```

```

procedure DEVELOP_ARQ4 ( Fator : real );

begin

  rewrite ( A0xA0 );

  reset ( A0 );

  clrscr;

```

```

( Monta o arquivo A0xA0 )

writeln ( 'Montando o conjunto A0xA0' );

for i := 1 to 32 do
begin
  seek ( A0, i-1 );
  read ( A0, tipoxy1 );
  tipoxyzw.x := tipoxy1.x*Fator;
  tipoxyzw.y := tipoxy1.y*Fator;
  for j := 1 to 32 do
  begin
    seek ( A0, j-1 );
    read ( A0, tipoxy2 );
    tipoxyzw.z := tipoxy2.x*Fator;
    tipoxyzw.w := tipoxy2.y*Fator;
    with tipoxyzw do
    begin
      cmx := cmx+x*Fator;
      cmy := cmy+y*Fator;
      cmz := cmz+z*Fator;
      cmw := cmw+w*Fator;
    end;
    write ( A0xA0, tipoxyzw );

  end;
end;

close ( A0 );
close ( A0xA0 );

WRITE_ARQ4;

end;

procedure CALC_ENERGIA;

var
  tipoxyzw : XYZW;
  enrg : real;
  i : integer;

begin
  reset ( A0xA0 );

  enrg := 0;

  for i := 1 to 1024 do
  begin
    seek ( A0xA0, i-1 );

```

```

    read ( A0xA0, tipoxyzw );
    with tipoxyzw do
        enrg := enrg + sqr ( cmx-x ) + sqr ( cmx-y ) + sqr ( cmz-z ) +
            sqr ( cmw-w );
    end;

    enrg := enrg / 1024;

    writeln ( 'Energia = ', enrg, ' * Delta 0 * Delta 0' );
    writeln ( arqres, 'Energia = ', enrg, ' * Delta 0 * Delta 0' );

    d0 := sqrt ( 1/enrg );

    writeln;
    writeln ( arqres );

    writeln ( 'Delta 0 = ', d0:6:4 );
    writeln ( arqres, 'Delta 0 = ', d0:6:4 );

    writeln ( arqres );

end;

```

```

procedure CALC_DIST;

```

```

var

```

```

    tipoxyzw1, tipoxyzw2 : XYZW;
    d : real;
    i, j : integer;

```

```

begin

```

```

    reset ( A0xA0 );

```

```

    dmin := 1e38;

```

```

    for i := 1 to 1024 do

```

```

    begin

```

```

        seek ( A0xA0, i-1 );

```

```

        read ( A0xA0, tipoxyzw1 );

```

```

        for j := ( i+1 ) to 1024 do

```

```

        begin

```

```

            seek ( A0xA0, j-1 );

```

```

            read ( A0xA0, tipoxyzw2 );

```

```

            with tipoxyzw1 do

```

```

                d := sqrt ( sqr ( x-tipoxyzw2.x ) + sqr ( y-tipoxyzw2.y ) +
                    sqr ( z-tipoxyzw2.z ) + sqr ( w-tipoxyzw2.w ) );

```

```

            if ( d < dmin ) then

```

```

                dmin := d;

```

```

            end;

```

```

        end;

```

```

close ( A0xA0 );

writeln ( 'Distancia minima em A0xA0 = ', dmin:6:4 );
writeln ( arqres, 'Distancia minima em A0xA0 = ', dmin:6:4 );

end;

begin

cmx := 0;
cmy := 0;
cmz := 0;
cmw := 0;

dmin := 0;

assign ( arqres, 'arquivo2.dat' );
rewrite ( arqres );
writeln ( arqres );
writeln ( arqres );

assign ( A0, 'A0.dat' );
assign ( A0xA0, 'A0xA0.dat' );

DEVELOP_ARQ4 ( 1 );

cmx := cmx/1024;
cmy := cmy/1024;
cmz := cmz/1024;
cmw := cmw/1024;

writeln;
writeln ( arqres );
writeln ( 'Centro de massa : ', cmx:6:4, ' ', cmy:6:4, ' ', cmz
        ' ', cmw:6:4 );

writeln ( arqres, 'Centro de massa : ', cmx:6:4, ' ', cmy:6:4,
        ' ', cmz:6:4, ' ', cmw:6:4 );

writeln ( arqres );

readln;

CALC_ENERGIA;

readln;

DEVELOP_ARQ4 ( d0 );

clrscr;

```

```
write ( 'Calculo da distancia do conjunto A0xA0' );
writeln;
writeln;

writeln ( arqres, 'Calculo da distancia do conjunto A0xA0' );
writeln ( arqres );

CALC_DIST;

writeln;
writeln ( 'Fim...' );
writeln;
writeln ( 'Tecle <ENTER>' );

readln;

close ( arqres );

end.
```

```
program READC;
```

```
uses
```

```
    CRT;
```

```
type
```

```
    XY = record  
        X, Y : Real;  
    end;
```

```
var
```

```
    ( arquivo dos resultados )  
    arqres : text;  
  
    delta0 : real;
```

```
procedure WRITE_ARQ2 ( Nome : string );
```

```
var
```

```
    arq : file of XY;  
    tipoxy : XY;  
    i, j : integer;
```

```
begin
```

```
    assign ( arq, Nome );  
    reset ( arq );
```

```
    clrscr;  
    write ( 'Conjunto ', Nome );  
    writeln;  
    writeln;
```

```
    write ( arqres, 'Conjunto ', Nome );  
    writeln ( arqres );  
    writeln ( arqres );
```

```
    for i := 1 to 8 do  
        begin
```

```
            seek ( arq, i-1 );  
            read ( arq, tipoxy );  
            with tipoxy do
```

```
                begin  
                    write ( i, ' => ( ', x:6:4, ', ', y:6:4, ' )' );  
                    write ( arqres, i, ' => ( ', x:6:4, ', ', y:6:4, ' )'
```

```
                end;  
                writeln;  
                writeln ( arqres );  
            end;  
        end;
```

```

end;

close ( arq );

writeln ( arqres );
writeln ( arqres );

readln;

end;

procedure READ_ARQ2 ( Nome : string );

var

    arq : file of XY;
    tipoXY : XY;
    certo : char;
    i : integer;

begin

    assign ( arq, Nome );

    certo := 'N';

    while ( certo = 'N' ) do
    begin

        clrscr;
        write ( 'Leitura ', Nome );
        writeln;
        writeln;

        rewrite ( arq );

        for i := 1 to 8 do
        begin
            write ( 'x', i, ' = ' );
            readln ( tipoXY.X );
            write ( 'y', i, ' = ' );
            readln ( tipoXY.Y );
            writeln;
            with tipoXY do
            begin
                X := X * delta0;
                Y := Y * delta0;
            end;
            seek ( arq, i-1 );
            write ( arq, tipoXY );
        end;

        writeln ( 'Esta certo (s/n) ? ' );
    end;

```

```
        readln ( certo );
        certo := upcase ( certo );

    end;

    close ( arq );

    WRITE_ARQ2 ( Nome );

end;

begin

    clrscr;

    write ( 'Valor de delta 0 : ' );
    readln ( delta0 );

    assign ( arqres, 'arquivc1.dat' );
    rewrite ( arqres );
    writeln ( arqres );
    writeln ( arqres );

    writeln ( arqres );
    writeln ( arqres, 'Delta 0 = ', delta0 );
    writeln ( arqres );

    READ_ARQ2 ( 'C0.dat' );
    READ_ARQ2 ( 'C1.dat' );
    READ_ARQ2 ( 'C2.dat' );
    READ_ARQ2 ( 'C3.dat' );

    close ( arqres );

end.
```

program DELTAOC;

uses

CRT;

type

XY = record
 X, Y : Real;
end;

XYZW = record
 X, Y, Z, W : Real;
end;

var

i, j : Integer;

tipoxy1, tipoxy2 : xy;

tipoxyzw : xyzw;

{ arquivo dos resultados }
arqres : text;

{ arquivo de pontos bidimensionais }
C0, C1, C2, C3 : file of XY;

{ arquivos de pontos 4-dimensionais }
C0xC0, C0xC1, C0xC2, C0xC3,
C1xC0, C1xC1, C1xC2, C1xC3,
C2xC0, C2xC1, C2xC2, C2xC3,
C3xC0, C3xC1, C3xC2, C3xC3,

C04, C14, C24, C34, C44, C54, C64, C74 : file of XYZW;

procedure WRITE_ARQ4 (Nome : string);

var

arq : file of XYZW;
tipoxyzw : xyzw;
i : integer;

begin


```

clrscr;
write ( 'Conjunto ', Nome );
writeln;
writeln;

write ( arqres, 'Conjunto ', Nome );
writeln ( arqres );
writeln ( arqres );

assign ( arq, Nome );
reset ( arq );

for i := 1 to 128 do
begin
  seek ( arq, i-1 );
  read ( arq, tipoxyzw );
  with tipoxyzw do
  begin
    write ( i, ' => ( ', x:6:4, ', ', y:6:4, ', ', z:6:4,
              ', ', w:6:4, ' )' );

    write ( arqres, i, ' => ( ', x:6:4, ', ', y:6:4, ', ',
              ', ', w:6:4, ' )' );

    end;
    writeln;
    writeln ( arqres );
  end;

  close ( arq );

  writeln ( arqres );
  writeln ( arqres );

  readln;

end;

```

```

procedure CALC_DIST ( Nome : string );

```

```

var

```

```

  arq : file of XYZW;
  tipoxyzw1, tipoxyzw2 : XYZW;
  dmin, d : real;
  i, j : integer;

```

```

begin

```

```

  assign ( arq, Nome );

```

```
assign ( C2xC2, 'C2xC2.dat' );
assign ( C2xC3, 'C2xC3.dat' );
```

```
assign ( C3xC0, 'C3xC0.dat' );
assign ( C3xC1, 'C3xC1.dat' );
assign ( C3xC2, 'C3xC2.dat' );
assign ( C3xC3, 'C3xC3.dat' );
```

```
assign ( C04, 'C04.dat' );
assign ( C14, 'C14.dat' );
assign ( C24, 'C24.dat' );
assign ( C34, 'C34.dat' );
assign ( C44, 'C44.dat' );
assign ( C54, 'C54.dat' );
assign ( C64, 'C64.dat' );
assign ( C74, 'C74.dat' );
```

```
clrscr;
```

```
{ Monta os arquivos C0x... }
```

```
writeln ( 'Montando os conjuntos C0x...' );
```

```
reset ( C0 );
reset ( C1 );
reset ( C2 );
reset ( C3 );
```

```
rewrite ( C0xC0 );
rewrite ( C0xC1 );
rewrite ( C0xC2 );
rewrite ( C0xC3 );
```

```
for i := 1 to 8 do
begin
```

```
    seek ( C0, i-1 );
    read ( C0, tipoxy1 );
    tipoxyzw.x := tipoxy1.x;
    tipoxyzw.y := tipoxy1.y;
    for j := 1 to 8 do
    begin
```

```
        { C0xC0 }
        seek ( C0, j-1 );
        read ( C0, tipoxy2 );
        tipoxyzw.z := tipoxy2.x;
        tipoxyzw.w := tipoxy2.y;
        with tipoxyzw do
            write ( C0xC0, tipoxyzw );
```

```
        { C0xC1 }
        seek ( C1, j-1 );
        read ( C1, tipoxy2 );
```

```

reset ( arq );

dmin := 1e38;
for i := 1 to 128 do
begin
    seek ( arq, i-1 );
    read ( arq, tipoxyzw1 );
    for j := ( i+1 ) to 128 do
    begin
        seek ( arq, j-1 );
        read ( arq, tipoxyzw2 );
        with tipoxyzw1 do
            d := sqrt ( sqr ( x-tipoxyzw2.x ) + sqr ( y-tipoxyzw2.y
                                + sqr ( z-tipoxyzw2.z ) + sqr ( w-tipoxyzw2.w
                                );

        if ( d < dmin ) then
            dmin := d;
    end;
end;

close ( arq );

writeln ( 'Distancia minima em ', Nome, ' = ', dmin:6:4 );
writeln ( arqres, 'Distancia minima em ', Nome, ' = ', dmin:6:4 );

end;

```

```

begin
    assign ( arqres, 'arquivc2.dat' );
    rewrite ( arqres );
    writeln ( arqres );
    writeln ( arqres );

    assign ( C0, 'C0.dat' );
    assign ( C1, 'C1.dat' );
    assign ( C2, 'C2.dat' );
    assign ( C3, 'C3.dat' );

    assign ( C0xC0, 'C0xC0.dat' );
    assign ( C0xC1, 'C0xC1.dat' );
    assign ( C0xC2, 'C0xC2.dat' );
    assign ( C0xC3, 'C0xC3.dat' );

    assign ( C1xC0, 'C1xC0.dat' );
    assign ( C1xC1, 'C1xC1.dat' );
    assign ( C1xC2, 'C1xC2.dat' );
    assign ( C1xC3, 'C1xC3.dat' );

    assign ( C2xC0, 'C2xC0.dat' );
    assign ( C2xC1, 'C2xC1.dat' );

```

```

reset ( arq );

dmin := 1e38;
for i := 1 to 128 do
begin
    seek ( arq, i-1 );
    read ( arq, tipoxyzw1 );
    for j := ( i+1 ) to 128 do
    begin
        seek ( arq, j-1 );
        read ( arq, tipoxyzw2 );
        with tipoxyzw1 do
            d := sqrt ( sqr ( x-tipoxyzw2.x ) + sqr ( y-tipoxyzw2.y
                                + sqr ( z-tipoxyzw2.z ) + sqr ( w-tipoxyzw2.w

        if ( d < dmin ) then
            dmin := d;
    end;
end;

close ( arq );

writeln ( 'Distancia minima em ', Nome, ' = ', dmin:6:4 );
writeln ( arqres, 'Distancia minima em ', Nome, ' = ', dmin

end;

```

```

begin

    assign ( arqres, 'arquivc2.dat' );
    rewrite ( arqres );
    writeln ( arqres );
    writeln ( arqres );

    assign ( C0, 'C0.dat' );
    assign ( C1, 'C1.dat' );
    assign ( C2, 'C2.dat' );
    assign ( C3, 'C3.dat' );

    assign ( C0xC0, 'C0xC0.dat' );
    assign ( C0xC1, 'C0xC1.dat' );
    assign ( C0xC2, 'C0xC2.dat' );
    assign ( C0xC3, 'C0xC3.dat' );

    assign ( C1xC0, 'C1xC0.dat' );
    assign ( C1xC1, 'C1xC1.dat' );
    assign ( C1xC2, 'C1xC2.dat' );
    assign ( C1xC3, 'C1xC3.dat' );

    assign ( C2xC0, 'C2xC0.dat' );
    assign ( C2xC1, 'C2xC1.dat' );

```

```

assign ( C2xC2, 'C2xC2.dat' );
assign ( C2xC3, 'C2xC3.dat' );

assign ( C3xC0, 'C3xC0.dat' );
assign ( C3xC1, 'C3xC1.dat' );
assign ( C3xC2, 'C3xC2.dat' );
assign ( C3xC3, 'C3xC3.dat' );

assign ( C04, 'C04.dat' );
assign ( C14, 'C14.dat' );
assign ( C24, 'C24.dat' );
assign ( C34, 'C34.dat' );
assign ( C44, 'C44.dat' );
assign ( C54, 'C54.dat' );
assign ( C64, 'C64.dat' );
assign ( C74, 'C74.dat' );

```

```

clrscr;

```

```

( Monta os arquivos C0x... )

```

```

writeln ( 'Montando os conjuntos C0x...' );

```

```

reset ( C0 );
reset ( C1 );
reset ( C2 );
reset ( C3 );

```

```

rewrite ( C0xC0 );
rewrite ( C0xC1 );
rewrite ( C0xC2 );
rewrite ( C0xC3 );

```

```

for i := 1 to 8 do
begin

```

```

    seek ( C0, i-1 );
    read ( C0, tipoxy1 );
    tipoxyzw.x := tipoxy1.x;
    tipoxyzw.y := tipoxy1.y;
    for j := 1 to 8 do
    begin

```

```

        { C0xC0 }
        seek ( C0, j-1 );
        read ( C0, tipoxy2 );
        tipoxyzw.z := tipoxy2.x;
        tipoxyzw.w := tipoxy2.y;
        with tipoxyzw do
            write ( C0xC0, tipoxyzw );

```

```

        { C0xC1 }
        seek ( C1, j-1 );
        read ( C1, tipoxy2 );

```

```

tipoxyzw.z := tipoxy2.x;
tipoxyzw.w := tipoxy2.y;
with tipoxyzw do
  write ( C0xC1, tipoxyzw );

{ C0xC2 }
seek ( C2, j-1 );
read ( C2, tipoxy2 );
tipoxyzw.z := tipoxy2.x;
tipoxyzw.w := tipoxy2.y;
with tipoxyzw do
  write ( C0xC2, tipoxyzw );

{ C0xC3 }
seek ( C3, j-1 );
read ( C3, tipoxy2 );
tipoxyzw.z := tipoxy2.x;
tipoxyzw.w := tipoxy2.y;
with tipoxyzw do
  write ( C0xC3, tipoxyzw );

end;
end;

close ( C0 );
close ( C1 );
close ( C2 );
close ( C3 );

close ( C0xC0 );
close ( C0xC1 );
close ( C0xC2 );
close ( C0xC3 );

{ Monta os arquivos C1x... }

writeln ( 'Montando os conjuntos C1x...' );

reset ( C0 );
reset ( C1 );
reset ( C2 );
reset ( C3 );

rewrite ( C1xC0 );
rewrite ( C1xC1 );
rewrite ( C1xC2 );
rewrite ( C1xC3 );

for i := 1 to 8 do
begin
  seek ( C1, i-1 );
  read ( C1, tipoxy1 );
  tipoxyzw.x := tipoxy1.x;

```

```

reset ( C1 );
reset ( C2 );
reset ( C3 );

rewrite ( C2xC0 );
rewrite ( C2xC1 );
rewrite ( C2xC2 );
rewrite ( C2xC3 );

for i := 1 to 8 do
begin
  seek ( C2, i-1 );
  read ( C2, tipoxy1 );
  tipoxyzw.x := tipoxy1.x;
  tipoxyzw.y := tipoxy1.y;
  for j := 1 to 8 do
  begin
    { C2xC0 }
    seek ( C0, j-1 );
    read ( C0, tipoxy2 );
    tipoxyzw.z := tipoxy2.x;
    tipoxyzw.w := tipoxy2.y;
    with tipoxyzw do
      write ( C2xC0, tipoxyzw );

    { C2xC1 }
    seek ( C1, j-1 );
    read ( C1, tipoxy2 );
    tipoxyzw.z := tipoxy2.x;
    tipoxyzw.w := tipoxy2.y;
    with tipoxyzw do
      write ( C2xC1, tipoxyzw );

    { C2xC2 }
    seek ( C2, j-1 );
    read ( C2, tipoxy2 );
    tipoxyzw.z := tipoxy2.x;
    tipoxyzw.w := tipoxy2.y;
    with tipoxyzw do
      write ( C2xC2, tipoxyzw );

    { C2xC3 }
    seek ( C3, j-1 );
    read ( C3, tipoxy2 );
    tipoxyzw.z := tipoxy2.x;
    tipoxyzw.w := tipoxy2.y;
    with tipoxyzw do
      write ( C2xC3, tipoxyzw );

  end;
end;

close ( C0 );
close ( C1 );

```

```
tipoxyzw.y := tipoxy1.y;
for j := 1 to 8 do
begin
    ( C1xC0 )
    seek ( C0, j-1 );
    read ( C0, tipoxy2 );
    tipoxyzw.z := tipoxy2.x;
    tipoxyzw.w := tipoxy2.y;
    with tipoxyzw do
        write ( C1xC0, tipoxyzw );

    ( C1xC1 )
    seek ( C1, j-1 );
    read ( C1, tipoxy2 );
    tipoxyzw.z := tipoxy2.x;
    tipoxyzw.w := tipoxy2.y;
    with tipoxyzw do
        write ( C1xC1, tipoxyzw );

    ( C1xC2 )
    seek ( C2, j-1 );
    read ( C2, tipoxy2 );
    tipoxyzw.z := tipoxy2.x;
    tipoxyzw.w := tipoxy2.y;
    with tipoxyzw do
        write ( C1xC2, tipoxyzw );

    ( C1xC3 )
    seek ( C3, j-1 );
    read ( C3, tipoxy2 );
    tipoxyzw.z := tipoxy2.x;
    tipoxyzw.w := tipoxy2.y;
    with tipoxyzw do
        write ( C1xC3, tipoxyzw );

    end;
end;

close ( C0 );
close ( C1 );
close ( C2 );
close ( C3 );

close ( C1xC0 );
close ( C1xC1 );
close ( C1xC2 );
close ( C1xC3 );

( Monta os arquivos C2x... )

writeln ( 'Montando os conjuntos C2x...' );

reset ( C0 );
```

```
close ( C2 );  
close ( C3 );
```

```
close ( C2xC0 );  
close ( C2xC1 );  
close ( C2xC2 );  
close ( C2xC3 );
```

```
( Monta os arquivos C3x... )
```

```
writeln ( 'Montando os conjuntos C3x...' );
```

```
reset ( C0 );  
reset ( C1 );  
reset ( C2 );  
reset ( C3 );
```

```
rewrite ( C3xC0 );  
rewrite ( C3xC1 );  
rewrite ( C3xC2 );  
rewrite ( C3xC3 );
```

```
for i := 1 to 8 do  
begin
```

```
    seek ( C3, i-1 );  
    read ( C3, tipoxy1 );  
    tipoxyzw.x := tipoxy1.x;  
    tipoxyzw.y := tipoxy1.y;  
    for j := 1 to 8 do  
    begin
```

```
        ( C3xC0 )  
        seek ( C0, j-1 );  
        read ( C0, tipoxy2 );  
        tipoxyzw.z := tipoxy2.x;  
        tipoxyzw.w := tipoxy2.y;  
        with tipoxyzw do  
            write ( C3xC0, tipoxyzw );
```

```
        ( C3xC1 )  
        seek ( C1, j-1 );  
        read ( C1, tipoxy2 );  
        tipoxyzw.z := tipoxy2.x;  
        tipoxyzw.w := tipoxy2.y;  
        with tipoxyzw do  
            write ( C3xC1, tipoxyzw );
```

```
        ( C3xC2 )  
        seek ( C2, j-1 );  
        read ( C2, tipoxy2 );  
        tipoxyzw.z := tipoxy2.x;  
        tipoxyzw.w := tipoxy2.y;  
        with tipoxyzw do  
            write ( C3xC2, tipoxyzw );
```

```

        ( C3xC3 )
        seek ( C3, j-1 );
        read ( C3, tipoxy2 );
        tipoxyzw.z := tipoxy2.x;
        tipoxyzw.w := tipoxy2.y;
        with tipoxyzw do
            write ( C3xC3, tipoxyzw );

    end;
end;

close ( C0 );
close ( C1 );
close ( C2 );
close ( C3 );

close ( C3xC0 );
close ( C3xC1 );
close ( C3xC2 );
close ( C3xC3 );

( Monta o arquivo C04 )

writeln ( 'Montando o arquivo C04' );

reset ( C0xC0 );
reset ( C2xC2 );

rewrite ( C04 );

for i := 1 to 64 do
begin
    seek ( C0xC0, i-1 );
    read ( C0xC0, tipoxyzw );
    write ( C04, tipoxyzw );
end;

for i := 1 to 64 do
begin
    seek ( C2xC2, i-1 );
    read ( C2xC2, tipoxyzw );
    write ( C04, tipoxyzw );
end;

close ( C0xC0 );
close ( C2xC2 );
close ( C04 );

( Monta o arquivo C14 )

```

```
writeln ( 'Montando o arquivo C14' );
```

```
reset ( C0xC1 );  
reset ( C2xC3 );
```

```
rewrite ( C14 );
```

```
for i := 1 to 64 do  
begin  
    seek ( C0xC1, i-1 );  
    read ( C0xC1, tipoxyzw );  
    write ( C14, tipoxyzw );  
end;
```

```
for i := 1 to 64 do  
begin  
    seek ( C2xC3, i-1 );  
    read ( C2xC3, tipoxyzw );  
    write ( C14, tipoxyzw );  
end;
```

```
close ( C0xC1 );  
close ( C2xC3 );  
close ( C14 );
```

```
( Monta o arquivo C24 )
```

```
writeln ( 'Montando o arquivo C24' );
```

```
reset ( C1xC1 );  
reset ( C3xC3 );
```

```
rewrite ( C24 );
```

```
for i := 1 to 64 do  
begin  
    seek ( C1xC1, i-1 );  
    read ( C1xC1, tipoxyzw );  
    write ( C24, tipoxyzw );  
end;
```

```
for i := 1 to 64 do  
begin  
    seek ( C3xC3, i-1 );  
    read ( C3xC3, tipoxyzw );  
    write ( C24, tipoxyzw );  
end;
```

```
close ( C1xC1 );  
close ( C3xC3 );  
close ( C24 );
```

(Monta o arquivo C34)

writeln ('Montando o arquivo C34');

reset (C1xC0);

reset (C3xC2);

rewrite (C34);

for i := 1 to 64 do

begin

seek (C1xC0, i-1);

read (C1xC0, tipoxyzw);

write (C34, tipoxyzw);

end;

for i := 1 to 64 do

begin

seek (C3xC2, i-1);

read (C3xC2, tipoxyzw);

write (C34, tipoxyzw);

end;

close (C1xC0);

close (C3xC2);

close (C34);

(Monta o arquivo C44)

writeln ('Montando o arquivo C44');

reset (C0xC2);

reset (C2xC0);

rewrite (C44);

for i := 1 to 64 do

begin

seek (C0xC2, i-1);

read (C0xC2, tipoxyzw);

write (C44, tipoxyzw);

end;

for i := 1 to 64 do

begin

seek (C2xC0, i-1);

read (C2xC0, tipoxyzw);

write (C44, tipoxyzw);

end;

close (C0xC2);

close (C2xC0);

close (C44);

{ Monta o arquivo C54 }

writeln ('Montando o arquivo C54');

reset (C0xC3);

reset (C2xC1);

rewrite (C54);

for i := 1 to 64 do

begin

seek (C0xC3, i-1);

read (C0xC3, tipoxyzw);

write (C54, tipoxyzw);

end;

for i := 1 to 64 do

seek (C2xC1, i-1);

read (C2xC1, tipoxyzw);

write (C54, tipoxyzw);

close (C0xC3);

close (C2xC1);

close (C54);

{ Monta o arquivo C64 }

writeln ('Montando o arquivo C64');

reset (C1xC3);

reset (C3xC1);

rewrite (C64);

for i := 1 to 64 do

begin

seek (C1xC3, i-1);

read (C1xC3, tipoxyzw);

write (C64, tipoxyzw);

end;

for i := 1 to 64 do

begin

seek (C3xC1, i-1);

read (C3xC1, tipoxyzw);

write (C64, tipoxyzw);

end;

```
close ( C1xC3 );
close ( C3xC1 );
close ( C64 );
```

```
( Monta o arquivo C74 )
```

```
writeln ( 'Montando o arquivo C74' );
```

```
reset ( C1xC2 );
reset ( C3xC0 );
```

```
rewrite ( C74 );
```

```
for i := 1 to 64 do
begin
    seek ( C1xC2, i-1 );
    read ( C1xC2, tipoxyzw );
    write ( C74, tipoxyzw );
end;
```

```
for i := 1 to 64 do
begin
    seek ( C3xC0, i-1 );
    read ( C3xC0, tipoxyzw );
    write ( C74, tipoxyzw );
end;
```

```
close ( C1xC2 );
close ( C3xC0 );
close ( C74 );
```

```
WRITE_ARQ4 ( 'C74.dat' );
```

```
clrscr;
```

```
write ( 'Calculo das distancias de cada subconjunto' );
writeln;
writeln;
```

```
writeln ( arqres, 'Calculo das distancias de cada subconjun
writeln ( arqres );
writeln ( arqres );
```

```
CALC_DIST ( 'C04.dat' );
CALC_DIST ( 'C14.dat' );
CALC_DIST ( 'C24.dat' );
CALC_DIST ( 'C34.dat' );
CALC_DIST ( 'C44.dat' );
```

```
      CALC_DIST ( 'G54.dat' );  
      CALC_DIST ( 'G64.dat' );  
      CALC_DIST ( 'G74.dat' );  
  
      writeln;  
      writeln ( 'Fim...' );  
      writeln;  
      writeln ( 'Tecla <ENTER>' );  
  
      readln;  
      close ( arqres );  
end.
```

REFÊRENCIAS
BIBLIOGRÁFICAS

- [1] - O. G. C. Gonçalves, R. Palazzo Jr., W.C. Borelli, S. Motoyama, "ESTUDO DO PROCESSO DE CODIFICAÇÃO DE CANAL — PARA O SISTEMA LOCKS — UTILIZANDO A TÉCNICA DE MODULAÇÃO POR CODIFICAÇÃO EM TRELICA". Contrato Telebrás 208/87 - RT 45 - ATA , 1987.
- [2] - O. G. C. Gonçalves, R. Palazzo Jr., "ANÁLISE DE ESTRUTURAS RETICULARES PARA MODULAÇÃO POR CODIFICAÇÃO EM TRELIÇA EM 2 DIMENSÕES". Contrato Telebrás 208/87 - RT - ATA Pub. JUL 89.
- [3] - O. G. C. Gonçalves, R. Palazzo Jr., "DETERMINAÇÃO E ANÁLISE DE CODIFICADORES TCM ÓTIMOS PARA RETICULADOS EM 2 DIMENSÕES" Contrato Telebrás 208/87 - RT- 61 ATA, Pub. 013/89
- [4] - O. G. C. Gonçalves, R. Palazzo Jr., "ANÁLISE DE ESTRUTURAS RETICULARES PARA MODULAÇÃO POR CODIFICAÇÃO EM TRELIÇA EM 4 DIMENSÕES" Contrato Telebrás - em publicação.
- [5] - Ungerböeck, G. : "TRELLIS CODED MODULATION WITH REDUNDANT SIGNALS SET - PART I AND PART II: STATE OF THE ART", IEEE Communications Magazine, February 1987- vol 25 n° 2.
- [6] - J.H. Conway and N.J. Sloane, "A FAST ENCODING METHOD FOR LATTICE CODES AND QUANTIZERS", IEEE Trans. Information Theory, vol IT-29, n° 6, november 1983.

- [7] - N.J.A. Sloane, "TABLES OF SPHERE PACKINGS AND SPHERICAL CODES", IEEE Transactions on Information Theory, vol IT-27, n° 3, may 1981.
- [8] - A. Boss, J.H. Conway and N.J.A. Sloane, "FURTHER LATTICE PACKINGS IN HIGH DIMENSIONS", Mathematika, vol.29.Part 2, december 1982, n° 58.
- [9] - J. Leech and N.J.A. Sloane, "SPHERE PACKINGS AND ERROR CORRECTING CODES", Can. J. Math., vol XXIII, n°4, 1971, pp.718-745.
- [10] - J.H. Conway and N.J.A. Sloane, "FAST QUANTIZING AND DECODING ALGORITHMS FOR LATTICE QUANTIZERS AND CODES", IEEE Transactions on Information Theory, vol. IT-28, march 1982.
- [11] - A.R. Calderbank and N.J.A. Sloane, "NEW TRELLIS CODES BASED ON LATTICES AND COSETS", IEEE Transactions on Information Theory, vol. IT-33, n° 2, march 1987.
- [12] - G.D. Forney, Jr., "CONVOLUTIONAL CODES I: ALGEBRAIC STRUCTURE", IEEE Transactions Information Theory, vol. IT-16, pp. 720-738, nov. 1970.
- [13] - A.R. Calderbank and N.J.A. Sloane, "FOUR-DIMENSIONAL MODULATION WITH AN EIGHT-STATE TRELLIS CODE", AT&T Tech. Jour., vol.64, pp. 1005 -1017, may-june 1985.
- [14] - L.F. Wei, "TRELLIS-CODED MODULATION WITH MULTIDIMENSIONAL CONSTELLATIONS", submitted to IEEE Trans. Information Theory, aug. 1985.

- [15] - A.R. Calderbank and N.J.A. Sloane, "AN EIGHT-DIMENSIONAL TRELLIS CODE", Proc. of IEEE, vol. 74, pp. 757-759 .
- [16] - G.D. Forney, Jr.; "COSET CODES I: GEOMETRY AND CLASSIFICATION, aug. 25, 1986.
- [17] - A. Gersho and V.B. Lawrence, "MULTIDIMENSIONAL SIGNAL CONSTELLATIONS FOR VOICEBAND DATA TRANSMISSION", IEEE Journal on Selected Areas in Communications, vol. SAC-2, n° 5, september 1984.
- [18] - N.J.A. Sloane, "BINARY CODES, LATTICES, AND SPHERE PACKINGS", in Combinatorial Surveys, P. J. Cameron, Ed. London and New York: Academic, 1977, pp. 117-164.
- [19] - Ray Wright - " TRELLIS CODED MODULATION FOR HIGH DATA RATES " - Motorola Information Systems LTD, UK - Communications International - september 1985.
- [20] - Peterson, W. W. and Weldon, E.S. - " ERROR CORRECTING CODES "- The Mit Press, Cambridge, Mass., U.S.A., 1972.
- [21] - S. Lin, and Costello, Jr. - " ERROR CONTROL CODING: FUNDAMENTALS AND APLICATIONS " . Prentice Hall, 1983
- [22] - Massey, J. L. - "CODING AND MODULATION IN DIGITAL COMMUNICATIONS" - Proc. 1974 Int. Zurich Seminar Digital Commun., Zurich, Switzerland, march 1974, p. E2.
- [23] - Palazzo, Jr., R. - "ANALISYS OF PERIODIC LINEAR AND NON LINEAR TRELLIS CODES " - Ph.D. Dissertation, University of California, Los Angeles, 1984.
- [24] - Forney, Jr. , G.D. - " THE VITERBI ALGORITHM " - Proceedings of the IEEE, vol 61,n.3, march 1973.

- [25] - Biglieri, E. - " UNGERBÖECK CODES DO NOT SHAPE THE
SINAL POWER SPECTRUM " - IEEE Trans. on Inf. Theory,
vol. IT - 32, july 1986, p. 595.
- [26] - Biglieri, E. - "HIGH LEVEL MODULATION AND CODING FOR
NON LINEAR SATELLITE CHANNELS " - IEEE Trans on
Commun., vol COM - 32 may 1984, p. 616.
- [27] - Portugheis, J. - " UMA CONTRIBUIÇÃO AO ESTUDO DE
MODULAÇÃO CODIFICADA M-PSK " - Tese de Mestrado,
Universidade Estadual de Campinas , S.P., março 1987.
- [28] - C.D. Forney Jr., R.G. Gallager, G. R. Lang, F.M.
Longstaff, and S.U. Qureshi, "EFFICIENT MODULATION FOR
BAND-LIMITED CHANNELS". IEEE J. Select. Areas
Commun., Vol. SAC- 2 pp. 632 - 647, 1984.
- [29] - Campopiano, C.N.; Glazer, B.G.; "A COHERENT DIGITAL AND
PHASE MODULATION SCHEME". IRE Transactions on
Communications Systems, november 1962.
- [30] - Markman, I., "ESQUEMAS DE MODULAÇÃO CODIFICADA EM TRÊS E
QUATRO DIMENSÕES". Tese de Mestrado, Unicamp - Campinas
1987.
- [31] - J.H. Conway and N.J. Sloane, "ON THE VORONOI REGION OF
CERTAIN LATTICES". Siam J. Algebraic Discret Methods, Vol
5, pp. 294 - 305, 1984.
- [32] - C.A. Rogers, "PACKING AND COVERING" Cambridge. The
University Press, 1964.

- [33] - J.W.S. Cassels, "AN INTRODUCTION TO THE GEOMETRY OF NUMBERS", New York: Springer-Verlag, 1971.
- [34] - H. S. M. Coxeter, "THE PROBLEM OF PACKING A NUMBER OF EQUAL NONOVERLAPING CIRCLES ON A SPHERE", Trans. New Acad. SCI, Series II, vol. 24, n^o 3, pp. 320-331, jan.62.
- [35] - Costelo, D.J., "CONSTRUCTION OF CONVOLUTIONAL CODES FOR SEQUENTIAL DECODING". Ph.D. thesis, University of Notre Dame, USA 1966.
- [36] - Peterson, W.W.; Weldon, E.S., "ERROR-CORRECTING CODES", The MIT Press, Cambridge Mass, USA 1972.
- [37] - Borelli, W.C. "CONVOLUTIONAL CODES FOR MULTINIVEL DATA TRANSMISSION". Ph.D. thesis, University of Kent at Canterbury, England june 1983.
- [38] - Marsan, M.A. et al., "THEORETICAL ASPECTS AND PERFORMANCE EVALUATION OF CHANNEL ENCODING WITH REDUNDANT MODULATOR ALPHABETS". Proc. GLOBECOM'84, Atlanta, USA 1984, p.688.
- [39] - Markman, I., Borelli, W.C., "SUBCONJUNTOS EFICIENTES DE CÓDIGOS DE ALFABETO REDUNDANTE PARA A PROCURA NÃO-EXAUSTIVA DE CÓDIGOS ÓTIMOS", 6^o SBT, Campina Grande PB set/89.
- [40] - Viterbi, A. J. " CONVOLUTIONAL CODES AND THEIR PERFORMANCE IN COMMUNICATION SYSTEMS ". IEEE Trans. on COMMUNICATIONS Technology, vol. com - 19. n. 5, october 1971.

- [41] - J.H. Conway and N.J. Sloane, "VORONOI REGIONS OF LATTICES, SECONDS MOMENTS OF POLYTOPES AND QUANTIZATION". IEEE Trans. Inf. Theory, Vol IT - 28 pp 211-226, 1982.
- [42] - Viterbi, A.J.; Omura, J.K., "PRINCIPLES OF DIGITAL COMMUNICATION AND CODING" - Mc Graw Hill Kogakusha, LTD 1979.
- [43] - Ungerböeck, G., "CHANNEL CODING WITH MULTILEVEL/PHASE SIGNAL" - IEEE on Inf. Theory, vol IT - 28, January 1982.
- [44] - Thapar, H. K. - " REAL - TIME APLICATION OF TRELLIS CODING TO HIGH-SPEED VOICEBAND DATA TRANSMISSION " - IEEE Journal on Selected Areas in Communications, vol.SAC - 2, september 1985, p. 648.
- [45] - Fonseca, K. O. - " UMA CONTRIBUIÇÃO A MODULAÇÃO PARA CÓDIGOS DE TRELIÇA PERIÓDICAMENTE VARIANDO NO TEMPO " - Tese de Mestrado, Universidade Estadual de Campinas, S.P., julho de 1988.
- [46] - Motoyama S., Borelli W. C., Palazzos Jr., R., Costa e Silva M., e O. G. C. Gonçalves - " ESTUDO DA VIABILIDADE PARA LOCXS - REDE LOCAL DE ACESSO INTEGRADO " Contrato TELEBRÁS 024/83 - RT- 37 -ATA.