ANÁLISE ESPECTRAL DE SINAIS DIGITAIS CODIFICADOS POR CÓDIGOS LINEARES OU CÓDIGOS DE BLOCO NÃO LINEARES COM O AUXÍLIO DE COMPUTADOR

JOÃO BATISTA BEZERRA
Orientador: DALTON SOARES ARANTES

Tese apresentada à Faculdade de Engenharia - FEC - UNICAMP, como parte dos requisitos exigidos para obtenção do título de MESTRE EM CIÊNCIAS.

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA DE CAMPINAS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

DEDICATÓRIA

Aos meus pais,
Luiz B da Silva e
Angelita F. Bezerra;

À minha esposa,
Luzia F. da Costa Bezerra;

Ao meu filho,
Cláudio Emanuel F.C. Bezerra.

AGRADECIMENTOS

Ao meu orientador, Prof. Dr. Dalton Soares Arantes, pela dedicação e sugestões dadas para solucionar as dificuldades surgidas durante a execução do trabalho;

À Universidade Federal do Rio Grande do Norte (UFRN), por ter me proporcionado esta oportunidade através do Departamento de Engenharia Elétrica do seu Centro de Tecnologia;

À TELEBRÁS pelo patrocínio dos trabalhos gráficos e de datilografia;

À Sra. Maria Júlia Dini Fray, pelos excelentes trabalhos de datilografia;

Aos Srs. Edson Pedro de Lima e Raimundo Nonato de Souza, pelos excelentes trabalhos de desenho;

A todos os colegas e amigos que, direta ou indiretamente, contribuiram para a realização deste trabalho.

ÍNDICE

SUMÁRIO

CAPÍTULO 1 - INTRODUÇÃO

1.1 - OBJETIVO DO TRABALHO
1.2 - TIPOS DE CÓDIGOS
1.3 - CÓDIGOS DE BLOCO
1.4 - CÓDIGOS DE ÁRVORE
1.5 - DISTÂNCIA DE HAMMING E PESO DE HAMMING

CAPÍTULO 2 - REVISÃO DE ÁLGEBRA

2.1 - GRUPOS
2.2 - ANÉIS
2.3 - CAMPOS
2.4 - SUB-GRUPOS E GRUPOS-FATOR
2.5 - ESPAÇOS VETORIAIS E ÁLGEBRAS LINEARES
2.6 - MATRIZES
2.7 - IDEAIS
2.8 - IDEAIS E CLASSES DE RESÍDUOS DE INTEIROS
2.9 - IDEAIS E CLASSES DE RESÍDUOS DE POLINÔMIOS
2.10 - A ÁLGEBRA DAS CLASSES DE RESÍDUOS DE POLINÔMIOS

CAPÍTULO 3 - MÁQUINAS SEQUENCIAIS DE ESTADO FINITO

3.1 - CIRCUITOS SEQUENCIAIS
3.2 - REPRESENTAÇÃO DE UMA MÁQUINA SEQUENCIAL SÍNCRONA
3.3 - ESPECIFICAÇÃO DO COMPORTAMENTO DE UMA MÁQUINA
3.4 - MÁQUINAS DETERMINÍSTICAS
3.5 - MÁQUINAS SEQUENCIAIS LINEARES
3.6 - SHIFT-REGISTERS
3.7 - O SHIFT-REGISTER COMO UM MULTIPLICADOR DE POLINÔMIOS
3.8 - MÁQUINAS INVERSAS
3.9 - O SHIFT-REGISTER COMO UM CIRCUITO DIVISOR DE POLINÔMIOS
3.10 - MÁQUINAS LINEARES COM ENTRADAS E SAÍDAS MÚLTIPLAS
3.11 - MATRIZ DE TRANSFERÊNCIA DE UMA MÁQUINA SEQUENCIAL LINEAR

.../.

CAPÍTULO 4 - CÓDIGOS DE BLOCO LINEARES

4.1 - INTRODUÇÃO
4.2 - DESCRIÇÃO DE CÓDIGOS DE BLOCO LINEARES POR MATRIZES
4.3 - ESTRUTURA ALGÉBRICA DOS CÓDIGOS DE BLOCO LINEARES
4.4 - CODIFICAÇÃO DE CÓDIGOS DE BLOCO LINEARES
4.5 - CÓDIGOS CÍCLICOS
4.6 - CODIFICAÇÃO DE CÓDIGOS CÍCLICOS

CAPÍTULO 5 - CÓDIGOS CONVOLUCIONAIS

5.1 - DESCRIÇÃO DE CÓDIGOS DE ÁRVORE POR MATRIZES
5.2 - ESTRUTURA DOS CÓDIGOS CONVOLUCIONAIS
5.3 - CODIFICADORES PARA CÓDIGOS CONVOLUCIONAIS
5.4 - O CODIFICADOR DE CÓDIGOS CONVOLUCIONAIS SOB O PONTO DE VISTA DA TEORIA DAS MÁQUINAS LINEARES DE ESTADO FINITO
5.5 - CÓDIGOS DE BLOCO LINEARES COMO UM CASO PARTICULAR DOS CÓDIGOS CONVOLUCIONAIS

CAPÍTULO 6 - ESPECTRO DE SINAIS DIGITAIS CODIFICADOS POR CÓDIGOS DE BLOCO

6.1 - INTRODUÇÃO
6.2 - ESPECIFICAÇÕES DO CODIFICADOR
6.3 - O SINAL DIGITAL
  6.3.1 - A MENSAGEM CODIFICADA
  6.3.2 - ESTATÍSTICA DA MENSAGEM CODIFICADA
  6.3.3 - DENSIDADE ESPECTRAL DO SINAL DIGITAL
6.4 - PROBABILIDADES E CORRELAÇÕES DAS PALAVRAS-CÓDIGO
6.5 - FÓRMULA EXATA PARA A DENSIDADE ESPECTRAL
6.6 - COMPORTAMENTO EM FREQUÊNCIA

CAPÍTULO 7 - O PROGRAMA E RESULTADOS

7.1 - GENERALIZAÇÃO DO MÉTODO DE CARIOLARO
7.2 - CONSTRUÇÃO DAS MATRIZES $\bar{A}$ e $\bar{E}$ PARA CÓDIGOS CONVOLUCIONAIS
7.3 - CONSTRUÇÃO DAS MATRIZES $\bar{A}$ e $\bar{E}$ PARA CÓDIGOS DE BLOCOS LINEARES

.../.

7.4 - CONSTRUÇÃO DAS MATRIZES $\bar{A}$ e $\bar{E}$ PARA CÓDIGOS DE BLOCOS NÃO LINEARES
7.5 - CÓDIGOS UTILIZADOS PARA TESTAR A PROGRAMAÇÃO QUE DETERMINA $X_c(f)$ e $X_d(f)$
7.6 - ALGUNS NOVOS RESULTADOS
7.7 - OUTROS RESULTADOS
7.8 - COMPARAÇÃO ENTRE CÓDIGOS
7.9 - ESPECTRO DE UM CÓDIGO CONVOLUCIONAL
7.10 - CONCLUSÕES

APÊNDICE A - RESUMO COMPUTACIONAL DAS CARACTERÍSTICAS DOS SUB-PROGRAMAS UTILIZADOS

APÊNDICE B - FLUXOGRAMA DOS PROGRAMAS E SUB-PROGRAMAS IMPLEMENTADOS

APÊNDICE C - LISTAGENS DOS PROGRAMAS E SUB-PROGRAMAS IMPLEMENTADOS

BIBLIOGRAFIA

...

SUMÁRIO

O problema da análise espectral se sinais digitais que apresentem características apropriadas para transmissão digital em banda-base vem sendo pesquisado desde o trabalho pioneiro de Bennett em 1958. Apesar da existência de muitos trabalhos sobre o assunto, poucos são os que tratam o problema com o auxílio da estrutura algébrica apropriada,a Teoria da Automação. Em 1974, Cariolaro e Tronca publicaram um trabalho que trata da análise espectral de sinais digitais codificados por códigos de bloco, utilizando a Teoria da Automação e de uma forma apropriada para o uso do computador digital na resolução do problema. Naquele trabalho, ênfase é dada aos códigos de bloco não lineares e assim os passos ali sugeridos para a implementação de um possível programa é apropriada apenas para essa classe de códigos.

Neste trabalho, usaremos a forte estrutura matemática de que dispõem os códigos lineares e, juntamente com a Teoria das Máquinas Sequencias Lineares. obteremos relações matemáticas que permitirão, após pequenas modificações nas sugestões de Cariolaro, implementar programas que determinarão ao propriedades espectrais de códigos lineares como também de códigos de bloco não lineares. Como resultados de aplicação mais imediata, obteremosas propriedades espectrais de alguns novos códigos sugeridos para transmissão por fibras óticas.

CAPÍTULO 1

INTRODUÇÃO

1 - OBJETIVO DO TRABALHO

Devido ao grande desenvolvimento observado nas últimas décadas nos sistemas de comunicações digitais, tem-se observado um crecente aumento no volume de dados a serem processados, como também maiores exigências com relação à precisão dos resultados a serem obtidos.

É através do estudo da "Teoria da Codificação" que se procura solucionar, na medida do possível, esses tipos de problemas, pois ela permite que se construam códigos capazes de corrigir ou detectar erros ou que façam as duas coisas simultaneamente, além de proporcionarem uma melhor utilização do meio de transmissão, isto é, uma melhor utilização das características do canal.

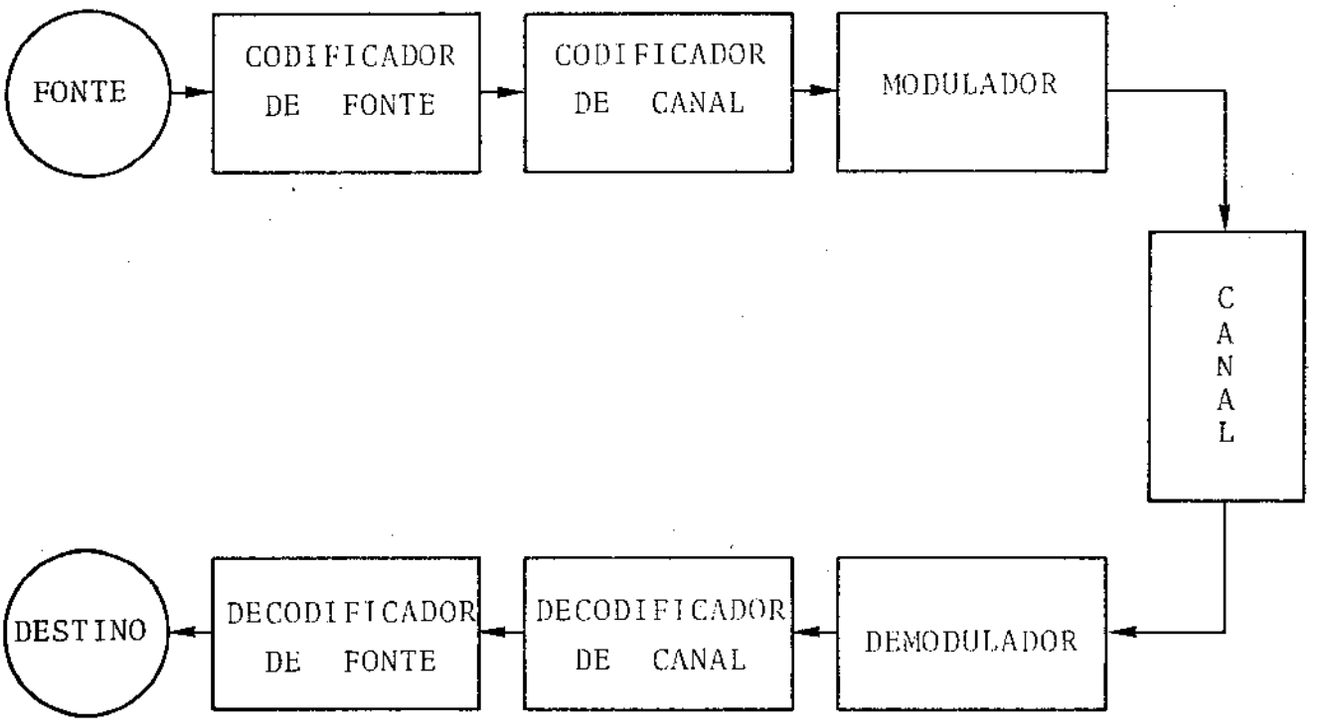


Fig. 1.1 - Representação, em diagrama de blocos, de um sistema que utiliza codificação

Na Fig. 1.1 está representado, em diagrama de blocos, um sistema que utiliza codificação. O gerador ou fonte de informação fornece o fluxo de informação sob a forma de mensagens a serem processadas pelo sistema. Essa fonte pode ser contínua ou discreta, dependendo apenas das características do sistema. Por exemplo, se a esquematização da Fig. 1.1 representar um sistema telefônico, a fonte será do tipo contínuo, enquanto que se representar um sistema

de teleprocessamento, a fonte será do tipo discreto. O fluxo de informação fornecido pela fonte alimenta a entrada de "codificador de fonte" que tem por finalidade transformar o fluxo de informação da fonte em um fluxo de informação com símbolos discretos p-ários, isto é, na saída do "codificador de fonte" temos uma réplica do fluxo de informação na forma de uma sequência de símbolos discretos p-ários. Por exemplo, se a Fig. 1.1 representar um sistema de comunicações digitais do tipo PCM, o codificador de fonte corresponderá ao conversor analógico-digital binário, p=2 (conversor A/D).Neste caso, a sequência binária que alimenta a entrada do"codificador de canal" é, em geral, transformada em uma sequência q-ária, $q \geq 2$, provavelmente utilizando um maior número de dígitos q-ários para representar cada dígito binário. Diz-se então que o "codificador de canal" introduz redundância e é essa redundância que permite a detetção, correção, ou ambas, de erros que ocorrem durante a transmissão da sequência através do canal.

A sequência codificada alimenta a entrada do modulador , cuja finalidade é transformar grupos de dígitos codificados em formas de onda (símbolos) aceitáveis pelo meio de transmissão, o canal. Assim, na saída do modulador, tem-se uma sequência de símbolos que são enviados através do canal.

Devido às imperfeições do canal, tais como ruído, distorção e mesmo pela interferência de canais adjacentes, os símbolos são modificados durante a transmissão, de modo que na saída do demodulador, cuja função é inversa à do modulador, obtém-se uma forma corrompida da sequência codificada. Essa sequência corrompida alimenta a entrada do "decodificador de canal" que a partir dessa sequência e normalmente através de um processo decisório estatístico, fornece em sua saída uma réplica da sequência q-ária codificada. Em geral, esse processo decisório é feito de modo que a probabilidade de erro na sequência codificada seja a mínima possível. A sequência decodificada alimenta o decodificador de fonte, cuja saída é uma réplica aproximada da mensagem transmitida;é essa réplica aproximada da mensagem transmitida que chega ao destino da informação. No caso de sistemas de comunicações digitais como o PCM , por exemplo, o conversor digital-analógico binário (conversor D/A) corresponde ao decodificador de fonte.

No diagrama de blocos da Fig. 1.1, o canal, devido suas imperfeições e principalmente por suas limitações quanto à quantidade de informação que pode ser transmitida por unidade de tempo ,

é um dos fatores de grande importância para a escolha do código a ser utilizado durante a codificação de canal. Uma das limitações mais comuns é a limitação em frequência, o que requer o conhecimento das propriedades espectrais do sinal codificado. Além dessa limitação em frequência, outros fatores inerentes às propriedades físicas de cada canal, impõem que o sinal codificado tenha propriedades espectrais específicas; é o caso por exemplo dos sistemas de transmissão em banda-base, isto é, sem modulação e demodulação.

Em certas aplicações, tais como transmissão digital em cabos multipares, coaxiais ou fibras óticas, torna-se necessária a utilização de códigos detetores de erros para a telesupervisão dos repetidores regenerativos. Neste caso, a transmissão é feita em banda-base e a escolha do código é crucial para o bom desempenho do sistema.

Com a perspectiva da utilização da fibra ótica como meio de transmissão em banda-base, muitas pesquisas têm sido realizadas com o fim de obter códigos que tenham as propriedades exigidas pelo referido meio de transmissão [5]. Uma dessas exigências está relacionada com o comportamento em frequência, o que implica em determianr o espectro de potência do sinal codificado para cada código que esteja sendo pesquisado. Assim, tendo em vista fornecer uma ferramenta que venha facilitar a pesquisa de códigos adequados , nos propusemos a implementar um programa para computador digital que possa fornecer as propriedades espectrais do código a ser estudado.

Nos fundamentamos num trabalho escrito por CARIOLARO e TRONCA [2] para uma classe particular de códigos e mais precisamente para os códigos analisados em [7]. Entretanto, mostraremos que as hipóteses assumidas em [2] não impõem quaisquer restrições a outros tipos de códigos e assim, através de uma associação entre a Teoria da Codificação, a Teoria das Máquinas Sequenciais de Estado Finito e de algumas relações por nós obtidas, foi possível a implementação de um programa geral.

O trabalho está estruturado da sequinte forma:

O restante deste capítulo é dedicado aos conceitos básicos da Teoria da Codificação. No capítulo 2 fazemos uma revisão de álgebra que permitirá uma melhor compreensão dos capítulos 4 e 5 , que tratam dos códidos lineares. No capítulo 3 apresentamos conceitos básicos da Teoria das Máquinas Sequenciais Síncronas. No capítulo 6 apresentaremos o trabalho de Cariolaro de uma forma mais de

talhada. No capítulo 7 são apresentados resultados existentes na literatura para alguns códigos bem conhecidos, além de novos resultados aqui obtidos com novos códigos propostos para transmissão por fibra ótica. Finalmente, apresentamos nos apêndices A, B e C um resumo dos programas e sub-programas implementados, dos fluxogramas e das listagens dos referidos programas.

2 - TIPOS DE CÓDIGOS

Levando-se em consideração a forma como o codificador de canal processa a sequência de informação, existem basicamente dois tipos de códigos: CÓDIGOS DE BLOCO E CÓDIGOS DE ÁRVORE.

Consideremos a sequência de informação a_0, a_1, a_2..... $a_i \in\{0,...p-1\}$ alimentando o codificador de canal da Fig. 1.1 . O índice i indica a dependência temporal. Se o código utilizado durante a codificação for um CÓDIGO DE BLOCO, o codificador processa a sequência de informação da seguinte forma;

A sequência é dividida em blocos de k dígitos de informação da forma $[a_0\ a_1 \cdots a_{k-1}][a_k\ a_{k+1} \cdots a_{2k-1}]\cdots$ pelo armazenamento de k dígitos na memória do codificador. Em seguida, cada bloco de k dígitos é processado independentemente, de acordo com as leis de formação do código, de modo que na saída do codificador obtém-se um bloco de n dígitos q-ários, n satisfazendo à seguinte desigualdade:

$$n \geqslant \frac{k \log p}{\log q} \qquad (1.1)$$

Assim, na saída do codificador obtém-se uma sequência da forma $[c_0\ c_1 \cdots c_{n-1}][c_n\ c_{n+1} \cdots c_{2n-1}]\cdots$. Cada bloco de n dígitos q-ários é denominado de PALAVRA-CÓDIGO. As palavras-código são enviadas através do canal, corrompidas pelo ruído e pelas imperfeições do canal de modo que a sequência na entrada do decodificador é da forma $[r_0\ r_1 \cdots r_{n-1}][r_n\ r_{n+1} \cdots r_{2n-1}]\cdots$ .O decodificador faz sua tomada de decisão e fornece em sua saída uma estimativa da sequência de informação, digamos $\hat{a}_0\ \hat{a}_1\ \hat{a}_2 \cdots$.

Quanto aos códigos de árvore, o procedimento é semelhante. A sequência de informação é dividida em blocos de k_0 dígitos p-ários e na saída do codificador obtém-se uma sequência q-ária dividida em blocos de n_0 dígitos

$$n_0 \geq \frac{k_0 \log p}{\log q} \qquad (1.2)$$

onde normalmente n_0 e k_0 são números pequenos. A diferença no processamento para os códigos de bloco e os códigos de árvore é que nos códigos de árvore, cada bloco de n_0 dígitos depende não só do bloco de informação de k_0 dígitos atual como também de todos os blocos de informação processados anteriormente. A dependência na codificação com códigos de árvore com relação a todos os blocos de k_0 dígitos de informação exige que o codificador tenha uma memória infinita, o que é impossível sob o ponto de vista prático; daí a pouca utilização dos códigos de árvore. Entretanto, existe uma subclasse dos códigos de árvore, os CÓDIGOS CONVOLUCIONAIS, que se caracterizam por possuirem uma dependência em um número finito de blocos de k_0 dígitos de informação anteriormente processados. Isto é, cada bloco de n_0 dígitos depende do bloco de k_0 dígitos atual e dos m-1 blocos de k_0 dígitos anteriormente processados de modo que a dependência é determinada através de um total de m blocos de informação de k_0 dígitos. Neste caso, o codificador possui uma memória finita e o número m é denominado de MEMÓRIA DO CÓDIGO.

Os códigos de bloco são divididos em duas classes: CÓDIGOS DE BLOCO LINEARES e CÓDIGOS DE BLOCO NÃO LINEARES. Os códigos convolucionais são códigos lineares. Na realidade, os códigos de bloco lineares são um caso particular dos códigos convolucionais com m=1. Esse fato será usado por nós na obtenção de resultados que facilitarão a nossa programação com o fim de determinar o espectro de potência de códigos de bloco lineares. Com relação à estrutura, são os códigos de bloco lineares que apresentam uma melhor estrutura matemática. Apesar dos códigos convolucionais ainda não possuirem uma estrutura matemática forte, eles podem ser facilmente compreendidos a partir da estrutura dos códigos de bloco lineares. Com relação ao desempenho, os códigos convolucionais são tão bons quanto os códigos de bloco em termos de capacidade corretora e de implementação do codificador e decodificador.

1.3 - CÓDIGOS DE BLOCO

DEFINIÇÃO 1.1 - Um código de bloco é um conjunto formado de M sequências q-árias de comprimento n. Cada sequência de comprimento n é uma palavra-código e n é denominado de comprimento do código.

Foi dito na seção 1.1 que o processo de decisão tomado pelo decodificador é um processo estatístico. Para códigos de bloco, esse processo pode ser compreendido através da construção de uma tabela denominada de TÁBUA DE DECODIFICAÇÃO. A construção dessa tábua é feita da seguinte maneira:

1) As palavras-código são dispostas em uma mesma linha.

2) Abaixo de cada palavra-código são colocadas as n-uplas mais semelhantes à respectiva palavra-código, isto é, abaixo de cada palavra-código são colocadas aquelas n-uplas mais prováveis de serem recebidas quando aquela palavra-código for transmitida.

3) Uma n-upla só pode aparecer uma única vez na tabela.

Após a construção dessa tabela, dispomos de uma listagem de todas as q^n n-uplas q-árias, todas possíveis de serem recebidas. A decodificação se processa da seguinte forma:

Identifica-se a n-upla recebida na tabela e assume-se que a palavra-código transmitida é aquela que encabeça a coluna onde a n-upla recebida está situada.

EXEMPLO 1 - Seja um código binário constituido por quatro palavras-código: 11000, 00110, 10011 e 01101. Observemos que para esse código, q=2, n=5 e k=2, já que M=4. Uma possível tábua de decodificação para esse código está apresentada na Fig. 1.2.

Palavras código	11000	00110	10011	01101
	11001	00111	10011	01100
	11010	00100	10001	01111
	11100	00010	10111	01001
	10000	01110	11011	00101
	01000	10110	00011	11101
	.....	.....	.....	.....
	11110	00000	01011	10101
	01010	10100	11111	00001

Fig. 1.2 - Uma possível tábua de decodificação para o código do exemplo 1

Analisando a tabela da Fig. 1.2,observamos que abaixo de cada palavra-código e até a linha tracejada estão todas as palavras possíveis de serem recebidas que diferem da palavra-código em apenas um dígito. Esse fato é bastante significativo, já que,utilizando o processo de decodificação descrito no parágrafo anterior ao exemplo 1, esse código pode corrigir todos os erros simples, isto é, erros de um dígito que tenham ocorrido durante a transmissão. Diz-se então que esse código tem capacidade corretora t=1.

DEFINIÇÃO 1.2 - Um código tem capacidade corretora t quando é capaz de corrigir todos os t ou menos erros possíveis de ocorrerem durante a transmissão.

Continuando com a análise da tabela da Fig. 1.2, observamos que as duas últimas palavras possíveis de serem recebidas , em cada coluna, diferem da correspondente palavra-código de dois dígitos. Isso implica em dizer que apenas alguns padrões de erros duplos podem ser corrigidos com esse código.

EXEMPLO 2 - Suponhamos que seja transmitida a palavra-código 00110 e que seja recebida a palavra 10100. Então o codificador assume que foi transmitido 00110 e assim um padrão de erro duplo foi corrigido. Entretanto, se a aplavra recebida for 10101,o decodificador assume que 01101 foi transmitida e portanto é cometido um erro de decodificação.

..4 - CÓDIGOS DE ÁRVORE

A razão para o nome "código de árvore" é que a codificação de uma sequência de informação pode ser realizada utilizando-se um diagrama que apresenta a configuração de uma árvore.Essa árvore é constituida por nós e ramos. Para um código (n_0,k_0) com dígitos de informação p-ários, os nós são espaçados de n_0 dígitos e de cada nó emanam p^{k0} ramos. A cada ramo da árvore está associado um bloco de n_0 dígitos de tal forma que ao conjunto dos ramos que emanam dos nós de mesma ordem está associado um código de bloco.Na Fig. 1.3 está esquematizado o modelo de uma árvore p-ária para um código (n_0,k_0).

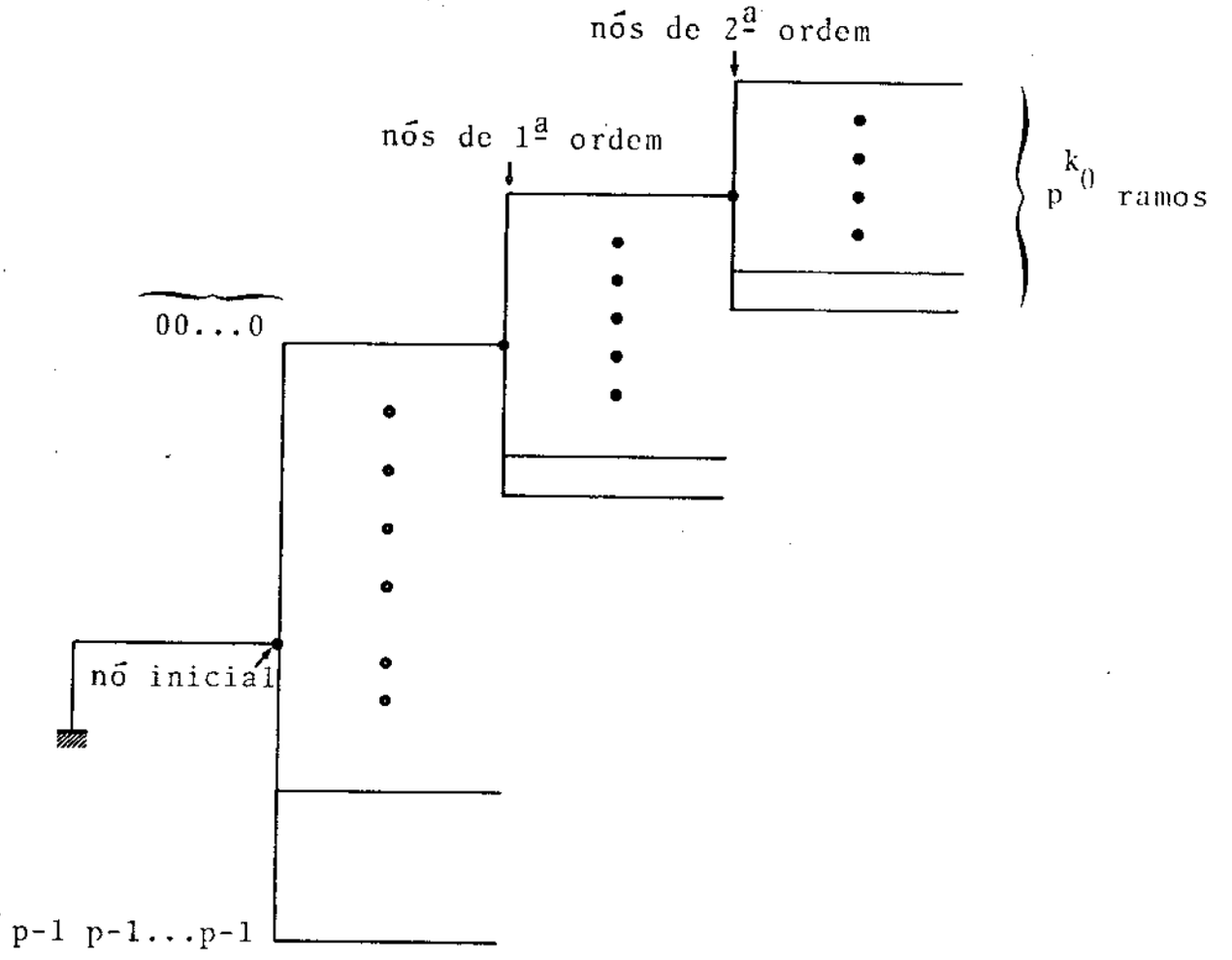


Fig. 1.3 - Modelo geral para uma árvore de um código de árvore (n_0, k_0)

A título de ilustração, consideremos a árvore binária associada a um código de árvore em que $n_0=3$ e $k_0=1$. Na Fig. 1.4 está representada a árvore associada a esse código. Consideremos agora a sequência de informação (0 1 0 1...) que deve ser codificada. Obedecendo a uma convenção, arbitrária, para percorrer os ramos da árvore, por exemplo, se "1" é recebido segue-se pelo ramo inferior e se "0" é recebido segue-se pelo ramo superior, obteremos a correspondente sequência codificada (000 111 001 101...). O processo segue indefinidamente. Na Fig. 1.4 está indicado o caminho seguido na codificação da sequência.

Em geral as árvores são infinitas; entretanto quando o código de árvore é convolucional a partir de um certo ponto os blocos de comprimento n_0 associados aos ramos dos nós de (m+1) - ésima

ordem começam a repetir-se o que caracteriza o fato dos códigos convolucionais serem uma classe especial dos códigos de árvore.

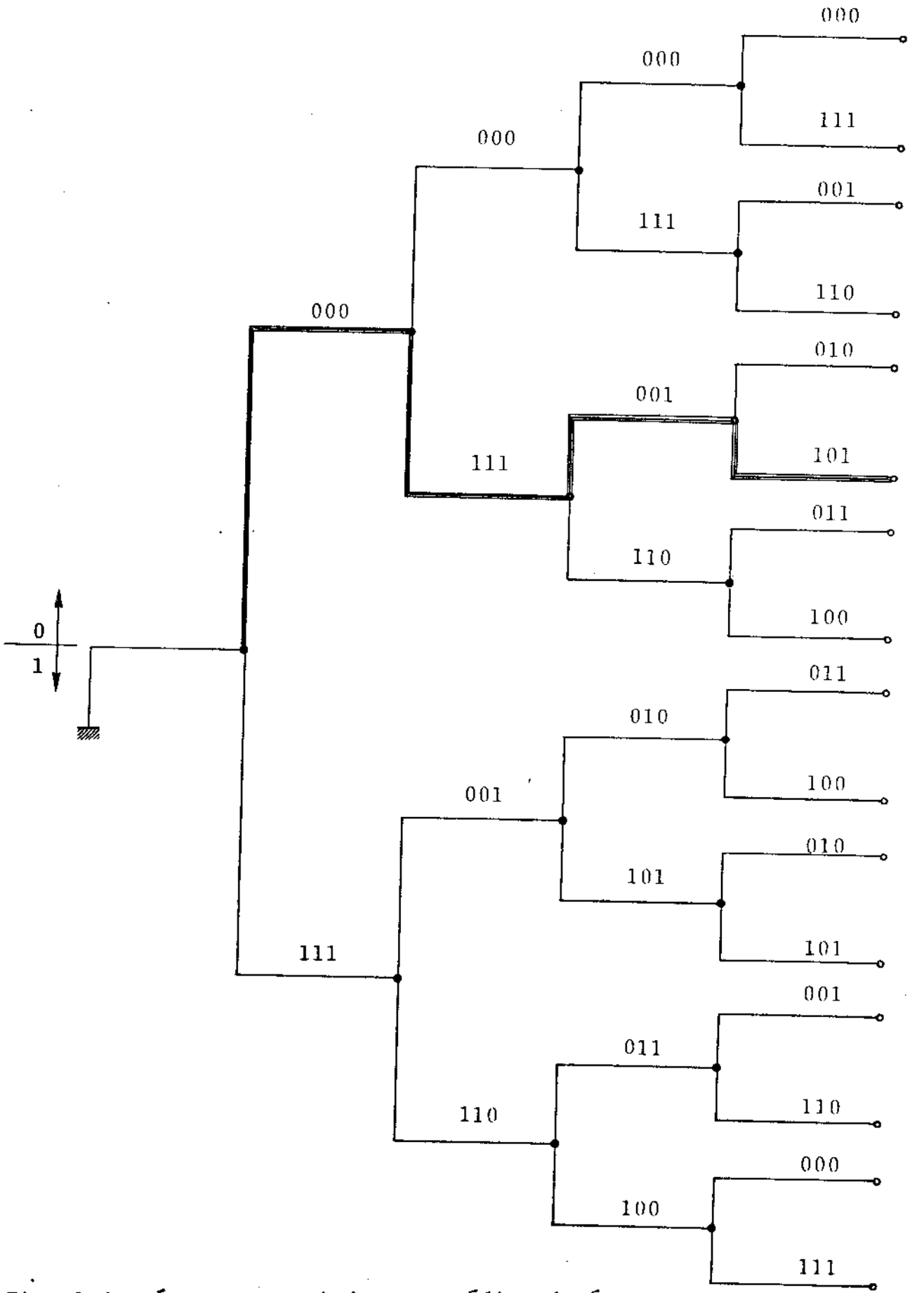


Fig. 1.4 - Árvore associada a um código de árvore em que $n_0=3$ e $k_0=1$.

1.5 - DISTÂNCIA DE HAMMING E PESO DE HAMMING

Durante a transmissão de uma palavra-código, é possível que o ruído do canal a transforme em uma outra e, nesse caso, apesar de erros terem sido cometidos, o decodificador não detetará qualquer erro, pois para ele o que está chegando é uma palavra - código sem qualquer alteração. Quando da escolha ou construção de um código, esse é um fator que deve ser observado com bastante atenção, isto é, um código deve ser escolhido de tal forma que a probabilidade do ruído de canal modificar uma palavra-código em outra seja mínima e isto é conseguido pela escolha de palavras-código bem diferentes. Uma das formas de medir quão diferentes são duas palavras-código é utilizar o conceito de DISTÂNCIA DE HAMMING.

DEFINIÇÃO 1.4 - Sejam dois vetores $\underset{\sim}{a}$ e $\underset{\sim}{b}$ de n componentes. A distância de Hamming entre $\underset{\sim}{a}$ e $\underset{\sim}{b}$, $d(\underset{\sim}{a},\underset{\sim}{b})$, é dada pelo número de componentes em que os dois vetores diferem.

EXEMPLO 5 - A distância de Hamming entre (0 1 0 1) e (1 1 1 1) é 2. Eles diferem na primeira e na terceira componentes.

DEFINIÇÃO 1.5 - Seja um vetor $\underset{\sim}{v}$. O peso de Hamming de $\underset{\sim}{v}$, $w(\underset{\sim}{v})$ é definido como sendo o número de componentes não nulas de $\underset{\sim}{v}$.

Os conceitos de distância de Hamming e peso de Hamming desempenham papel fundamental na teoria da codificação, em particular, a capacidade corretora de um código está relacionada com distância de Hamming [9] através da relação

$$d_{min} = 2t + 1 \qquad (1.3)$$

onde d_{min} é a menor distância de Hamming entre as palavras- código do código e t sua capacidade corretora. Em se tratando de códigos lineares, sua estrutura permite que sejam construídos códigos com as propriedades corretoras desejadas [8,9], Não é preocupação nossa neste trabalho a construção de códigos; entretanto nos próximos capítulos apresentaremos um pouco da estrutura dos códigos lineares com o fim de mostrar como os codificadores dos códigos lineares são obtidos como uma consequência de sua estrutura matemática.

CAPÍTULO 2

REVISÃO DE ÁLGEBRA

Neste capítulo fizemos uma revisão dos conceitos da Álgebra que são utilizados na estruturação dos códigos lineares. Muitos teoremas serão simplesmente enunciados; entretanto, daremos as provas daqueles que estão diretamente relacionados com o problema de estruturação dos códigos lineares. O desenvolvimento algébrico será feito apenas até o ponto de interesse para nosso trabalho. Em particular, não estamos interessados no problema da decodificação e assim serão desenvolvidos apenas os tópicos mínimos necessários para a compreensão do problema da codificação.

2.1 - GRUPOS

DEFINIÇÃO 2.1 - Um grupo G é um conjunto de objetos ou elementos para os quais uma operação é definida e são satisfeitos os seguintes axiomas:

AXIOMA G.1 (FECHAMENTO) - Se a operação do grupo é aplicada a quaisquer dois elementos pertencentes ao grupo, então o resultado também pertence ao grupo.

AXIOMA G.2 (ASSOCIATIVIDADE) - Para quaisquer três elementos pertencentes ao grupo temos:

$$(a+b)+c = a+(b+c) \qquad (2.1a)$$

$$(a.b).c = a.(b.c) \qquad (2.1b)$$

se a operação do grupo for respectivamente a adição ou a multiplicação (*).

AXIOMA G.3 (IDENTIDADE) - Existe um elemento identidade. Se a operação do grupo é chamada de adição, o elemento identidade é denominado de "zero", escrito 0 e é definido por:

$$0+a = a+0 \qquad (2.2a)$$

para cada elemento a pertencente ao grupo. Se a operação do grupo é chamada de multiplicação, o elemento identidade é chamado "um", escrito 1 e definido por:

$$1.a = a.1 \qquad (2.2b)$$

para todo a pertencente ao grupo.

(*) Adição e multiplicação aqui não significa, necessariamente, a adição e a multiplicação ordinárias.

AXIOMA G.4 (INVERSO) - Cada elemento do grupo possui um inverso. Se a operação do grupo é a adição, o inverso do elemento a é denotado -a e é definido pela equação:

$$a + (-a) = (-a) + a = 0 \qquad (2.3a)$$

Se a operação do grupo é a multiplicação, o inverso do elemento é denotado por a^{-1} e é definido pela equação:

$$a^{-1} \cdot a = a \cdot a^{-1} = 1 \qquad (2.3b)$$

DEFINIÇÃO 2.2 - Um grupo G é dito Abeliano ou Comutativo, se para os axiomas G.1 a G.4 satisfeitos, a lei comutativa é também satisfeita, isto é, para quaisquer dois elementos a e b pertencentes ao grupo,

$$a+b = b+a \qquad (2.4a)$$

$$a.b = b.a \qquad (2.4b)$$

se a operação do grupo for respectivamente a adição ou a multiplicação.

TEOREMA 2.1 - O elemento identidade em cada grupo é unico. O inverso de cada elemento do grupo é único.

2.2 - ANÉIS

DEFINIÇÃO 2.3 - Um anel R é um conjunto de elementos para os quais são definidas duas operações. Uma chamada adição e denotada por + e a outra chamada multiplicação e denotada por . ou x, não necessariamente a adição e a multiplicação ordinárias, e satisfazendo aos seguintes axiomas:

AXIOMA R.1 - O conjunto R é um grupo Abeliano sobre a adição.

AXIOMA R.2 (FECHAMENTO EM RELAÇÃO À MULTIPLICAÇÃO) - Para quaisquer dois elementos a e b pertencentes a R, o produto a . b é definido e pertence a R.

AXIOMA R.3 (ASSOCIATIVIDADE) - Para quaisquer três elementos a, b e c pertencentes a R,

$$a.(b.c) = (a.b).c \qquad (2.5)$$

AXIOMA R.4 (DESTRIBUTIVIDADE) - Para quaisquer três elementos a,b e c pertencentes a R,

$$a.(b+c) = a.b + a.c \qquad (2.6a)$$

$$(b+c).a = b.a + c.a \qquad (2.6b)$$

DEFINIÇÃO 2.4 - Um anel R é dito comutativo, se a operação multiplicação definida para R for também comutativa. Isto é, se para quaisquer a e b pertencentes a R,

$$a.b = b.a \qquad (2.7)$$

2.3 - CAMPOS

DEFINIÇÃO 2.5 - Um campo F é um anel comutativo com o elemento unidade, a identidade da multiplicação, onde cada elemento diferente de zero possui um inverso multiplicativo.

2.4 - SUB-GRUPOS E GRUPOS-FATOR

DEFINIÇÃO 2.6 - Um sub-conjunto H com elementos pertencentes a um grupo G é denominado de sub-grupo H, se satisfaz a todos os axiomas de grupo com relação à mesma operação definida no grupo G.

Para verificar se H é um sub-grupo, é necessário verificar apenas os axiomas do fechamento e do inverso. Isso porque se um conjunto H é fechado com relação à operação de um grupo G e o inverso do grupo está presente em H, a identidade obrigatoriamente está presente em H e a associatividade é sempre válida no sub-grupo se ela é válida no grupo.

Sejam g_1, g_2, g_3,..., os elementos de um grupo G e h_1, h_2, h_3,..., os elementos de um sub-grupo H obtido de G. Consideremos o arranjo formado da seguinte maneira:

1) A primeira linha é constituída pelos elementos do sub-grupo H com a identidade colocada na posição mais à esquerda e cada elemento pertencente a H aparecendo uma e somente uma vez.

2) O primeiro elemento da segunda linha é qualquer elemento pertencente a G que não esteja na primeira li

nha. Os elementos restantes da segunda linha são obtidos aplicando-se a operação do grupo G entre o primeiro elemento e cada um dos elementos pertencentes ao grupo H.

3) O primeiro elemento da terceira linha é qualquer elemento pertencente a G que não esteja na primeira ou segunda linha. Os elementos restantes são obtidos aplicando-se a operação do grupo entre o primeiro elemento e cada um dos elementos pertencentes a H.

O processo se repete para cada linha, sempre se usando para primeiro elemento da linha um elemento pertencente a G que não tenha ainda aparecido no arranjo, até que todos os elementos de G estejam presentes em algum lugar do arranjo. Na Fig. 2.1 fizemos uma apresentação do processo para um grupo cuja operação é a multiplicação.

$$
\begin{array}{llllll}
h_1=1, & h_3, & h_3, \ldots\ldots, & h_n \\
g_1 & , g_1h_2, & g_1h_3, \ldots\ldots, & g_1h_n \\
g_2 & , g_2h_2, & g_2h_3, \ldots\ldots, & g_2h_n \\
\vdots & \vdots & \vdots & \vdots \\
g_m & , g_mh_2, & g_mh_3, \ldots\ldots, & g_mh_n
\end{array}
$$

Fig. 2.1 - Decomposição de um grupo G em co-sets.

DEFINIÇÃO 2.7 - O conjunto formado pelos elementos de uma linha do arranjo descrito anteriormente é denominado de CO-SET À ESQUERDA. Co-sets à direita também podem ser formados, bastando apenas que cada elemento do sub-grupo seja operado à direita. O arranjo descrito é denominado de DECOMPOSIÇÃO DO GRUPO EM CO-SETS e o primeiro elemento em cada linha é denominado de LÍDER DO CO-SET.

TEOREMA 2.2 - Dois elementos g e g' pertencentes a um grupo G estão no mesmo co-set à esquerda de um sub-grupo H de G se, e somente se, $g^{-1}.g'$ pertence a H.

PROVA DA PROPOSIÇÃO DIRETA

"Se g e g' pertencem a G e estão no mesmo co-set à esquerda, então g^{-1} g' pertence a H."

Seja g_i o líder do co-set ao qual pertencem g e g'. Então, para algum j, $g = g_i h_j$ e para algum k, $g' = g_i h_k$ por construção. Assim,

$$g^{-1}.g' = (g_i h_j)^{-1}.g_i h_k \qquad (2.8a)$$

Mas

$$(g_i h_j)^{-1} = h_j^{-1}.g_i^{-1}$$

já que $$(g_i h_j).(h_j^{-1}\ g_i^{-1}) = g_i(h_j h_j^{-1})g_i^{-1} = g_i g_i^{-1} = 1\ . \qquad (2.8b)$$

Assim a equação (2.8a) pode ser escrita como:

$$g^{-1}g' = h_j^{-1}.g_i^{-1}.g_i.h_k = h_j^{-1}.(g_i^{-1}g_i)h_k = h_j^{-1}h_k \qquad (2.8c)$$

Pelos axiomas G.3 e G.4 concluimos que $g^{-1}g'$ é um elemento pertencente a H.

PROVA DA PROPOSIÇÃO INVERSA

"Se $g^{-1}.g'$ pertence ao sub-grupo H, então g e g' estão no mesmo co-set à esquerda."

Seja g_i o líder do co-set ao qual pertence g, então por construção

$$g = g_i h' \qquad (2.8d)$$

para algum h' pertencente a H.

Por hipótese,

$$g^{-1}g' = h \qquad (2.8e)$$

para algum h pertencente a H.

Multiplicando 2.8e à esquerda por g, usando o axioma G.4 e a equação 2.8d obtemos

$$g' = g_i hh' \qquad (2.8f)$$

Portando g' e g estão no mesmo co-set à esquerda.

Q.E.D.

TEOREMA 2.3 - Cada elemento pertencente a um grupo G está em um e somente um co-set de um sub-grupo H de G.

PROVA

Pela construção do arranjo da Fig. 2.1, cada elemento pertencente ao grupo G aparece pelo menos uma vez no arranjo. Devemos mostrar então que cada elemento pertencente a G aparece somente uma vez no arranjo. Primeiro suponhamos que dois elementos iguais estão na mesma linha do arranjo, digamos $g_i h_j$ e $g_i h_k$, isto é,

$$g_i h_j = g_i h_k \tag{2.9a}$$

Multiplicando a equação (2.9a) à esquerda por g_i^{-1} obtemos

$$h_j = h_k \tag{2.9b}$$

Como por construção do arranjo os elementos pertencentes a H aparecem apenas uma vez, (2.9b) é uma contradição e assim em um mesmo co-set todos os elementos são distintos.

Suponhamos agora que dois elementos iguais se encontram em diferentes co-sets, isto é,

$$g_i h_j = g_k h_\ell \tag{2.9c}$$

de modo que i > k.

Multiplicando (2.9c) à direita por h_j^{-1} obtemos

$$g_i = g_k h_\ell h_j^{-1} \tag{2.9d}$$

Como $h_\ell h_j^{-1}$ pertence ao sub-grupo pelos axiomas G.1 e G.4, concluimos que g_i pertence ao co-set cujo líder é g_k. Mas isto é uma contradição às regras de construção do arranjo. Assim, dois elementos iguais não podem pertencer a co-sets diferentes. Portanto, cada elemento do grupo aparece em um e somente um co-set.

Q.E.D.

DEFINIÇÃO 2.8 - O número de elementos em um grupo é denominado de ORDEM DO GRUPO. O número de co-sets de um grupo G com relação a um sub-grupo H de G é denominado de ÍNDICE DE G SOBRE H.

Da definição 2.8 e pelas regras de construção do arranjo da Fig. 2.1 obtemos a seguinte relação:

(ordem de H)(índice de G sobre H) = (ordem de G) (2.10)

DEFINIÇÃO 2.9 - Um sub-grupo H de um grupo G é denominado de sub-grupo normal se para cada elemento h pertencente a H e cada elemento g pertencente a G, $g^{-1}.h.\ g$ pertence a H.

Em geral, co-sets à esquerda são diferentes de co-sets à direita. Entretanto, cada co-set à esquerda de um grupo normal é também um co-set à direita e vice-versa. Em um grupo Abeliano , cada co-set à esquerda é, trivialmente, um co-set à direita e também cada sub-grupo é trivialmente um sub-grupo normal.

DEFINIÇÃO 2.10 - Se um sub-grupo H de um grupo G é normal, é possível definir uma operação entre co-sets para formar um novo grupo cujos elementos são os co-sets. Esse grupo assim formado é denominado de GRUPO FATOR e representado por G/H. O co-set que contém g é representado por {g}.

DEFINIÇÃO 2.11 - Sejam os co-sets $\{g_1\}$ e $\{g_2\}$. A multiplicação de co-sets é definida como:

$$\{g_1\}.\{g_2\} = \{g_1.g_2\} \qquad (2.11)$$

É importante observar que essa definição só é válida se, independentemente dos elementos escolhidos para representar os co-sets fatores, o co-set produto é o mesmo. Mostremos que isso é verdade.

Suponha $g_1^{-1}g_1' = h_1$ e $g_2^{-1}g_2' = h_2$. Então, desde que o sub-grupo seja normal, pela definição 2.9, $g_2'^{-1}.h_1.g_2'$ é um elemento de H, digamos h_3. Assim,

$$(g_1g_2)^{-1}g_1'g_2' = g_2^{-1}(g_1^{-1}g_1')g_2' = g_2^{-1}h_1g_2' = (g_2^{-1}g_2')(g_2'^{-1}h_1g_2') = h_2h_3 \qquad (2.12)$$

é um elemento pertencente a H pelo axioma G.1. Portanto, pelo teorema 2.2, g_1g_2 e $g_1'g_2'$ estão no mesmo co-set.

Mostremos que G/H é realmente um grupo para a operação multiplicação de co-sets definida pela definição 2.11. A operação está definida para todos os pares de co-sets e assim o axioma G.1 está satisfeito.

A associatividade pode ser checada através de:

$$\{g_1\}(\{g_2\}\{g_3\}) = \{g_1\}\{g_2g_3\} = \{g_1g_2g_3\} = \{g_1g_2\}\{g_3\} =$$

$$= \{g_1\}\{g_2\}\{g_3\} = (\{g_1\}\{g_2\})\{g_3\} \qquad (2.13)$$

O elemento identidade é o próprio sub-grupo H={1}, pois

$$\{1\}\{g\} = \{1.g\} = \{g\} \qquad (2.14a)$$

$$\{g\}\{1\} = \{g.1\} = \{g\} \qquad (2.14b)$$

O co-set inverso a {g} é o co-set que possui g^{-1} como elemento, isto é, $\{g^{-1}\}$. Assim

$$\{g\}\{g^{-1}\} = \{g.g^{-1}\} = \{1\} \qquad (2.15a)$$

$$\{g^{-1}\}\{g\} = \{g.g^{-1}\} = \{1\} \qquad (2.15b)$$

e o axioma G.4 está satisfeito. Como todos os axiomas para grupo estão satisfeitos o grupo fator definido em 2.10 é realmente um grupo sobre a multiplicação de co-sets definida em 2.11.

Pode-se mostrar que se o grupo original G é Abeliano, o grupo fator também o é.

2.5 - ESPAÇOS VETORIAIS E ÁLGEBRAS LINEARES

DEFINIÇÃO 2.12 - Um conjunto V de elementos é chamado de ESPAÇO VETORIAL sobre um campo F, se os seguintes axiomas são satisfeitos.

AXIOMA V.1 - O conjunto V é um grupo Abeliano sobre a adição.

AXIOMA V.2 - Para qualquer elemento $\underset{\sim}{v}$ pertencente a V, denominado vetor, e para qualquer elemento c pertencente a F, denominado escalar, o produto $c.\underset{\sim}{v}$ é também um vetor pertencente a V.

AXIOMA V.3 - Se $\underset{\sim}{u}$ e $\underset{\sim}{v}$ são vetores pertencentes a V e c é um escalar,

$$c(\underset{\sim}{u} + \underset{\sim}{v}) = c.\underset{\sim}{u} + c.\underset{\sim}{v} \qquad (2.16)$$

AXIOMA V.4 - Se $\underset{\sim}{v}$ é um vetor e c e d são escalares,

$$(c + d)\underset{\sim}{v} = c\underset{\sim}{v} + d\underset{\sim}{v} \qquad (2.17)$$

AXIOMA V.5 - Se $\underset{\sim}{v}$ é um vetor e c e d são escalares,

$$(cd)\underset{\sim}{v} = c(d\underset{\sim}{v}) \qquad (2.18a)$$

$$1\underset{\sim}{v} = \underset{\sim}{v} \qquad (2.18b)$$

DEFINIÇÃO 2.13 - Um conjunto A é denominado de uma ÁLGEBRA LINEAR associativa sobre um campo F, se os seguintes axiomas são satisfeitos:

AXIOMA A.1 - O conjunto A é um espaço vetorial sobre F.

AXIOMA A.2 - Para quaisquer dois elementos $\underset{\sim}{u}$ e $\underset{\sim}{v}$ pertencentes a A, existe um produto $\underset{\sim}{u}.\underset{\sim}{v}$ que também pertence a A.

AXIOMA A.3 - Para quaisquer três elementos $\underset{\sim}{u}$, $\underset{\sim}{v}$ e $\underset{\sim}{w}$ pertencentes a A,

$$\underset{\sim}{u}(\underset{\sim}{v}.\underset{\sim}{w}) = (\underset{\sim}{u}.\underset{\sim}{v})\underset{\sim}{w} \qquad (2.19)$$

AXIOMA A.4 - Se c e d são escalares pertencentes a F e $\underset{\sim}{u}$, $\underset{\sim}{v}$ e $\underset{\sim}{w}$ pertencem a A, então:

$$\underset{\sim}{u}(c\underset{\sim}{v} + d\underset{\sim}{w}) = c\underset{\sim}{u}\underset{\sim}{v} + d\underset{\sim}{u}\underset{\sim}{w} \qquad (2.20a)$$

$$(c\underset{\sim}{v} + d\underset{\sim}{w})\underset{\sim}{u} = c\underset{\sim}{v}\underset{\sim}{u} + d\underset{\sim}{w}\underset{\sim}{u} \qquad (2.20b)$$

DEFINIÇÃO 2.14 - Uma n-upla sobre um campo F é um conjunto ordenado de elementos e representada por $(a_1, a_2, \ldots, a_n)$, onde cada a_i pertence ao campo F e é denominado de i-ésima componente da n-upla. Adição de n-uplas é definida por:

$$(a_1,a_2,\ldots,a_n) + (b_1,b_2,\ldots,b_n) = (a_1+b_1,a_2+b_2,\ldots,a_n+b_n) \qquad (2.21a)$$

e multiplicação de uma n-upla por um elemento do campo é definida por:

$$c(a_1, a_2, \ldots, a_n) = (ca_1, ca_2, \ldots, ca_n) \qquad (2.21b)$$

Com essas duas operações definidas por (2.21a) e (2.21b), pode-se mostrar facilmente que o conjunto de todas as n-uplas sobre um campo F, forma um espaço vetorial. É esse espaço vetorial que desempenha um papel fundamental na teoria da codificação.

Multiplicação de n-uplas pode ser definida como:

$$(a_1,a_2,\ldots,a_n)(b_1,b_2,\ldots,b_n) = (a_1b_1,a_2b_2,\ldots,a_nb_n) \qquad (2.22)$$

Com essa última definição, o conjunto de todas as n-uplas forma uma álgebra linear.

DEFINIÇÃO 2.15 - Um sub-conjunto de um espaço vetorial é denominado de SUB-ESPAÇO, se ele satisfaz aos axiomas para espaço vetorial.

Para verificar se um conjunto forma um sub-espaço vetorial é suficiente verificar o fechamento com relação à adição e à multiplicação por escalares.

DEFINIÇÃO 2.16 - Sejam $\underset{\sim}{v}_1, \underset{\sim}{v}_2, \ldots, \underset{\sim}{v}_k$; k vetores pertencentes a um espaço vetorial V. Se um vetor $\underset{\sim}{u}$ pertencente a V é dado por:

$$\underset{\sim}{u} = a_i\underset{\sim}{v}_1 + a_2\underset{\sim}{v}_2 + \ldots + a_k\underset{\sim}{v}_k \qquad (2.23)$$

onde os coeficientes a_i são escalares, dizemos que $\underset{\sim}{u}$ é uma COMBINAÇÃO LINEAR de $\underset{\sim}{v}_1, \underset{\sim}{v}_2, \ldots, \underset{\sim}{v}_k$.

TEOREMA 2.4 - Um conjunto formado por todas as combinações lineares de $\underset{\sim}{v}_1, \underset{\sim}{v}_2, \ldots, \underset{\sim}{v}_k$, pertencentes a um espaço vetorial V, é um sub-espaço de V.

DEFINIÇÃO 2.17 - Um conjunto de vetores $\underset{\sim}{v}_1, \underset{\sim}{v}_2, \ldots, \underset{\sim}{v}_k$ são ditos LINEARMENTE DEPENDENTES, se existem escalares $c_1, c_2, \ldots, c_k$, não todos nulos, tal que:

$$c_1\underset{\sim}{v}_1 + c_2\underset{\sim}{v}_2 + \ldots + c_k\underset{\sim}{v}_k = \underset{\sim}{0} \qquad (2.24)$$

em caso contrário, eles são ditos LINEARMENTE INDEPENDENTES.

DEFINIÇÃO 2.18 - Um conjunto de vetores gera um espaço vetorial, se cada vetor pertencente ao espaço vetorial puder ser representa

do como uma combinação linear dos vetores do conjunto.

TEOREMA 2.5 - Se um conjunto de k vetores $\underset{\sim}{v}_1, \underset{\sim}{v}_2, \ldots, \underset{\sim}{v}_K$ gera um espaço vetorial que contém um conjunto de m vetores, $\underset{\sim}{v}_1, \underset{\sim}{v}_2, \ldots, \underset{\sim}{v}_m$ linearmente independentes, então $k \geqslant m$.

TEOREMA 2.6 - Se dois conjuntos de vetores linearmente independentes geram o mesmo espaço vetorial, então o número de vetores em cada conjunto é o mesmo.

DEFINIÇÃO 2.19 - Se um espaço vetorial é gerado por um conjunto de k vetores linearmente independentes, diz-se que o espaço tem dimensão k e que o conjunto de k vetores é uma base para o espaço.

TEOREMA 2.7 - Se V é um espaço vetorial de dimensão k , qualquer conjunto de k vetores linearmente independentes que pertença a V é uma base para V.

TEOREMA 2.8 - Se um espaço vetorial V_1 é sub-conjunto de um espaço vetorial V_2 e eles têm a mesma dimensão, então V_1 e V_2 são iguais.

DEFINIÇÃO 2.20 - Duas n-uplas $\underset{\sim}{v}_1$ e $\underset{\sim}{v}_2$ são ditas ortogonais quando o produto escalar $\underset{\sim}{v}_1 . \underset{\sim}{v}_2 = 0$.

TEOREMA 2.9 - O conjunto de todas as n-uplas ortogonais a um sub-espaço V_1 de n-uplas, forma um sub-espaço V_2 de n-uplas.

DEFINIÇÃO 2.21 - O espaço vetorial V_2, formado por todas as n-uplas ortogonais a um espaço V_1, é denominado de ESPAÇO NULO de V_1.

TEOREMA 2.10 - Se um vetor $\underset{\sim}{v}$ é ortogonal a cada vetor de um conjunto de vetores que gera o espaço V_1, então $\underset{\sim}{v}$ pertence ao espaço nulo de V_1.

TEOREMA 2.11 - Se a dimensão de um sub-espaço de n-uplas é k, então a dimensão do seu espaço nulo é n - k.

TEOREMA 2.12 - Se V_2 é um sub-espaço de n-uplas e V_1 é o espaço nulo de V_2, então V_2 é o espaço nulo de V_1.

2.6 - MATRIZES

DEFINIÇÃO 2.22 - Uma matriz m x n é um arranjo retangular de m linhas e n colunas onde os elementos das linhas ou colunas pertencem a um anel ou a um campo.

Na teoria da codificação, apenas as matrizes cujos elementos pertencem a um campo são de interesse. As m-linhas de uma matriz podem ser imaginadas como m n-uplas ou vetores e da mesma forma, as n colunas podem ser imaginadas como n m-uplas.

DEFINIÇÃO 2.23 - O conjunto de todas as combinações lineares das linhas de uma matriz M é denominado O ESPAÇO DAS LINHAS DE $\overline{M}$. Do mesmo modo, o conjunto de todas as combinações lineares das colunas de M é denominado O ESPAÇO DAS COLUNAS DE $\overline{M}$. A dimensão do espaço das linhas é denominado de RANK DAS LINHAS e a dimensão do espaço das colunas é denominado de RANK DAS COLUNAS.

DEFINIÇÃO 2.24 - Há um conjunto de OPERAÇÕES ELEMENTARES definidas para as linhas de uma matriz que são:

1) Permutação de duas linhas quaisquer.
2) Multiplicação de uma linha por um elemento do campo desde que seja diferente de zero.
3) A adição da múltipla de qualquer linha, obtida através de 2) a uma outra linha.

A inversa de cada operação elementar de linhas é também uma operação elementar de linhas.

TEOREMA 2.13 - Se uma matriz é obtida a partir de outra através de oeprações elementares de linhas, ambas matrizes têm o mesmo espaço de linhas.

DEFINIÇÃO 2.25 - Uma matriz $\overline{M}'$ é dita na forma canônica de Echelon, quando é obtida a partir de uma matriz $\overline{M}$, por meio de operações elementares de linhas, e satisfaz às seguintes condições:

1) O primeiro elemento diferente de zero de uma linha não nula é 1 e é denominado de LÍDER DA LINHA.

2) Cada coluna que contenha o elemento obtido em (1), possui todos os outros elementos nulos.

3) O líder de cada linha está à direita do líder da linha imediatamente anterior.

PROPRIEDADES DE UMA MATRIZ NA FORMA DE ECHELON:

1) As linhas não nulas são linearmente independentes.

2) Para cada espaço de linhas existe uma única matriz na forma canônica de Echelon.

DEFINIÇÃO 2.26 - Se todas as linhas de uma matriz n x n são linearmente independentes, então a matriz é dita NÃO SINGULAR.

Se uma matriz não singular é colocada na forma canônica de Echelon, cada linha obrigatoriamente contém um 1. Mas isso só pode ocorrer se os 1's estiverem na diagonal principal e essa matriz é chamada de matriz identidade.

2.7 - IDEAIS

DEFINIÇÃO 2.27 - Um ideal I é um sub-conjunto de elementos pertencentes a um anel R com as seguintes propriedades:

1) I é um sub-grupo do grupo aditivo de R.

2) Para qualquer elemento a pertencente a I e qualquer r pertencente a R, ar e ra também pertencem a I.

Como um ideal é um sub-grupo, co-sets podem ser formados como descrito na seção 2.4. Entretanto, no caso dos ideais eles recebem o nome de CLASSES DE RESÍDUOS. O arranjo descrito na seção 2.4 pode ser construído da mesma forma lembrando apenas que agora a operação do grupo é a adição. Como a operação do grupo é comutativa, então o ideal é um sub-grupo normal (ver comentários logo após Definição 2.9, seção 2.4). Assim, a adição de classes de resíduo pode ser definida como:

$$\{r\} + \{s\} = \{r+s\} \qquad (2.25)$$

onde {r} denota a classe de resíduo que contém r. Com esta definição, o conjunto das classes de resíduo forma um grupo, o grupo fator dado pela definição 2.10.

É possível também a multiplicação de classe de resíduo através da equação

$$\{r\}\{s\} = \{r.s\} \tag{2.26}$$

A multiplicação de classes de resíduo definida pela igualdade (2.26) só é válida se, independentemente do elemento escolhido para representar as classes de resíduo, o produto desses elementos pertencem à mesma classe de resíduo. Mostremos que esse fato é verdadeiro.

Suponhamos que r e r' pertencem à mesma classe de resíduo e que s e s' pertencem a uma mesma classe de resíduo diferente da classe de resíduo a que pertencem r e r'. Devemos mostrar que r.s e r'.s' pertencem a uma mesma classe de resíduo. Pelo teorema 2.2, isso só é verdade se r's'-rs pertencer ao ideal. Mas

$$r's'-rs = r's'-r's + r's-rs = r'(s'-s) + (r'- r)s \tag{2.27}$$

Pelo teorema 2.2, s'-s e r'-r pertencem ao ideal: assim as parcelas r'(s'-s) e (r'-r)s pertencem ao ideal, por definição. Então, pelo axioma G.1, r's'-rs pertence ao ideal e portanto a multiplicação de classes de resíduo definida por 2.26 é válida.

É fácil de verificar que a associatividade e distributividade são válidas:

$$\{a\}(\{b\}\{c\}) = \{a\}\{bc\} = \{abc\} = \{ab\}\{c\} = (\{a\}\{b\})\{c\} \tag{2.28a}$$

$$\{a\}(\{b\}+\{c\}) = \{a\}\{b+c\} = \{a(b+c)\} = \{ab+ac\} = \{ab\} + \{ac\} =$$

$$= \{a\}\{b\} + \{a\}\{c\} \tag{2.28b}$$

$$(\{b\} + \{c\})\{a\} = \{b+c\}\{a\} = \{(b+c)a\} = \{ba+ca\} =$$

$$= \{ba\} + \{ca\}\{b\}\{a\} + \{c\}\{a\} \tag{2.28c}$$

Portanto, os axiomas para anel estão satisfeitos e está provado o seguinte teorema:

TEOREMA 2.14 - O conjunto das classes de resíduo de um anel com relação a um ideal, forma um anel denominado de ANEL DAS CLASSES DE RESÍDUOS.

2.8 - IDEAIS E CLASSES DE RESÍDUO DE INTEIROS

DEFINIÇÃO 2.28 - Se r, s e t são inteiros e rs=t, então dizemos que t é divisível por r ou que r divide t.

DEFINIÇÃO 2.29 - Um inteiro $p \geqslant 1$ que é divisível apenas por $\pm p$ ou ± 1 é denominado de PRIMO.

DEFINIÇÃO 2.30 - O máximo divisor comum entre dois inteiros (MDC) é o maior inteiro positivo que divide a ambos.

DEFINIÇÃO 2.31 - Dois inteiros são ditos RELATIVAMENTE PRIMOS quando seu MDC é 1.

DEFINIÇÃO 2.32 - Para cada par de inteiros s e d existe um único par de inteiros q, o quociente, e r, o resto tal que

$$s = dq + r \quad , \quad 0 \leqslant r \leqslant |d| \tag{2.29}$$

Esse resultado é conhecido como o ALGORÍTMO DA DIVISÃO DE EUCLIDES.

A partir do algorítmo da divisão de Euclides pode-se mostrar que o MDC d entre dois inteiros r e s sempre pode ser expresso por:

$$d = ar + bs \tag{2.30}$$

onde a e b são inteiros.

TEOREMA 2.15 - Um conjunto de inteiros é um ideal se, e somente se, todos os seus elementos são múltiplos de algum inteiro.

PROVA DA PROPOSIÇÃO DIRETA

"Se um conjunto de inteiros é um ideal, então cada elemento do conjunto é múltiplo de algum inteiro."

Seja r o menor inteiro positivo pertencente ao ideal e seja s um outro inteiro pertencente ao ideal. Então pela equação (2.30) e pela definição de ideal (Definição 2.27), o MDC entre r e s pertence ao ideal. Como por hipótese r é o menor inteiro positivo pertencente ao ideal, então $r \leqslant d$. Como d divide r , então $d < r$. Então temos a desigualdade:

$$d \leq r \leq d \qquad (2.31a)$$

Portanto, r = d e assim r divide s. Como s é qualquer, r divide qualquer inteiro pertencente ao ideal e portanto, qualquer inteiro pertencente ao ideal é multiplo de r.

PROVA DA PROPOSIÇÃO INVERSA

"Se um conjunto é constituído de todos os múltiplos de algum inteiro, então ele forma um ideal."

Seja o conjunto de todos os múltiplos de r. Assim,cada elemento do conjunto pode ser escrito como

$$s = ar \qquad (2.31b)$$

Se r é um elemento de um ideal então pela definição de ideal todos os múltiplos de r também pertencem ao ideal.

Q.E.D.

DEFINIÇÃO 2.33 - Um ideal que consiste de todos os múltiplos de um elemento de um anel é denominado de IDEAL PRINCIPAL.

DEFINIÇÃO 2.34 - Um anel no qual cada ideal é um ideal principal é denominado de ANEL IDEAL PRINCIPAL.

DEFINIÇÃO 2.35 - O ideal que consiste de todos os múltiplos de um inteiro positivo m é representado por (m). O anel das classes de resíduo de (m) é denominado de ANEL DE INTEIROS MÓDULO m.

TEOREMA 2.16 - Cada classe de resíduo módulo m contém ou 0 ou um inteiro positivo menor que m. 0 é um elemento do ideal e cada inteiro positivo menor que m pertence a uma classe de resíduo distinta.

PROVA:

Seja s um elemento de uma classe de resíduo.Então s pode ser escrito como:

$$s = mq + r \qquad (2.32)$$

onde mq é um elemento pertencente a (m) e r é o líder da classe de resíduo que contém s. Pelo algorítmo da divisão de Euclides , $0 \leq r < m$. Como r e s estão na mesma classe de resíduo, pelo teo

rema 2.2 r-s pertence ao ideal (m) e portanto é um múltiplo de m. Pelo teorema 2.3 r≠s, então r e s não podem ambos ser menores que m, e não negativos, pois supondo r > s e ambos menores que m e positivos, implica em r - s < 0, o que é uma contradição com a hipótese r > s. Do mesmo modo, supondo r < s ambos menores que m e positivos, implica em r - s > 0, o que contradiz a hipótese r < s. Portanto, apenas um elemento menor que m pertence a uma classe de resíduo que não seja o ideal. Se r = s, r-s = 0 que pertence à classe de resíduo constituída pelo ideal.

Como consequência do teorema 2.16, se representarmos cada classe de resíduo pelo inteiro menor que m a que lhe pertence, o anel das classes de resíduo módulo m é constituído por:

$$\{\{0\}, \{1\}, \ldots, \{m-1\}\} \tag{2.33}$$

TEOREMA 2.17 - O anel das classes de resíduo módulo m é um campo se, e somente se, m é primo.

PROVA:

Suponhamos que m não é primo. Então m = rs para alguns inteiros r e s que não sejam múltiplos de m. Assim,

$$\{m\} = \{rs\} = \{r\}\{s\} = \{0\} = 0 \tag{2.34a}$$

Se $\{r\}$ tem um inverso, digamos $\{r\}^{-1}$, então

$$\{r\}^{-1}.\{m\} = \{r\}^{-1}.\{r\}.\{s\} = \{s\} = \{r\}^{-1}\ \{0\} = 0 \tag{2.34b}$$

o que implica em s ser múltiplo de m, o que é uma contradição.

Supondo agora que m é primo, devemos mostrar que cada classe de resíduo, exceto o ideal, possui uma inversa. Pelo teorema 2.16, cada classe de resíduo diferente do ideal possui um inteiro s < m. Como 1 é seu próprio inverso, podemos supor s > 1. Como m é, por hipótese, primo, o MDC entre s e m é 1 ou m. Mas s<m e assim m não divide s. Portanto, o MDC entre s e m é 1. Utilizando a equação 2.30 temos:

$$1 = am + bs \tag{2.34c}$$

para a e b inteiros. Assim,

$$\{1\} = \{a\}\{m\} + \{b\}\{s\} \tag{2.34d}$$

Como $\{m\}=0$ temos,

$$\{1\} = \{b\}\{s\} \tag{2.34e}$$

Portanto {b} é o inverso de {s}.

Q.E.D.

DEFINIÇÃO 2.36 - Os campos obtidos de acordo com o teorema 2.17 são chamados de CAMPOS PRIMOS ou CAMPOS DE GALOIS de p elementos e representados por GF(p).

2.9 - IDEAIS E CLASSES DE RESÍDUO DE POLINÔMIOS

Consideremos polinômios f(X) a uma variável com coeficientes pertencentes a um campo F.

$$f(X) = f_0 + f_1X + f_2X^2 + \ldots + f_nX^n \qquad (2.35)$$

DEFINIÇÃO 2.37 - O GRAU de um polinômio é dado pelo expoente da maior potência de X cujo coeficiente seja diferente de zero.

DEFINIÇÃO 2.38 - Um polinômio é chamado de MÔNICO se o coeficiente da maior potência de X é 1.

O conjunto de todos os polinômios forma um anel, já que eles podem ser somados e multiplicados no sentido usual de adição e multiplicação e pode-se mostrar facilmente que todos os axiomas para anéis são satisfeitos.

DEFINIÇÃO 2.39 - Se r(X), s(X) e t(X) são polinômios e r(X)s(X)= =t(X), dizemos que t(X) é divisível por r(X) ou que r(X) divide t(X).

DEFINIÇÃO 2.40 - Um polinômio p(X) de grau n maior que zero é dito irredutível se não é divisível por qualquer polinômio de grau menor que n.

DEFINIÇÃO 2.41 - O MDC de dois polimônios é o polinômio mônico de maior grau que divide a ambos.

DEFINIÇÃO 2.42 - Dois polinômios são ditos RELATIVAMENTE PRIMOS se seu MDC é 1.

DEFINIÇÃO 2.43 - Para cada par de polinômios s(X) e d(X), existe um único par de polinômios q(X), o quociente, e r(X), o resto,

tal que:

$$s(X) = d(X).q(X) + r(X) \qquad (2.36)$$

onde o grau de r(X) é menor que o grau de d(X). Este é o algorítmo da divisão de Euclides para polinômios.

Utilizando o algorítmo da divisão de Euclides é sempre possível colocar o MDC de dois polinômios r(X) e s(X) na forma:

$$d(X) = a(X)r(X) + b(X)s(X) \qquad (2.37)$$

onde a(X) e b(X) são polinômios e d(X) o MDC entre r(X) e s(X).

TEOREMA 2.18 - Um conjunto de polinômios é um ideal se,e somente se, todos os polinômios são múltiplos de algum polinômio.

PROVA DA PROPOSIÇÃO DIRETA

"Se um conjunto de polinômios é um ideal, então cada polinômio é múltiplo de algum polinômio".

Seja r(X) o polinômio mônico de menor grau pertencente ao ideal e seja s(X) um outro polinômio pertencente ao ideal.

Então, pela equação (2.37) e pela definição de ideal, o MDC d(X), entre r(X) e s(X) pertence ao ideal. Como por hipótese r(X) é o polinômio mônico de menor grau pertencente ao ideal , o grau de r(X), $g_r[r(X)] \leqslant g_r[d(X)]$. Como d(X) divide r(X) , então $g_r[d(X)] \leqslant g_r[r(X)]$. Assim, temos a seguinte desigualdade:

$$g_r[d(X)] \leqslant g_r[r(X)] \leqslant g_r[d(X)] \qquad (2.30a)$$

Portanto, $g_r[r(X)] = g_r[d(X)]$ e assim r(X) = d(X). Consequentemente, r(X) divide s(X) e portanto todos os elementos do ideal são múltiplos de r(X).

PROVA DA PROPOSIÇÃO INVERSA

"Se um conjunto de polinômios é constituido de todos os

polinômios múltiplos de algum polinômio, então o conjunto forma um ideal."

Seja o conjunto de todos os múltiplos de r(X). Assim , cada elemento s(X) do conjunto pode ser escrito como:

$$s(X) = r(X)\ a(X) \qquad (2.38b)$$

Se r(X) é um elemento de um ideal, então pela definição de ideal todos os múltiplos de r(X) pertencem ao ideal.

Q.E.D.

Pelo teorema 2.18 concluimos que o anel de polinômios é um anel ideal principal. O ideal que consiste de todos os múltiplos de f(X) é denotado por (f(X)). O anel das classes de resíduo formado a partir desse ideal é denominado de ANEL DE POLINÔMIOS MÓDULO f(X).

TEOREMA 2.19 - Cada classe de resíduo módulo um polinômio f(X) de grau n contém 0 ou um polinômio de grau menor que n. Zero é um elemento do ideal e cada polinômio de grau menor que n está em uma classe de resíduo distinta.

PROVA

Seja s(X) um elemento pertencente a uma classe de resíduo. Então s(X) pode ser escrito como:

$$s(X) = a(X)\ f(X) + r(X) \qquad (2.39)$$

onde a(X) f(X) é um elemento do ideal (f(X)) e r(X) é o líder da classe de resíduo que contém s(X). Pelo algorítmo da divisão de Euclides para polinômios, $0 \leqslant g_r[r(X)] < g_r[f(X)]$. Como r(X) e s(X) pertencem à mesma classe de resíduo, pelo teorema 2.2, r(X) -s(X) pertence ao ideal (d(X)) e portanto é um múltiplo de f(X) o que implica no $g_r[r(X) - s(X)] \geqslant g_r[f(X)]$. Mas $g_r[r(X)-s(X)] \geqslant g_r[f(X)]$ implica em $g_r[s(X)] > g_r[r(X)]$ para $s(X) \neq r(X)$, já que $g_r[r(X)] < g_r[f(X)]$. Portanto, na classe de resíduo que contém r(X) e s(X) apenas r(X) tem grau menor que n. Se r(X) = s(X) então r(X) - s(X) = 0 que é um elemento do ideal.

Q.E.D.

2.10 - A ÁLGEBRA DAS CLASSES DE RESÍDUO DE POLINÔMIOS

TEOREMA 2.20 - As classes de resíduo de polinômios módulo um po

linômio f(X) de grau n, formam uma álgebra linear comutativa de dimensão n sobre o campo dos coeficientes.

PROVA:

A multiplicação por escalares é definida pela equação

$$a\{r(X)\} = \{ar(X)\} \tag{2.39a}$$

onde a pertence ao campo dos coeficientes. É facilmente verificado que os axiomas para espaço vetorial e álgebra linear são satisfeitos.

Para mostrarmos que a dimensão é n, usamos o fato de que qualquer classe de resíduo pode ser escrita como uma combinação das classes de resíduo

$$\{1\}, \{X\}, \{X^2\}, \ldots, \{X^{n-1}\} \tag{2.39b}$$

isto é, o conjunto das classes de resíduo (2.30b) geram o espaço. A existência dessas n classes de resíduo é garantida pelo teorema 2.19. Mostremos que esse fato é verdadeiro.

Pelo teorema 2.19 cada classe de resíduo tem um polinômio de grau menor que n. Então uma classe de resíduo genérica é da forma

$$\{a_0 + a_1X + \ldots + a_{n-1}^{n-1}\} \tag{2.39c}$$

que pode ser escrita como

$$\{a_0\} + \{a_1X\} + \ldots + \{a_{n-1}X^{n-1}\} =$$

$$= a_0\{1\} + a_1\{X\} + \ldots + a_{n-1}\{X^{n-1}\} \tag{2.39d}$$

Assim, qualquer classe de resíduo pode ser expressa como uma combinação linear do conjunto das classes de resíduo (2.39b). A combinação linear (2.39d) só se anula se

$$a_0 + a_1X + \ldots + a_{n-1}X^{n-1} \tag{2.39e}$$

for múltiplo de f(X), o que é impossível pois o grau de f(X) é n, ou então se os coeficientes a_i forem todos nulos. Portanto, o conjunto (2.39b) forma uma base para o espaço vetorial.

Q.E.D.

TEOREMA 2.21 - Seja J um ideal na álgebra de polinômios módulo f(X) e seja g(X) um polinômio de menor grau, diferente de zero, tal que {g(X)} pertence a J. Então {s(X)} está em J se, e somente se, s(X) é divisível por g(X). E mais ainda g(X) divide f(X).

PROVA:

Pelo algorítmo da divisão de Euclides,

$$s(X) = g(X) \cdot q(X) + r(X) \qquad (2.40a)$$

onde r(X) tem grau menor que o grau de g(X). Assim,

$$\{s(X)\} = \{g(X)\}\{q(X)\} + \{r(X)\} \qquad (2.40b)$$

Se s(X) e g(X) estão em J, então

$$\{s(X)\} - \{g(X)\}\{q(X)\} = \{r(X)\} \qquad (2.40c)$$

também está, pelo teorema 2.2. Como r(X) tem grau menor que o de g(X) que foi suposto o polinômio não nulo de menor grau em J, então obrigatoriamente r(X)=0 e assim s(X) é múltiplo de g(X)e portanto divisível por g(X).

Inversamente, se s(X) é múltiplo de g(X), então

$$s(X) = g(X)\, q(X) \qquad (2.40d)$$

e portanto

$$\{s(X)\} = \{g(X)\}\{q(X)\} \qquad (2.40e)$$

Como por hipótese g(X) pertence a J, pela definição de ideal s(X) também pertence.

Pelo algorítmo da divisão de Euclides,

$$f(X) = g(X)\, q(X) + r(X) \qquad (2.40f)$$

onde r(X) tem grau menor que o grau de g(X). Assim,

$$\{f(X)\} = \{g(X)\}\{q(X)\} + \{r(X)\} = \{0\} \qquad (2.40g)$$

Portanto,

$$\{r(X)\} = \{g(X)\}.\{h(X)\} \qquad (2.40h)$$

onde h(X) = -g(X). Então {r(X)}está em J já que r(X) é um múltiplo de g(X). Mas o grau de r(X) é menor que o grau de g(X), então r(X)=0 e assim f(X) é múltiplo de g(X) e portanto g(X) divide f(X).

Q.E.D.

TEOREMA 2.22 - Para cada ideal J na álgebra de polinômios módulo f(X) há um único polinômio mônico g(X) de menor grau tal que {g(X)} pertence a J. Inversamente, cada polinômio mônico g(X)que divide f(X) gera um ideal J no qual g(X) é o polinômio mônico de mínimo grau em J.

PROVA:

Existe um polinômio

$$h(X) = h_0 + h_1X + \ldots + h_kX^k \qquad (2.41a)$$

de mínimo grau tal que $\{h(X)\}$ pertence a J. Então $h_k^{-1}h(X)$ é um polinômio mônico de mínimo grau cuja classe de resíduo está também em J e portanto existe pelo menos um polinômio mônico de mínimo grau que pertence a J. Se existem dois desses polinômios, digamos g(X) e g'(X), então pelo teorema 2.21 g(X) divide g'(X) e g'(X) divide g(X) e assim eles diferem no máximo por um fator que obrigatoriamente é um elemento do campo dos coeficientes . Como ambos foram supostos mônicos esse elemento do campo é 1 e assim g(X) = g'(X). Portanto, existe um único polinômio mônico de grau mínimo, g(X), tal que $\{g(X)\}$ pertence a J.

Suponha agora que g(X) é um polinômio mônico que divide f(X) e considere o ideal J gerado por $\{g(X)\}$, isto é, o ideal cujos elementos são todos os múltiplos de $\{g(X)\}$. Suponha também que $\{r(X)\}$ pertence a J. Então

$$\{r(X)\} = \{g(X)\}\{a(X)\} = \{g(X)\ a(X)\} \qquad (2.41b)$$

para algum a(X) . Portanto r(X) pode ser escrito como

$$r(X) = g(X)\ a(X) + f(X)\ b(X)\ ; \qquad (2.41c)$$

já que f(X) é múltiplo de g(X), por (2.41b), r(X) também o é. Como f(X) e r(X) ambos são múltiplos de g(X), se r(X) não é zero seu grau é maior ou igual ao grau de g(X). Assim g(X) é o polinômio mônico de menor grau tal que $\{g(X)\}$ está em J.

TEOREMA 2.23 - Seja f(X) = g(X).h(X) onde f(X) tem grau n e h(X) tem grau k. Então o ideal gerado por $\{g(X)\}$ na álgebra de polinômios módulo f(X) tem dimensão k.

PROVA:

No ideal gerado por $\{g(X)\}$, que é um sub-espaço, os vetores

$$\{g(X)\},\ \{Xg(X)\},\ldots,\ \{X^{k-1}g(X)\} \qquad (2.42a)$$

são linearmente independentes porque qualquer combinação linear deles é da forma

$$\{[a_0 + a_1X + a_2X^2 + \ldots + a_{k-1}X^{k-1}]\ g(X)\} \qquad (2.42b)$$

e pelo teorema 2.20 não pode ser zero a não ser que a_0, a_1, ..., a_{k-1} sejam zero pois a classe de resíduo (2.42b) contém um polinômio de grau n - 1 < n. Ainda mais, se {s(X)} pertence ao ideal, então s(X) é divisível pro g(X), pelo teorema 2.21, e se s(X) é o polinômio de mínimo grau na classe de resíduo; ele tem grau menor que n. Assim

$$s(X) = g(X)\, q(X) = g(X)\, (q_0 + q_1 X + \ldots + q_{k-1} X^{k-1}) \tag{2.42c}$$

e portanto

$$\{s(X)\} = q_0\, \{g(X)\} + q_1\, \{Xg(X)\} + \ldots + q_{k-1}\, \{X^{k-1} g(X)\} \tag{2.42d}$$

Assim o conjunto de vetores (2.42a) gera o espaço e portanto a dimensão do ideal é k.

Q.E.D.

Com estes resultados da álgebra, estamos aptos a entender o processo de obtenção de códigos de bloco lineares a serem discutidos no Cap. 4 e mais ainda, com o auxílio do Cap.3 ,estaremos capacitados a implementar o codificador que, como veremos ao final dos capítulos 6 e 7, é o ponto de partida para a obtenção do espectro de potência de um código.

CAPÍTULO 3

MÁQUINAS SEQUENCIAIS DE ESTADO FINITO

3.1 - CIRCUITOS SEQUENCIAIS

DEFINIÇÃO 3.1 - Um circuito é dito sequencial quando, em qualquer instante, suas saídas são funções das entradas, como também da informação armazenada até aquele instante, isto é, as saídas são função das entradas atuais e de todo o passado.

DEFINIÇÃO 3.2 - Uma MÁQUINA DE ESTADO FINITO é um modelo abstrato, que descreve um circuito sequencial síncrono também chamado de MÁQUINA SEQUENCIAL SÍNCRONA.

O comportamento de uma máquina de estado finito é descrito como uma sequência de eventos que ocorrem de uma forma discreta no tempo. De acordo com a definição 3.1, é necessário que a máquina síncrona tenha uma capacidade de armazenamento infinita para que seja possível o conhecimento de todo o passado. Entretanto, é impossível em termos práticos se construir máquinas que tenham capacidade de memória infinita. Assim, só existe interesse no estudo de máquinas cujo passado afeta seu comportamento futuro em apenas um número finito de formas.

DEFINIÇÃO 3.3 - Os estados internos de uma máquina são definidos como o número de classes de entradas passadas que uma maquina pode distinguir.

3.2 - REPRESENTAÇÃO DE UMA MÁQUINA SEQUENCIAL SÍNCRONA

O modelo geral de uma máquina sequencial síncrona está representado na Fig. 3.1. O circuito tem um número finito k de entradas. Os sinais que entram no circuito através dessas k entradas constituem o conjunto

$$\{x_1, x_2, \ldots, x_k\} \tag{3.1}$$

denominado de conjunto de variáveis de entrada onde cada x_i é uma variável binária, isto é,

$$x_i \in \{0,1\} \tag{3.2}$$

Uma k-upla ordenada de elementos pertencentes ao conjunto {0,1} é denominada de configuração de entrada ou simplesmente de entrada. O conjunto de todas as k-uplas binárias, em número de $K = 2^k$, é denominado de alfabeto de entrada I ou simplesmente de CONJUNTO DE ENTRADA.

$$I = \{\underline{i}_1, \underline{i}_2, \ldots, \underline{i}_K\} \tag{3.3}$$

onde cada $\underline{i}_k$ corresponde a uma k-upla.

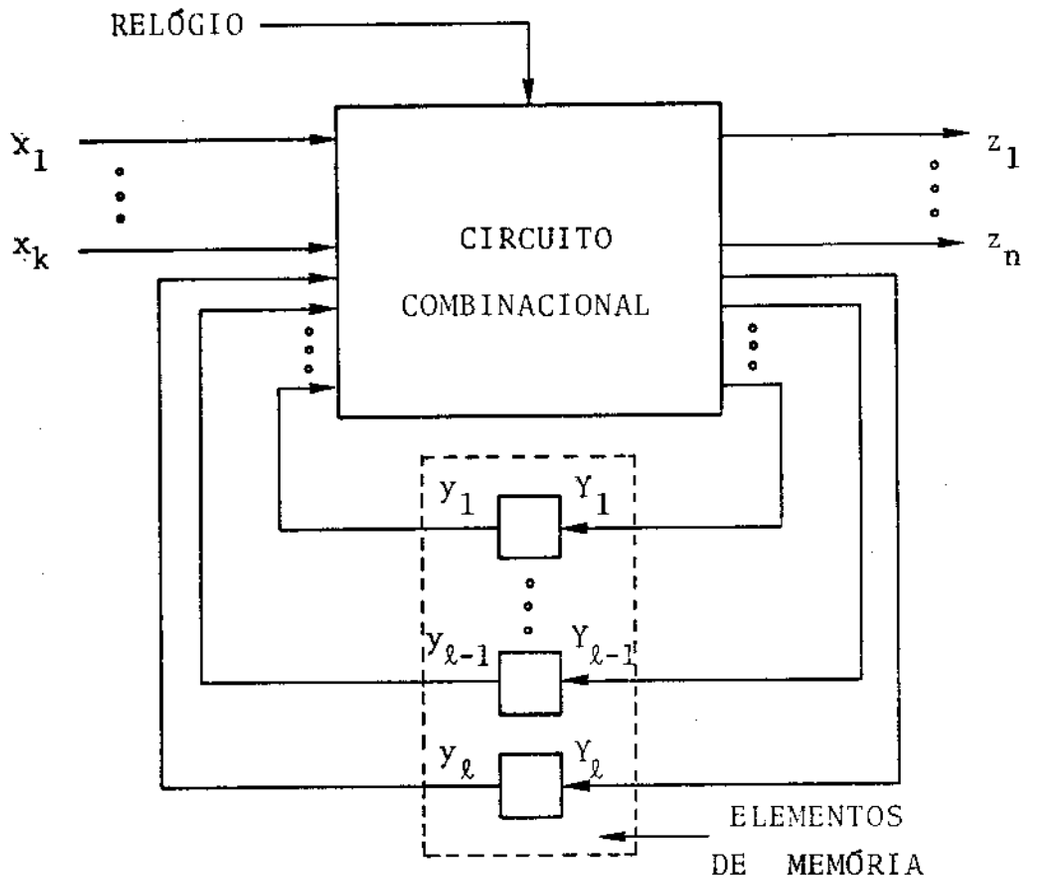


Fig. 3.1 - Modelo geral de uma máquina sequencial síncrona

Do mesmo modo, o circuito possui um número finito n de saídas e um conjunto de variáveis de saída

$$\{z_1, z_2, \ldots, z_n\} \tag{3.4}$$

onde cada z_i pertence ao conjunto $\{0,1\}$. Uma n-upla ordenada de elementos binários é denominada de configuração de saída ou simplesmente saída do circuito. O conjunto de todas as $N = 2^n$ saídas O, é denominado de CONJUNTO DE SAÍDA.

$$O = \{\underline{o}_1, \underline{o}_2, \ldots, \underline{o}_N\} \tag{3.5}$$

O valor do sinal na saída de cada elemento de memória é denominado de VARIÁVEL DE ESTADO e assim o conjunto

$$\{y_1, y_2, \ldots, y_\ell\} \tag{3.6}$$

é denominado de CONJUNTO DAS VARIÁVEIS DE ESTADO. A combinação das ℓ saídas dos elementos de memória define o ESTADO ATUAL DA MÁQUINA e o número $L=2^{\ell}$ de todas as combinações define o CONJUNTO DE ESTADOS DA MÁQUINA

$$S = \{\underset{\sim}{s}_1, \underset{\sim}{s}_2, \ldots, \underset{\sim}{s}_L\} \tag{3.7}$$

onde cada $\underset{\sim}{s}_i$ corresponde a um estado da máquina.

Os valores de $Y_1, Y_2, \ldots, Y_{\ell}$ na saída do circuito combinacional no instante t são iguais aos valores de $y_1, y_2, \ldots, y_{\ell}$ no instante t+1; assim $Y_1, Y_2, \ldots, Y_{\ell}$ definem o PRÓXIMO ESTADO da máquina. O sincronismo do circuito é controlado pelos pulsos do relógio.

3.3 - ESPECIFICAÇÃO DO COMPORTAMENTO DE UMA MÁQUINA

As relações entre entrada, estado atual, saída e próximo estado de uma máquina são especificadas de duas formas : através de uma TABELA DE ESTADO ou através de um DIAGRAMA DE ESTADO. Uma tabela de estado tem k colunas, uma para cada possível entrada, e L linhas, uma para cada estado. O elemento da tabela correspondente ao ℓ-ésimo estado e k-ésima entrada especifica a saída que será gerada e o próximo estado para o qual a máquina irá. Em um diagrama de estado, cada estado corresponde a um vértice no diagrama. A partir de cada vértice saem k ramos, cada ramo correspondendo a uma entrada e cada ramo se dirige para o vértice que corresponde ao estado a ser assumido pela máquina para a correspondente entrada. Cada ramo é identificado pela correspondente entrada e pela saída que é gerada para a transição de estado verificada.

A sucessão de estados através da qual passa uma máquina sequencial e a sequência de saídas que ela produz em resposta a uma sequência de entradas conhecidas, são univocamente determinadas pelo DIAGRAMA DE ESTADO ou TABELA DE ESTADO e pelo ESTADO INICIAL, onde o estado inicial corresponde ao estado em que se encontra a máquina antes da aplicação da sequência de entrada. O estado da máquina após a aplicação da sequência de entrada é denominado de ESTADO FINAL.

EXEMPLO 6 - Consideremos uma máquina em que k=2, n=2 e ℓ=1. Assim, temos para essa máquina:

$$I = \{\underset{\sim}{i}_1, \underset{\sim}{i}_2, \underset{\sim}{i}_3, \underset{\sim}{i}_4\} ,$$

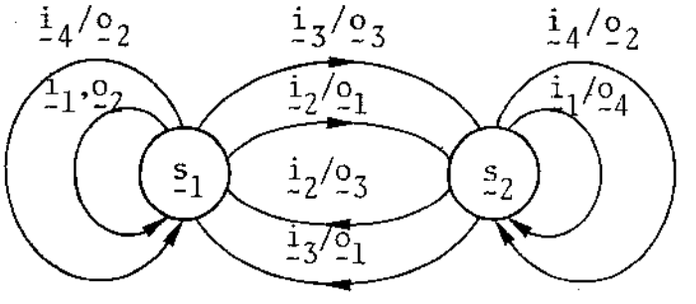
$$O = \{\underset{\sim}{o}_1, \underset{\sim}{o}_2, \underset{\sim}{o}_3, \underset{\sim}{o}_4\} ,$$

$$S = \{\underset{\sim}{s}_1, \underset{\sim}{s}_2\}$$

Especifiquemos um comportamento para a máquina através de uma tabela de estado e de um diagrama de estado mostrados respectivamente nas Figs. 3.2a e 3.2b.

S \ I	$\underset{\sim}{i}_1$	$\underset{\sim}{i}_2$	$\underset{\sim}{i}_3$	$\underset{\sim}{i}_4$
$\underset{\sim}{s}_1$	$\underset{\sim}{o}_2, \underset{\sim}{s}_1$	$\underset{\sim}{o}_1, \underset{\sim}{s}_2$	$\underset{\sim}{o}_3, \underset{\sim}{s}_2$	$\underset{\sim}{o}_2, \underset{\sim}{s}_1$
$\underset{\sim}{s}_2$	$\underset{\sim}{o}_4, \underset{\sim}{s}_2$	$\underset{\sim}{o}_3, \underset{\sim}{s}_1$	$\underset{\sim}{o}_1, \underset{\sim}{s}_1$	$\underset{\sim}{o}_2, \underset{\sim}{s}_2$

a) Tabela de estado



b) Diagrama de estado

Fig. 3.2 - Especificação de um possível comportamento para a máquina do exemplo 6

3.4 - MÁQUINAS DETERMINÍSTICAS

DEFINIÇÃO 3.4 - Uma máquina de estado finito é denominada DETERMINÍSTICA quando possui a propriedade do próximo estado, $\underset{\sim}{s}(t+1)$, ser determinado univocamente pelo estado atual, $\underset{\sim}{s}(t)$, e pela presente entrada $\underset{\sim}{i}(t)$. Isto é:

$$\underset{\sim}{s}(t+1) = g[\underset{\sim}{s}(t), \underset{\sim}{i}(t)] \qquad (3.8a)$$

onde a função g é denominada de FUNÇÃO TRANSIÇÃO DE ESTADO. Também o valor da saída $\underset{\sim}{o}(t)$ é função da entrada atual e do estado atual, isto é,

$$\underset{\sim}{o}(t) = h[\underset{\sim}{s}(t), \underset{\sim}{i}(t)] \qquad (3.8b)$$

onde a função h é denominada de FUNÇÃO DE SAÍDA.

DEFINIÇÃO 3.5 - Uma máquina que satisfaz às equações (3.8) é denominada de MÁQUINA DE MEALY.

Existem máquinas que são modeladas de modo que a função de saída dependa apenas do estado atual da máquina, isto é,

$$\underset{\sim}{o}(t) = h[\underset{\sim}{s}(t)] \qquad (3.9)$$

DEFINIÇÃO 3.6 - Uma máquina que satisfaz às equações (3.8a) e (3.9) é denominada de MÁQUINA DE MOORE.

As definições dadas até aqui podem ser resumidas numa definição mais geral para máquinas sequenciais síncronas.

DEFINIÇÃO 3.7 - Uma máquina sequencial m é uma quíntupla $m=(I,O,S,g,h)$ onde I, O e S são conjuntos finitos não vazios que representam respectivamente entrada, saída e estados da máquina e

$g: I \times S \rightarrow S$ é a função transição de estado.
$h: I \times S \rightarrow O$ é a função de saída para o modelo de Mealy.
$h: S \rightarrow O$ é a função de saída para o modelo de Moore.

I x S representa o produto cartesiano entre os conjuntos I e S cujos elementos são da forma $(\underset{\sim}{i}_k, \underset{\sim}{s}_\ell)$, $k=1,2,\ldots,K$ e $\ell=1,2,\ldots,L$. A função g associa cada par $(\underset{\sim}{i}_k, \underset{\sim}{s}_\ell)$ a um elemento $\underset{\sim}{s}$ pertencente a S, o PRÓXIMO ESTADO da máquina. A função h associa cada par $(\underset{\sim}{i}_k, \underset{\sim}{s}_\ell)$ a um elemento $\underset{\sim}{o}_n$ pertencente a O no modelo de Mealy e no caso do modelo de Moore associa cada elemento $\underset{\sim}{s}$ a um elemento $\underset{\sim}{o}_n$, $n=1,2,\ldots,N$.

3.5 - MÁQUINAS SEQUENCIAIS LINEARES

As máquinas sequenciais lineares formam uma sub-classe dos sistemas lineares nos quais a entrada, saída e transições de estado ocorrem em intervalos discretos, isto é, formam uma sub-classe dos sistemas lineares discretos. Em consequência, toda a teoria dos sistemas lineares discretos pode ser aplicada às máquinas sequenciais lineares.

DEFINIÇÃO 3.8 - Uma máquina sequencial linear é uma máquina de estado finito onde o circuito combinacional do modelo representado na Fig. 3.1 é constituido apenas de elementos lineares. As variáveis de entrada assumem valores pertencentes a GF(p) e as operações realizadas pelos elementos de circuito obedecem às regras das operações definidas em GF(p).

De acordo com a definição 3.8 podemos concluir que o circuito combinacional de uma máquina sequencial linear não pode ter como componentes de circuito portas AND, OR, NAND e NOR pois suas saídas não são combinações lineares das entradas.

Como elementos lineares temos:

1) SOMADORES MÓDULO-p - Um somador possui k entradas e uma saída que é a soma módulo-p das entradas.

2) MULTIPLICADORES POR ESCALAR MÓDULO-p - Um multiplicador c, onde c pertence a GF(p), possui uma entrada e uma saída. Se a entrada é x, a saída é cx módulo-p.

3) ELEMENTOS DE MEMÓRIA - É um elemento de dois terminais, cuja saída y(t) está relacionada com a entrada Y(t)através da relação

$$y(t) = Y(t-1). \qquad (3.10)$$

Na Fig. 3.3 estão representados os elementos de circuitos utilizados na implementação de uma máquina sequencial linear.

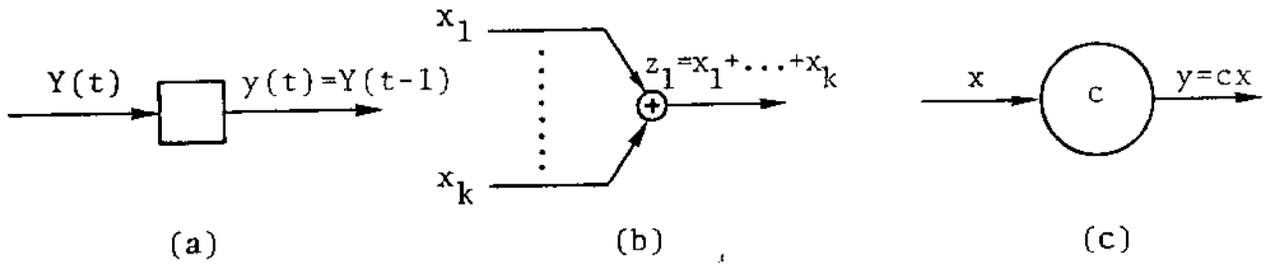


Fig. 3.3 - Componentes constituintes de circuitos sequenciais lineares: a) Elemento de memória
b) Somador módulo-p de k entradas
c) Multiplicador por escalar módulo-p

DEFINIÇÃO 3.9 - O número de elementos de memória em uma máquina linear é denominado de DIMENSÃO DA MÁQUINA e o espaço vetorial constituído por todas as ℓ-uplas ordenadas $(y_1, y_2, \ldots, y_\ell)$ é denominado de ESPAÇO DE ESTADO.

DEFINIÇÃO 3.10 - Uma máquina linear cujos elementos de memória se encontram inicialmente no estado zero é denominada de MÁQUINA INERTE.

São as máquinas lineares inertes que são usadas extensivamente na codificação e decodificação de códigos lineares, como também em aplicações que requeiram tranformações de sequências.

3.6 - SHIFT-REGISTERS

DEFINIÇÃO 3.11 - Shift-registers são máquinas lineares inertes com uma entrada e uma saída onde a saída é a soma módulo-p dos dígitos de entrada previamente escolhidos.

Na Fig. 3.4 está a representação de um modelo geral para um shift-register sobre GF(p).

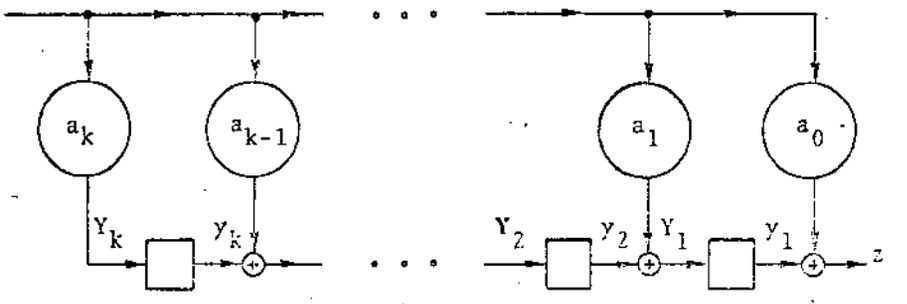


Fig. 3.4 - Representação geral de um shift-register

Observemos na Fig. 3.4 que a saída a cada instante t é a combinação linear módulo-p da entrada atual e das k entradas anteriores, isto é,

$$z(t) = a_0x(t) + a_1x(t-1) + \ldots + a_{k-1}x(t-k+1) + a_kx(t-k) \qquad (3.11)$$

Aplicando o operador linear D^i que, por definição, atrasa a variável que ele opera de i unidades de tempo, à equação (3.11), obtemos:

$$z = a_0x + a_1Dx + \ldots + a_{k-1}D^{k-1}x + a_kD^kx \qquad (3.12)$$

Dividindo a equação (3.12) por x, obtemos:

$$z/x = T(D) = a_0 + a_1D + \ldots + a_{k-1}D^{k-1} + a_kD^k \qquad (3.13)$$

DEFINIÇÃO 3.12 - A relação (3.13) é definida como a função de transferência do shift-register representado na Fig. 3.4.

3.7 - O SHIFT-REGISTER COMO UM MULTIPLICADOR DE POLINÔMIOS

Sejam os polinômios

$$a(X) = a_kX^k + a_{k-1}X^{k-1} + \ldots + a_1X + a_0$$

e $$b(X) = b_nX^n + b_{n-1}X^{n-1} + \ldots + b_1X + b_0$$

com coeficientes pertencentes a GF(p). O produto entre a(x) e b(x) é dado por:

$$p(X) = b_n a_k X^{n+k} + (b_{n-1}a_k + b_n a_{k-1})X^{n+k-1} +$$

$$+ (b_{n-2}a_k + b_{n-1}a_{k-1} + b_n a_{n-2})X^{n+k-2} + \ldots$$

$$\ldots + (b_0 a_2 + b_1 a_1 + b_2 a_0)X^2 + (b_1 a_0 + b_0 a_1)X + a_0 b_0$$

Consideremos agora o shift-register da Fig. 3.5 , que tem como multiplicadores escalares os coeficientes de a(X). Suponhamos também que os coeficientes de b(X) sejam as entradas do shift-register obedecendo à ordem decrescente das potências de X.

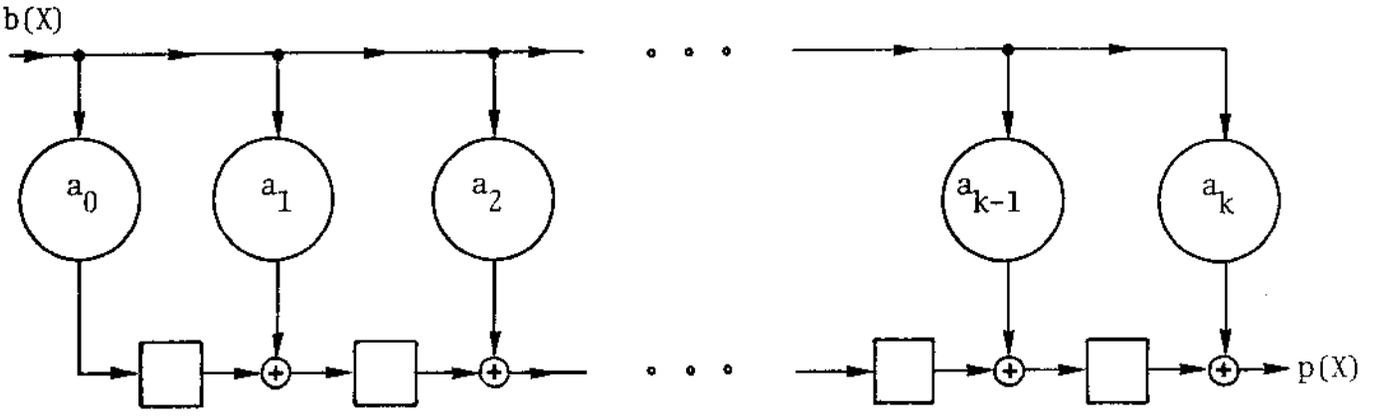


Fig. 3.5 - Circuito multiplicador de um polinômio qualquer b(X) pelo polinômio a(X)

Quando o primeiro coeficiente de b(X), b_n, aparece na entrada do circuito, a saída é $b_n a_k$, o primeiro coeficiente de p(X). O shift-register é deslocado pela aplicação de um pulso do relógio, não mostrado no circuito. Após o primeiro deslocamento, o conteúdo dos elementos de memória é $a_0 b_n$, $a_1 b_n, \ldots, a_{k-1} b_n$ e na entrada aparece o segundo coeficiente de b(X), b_{n-1}. Assim, a segunda saída do shift-register é $b_{n-1}a_k + b_n a_{k-1}$, o segundo coeficiente de p(X). Após outro deslocamento o conteúdo da memória é dado por

$$b_{n-1}a_0,\ b_n a_0 + b_{n-1}a_1,\ b_n a_1 + b_{n-1}a_2, \ldots, b_n a_{k-2} + b_{n-1}a_{k-1}$$

e a entrada é dada por b_{n-2}.

Assim, a saída será dada por

$b_{n-2}a_k + b_{n-1}a_{k-1} + b_n a_k$, o terceiro coeficiente de p(X).

O processo é repetido de modo que, após n+k-1 deslocamentos, aparece na saída o último coeficiente de p(X), $a_0 b_0$.

Portanto, o shift-register da Fig. 3.5 é capaz de efetuar a multiplicação de polinômios com coeficientes em GF(p). O shift-register da Fig. 3.4 também é um circuito multiplicador de polinômios, apenas os coeficientes do polinômio a ser multiplicado por a(X) devem entrar no circuito segundo as potências crescentes de X.

3.8 - MÁQUINAS INVERSAS

DEFINIÇÃO 3.13 - Um circuito cuja função de transferência é T(D) admite um inverso, se existe um circuito cuja função de transferência é 1/T(D).

Se uma máquina linear possui uma entrada e n saídas e se representarmos por $T_{j1}(D)$, j=1,2,...,n, a função de transferência entre a entrada e a j-ésima saída, então essa máquina admite uma inversa instantânea ou com atraso L, se existe um inteiro $L \geq 0$ tal que:

$$\text{MDC } [T_{11}(D), T_{21}(D), \ldots, T_{n1}(D)] = D^L \qquad (3.14)$$

Esse resultado está expresso em [13] como um teorema e é facilmente provado.

Estamos interessados em determinar sob que condições um shift-register admite um inverso e assim a divisão de polinômios seja possível de ser efetuada instantaneamente.

Para que a inversa seja instantânea (3.14) se reduz a:

$$\text{MDC } [T_{11}(D), T_{21}(D), \ldots, T_{n1}(D)] = 1 \qquad (3.15)$$

De (3.15) concluímos que $T_{1j}(D)$, j=1,...,n são polinômios relativamente primos e esta é a condição para o circuito admitir um inverso instantâneo. No caso de um shift-register, n = 1 e assim não tem sentido falar em MDC. Entretanto, podemos concluir que para que exista um inverso instantâneo, o coeficiente independente de D em T(D) tem que ser diferente de zero, pois do contrário T(D) pode ser colocado na forma $D^L T_1(D)$ para algum L > 0.

3.9 - O SHIFT-REGISTER COMO UM CIRCUITO DIVISOR DE POLINÔMIOS

Seja o polinômio

$$d(X) = d_n X^n + d_{n-1} X^{n-1} + \ldots + d_0$$

a ser dividido por

$$g(X) = g_r X^r + g_{r-1} X^{r-1} + \ldots + g_0.$$

Consideremos o circuito da Fig. 3.6 que é suposto inerte. As saídas do circuito durante os r primeiros deslocamentos são zeros e durante esse tempo, são armazenados nas memórias do shift-register os r primeiros coeficientes de d(X). No próximo deslocamento, a saída do circuito é dada por $d_n g_r^{-1}$, o primeiro coeficiente do quociente. Entretanto, para cada coeficiente do quociente obtido, digamos q_i, deve-se subtrair do dividendo d(X) o polinômio, $q_i(X)\ g(X)$ e no circuito essa operação é implementada através das conexões de realimentação. Após n deslocamentos o polinômio q(X) terá sido obtido na saída e o resto estará armazenado na memória do shift-register.

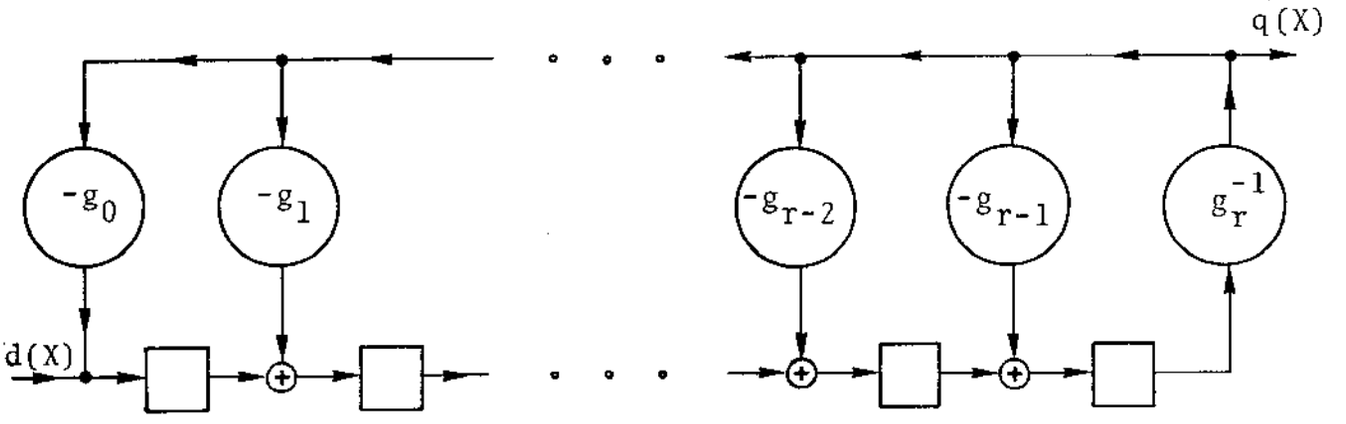


Fig. 3.6 - Circuito para dividir pelo polinômio
$g(X) = g_r X^r + g_{r-1} X^{r-1} + \ldots + g_1 X + g_0$

3.10 - MÁQUINAS LINEARES COM ENTRADAS E SAÍDAS MÚLTIPLAS

Consideremos o modelo geral para uma máquina de estado finito representado na Fig. 3.1. Para uma máquina linear, o circuito combinacional é constituído apenas de somadores módulo-p e multiplicadores escalares módulo-p. O próximo estado Y_i, do i-ésimo elemento de memória, de uma maneira geral, pode ser expresso como uma combinação linear sobre GF(p) das entradas externas e

do estado atual da máquina, isto é,

$$Y_i = (a_{i1}y_1 + a_{i2}y_2 + \ldots + a_{i\ell}y_\ell) + (b_{i1}x_1 + b_{i2}x_2 + \ldots$$

$$\ldots + b_{ik}x_k)$$

ou escrito de outra forma,

$$Y_i = \sum_{j=1}^{\ell} a_{ij}y_j + \sum_{j=1}^{k} b_{ij}x_j \qquad (3.16)$$

A equação (3.12) é denominada de EQUAÇÃO DO PRÓXIMO ESTADO para o elemento de memória Y_i. Se considerarmos as equações do próximo estado para todos elementos de memória temos a equação matricial:

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_\ell \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \cdots a_{1\ell} \\ a_{21} & a_{22} \cdots a_{2\ell} \\ \vdots & \vdots \quad \vdots \\ a_{\ell 1} & a_{\ell 2} \cdots a_{\ell\ell} \end{bmatrix} \cdot \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_\ell \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} \cdots b_{1k} \\ b_{21} & b_{22} \cdots b_{2k} \\ \vdots & \vdots \quad \vdots \\ b_{\ell 1} & b_{\ell 2} \cdots b_{\ell k} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{bmatrix} \qquad (3.17a)$$

ou então:

$$\underset{\sim}{Y}(t) = \underset{\sim}{y}(t+1) = \overline{A}\underset{\sim}{y}(t) + \overline{B}\underset{\sim}{x}(t) \; . \qquad (3.17b)$$

O vetor $\underset{\sim}{y}(t)$ é denominado de VETOR DE ESTADO ATUAL e seus componentes são as variáveis de estado. O vetor $\underset{\sim}{Y}(t)$ é o VETOR PRÓXIMO ESTADO e o vetor $\underset{\sim}{x}(t)$ é o VETOR DE ENTRADA e seus componentes são as variáveis de entrada. Cada componente $x_i(t)$ de $\underset{\sim}{x}(t)$ corresponde à entrada aplicada ao i-ésimo terminal de entrada no instante t. As matrizes $\overline{A}$ e $\overline{B}$ tem dimensões ℓ x ℓ e ℓ x k, respectivamente.

Da mesma forma, a i-ésima saída é uma combinação linear sobre GF(p)das entradas externas e do estado atual da máquina,isto é,

$$z_i = (c_{i1}y_1 + c_{i2}y_2 + \ldots + c_{i\ell}y_\ell) + (d_{i1}x_1 + \ldots$$

$$\ldots + (d_{i1}x_1 + d_{i2}x_2 + \ldots + d_{ik}x_k)$$

ou escrito de outra forma:

$$z_i = \sum_{j=1}^{\ell} c_{ij} y_j + \sum_{j=1}^{k} d_{ij} x_j \qquad (3.18)$$

A equação (3.18) é dneominada de EQUAÇÃO DE SAÍDA para o i-ésimo terminal de saída da máquina. Considerando as equações de saída para todos os terminais de saída da máquina temos a seguinte equação matricial.

$$\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} \cdots c_{1\ell} \\ c_{21} & c_{22} \cdots c_{2\ell} \\ \vdots & \vdots \quad \vdots \\ c_{n1} & c_{n2} \cdots c_{n\ell} \end{bmatrix} \cdot \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_\ell \end{bmatrix} + \begin{bmatrix} d_{11} & d_{12} \cdots d_{1k} \\ d_{21} & d_{22} \cdots d_{2k} \\ \vdots & \vdots \quad \vdots \\ d_{n1} & d_{n2} \cdots d_{nk} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{bmatrix} \qquad (3.19a)$$

ou então:

$$\underline{z}(t) = \overline{C}\underline{y}(t) + \overline{D}\underline{x}(t) \qquad (3.19b)$$

onde $\underline{z}(t)$ é o vetor de saída e seu i-ésimo componente $z_i(t)$ é a saída gerada no i-ésimo terminal de saída da máquina no instante t. As dimensões das matrizes $\overline{C}$ e $\overline{D}$ são n x ℓ e n x k respectivamente.

As matrizes $\overline{A}$, $\overline{B}$, $\overline{C}$ e $\overline{D}$ das equações (3.17b) e (3.19b) são denominadas de MATRIZES CONSTITUINTES da máquina linear.

DEFINIÇÃO 3.14 - Uma máquina sequencial é dita linear sobre um campo GF(p) se seus estados podem ser identificados como os elementos de um espaço vetorial sobre GF(p) e seu próximo estado e saída podem ser caracterizados pelo par de equações matriciais sobre GF(p) dado por:

$$\underline{Y}(t) = \overline{A}\,\underline{y}(t) + \overline{B}\,\underline{x}(t) \qquad (3.20a)$$

$$\underline{z}(t) = \overline{C}\,\underline{y}(t) + \overline{D}\,\underline{x}(t) \qquad (3.20b)$$

A dimensão da máquina é a dimensão do seu espaço do estado.

As equações (3.20) correspondem ao modelo de Mealy enquanto que se na equação (3.20b) $\overline{D}=[0]$, obtém-se o modelo de Moore. Na Fig. 3.7 está representado o modelo geral de uma máquina sequencial linear em termos de suas matrizes constituintes.

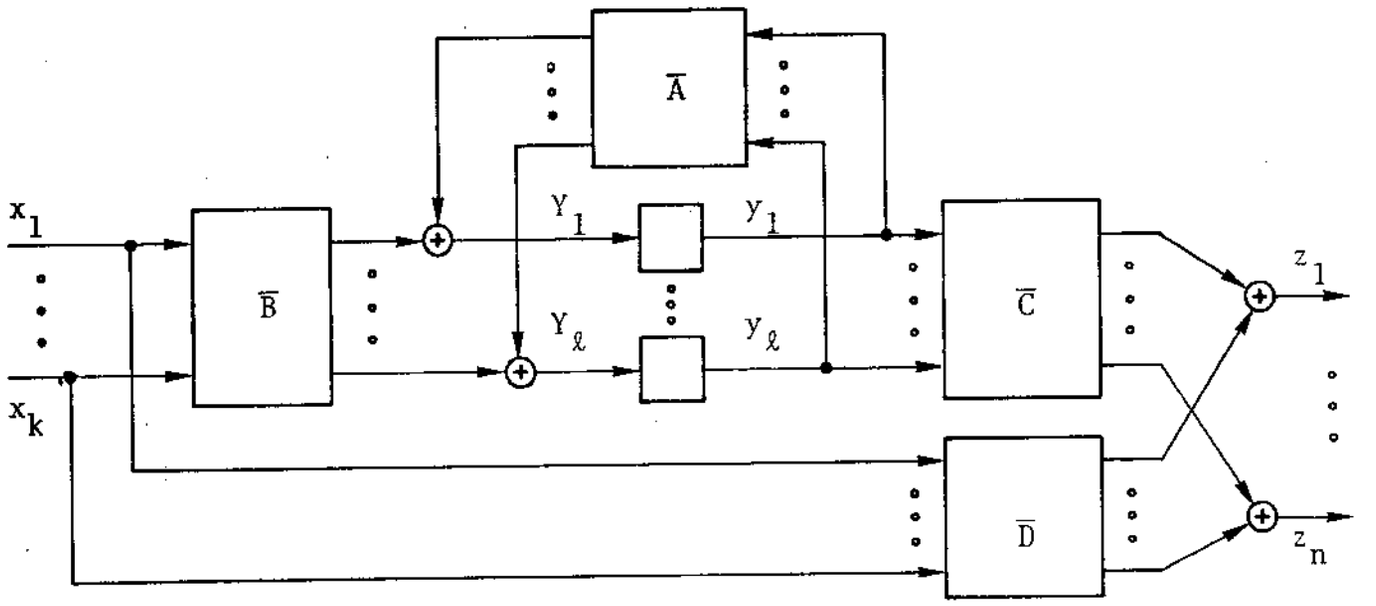


Fig. 3.7 - Modelo de uma máquina sequencial linear em termos de suas matrizes constituintes

3.11 - MATRIZ DE TRANSFERÊNCIA DE UMA MÁQUINA SEQUENCIAL LINEAR

Como as máquinas lineares formam uam sub-classe dos sistemas lineares discretos, toda a teoria para sistemas lineares discretos pode ser aplicada às máquinas sequenciais lineares. Da teoria de sistemas lineares discretos temos que a matriz de matriz de transferência do sistema é dada por: CHEN .

$$\overline{G}(z) = \overline{C}(z\overline{I} - \overline{A})^{-1}\,\overline{B} + \overline{D} \qquad (3.21)$$

onde z corresponde à transformada z , $\overline{I}$ é a matriz identidade e $\overline{A}$, $\overline{B}$, $\overline{C}$ e $\overline{D}$ são as matrizes constituintes do sistema. Entretanto, quando da análise de circuitos sequenciais, costuma-se usar o operador D apresentado na seção 3.6. É fácil de mostrar que a relação entre z e D é dada por $z=D^{-1}$. Assim, a equação 3.21 se transforma em:

$$\overline{G}(D) = \overline{C}(D^{-1}\overline{I} - \overline{A})^{-1}\,\overline{B} + \overline{D} \qquad (3.22)$$

que corresponde à matriz de transferência de uma máquina sequencial linear.

A equação (3.22) pode ser obtida diretamente das equações (3.20) pela aplicação do operador D a ambas equações e efetuando algumas operações matriciais.

Os tópicos apresentados até aqui sobre a teoria das máquinas de estado finito são suficientes para o objetivo do nosso trabalho e assim não nos aprofundaremos em outros aspectos da teoria principalmente com relação às máquinas lineares que possuem toda a estrutura dos sistemas lineares discretos.

CAPÍTULO 4

CÓDIGOS DE BLOCO LINEARES

4.1 - INTRODUÇÃO

DEFINIÇÃO 4.1 - Seja um espaço vetorial de n-uplas sobre um campo F. Um conjunto de n-uplas pertencentes a esse espaço vetorial é um CÓDIGO LINEAR se, e somente se, ele forma um sub-espaço do espaço das n-uplas.

Utilizando os resultados da álgebra apresentados no Cap.2, podemos concluir que se um conjunto de n-uplas sobre GF(p) forma um grupo, então esse conjunto é um sub-espaço vetorial e portanto um código linear. Assim, dado um conjunto de n-uplas com componentes em GF(p) para sabermos se ele forma um código linear basta mostrarmos que ele forma um grupo aditivo.

4.2 - DESCRIÇÃO DE CÓDIGOS DE BLOCO LINEARES POR MATRIZES

Como um código linear é um sub-espaço vetorial, uma forma conveniente de representá-lo é através de matrizes. Com essa idéia em mente, um código linear é o espaço das linhas de uma matriz não singular o que implica nas linhas da matriz serem linearmente independentes e portanto uma base para o código. Assim, qualquer palavra-código é uma combinação linear das linhas dessa matriz não singular.

DEFINIÇÃO 4.2 - Qualquer matriz $\overline{G}$ cujas linhas são uma base para um sub-espaço, correspondente a um código linear, é denominada de MATRIZ GERADORA DO CÓDIGO. Se o sub-espaço tem dimensão k, as dimensões de $\overline{G}$ são k x n.

Existe uma outra alternativa para a representação de um código linear através de matrizes que consiste em representar o código pela matriz geradora do espaço nulo do código; essa matriz é denominada de MATRIZ CHEQUE DE PARIDADE DO CÓDIGO. Embora esta representação seja de grande interesse na teoria da codificação, pois é a partir da matriz de paridade que se obtém as propriedades corretoras do código [8] além de desempenhar papel fundamental no problema da decodificação, nesse trabalho estamos interessados apenas com os aspectos que envolvem o processo da codificação e assim utilizaremos apenas as representações dos códigos lineares através da matriz $\overline{G}$.

DEFINIÇÃO 4.3 - Duas matrizes $\overline{G}$ e $\overline{G}'$ são ditas COMBINATORIAMENTE EQUIVALENTES quando uma é obtida a partir da outra através de operações elementares de linhas e permutações de colunas.

DEFINIÇÃO 4.4 - Dois códigos são equivalentes quando suas matrizes geradoras são combinatoriamente equivalentes.

Cada matriz geradora $\overline{G}$ é combinatoriamente equivalente a uma matriz $\overline{G}'$ na forma canônica de Echelon. A matriz $\overline{G}''$ pode ser obtida a partir de $\overline{G}$ seguindo-se os seguintes passos:

1) Como as linhas de $\overline{G}$ são linearmente independentes, obrigatoriamente existe pelo menos um elemento diferente de zero na i-ésima linha. Considere a j-ésima coluna na qual surge o primeiro elemento diferente de zero, digamos a_{ij}. Divida cada elemento da i-ésima linha pelo elemento a_{ij}. Após essa operação temos o novo elemento $a'_{ij}=1$.

2) Some a cada uma das outras linhas ℓ, a linha ℓ' obtida pela multiplicação da linha i pelo elemento $-a_{\ell j}$. O resultado após essas operações é que na j-ésima coluna a linha i possui 1 e todas as outras possuem zero.

3) Os passos 1 e 2 são repetidos para cada linha da nova matriz obtida e no final obtém-se a matriz $\overline{G}'$ na forma canônica de Echelon que possui k colunas, cada uma possuindo um 1 e k-1 zeros, e cada linha possui um desses 1's.

Observemos que as operações efetuadas até aqui são operações elementares de linhas e assim as duas matrizes $\overline{G}$ e $\overline{G}'$ geram o mesmo código.

4) Permutando-se as colunas de $\overline{G}'$, as k colunas que têm o primeiro 1 de cada linha podem ser colocadas na forma de uma sub-matriz identidade de dimensões k x k e assim a matriz combinatoriamente equivalente de $\overline{G}$, $\overline{G}''$, pode ser colocada na forma

$$\overline{G}'' = \left[\begin{array}{ccc|ccc} 1 & 0\ldots & 0 & p_{11} & p_{12}\cdots & p_{1,n-k} \\ 0 & 1\ldots & 0 & p_{21} & p_{22}\cdots & p_{2,n-k} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0\ldots & 1 & p_{k1} & p_{k2}\cdots & p_{k,n-k} \end{array}\right] = [I_k \mid P] \qquad (4.1)$$

que é dita estar na FORMA REDUZIDA DE ECHELON.

Seja agora um vetor de informação $\underline{v}=(a_1,a_2,\ldots,a_k)$ e suponhamos que desejamos obter a palavra-código, correspondente a esse bloco de informação, pertencente ao código gerado por $\overline{G}$". Seja $\underline{u}=(u_1,u_2,\ldots,u_n)$ a correspondente palavra-código. Observemos que se efetuarmos o produto $\underline{v}\overline{G}$" obtemos o vetor

$$\underline{u} = (a_1,a_2,\ldots,a_k\ c_1,c_2,\ldots,c_{n-k}) \qquad (4.2)$$

onde $$c_j = \sum_{i=1}^{k} a_i p_{ij}$$

A palavra-código $\underline{u}$, obtida em (4.2), tem as k primeiras componentes idênticas ao vetor de informação $\underline{v}$ e as n-k componentes restantes são combinações lineares das k componentes de $\underline{v}$.

DEFINIÇÃO 4.5 - Um código cujas palavras-código estão na forma da equação (4.2) é denominado de CÓDIGO SISTEMÁTICO.

Como cada matriz geradora possui uma matriz combinatoriamente equivalente na forma reduzida de Echelon, temos o seguinte teorema:

TEOREMA 4.1 - Cada código linear é equivalente a um código sistemático.

Esse resultado é muito importante porque se a codificação foi feita na forma sistemática, a decodificação é enormemente facilitada [8]. Tendo em vista o objetivo final do nosso trabalho, nesse ponto cabe a seguinte questão: Qual o comportamento das densidades espectrais de dois códigos equivalentes, um deles estando na forma sistemática? Essa é uma questão que procuraremos responder no final do trabalho.

DEFINIÇÃO 4.6 - Um código cuja matriz geradora tem dimensões kxn é denominado de CÓDIGO(n,k), onde n é o comprimento do código e k o número de dígitos nos blocos de informação.

4.3 - ESTRUTURA ALGÉBRICA DOS CÓDIGOS DE BLOCO LINEARES

A estrutura algébrica dos códigos de bloco lineares se fundamenta no fato de que sempre podemos fazer uma correspondência um a um entre n-uplas e um polinômio de grau menor que n. Isto é, $(a_0,a_1,a_2,\ldots,a_{n-1})$ pode ser representado pelo polinômio:

$$f(X) = a_0 + a_1X + a_2X^2 + \ldots + a_{n-1}X^{n-1}.$$

Tendo esta correspondência em mente, a álgebra de polinômios módulo um polinômio f(x) de grau n, apresentada no Cap. 2,, seção 2.10 , é equivalente a uma álgebra linear de n-uplas. Portanto, utilizando os teoremas 2.20, 2.21, 2.22 e 2.23 podemos fazer a seguinte correspondência:

Cada ideal J na álgebra de polinômios módulo um polinômio f(X) de grau n corresponde a um código linear (n,k), onde k é a dimensão do espaço vetorial que tem como uma de suas bases o conjunto de polinômios

$$\{g(X),\ Xg(X),\ldots,X^{k-1}g(X)\} \tag{4.3}$$

onde g(X) é um fator de f(X) e k é o grau de h(X) dado por f(X)/ /g(X).

Assim, a matriz geradora de um código linear é dada por

$$\overline{G} = \begin{bmatrix} X^{k-1}g(X) \\ \vdots \\ Xg(X) \\ g(X) \end{bmatrix} \tag{4.4}$$

4.4 - CODIFICAÇÃO DE CÓDIGOS DE BLOCO LINEARES

Como vimos pelo teorema 2.18, um ideal de polinômios é constituído por todos os múltiplos de um polinômio.Assim,o ideal gerado por g(X) na álgebra de polinômios módulo um polinômio f(X) de grau n, é constituído por todos os múltiplos de g(X) módulo f(X). Se considerarmos cada bloco de informação de k componentes como um polinômio b(X) de grau k-1, b(X) g(X) pertence ao ideal e portanto representa a palavra-código correspondente ao bloco de informação cujas componentes são os coeficientes de b(X). Portanto, a codificação será feita facilmente através do circuito multiplicador de polinômios discutido na seção 3.7. Na Fig. 3.5 os escalares dos multiplicadores correspondem aos coeficientes de g(X).

4.5 - CÓDIGOS CÍCLICOS

Dentre os códigos de bloco lineares, existe uma sub-

-classe de códigos denominados de códigos cíclicos que tem recebido maiores estudos devido suas propriedades particulares. A seguir, apresentamos algumas dessas particularidades.

DEFINIÇÃO 4.7 - Um sub-espaço V de n-uplas é chamado de SUB-ESPAÇO CÍCLICO se para cada vetor $\underline{v}=(a_{n-1},a_{n-2},\ldots,a_0)$ pertencente a V o vetor $\underline{v}'=(a_0,a_{n-1},a_{n-2},\ldots,a_1)$, obtido de $\underline{v}$ pelo deslocamento de suas componentes uma unidade para a direita, também pertence a V.

TEOREMA 4.2 - Na álgebra de polinômios módulo X^n-1, um sub- espaço é um sub-espaço cíclico se, e somente se, é um ideal.

PROVA:

O ponto básico para a prova é que multiplicação por$\{X\}$ é o mesmo que um deslocamento cíclico porque

$$X(a_{n-1}X^{n-1} + a_{n-2}X^{n-2} + \ldots + a_0) =$$

$$= a_{n-1}X^n + a_{n-2}X^{n-1} + \ldots + a_0X =$$

$$= a_{n-1}X^n + a_{n-2}X^{n-1} + \ldots + a_0X + a_{n-1} - a_{n-1} =$$

$$= a_{n-1}(X^n-1) + a_{n-2}X^{n-1} + \ldots + a_0X + a_{n-1} \qquad (4.5a)$$

O polinômio (4.5a) módulo X^{n-1} é simplesmente

$$a_{n-2}X^{n-1} + \ldots + a_0X + a_{n-1} \qquad (4.5b)$$

que é um deslocamento cíclico para a esquerda de

$$a_{n-1}X^{n-1} + a_{n-2}X^{n-2} + \ldots + a_0.$$

Assim, se o sub-espaço V é um ideal e $\underline{v}$ é um elemento pertencente a V, então $\{X\}\underline{v}$ pertence também a V. Como $\{X\}\underline{v}$ com deslocamento cíclico uma unidade para a esquerda, então V é um sub-espaço cíclico.

Suponhamos agora que V é um sub-espaço cíclico. Então, para qualquer $\underline{v}$ pertencente a V, $\{X\}\underline{v}$ pertence também a V e portanto para qualquer j

$$\{X\}^j\underline{v} = \{X^j\}\underline{v} \qquad (4.5c)$$

pertence também a V. Como V é por hipótese um sub-espaço , qualquer combinação linear

$$c_{n-1}\{X^{n-1}\}\,x + c_{n-2}\{X^{n-2}\}\,x + \ldots + c_0 v =$$

$$= \{c_{n-1}X^{n-1} + c_{n-2}X^{n-2} + \ldots + c_0\}\,x \qquad (4.5d)$$

também pertence a V. Assim, o produto de qualquer elemento da álgebra por qualquer elemento de V eprtence a V e portanto V é um ideal.

Q.E.D.

Os teoremas 2.20, 2.21, 2.22 e 2.23 aplicados à álgebra de polinômios módulo X^n-1 nos fornecem os códigos cíclicos cujas matrizes geradoras são da forma 4.4, sendo que g(X) obrigatoriamente é um fator de X^n-1.

4.6 - CODIFICAÇÃO DE CÓDIGOS CÍCLICOS

A codificação de códigos de bloco lineares discutida na seção 4.4 também se aplica a códigos cíclicos. Entretanto,devido as suas propriedades particulares, a codificação de códigos cíclicos pode ser efetuada de outra maneira. Essa outra forma de codificação se baseia no seguinte teorema que será apresentado sem prova, já que sua prova se baseia em alguns tópicos de álgebra não apresentados por nós no Cap. 2.

TEOREMA 4.3 - Seja

$$h(X) = \sum_{j=0}^{k} h_j X^j \quad , \quad h_0 \neq 0 \quad e \quad h_k = 1$$

e seja n o menor inteiro positivo, tal que X^n-1 é divisível por h(X). Seja ainda

$$g(X) = X^n-1/h(X).$$

Então as soluções de

$$\sum_{j=0}^{k} h_j a_{i+j} = 0$$

ou

$$a_{i+k} = \sum_{j=0}^{k-1} h_j a_{i+j} \qquad (4.6a)$$

conhecida como relação de recorrência, são periódicas de período

n e o conjunto constituído pelo primeiro período de cada possível solução, considerados como polinômios módulo X^n-1 da forma

$$a(X) = a_0 X^{n-1} + a_1 X^{n-2} + \ldots + a_{n-2}X + a_{n-1} \quad (4.6b)$$

é o ideal gerado por g(X) na álgebra de polinômios módulo X^n-1.

Observemos que em (4.6b) os coeficientes de maior grau correspondem aos de menor grau nos polinômios pertencentes ao ideal gerado por g(X) na álgebra de polinômios módulo X^n-1.

Baseado no teorema 4.3 obtém-se um codificador para códigos cíclicos que consiste simplesmente em um circuito divisor de polinômios discutido na seção 3.9 mas com a ordem dos coeficientes do polinômio divisor e com o coeficiente de maior grau igual a 1 já que $h_k=1$. Na Fig. 4.1 está a representação do circuito que é conhecido como o codificador de k estágios. O codificador discutido na sub-seção 4.4 é denominado de CODIFICADOR DE n-k ESTÁGIOS

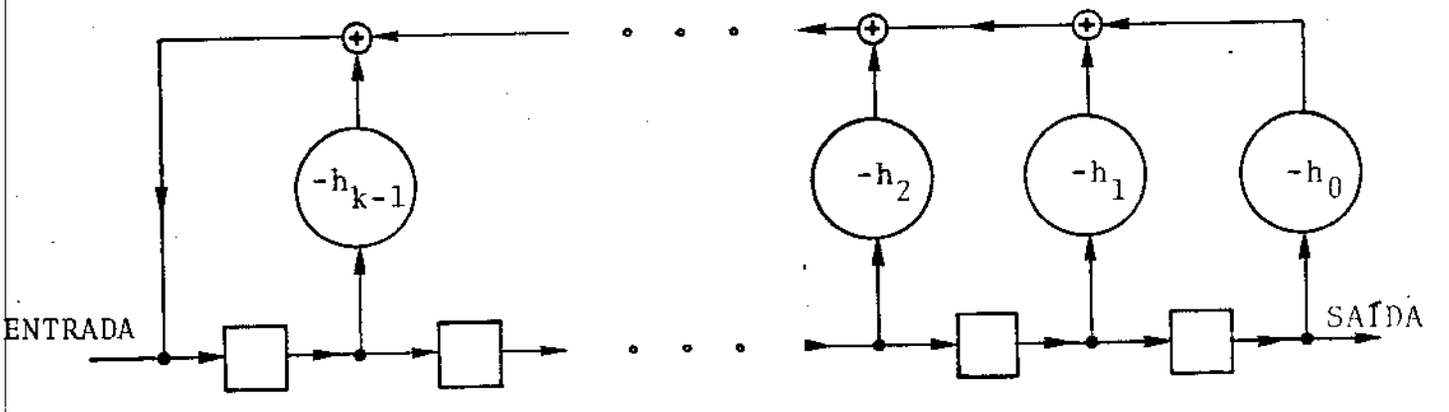


Fig. 4.1 - Codificador de k estágios para um código cíclico com polinômio gerador g(X); $h(X) = X^n-1/g(X)$.

Observemos que os códigos cíclicos dão uma opção para a escolha do codificador. Assim dependendo dos valores de n e k pode-se escolher um dos dois tipos de codificador. Os códigos cíclicos são realmente os códigos de bloco lineares mais estudados pois possuem uma estrutura algébrica, não apresentada aqui, que permite a construção de códigos com a capacidade corretora desejada. Essa estrutura algébrica é apresentada como própria de uma classe de códigos denominada de CÓDIGOS B.C.H., à qual pertencem todos os códigos cíclicos.

CAPÍTULO 5

CÓDIGOS CONVOLUCIONAIS

5.1 - DESCRIÇÃO DE CÓDIGOS DE ÁRVORE POR MATRIZES

Seja $\overline{F}_i$ uma matriz cujas k_0 linhas são vetores de comprimento semi-infinito, linearmente independentes e com componentes pertencentes a GF(p). Suponhamos que as primeiras $(i-1)n_0$ colunas de $\overline{F}_i$ são zeros e que alguns elementos nas colunas $(i-1)n_0+1$ até in_0 são diferentes de zero.

DEFINIÇÃO 5.1 - Um código de árvore linear é definido como sendo o conjunto dos semi-infinito vetores-linha que é o espaço das linhas da matriz

$$\overline{G} = \begin{bmatrix} \overline{F}_1 \\ \overline{F}_2 \\ \overline{F}_3 \\ \vdots \end{bmatrix} \tag{5.1}$$

que é a matriz geradora do código.

A forma geral da matriz geradora de um código de árvore linear está representada na Fig. 5.1 onde as áreas hachuriadas correspondem aos zeros das $(i-1)n_0$ colunas de cada matriz $\overline{F}_i$.

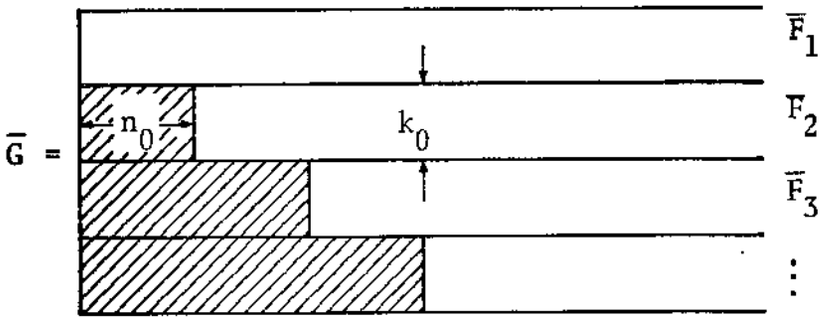


Fig. 5.1 - Modelo geral da matriz geradora de um código de árvore linear

Uma sequência codificada semi-infinita $\underline{c}$, num código de árvore linear, é obtida a partir de uma sequência de informação $\underline{i}$, também semi-infinita, através da seguinte relação [8] :

$$\underline{c} = \underline{i}\overline{G} \tag{5.2a}$$

Se considerarmos $\underline{c}$ e $\underline{i}$ como vetores-coluna temos:

$$\underline{c} = \overline{G}^T\underline{i} \tag{5.2b}$$

onde T indica transposição.

5.2 - ESTRUTURA DOS CÓDIGOS CONVOLUCIONAIS

Os códigos convolucionais, também chamados de códigos recorrentes, formam uma classe de códigos de árvore mais interessantes. A matriz geradora de um código convolucional é obtida através da imposição de que cada $\overline{F}_i$ em (5.1) seja uma versão transladada de $\overline{F}_1$. Isto é, F_i é obtida deslocando-se F_1 por $n_0 i$ posições à direita, $i > 1$, e com as $n_0 i$ posições situadas à esquerda preenchidas com zeros.

Observemos que as sequências-código são vetores semi-infinitos, o que implica no codificador e decodificador possuir um número semi-infinito de elementos de memória e assim suas implementações são impraticáveis.

Seja então n o número de dígitos que pode ser armazenado na memória do decodificador. O conjunto das palavras- código de comprimento n empregadas na decodificação dos primeiros n_0 dígitos constituintes de um bloco de informação de comprimento n_0, forma um grupo sobre a adição [8] e assim um sub-espaço linear do espaço das n-uplas.

A matriz geradora desse espaço tem dimensões k x n e tem a seguinte forma:

$$\overline{G} = \begin{bmatrix} \overline{G}_0 & \overline{G}_1 & \overline{G}_2 & \cdots & \overline{G}_{m-1} \\ & \overline{G}_0 & \overline{G}_1 & \cdots & \overline{G}_{m-2} \\ & & \overline{G}_0 & \cdots & \overline{G}_{m-3} \\ & & & & \vdots \\ & & & & \overline{G}_0 \end{bmatrix} \qquad (5.3)$$

onde as sub-matrizes $\overline{G}_i$ tem dimensões k_0 x n_0. A matriz $\overline{G}_0$ é sempre escolhida de modo que tenha rank k_0, o que implica em $\overline{G}$ ter rank $k = mk_0$.

DEFINIÇÃO 5.2 - O espaço das linhas da matriz $\overline{G}$ dada por (5.3) é dito um CÓDIGO CONVOLUCIONAL (n,k), onde $n = mn_0$, $k = mk_0$ e m a memória do código. As primeiras k_0 linhas de $\overline{G}$ formam uma matriz denominada de MATRIZ GERADORA BÁSICA DO CÓDIGO.

Através de operações elementares de linhas e de permutações de colunas em $\overline{G}$ é possível obter-se uma matriz G' combinatoriamente equivalente à matriz $\overline{G}$ que tem a seguinte forma

$$\overline{G}' = \begin{bmatrix} \overline{IP}_0 & \overline{OP}_1 & \overline{OP}_2 & \cdots & \overline{OP}_{m-1} \\ & \overline{IP}_0 & \overline{OP}_1 & \cdots & \overline{OP}_{m-2} \\ & & \ddots & \ddots & \vdots \\ & & & & \overline{IP}_0 \end{bmatrix} \qquad (5.4)$$

onde $\overline{I}$ é uma sub-matriz identidade de dimensões $k_0 \times k_0$, $\overline{P}_i$ é uma sub-matriz qualquer de dimensões $k_0 \times (n_0-k_0)$ e $\overline{O}$ é uma sub - matriz nula de dimensões $k_0 \times k_0$.

Como $\overline{G}$ e $\overline{G}'$ são combinatoriamente equivalentes, pelo teorema 4.1, $\overline{G}'$ gera um código sistemático equivalente ao código gerado por $\overline{G}$.

Do mesmo modo que os códigos de bloco lineares, os códigos convolucionais podem ser caracterizados pela matriz geradora do espaço nulo, a matriz cheque de paridade, que permite identificar as propriedades corretoras do código [8]. Entretanto, para os códigos convolucionais também é possível a obtenção das propriedades corretoras do código a partir da matriz $\overline{G}$. Em [9] esta propriedade está expressa sob a forma de um teorema que tem o seguinte significado:

"A distância mínima de um código convolucional é dada pela menor das distâncias de Hamming obtidas nas k_0 primeiras linhas de $\overline{G}$, isto é, a menor distância de Hamming obtida entre as linhas da matriz básica da código."

Conhecendo-se a distância mínima do código, a sua capacidade corretora é dada através da relação

$$d_{min} = 2t + 1$$

onde t é a capacidade corretora do código.

Esses resultados são muito importantes porque nos permitem a construção de códigos convolucionais com a capacidade corretora desejada. O único cuidado que se deve ter é atentar para manter a independência linear entre as linhas da matriz geradora básica.

5.3 - CODIFICADORES PARA CÓDIGOS CONVOLUCIONAIS

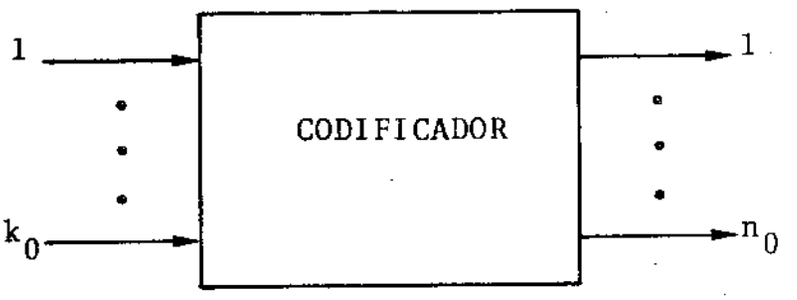


Fig. 5.2 - Forma geral de um codificador para um código convolucional (n,k), $n=mn_0$ e $k=mk_0$

A Fig. 5.2 representa um codificador geral para um código convolucional (mn_0, mk_0). O codificador aceita k_0 dígitos de informação, pertencentes a GF(p), como entrada e produz n_0 dígitos codificados na sua saída, $n_0 > k_0$. Esses dígitos de saída são combinações lineares sobre GF(p) dos m blocos de k_0 dígitos de informação recebidos anteriormente pelo codificador.

Existem dois tipos gerais de codificadores para códigos convolucionais. O codificador de $k=mk_0$ estágios que pode ser usado para qualquer código convolucional e o codificador (m-1) (n_0-k_0) estágios usado apenas para códigos convolucionais na forma sistemática.

5.3.1 - CODIFICADOR DE mk_0 ESTÁGIOS

Consideremos a matriz geradora do código (mn_0, mk_0) representada pela equação (5.3) e seja g (i,j) o elemento (i,j) das sub-matrizes $\overline{G}_\ell$, $\ell = 0,1,\ldots,m-1$, que constituem a matriz geradora básica, isto é, $\overline{G}_\ell$ é da forma

$$\overline{G}_\ell = \begin{bmatrix} g_\ell(1,1) & g_\ell(1,2) & \cdots & g_\ell(1,n_0) \\ g_\ell(2,1) & g_\ell(2,2) & \cdots & g_\ell(2,n_0) \\ \vdots & \vdots & & \vdots \\ g_\ell(k_0,1) & g_\ell(k_0,2) & \cdots & g_\ell(k_0,n_0) \end{bmatrix} \qquad (5.5)$$

Seja o polinômio sub-gerador $g_{ij}(D)$ definido por:

$$g_{ij}(D) \triangleq g_0(i,j) + g_1(i,j)D + \ldots + g_{m-1}(i,j)D^{m-1} \tag{5.6}$$

onde D é o operador atrasador. Existem n_0k_0 destes polinômios que especificam completamente a matriz geradora fundamental do código.

Seja

$$m_i(D) = \sum_{\ell=0}^{m-1} m_{\ell i}D^{\ell}$$

a representação polinomial da sequência de dígitos de informação que alimenta a i-ésima entrada do codificador. Observemos que essa sequência após m-1 unidades de tempo estará armazenada na memória do codificador o que corresponde a um armazenamento de mk_0 dígitos de informação, cada entrada tendo fornecido m - dígitos. Após transcorridas essas m-1 unidades de tempo é que obtém-se em cada saída do codificador um dígito codificado que é uma combinação linear dos mk_0 dígitos de informação armazenados na memória do codificador até o instante m-1.

A codificação poderá ser efetuada através dos produtos

$$m_i(D)\ g_{ij}(D)\ ;\ i=1,2,\ldots,k_0\ ;\ j=1,2,\ldots,n_0 \tag{5.7}$$

Esses produtos, para i fixo, representam a influência que os m dígitos armazenados através da i-ésima entrada dão a cada uma das saídas j. Então, se somarmos a influência que os m dígitos de informação obtidos através de cada entrada fornecem à j-ésima saída, temos em cada saída a sequência codificada correspondente, isto é,

$$c_j(D) = \sum_{i=1}^{k_0} m_i(D)\ g_{ij}(D)\ ;\ j=1,2,\ldots,n_0 \tag{5.8}$$

O j-ésimo dígito codificado obtido no instante m-1 será dado pelo coeficiente de D^{m-1} em (5.8) que é dado por

$$\sum_{i=1}^{k_0} [m_{0i}g_{m-1}(i,j) + m_{1i}g_{m-2}(i,j) + \ldots + m_{(m-1)i}g_0(i,j)] \tag{5.9}$$

onde os primeiros índices dos coeficientes m representam os instantes em que o dígito m entrou no codificador através da entrada i.

Observemos que a expressão (5.9) é a mesma coisa que o produto

$$(m_{01}\ m_{02}\cdots m_{0k_0} \mid m_{11}\ m_{12}\cdots m_{1k_0} \mid \cdots \mid m_{(m-1)1}\ m_{(m-1)2}\cdots m_{(m-1)k_0})\overline{G}.$$

Utilizando a expressão (5.9) podemos então esquematizar o codificador da Fig. 5.3.

5.3.2 - CODIFICADOR DE $(m-1)(n_0-k_0)$ ESTÁGIOS

O codificador da Fig. 5.3 é válido para qualquer código convolucional. Entretanto, em se tratando de códigos convolucionais sistemáticos, pode-se obter um codificador com um número bem menor de memórias, mais precisamente $(m-1)(n_0-k_0)$ elementos de memória. Isso se deve ao fato de precisarmos determinar apenas (n_0-k_0) dígitos codificados, pois k_0 já são conhecidos e, como consequência, são necessárias apenas (n_0-k_0) multiplicações do tipo representado pela equação (5.8).

Para um código sistemático,

$$\overline{G}_0 = [\overline{I}_{k_0} \mid \overline{P}_0] \quad \text{e} \quad \overline{G}_\ell = [\overline{O} \mid \overline{P}_\ell] \ ; \ \ell=1,2,\ldots,m-1.$$

Então, para $j \leqslant k_0$, os polinômios sub-geradores definidos por (5.6) são assim simplificados para

$$g_{ij}(D) = 1 \ ; \ i=j \leqslant k_0$$

$$= 0 \ ; \ i\neq j \leqslant k_0$$

Para $j > k_0$ temos:

$$g_{ij}(D) = p_0(i,k) + p_i(i,k)D + \ldots + p_{m-1}(i,k)D^{m-1}$$

$$k = 1,2,\ldots,n_0-k_0$$

Podemos então obter o codificador esquematizado da Fig. 5.4.

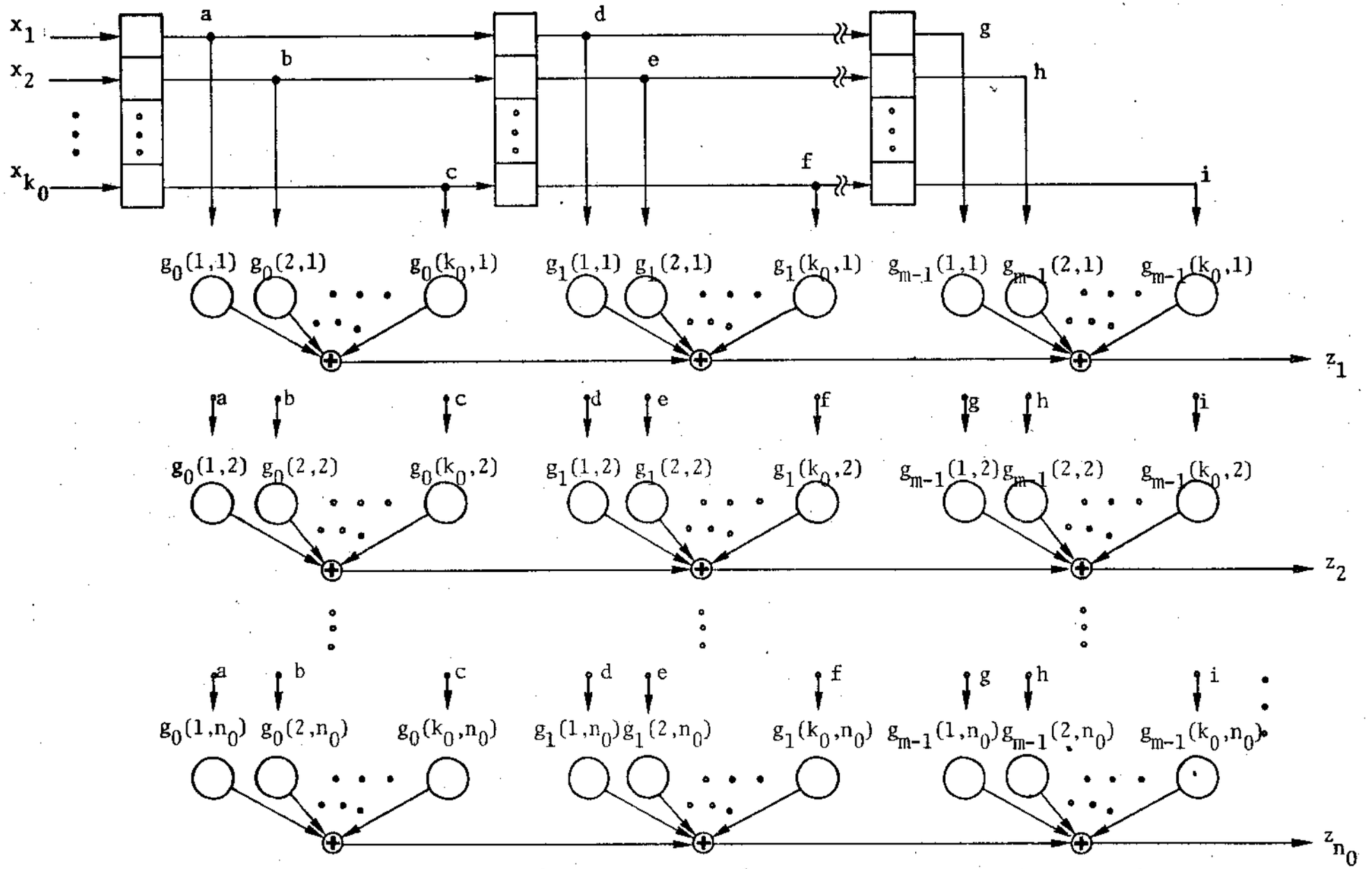
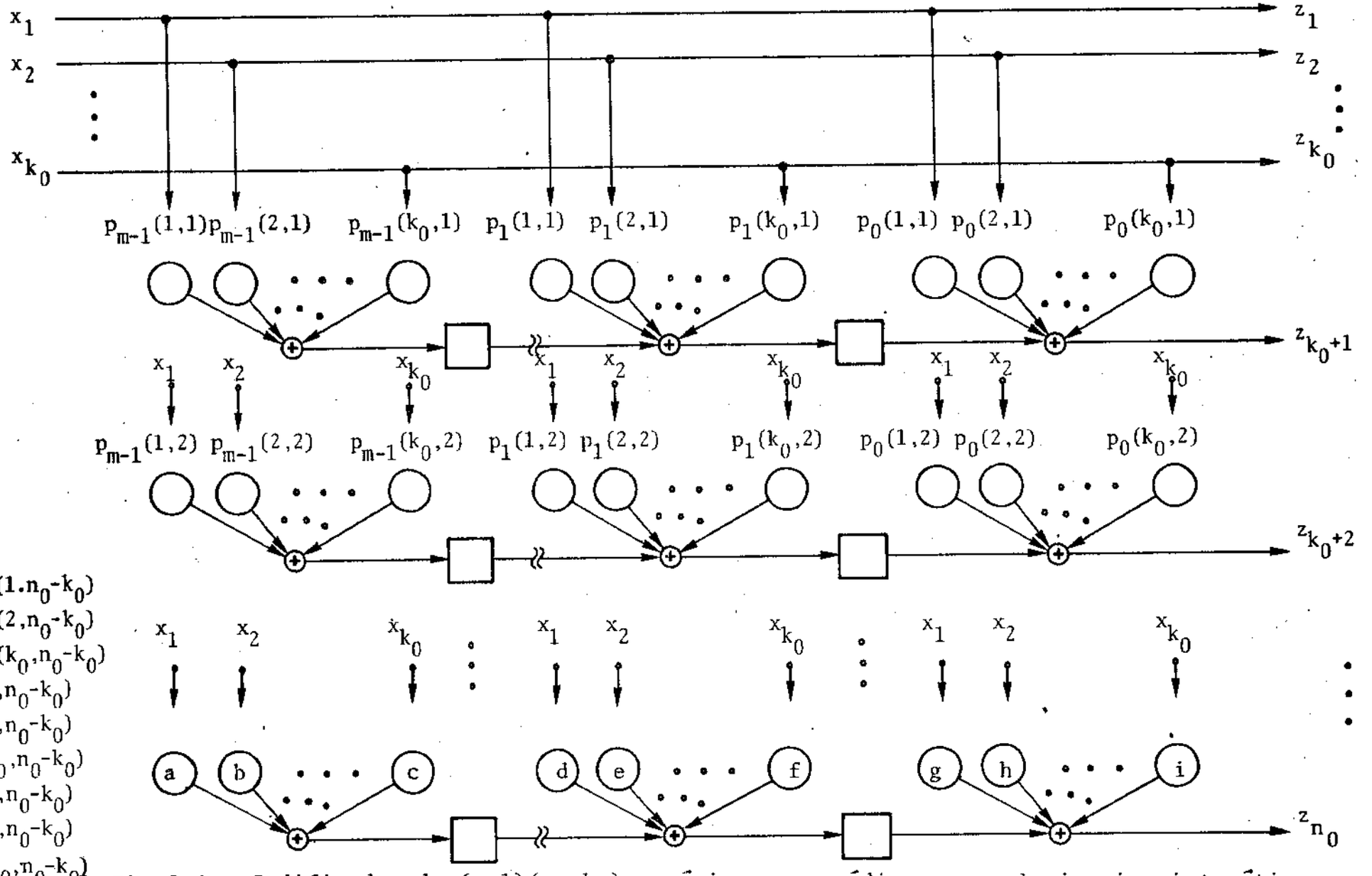


Fig. 5.3 - Codificador de $m.k_0$ memórias para um código convolucional (mn_0, m_{k_0})



a) $p_{m-1}(1.n_0-k_0)$
b) $p_{m-1}(2,n_0-k_0)$
c) $p_{m-1}(k_0,n_0-k_0)$
d) $p_1(1,n_0-k_0)$
e) $p_1(2,n_0-k_0)$
f) $p_1(k_0,n_0-k_0)$
g) $p_0(1,n_0-k_0)$
h) $p_0(2,n_0-k_0)$
i) $p_0(n_0,n_0-k_0)$

Fig.5.4 - Codificador de $(m-1)(n_0-k_0)$ memórias para códigos convolucionais sistemáticos

5.4 - O CODIFICADOR DE CÓDIGOS CONVOLUCIONAIS SOB O PONTO DE VISTA DA TEORIA DAS MÁQUINAS LINEARES DE ESTADO FINITO

5.4.1 - MATRIZES CONSTITUINTES DO CODIFICADOR

O codificador de um código convolucional sempre pode ser construído a partir da matriz geradora do código de acordo com um dos modelos apresentados nas Figs. 5.3 e 5.4. Como os componentes de circuito utilizados em sua implementação são todos lineares, todos os tópicos discutidos na seção 3.5 podem ser aplicados ao mesmo. Em particular, sob o ponto de vista da análise espectral de potência do código, é importante que o codificador possa ser modelado como uma máquina de Melay. Observando o codificador de mk_0 estágios apresentado na Fig. 3, verifica-se que as saídas do circuito são combinações lineares das saídas dos mk_0 elementos de memória do circuito, isto é, combinações lineares das mk_0 variáveis de estado do circuito. Assim, o modelo do codificador apresentado na Fig. 5.3 corresponde à máquina de Moore que, como veremos no Cap. 6, não tem interêsse sob o ponto de vista da análise do espectro de potência do código. Para transformarmos o codificador da Fig. 5.3 no modelo de Mealy basta suprimir o primeiro conjunto de k_0 memórias a partir da entrada do circuito. Feito isto, a saída do circuito a cada instante é função da entrada naquele instante e das m-1 entradas anteriores, isto é, é função da entrada atual e das $(m-1)k_0$ variáveis de estado atuais do circuito e isto corresponde exatamente ao modelo de Mealy. É fácil de observar que o codificador para códigos convolucionais sistemáticos apresentado na Fig. 5.4 já está de acordo com o modelo de Mealy.

O próximo passo após obter o codificador satisfazendo ao modelo de Mealy é obter as matrizes constituintes do codificador que satisfazem às equações 3.20. Para isso procede-se do seguinte modo:

1) Rotulam-se as saídas dos elementos de memória com as variáveis de estado y_i, $i=1,2,\ldots,(m-1)k_0$ para o codificador da Fig. 5.3 no modelo de Mealy e $i=1,2,\ldots,(m-1)(n_0-k_0)$ para o codificador da Fig. 5.4.

2) Rotulam-se os entradas dos elementos de memória com o próximo estado Y_i, i com a mesma variação do item 1.

3) A partir do circuito escrevem-se as equações de próximo estado e de saída dadas por (3.17a) e (3.19a), respectivamente.

EXEMPLO 7 - Consideremos o código convolucional sistemático binário (12,9) [8] cuja matriz geradora é dada por

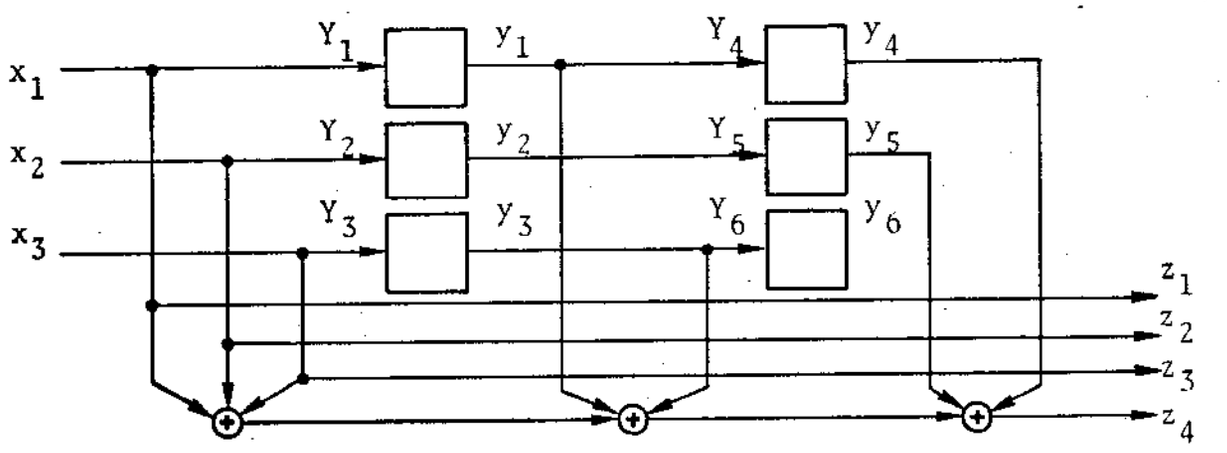
$$\overline{G} = \begin{bmatrix} 1\ 0\ 0\ 1 & 0\ 0\ 0\ 1 & 0\ 0\ 0\ 1 \\ 0\ 1\ 0\ 1 & 0\ 0\ 0\ 0 & 0\ 0\ 0\ 1 \\ 0\ 0\ 1\ 1 & 0\ 0\ 0\ 1 & 0\ 0\ 0\ 0 \\ & 1\ 0\ 0\ 1 & 0\ 0\ 0\ 1 \\ & 0\ 1\ 0\ 1 & 0\ 0\ 0\ 0 \\ & 0\ 0\ 1\ 1 & 0\ 0\ 0\ 1 \\ & & 1\ 0\ 0\ 1 \\ & & 0\ 1\ 0\ 1 \\ & & 0\ 0\ 1\ 1 \end{bmatrix}$$

Observemos que $k_0=3$, $n_0=4$ e $m=3$. As sub-matrizes que formam a matriz geradora básica são:

$$\overline{G}_0 = \begin{bmatrix} 1\ 0\ 0\ 1 \\ 0\ 1\ 0\ 1 \\ 0\ 0\ 1\ 1 \end{bmatrix} \quad ; \quad \overline{G}_1 = \begin{bmatrix} 0\ 0\ 0\ 1 \\ 0\ 0\ 0\ 0 \\ 0\ 0\ 0\ 1 \end{bmatrix} \quad ; \quad \overline{G}_2 = \begin{bmatrix} 0\ 0\ 0\ 1 \\ 0\ 0\ 0\ 1 \\ 0\ 0\ 0\ 0 \end{bmatrix}$$

Para o caso binário, os multiplicadores escalares se reduzem a conexões ou curto-circuito quando o escalar multiplicador é 1 e a não conexões ou circuito aberto quando o escalar multiplidor é 0. Os somadores módulo-2 são portas OU-EXCLUSIVO.

O codificador para o código (12,9) de acordo com o modelo da Fig. 5.3 e já na forma de Mealy é o seguinte:



Escrevendo as equações de próximo estado a partir do codificador temos:

$$Y_1 = x_1$$

$$Y_2 = x_2$$

$$Y_3 = x_3$$

$$Y_4 = y_1$$

$$Y_5 = y_2$$

$$Y_6 = y_3$$

Escrevendo as equações de saída a partir do codificador temos:

$$z_1 = x_1$$

$$z_2 = x_2$$

$$z_3 = x_3$$

$$z_4 = x_1+x_2+x_3+y_1+y_3+y_4+y_5$$

Na forma matricial, obtemos:

$$\begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \\ Y_5 \\ Y_6 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

e as matrizes constituintes do codificador são:

$$\overline{A} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad ; \quad \overline{B} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\overline{C} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \quad ; \quad \overline{D} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

5.4.2 - MATRIZ DE TRANSFERÊNCIA DO CODIFICADOR

A matriz de transferência do codificador de um código convolucional pode ser obtida a partir da equação (3.22), desde que as matrizes constituintes sejam conhecidas e já mostramos como proceder para obtê-las. Entretanto, o uso da equação (3.22) implica numa inversão de matriz normalmente de dimensões maiores que três, o que torna o processo muito trabalhoso. Assim procuramos desenvolver um método que nos permita obter a matriz de transferência do codificador sem efetuar a inversão de matrizes e, como veremos, o resultado é bastante simples e nos permite obter a matriz de transferência diretamente a partir da matriz geradora do código.

Consideremos então a expressão (5.8) que representa a sequência codificada da j-ésima saída do codificador e repetida aqui por conveniência como (5.10).

$$c_j(D) = \sum_{i=1}^{k_0} m_i(D)\, g_{ij}(D) \quad ; \quad j=1,2,\ldots,n_0 \tag{5.10}$$

Sejam os vetores-linha $\underline{z}(D)$ e $\underline{x}(D)$ definidos por

$$\underline{z}(D) \triangleq (c_1(D)\ ,\ c_2(D)\ ,\ldots..,\ c_{n_0}(D)) \tag{5.11}$$

$$\underline{x}(D) \triangleq (m_1(D)\ ,\ m_2(D)\ ,\ldots..,\ m_{k_0}(D)) \tag{5.12}$$

Seja ainda a matriz $\overline{H}(D)$ definida por

$$\overline{H}(D) \triangleq \begin{bmatrix} g_{11}(D) & g_{12}(D) & \cdots & g_{1n_0}(D) \\ g_{21}(D) & g_{22}(D) & \cdots & g_{2n_0}(D) \\ \vdots & \vdots & & \vdots \\ g_{k_0 1}(D) & g_{k_o 2}(D) & & g_{k_0 n_0}(D) \end{bmatrix} \tag{5.13}$$

Observemos que a equação (5.10) representa, para cada j, o produto de $\underline{x}(D)$ pela j-ésima coluna de $\overline{H}(D)$. Assim, podemos escrever a relação entre $\underline{x}(D)$, $\underline{z}(D)$ e $\overline{H}(D)$ como:

$$\underline{z}(D) = \underline{x}(D).\overline{H}(D). \tag{5.14}$$

tomando a transposta em ambos os membros da equação (5.18) obtemos:

$$\underline{z}(D)^T = \overline{H}(D)^T.\underline{x}(D)^T \tag{5.15}$$

Agora $\underline{z}(D)^T$ e $\underline{x}(D)^T$ são vetores-coluna de comprimento n_0 e k_0 respectivamente. As componentes de $\underline{z}(D)^T$ correspondem às sequências codificadas nas n_0 saídas do codificador enquanto que as componentes de $\underline{x}(D)^T$ correspondem às sequências de informação nas k_0 entradas do codificador. Assim, a equação (5.19) representa uma relação entre a entrada e a saída do codificador e portanto $\overline{H}(D)^T$ representa a matriz de transferência do codificador, isto é,

$$\overline{G}(D) = \overline{H}(D)^T = \begin{bmatrix} g_{11}(D) & g_{21}(D) & \cdots & g_{k_0 1}(D) \\ g_{12}(D) & g_{22}(D) & \cdots & g_{k_0 2}(D) \\ \vdots & \vdots & & \vdots \\ g_{1n_0}(D) & g_{2n_0}(D) & & g_{k_0 n_0}(D) \end{bmatrix} \quad (5.16)$$

Observemos que as dimensões de $\overline{G}(D)$ em (5.16) são n_0 x k_0 que são as dimensões obtidas através de (3.22). Esse resultado coincide com a definição dada em [13] para a matriz geradora de um código convolucional. Entretanto em [13] não é mostrado como obter os elementos dessa matriz $\overline{G}(D)$.

Desenvolvendo cada elemento de $\overline{G}(D)$em (5.16) obtemos:

$$\overline{G}(D) = \begin{bmatrix} g_0(1,1)+g_1(1,1)D+\ldots+g_{m-1}(1,1)D^{m-1} & g_0(2,1)+g_1(2,1)D+\ldots+g_{m-1}(2,1)D^{m-1} \ldots \\ g_0(1,2)+g_1(1,2)D+\ldots+g_{m-1}(1,2)D^{m-1} & g_0(2,2)+g_1(2,2)D+\ldots+g_{m-1}(2,2)D^{m-1} \ldots \\ \vdots & \vdots \\ g_0(1,n_0)+g_1(1,n_0)D+\ldots+g_{m-1}(1,n_0)D^{m-1} & g_0(2,n_0)+g_1(2,n_0)D+\ldots+g_{m-1}(2,n_0)D^{m-1} \ldots \end{bmatrix}$$

$$\begin{bmatrix} \ldots\ g_0(k_0,1)+g_1(k_0,1)D+\ldots+g_{m-1}(k_0,1)D^{m-1} \\ \ldots\ g_0(k_0,2)+g_1(k_0,2)D+\ldots+g_{m-1}(k_0,2)D^{m-1} \\ \vdots \\ \ldots\ g_0(k_0,n_0)+g_1(k_0,n_0)D+\ldots+g_{m-1}(k_0,n_0)D^{m-1} \end{bmatrix} \quad (5.17)$$

Podemos escrever (5.17) como:

$$\overline{G}(D) = \begin{bmatrix} g_0(1,1) & g_0(2,1)\ldots g_0(k_0,1) \\ g_0(1,2) & g_0(2,2)\ldots g_0(k_0,2) \\ g_0(1,n_0) & g_0(2,n_0)\ldots g_0(k_0,n_0) \end{bmatrix} + D \begin{bmatrix} g_1(1,1) & g_1(2,1)\ldots g_1(k_0,1) \\ g_1(1,2) & g_1(2,2)\ldots g_1(k_0,2) \\ g_1(1,n_0) & g_1(2,n_0)\ldots g_1(k_0,n_0) \end{bmatrix} + \ldots$$

$$\ldots + D^{m-1} \begin{bmatrix} g_{m-1}(1,1) & g_{m-1}(2,1) & \ldots & g_{m-1}(k_0,1) \\ g_{m-1}(1,2) & g_{m-1}(2,2) & \ldots & g_{m-1}(k_0,1) \\ \vdots & \vdots & & \vdots \\ g_{m-1}(1,n_0) & g_{m-1}(2,n_0) & \ldots & g_{m-1}(k_0,n_0) \end{bmatrix}$$

ou então $\overline{G}(D) = \overline{G}_0^T + D\,\overline{G}_1^T + \ldots + D^{m-1}\,\overline{G}_{m-1}^T$ (5.18)

Portanto, chegamos a uma expressão que permite determinar a matriz de transferência do codificador de um código convolucional diretamente, a partir da matriz geradora do código.

5.5 - CÓDIGOS DE BLOCO LINEARES COMO UM CASO PARTICULAR DOS CÓDIGOS CONVOLUCIONAIS

A diferença fundamental entre os códigos de bloco lineares e os códigos convolucionais é que nos códigos de bloco, o codificador fornece em sua saída palavras-código de comprimento n de tal modo que cada palavra-código codificada num certo instante depende unicamente do bloco de informação de comprimento k que teve acesso ao codificador naquele instante. Nos códigos convolucionais, quando o codificador se encontra no modelo de Mealy , a cada instante um bloco codificado de comprimento n_0 depende do bloco de informação de comprimento k_0 que teve acesso ao codificador naquele instante como também dos m-1 blocos de informação processados anteriormente. Assim os códigos de bloco lineares são códigos convolucionais em que m=1.

Analisando os códigos de bloco lineares como um caso particular dos códigos convolucionais, podemos obter um codificador para códigos de bloco lineares na forma paralela que nos é muito mais útil sob o ponto de vista computacional.O codificador da Fig. 5.3, quando sob o modelo de Mealy, se reduz ao codificador mostrado na Fig. 5.5, que como vemos, não possui elementos de memória, o que implica em ser uma máquina de dimensão zero.

Observemos que a matriz geradora dos códigos convolucionais dada por (5.3), para m=1, se reduz a:

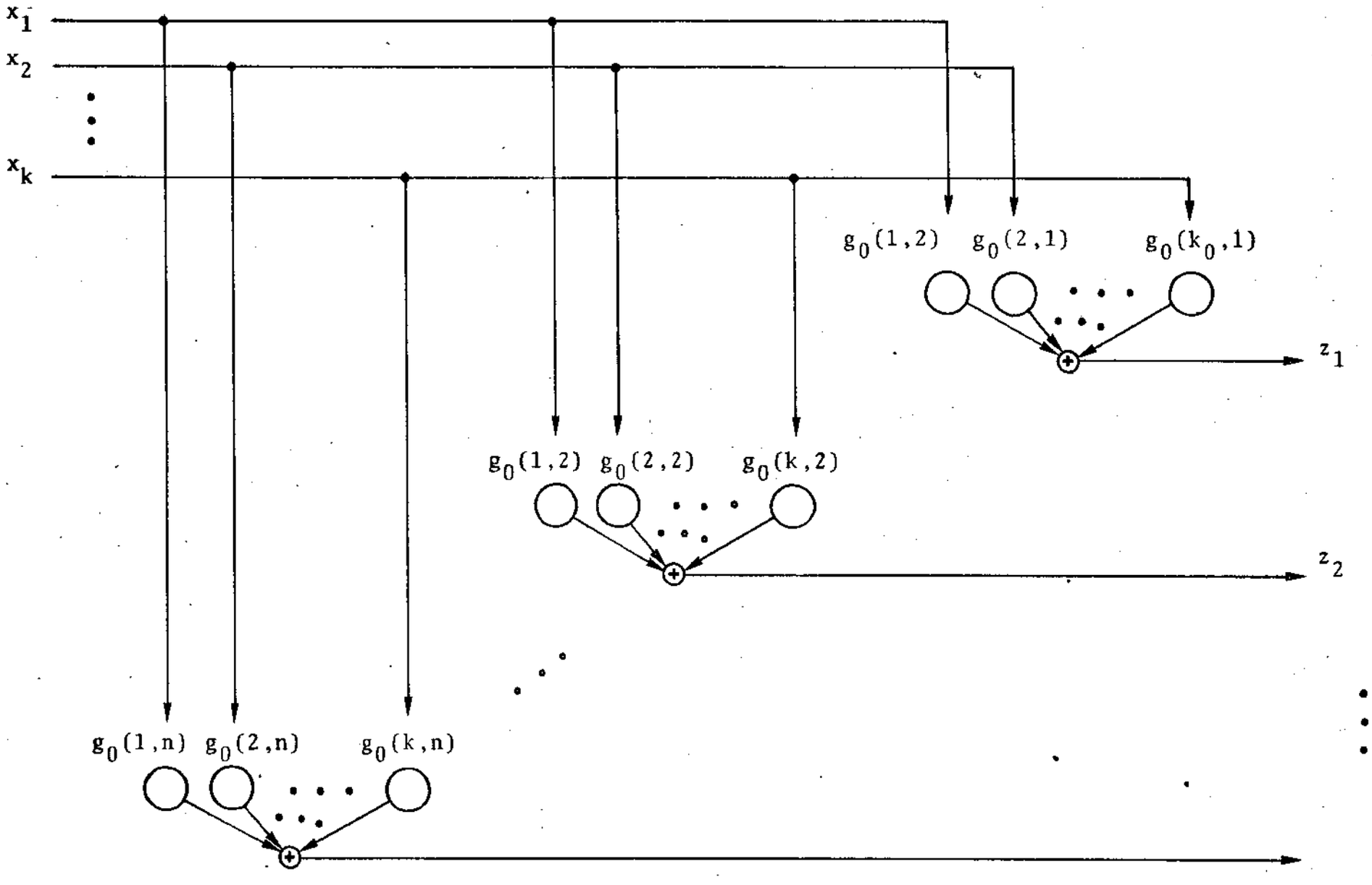


Fig.5.5 - Modelo paralelo para o codificador de um código de bloco linear (n,k), onde $\overline{G}_0 = \overline{G}$

$$\overline{G} = [\overline{G}_0] \tag{5.19}$$

isto é, a sub-matriz G_0 é a própria matriz geradora do código.

A matriz de transferência do codificador para códigos de bloco lineares, na forma paralela, pode ser obtida da equação (5.18) fazendo m=1 resultando em:

$$\overline{G}(D) = \overline{G}_0^T = \overline{G}^T \tag{5.20}$$

A equação (5.20) é o ponto básico para a obtenção das palavras-código de um código de bloco linear (n,k) durante a confecção do programa para determinar a densidade espectral de potência de códigos de bloco lineares.

CAPÍTULO 6

ESPECTRO DE SINAIS DIGITAIS CODIFICADOS POR CÓDIGOS DE BLOCO

6.1 - INTRODUÇÃO

Apresentamos agora os resultados obtidos por Cariolaro e Tronca no trabalho em que nós nos baseamos para escrever o programa para computador que permite determinar o espectro de potência de um código na forma normalizada [2,3]. O trabalho de Cariolaro, como veremos, é desenvolvido apenas apra códigos de bloco e mais precisamente para os códigos apresentados em [7]. Entretanto, as hipóteses apresentadas não impõem quaisquer restrições à extensão dos resultados para códigos lineares. Como veremos, a estrutura matemática dos códigos lineares juntamente com a teoria das máquinas sequenciais lineares nos permitem uma programação bastante elegante sob o ponto de vista da quantidade de dados que devem ser fornecidos ao computador. Apenas a título de comparação com os resultados de Bennett apresentamos as deduções dos espectros de potência de sinais digitais no caso em que a sequência codificada é uma versão modificada, mas de mesmo comprimento da sequência de entrada, isto é, o código utilizado é da forma (1,1) e para o caso em que a sequência codificada tem comprimento diferente da sequência de informação, isto é, quando o código utilizado é da forma (n,k).

6.2 - ESPECIFICAÇÕES DO CODIFICADOR

Sejam $B = \{0,1\}$ um conjunto binário e $A_Q = \{k_1, k_2, \ldots, k_Q\}$ um conjunto Q-ário com $Q \geqslant 2$. Sejam ainda B^k o conjunto de todas as sequências binárias de comprimento k e A_Q^n o conjunto de todas as sequências Q-árias de comprimento n.

DEFINIÇÃO 6.1 - Um processo de codificação é definido como um mapeamento $\phi: B^k \to A_Q^n$ que é necessáriamente inejtivo, isto é, um para um, de modo que seja preservada a informação e consequentemente seja possível a decodificação, o que implica em $Q^n \geqslant 2^k$.

DEFINIÇÃO 6.2 - Uma n-upla Q-ária $\underset{\sim}{\alpha} \in A_Q^n$, tal que seja a imagem de uma k-upla binária $\underset{\sim}{\beta} \in B^k$ através de ϕ é denominada de palavra-código. Isto é, se $\phi(\underset{\sim}{\beta}) = \underset{\sim}{\alpha}$ então $\underset{\sim}{\alpha}$ é uma palavra código.

Sejam O o conjunto de todas as palavras-código e $J = B^k$ o conjunto de todas as k-uplas binárias denominadas a par

tir de agora de VETORES DE INFORMAÇÃO. Em geral, $\mathcal{C} \subset A_Q^n$ já que algumas n-uplas Q-árias não são imagem de nenhum vetor de informação através da transformação ϕ.

DEFINIÇÃO 6.3 - Um codificador de um código (n,k) sempre pode ser representado por uma máquina sequencial síncrona; ou mais precisamente por uma máquina de Mealy, isto é, uma quíntupla m=(I,O,S, g,h) onde os componentes da quíntupla tem o significado dado na definição 3.7.

É comum descrever a função de saída h, como um conjunto de funções $\{h_i\}$, onde h_i é a função de saída associada ao estado s_i, isto é, $h_i(\underline{\beta}) = h(s_i,\underline{\beta})$ para cada $s_i \in S$ e $\underline{\beta} \in I$. Entretanto, para cálculos de densidade espectrais é mais conveniente uma ordenação baseada nos vetores de informação em vez de uma ordenação baseada nos estados. Em se tratando de códigos de bloco, é essencial um procedimento com notação matricial. Assim, como o número de vetores de informação é $K = 2^k$ e forma um espaço vetorial de dimensão k sobre GF(2), podemos arranjá-los em uma matriz $\overline{B}$ cujas linhas são os vetores de informação $\underline{\beta}_u=(\beta_{u1},\beta_{u2},\ldots,\beta_{uk})$, u=1,2,...,K e portanto as dimensões de B são K x k, isto é,

$$\overline{B} = \begin{bmatrix} \underline{\beta}_1 \\ \underline{\beta}_2 \\ \vdots \\ \underline{\beta}_K \end{bmatrix} = \begin{bmatrix} \beta_{12} & \cdots & \beta_{1k} \\ \beta_{22} & \cdots & \beta_{2k} \\ \vdots & & \vdots \\ \beta_{K2} & \cdots & \beta_{Kk} \end{bmatrix} \qquad (6.1)$$

Da mesma forma, as palavras-código são elementos de um espaço vetorial de dimensão n; então, definindo $\underline{\alpha}_{iu}$ como a palavra-código obtida quando o codificador se encontra no estado s_i e o vetor de informação em sua entrada é $\underline{\beta}_u$, isto é, $\underline{\alpha}_{iu}=h(s_i,\underline{\beta}u)$, podemos arranjar todas as palavras código em uma matriz $\overline{A}$ formada a partir de K sub-matrizes $\overline{A}_u$ de dimensões L x n, onde as sub-matrizes $\overline{A}_u$ tem como linhas as palavras-código

$$\underline{\alpha}_{iu} = [\alpha_{iu}^{(1)}, \alpha_{iu}^{(2)},\ldots,\alpha_{iu}^{(n)}] \quad , (i=1,2,\ldots,L \quad e \quad u=1,2,\ldots,K),$$

isto é,

$$\bar{A}_u = \begin{bmatrix} \underset{\sim}{\alpha}_{1u} \\ \underset{\sim}{\alpha}_{2u} \\ \vdots \\ \underset{\sim}{\alpha}_{Lu} \end{bmatrix} = \begin{bmatrix} \alpha_{1u}^{(1)} & \alpha_{1u}^{(2)} & \cdots & \alpha_{1u}^{(n)} \\ \alpha_{2u}^{(1)} & \alpha_{2u}^{(2)} & \cdots & \alpha_{2u}^{(n)} \\ \vdots & \vdots & & \vdots \\ \alpha_{Lu}^{(1)} & \alpha_{Lu}^{(2)} & \cdots & \alpha_{Lu}^{(n)} \end{bmatrix} \quad ; \; u=1,2,\ldots, K. \tag{6.2}$$

Observemos que cada sub-matriz $\bar{A}_u$ tem como linhas todas as palavras-código obtidas quando, para a entrada do codificador fixa no vetor $\underset{\sim}{\beta}_u$, fazemos o codificador passar por todos os L estados. Assim a matriz $\bar{A}$ é da forma

$$\bar{A} = \begin{bmatrix} \bar{A}_1 \\ \cdots \\ \bar{A}_2 \\ \cdots \\ \vdots \\ \cdots \\ \bar{A}_K \end{bmatrix} \tag{6.3}$$

com dimensões K_a x n, onde K_a=K.L. Assim, $\bar{A}$ especifica completamente a função de saída, h, do codificador.

Precisamos ainda de uma representação matricial que caracterize a função transição de estado g. Para isto, definamos a seguinte matriz binária:

$$\bar{E}_u = [e_u(i,j)] \quad ; \quad (i,j=1,2,\ldots,L) \tag{6.4}$$

onde $e_u(i,j)$=1 se, e somente se, $s_j = g(s_i, \underset{\sim}{\beta}_u)$ e em caso contrário $e_u(i,j)$=0. Seja o conjunto de vetores de informação

$$B_{ij} \triangleq \{\underset{\sim}{\beta}_u \mid s_j = g(s_i, \underset{\sim}{\beta}_u)\} \; . \tag{6.5}$$

isto é, o conjunto B_{ij} é o conjunto dos vetores de informação que, ao serem aplicados à entrada do codificador quando no estado s_i, o próximo estado é s_j. Assim, podemos escrever

$$e_u(i,j) = \begin{cases} 1 \; ; \; \text{se} \; \underset{\sim}{\beta}_u \in B_{ij} \\ \\ 0 \; ; \; \text{se} \; \underset{\sim}{\beta}_u \notin B_{ij} \end{cases} \tag{6.6}$$

Ordenando-se as sub-matrizes $\bar{E}_u$ na mesma ordem das sub-matrizes $\bar{A}_u$, obtemos uma matriz $\bar{E}$ que caracteriza a função transição de estado g. Assim, $\bar{E}$ é uma matriz de dimensão K_a x L com a seguinte forma:

$$\bar{E} = \begin{bmatrix} \bar{E}_1 \\ \cdots \\ \bar{E}_2 \\ \cdots \\ \vdots \\ \cdots \\ \bar{E}_K \end{bmatrix} \qquad (6.7)$$

6.3 - O SINAL DIGITAL

Suporemos que o sinal digital é obtido a partir de uma sequência de informação binária, fornecida por um codificador de fonte, que alimenta a entrada do codificador de canal cuja saída alimenta a entrada do modulador digital. Na Fig. 6.1 está uma esquematização do sistema.

Fig. 6.1 -. Geração do sinal digital x(t)

6.3.1 - A MENSAGEM CODIFICADA

O codificador é suposto como uma máquina sequencial síncrona geral descrita na seção anterior. Assim, a sequência de informação é dividida em sequências binárias de comprimento k, os vetores de informação:

$$\underset{\sim}{b}_m = [b_m^{(1)}, b_m^{(2)}, \ldots, b_m^{(k)}] \in B^k \ , \ -\infty < m < \infty \qquad (6.8)$$

que são codificados em palavras Q-árias de comprimento n:

$$\underset{\sim}{a}_m = [a_m^{(1)}, a_m^{(2)}, \ldots, a_m^{(n)}] \in O \subset A_Q^n, \ -\infty < m < \infty \qquad (6.9)$$

Agora as funções do codificador podem ser reescritas mostrando explicitamente a dependência temporal, através de m, como

$$\underset{\sim}{a}_m = h(s_m, \underset{\sim}{b}_m) \tag{6.10a}$$

$$s_{m+1} = g(s_m, \underset{\sim}{b}_m) \tag{6.10b}$$

onde s_m e s_{m+1} são o estado atual e o próximo estado, respectivamente.

6.3.2a - ESTATÍSTICA DA MENSAGEM CODIFICADA

Uma sequência infinita de símbolos, digamos $\{c_r\}(-\infty<r<+\infty)$, é considerada como um processo aleatório que é discreto tanto no tempo como em amplitude desde que r é suposto assumir somente valores inteiros e as amplitudes assumem valores pertencentes a um conjunto de comprimento finito.

O processo $\{c_r\}$ é dito estacionário no sentido fraco se o valor médio,

$$m_c \triangleq E\{c_r\} , \tag{6.11}$$

e a função de autocorrelação,

$$R_c(k) \triangleq E\{c_r\ c_{r+k}\} \ ; \quad -\infty < k < \infty, \tag{6.12}$$

são independentes de r.

Por outro lado, uma sequência infinita de palavras de comprimento n, digamos $\{\underset{\sim}{a}_m\}$ $(-\infty<m<+\infty)$, onde $\underset{\sim}{a}_m$ é da forma da eq. (6.9), é considerado um processo aleatório discreto n - dimensional, isto é, um vetor aleatório cujas componentes $\{a_m^q\}(q=1,2,\ldots,n)$ são processos aleatórios discretos.

O processo $\{\underset{\sim}{a}_m\}$ é dito estacionário no sentido fraco, se o vetor média $\underset{\sim}{m}_a$ e a matriz de auto-correlação $\overline{R}_a(k)$ são independentes de m, onde:

$$\underset{\sim}{m}_a = [m_a^1, m_a^2, \ldots, m_a^n] \triangleq E\{\underset{\sim}{a}_m\} \tag{6.13a}$$

$$\overline{R}_a(k) = ||\, R_a^{pq}(k)\,|| = E\{\underset{\sim}{a}_m^T \cdot \underset{\sim}{a}_{m+k}\} \ , \ p,q=1,2,\ldots,n \tag{6.13b}$$

onde T indica transposição.

A estacionaridade de um processo aleatório vetorial não somente implica na estacionaridade de cada componente do processo como também em sua estacionaridade conjunta[3]. Isto significa que a correlação entre $\{a_m^p\}$ e $\{a_m^q\}$, $R_a^{pq}(k) \triangleq E\{a_m^p . a_{m+k}^q\}$, é independente de m tanto para p=q como para p≠q.

Agora, dado um processo aleatório vetorial $\{\underline{a}_m\}$, suponhamos que os símbolos de cada palavra de comprimento n sejam transmitidos sequencialmente, como em transmissão MAP. Desta forma, um processo escalar $\{y_r\}$ é gerado, onde

$$y_{mn+p} = a_m^p \quad (p=1,2,\ldots,n \; ; \; -\infty < m < +\infty) \tag{6.14}$$

Em geral, este processo é não estacionário, mesmo se o processo original o é. Entretanto, ele exibe um tipo especial de estacionaridade, denominada de estacionaridade periódica que surge a partir das seguintes relações:

$$E\{y_{m.n+p}\} = E\{y_p\} = m_a^p \quad ; \quad 1 \leqslant p \leqslant n \tag{6.15a}$$

$$E\{y_{m.n+p} \; y_{(m+k)n+q}\} = E\{y_p y_{kn+q}\} = R_a^{pq}(k) \; ;$$

$$1 \leqslant p,q \leqslant n \quad e \quad -\infty < k < +\infty \tag{6.15b}$$

Os processos que satisfazem às equações (6.15) também são denominados de CICLO-ESTACIONÁRIOS.

Assim, o valor médio $E\{y_r\}$ não é independente de r mas é uma função periódica de n; a autocorrelação $E\{y_r y_s\}$ depende tanto de r como de s e também exibe uma periodicidade dada por (6.15b). Com certeza, estacionaridade periódica é mais fraca que estacionaridade no sentido fraco; entretanto, o processo $\{y_r\}$ pode ser tratado como estacionário se a origem dos tempos é considerada como uma variável aleatória conveniente.

Em resumo, a sequência de símbolos obtida de uma sequência de palavras de comprimento n, estacionária, é periodicamente estacionária de período n.

6.3.2b - A SEQUÊNCIA DE PALAVRAS-CÓDIGO COMO UM PROCESSO ALEATÓRIO

As considerações da seção 6.3.2a podem ser aplicadas às sequências de vetores de informação e de palavras-código, digamos:

$$\{\underset{\sim}{b}_m\} \quad e \quad \{\underset{\sim}{a}_m\} \; ; \; -\infty < m < +\infty \; , \tag{6.16}$$

onde $\underset{\sim}{b}_m$ e $\underset{\sim}{a}_m$ tem o significado dado pelas equações (6.8) e (6.9) respectivamente.

Em geral, a estatística da sequência de palavras-código depende da estatística dos símbolos de informação e das especificações do código. Suporemos apenas que a sequência de vetores de informação seja estacionária no sentido fraco e que $\underset{\sim}{b}_m$ sejam estatisticamente independentes, enquanto que as funções do codificador g e h são arbitrárias. Também as probabilidades dos vetores de informação são supostas arbitrárias.

6.3.3 - DENSIDADE ESPECTRAL DO SINAL DIGITAL

6.3.3a - SINAL DIGITAL MAP COM UMA MENSAGEM CODIFICADA SÍMBOLO A SÍMBOLO (CÓDIGO UTILIZADO (1,1))

Em um sistema digital MAP cada símbolo da mensagem codificada, digamos $c_r (-\infty<r<+\infty)$, é transmitido como um pulso padrão de forma s(t) a intervalos de duração T, o período de símbolos, de modo que o sinal modulado resultante pode ser expresso por:

$$z(t) = \sum_{r=-\infty}^{\infty} c_r \, s(t-rT-\theta) \; , \tag{6.17}$$

onde θ é a fase da sequência de pulsos.

Na expressão (6.17) é costumeiro supor $\theta = 0$. O sinal resultante não é estacionário mas sim ciclo-estacionário. Então não é possível a aplicação direta do conceito de densidade espectral de potência. Antes, a não estacionaridade tem que ser removida e isto é feito tomando a média e auto-correlação temporais em um período.

Este procedimento de tratar um processo ciclo-estacionário como um processo estacionário é indireto e normalmente difícil e mesmo em alguns casos, como na avaliação da probabilidade de erro da extração do relógio, é importante manter a ciclo-estacionaridade do processo. Um procedimento equivalente de tratar o problema, que é direto e fisicamente palpável, é supor que a fase θ é uma variável aleatória uniformemente distribuída num período de símbolo, T. A interpretação física dessa suposição é que um

observador do processo z(t), apesar de saber que o mesmo é constituído por pulsos emitidos a intervalos de tempo iguais, não sabe nada a respeito do referencial temporal para os instantes de emissão.

Com a origem dos tempos assim imprevisível, o processo z(t) se torna estacionário no sentido fraco, e assim considerações de espectro de potência podem ser aplicadas. Mais precisamente, supondo $\{c_r\}$ estacionária no sentido fraco, e que θ é uniformemente distribuído entre $0 \leqslant \theta < T$ pode-se mostrar que

$$m_z \triangleq E\{z(t)\} = 1/T \; m_c A_s \;, \tag{6.18a}$$

$$R_z(\tau) \triangleq E\{z(t).z(t+\tau)\} = 1/T \sum_{k=-\infty}^{+\infty} R_c(k) \; C_s(\tau - kT) \;, \tag{6.18b}$$

onde m_c e $R_c(k)$ são o valor médio e a auto-correlação (estatística) de $\{c_r\}$ respectivamente; A_s e $C_s(\tau)$ são a área e a auto-correlação (temporal) do pulso s(t), isto é,

$$A_s \triangleq \int_{-\infty}^{+\infty} s(t)dt \;, \tag{6.19a}$$

$$C_s(\tau) \triangleq \int_{-\infty}^{+\infty} s(t)\; s(t+\tau)\; dt \tag{6.18b}$$

Mostremos a veracidade das equações (6.18) sujeitas às suposições de $\{c_r\}$ ser estacionária no sentido fraco e θ uniformemente distribuída entre $0 \leqslant \theta < T$.

$$m_z \triangleq E\{z(t)\} = E\{ \sum_{r=-\infty}^{+\infty} c_r \; s(t-rT-\theta)\} =$$

$$= \sum_{r=-\infty}^{+\infty} E\{c_r \; s(t-rT-\theta)\},$$

como os dígitos codificados são independentes da forma de pulso utilizada na modulação, temos:

$$m_z = \sum_{r=-\infty}^{+\infty} E\{c_r\}.E\{s(t-rT-\theta)\} \;;$$

mas por (6.11), temos:

$$m_z = \sum_{r=-\infty}^{+\infty} m_c \, E\{s(t-rT-\theta)\} = m_c \sum_{r=-\infty}^{+\infty} E\{s(t-rT-\theta)\}$$

$$m_z = m_c \sum_{r=-\infty}^{+\infty} \int_{-\infty}^{\infty} s(t-rT-\theta) \; p_\theta(\theta) \; d\theta$$

onde $p_\theta(\theta)$ é a função densidade de probabilidade da variável aleatória θ. Como θ é uniformemente distribuída entre 0 e T por hipótese,

$$m_z = m_c \sum_{r=-\infty}^{+\infty} 1/T \int_0^T s(t-rT-\theta) \; d\theta$$

Fazendo-se a mudança de variável $y=t-rT-\theta$

$$m_z = 1/T . m_c \sum_{r=-\infty}^{+\infty} - \int_{t-rT}^{t-rT+T} s(y) \; dy =$$

$$= 1/T . m_c \left[\sum_{r=-\infty}^{+\infty} \int_{t-rT+T}^{t-rT} s(y) \; dy \right]$$

Observemos que a integral entre colchetes representa a área abaixo de $s(y)$ em um período T para r fixo. Como r varia no intervalo $(-\infty,+\infty)$ temos:

$$m_z = 1/T . m_c \int_{-\infty}^{\infty} s(y) \; dy$$

Utilizando a equação (6.19a) obtemos finalmente (6.18a).

$$m_z = 1/T \; m_c A_s \qquad (6.20)$$

A auto-correlação para $z(t)$ é dada por

$$R_z(\tau) = E\{z(t) \; z(t+\tau)\} = E\{ \sum_{r=-\infty}^{+\infty} c_r \; s(t-rT-\theta) . \sum_{s=-\infty}^{+\infty} c_s \; s(t+\tau-sT-\theta)\} =$$

$$= E\{ \sum_{r,s=-\infty}^{+\infty} c_r \; c_s \; s(t-rT-\theta) \; s(t+\tau-sT-\theta)\} =$$

$$= \sum_{r,s=-\infty}^{+\infty} E\{c_r \; c_s \; s(t-rT-\theta) \; s(t+\tau-sT-\theta)\} =$$

$$= \sum_{r,s=-\infty}^{+\infty} E\{c_r\ c_s\}\ E\{s(t-rT-\theta)\ s(t+\tau-sT-\theta)\}$$

Como por hipótese $\{c_r\}$ é estacionária no sentido fraco, utilizando (6.12), $E\{c_r\ c_s\} = R_c(s-r)$. Assim,

$$R_z(\tau) = \sum_{r,s=-\infty}^{+\infty} R_c(s-r)\ E\{s(t-rT-\theta)\ s(t+\tau-sT-\theta)\}$$

$$R_z(\tau) = \sum_{r,s=-\infty}^{+\infty} R_c(s-r) \int_{-\infty}^{\infty} s(t-rT-\theta)s(t+\tau-sT-\theta)p_\theta(\theta)\ d\theta =$$

$$= \sum_{r,s=-\infty}^{+\infty} R_c(s-r)1/T \int_0^T s(t-rT-\theta)s(t+\tau-sT-\theta)d\theta$$

Fazendo-se a mudança de variável

$$y_1 = t - rT - \theta \quad \text{e} \quad y_2 = t + \tau - sT - \theta$$

e integrando-se com relação a y_1, temos:

$$R_z(\tau) = 1/T \sum_{r,s=-\infty}^{+\infty} R_c(s-r) \int_{t-rT-\theta}^{t-rT} s(y_1)\ s(y_2)\ dy_1 =$$

$$= 1/T \sum_{s=-\infty}^{+\infty} R_c(s-r)\left[\int_{-\infty}^{\infty} s(y_1)\ s(y_2)\ dy_1\right]$$

Observemos que a integral entre colchetes representa a auto-correlação $C_s(y_2-y_1)$ (pela definição de auto-correlação). Assim,

$$R_z(\tau) = 1/T \sum_{s=-\infty}^{+\infty} R_c(s-r)\ C_s(y_2-y_1) =$$

$$= 1/T \sum_{s=-\infty}^{+\infty} R_c(s-r)\ C_s[\tau-(s-r)T] =$$

$$= 1/T \sum_{k=-\infty}^{+\infty} R_c(k).C_s(\tau-kT) \qquad (6.21)$$

onde $k=s-r$ e $C_s(\tau)$ é como definido por (6.19b).

A densidade espectral de potência do processo $z(t)$ é simplesmente a transformada de Fourier de sua auto-correlação, isto é,

$$W_z(f) = F[R_z(\tau)] = F[1/T \sum_{k=-\infty}^{+\infty} R_c(k)\ C_s(\tau-kT)] =$$

$$= 1/T \sum_{k=-\infty}^{+\infty} F[R_c(k)\ C_s(\tau-kT)] =$$

$$= 1/T \sum_{k=-\infty}^{+\infty} F[R_c(k)] * F[C_s(\tau-kT)] =$$

$$= 1/T \sum_{k=-\infty}^{+\infty} R_c(k)\delta(f) * F[C_s(\tau)]\ e^{-j2\pi fkT} =$$

$$= 1/T \sum_{k=-\infty}^{+\infty} R_c(k).F[C_s(\tau)]\ e^{-j2\pi fkT} =$$

$$= 1/T \sum_{k=-\infty}^{+\infty} R_c(k)\ e^{-j2\pi fkT}.F[C_s(\tau)]$$

É fácil de ver [10] que $F[C_s(\tau)] = |S(f)|^2$. Assim,

$$W_z(f) = 1/T\ |S(f)|^2.W_c(f)\ , \qquad (6.22)$$

onde

$$W_c(f) = \sum_{k=-\infty}^{+\infty} R_c(k)\ e^{-j2\pi fkT} \qquad (6.23)$$

$$S(f) = F[s(t)]$$

e $|S(f)|^2$ representa a densidade espectral de energia de $s(t)$. $W_c(f)$ é definida como a transformada de Fourier discreta da auto-correlação discreta $R_c(k)$ e assim ele representa a densidade espectral do processo aleatório discreto $\{c_r\}$. $W_c(f)$ também pode ser considerado como a densidade espectral do processo $z(t)$, normalizada com relação à envoltória $1/T|S(f)|^2$.

Podemos ainda rearranjar (6.23) de modo a obtermos

$$W_c(f) = \sum_{k=-\infty}^{+\infty} [R_c(k)-R_c(\infty)]\ e^{-j2\pi fkT} + \sum_{k=-\infty}^{+\infty} R_c(\infty)\ e^{-j2\pi fkT} =$$

$$= \sum_{k=-\infty}^{+\infty} \varepsilon_k [R_c(k)-R_c(\infty)] \cos 2\pi fkT + R_c(\infty) \sum_{k=-\infty}^{+\infty} e^{-j2\pi fkT}$$

onde utilizamos a propriedade de paridade de auto-correlação e a mudança da transformada de Fourier exponencial para a forma trigonométrica com $\varepsilon_0=1$, $\varepsilon_k=2$ para $k>0$ e $R_c(\infty)=m_c^2$; fato que caracteriza a independência dos símbolos quando estão bastante afastados. Podemos fazer uma modificação no segundo termo da expressão de modo a obtermos:

$$W_c(f) = \sum_{k=0}^{+\infty} [\varepsilon_k\ R_c(k)-R_c(\infty)]\ \cos 2\pi fkT + R_c(\infty).1/T \sum_{k=-\infty}^{+\infty} \delta(f-k/T) \qquad (6.24)$$

A expressão (6.24) é semelhante à forma como Bennett apresenta a densidade espectral para o sinal digital codificado símbolo a símbolo.

6.3.3b - SINAL DIGITAL MAP COM CODIFICAÇÃO DA MENSAGEM POR UM CÓDIGO (n,k)

Quando a mensagem é composta de palavras-código de comprimento n, $\{\underline{a}_m\}$, o sinal digital MAP, pode ser escrito como:

$$x(t) = \sum_{m=-\infty}^{+\infty} \sum_{p=1}^{n} a_m^p\ s[t-(p-1)T - mT_n - \theta_n] \ , \qquad (6.25)$$

onde: a_m^p é o p-ésimo símbolo da m-ésima palavra-código

T é o período dos símbolos

$T_n \triangleq nT$ é o período das palavras-código

θ_n é a fase

Considerando o vetor palavra-código $\underline{a}_m=(a_m^1, a_m^2, \ldots, a_m^n)$ e o vetor-coluna de pulsos $\underline{s}(t)$ onde

$$\underline{s}(t) = \begin{bmatrix} s(t) \\ s(t-T) \\ \vdots \\ s(t-(n-1)T) \end{bmatrix} , \qquad (6.26)$$

a expressão (6.25) pode ser escrita mais compactamente como

$$x(t) = \sum_{m=-\infty}^{+\infty} \underline{a}_m \, \underline{s}(t-mT_n - \theta_n) . \qquad (6.27)$$

Supondo que a sequência codificada $\{\underline{a}_m\}$ seja estacionária no sentido fraco e que a fase θ_n seja uma variável aleatória uniformemente distribuída sobre um período de palavra-código, $0 \leqslant \theta_n < T_n$, o peocesso x(t) em (6.27) é estacionário no sentido fraco, com média e auto-correlação dadas por (6.28a) e (6.28b) respectivamente.

$$m_x = 1/T_n \, A_s \sum_{p=1}^{n} m_a^p , \qquad (6.28a)$$

$$R_x(\tau) = 1/T_n \sum_{k=-\infty}^{+\infty} \sum_{p,q=1}^{n} R_a^{pq}(k) \, C_s[\tau - kT_n - (q-p)T], \qquad (6.28b)$$

onde m_a^p é o p-ésimo componente do vetor média $\underline{m}_a$, $R_a^{pq}(k)$ é o elemento (p,q) da matriz de auto-correlação $\overline{R}_a(k)$ enquanto A_s e $C_s(\tau)$ são os parâmetros do pulso dados por (6.19a) e (6.19b) respectivamente.

Mostremos a veracidade das equações (6.28).

$$m_x \triangleq E\{x(t)\} = E\{ \sum_{m=-\infty}^{+\infty} \underline{a}_m \, \underline{s}[t-m.T_n - \theta_n]\} =$$

$$= \sum_{m=-\infty}^{+\infty} E\{\underline{a}_m \, \underline{s}[t-mT_n-\theta_n]\} = \sum_{m=-\infty}^{+\infty} E\{\underline{a}_m\} \, E\{\underline{s}[t-mT_n-\theta_n]\} =$$

$$= \sum_{m=-\infty}^{+\infty} E\{\underline{a}_m\} \int_{-\infty}^{\infty} \underline{s}[t-mT_n-\theta_n] \, p_{\theta_n}(\theta_n) \, d\theta ,$$

onde $P_{\theta_n}(\theta_n)$ é a função densidade de probabilidade da variável aleatória θ_n. Utilizando a definição dada por (6.13a) temos:

$$m_x = \underset{\sim}{m}_a \sum_{m=-\infty}^{+\infty} 1/T_n \int_0^{T_n} \underset{\sim}{s}(t-mT_n-\theta_n)\, d\theta_n =$$

$$= 1/T_n \; \underset{\sim}{m}_a \sum_{m=-\infty}^{+\infty} \int_0^{T_n} \underset{\sim}{s}(t-mT_n-\theta_n)\, d\theta_n .$$

Fazendo a mudança de variável $y=t-mT_n-\theta_n$, temos:

$$m_x = 1/T_n \; \underset{\sim}{m}_a \sum_{m=-\infty}^{+\infty} - \int_{t-mT_n}^{t-mT_n-T_n} \underset{\sim}{s}(y)\, dy =$$

$$= 1/T_n \cdot \underset{\sim}{m}_a \sum_{m=-\infty}^{+\infty} \int_{t-mT_n-T_n}^{t-mT_n} \underset{\sim}{s}(y)\, dy =$$

$$= 1/T_n \; \underset{\sim}{m}_a \int_{-\infty}^{\infty} \underset{\sim}{s}(y)\, dy = 1/T_n \; \underset{\sim}{m}_a \; A_s ,$$

Finalmente,

$$m_x = 1/T_n \; A_s \sum_{p=1}^{n} m_a^p$$

Para a auto-correlação temos:

$$R_x(\tau) \triangleq E\{x(t)\, x(t+\tau)\}.$$

Utilizando a expressão escalar para $x(t)$, (eq. 6.25), por conveniência, temos:

$$R_x(\tau) = E\{ \sum_{m=-\infty}^{+\infty} \sum_{p=1}^{n} a_m^p \; s[t-(p-1)T - mT_n - \theta_n] \cdot$$

$$\cdot \sum_{\ell=-\infty}^{+\infty} \sum_{q=1}^{n} a_\ell^q \; s[t+\tau-(q-1)T - \ell T_n - \theta_n]\} =$$

$$= E\{ \sum_{m,\ell=-\infty}^{+\infty} \sum_{p,q=1}^{n} a_m^p \, a_\ell^q \; s[t-(p-1)T - mT_n - \theta_n] s[t+\tau-(q-1)T - \ell T_n - \theta_n]\}$$

$$= \sum_{m,\ell=-\infty}^{+\infty} \sum_{p,q=1}^{n} E\{a_m^p\, a_\ell^q\, s[t-(p-1)T-mT_n-\theta_n]s[t+\tau-(q-1)T-\ell T_n-\theta_n]\} =$$

$$= \sum_{m,\ell=-\infty}^{+\infty} \sum_{p,q=1}^{n} E\{a_m^p\, a_\ell^q\}E\{s[t-(p-1)T-mT_n-\theta_n]s[t+\tau-(q-1)T-\ell T_n-\theta_n]\}\ ;$$

como a sequência $\{a_m\}$ é suposta estacionária no sentido fraco, aplicando-se a definição (6.13b) temos:

$$R_x(\tau) = \sum_{m\ell=-\infty}^{+\infty} \sum_{p,q=1}^{n} R_a^{pq}(\ell-m)\, E\{s[t-(p-1)T-mT_n-\theta_n]s[t+\tau-(q-1)T-\ell T_n-\theta_n]\}$$

$$= \sum_{m,\ell=-\infty}^{+\infty} \sum_{p,q=1}^{n} R_a^{pq}(\ell-m) \int_{-\infty}^{+\infty} s[t-(p-1)T-mT_n-\theta_n]s[t+\tau-(q-1)T-\ell T_n-\theta_n]\, p_{\theta_n}(\theta_n)d\theta_n =$$

$$= 1/T_n \sum_{m,\ell=-\infty}^{+\infty} \sum_{p,q=1}^{n} R_a^{pq}(\ell-m) \int_0^{T_n} s[t-(p-1)T-mT_n-\theta_n]s[t+\tau-(q-1)T-\ell T_n-\theta_n]d\theta_n .$$

Fazendo-se a mudança de variáveis: $y_1 = t - (p-1)T - mT_n - \theta_n$, $y_2 = t + \tau - (q-1)T - \ell T_n - \theta_n$ e integrando-se com relação a y_1, tem-se:

$$R_x(\tau) = 1/T_n \sum_{m,\ell=-\infty}^{+\infty} \sum_{p,q=1}^{n} R_a^{pq}(\ell-m) \int_{t-(q-1)T-mT_n-T_n}^{t-(p-1)T-mT_n} s(y_1)s(y_2)dy_1 =$$

$$= 1/T_n \sum_{\ell=-\infty}^{+\infty} \sum_{p,q=1}^{n} R_a^{pq}(\ell-m) \int_{-\infty}^{+\infty} s(y_1)s(y_2)\, dy_1 =$$

$$= 1/T_n \sum_{\ell=-\infty}^{+\infty} \sum_{p,q=1}^{n} R_a^{pq}(\ell-m)\, C_s(y_2-y_1) =$$

$$= 1/T_n \sum_{\ell=-\infty}^{+\infty} \sum_{p,q=1}^{n} R_a^{pq}(\ell-m)\, C_s[\tau - (\ell-m)T_n - (q-p)T] =$$

$$= 1/T_n \sum_{k=-\infty}^{+\infty} \sum_{p,q=1}^{n} R_a^{pq}(k) C_s[\tau - kT_n - (q-p)T]. \qquad (6.29)$$

onde $k = \ell - m$.

A densidade espectral do processo $x(t)$ é simplesmente a transformada de Fourier de (6.29), isto é,

$$W_x(f) = 1/T_n \sum_{k=-\infty}^{+\infty} \sum_{p,q=1}^{n} R_a^{pq}(k)\,\delta(f) * F[C_s(\tau)]\, e^{-j2\pi f[kT_n-(q-p)T]} =$$

$$= 1/T_n \sum_{k=-\infty}^{+\infty} \sum_{p,q=1}^{n} R_a^{pq}(k)\; e^{-j2\pi f[kT_n-(q-p)T]} \cdot |S(f)|^2 =$$

$$= |S(f)|^2/T_n \sum_{k=-\infty}^{+\infty} \cdot \sum_{p=1}^{n} \sum_{q=1}^{n} R_a^{pq}(k)\; e^{-j2\pi f(q-p)T} \cdot e^{-j2\pi fkT_n} =$$

$$= |S(f)|^2/T_n \sum_{k=-\infty}^{+\infty} [e^{j2\pi fT}, e^{j4\pi fT} \cdots, e^{j2\pi fnT}]\, \bar{R}_a(k) \begin{bmatrix} e^{-j2\pi fT} \\ e^{-j4\pi fT} \\ \vdots \\ e^{-j2\pi fnT} \end{bmatrix} e^{-j2\pi fkT_n} =$$

$$= |S(f)|^2/T_n \sum_{k=-\infty}^{+\infty} \bar{V}.\bar{R}_a(k)\; \bar{V}^*\, e^{-j2\pi fkT_n} =$$

$$= |S(f)|^2/T_n\; \bar{V} \left(\sum_{k=-\infty}^{+\infty} \bar{R}_a(k)\, e^{-j2\pi fkT_n}\right) \bar{V}^* =$$

$$= |S(f)|^2/T_n\; \bar{V}\, \bar{G}_a(f)\, \bar{V}^*. \qquad (6.30)$$

onde $\bar{V}$ é o vetor $[e^{j2\pi fT}, e^{-j4\pi fT}, \ldots, e^{j2\pi fnT}]$, $\bar{V}^*$ é seu transposto conjugado e $\bar{G}_a(f)$ é a matriz densidade espectral do processo $\{\underline{a}_m\}$, n-dimensional, que é definida por:

$$\bar{G}_a(f) \triangleq \sum_{k=-\infty}^{+\infty} \bar{R}_a(k)\, e^{-j2\pi fkT_n} \qquad (6.31)$$

É possível também a obtenção de uma expressão semelhante à de Bennett nesse caso e para isto basta adicionar e subtrair o limite da matriz de auto-correlação para $k \to \infty$, isto é,

$$\bar{G}_a(f) = \sum_{k=-\infty}^{+\infty} [\bar{R}_a(k)-\bar{R}_a(\infty)]\, e^{-j2\pi fkT_n} + \bar{R}_a(\infty) \sum_{k=-\infty}^{\infty} e^{-j2\pi fkT_n} =$$

$$= \sum_{k=-\infty}^{+\infty} [\bar{R}_a(k) - \bar{R}_a(\infty)] e^{-j2\pi fkT_n} + 1/T_n\, \bar{R}_a(\infty) \sum_{k=-\infty}^{+\infty} \delta(f-k/T_n). \tag{6.32a}$$

Utilizando o fato de $\bar{R}_a(-k) = \bar{R}_a(k)$ obtemos ainda:

$$\bar{G}_a(f) = \sum_{k=0}^{+\infty} \varepsilon_k [\bar{R}_a(k) - \bar{R}_a(\infty)] \cos 2\pi fkT_n + \bar{R}_a(\infty)/T_n \sum_{k=-\infty}^{+\infty} \delta(f-k/T_n) \tag{6.32b}$$

onde $\varepsilon_0=1$ e $\varepsilon_k=2$ para $k > 0$.

Substituindo (6.32a) em (6.30) obtemos:

$$W_x(f) = |S(f)|^2/T_n\, \bar{V} \left\{ \sum_{k=-\infty}^{+\infty} [\bar{R}_a(k)-\bar{R}_a(\infty)]\, e^{-j2\pi fkT_n} + \bar{R}_a(\infty)/T_n \sum_{k=-\infty}^{+\infty} \delta(f-k/T_n) \right\} \bar{V}^* =$$

$$= |S(f)|^2/T_n \left\{ \bar{V} \sum_{k=-\infty}^{+\infty} [\bar{R}_a(k)-\bar{R}_a(\infty)] e^{-j2\pi fkT_n} \bar{V}^* + \bar{V}\, \bar{R}_a(\infty)/T_n \sum_{k=-\infty}^{+\infty} (f-k/T_n) \bar{V}^* \right\} =$$

$$= |S(f)|^2/T_n [X_c(f) + X_d(f)/T_n] = |S(f)|^2/T_n\, X(f). \tag{6.33}$$

onde
$$X_c(f) = \bar{V} \sum_{k=-\infty}^{+\infty} [\bar{R}_a(k)-\bar{R}_a(\infty)]\, e^{-j2\pi fkT_n} \cdot \bar{V}^* \tag{6.34}$$

$$X_d(f) = \bar{V} \cdot \bar{R}_a(\infty) \cdot \bar{V}^* \sum_{k=-\infty}^{+\infty} \delta(f-k/T_n). \tag{6.35}$$

$$X(f) = X_c(f) + X_d(f)/T_n. \tag{6.36}$$

$X_c(f)$ é denominado de componente contínuo do espectro de potência do código enquanto que $X_d(f)$ é denominado de componente discreto.

Observando a expressão (6.33), verificamos que o fator X(f) é determinado única e exclusivamente pela codificação , enquanto que o outro fator é determinado somente pelo modulador digital (veja Fig. 6.1). Assim, $X(f).1/T_n$ é interpretado como sendo o espectro de potência do sinal digital x(t) quando o modulador digital produz pulsos ideais $s(t)=\delta(t)$. Observemos que X(f) é uma função periódica de f com um período igual à taxa de símbolos B=1/T e portanto basta ser examinada no intervalo $0 \leqslant f \leqslant B$.

O componente contínuo $X_c(f)$ é composto de uma combinação de exponenciais da forma

$$e^{-j2\pi f[kT_n+(q-p)T]}$$

de modo que (6.34) pode ser interpretado como uma maneira diferente de expressar a expansão em série de Fourier de $X_c(f)$.

É de interesse o comportamento de $X_c(f)$ nas proximidades de f=0 e isto pode ser feito fazendo-se f=0 em (6.34) , isto é:

$$X_c(0) = \overline{V}(0) \sum_{k=-\infty}^{+\infty} [\overline{R}_a(k) - \overline{R}_a(\infty)]\, \overline{V}^*(0) =$$

$$= \sum_{k=-\infty}^{+\infty} \sum_{p,q=1}^{n} [R_a^{pq}(k) - R_a^{pq}(\infty)] \qquad (6.37)$$

Normalmente se deseja que $X_c(0)=0$. Uma condição suficiente para que isto ocorra é que a série de matrizes de covariância $\overline{R}_a(k)-\overline{R}_a(\infty)$ convirja para zero, isto é,

$$X_c(f) = 0$$

se $$\sum_{k=-\infty}^{+\infty} [\overline{R}_a(k) - \overline{R}_a(\infty)] = 0. \qquad (6.38)$$

O componente discreto $X_d(f)$ apresenta n linhas espectrais no intervalo $0 \leqslant f < B$, nas frequências

$$f = 0, B_n, \ldots, (n-1)B_n$$

onde $$B_n = B/n.$$

A presença desses componentes discretos não é de todo indesejável já que podem ser usados para a extração do relógio a

partir do sinal x(t).

É fácil de ver, a partir de (6.32b) e (6.30) que $X_c(f)$ e $X_d(f)$ podem também ser escritos como:

$$X_c(f) = \bar{V}\,\bar{\Gamma}_1\,\bar{V}^* + 2\,R_e[\bar{V}\,\bar{\Gamma}_2\,\bar{V}^*] \tag{6.39a}$$

$$X_d(f) = \bar{V}\,\bar{R}(\infty)\,\bar{V}^* \tag{6.39b}$$

onde $\bar{\Gamma}_1 \triangleq \bar{R}(0) - \bar{R}(\infty)$

e $$\bar{\Gamma}_2 \triangleq \sum_{k=1}^{\infty} [\bar{R}_a(k) - \bar{R}_u(\infty)]\, e^{-j2\pi f k T_n} \tag{6.40}$$

6.4 - PROBABILIDADES E CORRELAÇÕES DAS PALAVRAS-CÓDIGO

Seja o vetor de probabilidades das palavras-código definido por:

$$\underline{p} = (p_1, p_2, \ldots, p_{K_a}) \tag{6.41}$$

onde $p_r \triangleq \text{Prob}(\underline{a}_m = \underline{\alpha}_r)$, isto é, p_r é a probabilidade de que no instante m $\underline{\alpha}_r$ seja transmitida. Seja ainda a matriz de probabilidades juntas das palavras-código dada por:

$$\bar{P}_a(k) = \| P_a^{rs}(k) \| \quad , (r,s=1,2,\ldots,K_a), \tag{6.42}$$

onde $$P_a^{rs}(k) = \text{Prob}(\underline{a}_m = \underline{\alpha}_r \cap \underline{a}_{m+k} = \underline{\alpha}_s) \ , \tag{6.43}$$

isto é, $P_a^{rs}(k)$ é a probabilidade de que no instante m $\underline{\alpha}_r$ seja transmitida e que no instante m+k $\underline{\alpha}_s$ seja transmitida.

Considerando a matriz de palavras-código definida por (6.2) e observando a sua constituição, pode-se mostrar [3] que:

$$\underline{m}_a = \underline{p}\,\bar{A} \tag{6.43a}$$

$$\bar{R}_a(k) = \bar{A}^T\,\bar{P}_a(k)\,\bar{A} \ . \tag{6.43b}$$

Quando k tende para o infinito, quisquer duas palavras-código $\underline{a}_m$ e $\underline{a}_{m+k}$ são completamente independentes e assim temos:

$$\bar{P}_a(\infty) = \underline{p}^T\,\underline{p} \tag{6.44a}$$

$$\bar{R}_a(\infty) = \underline{m}_a^T \cdot \underline{m}_a \qquad (6.44b)$$

A expressão (6.43) representa o elemento genérico da matriz $\bar{P}_a(k)$ para cada k. Introduzindo os índices i,j posicionadores dos elementos na matriz, (6.43) se transforma em

$$P_{ak}^{uv}(i,j) = \text{Prob}\,(\underline{a}_m = \underline{\alpha}_{iu} \cap \underline{a}_{m+k} = \underline{\alpha}_{jv}) \qquad (6.45)$$

Como a matriz $\bar{A}$ dada por (6.2) está dividida em $K = 2^k$ sub-matrizes, a matriz $\bar{P}_a(k)$ dada por (6.42) pode também ser dividida em K^2 submatrizes $\bar{P}_{ak}^{uv}$ com elementos dados por (6.45) e (u,v=1,2,...,K). Utilizando essas partições de matrizes, (6.43b) pode ser escrita como:

$$R_a(k) = \left[\, A_1^T \;\middle|\; A_2^T \;\middle|\; \ldots\ldots \;\middle|\; A_K^T \,\right] \begin{bmatrix} \bar{P}_{ak}^{11} & \bar{P}_{ak}^{12} & \ldots\ldots & \bar{P}_{ak}^{1K} \\ \bar{P}_{ak}^{21} & \bar{P}_{ak}^{22} & \ldots\ldots & \bar{P}_{ak}^{2K} \\ \vdots & \vdots & & \vdots \\ \bar{P}_{ak}^{K1} & \bar{P}_{ak}^{K2} & \ldots\ldots & \bar{P}_{ak}^{KK} \end{bmatrix} \begin{bmatrix} \bar{A}_1 \\ \bar{A}_2 \\ \vdots \\ \bar{A}_K \end{bmatrix} \qquad (6.46)$$

Para facilitar a notação, abandonamos o índice a e usaremos $\bar{R}_a(k) \triangleq \bar{R}_k$, $\bar{P}_a^{uv}(k) \triangleq P_k^{uv}$. Assim, (6.46) pode ser escrita como:

$$\bar{R}_k = \left[\sum_{u=1}^{K} \bar{A}_u^T \bar{P}_k^{u1} \;,\; \sum_{u=1}^{K} \bar{A}_u^T \bar{P}_k^{u2} \;,\; \ldots\ldots,\; \sum_{u=1}^{K} \bar{A}_u^T \bar{P}_k^{uK} \right] \begin{bmatrix} \bar{A}_1 \\ \bar{A}_2 \\ \vdots \\ \bar{A}_K \end{bmatrix} =$$

$$= \sum_{u=1}^{K} \bar{A}_u^T \bar{P}_k^{u1} \bar{A}_1 + \sum_{u=1}^{K} \bar{A}_u^T \bar{P}_k^{u2} \bar{A}_2 + \ldots\ldots + \sum_{u=1}^{K} \bar{A}_u \bar{P}_k^{uK} \bar{A}_K =$$

$$= \sum_{u=1}^{K} \sum_{v=1}^{K} \bar{A}_u^T \bar{P}_k^{uv} \bar{A}_v \qquad (6.47)$$

Pelo que foi visto até aqui, observa-se que as matrizes de auto-correlação dependem das probabilidades das palavras-código, que teremos que calcular, e mostraremos que dependem apenas das probabilidades dos vetores de informação e das especificações do codificador.

6.4.1 - PROBABILIDADES DOS ESTADOS DO CODIFICADOR

As suposições feitas para o cálculo das probabilidades dos estados são que:

1) Os vetores de informação $\underset{\sim}{b}_m$ sejam independentes;

2) O processo $\{\underset{\sim}{b}_n\}$ seja estacionário no sentido fraco.

Considerando a função transição de estado do codificador dada por (6.10b) e considerando as suposições (1) e (2) é fácil de mostrar [3] que a sequência de estados $\{S_m\}$ representa uma cadeia de Markov homogênea. Assim, $\{S_m\}$ pode ser caracterizada exaustivamente pela matriz transição de estado

$$\overline{\Pi} = \| \Pi(j/i) \| , \quad (i,j=1,2,\ldots,L) \tag{6.48}$$

onde L é o número de estados do codificador e $\Pi(j/i)$ é a probabilidade de, no instante m+1, encontrar o codificador no estado s_j, dado que no instante m ele se encontra no estado s_i, isto é,

$$\Pi(j/i) \triangleq \text{Prob}\,[S_{m+1} = s_j \mid S_m = s_i] \tag{6.49}$$

Essa probabilidade condicional é obtida diretamente como sendo a probabilidade do evento $\{\underset{\sim}{b}_m \in \mathcal{B}_{ij}\}$ onde $\mathcal{B}$ é o conjunto dado por (6.5). Assim, temos:

$$\Pi(j/i) = \text{Prob}\,[\underset{\sim}{b}_m \in \mathcal{B}_{ij}] = \sum_{\underset{\sim}{\beta}_u \in \mathcal{B}_{ij}} q_u \tag{6.50}$$

onde $q_u = \text{Prob}[\underset{\sim}{b}_n = \underset{\sim}{\beta}_u]$, (u=1,...,K), são as probabilidades dos vetores de informação e o somatório é sobre todos u's tal que

$$\underset{\sim}{\beta}_u \in \mathcal{B}_{ij}$$

Utilizando as matrizes $\overline{E}_u$ definidas por (6.6), obtém-se a seguinte relação para a matriz transição de estado [2].

$$\overline{\Pi} = \sum_{u=1}^{K} q_u \overline{E}_u \tag{6.51}$$

Com a matriz transição de estado calculada, as outras probabilidades de interesse podem ser obtidas da seguinte maneira [11] :

a) O vetor coluna dado por (6.52), que corresponde às probabilidades absolutas dos estados do codificador,

$$\underline{\overset{\infty}{p}} \triangleq [\overset{\infty}{p}_1 \ , \ \overset{\infty}{p}_2 \ , \ldots , \ \overset{\infty}{p}_L]^T \tag{6.52}$$

pode ser determinado como sendo o autovetor da matriz $\overline{\Pi}$ associado ao autovalor $\lambda = 1$. Assim $\underline{\overset{\infty}{p}}$ é obtido pela solução do sistema de equações

$$\begin{cases} \overline{\Pi}^T \ \underline{\overset{\infty}{p}} = \underline{\overset{\infty}{p}} & (6.53a) \\ \sum_{i=1}^{L} \overset{\infty}{p}_i = 1 & (6.53b) \end{cases}$$

onde $\overset{\infty}{p}_i = \text{Prob}\,[S_m = s_i]$.

A matriz de transição de estado em k passos $\overline{\Pi}_k$, é dada por:

$$\overline{\Pi}_k = \overline{\Pi}^k \ , \quad k=1,2,\ldots \tag{6.54}$$

onde o (i,j) elemento de $\overline{\Pi}_k$ é a probabilidade condicional

$$\Pi_k(j/i) \triangleq \text{Prob}\,[S_{m+k} = s_j / S_m = s_i].$$

Isto é, $\Pi_k(j/i)$ é a probabilidade do codificador se encontrar no estado s_j no instante m+k, sabendo-se que no instante m ele se encontra no estado s_i.

b) Supondo que a cadeia de Markov é regular, o limite da matriz de k transições quando $k \to \infty$ é dado por

$$\overline{\Pi}_\infty = \lim_{k\to\infty} \overline{\Pi}_k = \| \ \Pi_\infty(j/i) \ \| \ , \ (i,j=1,\ldots,L) \tag{6.55}$$

onde $\Pi_\infty(j/i) = \overset{\infty}{p}_j$. Isto é, $\overline{\Pi}_\infty$ é univocamente determinada pelas probabilidades absolutas dos estados.

6.4.2 - PROBABILIDADES DAS PALAVRAS-CÓDIGO

Consideremos a seguinte matriz diagonal:

$$\overline{\overline{D}}^{\infty} \triangleq || \overline{p}_j^{\infty} \delta_{ij} || , \quad (i,j=1,2,\ldots,L) \tag{6.56}$$

que é univocamente caracterizada por $\underline{\overline{p}}^{\infty}$.

Mostraremos agora que as sub-matrizes das probabilidades conjuntas das palavras-código dadas por (6.45) são dadas por:

$$\overline{P}_0^{uv} = \delta_{uv} q_u \overline{\overline{D}}^{\infty} , \quad (u,v=1,2,\ldots,K) \tag{6.57}$$

$$\overline{P}_k^{uv} = q_u q_v \overline{\overline{D}}^{\infty} E_u \overline{\Pi}^{k-1} , \quad k \geq 1 \text{ e } (u,v=1,2,\ldots,K) \tag{6.58}$$

onde δ_{uv} é o símbolo de Kronecker e q_u são as probabilidades dos vetores de informação.

PROVA:

Através dos dois índices de ordenação introduzidos na seção 6.2, foi estabelecida uma correspondência um-a-um entre os pares ordenados estado-vetor de informação e as palavras- código, isto é, $(s_i,\underline{\beta}_u) \leftrightarrow \underline{\alpha}_{iu}$, $(s_j,\underline{\beta}_v) \leftrightarrow \underline{\alpha}_{jv}$, etc. Assim, a probabilidade conjunta de palavras-código dada por (6.45) pode ser calculada por:

$$P_k^{uv}(i,j) = \text{Prob}[S_m = s_i \cap \underline{b}_m = \underline{\beta}_u \cap S_{m+k} = s_j \cap \underline{b}_{m+k} = \underline{\beta}_v] . \tag{6.59}$$

Para k=0, temos:

$$P_0^{uv}(i,j) = \text{Prob}[S_m = s_i \cap \underline{b}_m = \underline{\beta}_u \cap S_m = s_j \cap \underline{b}_m = \underline{\beta}_v] . \tag{6.60a}$$

Como é impossível num mesmo instante o codificador assumir dois estados distintos e receber dois vetores de informação distintos, a expressão (6.60a) pode ser escrita como:

$$P_o^{uv}(i,j) = \delta_{uv} \delta_{ij} \text{Prob}[S_m = s_i \cap \underline{b}_m = \underline{\beta}_u] \tag{6.60b}$$

Considerando que o estado atual S_m é estatiscamente independente do vetor de informação $\underline{b}_m$ que chega ao codificador no instante m, temos:

$$P_0^{uv}(i,j) = \delta_{uv} . \delta_{ij} . \text{Prob}[S_m = s_i] . \text{Prob}[\underline{b}_m = \underline{\beta}_u] =$$

$$= \delta_{uv} \cdot \delta_{ij} \cdot \overset{\infty}{p}_i \cdot q_u = \delta_{uv}\, q_u \cdot \overset{\infty}{p}_j\, \delta_{ij} \qquad (6.60c)$$

Finalmente temos:

$$\overline{P}_0^{uv} \triangleq ||P_0^{uv}(i,j)|| = \delta_{uv}\, q_u\, ||\overset{\infty}{p}_j\, \delta_{ij}|| = \delta_{uv}\, q_u\, \overline{\overset{\infty}{D}} \qquad (6.60d)$$

Para $k \geq 1$, supondo-se que $\underset{\sim}{b}_{m+k}$ é independente de $\underset{\sim}{b}_m$, S_m e S_{m+k}, (6.59) pode ser escrita como:

$$p_k^{uv}(i,j) = \text{Prob.}[S_m = s_i \cap \underset{\sim}{b}_m = \underset{\sim}{\beta}_u \cap S_{m+k} = s_j] \cdot \text{Prob.}[\underset{\sim}{b}_{m+k} = \underset{\sim}{\beta}_v]$$

$$= q_v \cdot \text{Prob.}[S_m = s_i \cap \underset{\sim}{b}_m = \underset{\sim}{\beta}_u \cap S_{m+k} = s_j] =$$

$$= q_v \cdot \text{Prob.}[S_{m+k} = s_j / S_m = s_i \cap \underset{\sim}{b}_m = \underset{\sim}{\beta}_u] \cdot \text{Prob}[S_m = s_i \cap \underset{\sim}{b}_m = \underset{\sim}{\beta}_u] =$$

$$= q_v \cdot \text{Prob}[S_{m+k} = s_j / S_m = s_i \cap \underset{\sim}{b}_m = \underset{\sim}{\beta}_u] \cdot \text{Prob}[S_m = s_i] \cdot \text{Prob}[\underset{\sim}{b}_m = \underset{\sim}{\beta}_u] =$$

$$= q_v \cdot Z_k(j/i,u) \cdot \overset{\infty}{p}_i \cdot q_u$$

$$= q_u\, q_v\, Z_k(j/i,u) \cdot \overset{\infty}{p}_i \qquad (6.61)$$

onde $$Z_k(j/i,u) \overset{\Delta}{=} \text{Prob.}[S_{m+k} = s_j / S_m = s_i \cap \underset{\sim}{b}_m = \underset{\sim}{\beta}_u] \qquad (6.62)$$

Para k=1, dado que $S_m = s_i$ e $\underset{\sim}{b}_m = \underset{\sim}{\beta}_u$, o próximo estado S_{m+1} é s_j, com probabilidade 1, se $\underset{\sim}{\beta}_u \in \underset{\sim}{B}_{ij}$ enquanto que $S_{m+1} \neq s_j$ se $\underset{\sim}{\beta}_u \notin B_{ij}$. Assim,

$$Z_1(j/i,u) = \begin{cases} 1 \ ; \text{ se } \underset{\sim}{\beta}_u \in B_{ij} \\ \\ 0 \ ; \text{ se } \underset{\sim}{\beta}_u \notin B_{ij} \end{cases}$$

e portanto $Z_1(j/i,u) = e_u(i,j)$ (dado por (6.6)). Assim,

$$P_1^{uv}(i,j) = q_u\, q_v\, \overset{\infty}{\gamma}_i\, e_u(i,j) \qquad (6.63)$$

Para $k \geq 2$, consideremos a seguinte partição do evento $\{S_{m+k} = s_j\}$.

$$\{S_{m+k} = s_j\} = \bigcup_{\ell=1}^{L} \{S_{m+k-1} = s_\ell \cap \underset{\sim}{b}_{m+k-1} \in B_{\ell j}\} \qquad (6.64)$$

que representa as L diferentes maneiras de chegar ao estado s_j na (m+k)-ésima transição tendo como início a transição anterior, isto é, a (m+k-1)-ésima transição. Substituindo (6.64) em (6.62) temos:

$$Z_k(j/i,u) = \text{Prob.}\left[\bigcup_{\ell=1}^{L} \{S_{m+k-1}=s_\ell \cap b_{m+k-1} \in B_{\ell j}\}/S_m=s_i \cap \underset{\sim}{b}_m=\underset{\sim}{\beta}_u\right] =$$

$$= \sum_{\ell=1}^{L} \text{Prob.}\left[S_{m+k-1}=s_\ell \cap \underset{\sim}{b}_{m+k-1} \in B_{\ell j}/S_m=s_i \cap \underset{\sim}{b}_m=\underset{\sim}{\beta}_u\right] \tag{6.65}$$

Observemos que em (6.65) usamos o fato de que a partição (6.64) é disjunta.

Desde que, para $k \geqslant 2$, $\underset{\sim}{b}_{m+k-1}$ seja independente de S_{m+k-1}, S_m e $\underset{\sim}{b}_m$, o que é verdade pelas suposições feitas anteriormente,

$$Z_k(j/i,u) = \sum_{\ell=1}^{L} \text{Prob.}[\underset{\sim}{b}_{m+k-1} \in B_{\ell j}] \cdot \text{Prob.}[S_{m+k-1}=s_\ell /$$

$$/ S_m=s_i \cap \underset{\sim}{b}_m=\underset{\sim}{\beta}_u] = \sum_{\ell=1}^{L} \Pi(j/\ell) \cdot Z_{k-1}(\ell/i,u) \tag{6.66}$$

onde em (6.66) usamos (6.50) e (6.62).

Multiplicando ambos os membros de (6.66) por $q_u\, q_v\, \overset{\infty}{p}_i$, obtemos:

$$P_k^{uv}(i,j) = q_u\, q_v\, Z_k(j/i,u)\overset{\infty}{p}_i = \sum_{\ell=1}^{L} \Pi(j/\ell).Z_{k-1}(\ell/i,u) q_u\, q_v\, \overset{\infty}{p}_i =$$

$$= \sum_{\ell=1}^{L} \Pi(j/\ell).q_u\, q_v\, Z_{k-1}(\ell/i,u)\, \overset{\infty}{p}_i =$$

$$= \sum_{\ell=1}^{L} \Pi(j/\ell).P_{k-1}^{uv}(i,\ell) = \sum_{\ell=1}^{L} P_{k-1}^{uv}(i,\ell).\Pi(j/\ell) \tag{6.67}$$

A expressão (6.67) representa o produto entre a i-ésima linha de $\overline{P}_{k-1}^{uv}$ e a j-ésima coluna de $\overline{\Pi}$. Assim,

$$\overline{P}_k^{uv} = \overline{P}_{k-1}^{uv} \cdot \overline{\Pi} \tag{6.68}$$

O uso repetitivo de (6.68) finalmente nos dá:

$$\overline{P}_k^{uv} = q_u.q_v.\overline{\overset{\infty}{D}}.\overline{E}_u.\overline{\Pi}^{k-1} \tag{6.69}$$

Aplicando as equações (6.60d) e (6.69) à equação (6.47), obtém-se as matrizes de correlação das palavras-código dependendo apenas das probabilidades dos vetores de informação das probabilidades absolutas dos estados e da matriz transição de estado. Isto é,

$$\overline{R}_a(0) \triangleq R_0 = \sum_{u=1}^{K} q_u \, \overline{A}_u^T . \overline{\overline{D}}^{\infty} . \overline{A}_u \tag{6.70}$$

$$\overline{R}_a(k) \triangleq \overline{R}_k = \sum_{u=1}^{K} \sum_{v=1}^{K} q_u . q_v \, \overline{A}_u^T \, \overline{\overline{D}}^{\infty} . \overline{E}_u . \overline{\Pi}^{k-1} . \overline{A}_v =$$

$$= \sum_{u=1}^{K} \sum_{v=1}^{K} (q_u \, \overline{A}_u^T \, \overline{\overline{D}}^{\infty} \, \overline{E}_u) . \overline{\Pi}^{k-1} . (q_v \overline{A}_v) =$$

$$= \sum_{u=1}^{K} (q_u \, \overline{A}_u \, \overline{\overline{D}}^{\infty} \, \overline{E}_u) . \overline{\Pi}^{k-1} . \sum_{v=1}^{K} q_v \, \overline{A}_v =$$

$$= \overline{C}_1^T . \overline{\Pi}^{k-1} . \overline{C}_2 . \tag{6.71}$$

onde $$\overline{C}_1 = \sum_{u=1}^{K} q_u \, \overline{E}_u^T \, \overline{\overline{D}}^{\infty} \, \overline{A}_u , \tag{6.72}$$

$$\overline{C}_2 = \sum_{u=1}^{K} q_u \, \overline{A}_u . \tag{6.73}$$

Tomando o limite de (6.71), quando $k \to \infty$ temos:

$$\overline{R}_a(\infty) \triangleq \overline{R}_\infty = \overline{C}_1^T . \overline{\Pi}_\infty . \overline{C}_2 \tag{6.74}$$

Usando (6.70) e (6.74) em (6.40) obtemos

$$\overline{\Gamma}_1 = \sum_{u=1}^{K} q_u \, \overline{A}_u^T . \overline{\overline{D}}^{\infty} . \overline{A}_u - \overline{C}_1^T . \overline{\Pi}_\infty \, \overline{C}_2 \tag{6.75}$$

$$\overline{\Gamma}_2 = \sum_{k=1}^{\infty} (\overline{C}_1^T . \overline{\Pi}^{k-1} . \overline{C}_2 - \overline{C}_1^T \, \overline{\Pi}_\infty \, \overline{C}_2) \, e^{-j2\pi f k T_n} =$$

$$= \overline{C}_1^T \sum_{k=1}^{\infty} w^{-k} . \overline{\Pi}^{k-1} . \overline{C}_2 - \overline{C}_1^T \sum_{k=1}^{\infty} w^{-k} \, \overline{\Pi}_\infty . \overline{C}_2 =$$

$$= \bar{C}_1^T \left[\sum_{k=1}^{\infty} w^{-k} . \bar{\Pi}^{k-1} . \bar{C}_2 - \sum_{k=1}^{\infty} w^{-k} . \bar{\Pi}_\infty . \bar{C}_2 \right] =$$

$$= \bar{C}_1^T \left[\sum_{k=1}^{\infty} w^{-k} . (\bar{\Pi}^{k-1} - \bar{\Pi}_\infty) \right] \bar{C}_2 \ , \qquad (6.76)$$

onde $w = e^{j2\pi f T_n}$.

6.5 - FÓRMULA EXATA PARA A DENSIDADE ESPECTRAL

Pelas equações (6.39a), (6.39b) e (6.40) vemos que a avalição exata da densidade espectral do processo x(t) implica na avaliação de uma série de somas de matrizes dada por $\bar{\Gamma}_2$ reescrita na forma (6.76).

Mostremos que:

$$\bar{Y}_n \triangleq \sum_{k=1}^{n} w^{-k} (\bar{\Pi}^{k-1} - \bar{\Pi}_\infty) \qquad (6.77)$$

converge para

$$\bar{Y}_\infty \triangleq (\bar{U} - \bar{\Pi}_\infty)[w\bar{U} - (\bar{\Pi} - \bar{\Pi}_\infty)]^{-1} \qquad (6.78)$$

quando $n \to \infty$ (*).

A convergência de $\bar{Y}_n$ quando $n \to \infty$ está intimamente relacionada com o fato de $\bar{\Pi}$ ser uma matriz estocástica.

A matriz $\bar{\Pi}_\infty$ só existe se, e somente se, a cadeia homogênea de Markov é regular [11]. Com essa hipótese, a matriz transição de estado satisfaz às seguintes propriedades:

1) Os autovalores μ_i, (i=1.2,...,L) de uma matriz de transição regular satisfazem às seguintes relações:

$$\mu_i = 1, \ |\mu_i| < 1 \ , \ (i=1,2,\ldots,L) \qquad (6.79)$$

2) A matriz limite $\bar{\Pi}_\infty$ é indepotente, isto é,

$$\bar{\Pi}_\infty^{\,2} = \bar{\Pi}_\infty \quad \text{e} \quad \bar{\Pi}_\infty . \bar{\Pi} = \bar{\Pi} . \bar{\Pi}_\infty = \bar{\Pi}_\infty \qquad (6.80)$$

3) Os auto-valores λ_i, (i=1,2,...,L) da matriz

$$\bar{F} \triangleq \bar{\Pi} - \bar{\Pi}_\infty \qquad (6.81)$$

são dados por:

(*) $\bar{U}$ é a matriz identidade.

$$\lambda_1 = 0 \ , \ \lambda_i = \mu_i \ , \quad |\lambda_i| < 1, \ (i=2,\ldots,L)$$

onde μ_i são os auto-valores de $\overline{\Pi}$.

Como $\overline{F}$ é na realidade dada por

$$\overline{F} = \lim_{k\to\infty} (\overline{\Pi} - \overline{\Pi}^k) = \overline{\Pi} - \overline{\Pi}_\infty,$$

da teoria de polinômios com variável matricial, temos que se μ_i são os auto-valores de uma matriz $\overline{\Pi}$, então $g_k(\mu_i)$ são os autovalores de $g_k(\overline{\Pi})$, onde $g_k(\overline{\Pi})$ é um polinômio de variável matricial [11].

Considerando

$$g_k(\mu_i) = \mu_i - \mu_i^k \ ,$$

$$\overline{F} = \lim_{k\to\infty} g_k(\Pi).$$

Assim,

$$\lambda_i = \lim_{k\to\infty} g_k(\mu i) = \mu_i$$

pela propriedade (1) e

$$\lambda_1 = (\mu_1 - \mu_1) = 0.$$

Usando a propriedade (2),

$$(\overline{\Pi}-\overline{\Pi}_\infty)^k \equiv \binom{k}{0}\overline{\Pi}^k - \binom{k}{1}\overline{\Pi}^{k-1} \cdot \overline{\Pi}_\infty + \ldots + \binom{n}{n}\overline{\Pi}_\infty^k =$$

$$\equiv \overline{\Pi}^k - \overline{\Pi}_\infty \qquad (k=1,2,\ldots) \qquad (6.82a)$$

$$(\overline{\Pi} - \overline{\Pi}_\infty)^k - \overline{\Pi}_\infty \equiv \overline{\Pi}^k - \overline{\Pi}_\infty, \ k=0 \qquad (6.82b)$$

Após todas essas considerações, (6.77) pode ser escrita como,

$$Y_n = \sum_{k=0}^{n-1} w^{-k-1}(\overline{\Pi}^k-\overline{\Pi}_\infty) = w^{-1}\left[\sum_{k=0}^{n-1} w^{-k}(\overline{\Pi}^k-\overline{\Pi}_\infty)\right] =$$

$$= w^{-1}\left[w^{-k}(\overline{\Pi}^k-\overline{\Pi}_\infty)\Big|_{k=0} + \sum_{k=1}^{n-1} w^{-k}(\overline{\Pi}^k-\overline{\Pi}_\infty)\right] =$$

$$= w^{-1}\left[w^{-k}(\overline{\Pi}-\overline{\Pi}_\infty)^k\Big|_{k=0} - \overline{\Pi}_\infty + \sum_{k=1}^{n-1} w^{-k}(\overline{\Pi}-\overline{\Pi}_\infty)^k\right] =$$

$$= w^{-1}[w^{-k}\ \bar{F}^{k}_{k=0} - \bar{\Pi}_{\infty} + \sum_{k=1}^{n-1} (w^{-1}\ \bar{F})^{k}] =$$

$$= w^{-1}[\sum_{k=0}^{n-1} (w^{-1}\ \bar{F})^{k} - \bar{\Pi}_{\infty}]$$

$$= u[\sum_{k=0}^{n-1} (u.\bar{F})^{k} - \bar{\Pi}_{\infty}] \ . \qquad (6.83)$$

onde $u = w^{-1}$ e $\bar{F} = \bar{\Pi} - \bar{\Pi}_{\infty}$.

Como a matriz $\bar{F}$ tem auto-valores que satsifazem à condição $|\lambda_i| < 1$, satisfaz à propriedade[11]

$$\sum_{k=0}^{\infty} \bar{F}^{k} = (\bar{U} - \bar{F})^{-1} \qquad (6.84)$$

Assim, (6.83), quando n tende para o infinito, pode ser escrita como

$$Y_{\infty} = u[(\bar{U} - u\bar{F})^{-1} - \bar{\Pi}_{\infty}] =$$

$$= u[\bar{U}(\bar{U} - u\bar{F})^{-1} - \bar{\Pi}_{\infty}(\bar{U} - u\bar{F})(\bar{U} - u\bar{F})^{-1}] =$$

$$= u\{[\bar{U} - \bar{\Pi}_{\infty}(\bar{U} - u\bar{F})](\bar{U} - u\bar{F})^{-1}\} =$$

$$= u\{[\bar{U} - \bar{\Pi}_{\infty} - u\bar{\Pi}_{\infty}(\bar{\Pi} - \bar{\Pi}_{\infty})](\bar{U} - u\bar{F})^{-1}\} =$$

$$= u\{[\bar{U} - \bar{\Pi}_{\infty} - u\bar{\Pi}_{\infty} + u\bar{\Pi}_{\infty}](\bar{U} - u\bar{F})^{-1}\} =$$

$$= u\{[\bar{U} - \bar{\Pi}_{\infty}](\bar{U} - u\bar{F})^{-1}\} =$$

$$= (\bar{U} - \bar{\Pi}_{\infty})\ u\ (\bar{U} - u\bar{F})^{-1} =$$

$$= (\bar{U} - \bar{\Pi}_{\infty})\ u[u(w\bar{U} - \bar{F})]^{-1} =$$

$$= (\bar{U} - \bar{\Pi}_{\infty})(w\bar{U} - \bar{F})^{-1} =$$

$$= (\bar{U} - \bar{\Pi}_{\infty})\ [w\bar{U} - (\bar{\Pi} - \bar{\Pi}_{\infty})]^{-1} \ . \qquad (6.85)$$

Portanto a equação (6.76) pode ser escrita como:

$$\bar{\Gamma}_2 = \bar{C}_1^T.(\bar{U} - \bar{\Pi}_\infty)\,[w\bar{U} - (\bar{\bar{\Pi}} - \bar{\Pi}_\infty)]^{-1}.\bar{C}_2 \qquad (6.86)$$

e assim o espectro de potência do processo x(t) pode ser avaliado de uma forma exata. Isto é,

$$X_c(f) = \bar{V}(\bar{R}_0 - \bar{R}_\infty)\bar{V}^* + 2\,Re\{\bar{V}\,\bar{C}_1^T(\bar{U}-\bar{\Pi}_\infty)[w\bar{U} - (\bar{\Pi} - \bar{\Pi}_\infty)]^{-1}\,\bar{C}_2\,\bar{V}^*\} \qquad (6.87a)$$

$$X_d(f) = \bar{V}\,\bar{R}_\infty\,\bar{V}^* \qquad (6.87b)$$

com $W_x(f)$ dado por (6.33).

A equação (6.87a) pode ser escrita ainda como:

$$X_c(f) = X_1(f) + 2Re\{X_2(f)\} \qquad (6.88)$$

onde

$$X_1(f) \triangleq \bar{V}.(\bar{R}_0 - \bar{R}_\infty)\,.\,\bar{V}^*\,, \qquad (6.89a)$$

$$X_2(f) \triangleq \bar{V}\,\bar{C}_1^T(\bar{U} - \bar{\Pi}_\infty)[w\bar{U} - (\bar{\Pi} - \bar{\Pi}_\infty)]^{-1}\,\bar{C}_2.\bar{V}^* \qquad (6.89b)$$

6.6 - COMPORTAMENTO EM FREQUÊNCIA

Na seção anterior obtivemos resultados exatos para o espectro de potência do sinal digital x(t). Entretanto as expressões obtidas não mostrem explicitamente o comportamento em frequência do espectro de x(t). Assim, nesta seção nos preocupamos em reescrever os resultados obtidos anteriormente, de modo a explicitar esse comportamento em frequência do processo x(t).

Consideremos inicialmente a equação (6.87b) escrita como:

$$X_d(f) = [e^{j\omega T},\ e^{j2\omega T},\ldots,e^{jn\omega T}] \begin{bmatrix} R_\infty(1,1) & R_\infty(1,2) & \ldots & R_\infty(1,n) \\ R_\infty(2,1) & R_\infty(2,2) & \ldots & R_\infty(2,n) \\ \vdots & \vdots & & \vdots \\ R_\infty(n,1) & R_\infty(n,2) & \ldots & R_\infty(n,n) \end{bmatrix} \begin{bmatrix} e^{-j\omega T} \\ e^{-j2\omega T} \\ \vdots \\ e^{-jn\omega T} \end{bmatrix}$$

onde $\omega \triangleq 2\pi f$. Após a multiplicação das matrizes indicadas, podemos escrever $X_d(f)$ como:

$$X_d(f) = \sum_{k=1}^{n} R_\infty(k,1)e^{j\omega(k-1)T} + \sum_{k=1}^{n} R_\infty(k,2)e^{j\omega(k-2)T} + \ldots$$

$$\ldots + \sum_{k=1}^{n} R_\infty(k,n)e^{j\omega(k-n)T}$$

Como a matriz de auto-correlação é simétrica, após o desenvolvimento de cada somatório e a ordenação dos termos, temos:

$$X_d(f) = \sum_{i=1}^{n} R_\infty(i,i) + \sum_{i=1}^{n-1} R_\infty(i,i+1)(e^{j\omega T} + e^{-j\omega T}) +$$

$$+ \sum_{i=1}^{n-2} R_\infty(i,i+2)(e^{j2\omega T} + e^{-j2\omega T}) + \ldots$$

$$\ldots + \sum_{i=1}^{n-(n-1)} R_\infty(i,i+n-1)\ (e^{j(n-1)\omega T} + e^{-j(n-1)\omega T}) =$$

$$= \sum_{i=1}^{n} R_\infty(i,i) + 2 \sum_{i=1}^{n-1} R_\infty(i,i+1)\ \cos\omega T + \ldots$$

$$\ldots + 2 \sum_{i=1}^{n-(n-1)} R_\infty(i,i+n-1)\ \cos(n-1)\omega T =$$

$$= \sum_{k=0}^{n-1} \varepsilon_k \mu_k\ \cos k\omega T. \qquad (6.90)$$

onde: $\varepsilon_k = 1$, se $k=0$

$= 2$, se $k \geqslant 1$ (6.91)

$$\mu_k = \sum_{i=1}^{n-k} R_\infty(i,i+k) \qquad (6.92)$$

Portanto, a forma explícita para $X_d(f)$ é dada por (6.90), isto é,

$$X_d(f) = \sum_{k=0}^{n-1} \varepsilon_k \mu_k\ \cos k\omega T \qquad (6.93)$$

Todas as considerações feitas na obtenção de (6.93) podem ser aplicadas a $X_1(f)$ dado por (6.89a) de modo que obtém-se:

$$X_1(f) = \sum_{k=0}^{n-1} \varepsilon_k \nu_k \cos k\omega T \qquad (6.94)$$

com ε_k dado por (6.91) e

$$\nu_k = \sum_{i=1}^{n-k} [R_0(i,i+k) - R_\infty(i,i+k)] \qquad (6.95)$$

Ao componente $X_2(f)$, dado por (6.89b), não se pode fazer as considerações feitas anteriormente já que a matriz $\overline{\Gamma}_2$ não possui qualquer simetria particular, além de depender do termo

$$w = e^{j2\pi f T_n} .$$

É necessário que seja efetuada a inversão de matrizes $w\overline{U}-(\overline{\Pi}-\overline{\Pi}_\infty)$. Este é um problema bem conhecido na teoria de funções matriciais que é resolvido através da relação:

$$(w\overline{U}-\overline{F})^{-1} = D(w)^{-1}\,\overline{G} \qquad (6.96)$$

onde $\overline{F} = \overline{\Pi}-\overline{\Pi}_\infty$, $D(w)$ é o polinômio característico de $\overline{F}$ e $\overline{G}$ é denominada de matriz conjunta de $\overline{F}$ [11]. Isto é,

$$D(w) \triangleq |w\overline{U} - \overline{F}| = \sum_{k=1}^{L} d_{L-k}\, w^k \ , \quad (d_0=1,\ d_L=0) \qquad (6.97)$$

$$\overline{G} = \sum_{k=0}^{L} \overline{G}_{L-k}\, w^{k-1} \ , \quad (\overline{G}_L=\overline{0},\ \overline{G}_0=\overline{U}) \qquad (6.98)$$

Os coeficientes d_k e as matrizes coeficientes $\overline{G}_k$ estão relacionados através da expressão

$$\overline{G}_k = \overline{F}^k + d_1\overline{F}^{k-1} + \ldots + d_{k-1}\overline{F} + d_k\overline{U} \qquad (6.99)$$

de modo que se o polinômio característico de $\overline{F}$ é conhecido, as matrizes coeficientes $\overline{G}_k$ podem ser determinadas. Aplicando as considerações anteriores à matriz $\overline{\Gamma}_2$ temos:

$$\overline{\Gamma}_2 = \overline{C}_1^T(\overline{U} - \overline{\Pi}_\infty)(w\overline{U} - \overline{F})^{-1}\,\overline{C}_2 =$$

$$= \overline{C}_1^T(\overline{U} - \overline{\Pi}_\infty)D(w)^{-1}\,\overline{G}\,\overline{C}_2 =$$

$$= \overline{C}_1^T(\overline{U} - \overline{\Pi}_\infty)\,D(w)^{-1}\,\Big[\sum_{k=0}^{L} \overline{G}_{L-k}\, w^{k-1}\Big]\,\overline{C}_2 =$$

$$= \frac{\sum_{k=0}^{L} \overline{C}_1^T(\overline{U} - \overline{\Pi}_\infty)\ \overline{G}_{L-k}\ \overline{C}_2\ w^{k-1}}{D(w)} =$$

$$= \frac{\sum_{k=0}^{L} \overline{C}_1^T(\overline{U} - \overline{\Pi}_\infty)\ \overline{G}_{L-k}\ \overline{C}_2\ w^{k-1}}{\sum_{k=1}^{L} d_{L-k}\ w^k} =$$

$$= \frac{\sum_{k=0}^{L} \overline{C}_1^T(\overline{U} - \overline{\Pi}_\infty)\ \overline{G}_{L-k}\ \overline{C}_2\ e^{jn(k-1)\omega T}}{\sum_{k=1}^{L} d_{L-k}\ e^{jnk\omega T}} \qquad (6.100)$$

De (6.100) podemos escrever $X_2(f)$ como

$$X_2(f) = \frac{\overline{V} \sum_{k=0}^{L} \overline{C}_1^T(\overline{U} - \overline{\Pi}_\infty)\ \overline{G}_{L-k}\ \overline{C}_2\ e^{jn(k-1)\omega t} \cdot \overline{V}^*}{\sum_{k=1}^{L} d_{L-k}\ e^{jnk\omega T}}$$

$$= \frac{\overline{V} \sum_{k=0}^{L} \overline{H}_{k-1}\ e^{jn(k-1)\omega T} \cdot \overline{V}^*}{\sum_{k=1}^{L} d_{L-k}\ e^{jnk\omega T}} =$$

$$= \frac{\sum_{k=0}^{L} \overline{V}\ \overline{H}_{k-1}\ \overline{V}^*\ e^{jn(k-1)\omega T}}{\sum_{k=1}^{L} d_{L-k}\ e^{jnk\omega T}} \qquad (6.101)$$

onde $\overline{H}_{k-1} = \overline{C}_1^T(\overline{U} - \overline{\Pi}_\infty)\ \overline{G}_{L-k}\ \overline{C}_2$. (6.102)

Explicitando-se o produto $\overline{V}\ \overline{H}_{k-1}\ \overline{V}^*$ em (6.101) obtém-se

$$\overline{V}\ \overline{H}_{k-1}\ \overline{V}^* = \sum_{i=1}^{n} H_{k-1}(i,1)\ e^{j(i-1)\omega T} +$$

$$+ \sum_{i=1}^{n} H_{k-1}(i,2)\ e^{j(i-2)\omega T} + \ldots$$

$$\ldots + \sum_{i=1}^{n} H_{k-1}(i,n)\, e^{j(i-n)\omega T} ,$$

que após o desenvolvimento de cada um dos somatórios e rearranjamento dos termos pode ser colocado na forma:

$$\overline{V}\,\overline{H}_{k-1}\,\overline{V}^{*} = \sum_{i=1}^{n} H_{k-1}(i,i) + \sum_{i=1}^{n-1} H_{k-1}(i,i+1)\, e^{j\omega T} +$$

$$+ \sum_{i=1}^{n-1} H_{k-1}(i+1,i)\, e^{-j\omega T} + \ldots$$

$$\ldots + \sum_{i=1}^{n-(n-1)} H_{k-1}(i,i+n-1)\, e^{-j(n-1)\omega T} +$$

$$+ \sum_{i=1}^{n-(n-1)} H_{k-1}(i+n-1,i)\, e^{-j(n-1)\omega T} \qquad (6.103)$$

Multiplicando-se (6.103) por $e^{jn(k-1)\omega T}$ obtém-se:

$$\overline{V}\,\overline{H}_{k-1}\,\overline{V}^{*}\, e^{jn(k-1)\omega T} = \sum_{i=1}^{n} H_{k-1}(i,i)\, e^{jn(k-1)\omega T} +$$

$$+ \sum_{i=1}^{n-1} H_{k-1}(i,i+1)\, e^{j[n(k-1)+1]\omega T} +$$

$$+ \sum_{i=1}^{n-1} H_{k-1}(i+i,i)\, e^{j[n(k-1)-1]\omega T} + \ldots$$

$$\ldots + \sum_{i=1}^{n-(n-1)} H_{k-1}(i,i+n-1) e^{j[n(k-1)+n-1]\omega T} +$$

$$+ \sum_{i=1}^{n-(n-1)} H_{k-1}(i+n-1,i) e^{j[n(k-1)-(n-1)]\omega T} . \qquad (6.104)$$

Substituindo-se (6.104) em (6.101) e após alguns rearranjamentos, obtém-se:

$$X_2(f) = \frac{\sum_{s=-(n-1)}^{nL-1} n_s\, e^{js\omega T}}{\sum_{k=1}^{L} d_{L-k}\, e^{jkn\omega T}} \qquad (6.105)$$

onde $s=n(k-1)+p$, $(k=0,1,\ldots,L$ e $p=0,1,\ldots,n-1)$.

Tomando-se a parte real de (6.105) multiplicada por 2 (equação (6.39a)) e após alguma álgebra obtém-se:

$$2R_e[X_2(f)] = \frac{\sum_{k=0}^{n(L+1)-1} \varepsilon_k N_k \cos k\omega T}{\sum_{k=0}^{L} \varepsilon_k D_K \cos nk\omega T} \qquad (6.106)$$

onde $\varepsilon_0=1$, $\varepsilon_k=2$ $(k\geqslant 1)$ e

$$N_k \triangleq N_{p+qn} = \sum_{h=0}^{L-q} [d_{L-h}\, n_{(h+q)n+p} + d_{L-(h+q)} \cdot n_{hn-p}] \qquad (6.107a)$$

$$D_k \triangleq D_q = \sum_{h=0}^{L-q} d_{L-h} \cdot d_{L-(n+q)} \qquad (6.107b)$$

com $p=0,1,\ldots,n-1$ e $q=0,1,\ldots,L$.

Assim temos finalmente $X_c(f)$ explícito em termos de frequência dado por:

$$X_c(f) = \sum_{k=0}^{n-1} \varepsilon_k \nu_k \cos k\omega T + \frac{\sum_{k=0}^{n(L+1)-1} \varepsilon_k N_k \cos k\omega T}{\sum_{k=0}^{L} \varepsilon_k D_k \cos nk\omega T} \qquad (6.108)$$

Para finalizar precisamos explicitar a forma como os coeficientes n_s em (6.105) são calculados. Na realidade, esses coeficientes são obtidos em cada um dos somatórios de (6.104) quando k varia de 0 a L. Entretanto, observemos que para cada k, em (6.102) obtém-se uma matriz coeficiente $\overline{G}_k$ diferente, o que implica num grande trabalho de cálculo para obter-se os coeficientes n_s.

Entretanto, existe um método, atribuído a Faddeev, que permite calcular simultaneamente os coeficientes d_k do polinômio característico e as matrizes coeficientes $\overline{G}_k$. As relações são as seguintes:

$$d_k = 1/k.\text{traço}[\overline{F}.\overline{G}_{k-1}] \qquad (6.109a)$$

$$\overline{G}_k = \overline{F}\,\overline{G}_{k-1} + d_k\,\overline{U} \qquad (k=1,\ 2,\ldots,L) \qquad (6.109b)$$

com condições iniciais $d_0=1$ e $\overline{G}_0=\overline{U}$.

Para cada $\overline{G}_k$ calculado podemos efetuar o produto de matrizes.

$$\overline{H}_k \triangleq \overline{C}_1^T(\overline{U} - \overline{\Pi}_\infty)\ \overline{G}_{L-k-1}\ \overline{C}_2 \qquad (k=0,1,\ldots,L-1) \qquad (6.110)$$

e assim obtém-se

$$n_s \triangleq n_{p+kn} = \sum_{i=1}^{n-p} H_k(i+p,i) + \sum_{i=1}^{p} H_{k+1}(i,i+n-p) \qquad (6.111)$$

onde $p=0,1,\ldots,n-1$, $k=1,0,1,0,\ldots,L-1$ e

$$\overline{H}_{-1} = \overline{H}_L = \overline{0} \qquad (6.112)$$

No trabalho de Cariolaro foi feita a seguinte sugestão para a implementação de um programa, para computador digital, que calcule o espectro de potência de códigos de bloco (n.k):

1) A partir da função de saída do codificador determine as submatrizes $\overline{A}_u$ $(u=1,\ldots,K)$.
2) A partir da função transição de estado do codificador determine as submatrizes $\overline{E}_u$.
3) Calcule as probabilidades dos vetores de informação
4) Calcule a matriz transição de estado $\overline{\Pi}$.
5) Calcule o auto-vetor $\overset{\infty}{p} = [\overset{\infty}{p}_1\ \overset{\infty}{p}_2 \ldots \overset{\infty}{p}_L]^T$ das probabilidades absolutas dos estados do codificador, as matrizes $\overline{\Pi}_\infty$ e $\overline{D}^\infty$.
6) Calcule as matrizes $\overline{C}_1$ e $\overline{C}_2$.
7) Calcule $\overline{R}_0$ e $\overline{R}_\infty$.
8) Calcule os coeficientes d_k e as matrizes coeficientes $\overline{G}_k$.
9) Calcule os coeficientes n_s.

CAPÍTULO 7

O PROGRAMA E RESULTADOS

7.1 - GENERALIZAÇÃO DO MÉTODO DE CARIOLARO

Analisando-se os passos a serem seguidos para a implementação do programa para computador digital sugeridos no trabalho de Cariolaro (seção 6.6) observamos que eles são apropriados apenas para os códigos de bloco não lineares, já que não existem leis matemáticas que possam caracterizar, para qualquer código de bloco não linear, a função de saída h, e a função transição de estado g, associadas ao codificador. Assim, no programa sugerido os passos (1) e (2) devem ser seguidos de modo que as submatrizes $\bar{A}_u$ e $\bar{E}_u$ (u=1,2,...K) sejam fornecidas como dados de entrada [2].

Estamos interessados em estender o método proposto por Cariolaro aos códigos lineares (Bloco e Convolucionais). Para isso, revisemos as hipóteses assumidas em [2]:

(1) O codificador possa ser modelado por uma máquina sequencial de estado finito segundo o modelo de Mealy.

(2) A sequência constituida pelos vetores de informação $\{\underline{b}_k\}(-\infty < k < +\infty)$, seja estacionária no sentido fraco.

(3) Os vetores de informação $\underline{b}_k$ sejam estatisticamente independentes.

Observemos que as hipóteses (2) e (3) não impõem e nem implicam em qualquer restrição ao tipo de código utilizado. Entretanto, a hipótese (1) impõe que o codificador possa ser modelado segundo uma máquina de Mealy.

Como vimos no Capítulo 5, o codificador para códigos convolucionais pode ser posto de modo a satisfazer o modelo de Mealy e esse é realmente o modelo utilizado, já que economiza um conjunto de k_0 elementos de memória [9]. E ainda mais, o codificador para códigos convolucionais é uma máquina sequencial de estado finito LINEAR, o que garante funções de saída e transição de estado bem estruturadas (veja Cap. 3 , seção 3.5). Esse fato evita que as sub-matrizes $\bar{A}_u$ e $\bar{E}_u$ sejam fornecidas como dados de entrada. Assim, para os códigos convolucionais, as hipóteses (1), (2) e (3) são satisfeitas.

Em se tratando de códigos de bloco lineares (veja Cap.4, seção 4.4), o codificador (Shift-Register multiplicador de polinômios) apesar de ser uma máquina sequencial linear não satisfaz ao modelo de Mealy, já que os vetores de informação são processados

em série. Assim, precisamos obter um codificador para códigos de bloco lineares que opere na forma paralela. Esse codificador na forma paralela foi obtido da seguinte forma:

Através da associação entre os resultados da Teoria da Codificação e a Teoria de Sistemas Lineares Discretos (ver seção 5.4), mostramos que a matriz de transferência do codificador para um código convolucional é dada pela equação (5.18) e repetida aqui como (7.1).

$$\overline{G}(D) = G_0^T + D\, G_1^T + \ldots + D^{m-1}\, G_{m-1}^T \,. \qquad (7.1)$$

Mas os códigos de bloco lineares são um caso particular dos códigos convolucionais em que m=1. Assim, obtivemos a matriz de transferência do codificador de um código de bloco linear na forma paralela dada pela expressão (5.20) e repetida aqui como (7.2).

$$\overline{G}(D) = G_0^T = G^T \qquad (7.2)$$

O codificador na forma paralela para os códigos de bloco lineares está esquematizado na Fig. 5.5. Portanto as hipóteses (1), (2) e (3) também são satisfeitas para os códigos de bloco lineares.

Finalizando essa seção, observemos que as equações obtidas no cap. 6 para o cálculo dos componentes contínuo e discreto do sinal digital codificado dependem direta ou indiretamente das sub-matrizes $\overline{A}_u$ e $\overline{E}_u$. Assim, nossa programação foi estruturada de acordo com a Fig. 7.1.

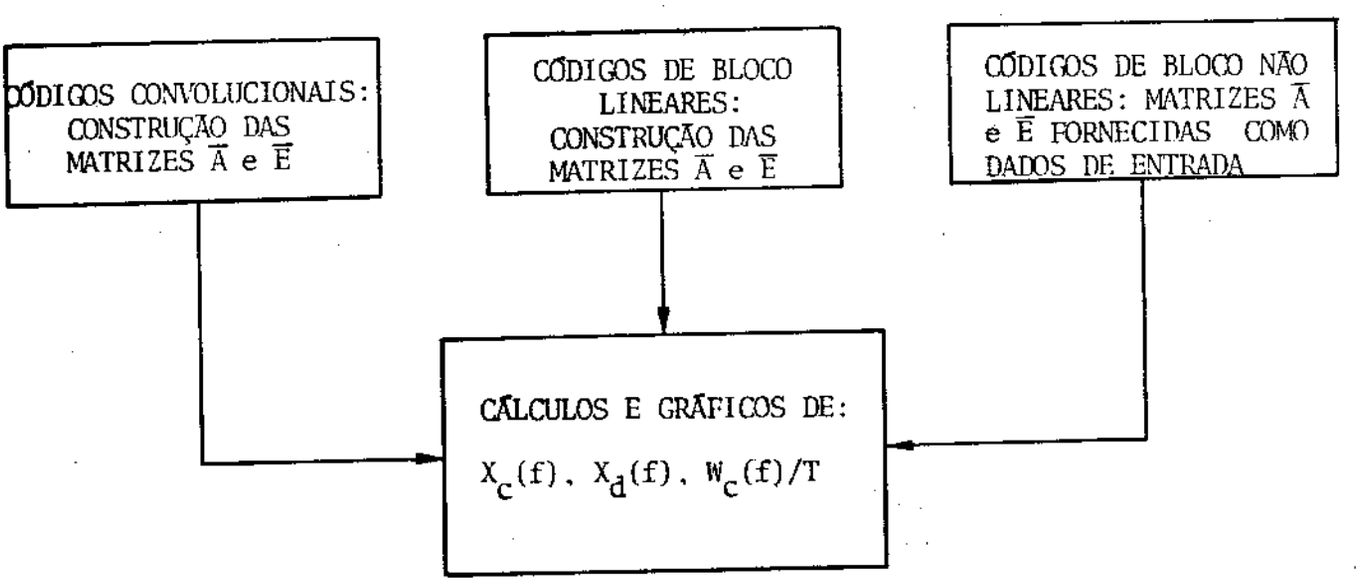


Fig. 7.1 - Estrutura da programação

.2 - CONSTRUÇÃO DAS MATRIZES $\bar{A}$ e $\bar{E}$ PARA CÓDIGOS CONVOLUCIONAIS

O programa implementado para construção das matrizes $\bar{A}$ e $\bar{E}$ para códigos convolucionais tem como base as equações constituintes do codificador (Cap. 3, seção 3.10) que são fornecidas como dados de entrada e para obtê-las os passos a serem seguidos são os descritos na seção 5.4, Cap. 5. Em resumo, o procedimento para uso do programa que determina os componentes contínuo e discreto do espectro de potência de um sinal digital codificado por código convolucional é o seguinte:

(1) - Obter a matriz geradora $\bar{G}$ com as propriedades corretoras desejadas.

(2) - A partir de $\bar{G}$ esquematizar o circuito do codificador de acordo com a Fig. 5.3 ou 5.4 (se o código for sistemático).

(3) - A partir do codificador obtido em (2) obter as suas matrizes constituintes como descrito na seção 5.4.

(4) - Fornecer como dados de entrada:

4.1 - Probabilidade de ocorrer "1" no sinal a ser codificado.

4.2 - Número de memórias do codificador obtido em (2).

4.3 - Os valores de n_0 e k_0 para o código utilizado.

4.4 - As matrizes constituintes obtidas em (3).

.3 - CONSTRUÇÃO DAS MATRIZES $\bar{A}$ e $\bar{E}$ PARA CÓDIGOS DE BLOCO LINEARES

Para a construção das matrizes $\bar{A}$ e $\bar{E}$ de códigos lineares, utilizamos a equação (7.2) e o fato de que o codificador, na forma paralela, para os códigos de bloco lineares é uma máquina sequencial linear de dimensão 0 (zero). Em resumo, o procedimento para uso do programa que determina os componentes contínuo e discreto do espectro de potência do sinal digital codificado por códigos de bloco lineares é como se segue:

(1) - Obter polinômio gerador do código desejado.

(2) - A partir de g(X), obtido em (1), obter matriz geradora $\bar{G}$ de acordo com a equação (4.4).

(3) - A partir de $\overline{G}$, obtida em (2), escrever $\overline{G}^T$.

(4) - Fornecer como dados de entrada:

4.1 - Probabilidade de ocorrer "1" no sinal a ser codificado.

4.2 - Os valores de n e k para o código utilizado.

4.3 - Matriz $\overline{G}^T$ obtida em (3).

4 - CONSTRUÇÃO DAS MATRIZES $\overline{A}$ e $\overline{E}$ PARA CÓDIGOS DE BLOCO NÃO LINEARES

Como já dissemos anteriormente, os códigos de bloco não lineares não possuem uma estrutura matemática que permita uma caracterização genérica das funções transição de estado e de saída da máquina sequencial associada ao codificador. Por isso, para cada código, elas devem ser construídas e fornecidas como dados de entrada. Para obter as matrizes $\overline{A}$ e $\overline{E}$ e utilizar o programa que determina os componentes contínuo e discreto de um sinal digital codificado por código de bloco não linear os passos a serem seguidos são os seguintes:

(1) - Obter o modelo da máquina sequencial de estado finito associado ao codificador.

(2) - Obter os alfabetos* correspondentes a cada estado da maquina obtida em (1).

(3) - A partir dos alfabetos obtidos em (2) escrever as sub-matrizes $\overline{A}_u$ obedecendo sua definição dada pela equação (6.2).

(4) - A partir do diagrama de estado obtido em (1) e de acordo com as equações (6.5) e (6.6) obter as sub-matrizes $\overline{E}_u$.

(5) - Fornecer como dados de entrada:

5.1 - Probabilidade de ocorrer "1" no sinal a ser codificado.

5.2 - Os valores de n e k para o código utilizado e o número de estados associados ao codificador.

(*) Alfabeto é o conjunto das palavras-código associadas aos vetores de informação para o codificador num mesmo estado.

5.3 - As sub-matrizes $\bar{A}_u$ e $\bar{E}_u$ obtidas em (3) e (4).

Nos apêndices A, B e C apresentamos um resumo das características dos programas e sub-programas utilizados na programação, como também seus fluxogramas e listagens.

7.5 - CÓDIGOS UTILIZADOS PARA TESTAR A PROGRAMAÇÃO QUE DETERMINA $X_c(f)$ E $X_d(f)$

Na literatura existem códigos cujas densidades espectrais de potência já são bem conhecidas. Assim, usamos esses códigos para testar a nossa programação. Utilizamos dois códigos, o código de Franaszek ou código MS-43 [2,3] e o código 3B-4B [5].

7.5.1 - CÓDIGO DE FRANASZEK OU MS-43

O código MS-43, como o 3B-4B, pertence a uma classe de códigos de bloco não lineares denominados de "CÓDIGOS CONTADORES". Para esses códigos, a função transição de estado g é dada por:

$$s_{n+1} = S_n + \gamma \ , \quad \gamma = \sum_{k=1}^{n} a_k - \langle a_k \rangle \ . \tag{7.3}$$

onde: s_n é o estado atual.

n é o comprimento do código.

γ é a característica da palavra-código transmitida quando o codificador se encontra no estado S_n.

a_k representa os dígitos utilizados na palavra-código transmitida quando o codificador se encontra no estado S_n.

$\langle a_k \rangle$ é a média dos dígitos sobre todas as palavras- código do código utilizado. É obtida somando-se os dígitos de todas as palavras-código e dividindo essa soma pelo número total de dígitos.

Para o código MS-43,o modelo da máquina sequencial de estado finito associado ao codificador está representado na Fig.7.2 [3].

Utilizando a função de saída do codificador caracterizada pela tabela I de [2] juntamente com as sub-matrizes $\bar{A}_u$ e $\bar{E}_u$ dadas na tabela II de [2], obtivemos o resultado mostrado na Fig. 7.3 para prob(1) = 0,5. Para esse código só existem palavras-código com característica -3, -2, -1, 0, 1, 0, 1, 2 e 3 e $\langle a_n \rangle = 0$. Como $\langle a_k \rangle = 0$, significa que o código apresenta nível DC nulo.

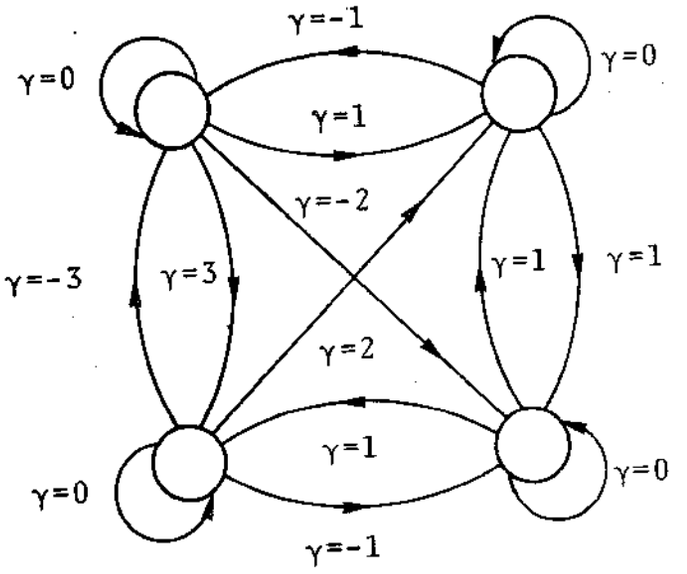


Fig. 7.2 - Diagrama de estado para o codificador do código de Franaszek

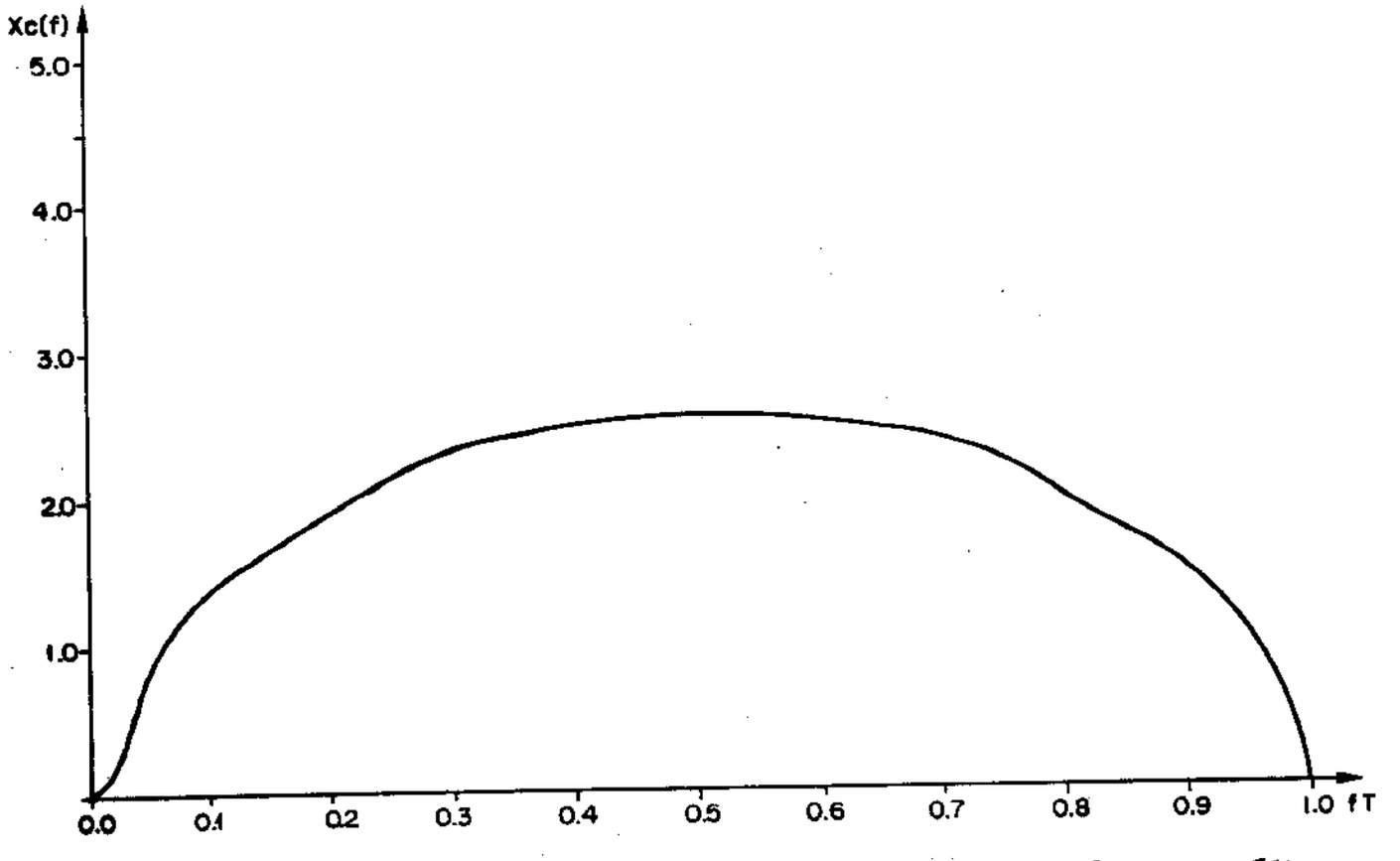


Fig.7.3 - Componente contínuo, normalizado, do espectro de potência do código de Franaszek para prob(1) = 0.5.

7.5.2 - CÓDIGO 3B-4B (TELETTRA) PROPOSTO PARA TRANSMISSÃO EM FIBRA ÓTICA [5]

O diagrama de estado para o codificador está representado na Fig. 7.4.A função de saída para o codificador é dada pela tabela 7.1. Para esse código $\langle a_k \rangle = 0,5$.

u	$\underline{\beta}_u$	$\underline{\alpha}_{iu} = h(s_i, \underline{\beta}_u)$	
		$s_1 = -1/2$	$s_2 = 1/2$
1	0 0 0	0 1 0 1	0 1 0 1
2	0 0 1	1 0 0 1	1 0 0 1
3	0 1 0	1 1 1 0	0 1 0 0
4	0 1 1	1 1 0 1	1 0 0 0
5	1 0 0	0 1 1 1	0 0 1 0
6	1 0 1	1 0 1 1	0 0 0 1
7	1 1 0	0 1 1 0	0 1 1 0
8	1 1 1	1 0 1 0	1 0 1 0

Tabela 7.1
Função de saída para o código 3B-4B da TELETTRA

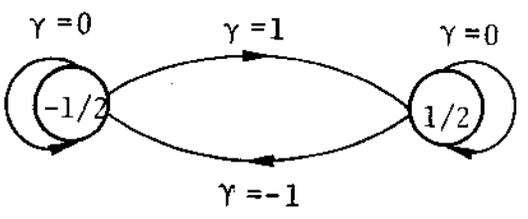


Fig. 7.4
Diagrama de estado para o codificador do código 3B-4B

Para esse código, as sub-matrizes $\overline{A}_u$ são:

$$\overline{A}_1 = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} ; \quad \overline{A}_2 = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix} ; \quad \overline{A}_3 = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} ; \quad \overline{A}_4 = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} ;$$

$$\overline{A}_5 = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} ; \quad \overline{A}_6 = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} ; \quad \overline{A}_7 = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix} ; \quad \overline{A}_8 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix} .$$

Enquanto que as sub-matrizes $\overline{E}_u$ são:

$$\overline{E}_1 = \overline{E}_2 = \overline{E}_7 = \overline{E}_8 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} ; \quad \overline{E}_3 = \overline{E}_4 = \overline{E}_5 = \overline{E}_6 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} .$$

Temos também n=4, k=3 e o número de estados igual a 2.Com esses dados de entrada e para prob(1) = 0,5 obtivemos os resultados mostrados nas Figs. 7.5a e 7.5b.

Os resultados obtidos para os dois códigos - teste estão de acordo com aqueles obtidos em [2,3] para o MS-43 e em [5] para o 3B-4B. Para o código 3B-4B temos como nova informação seu componente discreto que não é apresentado em [5]. O componente discreto do código MS-43 é nulo para prob(1) = 0,5 [2]. Os resultados obtidos nos asseguram o bom funcionamento do corpo principal do programa (ver Fig. 7.1) que consiste de uma sub-rotina identificada por SUBROTINE SPECTR (ver apêndices).

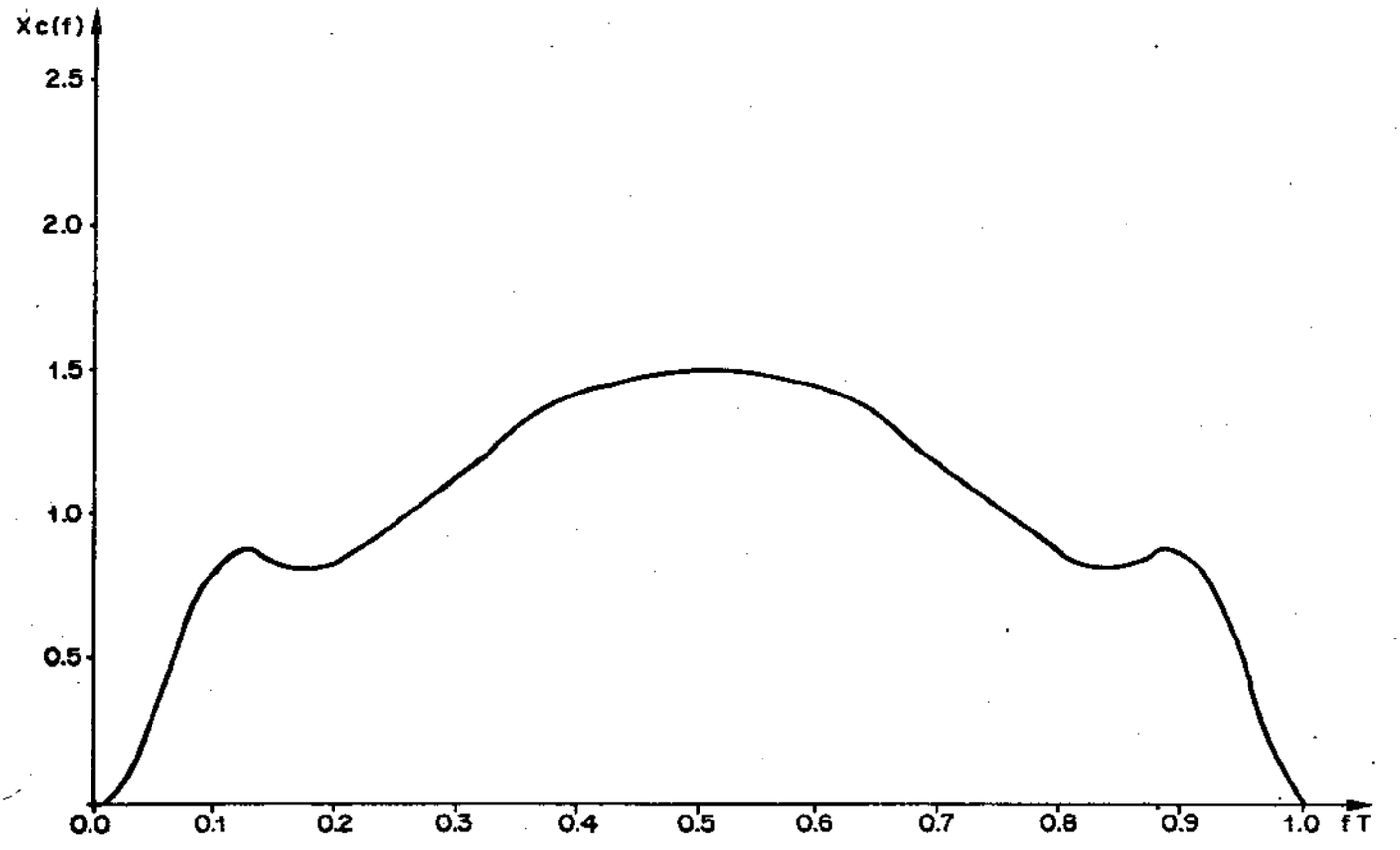


Fig. 7.5a - Componente contínuo,normalizado, do espectro de potência do código 3B-4B proposto em [5] para prob(1) = 0,5.

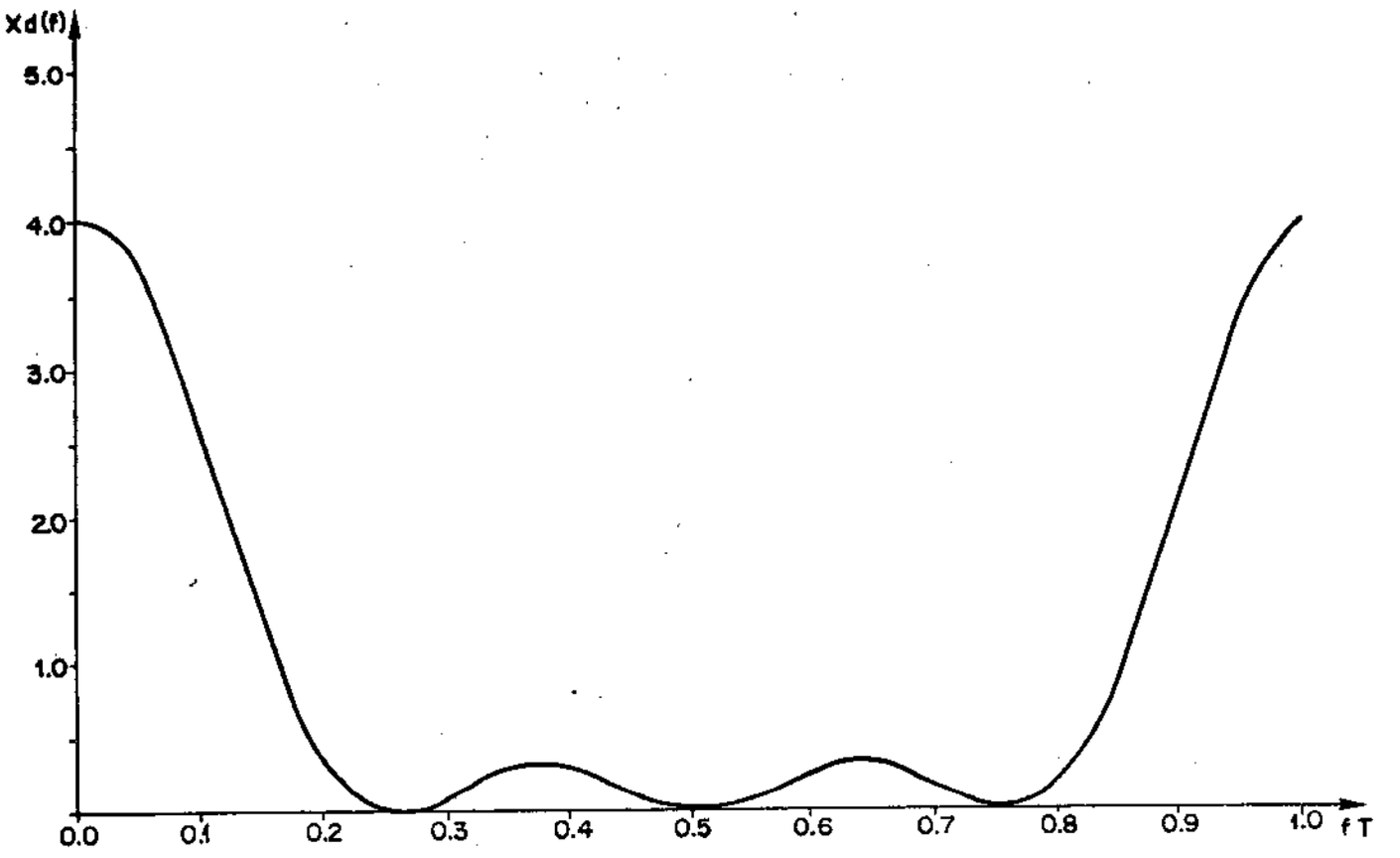


Fig. 5.7b - Componente discreto, normalizado, do espectro de potência do código 3B-4B proposto em [5] para prob(1) = 0,5 As raias estão localizadas nas frequências m/4 , m=0,1, 2,...

7.6 - ALGUNS NOVOS RESULTADOS

O código 3B-4B apresentado na seção 7.5 foi sugerido em [5], dentre um conjunto de outros códigos, como sendo o que apresenta melhores características para transmissão por fibra ótica. Entretanto, num trabalho que vem sendo realizado aqui na UNICAMP e que trata da capacidade dos códigos em recobrarem o sincronismo, visando a monitoração da taxa de erros no sistema, mostrou-se que esse código apresenta alguns problemas a esse respeito em algumas situações. Assim, foram propostos dois novos códigos 3B-4B por nós denominados de 3B-4B* e 3B-4B**. Esses novos códigos sugeridos apresentam uma boa capacidade de recobrar o sincronismo. Entretanto , não se sabia se suas propriedades espectrais satisfaziam às exigências para transmissão por fibra ótica [5]. Assim, através do nosso trabalho foi possível observar suas propriedades espectrais além de procurarmos verificar a sensibilidade dessas propriedades às variações da estatística do sinal a ser codificado.

7.6.1 - CÓDIGO 3B-4B*

Para esse código, o diagrama de estado associado ao codificador é o mesmo da Fig. 7.4. Entretanto, a tabela que caracteriza a função de saída do codificador é dada na tabela 7.2.

u	$\underline{\beta}_u$	$\underline{\alpha}_{iu} = h(s_i, \underline{\beta}_u)$	
		s_1=-1/2	s_2=1/2
1	0 0 0	0 1 0 1	0 1 0 1
2	0 0 1	1 1 0 0	1 1 0 0
3	0 1 0	1 1 1 0	0 1 0 0
4	0 1 1	1 1 0 1	1 0 0 0
5	1 0 0	0 1 1 1	0 0 1 0
6	1 0 1	1 0 1 1	0 0 0 1
7	1 1 0	0 0 1 1	0 0 1 1
8	1 1 1	1 0 1 0	1 0 1 0

Tabela 7.2
Função de saída para o codificador do código 3B-4B*

u	$\underline{\beta}_u$	$\underline{\alpha}_{iu} = h(s_i, \underline{\beta}_u)$	
		s_1=-1/2	s_2=1/2
1	0 0 0	0 0 1 1	0 0 1 1
2	0 0 1	0 1 0 1	0 1 0 1
3	0 1 0	1 0 0 1	1 0 0 1
4	0 1 1	0 1 1 0	0 1 1 0
5	1 0 0	1 0 1 0	1 0 1 0
6	1 0 1	1 1 0 0	1 1 0 0
7	1 1 0	1 1 0 1	0 0 1 0
8	1 1 1	1 0 1 1	0 1 0 0

Tabela 7.3
Função de saída para o codificador do código 3B-4B**

As sub-matrizes $\bar{A}_u$ são dadas por:

$$\bar{A}_1 = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}; \quad \bar{A}_2 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}; \quad \bar{A}_3 = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}; \quad \bar{A}_4 = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix};$$

$$\bar{A}_5 = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}; \quad \bar{A}_6 = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad \bar{A}_7 = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}; \quad \bar{A}_8 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}.$$

As sub-matrizes E_u são as mesmas do código 3B-4B original.

Nas Figs. 7.6a e 7.6b estão representados os componentes contínuo e discreto, normalizados, para p(1)=0,5, p(1) = 0,25 e p(1)=0,75, enquanto que na Fig. 7.6c está representado o espectro de potência do código considerando que os pulsos transmitidos são retangulares,

com um fator de ocupação de 100% para o mesmo conjunto de probabilidade. Isto é,

$$W_c(f)\ /T = \frac{Sinc^2(fT)}{n} \cdot X_c(f) \qquad (7.4)$$

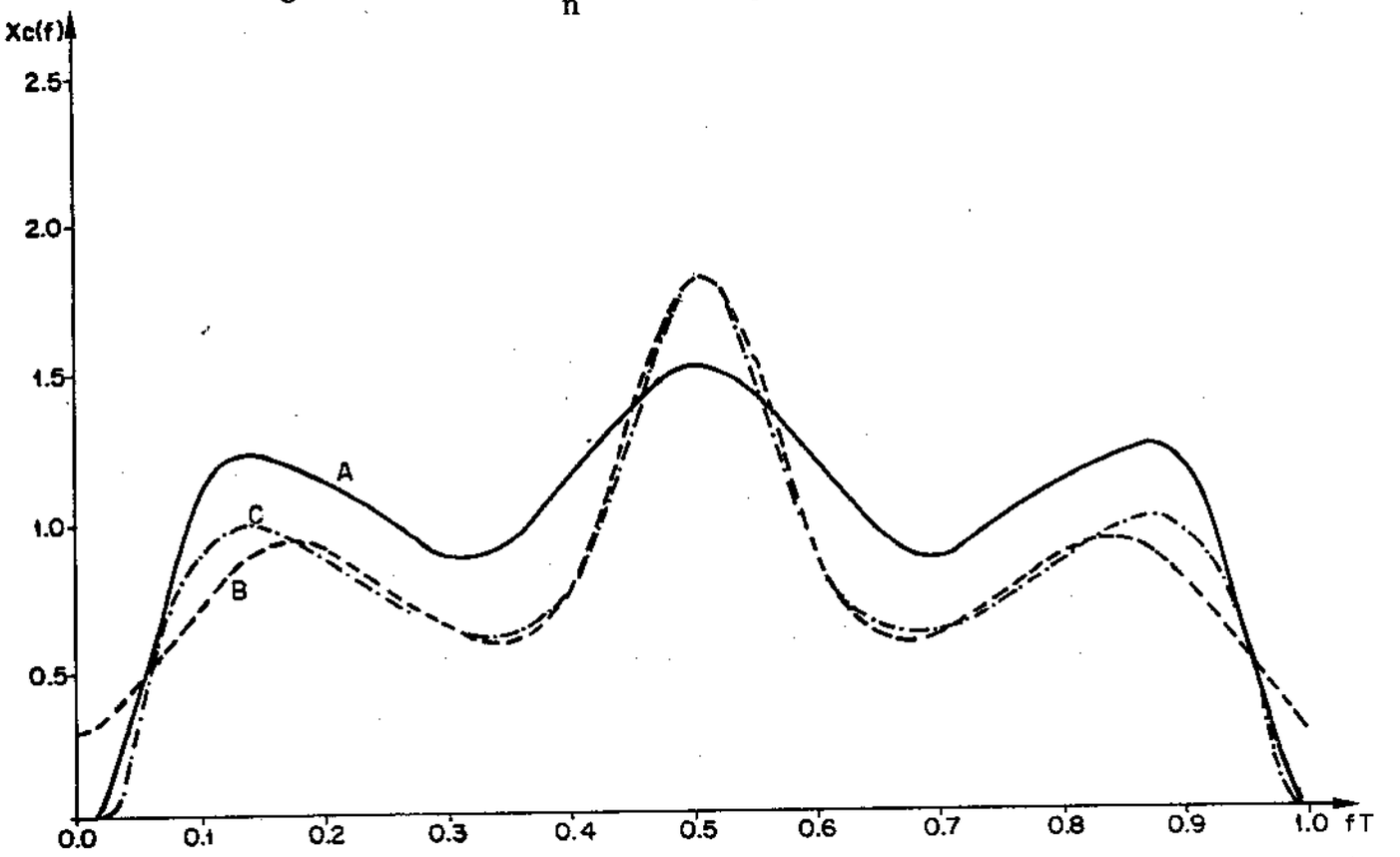


Fig. 7.6a - Componente contínuo do espectro de potência do código 3B-4B*. a)Prob(1)=0,5 b)Prob(1)=0,25 c)Prob(1)=0,75

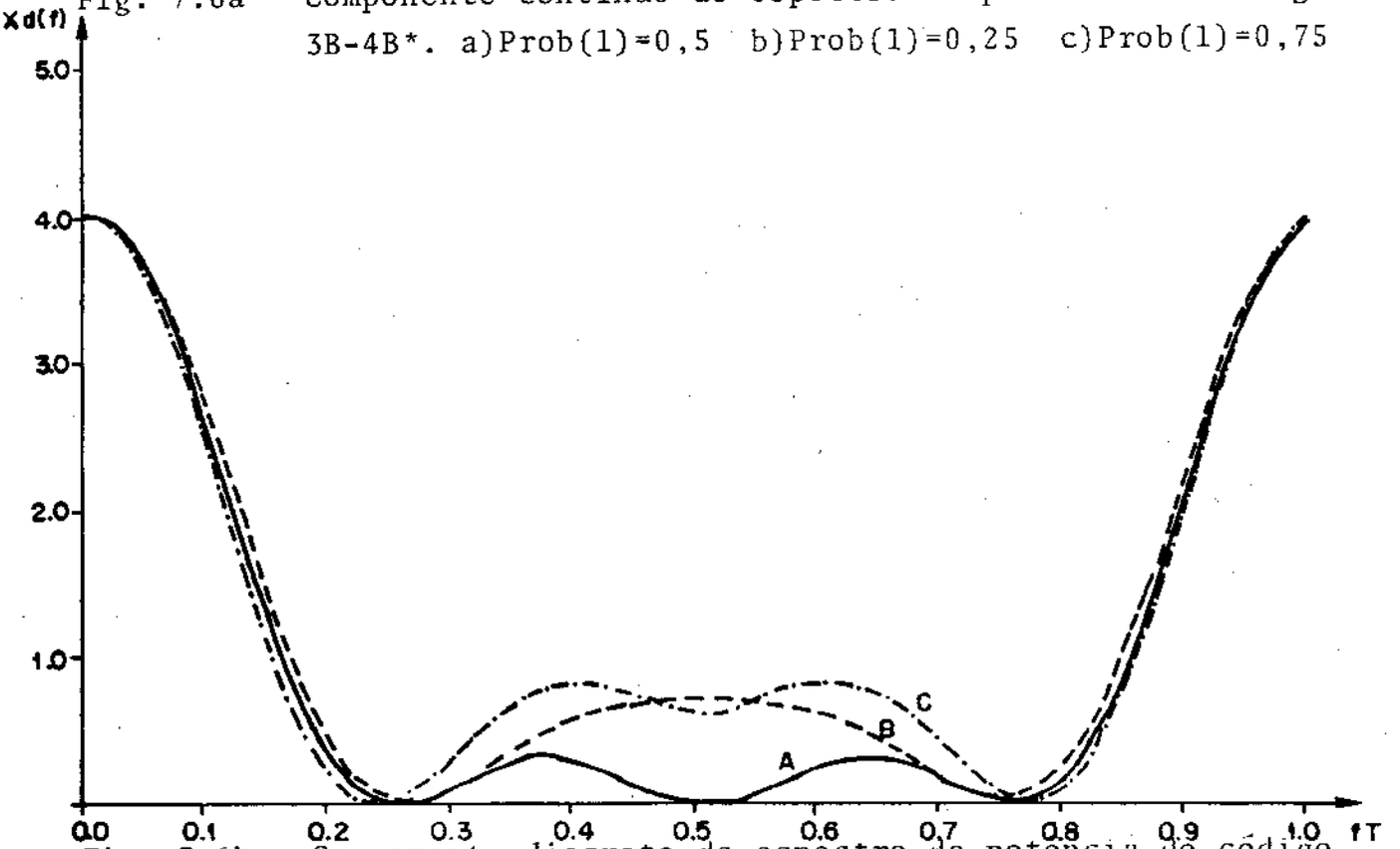


Fig. 7.6b - Componente discreto do espectro de potência do código 3B-4B*. a)Prob(1)=0,5 b)Prob(1)=0,25 c)Prob(1)=0,75

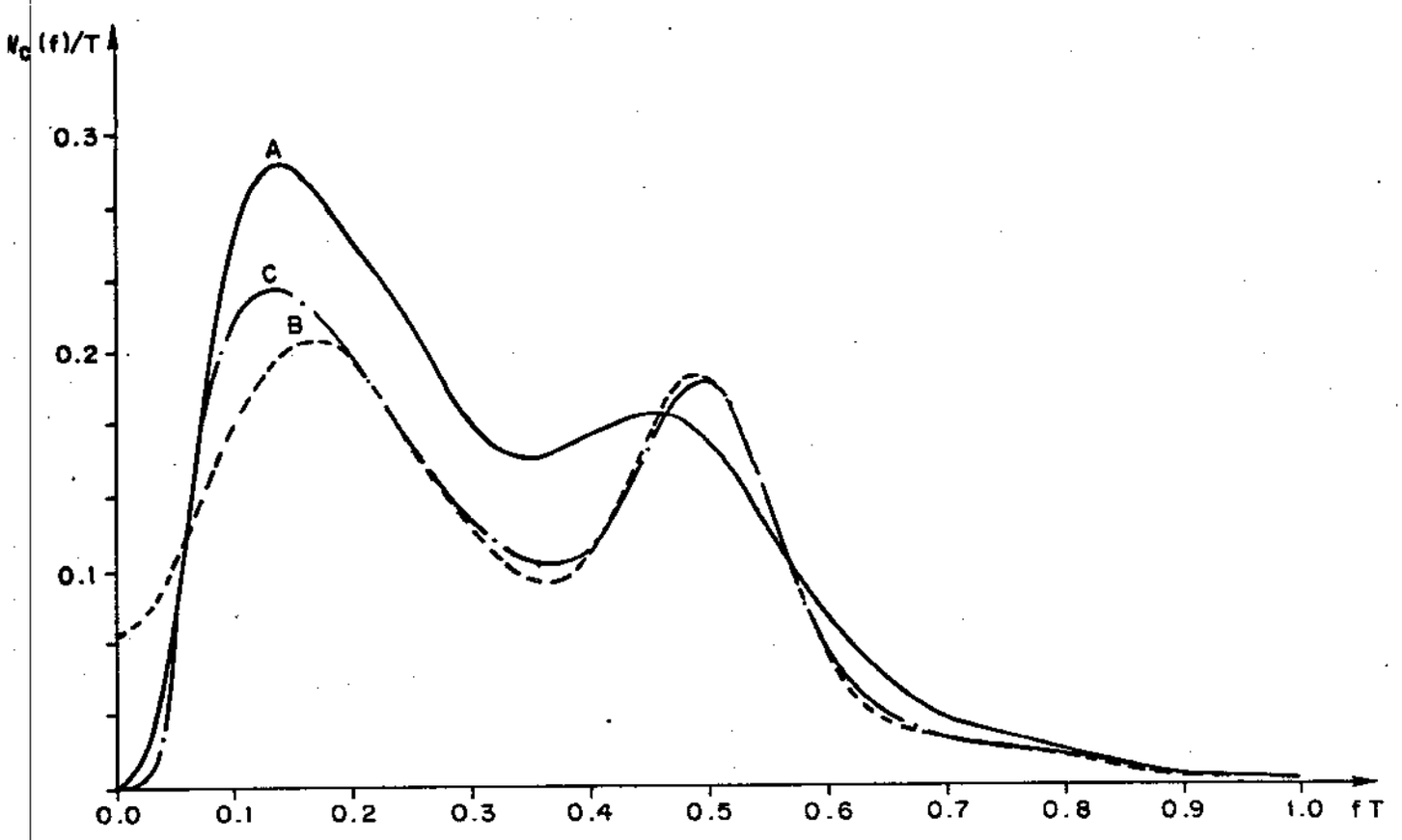


Fig. 7.6c - Componente contínuo , normalizado, do espectro de potência do código 3B-4B* para pulsos retangulares com fator de ocupação de 100%. a)Prob(1) = 0,5 b)Prob(1) = 0,25 c)Prob(1) = 0,75

7.6.2 - CÓDIGO 3B-4B**

Para esse código, a função de saída do codificador é caracterizada pela tabela 7.3, enquanto que o diagrama de estado é ainda dado pela Fig. 7.4. As sub-matrizes $\overline{A}_u$ são dadas por:

$$\overline{A}_1 = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}; \quad \overline{A}_2 = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}; \quad \overline{A}_3 = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix}; \quad \overline{A}_4 = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix};$$

$$\overline{A}_5 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}; \quad \overline{A}_6 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}; \quad \overline{A}_7 = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}; \quad \overline{A}_8 = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

As sub-matrizes $\overline{E}_u$ são dadas por:

$$\overline{E}_1 = \overline{E}_2 = \overline{E}_3 = \overline{E}_4 = \overline{E}_5 = \overline{E}_6 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; \quad \overline{E}_7 = \overline{E}_8 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Nas Figs, 7.7a e 7,7b estão representados os componentes contínuo e discreto, normalizados, para o espectro de potência do código 3B-4B** para Prob (1) no sinal a ser codificado de 0,5, 0,25 e 0,75. Na Fig. 7.7c está representado $W_c(f)/T$ (eq.7.4) para o mesmo conjunto de probabilidades.

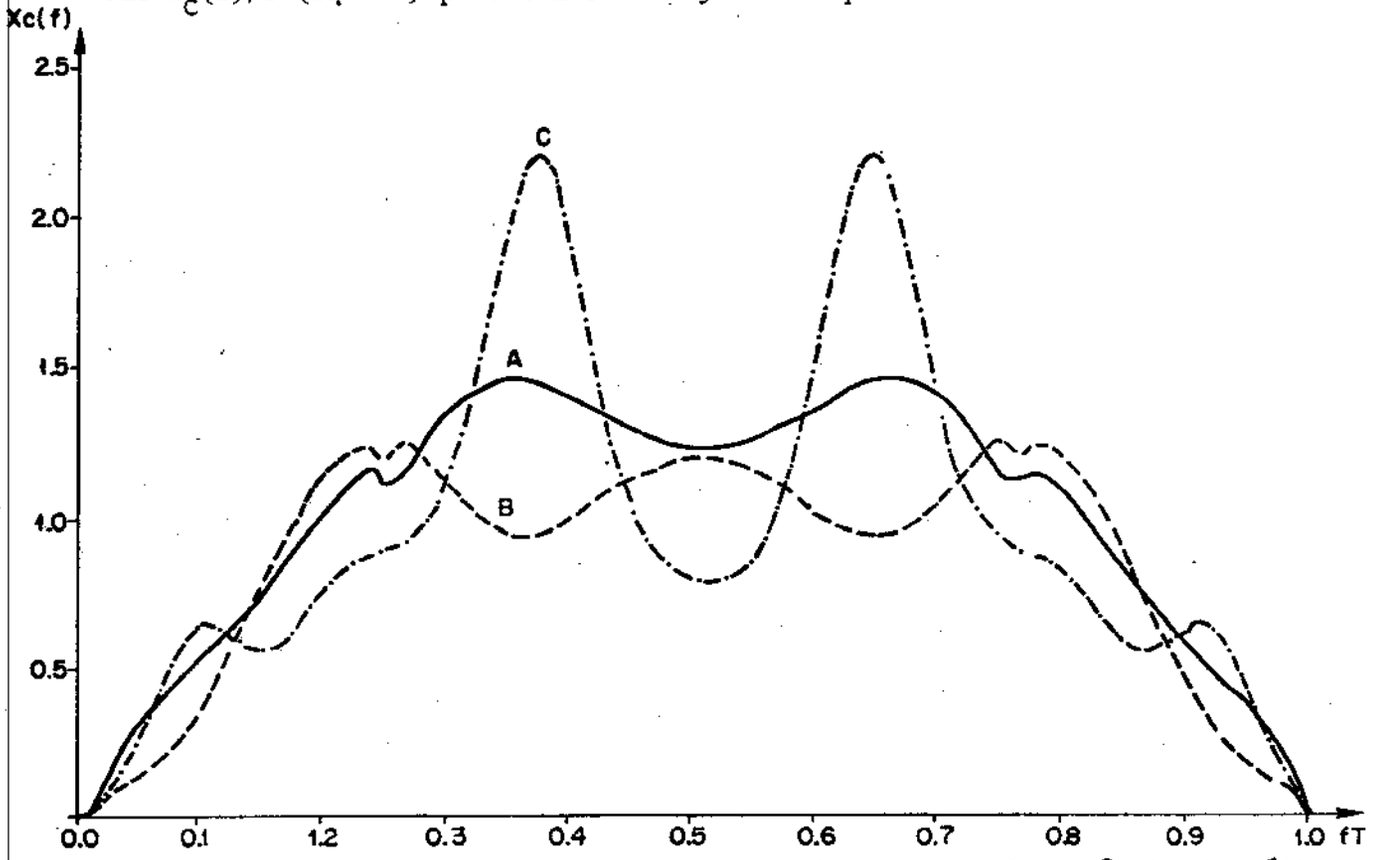


Fig.7.7a - Componente contínuo, normalizado, do espectro de potência do código 3B-4B**. a)Prob(1) = 0,5 b)Prob(1) = 0,25 c)Prob(1) = 0,75.

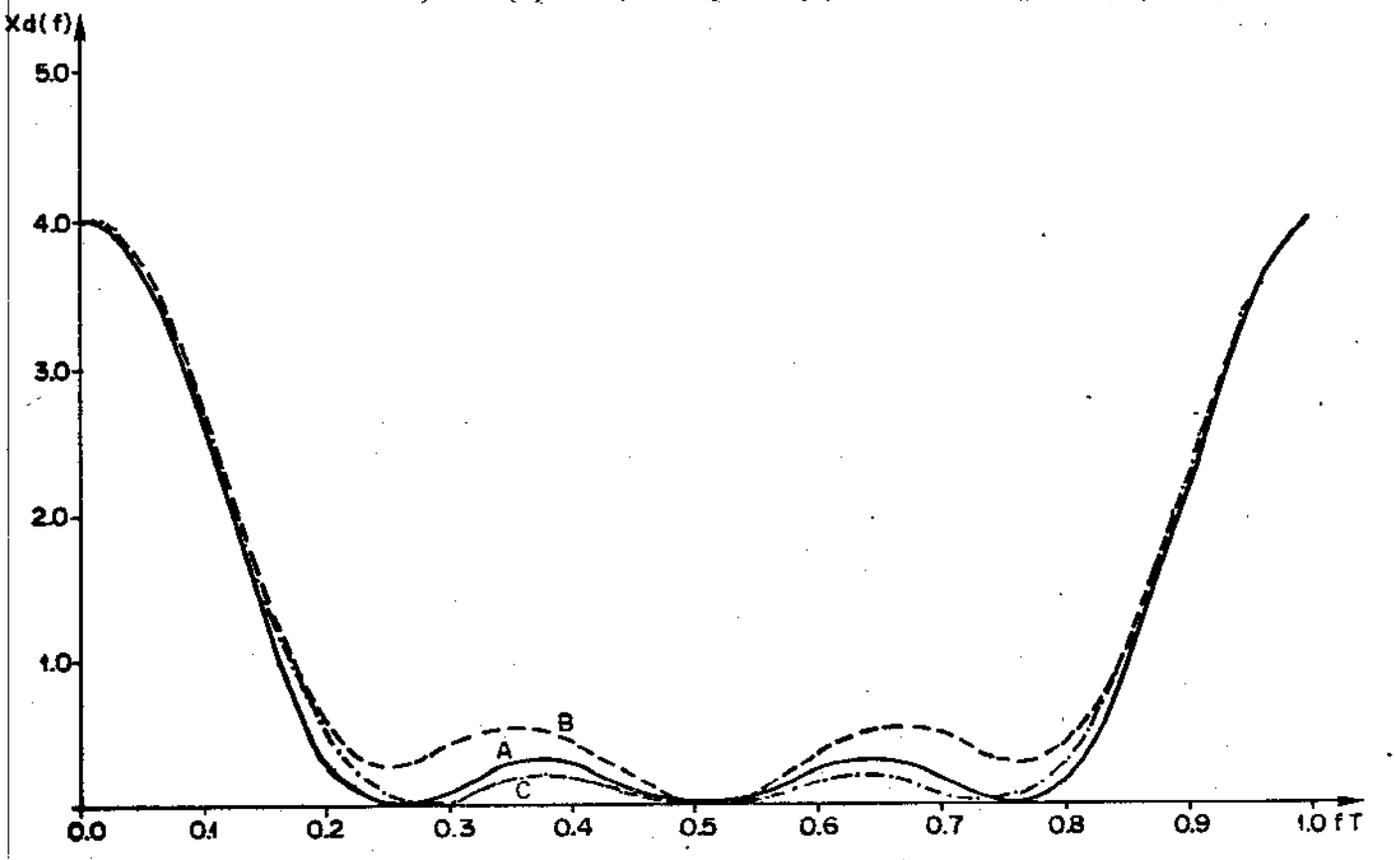


Fig.7.7b - Componente discreto, normalizado, do espectro de potência do código 3B-4B**, a)Prob(1) = 0,5 b)Prob(1) = 0,25 c)Prob(1) = 0,75

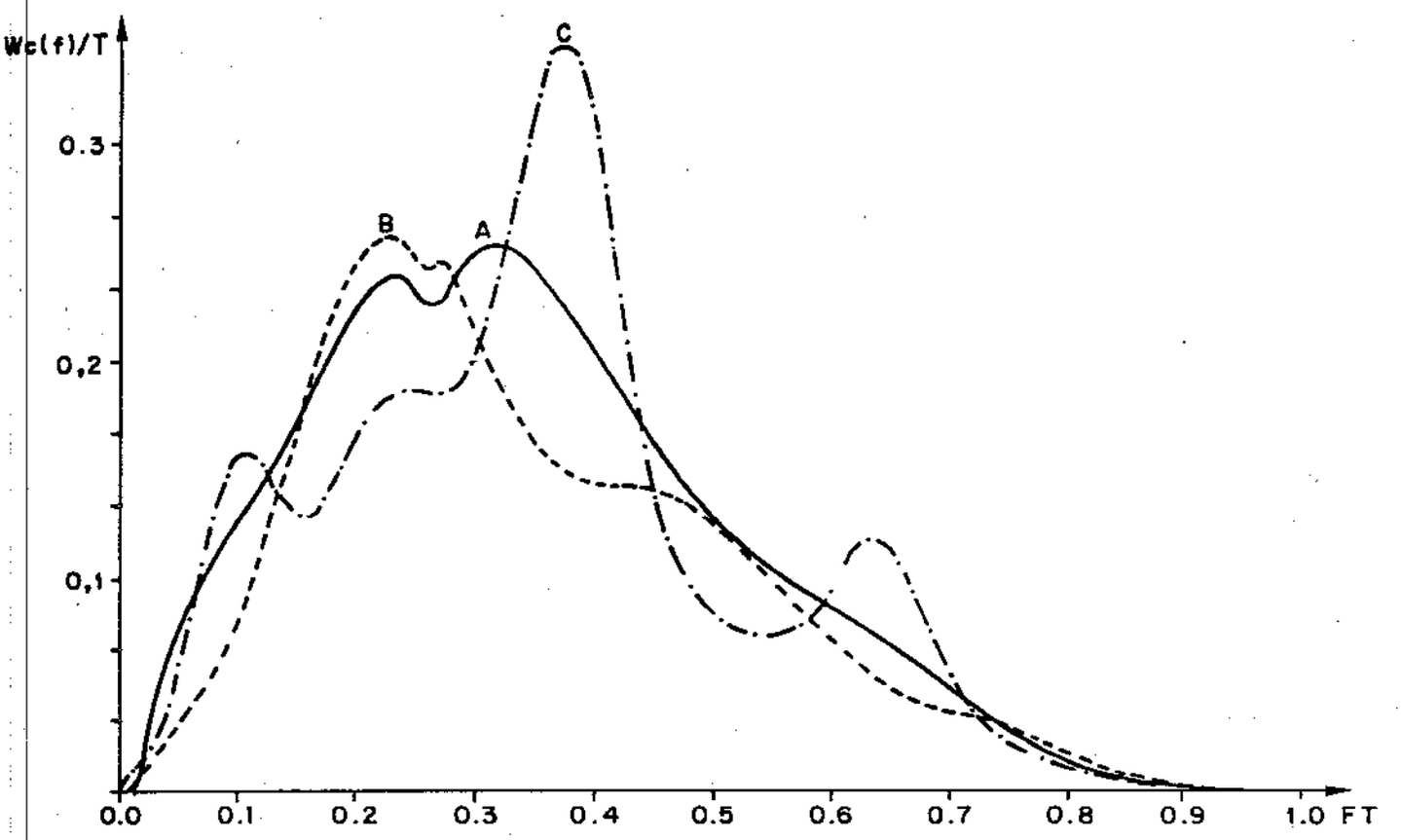


Fig. 7.7c - Componente contínuo, normalizado, do espectro de potência do código 3B-4B** para pulsos retangulares com fator de ocupação de 100%.
a) Prob(1) = 0,5 b) Prob(1) = 0,25 c)Prob(1) = 0,75

7 - OUTROS RESULTADOS

7.7.1 - CÓDIGO 5B-6B

O código 5B-6B é um outro código proposto em [5] para transmissão por fibra ótica. Há um certo interesse na utilização desse código pelo fato dele proporcionar uma economia de faixa de transmissão; enquanto códigos como os 3B-4B exigem 1,33 vezes a faixa do sinal a ser codificado, o código 5B-6B exige 1,2,o que representa uma economia de 13% em faixa para um mesmo sinal a ser codificado. Além desse fato bastante significativo, parece não haver muita diferença na dificuldade de implementação do codificador, decodificador e repetidores para esses códigos. Assim, procuramos verificar qual o comportamento das suas propriedades espectrais às variações da estatística do sinal a ser codificado.

Devido ao número elevado das sub-matrizes $\overline{A}_u$ e $\overline{E}_u$ deixamos de apresentá-las aqui. O diagrama de estado associado ao codificador é o mesmo da Fig. 7.4. A tabela que caracteriza a função de saída para esse código está apresentada em [5]. Nas Figs.7.8a e 7.8b estão os componentes contínuo e discreto, normalizados, para prob(1) no sinal a ser codificado de 0,5, 0,25 e 0,75. Na Fig.7.8c está representado $w_c(f)/T$ dado pela eq. (7.4) para o mesmo conjunto de probabilidades.

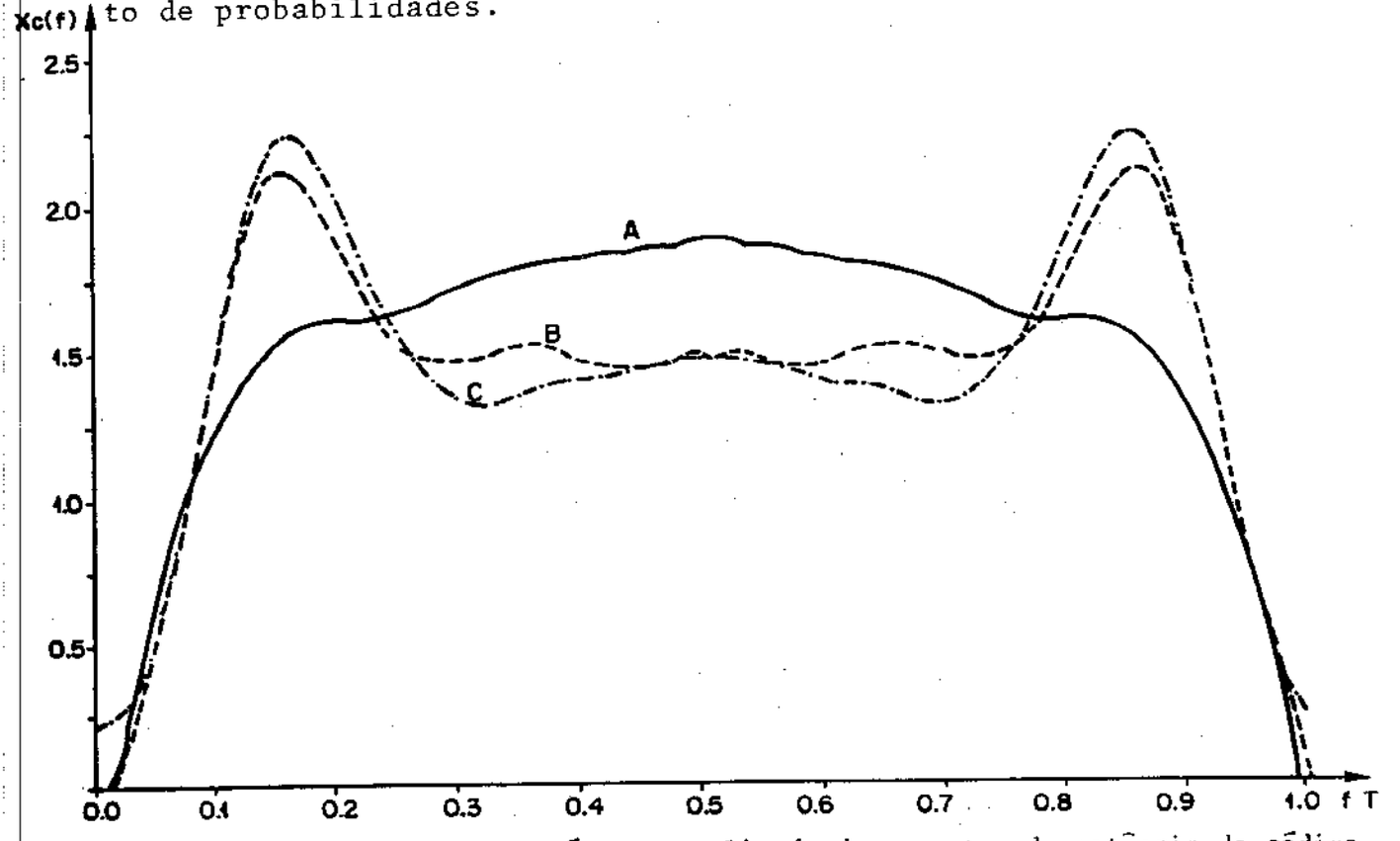


Fig. 7.8a - Componente contínuo,normalizado,do espectro de potência do código 5B-6B. a)Prob(1)=0,5 b)Prob(1)=0,25 c)Prob(1)=0,75

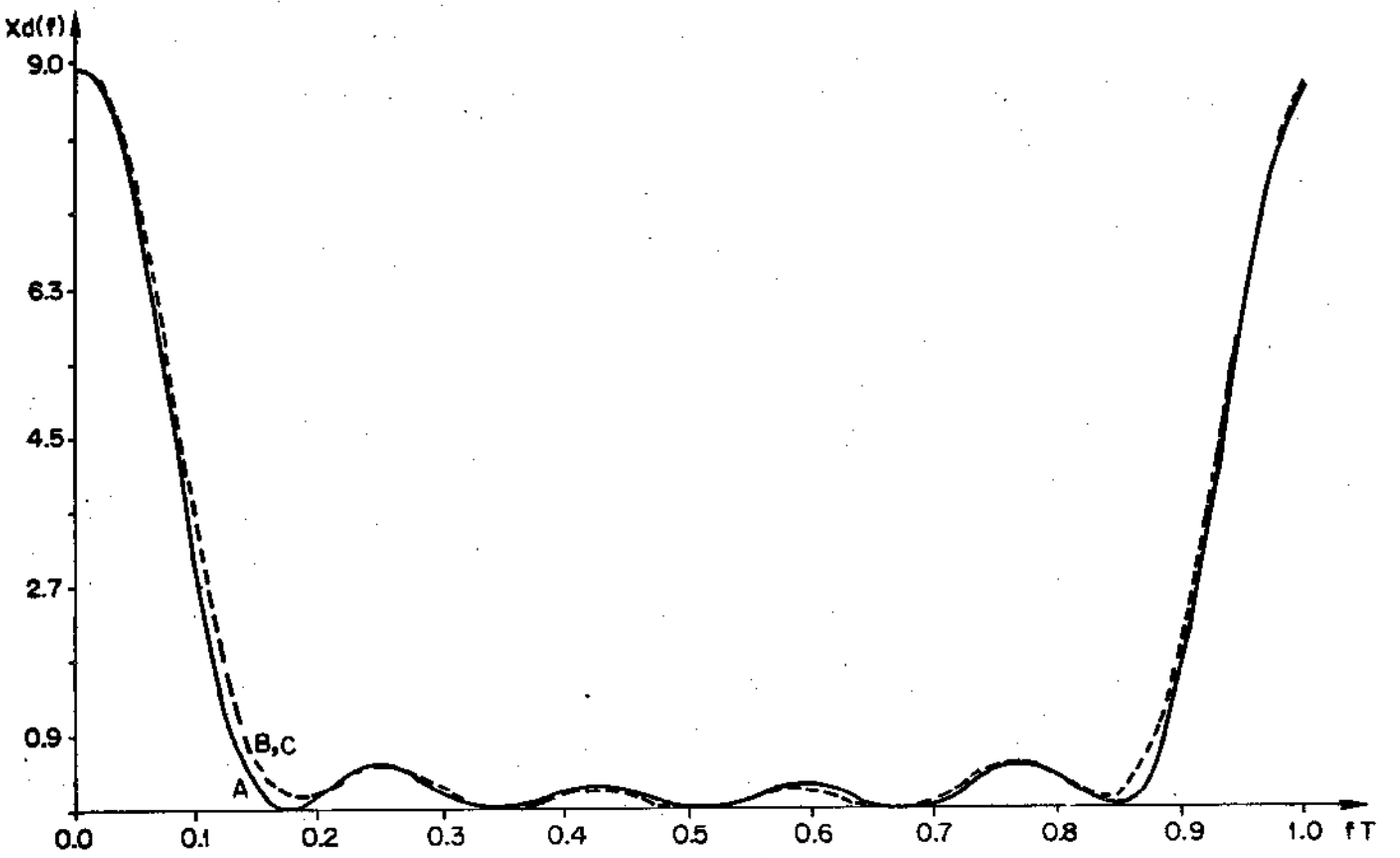


Fig.7.8b - Componente discreto, normalizado, do espectro de potência do código 5B-6B. a) Prob(1) = 0,5 b) Prob(1) = 0,25 c) Prob(1) = 0,75
As raias estão situadas às frequências m/6 , m = 0,1,2,.... .

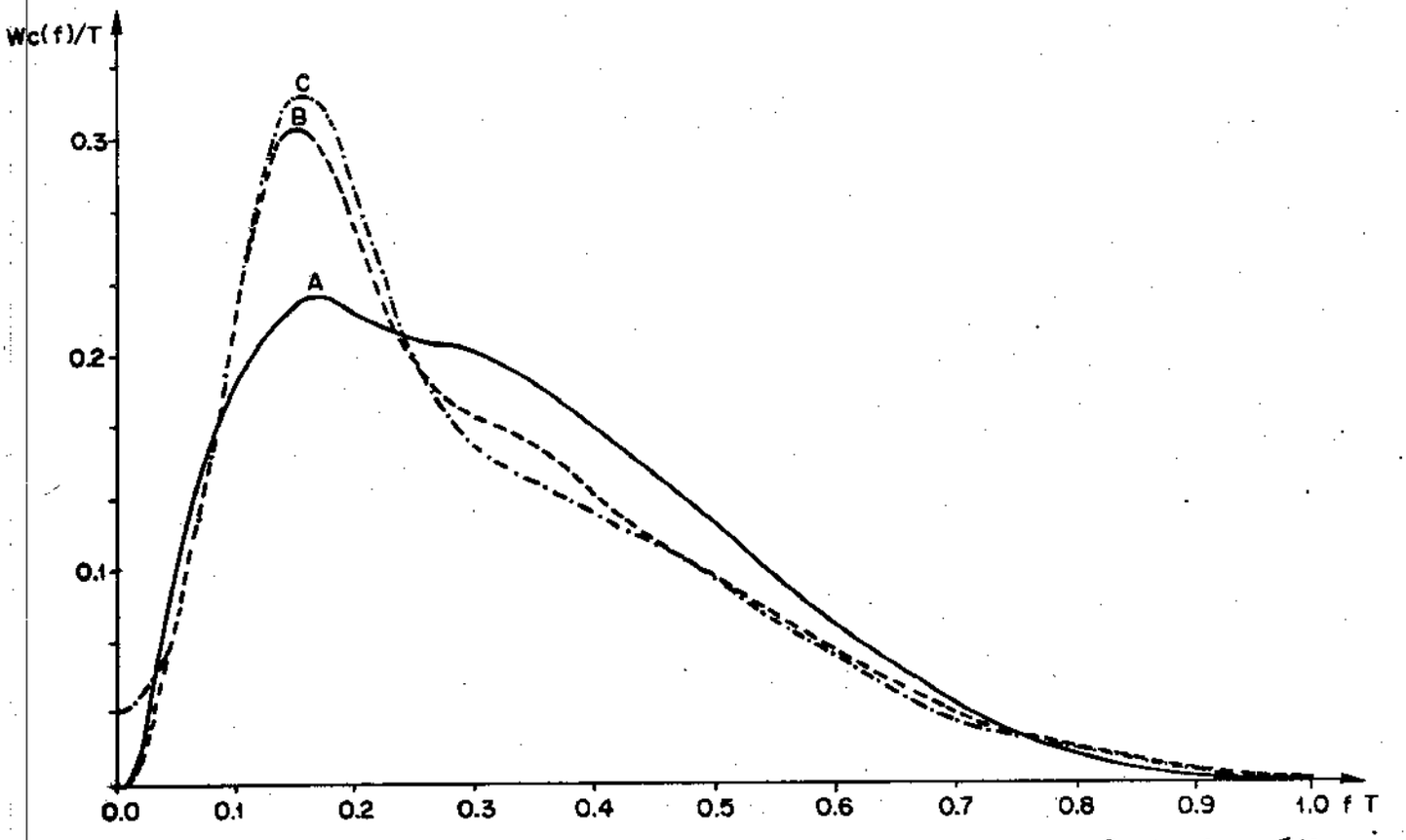


Fig.7.8c - Componente contínuo, normalizado, do espectro de potência do código 5B-6B para pulsos retangulares com fator de ocupação de 100%.
a) Prob(1) = 0,5 b) Prob(1) = 0,25 c) Prob(1) = 0,75

7.7.2 - CÓDIGO DE PETROVIC

Um outro código para transmissão por fibra ótica foi proposto em [6]. A função de saída para o codificador desse código é especificada da seguinte forma:

"1" é codificado por "11" e "00" alternadamente.

"0" é codificado por "10" após a transmissão de "11" ou "01" e por "01" após a transmissão de "00" ou "10".

A expressão para o componente contínuo do espectro de potência do código para pulsos retangulares com fator de ocupação de 100% é dada por [6]:

$$W_c(f) = T/4 \left[\frac{\text{sen}(x/2)}{x/2}\right]^2 \left\{ 1 + 1/3\cos x - \cos 2x - 2/3\cos 3x + \right.$$

$$\left. + 1/3 \left[\frac{(\cos 4x + 2\cos 5x)(2 + \cos 2x) + (\text{sen}4x + 2\text{sen}5x)\text{sen}2x}{(2 + \cos 2x)^2 + \text{sen}^2 2x}\right] \right\} \qquad (7.5)$$

onde $x = 2\pi fT$ e a expressão (7.5) é válida para probabilidade de 1 no sinal a ser codificado de 0,5.

Muitas dúvidas tem sido levantadas com relação ao espectro do sinal codificado por esse código. Nós, em particular, achamos que os resultados obtidos em [6] não são de todo verdadeiros. A justificativa para essa afirmação é que para obter a equação (7.5) o autor usou o resultado obtido por Bennett que, como vimos no Cap. 6, é um caso particular do resultado obtido por Cariolaro para n=1. Assim, procuramos obter o espectro de potência desse código usando o método de Cariolaro, no qual se baseia nossa programação. E mais ainda, estávamos interessados em obter a expressão exata para $W_c(f)$ para podermos comparar com a equação (7.5).

O primeiro passo então foi obter um modelo de máquina sequencial que implementasse a função de saída do código. Inicialmente, conseguimos um modelo com seis estados que em termos computacionais solucionava o nosso problema mas em termos de obter analiticamente $W_c(f)$ era impraticável sob o ponto de vista da quantidade de álgebra envolvida no problema. Posteriormente, nos foi sugerido o modelo que está respresentado na Fig. 7.9a, que facilitou a quantidade de álge

bra envolvida no problema e nos permitiu obter a expressão exata para o componente contínuo do espectro do código, utilizando o método de Cariolaro. A função de saída para o código pode então ser caracterizada pela tabela 7.4.

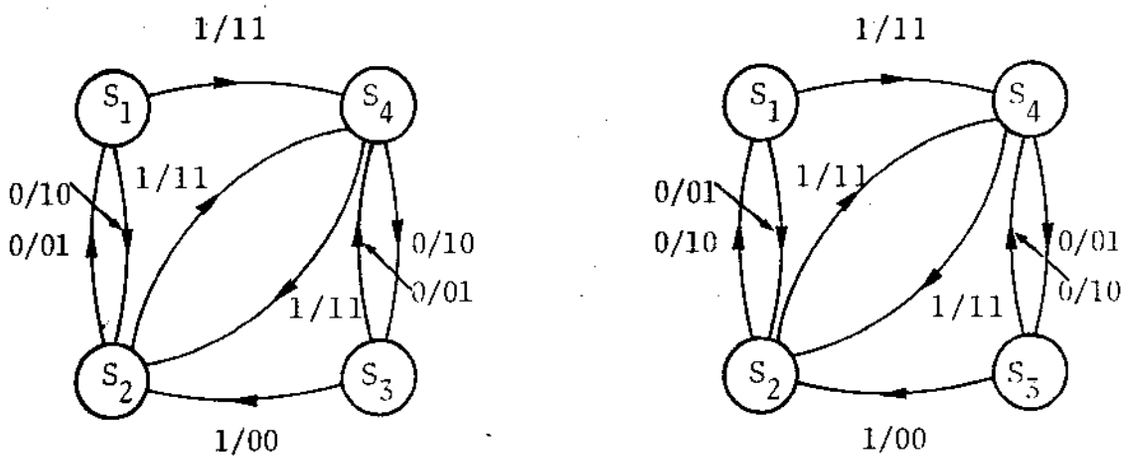


Fig. 7.9 - Modelos de máquinas sequenciais. a) Para o codificador do código de Petrovic. b) Para o codificador do código de Petrovic modificado.

u	$\underset{\sim}{\beta}_u$	$\underset{\sim}{\alpha}_{iu} = h(s_i, \underset{\sim}{\beta}_u)$			
		s_1	s_2	s_3	s_4
1	0	10	01	01	10
2	1	11	11	00	00

Tabela 7.4
Função de saída para o codificador do código de Petrovic

u	$\underset{\sim}{\beta}_u$	$\underset{\sim}{\alpha}_{iu} = h(s_i, \underset{\sim}{\beta}_u)$			
		s_1	s_2	s_3	s_4
1	0	01	10	10	01
2	1	11	11	00	00

Tabela 7.5
Função de saída para o codificador do código de Petrovic modificado

As sub-matrizes $\overline{A}_u$ e $\overline{E}_u$ para o código puderam então ser determinadas como:

$$\overline{A}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \quad ; \quad \overline{A}_2 = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad ; \quad \overline{E}_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad ; \quad \overline{E}_2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Seguindo os passos da resolução do método de Cariolaro e para prob.(1) = 1/2, obtivemos:

$$\underset{\sim}{\overset{\infty}{p}} = 1/6\ (1,2,2,1)$$

$$\overline{R}_0 = 1/4 \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} ; \quad \overline{C}_1^T = 1/12 \begin{bmatrix} 0 & 1 & 2 & 3 \\ 2 & 0 & 0 & 4 \end{bmatrix} ; \quad \overline{C}_2 = 1/2 \begin{bmatrix} 2 & 1 \\ 1 & 2 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} ;$$

$$\overline{R}_\infty = 1/4 \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} ; \quad \overline{\Gamma}_2 = \frac{-1}{72w(4w^2+4w+1)} \begin{bmatrix} 24w^2+36w+12 & 24w^2+12w \\ -24w^2+6 & 48w^2+24w \end{bmatrix} ;$$

$$\overline{\Gamma}_1 = 1/4 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

onde: $w = e^{j2\omega T}$.

Assim,

$$X_1(f) = [v\ v^2]\ \overline{\Gamma}_1 \begin{bmatrix} v^{-1} \\ v^{-2} \end{bmatrix} = 1/2.$$

$$2R_e\ [X_2(f)] = 2R_e \left\{ [v\ v^2]\ \overline{\Gamma}_2 \begin{bmatrix} v^{-1} \\ v^{-2} \end{bmatrix} \right\} =$$

$$= -1/6 \left[\frac{58-9\cos\omega T+102\cos 2\omega T+24\cos 3\omega T+48\cos 4\omega T+12\cos 5\omega T+8\cos 6\omega T}{33+40\cos 2\omega T+8\cos 4\omega T} \right],$$

onde: $v = e^{j\omega T}$.

Portanto,

$$X_c(f) = 1/2 \left\{ 1 - 1/3 \left[\frac{58-9\cos\omega T+102\cos 2\omega T+24\cos 3\omega T+48\cos 4\omega T+12\cos 5\omega T+8\cos 6\omega T}{33+40\cos 2\omega T+8\cos 4\omega T} \right] \right\}. \tag{7.6}$$

Considerando-se pulsos retangulares com fator de ocupação de 100%, temos:

$$W_c(f)=T/4\left[\frac{\operatorname{sen}(\omega T/2)}{\omega T/2}\right]^2\left\{1-1/3\left[\frac{58-9\cos\omega T+102\cos 2\omega T+24\cos 3\omega T+48\cos 4\omega T+12\cos 5\omega T+8\cos 6\omega T}{33+40\cos 2\omega T+8\cos 4\omega T}\right]\right\} \tag{7.7}$$

O componente discreto é dado por:

$$X_d(f) = [v\ v^2]\ \bar{R}_\infty \begin{bmatrix} v^{-1} \\ v^{-2} \end{bmatrix} = 1/2\ (1 + \cos\omega T) \tag{7.8}$$

Finalmente, temos para um pulso s(t) qualquer:

$$W(f)= \frac{|S(f)|^2}{4T}\left\{1-1/3\left[\frac{58-9\cos\omega T+102\cos 2\omega T+24\cos 3\omega T+48\cos 4\omega T+12\cos 5\omega T+8\cos 6\omega T}{33+40\cos 2\omega T+8\cos 4\omega T}\right] + \right.$$

$$\left. + \frac{(1 + \cos\omega T)}{2T} \sum_{k=-\infty}^{+\infty} \delta(f - k/2T)\right\} \tag{7.9}$$

e esta é a expressão exata para a densidade espectral de potência do sinal digital codificado pelo código de Petrovic.

Com o fim de melhor compararmos os resultados obtidos pelos dois métodos, escreveremos a equação (7.5) na forma:

$$W_c(f)=T/4\left[\frac{\operatorname{sen}(x/2)}{x/2}\right]^2\left\{1 + 1/3\cos x - \cos 2x - 2/3\cos 3x + \right.$$

$$\left. + 1/3\left[\frac{\cos 2x + 2\cos 3x + 2\cos 4x + 4\cos 5x}{5 + 4\cos 2x}\right]\right\}. \tag{7.10}$$

Na Fig. 7.10 estão representados os gráficos de $W_c(f)$ dados pelas equações (7.7) e (7.10) em função da frequência normalizada fT_0, T_0 sendo o período de símbolos no sinal a ser codificado e relacionado com T através da expressão:

$$T = k/n\ T_0 \tag{7.11}$$

onde n e k são os parâmetros do código (n,k). Como podemos observar, os gráficos se superpõem e assim pode-se concluir que, nesse caso, em termos de resultados, os métodos de Cariolaro e de Bennett são equivalentes. Entretanto, como pode-se verificar em [1], as suposições

impostas por Bennett à estatística do sinal a ser codificado são bem mais fortes que as impostas por Cariolaro.

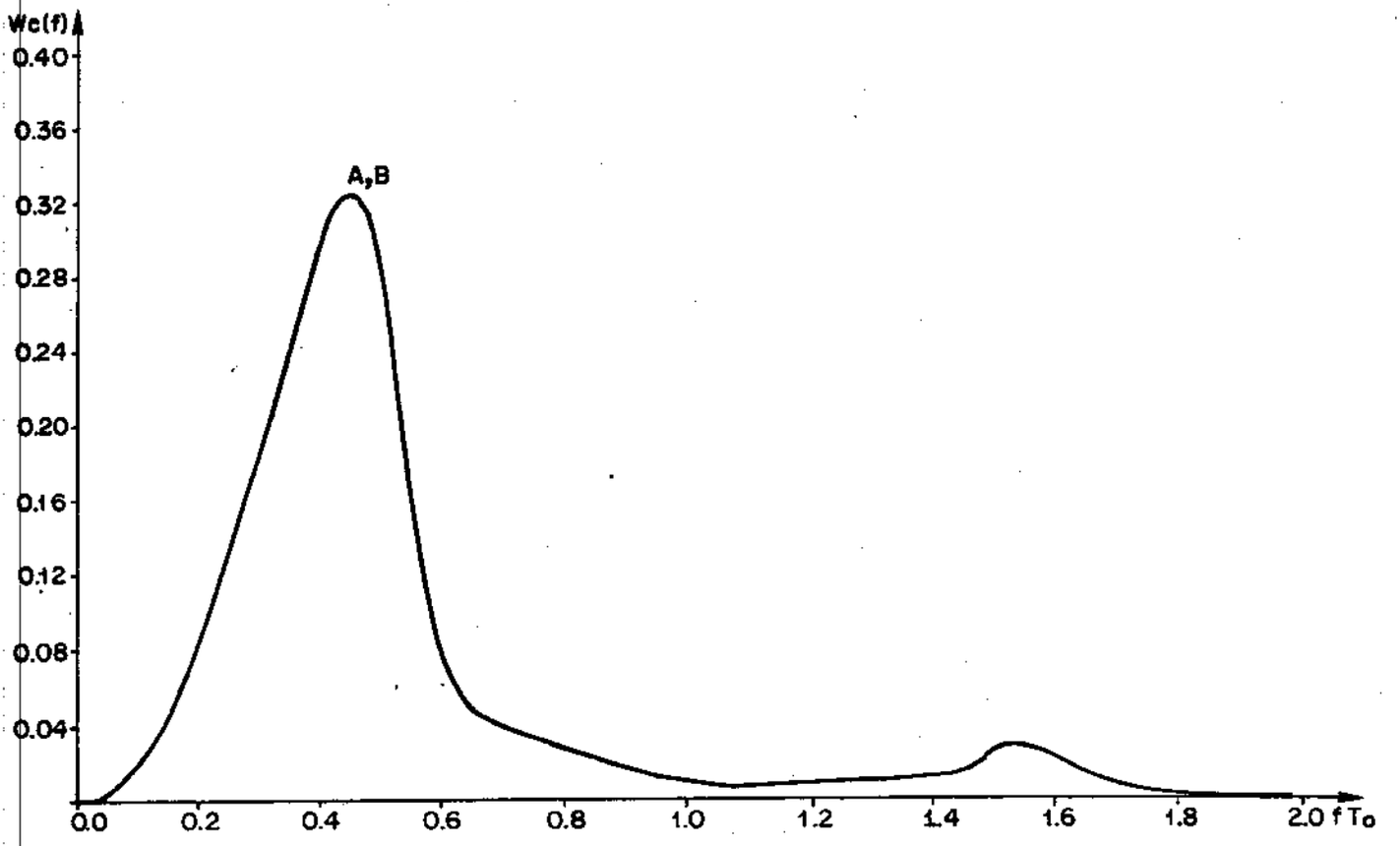


Fig. 7.10 - Gráficos de $W_c(f)$ em função da frequência normalizada fT_0. a) Resultado original
b) Novo resultado

7.7.3 - CÓDIGO DE PETROVIC MODIFICADO

Nas pesquisas que vêem sendo realizadas na TELEBRÁS, com o fim de implementar um sistema de comunicações do tipo PCM para transmissão por fibra ótica, um dos códigos que tem recebido atenção, devido à simplicidade de implementação do codificador e decodificador [6], é o código de Petrovic. Entretanto, numa tentativa de melhorar as propriedades espectrais do código com respeito ao seu conteúdo espectral nas baixas frequências, visando minimizar a interferência de intersímbolos devido à supressão de componentes de baixa frequência durante a transmissão, foi proposta a seguinte modificação:

> "0" é transmitido por "01" após a transmissão de "11" ou "10" e é transmitido por "10" após a transmissão de "00" ou "01".

Na Fig. 7.11 estão representados os sinais digitais codificados correspondentes a uma mesma sequência de informação. À primeira vista, nos parece que a modificação proposta suprime bastante os componentes de baixa frequência, além de melhorar a extração do relógio, já que o número de transições é aumentado consideravelmente.

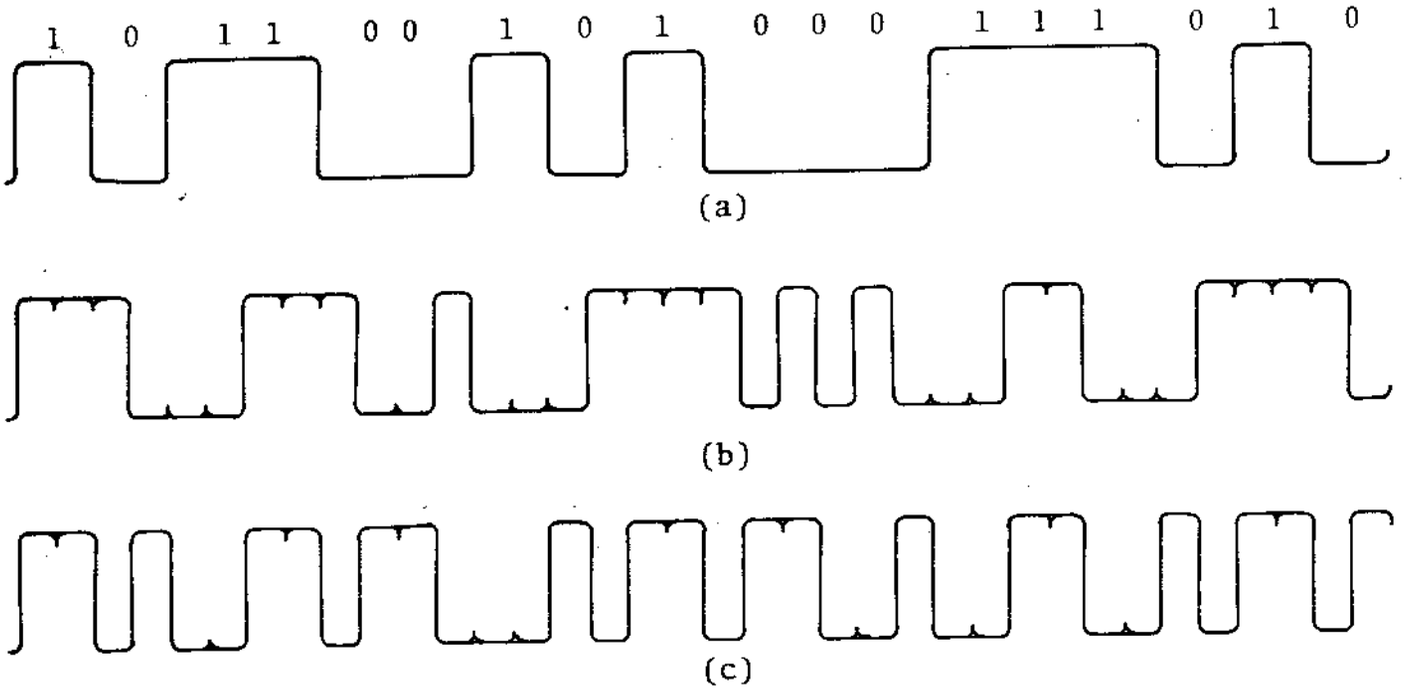


Fig. 7.11 - Sinais digitais codificados, a) Sequência a ser codificada. b) Sequência codificada utilizando o código de Petrovic original. c) Sequência codificada utilizando o código de Petrovic modificado.

O diagrama de estado correspondente ao novo código está representado na Fig. 7.9b e a função de saída do codificador está caracterizada pela tabela 7.5. Como podemos observar, apenas a sub-matriz $\bar{A}_1$ sofreu alterações com as mudanças introduzidas, passando a ser dada por:

$$\bar{A}_1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Repetindo a mesma sequência de cálculos efetuada anteriormente obtivemos:

$$X_c(f) = 1/2 \left\{ 1-1/3 \left[\frac{58+22\cos\omega T+102\cos 2\omega T-3\cos 3\omega T+48\cos 4\omega T+4\cos 5\omega T+8\cos 6\omega T+4\cos 7\omega T}{33+40\cos 2\omega T+8\cos 4\omega T} \right] \right\} \tag{7.12}$$

$$X_d(f) = 1/2\ (1 + \cos\omega T) \tag{7.13}$$

e assim,

$$W(f) = \frac{|S(f)|^2}{4T} \left\{ 1-1/3 \left[\frac{58+22\cos\omega T+102\cos 2\omega T-3\cos 3\omega T+48\cos 4\omega T+4\cos 5\omega T+8\cos 6\omega T+4\cos 7\omega T}{33+40\cos 2\omega T+8\cos 4\omega T} \right] + \right.$$

$$\left. + \frac{(1 + \cos\omega T)}{2T} \sum_{k=-\infty}^{+\infty} \delta(f - k/2T) \right\} . \tag{7.14}$$

Na Fig. 7.12 estão representados os componentes contínuos normalizados para os dois códigos. Como podemos observar, os componentes de baixa frequência são fortemente suprimidos com o novo código. Entretanto, o comportamento em torno da frequência de Nyquist permaneceu o mesmo para os dois códigos.

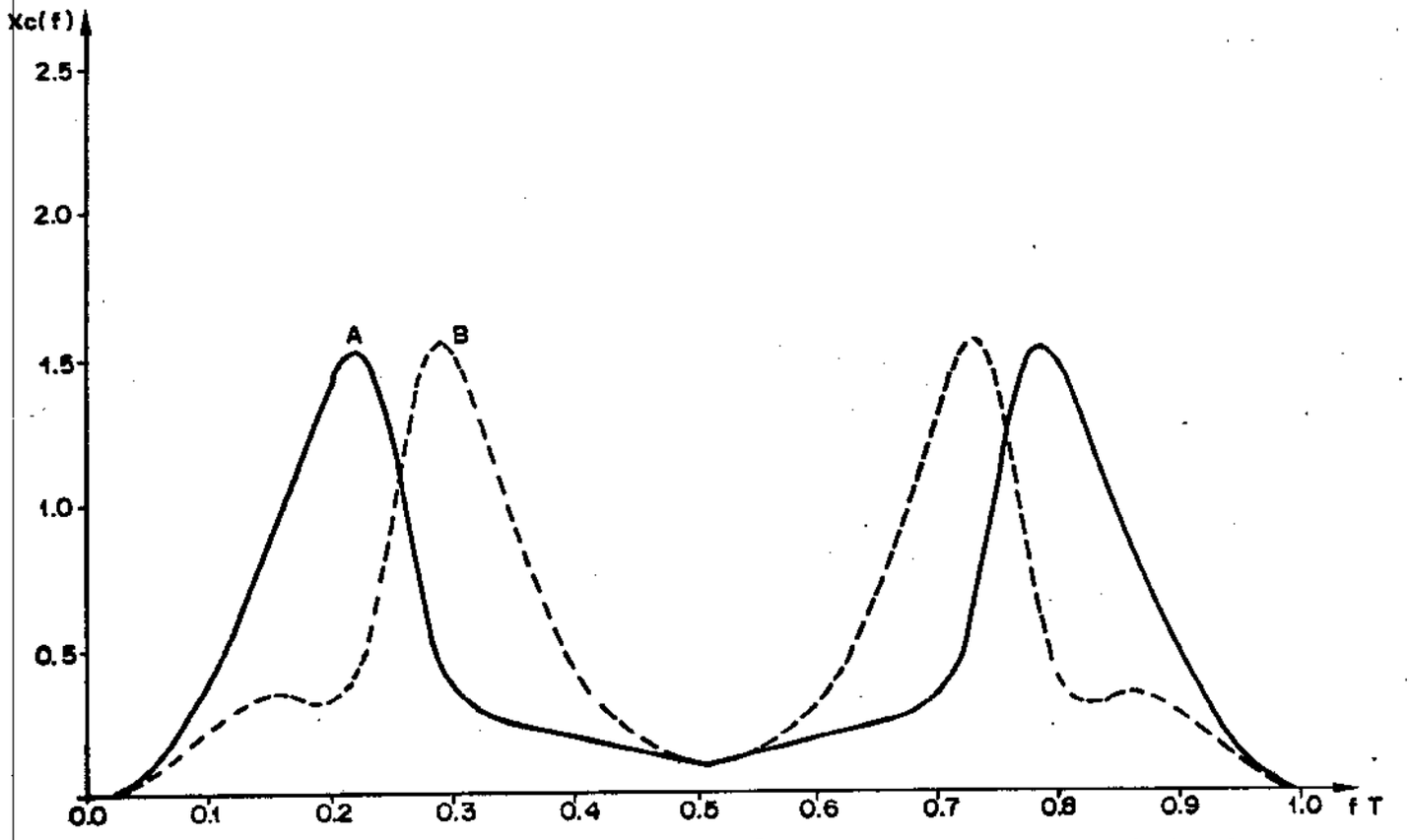


Fig.7.12 - $X_c(f)$ normalizado. a) Código de Petrovic original.
b) Código de Petrovic modificado.

.8 - COMPARAÇÃO ENTRE CÓDIGOS

Uma outra característica da nossa programação é que se pode determinar, a partir dos dados de saída e com o uso das equações (6.93), (6.94) e (6.106), expressões aproximadas para $X_c(f)$ e $X_d(f)$, e portanto para $W_c(f)$. A precisão dessas expressões está relacionada com a precisão das probabilidades dos vetores de informação e com a precisão da solução do sistema de equações (6.53). Para alguns códigos, chegamos a obter resultados exatos para prob(1)=0,5.

Quando se deseja comparar as propriedades espectrais de códigos diferentes, os gráficos de $X_c(f)$, normalizado não são os mais eficientes, a não ser que os códigos a serem comparados tenham a mesma taxa de informação k/n. Assim, com o fim de compararmos códigos com taxas de informação diferentes, obtivemos as expressões de $X_c(f)$ a partir dos dados de saída da nossa programação e implementamos um programa para plotar os gráficos de $W_c(f)$ para os diferentes códigos.

Inicialmente, faremos uma comparação entre os códigos 3B-4B. Como eles possuem a mesma taxa, usaremos os gráficos de $X_c(f)$ normalizado para a comparação.

7.8.1 - CÓDIGOS 3B-4B

Na Fig. 7.13 estão representados os componentes contínuos, normalizados, dos espectros de potência dos códigos 3B-4B , 3B-4B* e 3B-4B**. Como podemos observar, nas baixas frequências o código que apresenta melhores características para transmissão por fibra ótica é o código 3B-4B**. Entretanto, em torno da frequência de Nyquist o código 3B-4B** perde em desempenho para os códigos 3B-4B e 3B-4B* que, nessa frequência, apresentam máximos absolutos. Assim, num sistema de transmissão em que não seja necessário equalização, o código 3B-4B** deve ser o preferido [5]. Entretanto, se for utilizado equalização, o código 3B-4B* será o preferido, já que o 3B-4B original apresenta problemas de sincronismo.

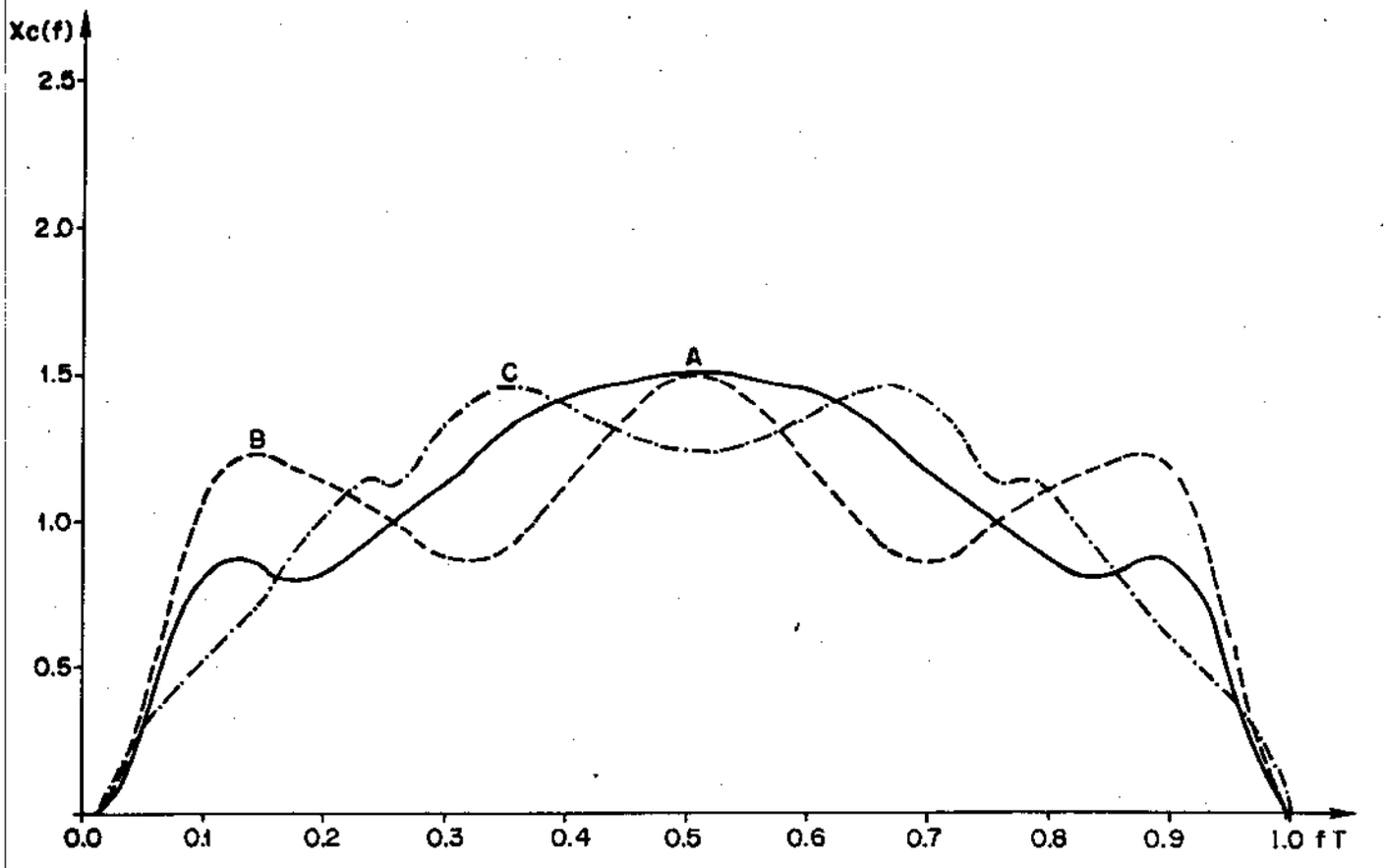


Fig. 7.13 - Componente contínuo, normalizado, para os códigos a) 3B-4B b) 3B-4B* c) 3B-4B**

7.8.2 - 3B-4B** , 5B-6B E PETROVIC MODIFICADO

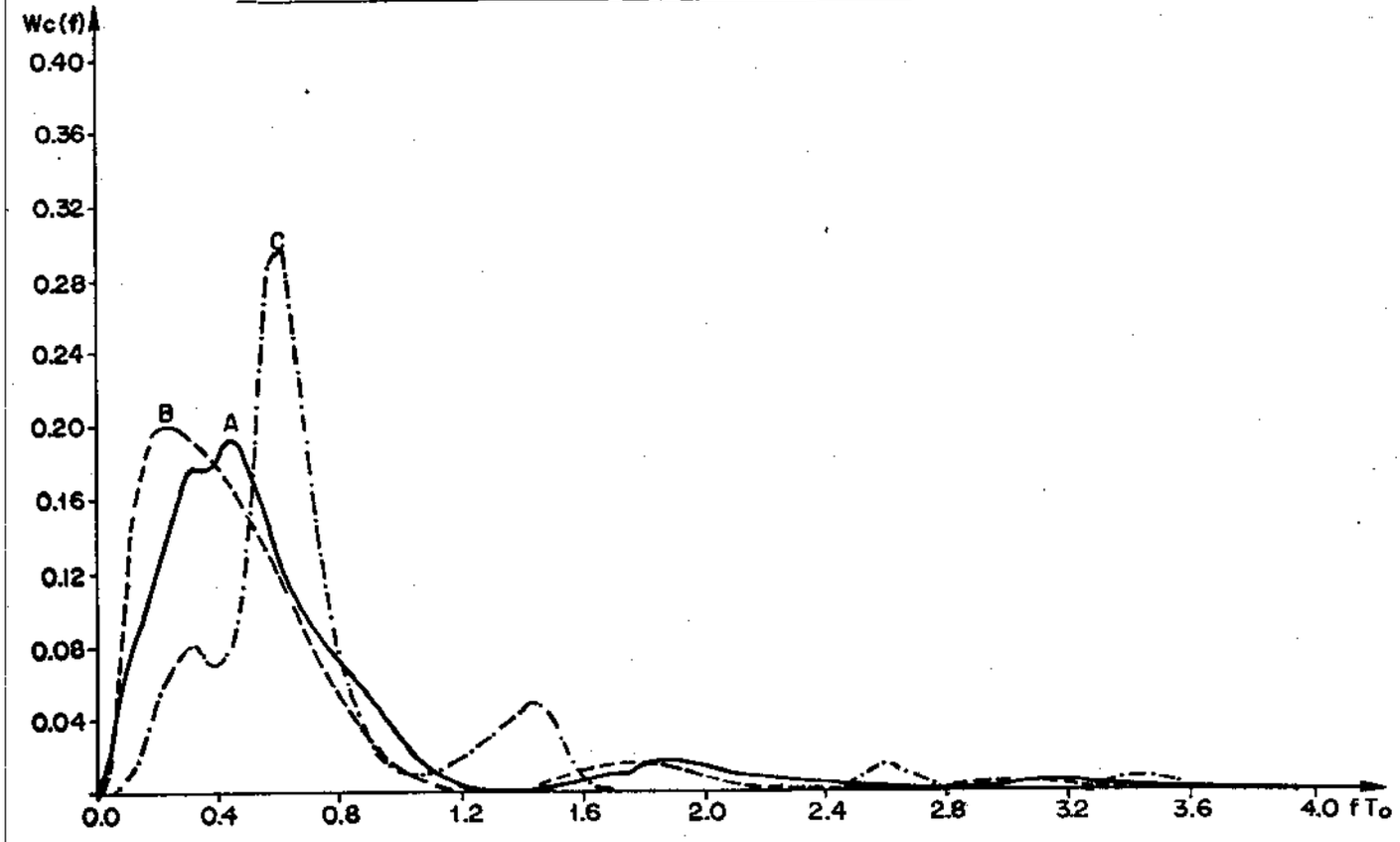


Fig.7.14 - Componente contínuo $W_c(f)$ em função da frequência normalizada fT_0 para pulsos retangulares com fator de ocupação de 100% dos códigos: a) 3B-4B** b) 5B-6B c) Petrovic modificado.

Como podemos observar pela Fig. 7.14, é o código de Petrovic modificado que apresenta melhor desempenho nas baixas frequências, de modo que se não houver problemas de economia de faixa de transmissão, será esse o código escolhido. Entretanto , se houver necessidade de equalização e economia de faixa, o código 5B-6B será o preferido, supondo-se que as dificuldades de implementação do codificador e decodificador sejam as mesmas para os códigos 3B-4B** e 5B-6B.

7.9 - ESPECTRO DE UM CÓDIGO CONVOLUCIONAL

O código convolucional do exemplo 7 (Cap.6) está na forma sistemática. Naquele exemplo, utilizamos o codificador do $(m-1)n_0$ estágios para obtermos as matrizes constituintes do codificador.Se utilizarmos como codificador o modelo de $(m-1)(n_0-k_0)$ estágios, válido apenas para códigos sistemáticos , obteremos como matrizes constituintes do codificador,

$$\overline{A} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \quad ; \quad \overline{B} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

$$\overline{C} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \quad ; \quad \overline{D} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Nesse caso, o número de memórias do codificador é 2. Nas Figs. 7.15a, 7.15b estão representados os componentes contínuo e discretos, normalizados, para prob.(1) = 0,5. Na Fig. 7.15c está representado o componente contínuo, normalizado, para pulsos retangulares com fator de ocupação de 100% e prob(1) = 0,5. Observemos pela Fig. 7.15a que o codificador para esse código convolucional se comporta como uma fonte de ruído branco.

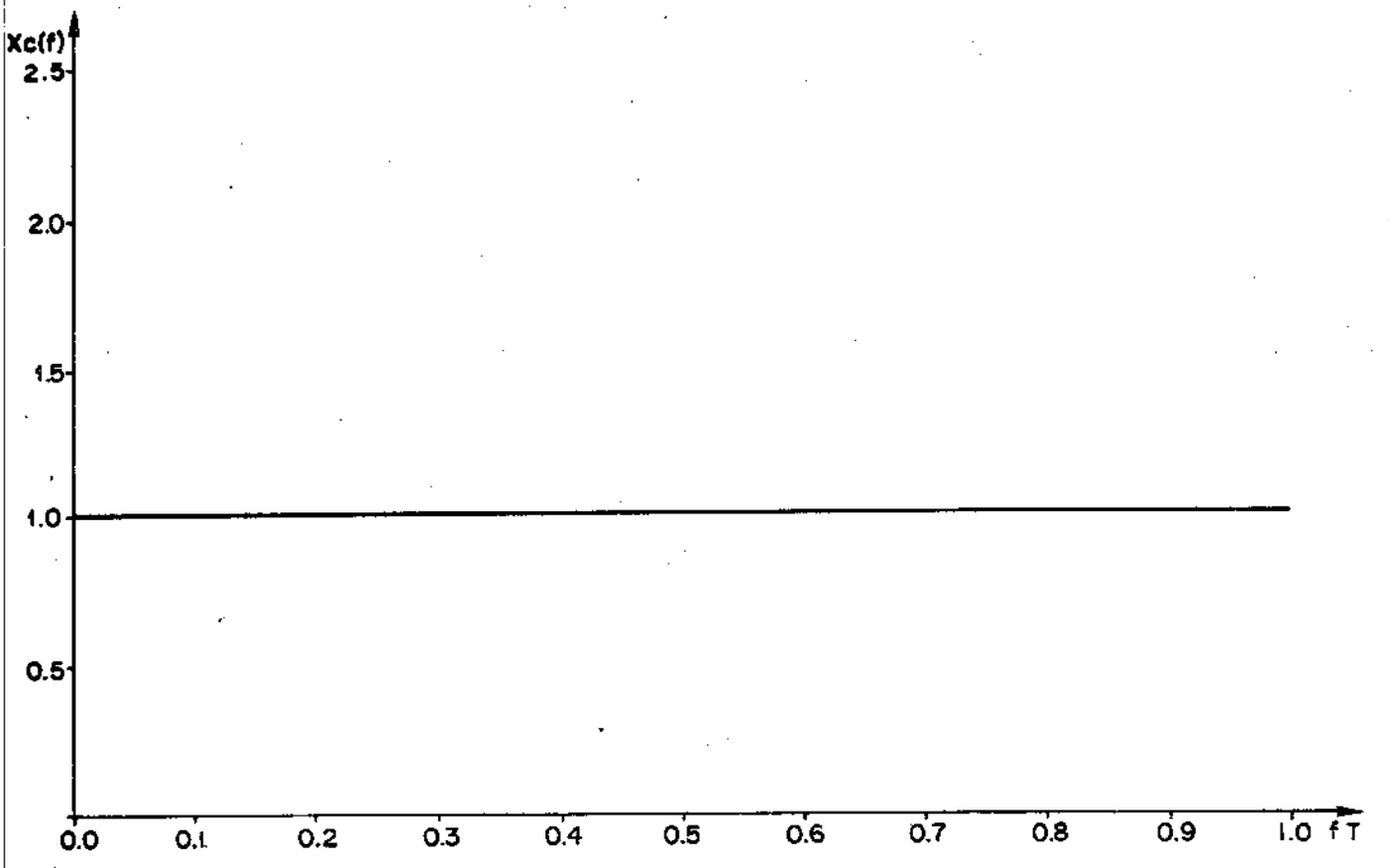


Fig. 7.15a - Componente contínuo, normalizado, do espectro de potência do código convolucional sistemático (12,9) para prob.(1) = 0,5.

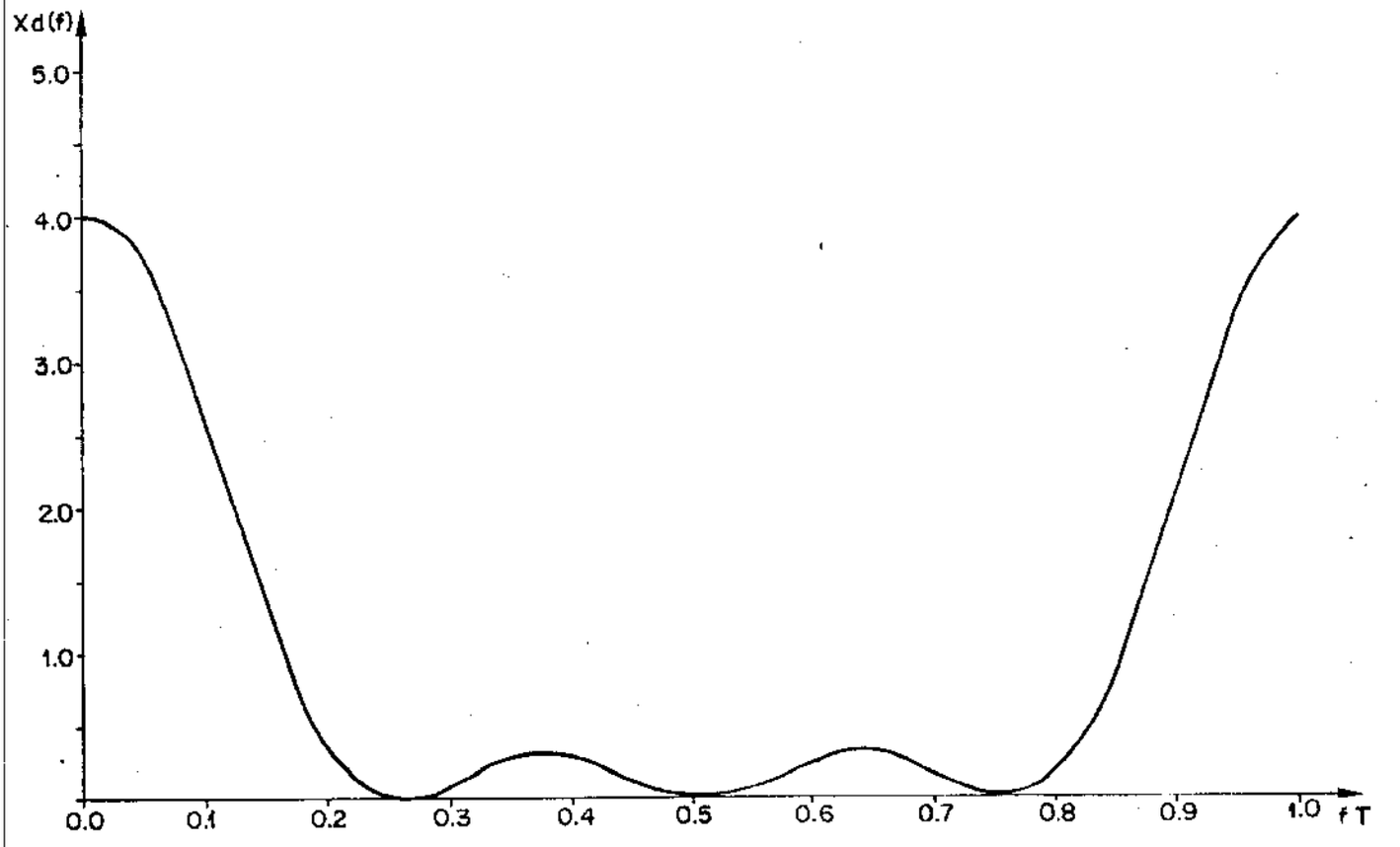


Fig. 7.15b - Componente discreto, normalizado, do espectro de potência do código convolucional sistemático (12,9) para prob.(1) = 0,5. As raias estão situadas nas frequências m/4, m=0,1,2,...

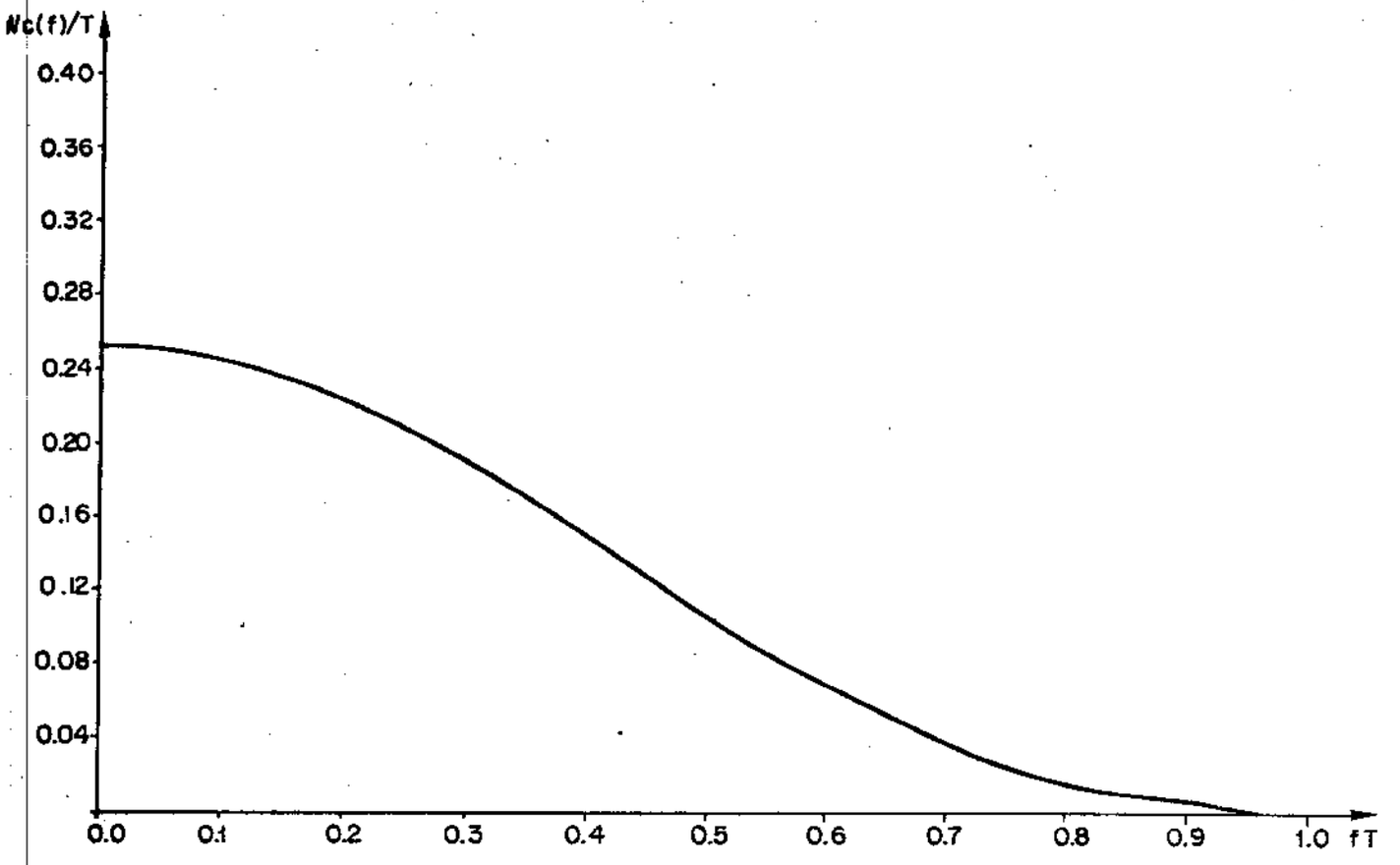


Fig. 7.15c - Componente contínuo, normalizado, do espectro de potência do código convolucional (12,9) para pulsos retangulares com fator de ocupação de 100% e prob.(1) = 0,5.

.10 - CONCLUSÕES

Nesse trabalho, através de uma associação entre a Teoria da Codificação e a Teoria das Máquinas Sequenciais Lineares, obtivemos relações que nos permitiram obter a matriz de transferência dos codificadores de códigos convolucionais a partir da matriz geradora do código. Este resultado permitiu obtermos o modelo paralelo do codificador dos códigos de bloco lineares e assim estender o método de Cariolaro, para o cálculo do espectro de potência de um sinal digital codificado, aos sinais codificados por códigos lineares.

Foram determinadas as propriedades espectrais de novos códigos, sugeridos para transmissão por fibra ótica, além de uma nova expressão para o espectro de potência do código de Petrovic.

Apesar de termos apresentado a maioria dos resultados com a utilização de códigos de bloco não lineares, esperamos ter contribuído para uma futura utilização dos códigos lineares. Em particular, cremos que os códigos convolucionais na forma não sistemática assim como os códigos cíclicos sejam bem promissores. Entre

tanto, os códigos convolucionais na forma não sistemática exigem um elevado número de memórias para o codificador, o que implica em termos sub-matrizes $\bar{A}_u$ e $\bar{E}_u$ de dimensões elevadas e,consequentemente, cresce rapidamente a quantidade de memória de computador exigida na execução do programa. Da mesma forma, os códigos cíclicos mais interessantes já apresentam um comprimento razoável . Assim, vemos que técnicas de programação mais sofisticadas devem ser utilizadas para uma pesquisa sistemática nas propriedades espectrais dos códigos lineares.

APÊNDICE A

RESUMO COMPUTACIONAL DAS CARACTERÍSTICAS DOS SUB-PROGRAMAS UTILIZADOS

SUMÁRIO DAS CARACTERÍSTICAS DA SUB-ROTINA SOPRMA

IDENTIFICAÇÃO: SUBROUTINE SOPRMA (NSUB,M,N,K,A,F,B,Q,S)

PROPÓSITO: Dadas duas matrizes A e B com o mesmo número de linhas , dividí-las em sub-matrizes de mesmo número de linhas; dada uma terceira matriz constante efetuar a seguinte operação

$$S = \sum_{u=i}^{K} q_u A_u^T F B_u$$

onde: A_u e B_u são as sub-matrizes;

F é a matriz fixa e

q_u são constantes.

SUBPROGRAMAS ADICIONAIS REQUERIDOS:

Esta sub-rotina chama as sub-rotinas identificadas por:

SUBROUTINE PRIMARE (M,N,K,A,B,P)

SUBROUTINE SOMAT (M,N,A,B,C)

ARGUMENTOS DE ENTRADA:

NSUB — Variável associada ao número de sub-matrizes em que serão divididas as matrizes A e B.

M — Variável associada ao número de linhas das sub-matrizes A_u e B_u.

N — Variável associada ao número de colunas das sub-matrizes A_u.

K — Variável associada ao número de colunas das sub-matrizes B_u.

A — Variável bidimensional onde são armazenados os elementos da matriz A.

B — Variável bidimensional onde são armazenados os elementos da matriz B.

F — Variável bidimensional onde são armazenados os elementos da matriz fixa no somatório.

Q — Variável unidimensional onde são armazenados, de forma ordenada, as constantes multiplicativas q_u.

ARGUMENTOS DE SAÍDA:

S — Variável bidimensional onde são armazenados os elementos da matriz resultante da operação

SUMÁRIO DAS CARACTERÍSTICAS DA SUB-ROTINA SOMSUB

IDENTIFICAÇÃO: SUBROUTINE SOMSUB (M,N,NSUB,X,CONST,SOM)

PROPÓSITO: Dada uma matriz, determinar uma partição de sub-matrizes com mesmo número de linhas e com o número de colunas igual ao da matriz original, multiplicar cada sub-matriz obtida por uma respectiva constante e finalmente efetuar a soma dessas sub-matrizes após a multiplicação por suas respectivas constantes, isto é, efetuar operações da forma:

$$S = \sum_{u=i}^{K} q_u A_u$$

SUB-PROGRAMAS ADICIONAIS REQUERIDOS: Nenhum.

ARGUMENTOS DE ENTRADA:

M — Variável associada ao número de linhas das sub-matrizes.

N — Variável associada ao número de colunas das sub-matrizes.

NSUB — Variável associada ao número de sub-matrizes em que é dividida a matriz dada.

X — Variável bidimensional onde são armazenados os elementos da matriz a sofrer partição.

CONST — Variável unidimensional onde são armazenadas as constantes multiplicadoras das sub-matrizes.

ARGUMENTOS DE SAÍDA:

SOM — Variável bidimensional onde são armazenados os elementos da matriz resultante da operação.

SUMÁRIO DAS CARACTERÍSICAS DA SUB-ROTINA SPACE

IDENTIFICAÇÃO: SUBROUTINE SPACE (IDIM,MAT)

PROPÓSITO: Construir um espaço vetorial sobre GF(2) com os elementos do espaço ordenados segundo a representação binária dos inteiros $0,1,\ldots,2^K-1$, onde K é a dimensão do espaço.

SUB-PROGRAMAS ADICIONAIS REQUERIDOS. Nenhum.

ARGUMENTOS DE ENTRADA:

IDIM — Variável associada à dimensão do espaço.

ARGUMENTOS DE SAÍDA:

MAT — Variável bidimensional onde, em cada linha, são armazenados os elementos do espaço vetorial.

SUMÁRIO DAS CARACTERÍSTICAS DA SUB-ROTINA SOMAT

IDENTIFICAÇÃO: SUBROUTINE SOMAT (M,N,A,B,C)

PROPÓSITO: Efetuar a soma entre duas matrizes.

SUB-PROGRAMAS ADICIONAIS REQUERIDOS: Nenhum.

ARGUMENTOS DE ENTRADA:

M — Variável associada ao número de linhas das matrizes a serem somadas.

N — Variável associada ao número de colunas das matrizes a serem somadas.

A — Variável bidimensional onde são armazenados os elementos da primeira matriz.

B — Variável bidimensional onde são armazenados os elementos da segunda matriz.

ARGUMENTOS DE SAÍDA:

C — Variável bidimensional onde são armazenados os elementos da matriz resultante da soma das matrizes.

SUMÁRIO DAS CARACTERÍSTICAS DA SUB-ROTINA SOMAHK

IDENTIFICAÇÃO: SUBROUTINE SOMAHK (ISOMID,N,JP,HKNXT,SN)

PROPÓSITO: Efetuar um somatório com os elementos de duas matrizes dadas, através da expressão:

$$n_{p+k.N} = \sum_{i=1}^{N-p} H_k(i+p,i) + \sum_{i=1}^{p} H_{k+1}(i,i+N-p)$$

SUB-PROGRAMAS ADICIONAIS REQUERIDOS: Nenhum.

ARGUMENTOS DE ENTRADA

ISOMID — Variável associada ao índice p+k.N.

N — Variável associada à ordem das matrizes H_k e H_{k+1}.

IP — Variável associada ao termo p.

HK — Variável bidimensinal onde são armazenados os elementos da primeira matriz.

HKNXT — Variável bidimensional onde são armazenados os elementos da segunda matriz.

ARGUMENTOS DE SAÍDA:

SN — Variável associada ao resultado do somatório efetuado.

SUMÁRIO DAS CARACTERÍSTICAS DA SUB-ROTINA PRMARE

IDENTIFICAÇÃO: SUBROUTINE PRMARE (M,N,K,A,B,P)

PROPÓSITO: Determinar o produto entre duas matrizes com elementos reais.

SUB-PROGRAMAS ADICIONAIS REQUERIDOS: Nenhum.

ARGUMENTOS DE ENTRADA:

M — Variável associada ao número de linhas da primeira matriz fator.

N — Variável associada ao número de colunas da primeira matriz fator e ao número de linhas da segunda matriz fator.

K — Variável associada ao número de colunas da segunda matriz fator.

A — Variável bidimensional onde são armazenados os elementos da primeira matriz fator.

B — Variável bidimensional onde são armazenados os elementos da segunda matriz fator.

ARGUMENTOS DE SAÍDA:

p — Variável bidimensional onde são armazenados os elementos da matriz produto.

SUMÁRIO DAS CARACTERÍSTICAS DA SUB-ROTINA TRASO

IDENTIFICAÇÃO: SUBROUTINE TRASO (M,FG,K,DK)

PROPÓSITO: Determinar o traço de uma matriz e dividí-lo pelo negativo de uma constante.

SUB-PROGRAMAS ADICIONAIS REQUERIDOS: Nenhum.

ARGUMENTOS DE ENTRADA:

M — Variável associada à ordem da matriz cujo trabalho deve ser determinado.

FG — Variável bidimensional onde são armazenados os elementos da matriz cujo traço deve ser determinado.

K — Variável associado à constante que divide o traço da matriz.

ARGUMENTOS DE SAÍDA:

DK — Variável associada ao traço da matriz após a divisão pelo negativo da constante.

SUMÁRIO DAS CARACTERÍSTICAS DA SUB-ROTINA PLOT

IDENTIFICAÇÃO: SUBROUTINE PLOT (y,M,NF,NS,XK,ESC)

PROPÓSITO: Esboçar o gráfico de um máximo de duas funções através da determinação de um máximo de 100 pontos para cada função.

SUB-PROGRAMAS ADICIONAIS REQUERIDOS: Nenhum.

ARGUMENTOS DE ENTRADA:

y — Variável bidimensional aonde são armazenados as ordenadas das funções a serem plotadas.

M — Variável associada ao número de funções a serem plotadas.

NF — Variável associada ao número de pontos que se deseja plotar.

NS — Variável associada ao máximo valor da escala das ordenadas usado para plotar as funções.

XK — Variável usada para transformar as abssissas (inteiros múltiplos de 10) nos valores desejados.

ESC — Variável associada a um fator de escala que pode ser utilizado para uma melhor apresentação dos gráficos das funções.

ARGUMENTOS DE SAÍDA: Nenhum.

SUMÁRIO DAS CARACTERÍSTICAS DA SUB-ROTINA PRIDCT

IDENTIFICAÇÃO: SUBROUTINE PRIDCT (M,CONST,DIAG)

PROPÓSITO: Construir uma matriz que seja o resultado do produto de uma constante pela matriz identidade.

SUB-PROGRAMAS ADICIONAIS REQUERIDOS: Nenhum.

ARGUMENTOS DE ENTRADA:

M — Variável associada à ordem da matriz identidade.

CONST — Variável associada à constante multiplicadora da matriz identidade.

ARGUMENTOS DE SAÍDA:

DIAG — Variável bidimensional onde são armazenados os elementos da matriz diagonal resultante.

SUMÁRIO DAS CARACTERÍSTICAS DA SUB-ROTINA PRIMAJN

IDENTIFICAÇÃO: SUBROUTINE PRIMAJN (M,N,K,JA,JB,JP)

PROPÓSITO: Determinar o produto entre duas matrizes com elementos inteiros.

SUB-PROGRAMAS ADICIONAIS REQUERIDOS: Nenhum.

ARGUMENTOS DE ENTRADA:

M — Variável associada ao número de linhas da primeira matriz fator.

N — Variável associada ao número de colunas da primeira matriz fator e ao número de linhas da segunda matriz fator.

K — Variável associada ao número de colunas da segunda matriz fator.

JA — Variável bidimensional onde são armazenados os elementos da primeira matriz fator.

JB — Variável bidimensional onde são armazenados os elementos da segunda matriz fator.

ARGUMENTOS DE SAÍDA:

JP — Variável bidimensional onde são armazenados os elementos da matriz produto.

SUMÁRIO DAS CARACTERÍSTICAS DA SUB-ROTINA SPECTR

IDENTIFICAÇÃO: SUBROUTINE SPECTR (N,NSTAT,NVI)

PROPÓSITO: Determinar os gráficos dos componentes contínuo e discretos, normalizados, do espectro de potência de um código. Determinar um gráfico do componente contínuo considerando pulsos retangulares com um fator de ocupação de 100%.

SUB-PROGRAMAS ADICIONAIS REQUERIDOS: Esta sub-rotina chama as sub-rotinas identificadas por:

SOMSUB (M,N,NSUB,X,CONST,SOM)

SOPRMA (NSUB,M,N,K,A,F,B,Q,S)

PRMARE (M,N,K,A,B,P)

PRIDCT (M,CONST,DIAG)

TRASO (M,FG,K,DK)

SOMAT (M,N,A,B,C)

SOMAHK (ISOMID,N,IP,HK,HKNXT,SN)

PLOT (y,M,NF,NS,XK,ESC)

ARGUMENTOS DE ENTRADA:

N — Variável associada ao comprimento das palavras-código do código utilizado.

NSTAT — Variável associada ao número de estados da máquina sequencial de estado finito associada ao codificador.

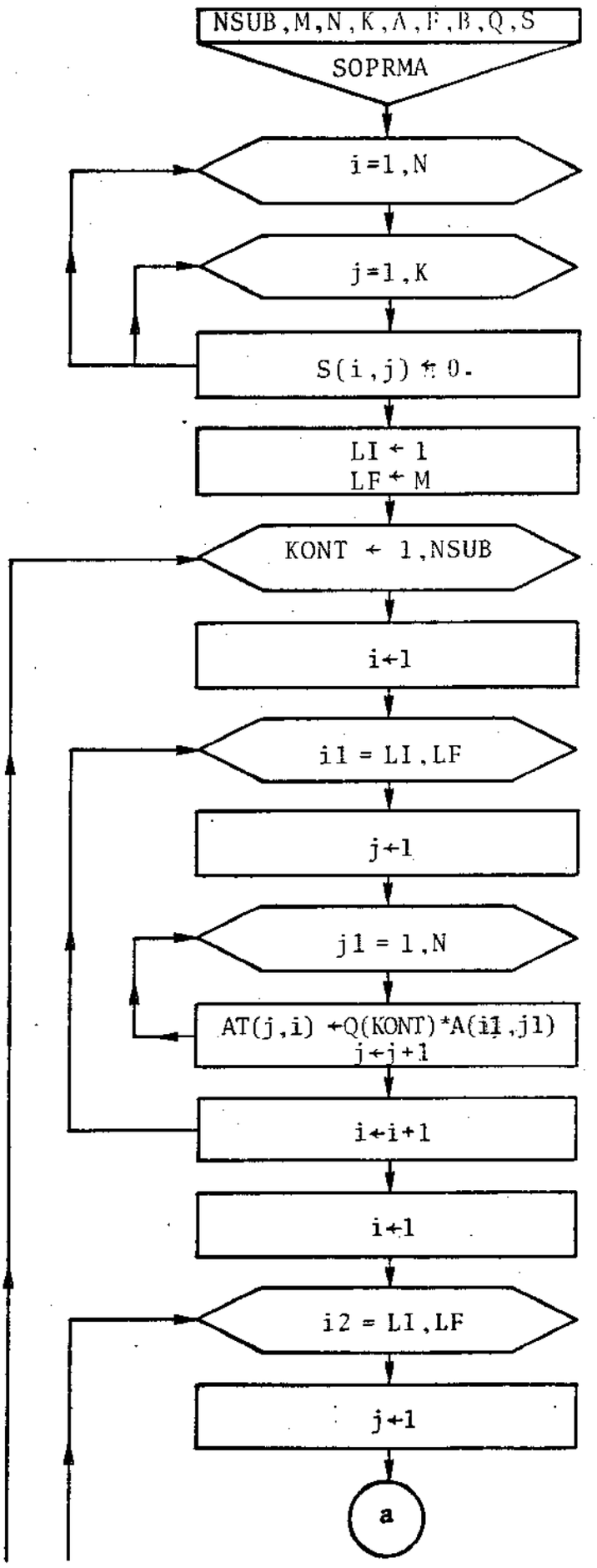
NVI — Variável associada ao número de vetores de informação.

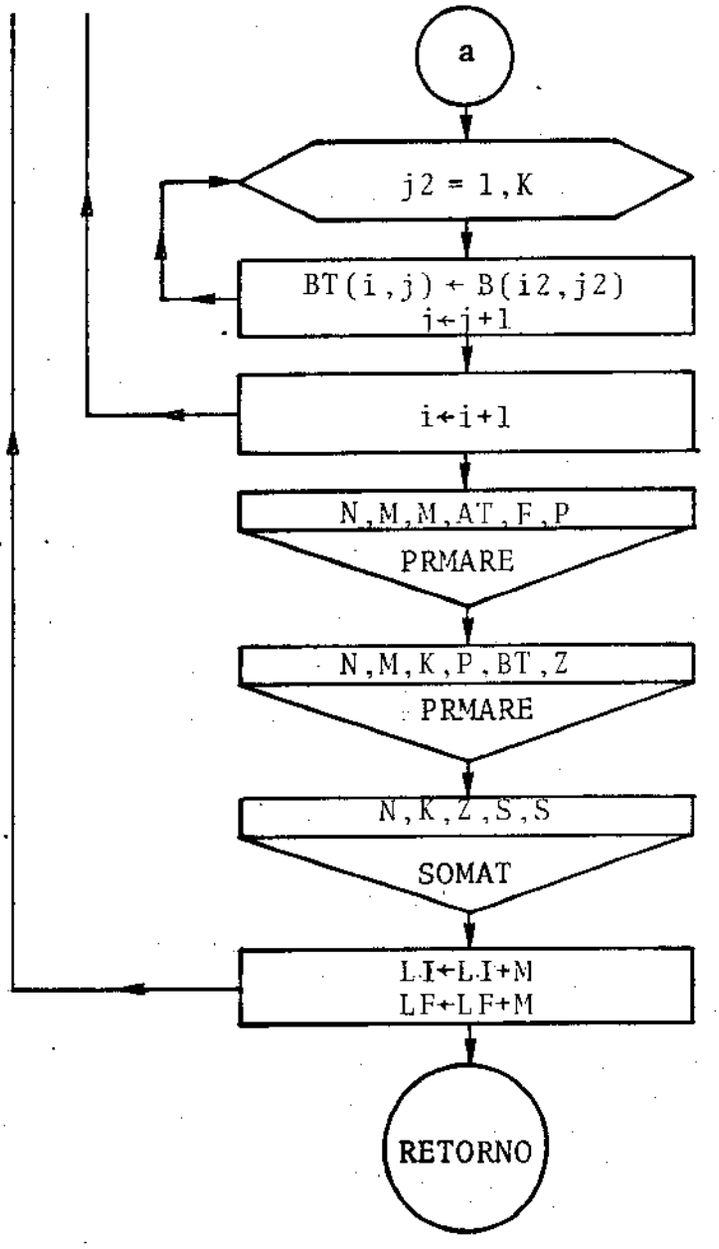
ARGUMENTOS DE SAÍDA: Nenhum.

APÊNDICE B

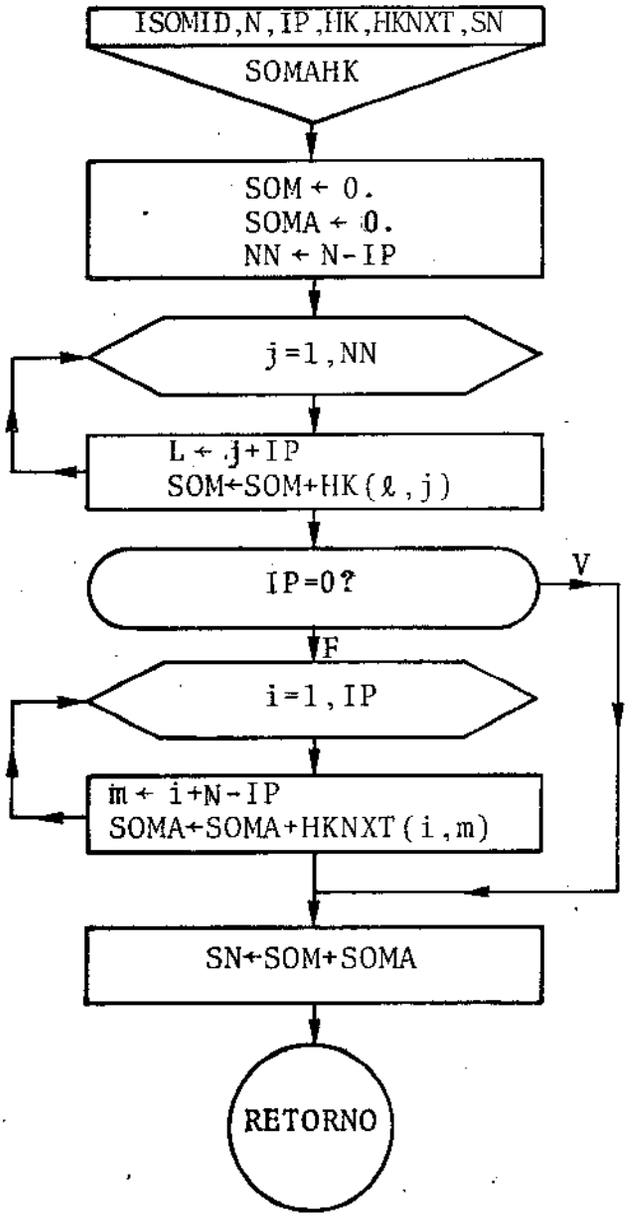
FLUXOGRAMAS DOS PROGRAMAS E SUB-PROGRAMAS IMPLEMENTADOS

FLUXOGRAMA DA SUBROTINA SOPRMA

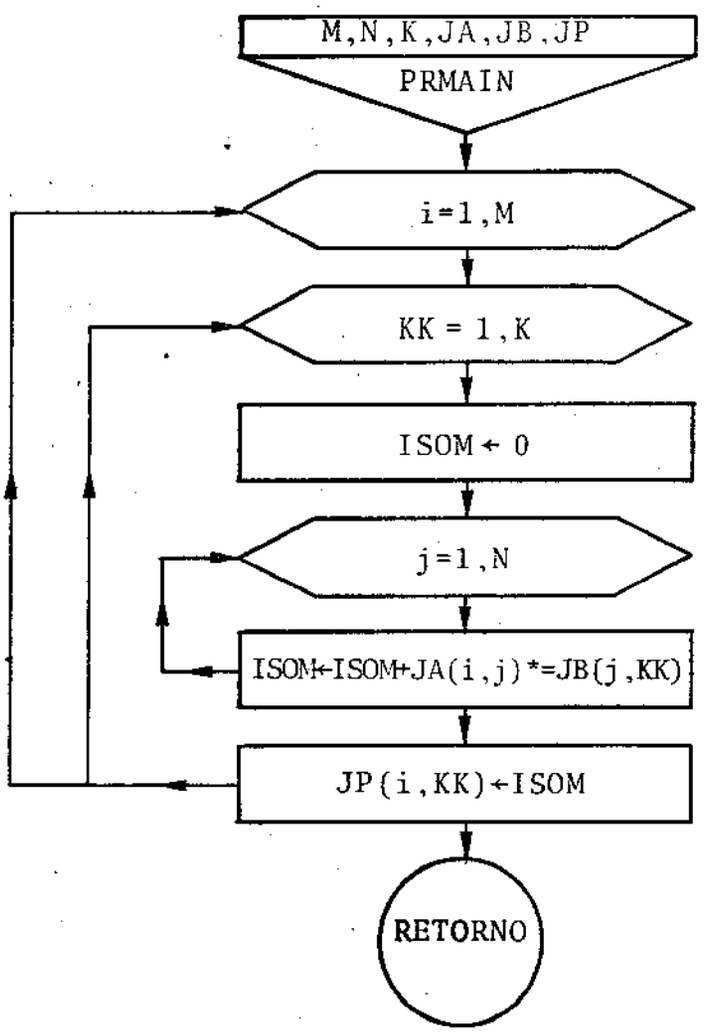



a
j2 = 1,K
BT(i,j) ← B(i2,j2)
j←j+1
i←i+1
N,M,M,AT,F,P
PRMARE
N,M,K,P,BT,Z
PRMARE
N,K,Z,S,S
SOMAT
LI←LI+M
LF←LF+M
RETORNO

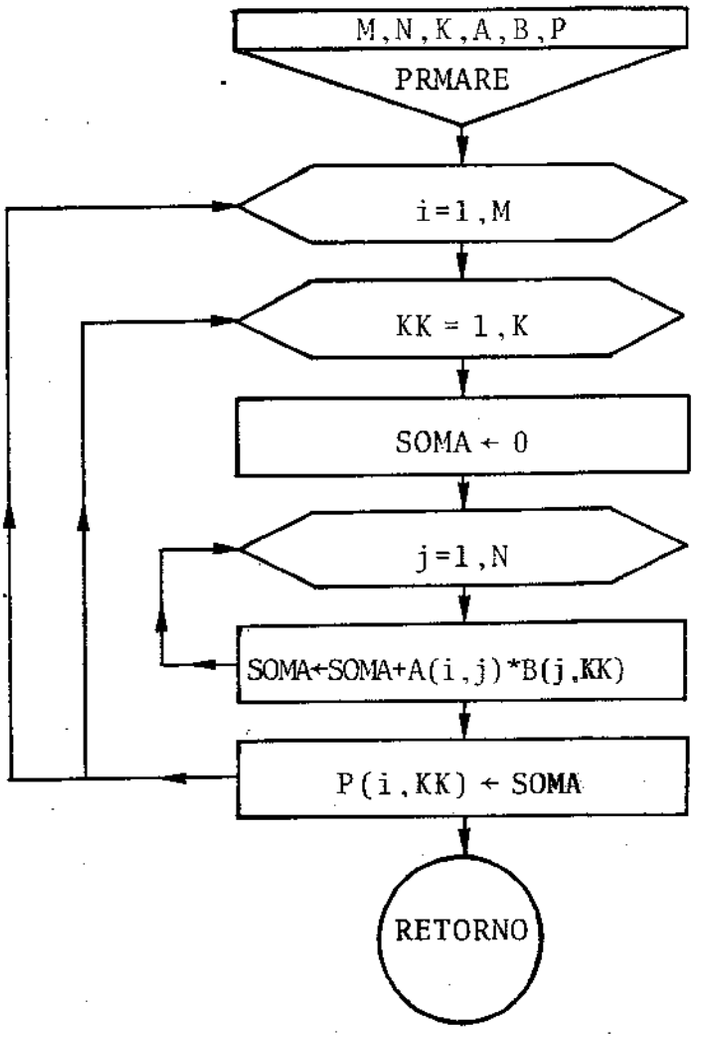
FLUXOGRAMA DA SUBROTINA SOMAHK



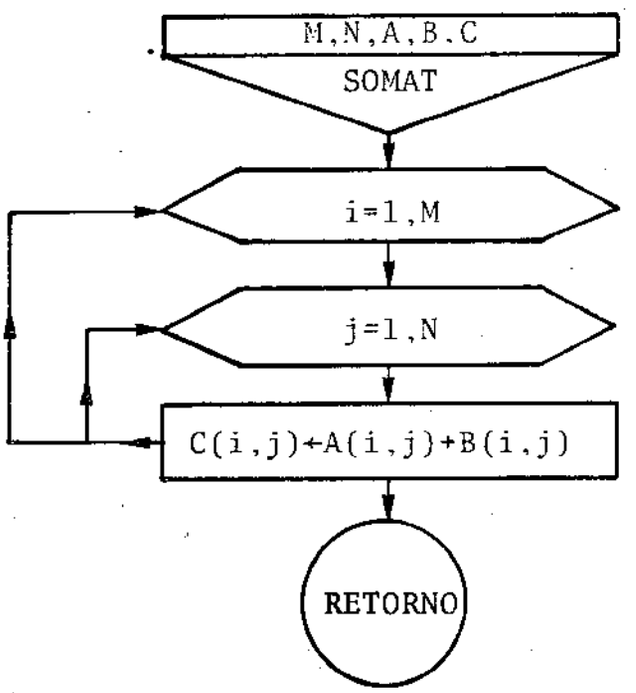
FLUXOGRAMA DA SUBROTINA PRMAIN



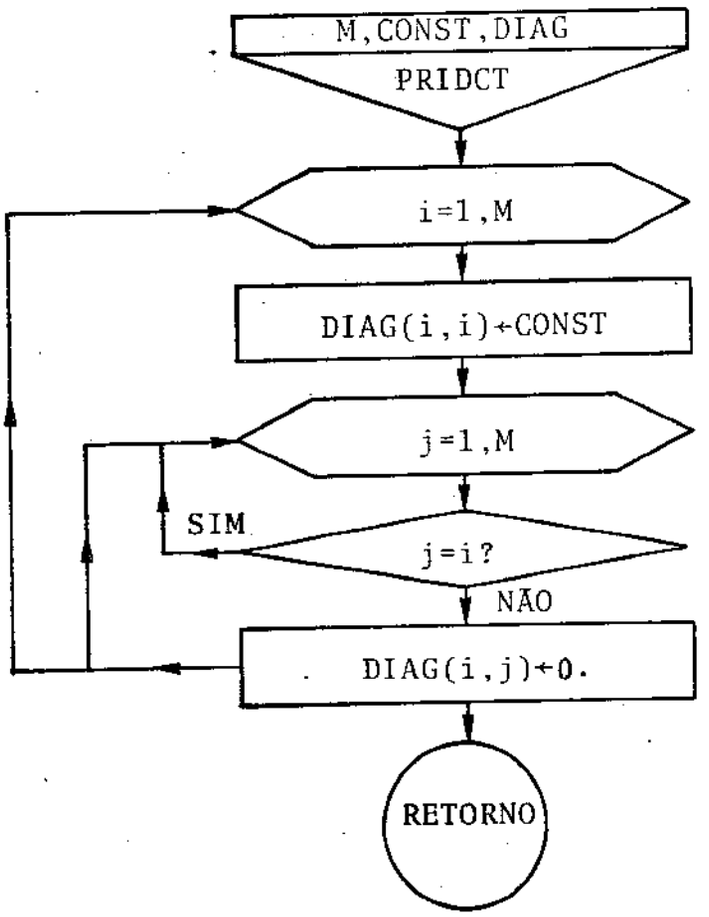
FLUXOGRAMA DA SUBROTINA PRMARE



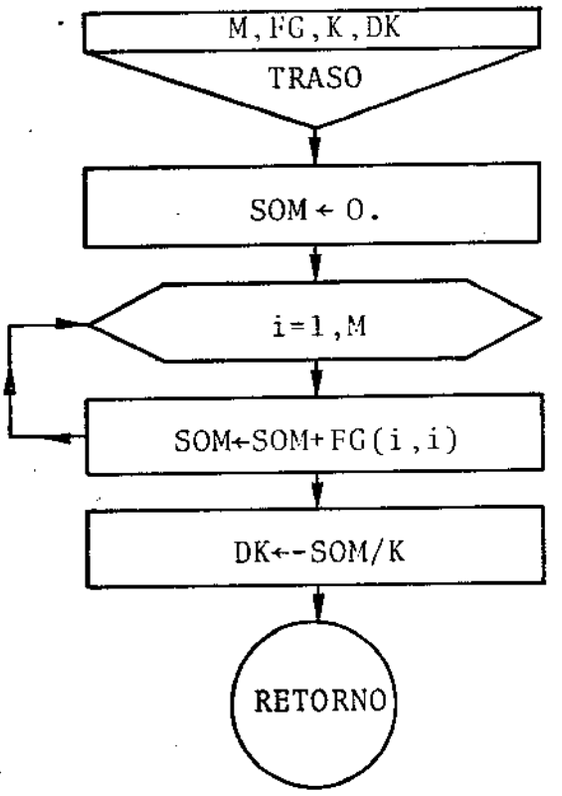
FLUXOGRAMA DA SUBROTINA SOMAT



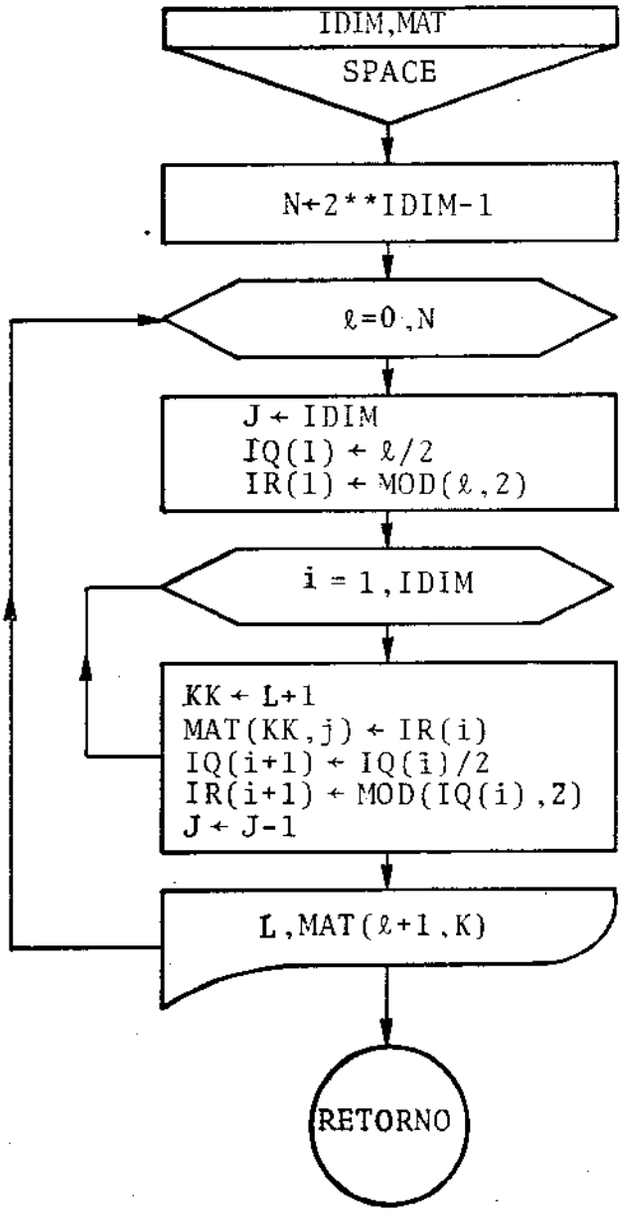
FLUXOGRAMA DA SUBROTINA PRIDCT



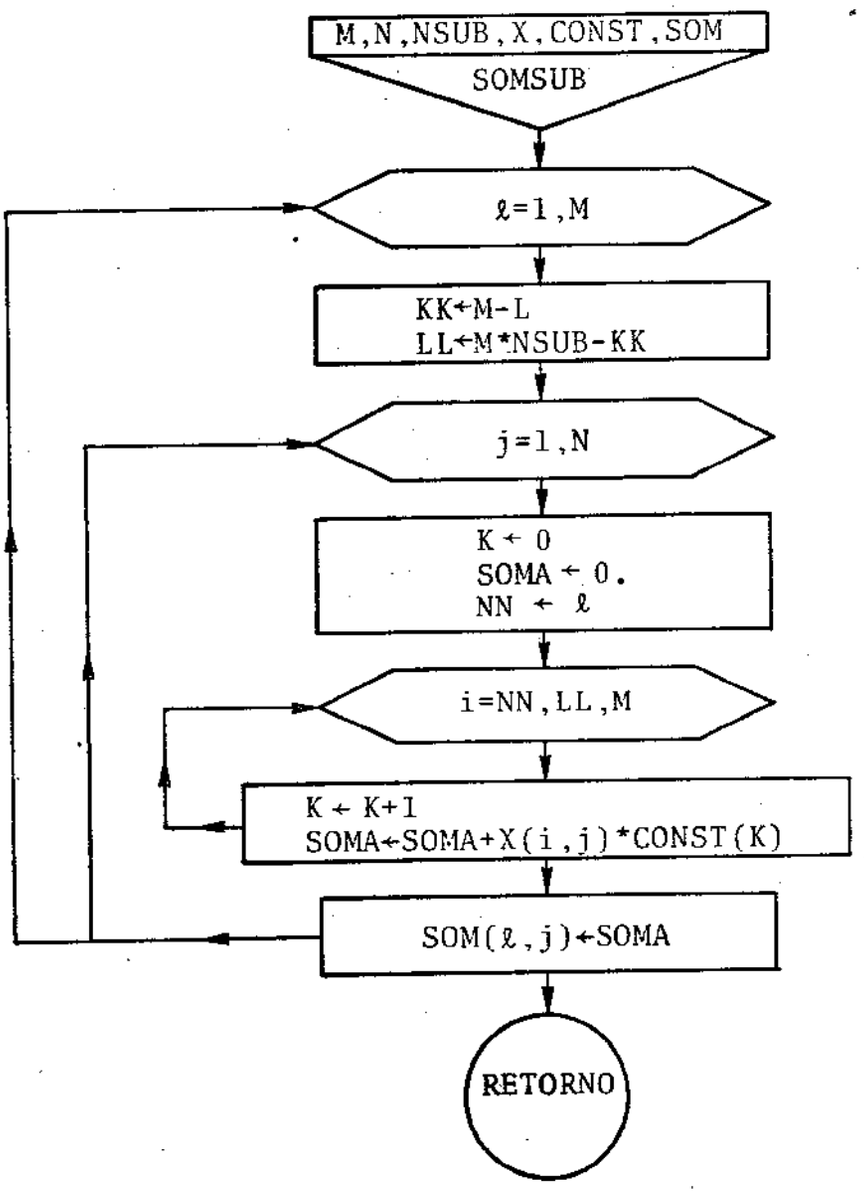
FLUXOGRAMA DA SUBROTINA TRASO



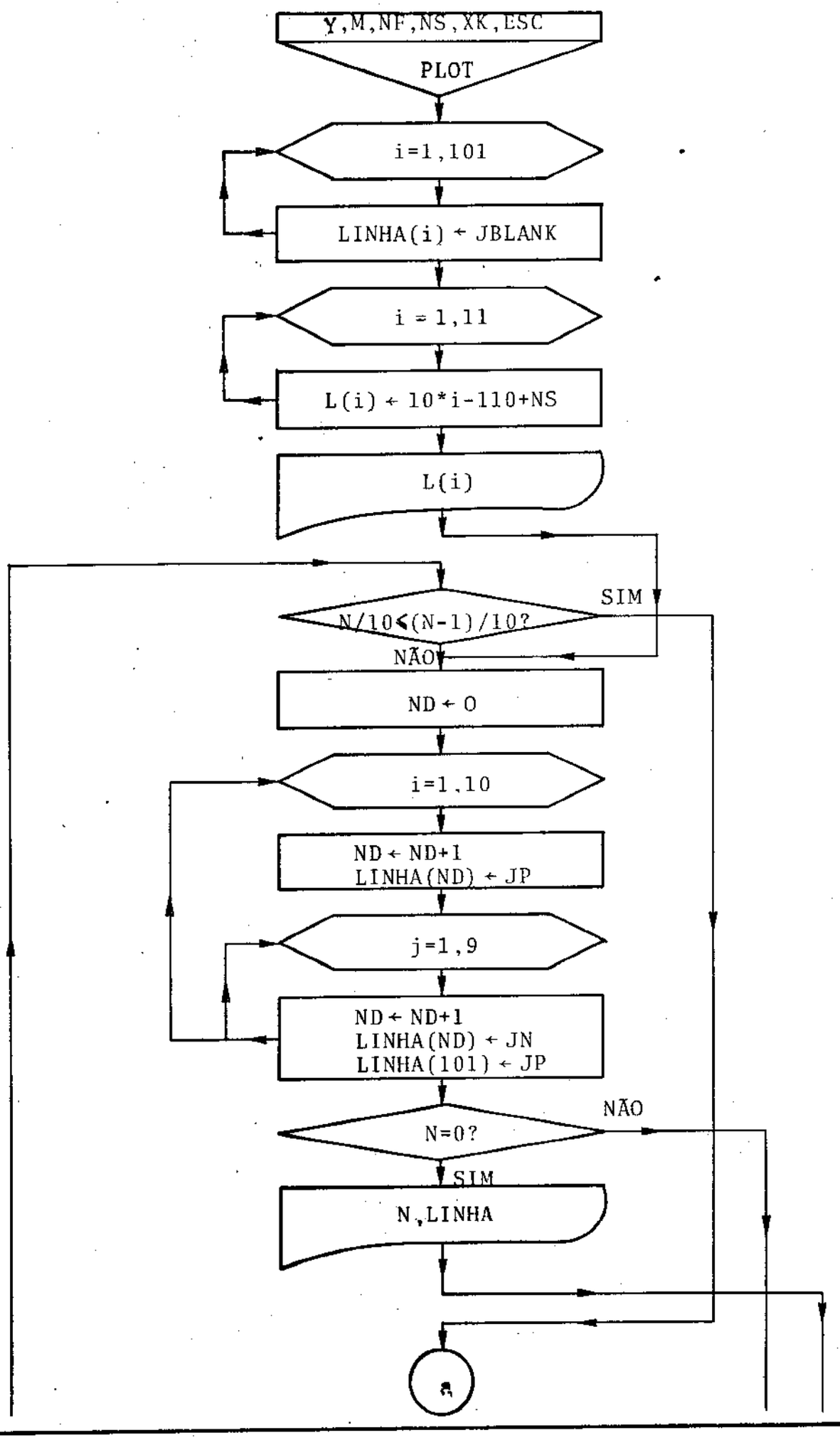
FLUXOGRAMA DA SUBROTINA SPACE

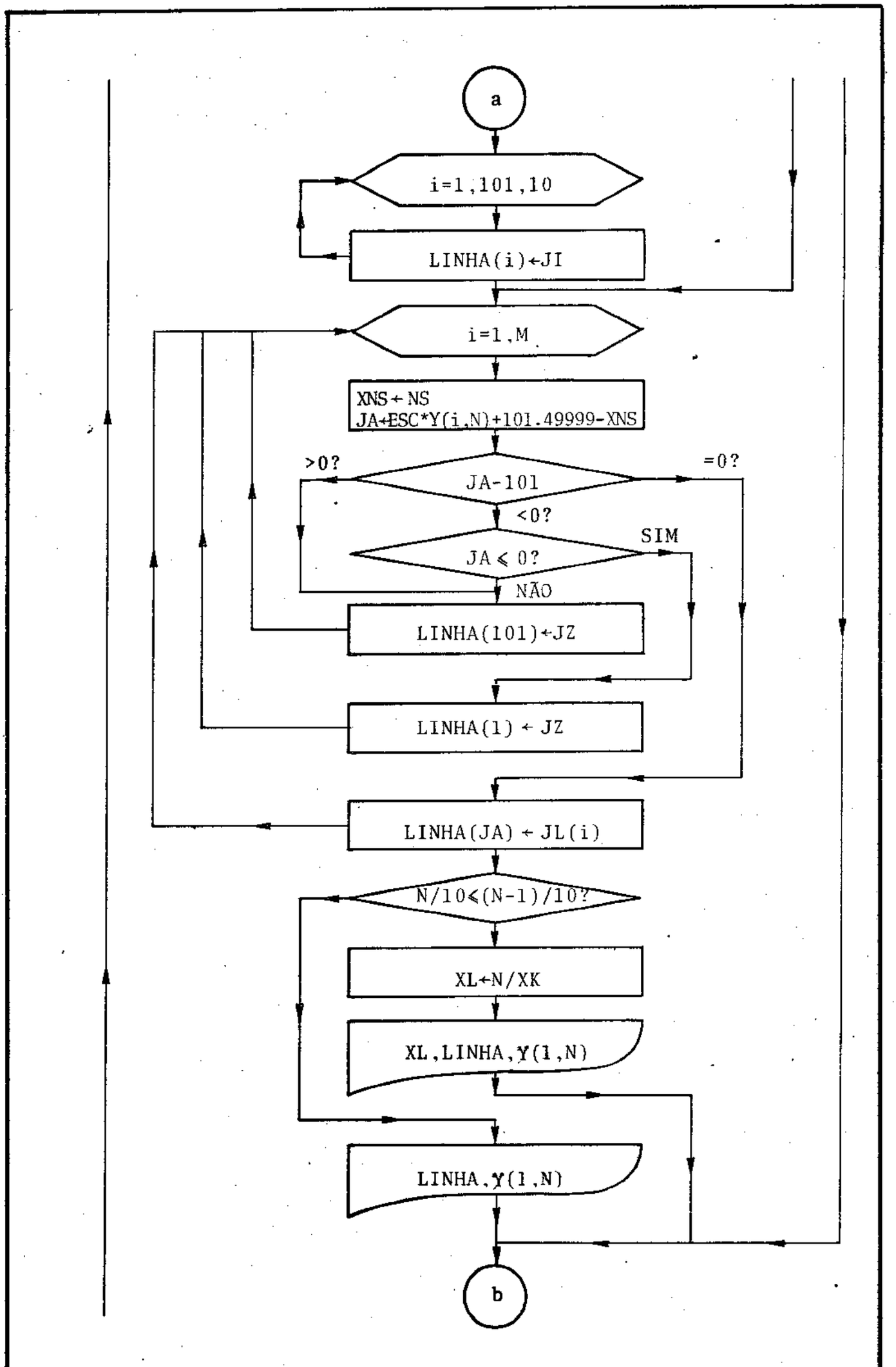


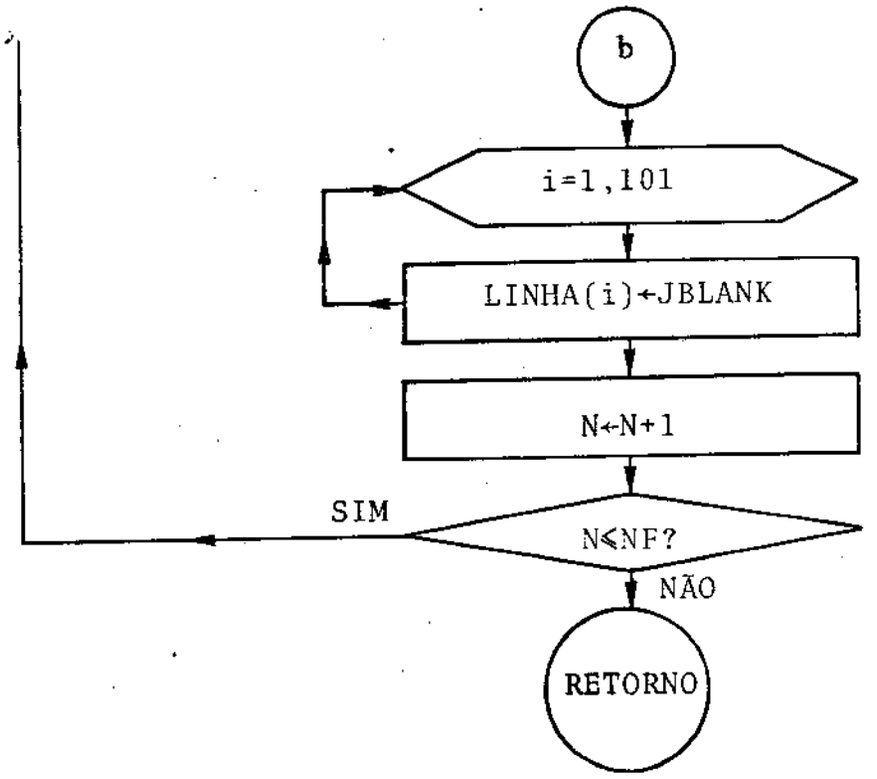
FLUXOGRAMA DA SUBROTINA SOMSUB



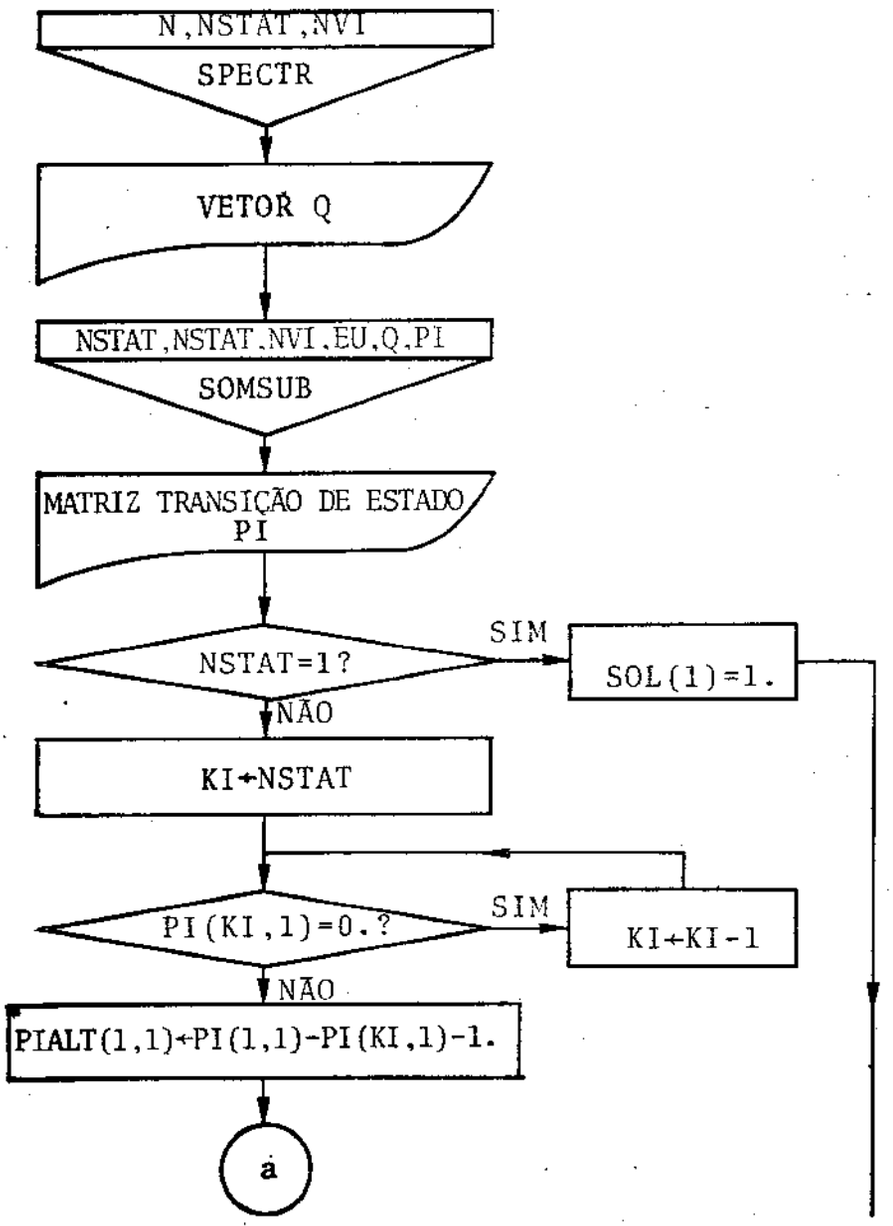
FLUXOGRAMA DA SUBROTINA PLOT

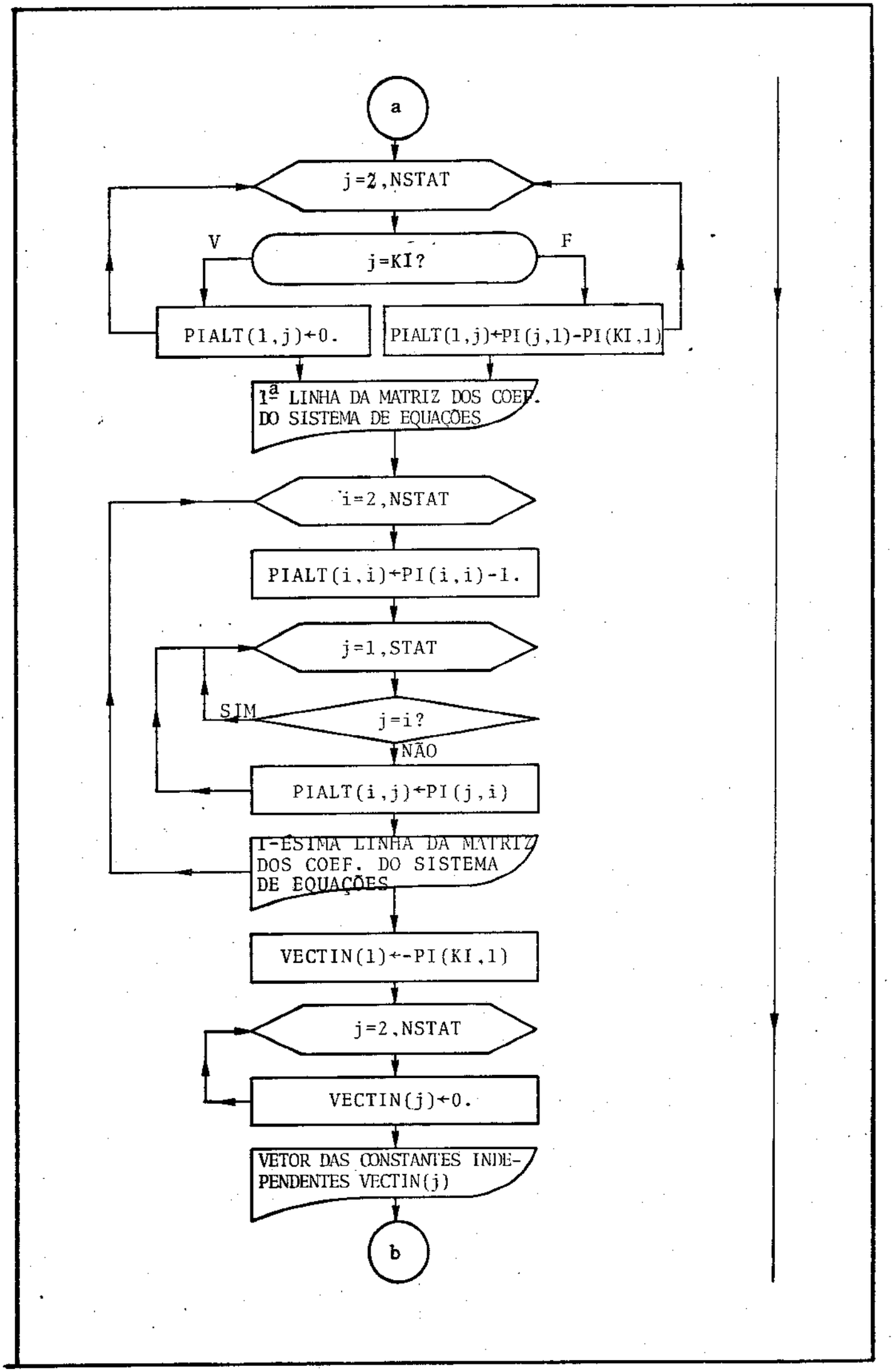


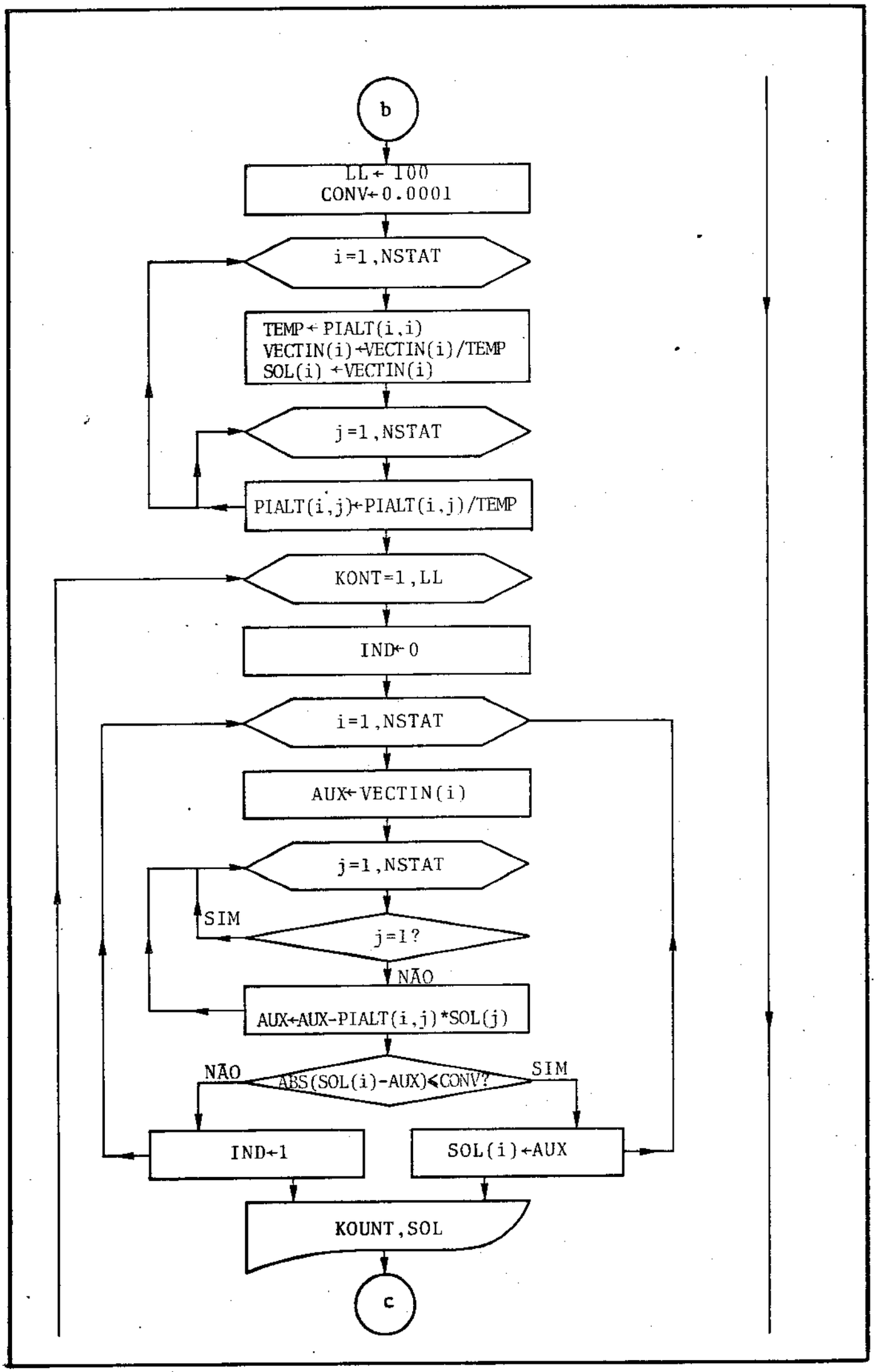

a
i=1,101,10
LINHA(i)←JI
i=1,M
XNS←NS
JA←ESC*Y(i,N)+101.49999-XNS
>0?
JA-101
=0?
<0?
SIM
JA ⩽ 0?
NÃO
LINHA(101)←JZ
LINHA(1) ← JZ
LINHA(JA) ← JL(i)
N/10⩽(N-1)/10?
XL←N/XK
XL,LINHA,Y(1,N)
LINHA,Y(1,N)
b

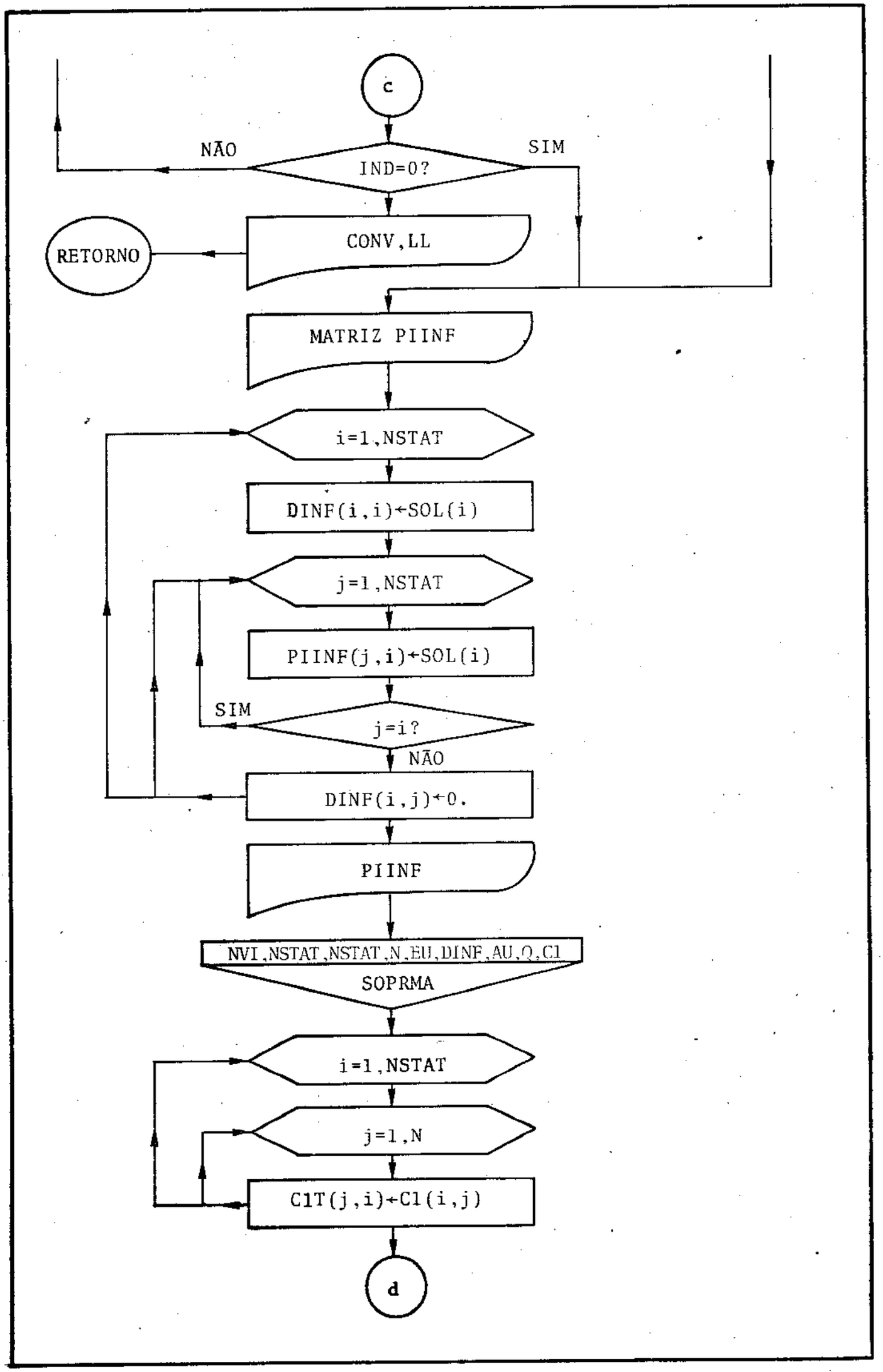

b
i=1,101
LINHA(i)←JBLANK
N←N+1
SIM
N≤NF?
NÃO
RETORNO

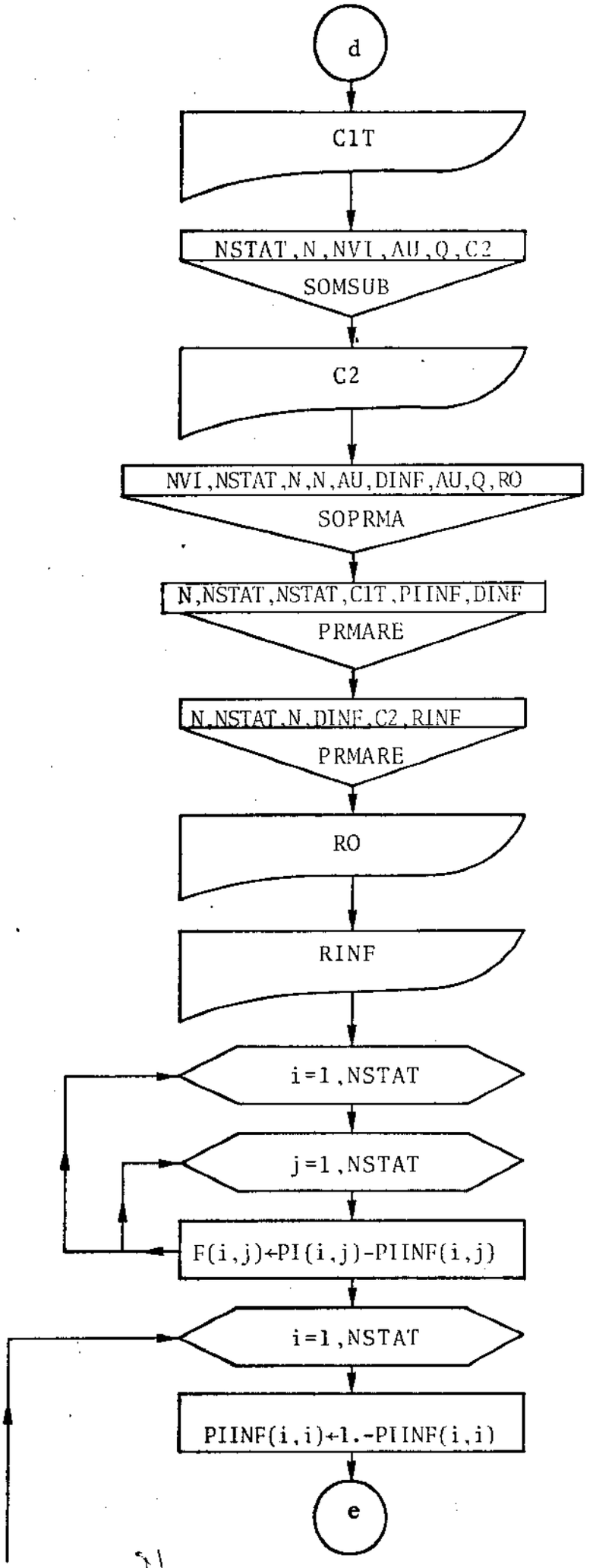
FLUXOGRAMA DA SUBROTINA SPECTR

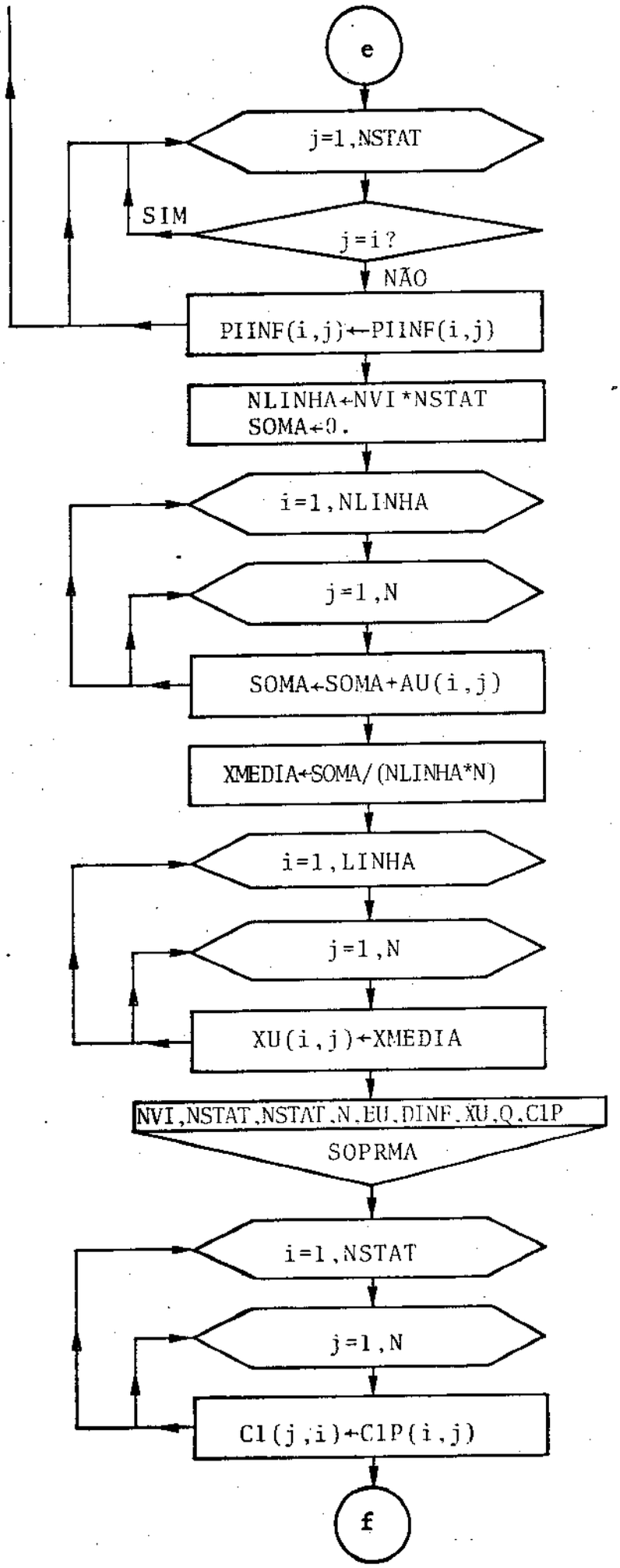


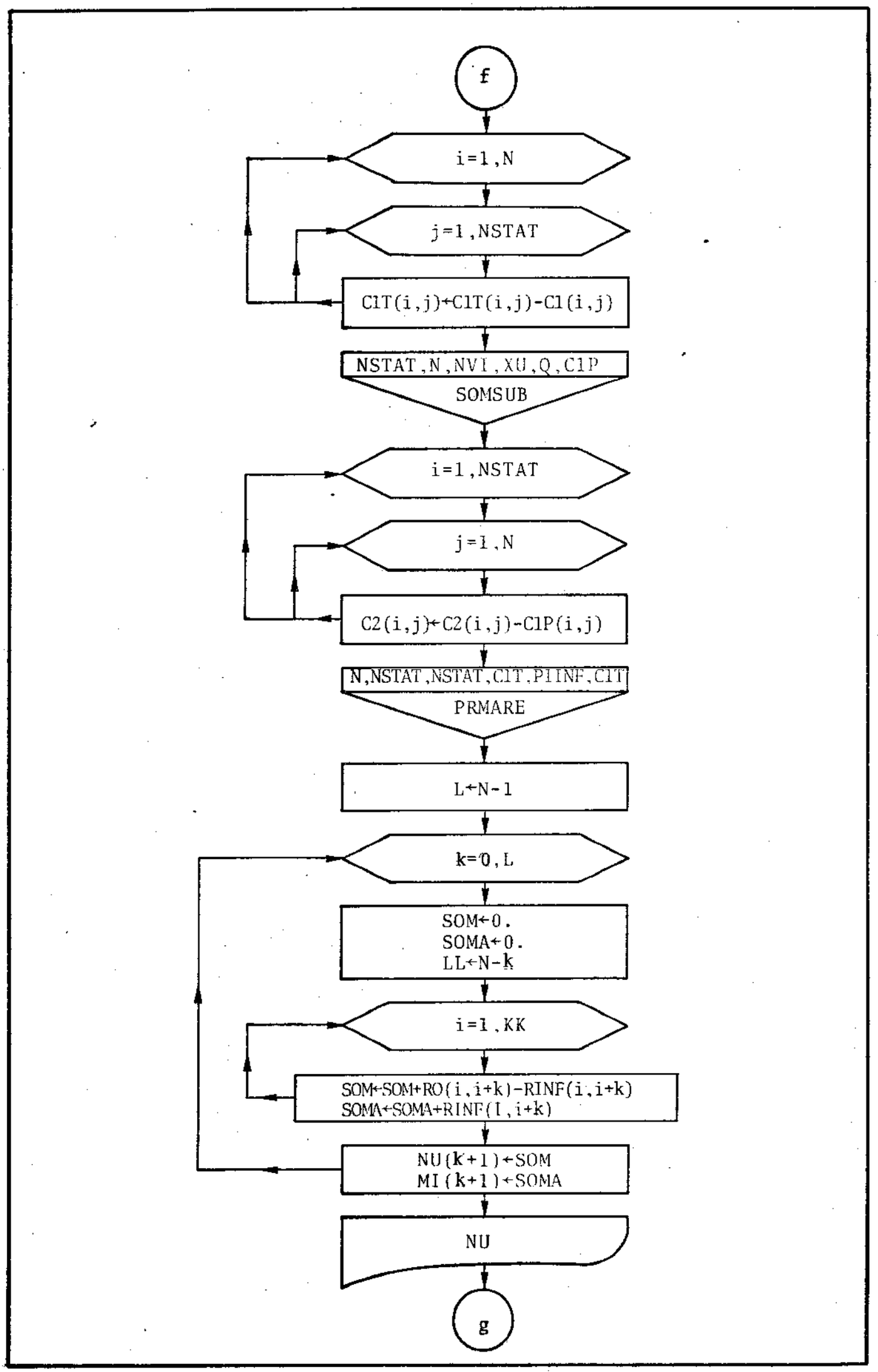

a
j=2,NSTAT
V
j=KI?
F
PIALT(1,j)←0.
PIALT(1,j)←PI(j,1)-PI(KI,1)
1ª LINHA DA MATRIZ DOS COEF. DO SISTEMA DE EQUAÇÕES
i=2,NSTAT
PIALT(i,i)←PI(i,i)-1.
j=1,STAT
SIM
j=i?
NÃO
PIALT(i,j)←PI(j,i)
I-ÉSIMA LINHA DA MATRIZ DOS COEF. DO SISTEMA DE EQUAÇÕES
VECTIN(1)←-PI(KI,1)
j=2,NSTAT
VECTIN(j)←0.
VETOR DAS CONSTANTES INDEPENDENTES VECTIN(j)
b

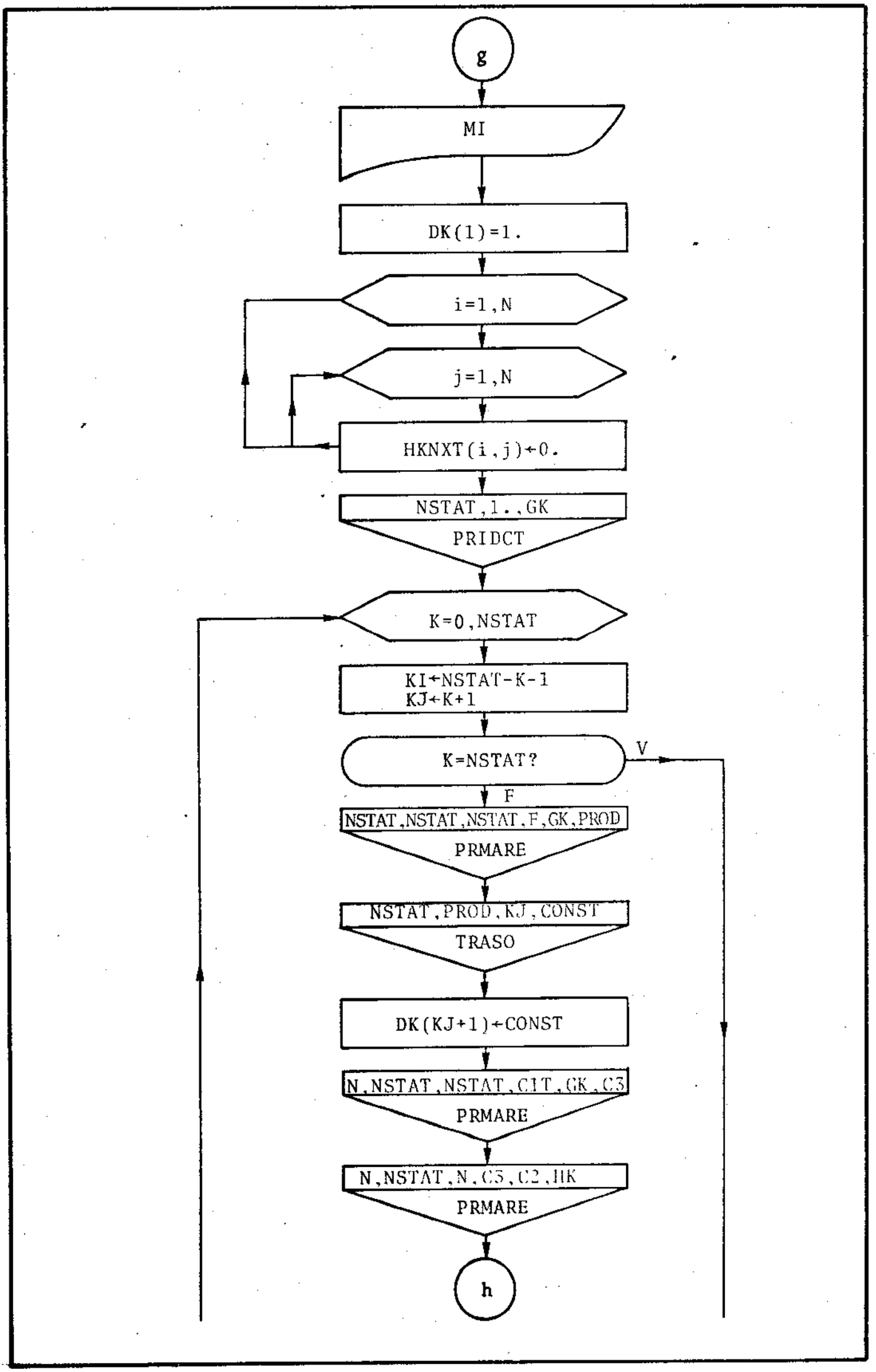

b
LL ← 100
CONV ← 0.0001
i=1,NSTAT
TEMP ← PIALT(i,i)
VECTIN(i) ← VECTIN(i)/TEMP
SOL(i) ← VECTIN(i)
j=1,NSTAT
PIALT(i,j) ← PIALT(i,j)/TEMP
KONT=1,LL
IND ← 0
i=1,NSTAT
AUX ← VECTIN(i)
j=1,NSTAT
SIM
j=1?
NÃO
AUX ← AUX-PIALT(i,j)*SOL(j)
NÃO
ABS(SOL(i)-AUX)<CONV?
SIM
IND ← 1
SOL(i) ← AUX
KOUNT,SOL
c

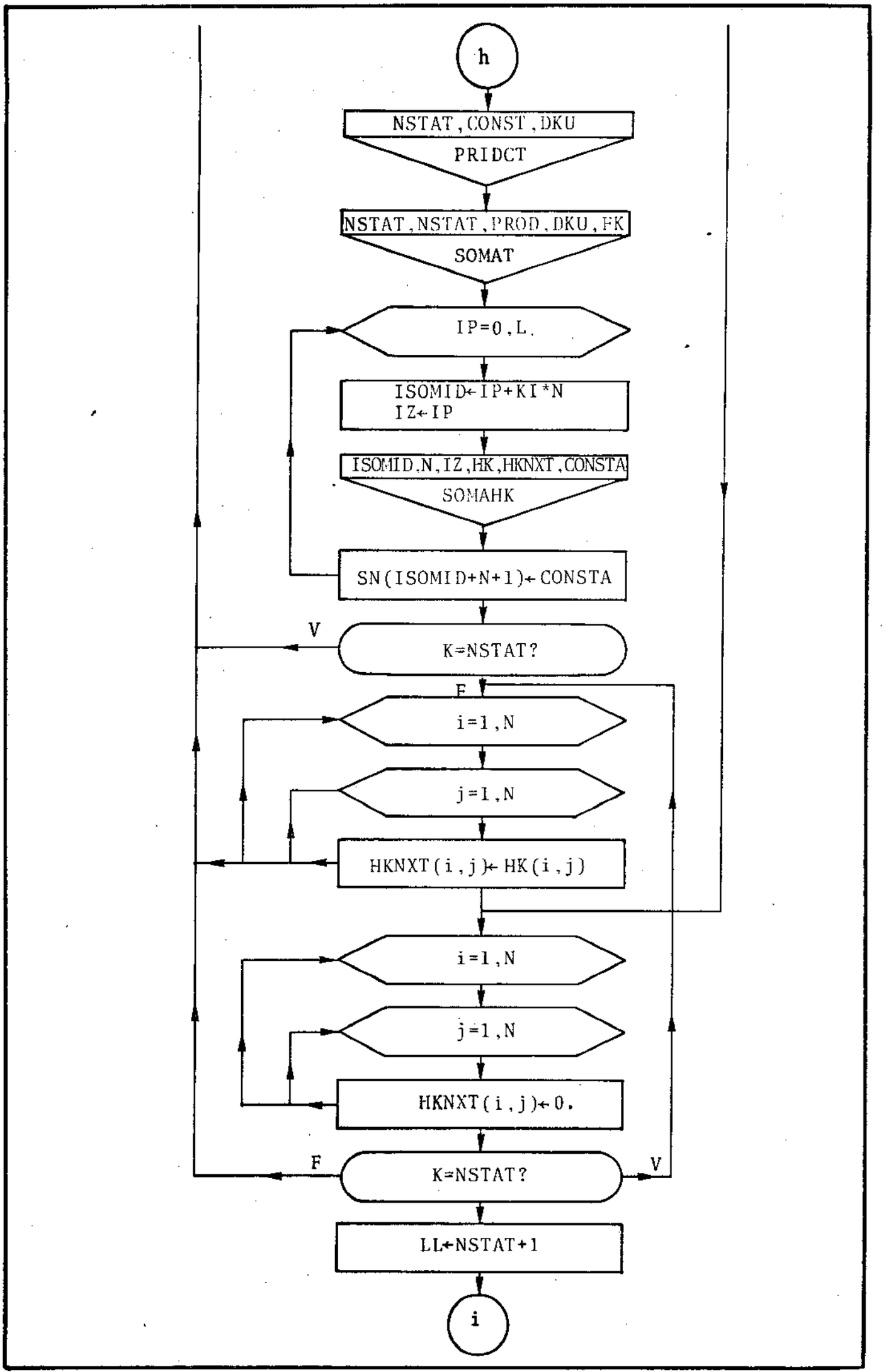

c
NÃO
IND=0?
SIM
RETORNO
CONV,LL
MATRIZ PIINF
i=1,NSTAT
DINF(i,i)←SOL(i)
j=1,NSTAT
PIINF(j,i)←SOL(i)
SIM
j=i?
NÃO
DINF(i,j)←0.
PIINF
NVI,NSTAT,NSTAT,N,EU,DINF,AU,Q,C1
SOPRMA
i=1,NSTAT
j=1,N
C1T(j,i)←C1(i,j)
d

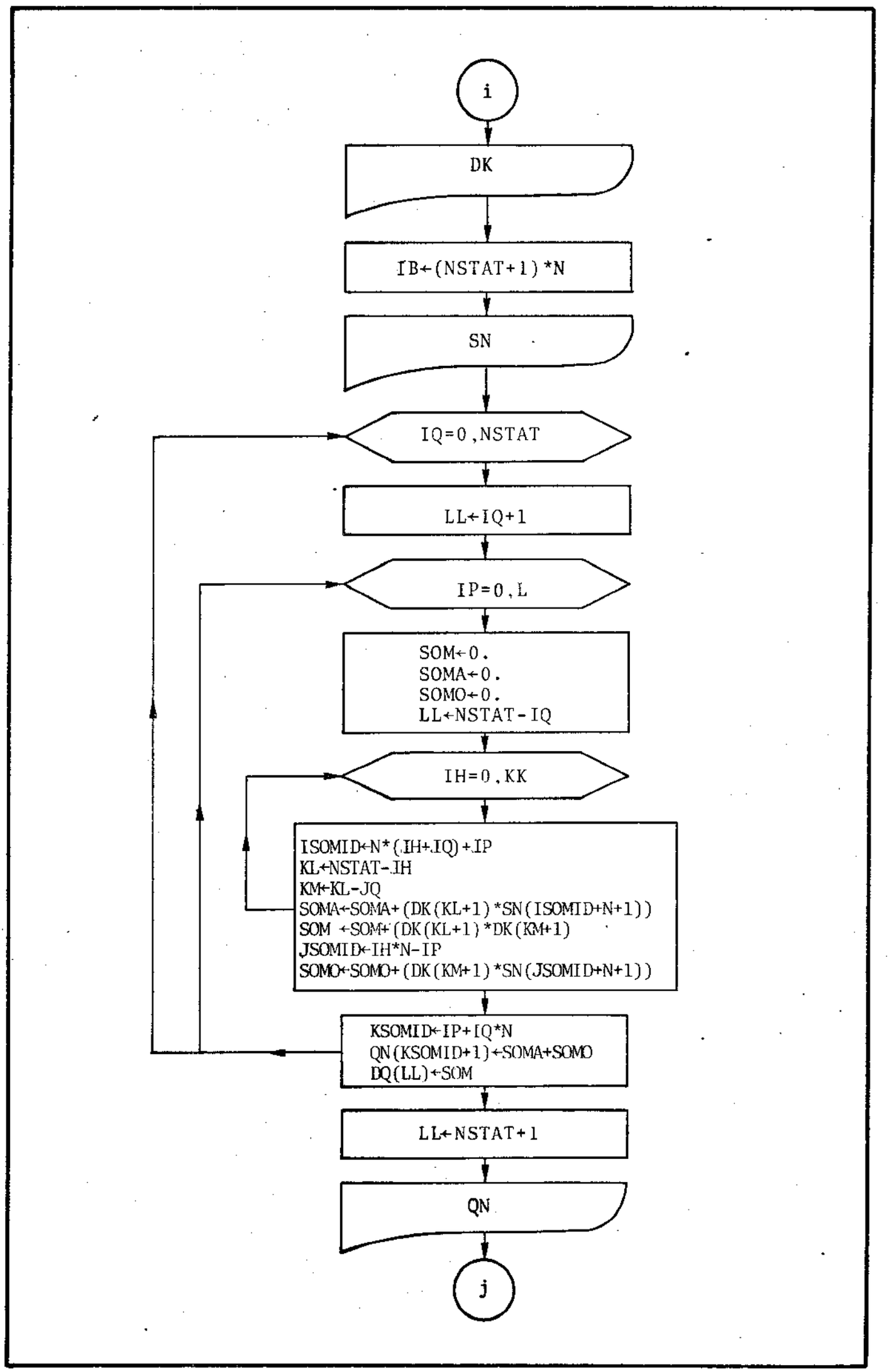

d
C1T
NSTAT,N,NVI,AU,Q,C2
SOMSUB
C2
NVI,NSTAT,N,N,AU,DINF,AU,Q,RO
SOPRMA
N,NSTAT,NSTAT,C1T,PIINF,DINF
PRMARE
N,NSTAT,N,DINF,C2,RINF
PRMARE
RO
RINF
i=1,NSTAT
j=1,NSTAT
F(i,j)←PI(i,j)-PIINF(i,j)
i=1,NSTAT
PIINF(i,i)←1.-PIINF(i,i)
e

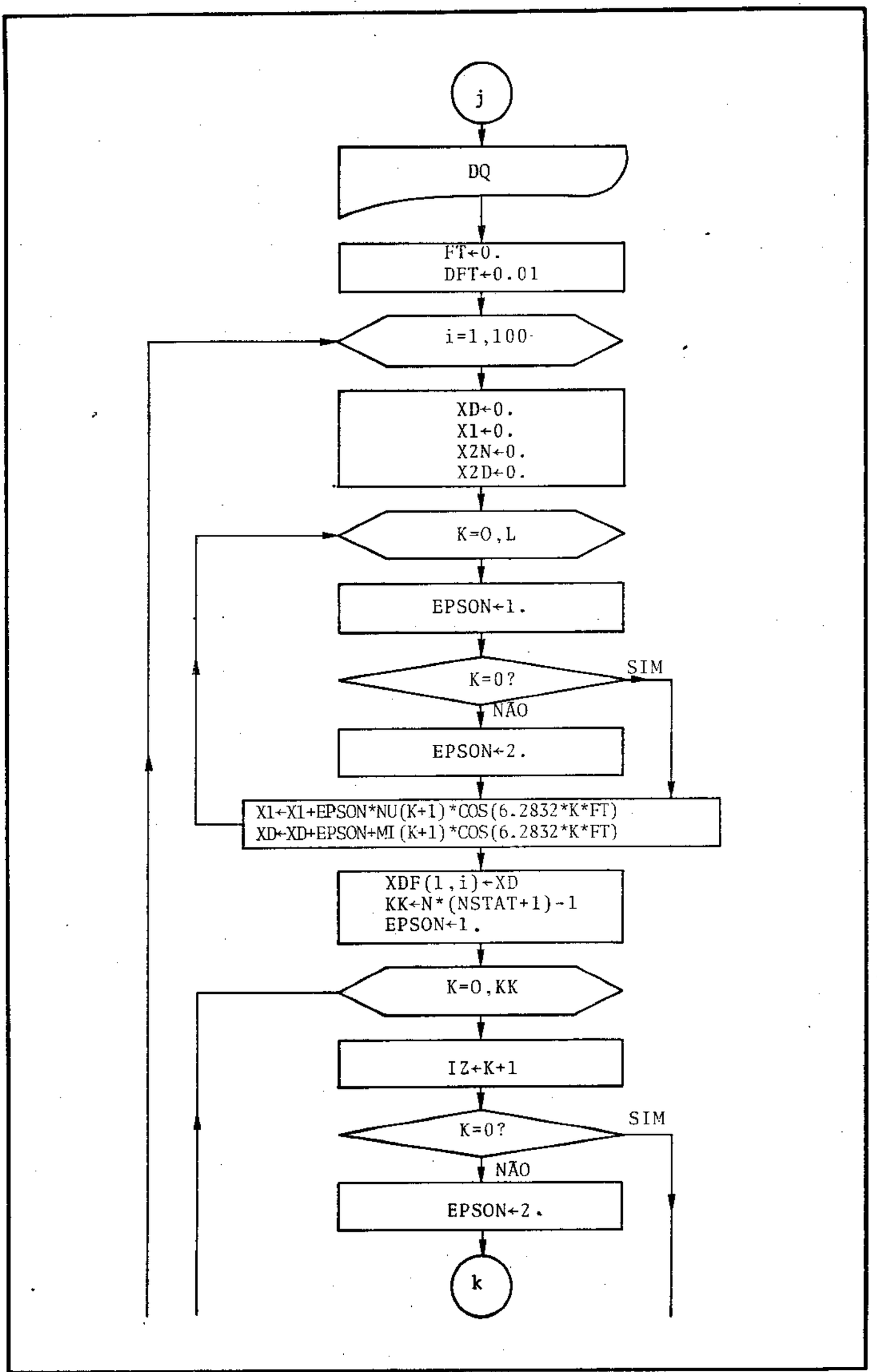

e
j=1,NSTAT
SIM
j=i?
NÃO
PIINF(i,j)←PIINF(i,j)
NLINHA←NVI*NSTAT
SOMA←0.
i=1,NLINHA
j=1,N
SOMA←SOMA+AU(i,j)
XMEDIA←SOMA/(NLINHA*N)
i=1,LINHA
j=1,N
XU(i,j)←XMEDIA
NVI,NSTAT,NSTAT,N,EU,DINF,XU,Q,C1P
SOPRMA
i=1,NSTAT
j=1,N
C1(j,i)←C1P(i,j)
f

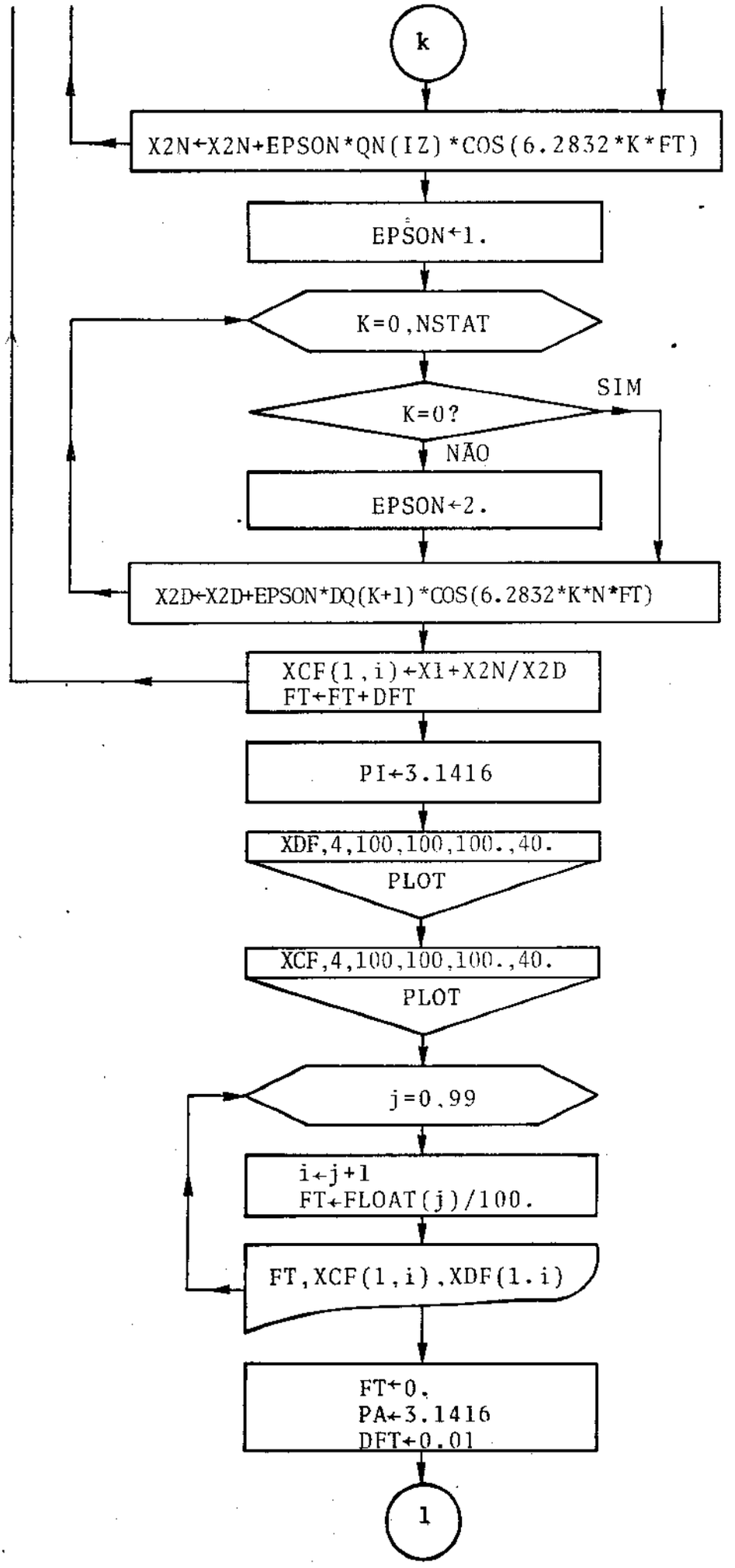

f
i=1,N
j=1,NSTAT
C1T(i,j)←C1T(i,j)-C1(i,j)
NSTAT,N,NVI,XU,Q,C1P
SOMSUB
i=1,NSTAT
j=1,N
C2(i,j)←C2(i,j)-C1P(i,j)
N,NSTAT,NSTAT,C1T,PIINF,C1T
PRMARE
L←N-1
k=0,L
SOM←0.
SOMA←0.
LL←N-k
i=1,KK
SOM←SOM+RO(i,i+k)-RINF(i,i+k)
SOMA←SOMA+RINF(1,i+k)
NU(k+1)←SOM
MI(k+1)←SOMA
NU
g

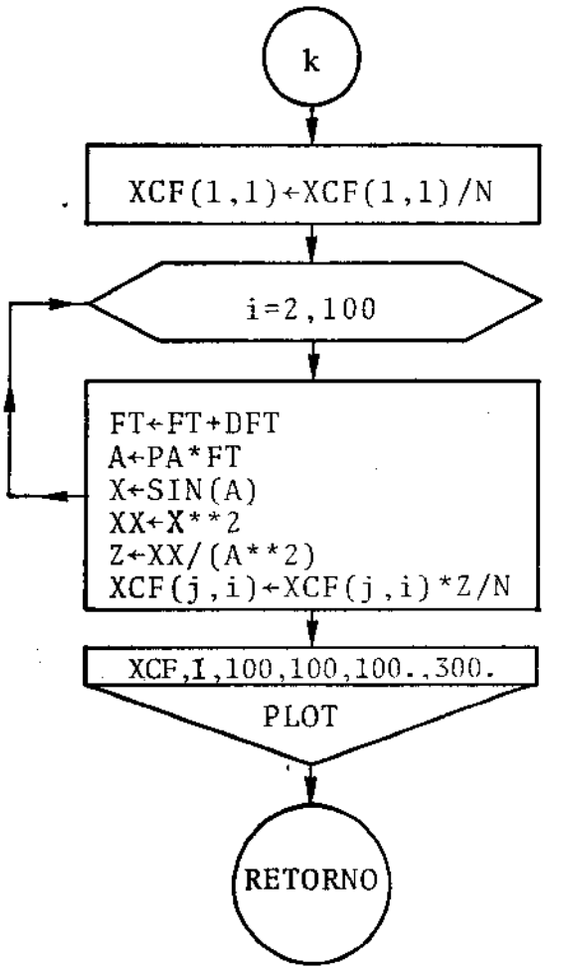

g
MI
DK(1)=1.
i=1,N
j=1,N
HKNXT(i,j)←0.
NSTAT,1.,GK
PRIDCT
K=0,NSTAT
KI←NSTAT-K-1
KJ←K+1
K=NSTAT?
V
F
NSTAT,NSTAT,NSTAT,F,GK,PROD
PRMARE
NSTAT,PROD,KJ,CONST
TRASO
DK(KJ+1)←CONST
N,NSTAT,NSTAT,C1T,GK,C3
PRMARE
N,NSTAT,N,C3,C2,HK
PRMARE
h


h
NSTAT,CONST,DKU
PRIDCT
NSTAT,NSTAT,PROD,DKU,FK
SOMAT
IP=0,L.
ISOMID←IP+KI*N
IZ←IP
ISOMID,N,IZ,HK,HKNXT,CONSTA
SOMAHK
SN(ISOMID+N+1)←CONSTA
V
K=NSTAT?
F
i=1,N
j=1,N
HKNXT(i,j)←HK(i,j)
i=1,N
j=1,N
HKNXT(i,j)←0.
F
K=NSTAT?
V
LL←NSTAT+1
i

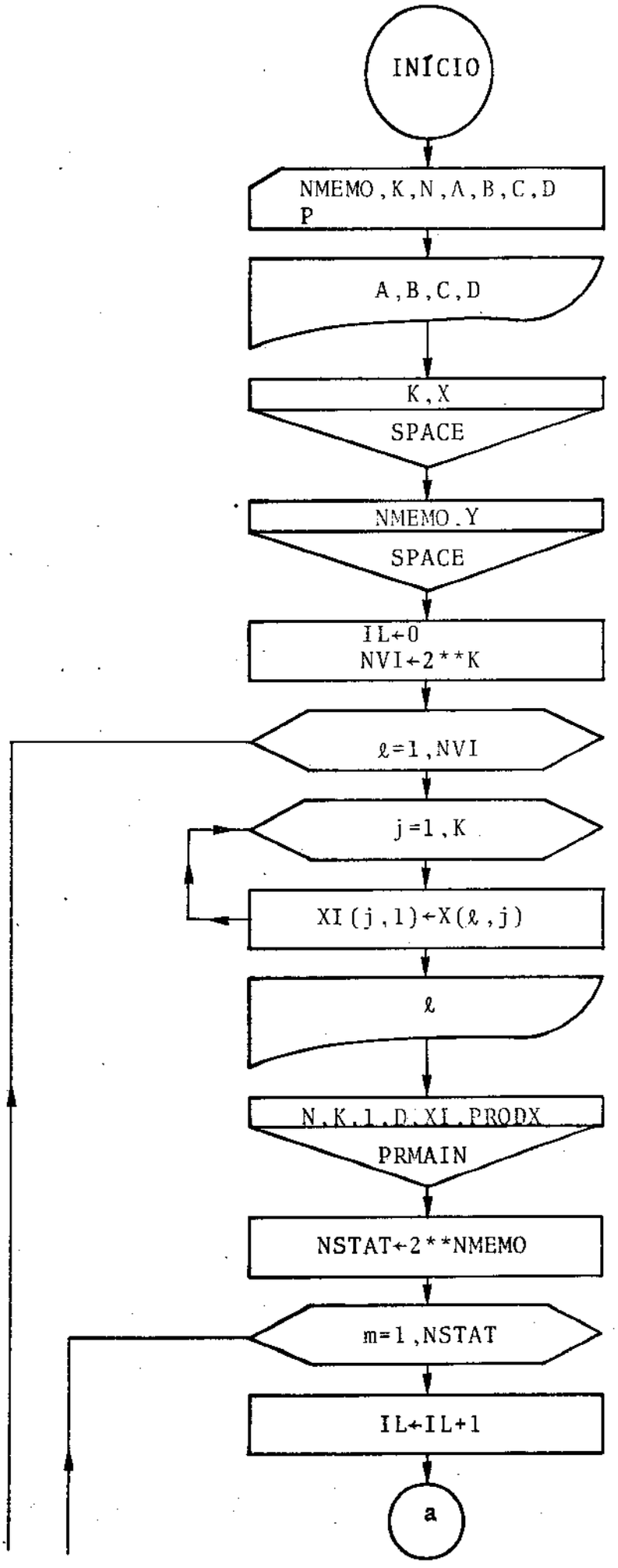

i
DK
IB←(NSTAT+1)*N
SN
IQ=0,NSTAT
LL←IQ+1
IP=0,L
SOM←0.
SOMA←0.
SOMO←0.
LL←NSTAT-IQ
IH=0,KK
ISOMID←N*(IH+IQ)+IP
KL←NSTAT-IH
KM←KL-JQ
SOMA←SOMA+(DK(KL+1)*SN(ISOMID+N+1))
SOM ←SOM+(DK(KL+1)*DK(KM+1)
JSOMID←IH*N-IP
SOMO←SOMO+(DK(KM+1)*SN(JSOMID+N+1))
KSOMID←IP+IQ*N
QN(KSOMID+1)←SOMA+SOMO
DQ(LL)←SOM
LL←NSTAT+1
QN
j

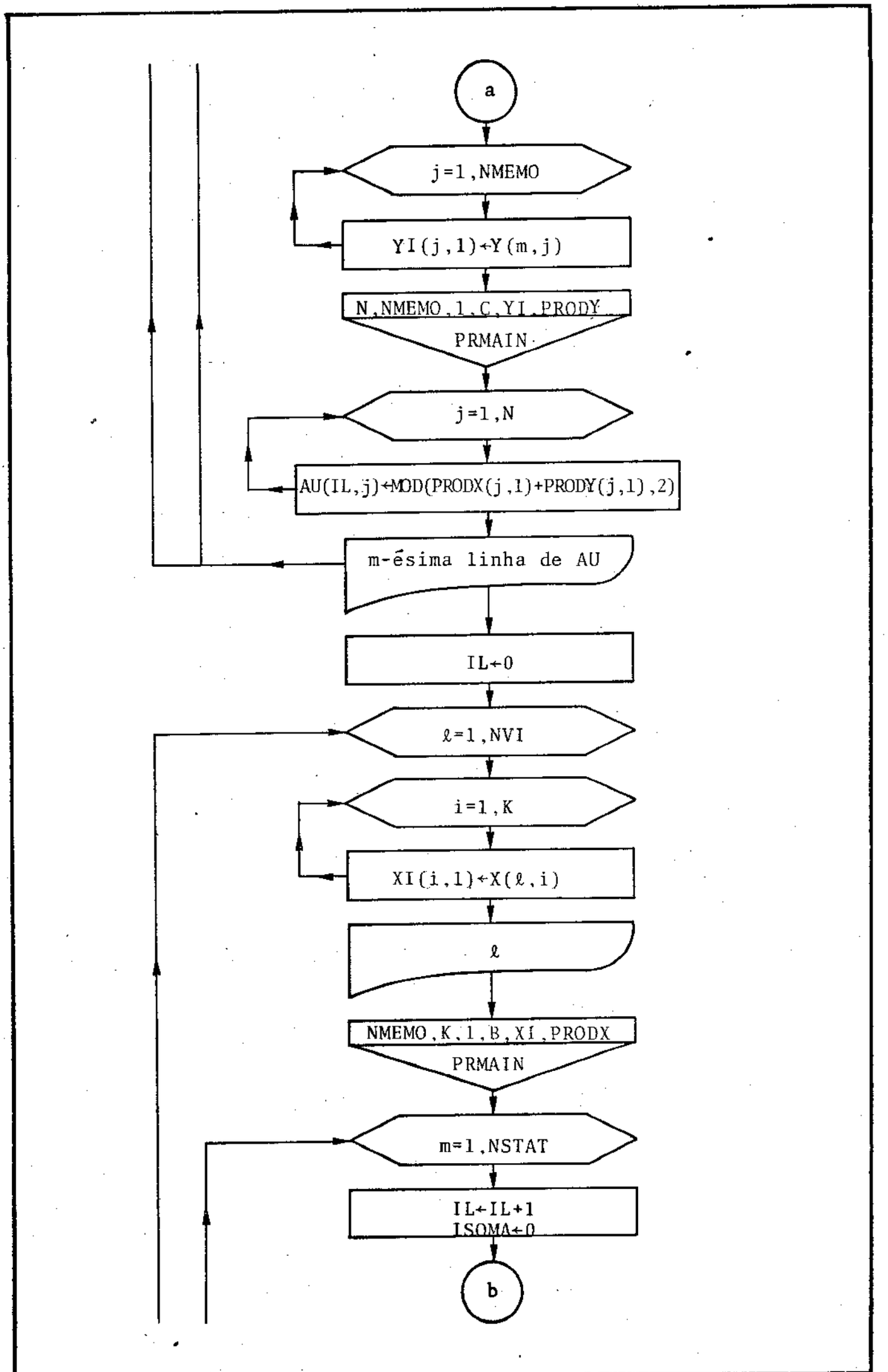

j
DQ
FT←0.
DFT←0.01
i=1,100
XD←0.
X1←0.
X2N←0.
X2D←0.
K=0,L
EPSON←1.
K=0?
SIM
NÃO
EPSON←2.
X1←X1+EPSON*NU(K+1)*COS(6.2832*K*FT)
XD←XD+EPSON+MI(K+1)*COS(6.2832*K*FT)
XDF(1,i)←XD
KK←N*(NSTAT+1)-1
EPSON←1.
K=0,KK
IZ←K+1
K=0?
SIM
NÃO
EPSON←2.
k

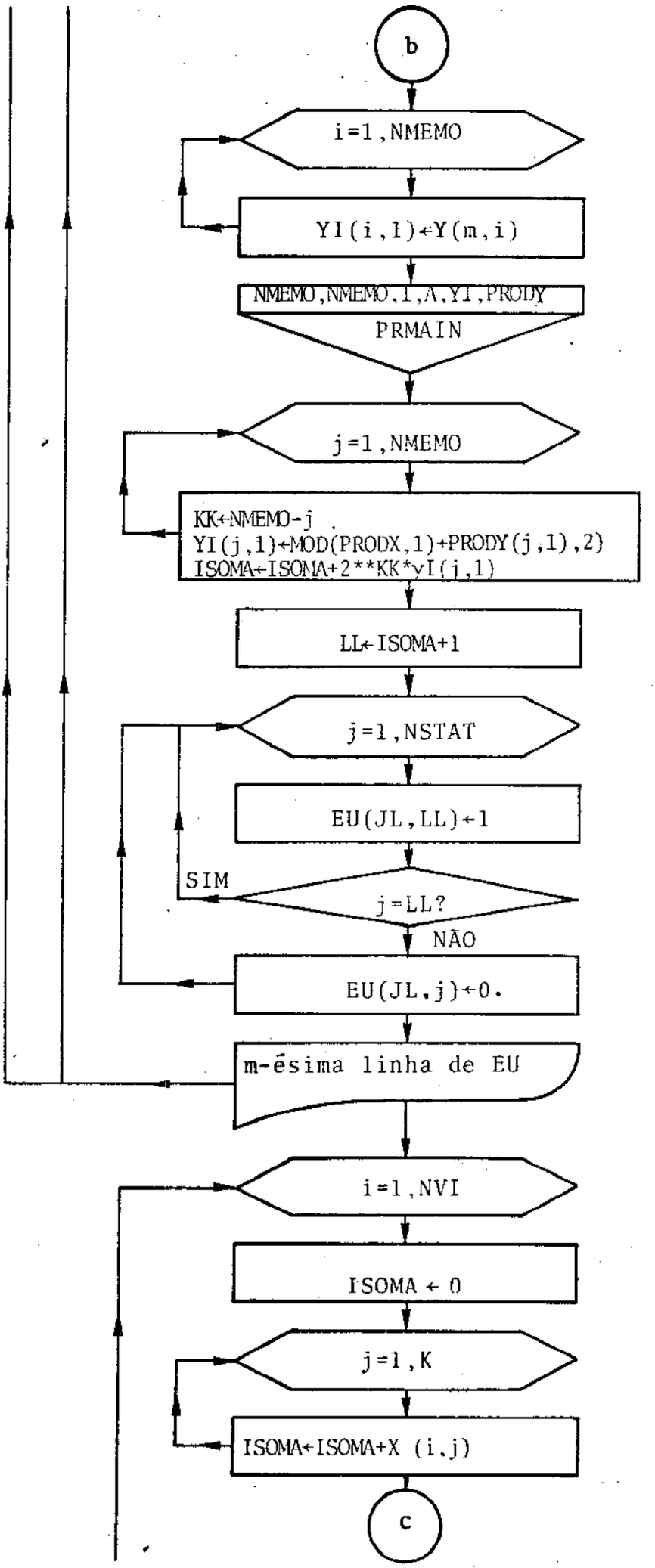

k
X2N←X2N+EPSON*QN(IZ)*COS(6.2832*K*FT)
EPSON←1.
K=0,NSTAT
K=0?
SIM
NÃO
EPSON←2.
X2D←X2D+EPSON*DQ(K+1)*COS(6.2832*K*N*FT)
XCF(1,i)←X1+X2N/X2D
FT←FT+DFT
PI←3.1416
XDF,4,100,100,100.,40.
PLOT
XCF,4,100,100,100.,40.
PLOT
j=0,99
i←j+1
FT←FLOAT(j)/100.
FT,XCF(1,i),XDF(1.i)
FT←0.
PA←3.1416
DFT←0.01
1

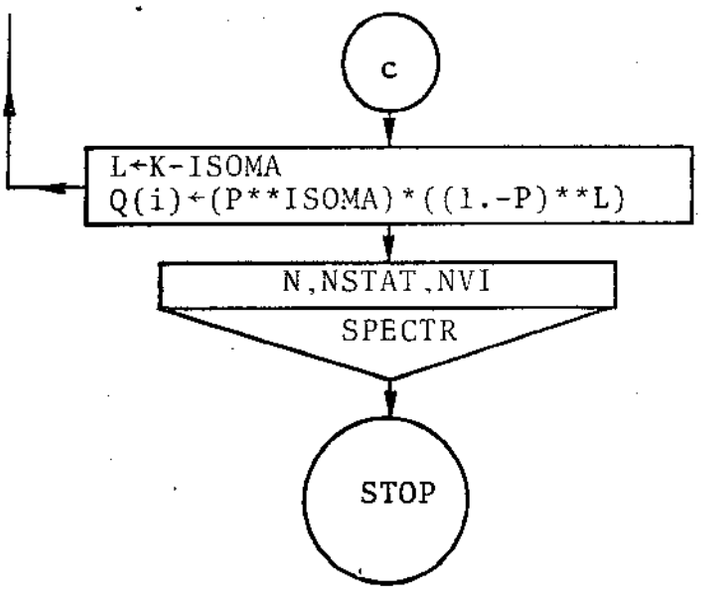

k
XCF(1,1)←XCF(1,1)/N
i=2,100
FT←FT+DFT
A←PA*FT
X←SIN(A)
XX←X**2
Z←XX/(A**2)
XCF(j,i)←XCF(j,i)*Z/N
XCF,I,100,100,100.,300.
PLOT
RETORNO

FLUXOGRAMA DO PROGRAMA CDCONV (CÓDIGOS CONVOLUCIONAIS)

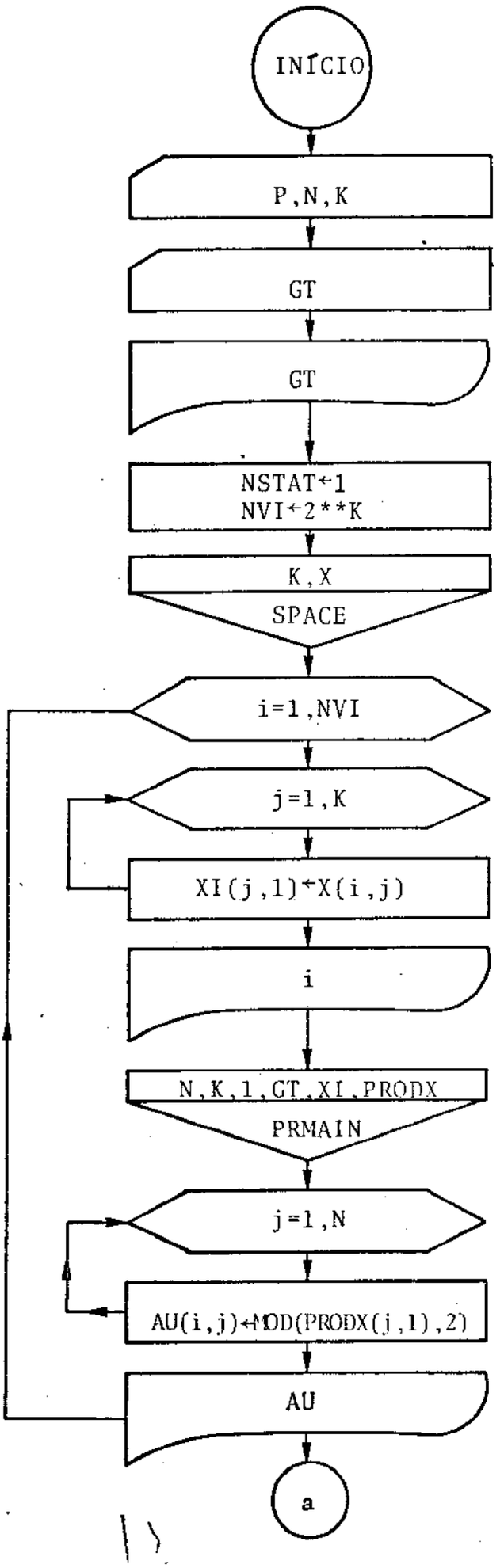


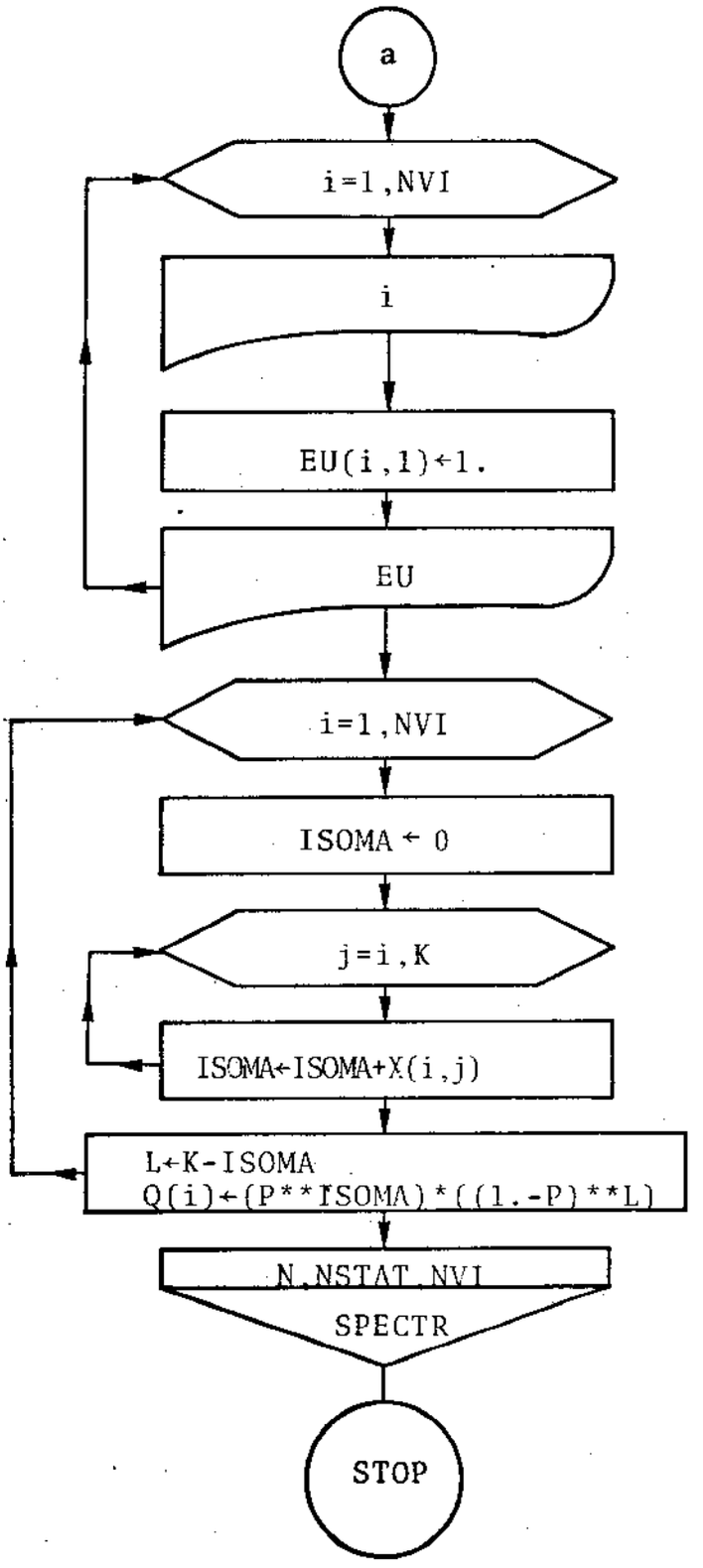

a
j=1,NMEMO
YI(j,1)←Y(m,j)
N,NMEMO,1,C,YI,PRODY
PRMAIN
j=1,N
AU(IL,j)←MOD(PRODX(j,1)+PRODY(j,1),2)
m-ésima linha de AU
IL←0
ℓ=1,NVI
i=1,K
XI(i,1)←X(ℓ,i)
ℓ
NMEMO,K,1,B,XI,PRODX
PRMAIN
m=1,NSTAT
IL←IL+1
ISOMA←0
b


b
i=1,NMEMO
YI(i,1)←Y(m,i)
NMEMO,NMEMO,1,A,YI,PRODY
PRMAIN
j=1,NMEMO
KK←NMEMO-j
YI(j,1)←MOD(PRODX,1)+PRODY(j,1),2)
ISOMA←ISOMA+2**KK*YI(j,1)
LL←ISOMA+1
j=1,NSTAT
EU(JL,LL)←1
SIM
j=LL?
NÃO
EU(JL,j)←0.
m-ésima linha de EU
i=1,NVI
ISOMA ← 0
j=1,K
ISOMA←ISOMA+X (i,j)
c

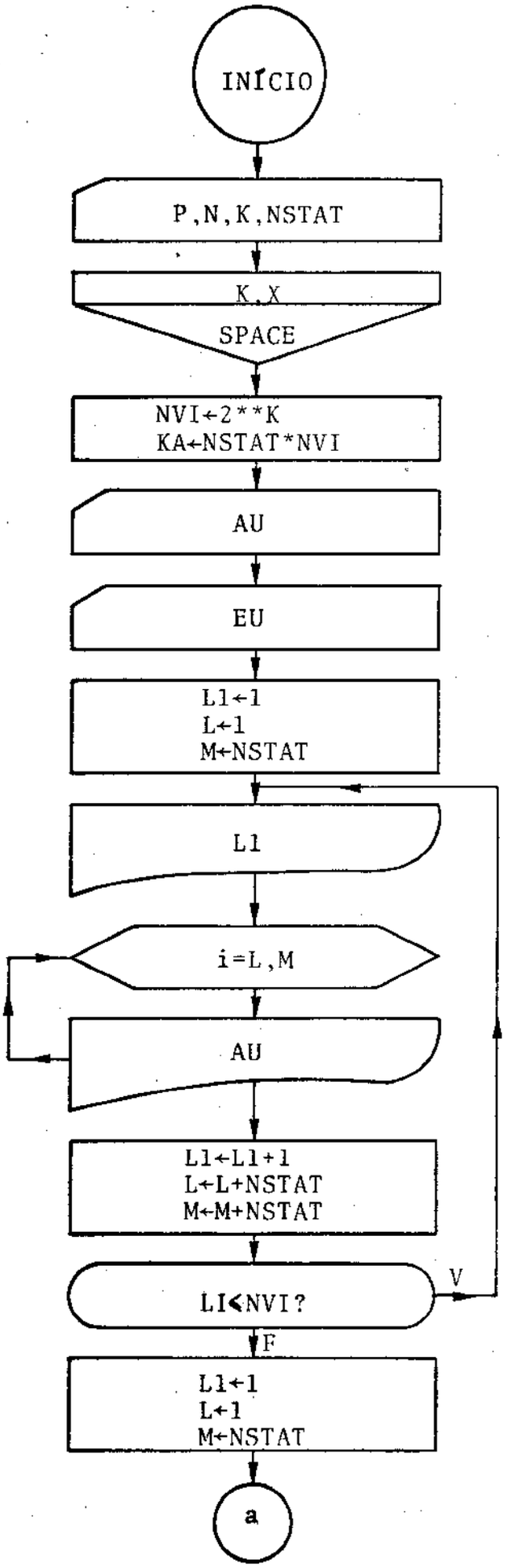

C
L←K-ISOMA
Q(i)←(P**ISOMA)*((1.-P)**L)
N,NSTAT,NVI
SPECTR
STOP

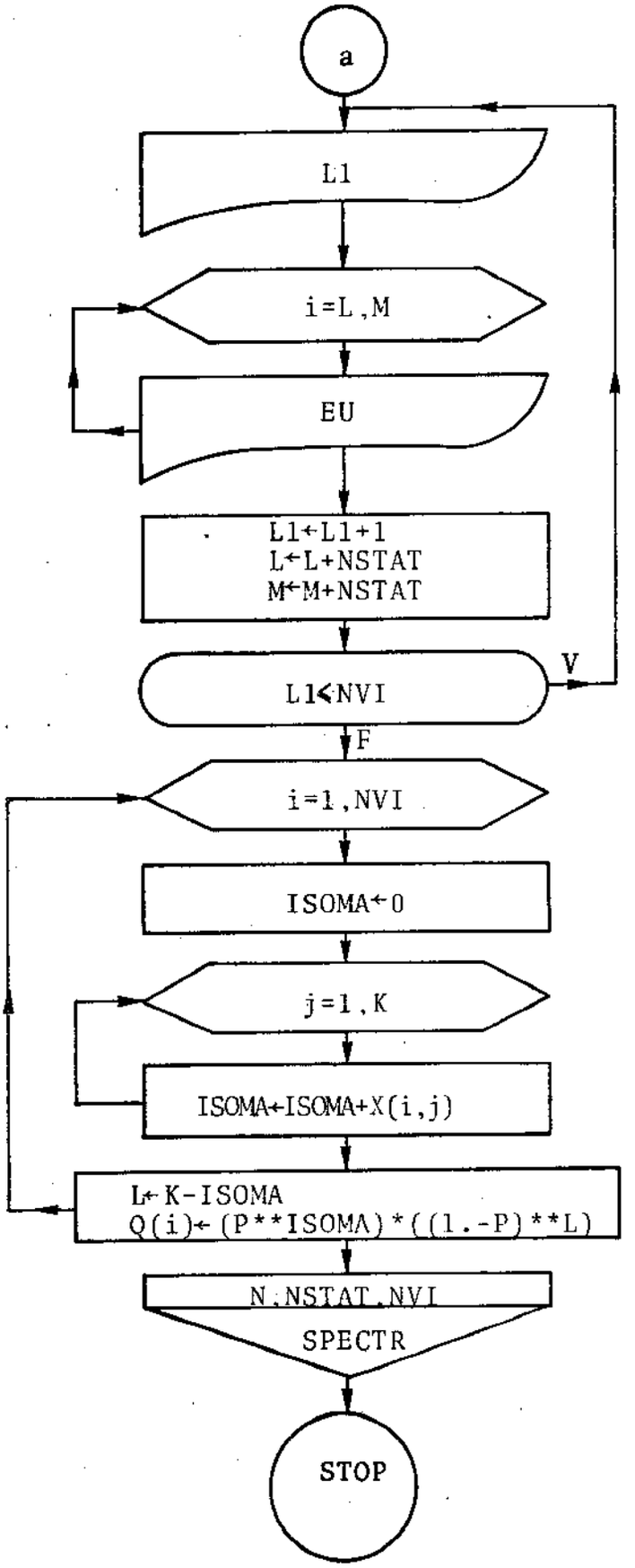
FLUXOGRAMA DO PROGRAMA CDBLLI (CODIGOS DE BLOCO LINEARES)




a
i=1,NVI
i
EU(i,1)←1.
EU
i=1,NVI
ISOMA ← 0
j=i,K
ISOMA←ISOMA+X(i,j)
L←K-ISOMA
Q(i)←(P**ISOMA)*((1.-P)**L)
N,NSTAT,NVI
SPECTR
STOP

FLUXOGRAMA DO PROGRAMA CDBLNL (CÓDIGOS DE BLOCO NÃO LINEARES)




a
L1
i=L,M
EU
L1←L1+1
L←L+NSTAT
M←M+NSTAT
L1≤NVI
V
F
i=1,NVI
ISOMA←0
j=1,K
ISOMA←ISOMA+X(i,j)
L←K-ISOMA
Q(i)←(P**ISOMA)*((1.-P)**L)
N,NSTAT,NVI
SPECTR
STOP

APÊNDICE C

LISTAGENS DOS PROGRAMAS E

SUB-PROGRAMAS IMPLEMENTADOS

```
      SUBROUTINE SOPRMA(NSUB,M,N,K,A,F,B,Q,S)
      DIMENSION A(96,6),B(96,6),F(6,6),S(6,6),AT(6,6)
     *,BT(6,6),P(6,6),Z(6,6),Q(32)
C     INICIALIZACAO DA MATRIZ S
      DO 5 I=1,N
      DO 10 J=1,K
      S(I,J)=0.
   10 CONTINUE
    5 CONTINUE
C     INICIALIZACAO DOS CONTADORES DAS LINHAS
      LI=1
      LF=M
C     INICIO DA REALIZACAO DAS OPERACOES
      DO 15 KONT=1,NSUB
C     TRANSPOZICAO DAS SUB-MATRIZES DE A
      I=1
      DO 20 I1=LI,LF
      J=1
      DO 25 J1=1,N
      AT(J,I)=Q(KONT)*A(I1,J1)
      J=J+1
   25 CONTINUE
      I=I+1
   20 CONTINUE
C     OBTENSAO DAS SUBMATRIZES DE B
      I=1
      DO 30 I2=LI,LF
      J=1
      DO 35 J2=1,K
      BT(I,J)=B(I2,J2)
      J=J+1
   35 CONTINUE
      I=I+1
   30 CONTINUE
      CALL PRMARE(N,M,M,AT,F,P)
      CALL PRMARE(N,M,K,P,BT,Z)
      CALL SOMAT(N,K,Z,S,S)
      LI=LI+M
      LF=LF+M
   15 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SOMAHK(ISOMID,N,IP,HK,HKNXT,SN)
      DIMENSION HK(6,6),HKNXT(6,6)
      SOM=0.
      SOMA=0.
      NN=N-IP
      DO 5 J=1,NN
      L=J+IP
      SOM=SOM+HK(L,J)
    5 CONTINUE
      IF(IP.EQ.0)GO TO 1
     DO 10 I=1,IP
      M=I+N-IP
      SOMA=SOMA+HKNXT(I,M)
   10 CONTINUE
    1 SN=SOM+SOMA
      RETURN
      END



      SUBROUTINE PRMAIN(M,N,K,JA,JB,JP)
      DIMENSION JA(8,8),JB(8,8),JP(8,8)
      DO 5 I=1,M
      DO 10 KK=1,K
      ISOM=0
      DO 15 J=1,N
      ISOM=ISOM+JA(I,J)*JB(J,KK)
   15 CONTINUE
      JP(I,KK)=ISOM
   10 CONTINUE
    5 CONTINUE
      RETURN
      END



      SUBROUTINE PRMARE(M,N,K,A,B,P)
      DIMENSION A(6,6),B(6,6),P(6,6)
      DO 5 I=1,M
      DO 10 KK=1,K
      SOMA=0.
      DO 15 J=1,N
      SOMA=SOMA+A(I,J)*B(J,KK)
   15 CONTINUE
      P(I,KK)=SOMA
   10 CONTINUE
    5 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SOMAT(M,N,A,B,C)
      DIMENSION A(6,6),B(6,6),C(6,6)
      DO 5 I=1,M
      DO 10 J=1,N
      C(I,J)=A(I,J)+B(I,J)
10    CONTINUE
 5    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PRIDCT(M,CONST,DIAG)
      DIMENSION DIAG(6,6)
      DO 5 I=1,M
      DIAG(I,I)=CONST
      DO 10 J=1,M
      IF(J-I)1,10,1
 1    DIAG(I,J)=0.
10    CONTINUE
 5    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE TRASO(M,FG,K,DK)
      DIMENSION FG(6,6)
      SOM=0.
      DO 5 I=1,M
      SOM=SOM+FG(I,I)
 5    CONTINUE
      DK=-SOM/K
      RETURN
      END
```

```
      SUBROUTINE SPACE(IDIM,MAT)
      DIMENSION IR(6),IQ(6),MAT(32,6)
      IW=21
      N=2**IDIM-1
      DO 5 L=0,N
      J=IDIM
      IQ(1)=L/2
      IR(1)=MOD(L,2)
      DO 10 I=1,IDIM
      KK=L+1
      MAT(KK,J)=IR(I)
      IQ(I+1)=IQ(I)/2
      IR(I+1)=MOD(IQ(I),2)
      J=J-1
10    CONTINUE
      WRITE(IW,2)L,(MAT(L+1,K),K=1,IDIM)
      TYPE 2,L,(MAT(L+1,K),K=1,IDIM)
 2    FORMAT(1H ,I4,4X,8(I1,1X))
 5    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SOMSUB(M,N,NSUB,X,CONST,SOM)
      DIMENSION X(96,6),CONST(32),SOM(6,6)
      DO 5 L=1,M
      KK=M-L
      LL=M*NSUB-KK
      DO 10 J=1,N
      K=0
      SOMA=0.
      NN=L
      DO 15 I=NN,LL,M
      K=K+1
      SOMA=SOMA+X(I,J)*CONST(K)
15    CONTINUE
      SOM(L,J)=SOMA
10    CONTINUE
 5    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PLOT(Y,M,NF,NS,XK,ESC)
      DIMENSION Y(1,100),LINHA(101),L(11),JL(2)
      DATA(JL(I),I=1,2)/1H*,1H./,JN,JP,JI,JBLANK,JZ/1H-,1H+,1HI,1H ,1H$/
      IW=21
      DO 99 I=1,101
      LINHA(I)=JBLANK
   99 N=0
      DO 101 I=1,11
      L(I)=10*I-110+NS
  101 CONTINUE
      WRITE(IW,105)(L(I),I=1,11)
  105 FORMAT(3X,11(I4,6X),6HY(1,I))
      GO TO 115
  110 IF(N/10-(N-1)/10)125,125,115
C     CONSTRUCAO DAS LINHAS DAS ORDENADAS DO GRAFICO
  115 ND=0
      DO 120 I=1,10
      ND=ND+1
      LINHA(ND)=JP
      DO 120 J=1,9
      ND=ND+1
  120 LINHA(ND)=JN
      LINHA(101)=JP
      IF(N)135,121,135
  121 WRITE(IW,170)N,LINHA
      GO TO 185
C     CONSTRUCAO DE LINHAS QUE CORRESPONDEM A ABSSISSA
  125 DO 130 I=1,101,10
      LINHA(I)=JI
  130 CONTINUE
C     MUDANCA DE VALORES NUMERICOS PARA VALORES ALFABETICOS
  135 DO 160 I=1,M
      XNS=NS
      JA=ESC*Y(I,N)+101.49999-XNS
      IF(JA-101)140,155,145
  140 IF(JA)150,150,155
  145 DINHA(101)=JZ
      GO TO 160
  150 LINHA(1)=JZ
      GO TO 160
  155 LINHA(JA)=JL(I)
  160 CONTINUE
C     IMPRESSAO DE UMA LINHA DE DADOS
      IF(N/10-(N-1)/10)175,175,165
  165 XL=N/XK
      WRITE(IW,170)XL,LINHA,Y(1,N)
  170 FORMAT(1X,F4.1,101A1,1X,1PE12.5)
      GO TO 185
  175 WRITE(IW,180)LINHA,Y(1,N)
  180 FORMAT(5X,101A1,1X,E12.5)
C     ZERANDO AS VARIAVEIS DE LINHA
  185 DO 190 I=1,101
      LINHA(I)=JBLANK
  190 CONTINUE
  195 N=N+1
      IF(N-NF)110,110,200
  200 RETURN
      END
```

```
      SUBROUTINE SPECTR(N,NSTAT,NVI)
      REAL PI(6,6),PIALT(6,6),VECTIN(7),SOL(7),DINF(6,6),
     *  PIINF(6,6),C1(6,6),C2(6,6),C3(6,6),C1T(6,6),RO(6,6),
     *  RINF(6,6),F(6,6),NU(7),MI(7),HK(6,6),HKNXT(6,6),SN(30),
     *  DKU(6,6),DK(7),Q(32),GK(6,6),PROD(6,6),QN(30),DQ(7),
     *  C1P(6,6),XDF(1,100),XCF(1,100),AU(96,6),EU(96,6),XU(96,6)
      COMMON AU,EU,Q
      IW=21
      IR=22
      WRITE(IW,77)(Q(I),I=1,NVI)
   77 FORMAT(//,' PROB. DOS VETORES DE INFORMACAO '/(1H0,8(F12.6,1X)))
C     CALCULO DA MATRIZ TRANSICAO DE ESTADO,PI
      CALL SOMSUB(NSTAT,NSTAT,NVI,EU,Q,PI)
      WRITE(IW,512)
  512 FORMAT(1H0,' MATRIZ TRANSICAO DE ESTADO PI')
      DO 515 I=1,NSTAT
      WRITE(IW,513)(PI(I,J),J=1,NSTAT)
  513 FORMAT(1H0,8(F12.6,1X))
  515 CONTINUE
      IF(NSTAT-1)50,40,50
   40 SOL(1)=1.
      GO TO 24
C     CALCULO DAS PROBABILIDADES ABSOLUTAS DOS ESTADOS
C     INICIO DO CALCULO.CONSTRUCAO DA MATRIZ DOS COEFICIENTES
   50 KI=NSTAT
  517 IF(PI(KI,1))516,514,516
  514 KI=KI-1
      GO TO 517
  516 PIALT(1,1)=PI(1,1)-1.-PI(KI,1)
      DO 70 J=2,NSTAT
      IF(J.EQ.KI)GO TO 518
      PIALT(1,J)=PI(J,1)-PI(KI,1)
      GO TO 70
  518 PIALT(1,J)=0.
   70 CONTINUE
      WRITE(IW,13)(PIALT(1,J),J=1,NSTAT)
   13 FORMAT(1H0,'MATRIZ DOS COEFICIENTES'/(1H ,12(F10.6,1X)))
      DO 75 I=2,NSTAT
      PIALT(I,I)=PI(I,I)-1.
      DO 80 J=1,NSTAT
      IF(J-I)14,80,14
   14 PIALT(I,J)=PI(J,I)
   80 CONTINUE
      WRITE(IW,16)(PIALT(I,J),J=1,NSTAT)
   16 FORMAT(1H ,12(F10.6,1X))
   75 CONTINUE
C     CONSTRUCAO DO VETOR DAS CONSTANTES INDEPENDENTES
      VECTIN(1)=-PI(KI,1)
      DO 85 J=2,NSTAT
      VECTIN(J)=0.
   85 CONTINUE
      WRITE(IW,17)(VECTIN(I),I=1,NSTAT)
   17 FORMAT(1H0,'VETOR DAS CONST. INDEP.'/(1H ,12(F10.6,1X)))
C     INICIO DA RESOL. DO SISTEMA-METODO DE GAUSS-SEIDEL
C     NORMALIZACAO DA MATRIZ DOS COEFICIENTES
      LL=100
      CONV=0.0001
      DO 90 I=1,NSTAT
      TEMP=PIALT(I,I)
      VECTIN(I)=VECTIN(I)/TEMP
      SOL(I)=VECTIN(I)
*
```

```
      DO 91 J=1,NSTAT
      PIALT(I,J)=PIALT(I,J)/TEMP
   91 CONTINUE
   90 CONTINUE
C     INICIO DAS ITERACOES
      DO 95 KOUNT=1,LL
C     INDICADOR DE CONVERGENCIA
      IND=0
      DO 100 I=1,NSTAT
      AUX=VECTIN(I)
      DO 105 J=1,NSTAT
      IF(J-I)18,105,18
   18 AUX=AUX-PIALT(I,J)*SOL(J)
  105 CONTINUE
      IF(ABS(SOL(I)-AUX)-CONV)19,19,21
   21 IND=1
   19 SOL(I)=AUX
  100 CONTINUE
      WRITE(IW,22)KOUNT,(SOL(I),I=1,NSTAT)
   22 FORMAT(1H0,' SOL. APOS',I3,' ITERACOES'/(1H0,12(F10.6,1X)))
C     TESTE PARA A CONVERGENCIA
      IF(IND)95,24,95
   95 CONTINUE
C     CASO DE DIVERGENCIA
      TYPE 23,CONV,LL
      WRITE(IW,23)CONV,LL
   23 FORMAT(1H0,'A CONVERG. DESEJADA DE',F10.8,'NAO FOI OBTIDA EM
     * ',I3,' ITERACOES')
      GO TO 777
C     CALCULO DA MATRIZ DAS PROBABILIDADES DE K TRANSICOES PARA
C     K TENDENDO A INFINITO
   24 WRITE(IW,27)
      DO 110 I=1,NSTAT
      DINF(I,I)=SOL(I)
      DO 115 J=1,NSTAT
      PIINF(J,I)=SOL(I)
      IF(J-I)26,115,26
   26 DINF(I,J)=0.
  115 CONTINUE
   27 FORMAT(///,60X'MATRIZ PIINF')
  110 CONTINUE
      DO 500 I=1,NSTAT
      WRITE(IW,16)(PIINF(I,J),J=1,NSTAT)
  500 CONTINUE
C     CONSTRUCAO DA MATRIZ C1
      CALL SOPRMA(NVI,NSTAT,NSTAT,N,EU,DINF,AU,Q,C1)
C     CALCULO DA MATRIZ C1T
      DO 130 I=1,NSTAT
      DO 135 J=1,N
      C1T(J,I)=C1(I,J)
  135 CONTINUE
  130 CONTINUE
      WRITE(IW,28)
      DO 140 I=1,N
      WRITE(IW,16)(C1T(I,J),J=1,NSTAT)
  140 CONTINUE
   28 FORMAT(///,60X'MATRIZ C1T')
C     CONSTRUCAO DA MATRIZ C2
      CALL SOMSUB(NSTAT,N,NVI,AU,Q,C2)
      WRITE(IW,29)
   29 FORMAT(///,20X'MATRIZ C2')
*
```

```
      DO 145 I=1,NSTAT
      WRITE(IW,16)(C2(I,J),J=1,N)
145   CONTINUE
C     CALCULO DA MATRIZ RO
      CALL SOPRMA(NVI,NSTAT,N,N,AU,DINF,AU,Q,RO)
C     CONSTRUCAO DA MATRIZ RINF
      CALL PRMARE(N,NSTAT,NSTAT,C1T,PIINF,DINF)
      CALL PRMARE(N,NSTAT,N,DINF,C2,RINF)
      WRITE(IW,31)
 31   FORMAT(///,' MATRIZ RO')
      DO 160 IJ=1,N
      WRITE(IW,32)(RO(IJ,J),J=1,N)
 32   FORMAT(1H ,6(F11.6,1X))
160   CONTINUE
      WRITE(IW,11)
 11   FORMAT(///,' MATRIZ RINF')
      DO 5 I=1,N
      WRITE(IW,32)(RINF(I,J),J=1,N)
  5   CONTINUE
C     CONSTRUCAO DA MATRIZ F
      DO 165 IJ=1,NSTAT
      DO 170 J=1,NSTAT
      F(IJ,J)=PI(IJ,J)-PIINF(IJ,J)
170   CONTINUE
165   CONTINUE
C     ALTERACAO DA MATRIZ PIINF PARA U-PIINF
      DO 175 IJ=1,NSTAT
      PIINF(IJ,IJ)=1.-PIINF(IJ,IJ)
      DO 180 J=1,NSTAT
      IF(J-IJ)33,180,33
 33   PIINF(IJ,J)=-PIINF(IJ,J)
180   CONTINUE
175   CONTINUE
      NLINHA=NVI*NSTAT
      SOMA=0.
      DO 300 I=1,NLINHA
      DO 300 J=1,N
      SOMA=SOMA+AU(I,J)
300   CONTINUE
      XMEDIA=SOMA/(NLINHA*N)
      DO 305 I=1,NLINHA
      DO 305 J=1,N
      XU(I,J)=XMEDIA
305   CONTINUE
      CALL SOPRMA(NVI,NSTAT,NSTAT,N,EU,DINF,XU,Q,C1P)
      DO 310 I=1,NSTAT
      DO 310 J=1,N
      C1(J,I)=C1P(I,J)
310   CONTINUE
      DO 315 I=1,N
      DO 315 J=1,NSTAT
      C1T(I,J)=C1T(I,J)-C1(I,J)
315   CONTINUE
      CALL SOMSUB(NSTAT,N,NVI,XU,Q,C1P)
      DO 320 I=1,NSTAT
      DO 320 J=1,N
      C2(I,J)=C2(I,J)-C1P(I,J)
320   CONTINUE
C     CALCULO DO FATOR C1T=C1T*PIINF
      CALL PRMARE(N,NSTAT,NSTAT,C1T,PIINF,C1T)
C    CALCULO DOS COEFICIENTES MI E NU
*
```

```
      L=N-1
      DO 185 K=0,L
      SOM=0.
      SOMA=0.
      KK=N-K
      DO 190 I=1,KK
      SOM=SOM+RO(I,I+K)-RINF(I,I+K)
      SOMA=SOMA+RINF(I,I+K)
  190 CONTINUE
      NU(K+1)=SOM
      MI(K+1)=SOMA
  185 CONTINUE
      WRITE(IW,34)(NU(I5),I5=1,N)
   34 FORMAT(1H0,' COEFICIENTES NU'/(1H0,4(F12.6,1X)))
      WRITE(IW,9)(MI(I),I=1,N)
    9 FORMAT(///,' COEFICIENTES MI'/(1H0,6(F12.6,1X)))
C     CALCULO DOS COEFICIENTES DK E SN
      DK(1)=1.
C     INICIALIZACAO DA MATRIZ HKNXT
      DO 195 I=1,N
      DO 200 J=1,N
      HKNXT(I,J)=0.
  200 CONTINUE
  195 CONTINUE
C     INICIALIZACAO DA MATRIZ GK
      CALL PRIDCT(NSTAT,1.,GK)
C     CALCULO DAS MATRIZES HK
      DO 205  K=0,NSTAT
      KI=NSTAT-K-1
      KJ=K+1
      IF(K.EQ.NSTAT)GO TO 111
C     CONSTRUCAO DOS COEFICIENTES DK
      CALL PRMARE(NSTAT,NSTAT,NSTAT,F,GK,PROD)
      CALL TRASO(NSTAT,PROD,KJ,CONST)
      DK(KJ+1)=CONST
C     CONSTRUCAO DO NOVO HK
      CALL PRMARE(N,NSTAT,NSTAT,C1T,GK,C3)
      CALL PRMARE(N,NSTAT,N,C3,C2,HK)
C     COSTRUCAO DO NOVO GK
      CALL PRIDCT(NSTAT,CONST,DKU)
      CALL SOMAT(NSTAT,NSTAT,PROD,DKU,GK)
C     CALCULO DOS COEFICIENTES NS
  112 DO 210 IP=0,L
      ISOMID=IP+KI*N
      IZ=IP
      CALL SOMAHK(ISOMID,N,IZ,HK,HKNXT,CONSTA)
      SN(ISOMID+N+1)=CONSTA
  210 CONTINUE
C     SUBSTITUICAO DE HKNXT PELO HK ATUAL
      IF(K.EQ.NSTAT)GO TO 205
      DO 215 I=1,N
      DO 220 J=1,N
      HKNXT(I,J)=HK(I,J)
  220 CONTINUE
  215 CONTINUE
      GO TO 205
  111 DO 225 I=1,N
      DO 225 J=1,N
      HK(I,J)=0.
  225 CONTINUE
      IF(K.EQ.NSTAT)GO TO 112
*
```

```
  205   CONTINUE
        LL=NSTAT+1
        WRITE(IW,999)(DK(I),I=1,LL)
  999   FORMAT(1H0,' COEFICIENTES DK'/1H0,5(F12.6,2X))
        WRITE(IW,36)
   36   FORMAT(///,23X'COEFICIENTES NS')
        IB=(NSTAT+1)*N
        WRITE(IW,37)(SN(J),J=1,IB)
   37   FORMAT(1H ,7(F12.6))
C       CALCULO DOS COEFICIENTES DQ E QN
        DO 240 IQ=0,NSTAT
        LL=IQ+1
        DO 245 IP=0,L
        SOM=0.
        SOMA=0.
        SOMQ=0.
        KK=NSTAT-IQ
        DO 250 IH=0,KK
        ISOMID=N*(IH+IQ)+IP
        KL=NSTAT-IH
        KM=NSTAT-(IH+IQ)
        SOMA=SOMA+(DK(KL+1)*SN(ISOMID+N+1))
        SOM=SOM+(DK(KL+1)*DK(KM+1))
        JSOMID=IH*N-IP
        SOMQ=SOMQ+(DK(KM+1)*SN(JSOMID+N+1))
  250   CONTINUE
        KSOMID=IP+IQ*N
        QN(KSOMID+1)=SOMA+SOMQ
        DQ(LL)=SOM
  245   CONTINUE
  240   CONTINUE
C       IMPRESSAO DOS COEFICIENTES NK
        LL=NSTAT+1
        WRITE(IW,38)
   38   FORMAT(///,30X'COEFICIENTES NK')
        WRITE(IW,37)(QN(J),J=1,IB)
C       IMPRESSAO DOS COEFICIENTES DQ
        WRITE(IW,39)
   39   FORMAT(///,30X'COEFICIENTES DQ')
        WRITE(IW,37)(DQ(I),I=1,LL)
C       INICIO DO CALCULO DE 100 PONTOS DOS COMPONENTES CONTINUOS
C       E DISCRETO DO ESPECTRO DE POTENCIA DO CODIGO
        FT=0.
        DFT=0.01
        DO 260 I=1,100
        XD=0.
        X1=0.
        X2N=0.
        X2D=0.
        DO 265 K=0,L
        EPSON=1.
        IF(K)41,42,41
   41   EPSON=2.
   42   X1=X1+EPSON*NU(K+1)*COS(6.28*K*FT)
       XD=XD+EPSON*MI(K+1)*COS(6.28*K*FT)
  265   CONTINUE
        XDF(1,I)=XD
        KK=N*(NSTAT+1)-1
        EPSON=1.
        DO 270 K=0,KK
        IZ=K+1
*
```

```
      IF(K)43,44,43
   43 EPSON=2.
   44 X2N=X2N+(EPSON*QN(IZ)*COS(6.28*K*FT))
  270 CONTINUE
      EPSON=1.
      DO 275 K=0,NSTAT
      IF(K)46,47,46
   46 EPSON=2.
   47 X2D=X2D+EPSON*DQ(K+1)*COS(6.28*K*N*FT)
  275 CONTINUE
      XCF(1,I)=X1+X2N/X2D
      FT=FT+DFT
  260 CONTINUE
C     CONSTRUCAO DOS GRAFICOS DE XC(F) E XD(F)
      WRITE(IW,48)
   48 FORMAT(1H1,80X'XD(F) NORMALIZADO')
      CALL PLOT(XDF,1,100,100,100.,80.)
      WRITE(IW,49)
   49 FORMAT(1H1,80X,'XC(F) NORMALIZADO')
      CALL PLOT(XCF,1,100,100,100.,40.)
      WRITE(IW,51)
   51 FORMAT(1H1,' FT'10X'XC(F)'10X'XD(F)')
      DO 280 J=0,99
      I=J+1
      FT=FLOAT(J)/100.
      WRITE(IW,52)FT,XCF(1,I),XDF(1,I)
   52 FORMAT(1H0,F5.3,5X,F12.8,3X,F12.8)
  280 CONTINUE
      XCF(1,1)=XCF(1,1)/N
      FT=0.
      PA=3.14159
      DFT=0.01
      DO 285 I=2,100
      FT=FT+DFT
      A=PA*FT
      X=SIN(A)
      XX=X**2
      Z=XX/(A**2)
      XCF(1,I)=XCF(1,I)*Z/N
  285 CONTINUE
      WRITE(IW,53)
   53 FORMAT(1H1,////,80X,'XC(F)*E(F)/T')
      CALL PLOT(XCF,1,100,100,100.,80.)
  777 RETURN
      END
*
```

```
      ESTE PROGRAMA DETERMINA OS COMPONENTES CONTINU E DISCRE-
      TOS,NORMALIZADOS,DO ESPECTRO DE POTENCIA DE UM CODIGO CON-
      VOLUCIONAL
      REAL 0(32),AU(96,6),EU(96,6)
      INTEGER A(8,8),B8,8),C(8,8),D(8,8),X(32,6),
     *Y(32,6),XI(8,8),YI(8,8),PRODX(8,8),PRODY(8,8)
      COMMON AU,EU,0
      LEITURA DAS CARACTERISTICAS DO CODIFICADOR
      IW=21
      IR=22
      READ(IR,13)P
      READ(IR,1)NMEMO,K,N,((A(I,J),J=1,NMEMO),I=1,NMEMO),((B(I,J),
     *J=1,K),I=1,NMEMO),((C(I,J),J=1,NMEMO),I=1,N),((D(I,J),J=1,
     **K),I=1,N)
    1 FORMAT(10(I1,1X))
      WRITE(IW,2)
    2 FORMAT(1H0,'CARACTERIZACAO DO CODIFICADOR'/(1H0,'Y=A*Y+B*X')
     */(1H ,'Z=C*Y+D*X'))
      WRITE(IW,3)
    3 FORMAT(///,4X'MATRIZ A')
      DO 5 I=1,NMEMO
      WRITE(IW,4)(A(I,J),J=1,NMEMO)
    5 CONTINUE
      WRITE(IW,21)
   21 FORMAT(///,4X'MATRIZ B')
      DO 70 I=1,NMEMO
      WRITE(IW,4)(B(I,J),J=1,K)
   70 CONTINUE
      WRITE(IW,22)
   22 FORMAT(///,4X'MATRIZ C')
      DO 75 I=1,N
      WRITE(IW,4)(C(I,J),J=1,NMEMO)
   75 CONTINUE
    4 FORMAT(1H ,8(I1,1X))
      WRITE(IW,6)
    6 FORMAT(///,4X'MATRIZ D')
      DO 10 I=1,N
      WRITE(IW,4)(D(I,J),J=1,K)
   10 CONTINUE
      CONSTRUCAO DAS SUB-MATRIZES AU
      WRITE(IW,23)
   23 FORMAT(///,' ESPACO DE INFORMACAO')
      CALL SPACE(K,X)
      WRITE(IW,24)
   24 FORMAT(///,' ESPACO DE ESTADO')
      CALL SPACE(NMEMO,Y)
      IL=0
      NVI=2**K
      DO 15 L=1,NVI
      DO 20 J=1,K
      XI(J,1)=X(L,J)
   20 CONTINUE
      WRITE(IW,7)L
      TYPE 7,L
    7 FORMAT(///,' MATRIZ A'I2)
      CALL PRMAIN(N,K,1,D,XI,PRODX)
      NSTAT=2**NMEMO
      DO 25 M=1,NSTAT
      IL=IL+1
```

```
      DO 30 J=1,NMEMO
      YI(J,1)=Y(M,J)
30    CONTINUE
      CALL PRMAIN(N,NMEMO,1,C,YI,PRODY)
      DO 35 J=1,N
      AU(IL,J)=FLOAT(MOD(PRODX(J,1)+PRODY(J,1),2))
35    CONTINUE
      SAIDA DAS LINHAS DE AU
      TYPE 8,(AU(IL,J),J=1,N)
      WRITE(IW,8)(AU(IL,J),J=1,N)
 8    FORMAT(1H ,8(F3.1,2X))
25    CONTINUE
15    CONTINUE
      CONSTRUCAO DAS SUB-MATRIZES EU
      IL=0
      DO 40 L=1,NVI
      DO 45 I=1,K
      XI(I,1)=X(L,I)
45    CONTINUE
      TYPE 9,L
      WRITE(IW,9)L
 9    FORMAT(///,27X'MATRIZ E'I2)
      CALL PRMAIN(NMEMO,K,1,B,XI,PRODX)
      DO 50 M=1,NSTAT
      IL=IL+1
      DO 55 I=1,NMEMO
      YI(I,1)=Y(M,I)
55    CONTINUE
      CALL PRMAIN(NMEMO,NMEMO,1,A,YI,PRODY)
      ISOMA=0
      DO 60 J=1,NMEMO
      KK=NMEMO-J
      YI(J,1)=MOD(PRODX(J,1)+PRODY(J,1),2)
      ISOMA=ISOMA+2**KK*YI(J,1)
60    CONTINUE
      LL=ISOMA+1
      DO 65 J=1,NSTAT
      EU(IL,LL)=1.
      IF(J-LL)11,65,11
11    EU(IL,J)=0.
65    CONTINUE
      SAIDA DAS LINHAS DE EU
      TYPE 12,(EU(IL,J),J=1,NSTAT)
      WRITE(IW,12)(EU(IL,J),J=1,NSTAT)
12    FORMAT(1H ,32(F3.1,2X))
50    CONTINUE
40    CONTINUE
      CALCULO DAS PROBABILIDADES DOS VETORES DE INFORMACAO
13    FORMAT(F4.2)
      DO 80 I=1,NVI
      ISOMA=0
      DO 85 J=1,K
      ISOMA=ISOMA+X(I,J)
85    CONTINUE
      L=K-ISOMA
      Q(I)=(P**ISOMA)*((1.-P)**L)
80    CONTINUE
      CALL SPECTR(N,NSTAT,NVI)
      STOP
      END
```

```
      ESTE PROGRAMA DETERMINA OS COMPONENTES CONTINUO E DISCRETO,
      NORMALIZADOS,DO ESPECTRO DE POTENCIA DE UM CODIGO DE BLOCO
      LINEAR
      REAL Q(16)
      INTEGER GT(8,8),X(16,4),XI(8,1),PRODX(8,1),AU(64,8),EU(64,8)
      COMMON AU,EU,Q
      IW=21
      IR=22
      READ(IR,1)N,K,P
      READ(IR,2)((GT(I,J),J=1,K),I=1,N)
    1 FORMAT(I2,F8.6)
    2 FORMAT(8I1)
      WRITE(IW,7)
    7 FORMAT(1H0,' MATRIZ DE TRANSFERENCIA DO CODIFICADOR')
      DO 25 I=1,N
      WRITE(IW,8)(GT(I,J),J=1,K)
   25 CONTINUE
    8 FORMAT(1H0,8(I3,1X))
      NSTAT=1
      NVI=2**K
      CONSTRUCAO DO ESPACO DE INFORMACAO
      CALL SPACE(K,X)
      CONSTRUCAO DAS SUB-MATRIZES AU
      DO 5 I=1,NVI
      DO 10 J=1,K
      XI(J,1)=X(I,J)
   10 CONTINUE
      WRITE(IW,3)I
    3 FORMAT(1H0,' MATRIZ A'I2)
      CALL PRMAIN(N,K,1,GT,XI,PRODX)
      DO 15 J=1,N
      AU(I,J)=MOD(PRODX(J,1),2)
   15 CONTINUE
      WRITE(IW,4)(AU(I,J),J=1,N)
    4 FORMAT(1H0,8(I3,1X))
    5 CONTINUE
      CONSTRUCAO DAS SUB-MATRIZES EU
      DO 20 I=1,NVI
      WRITE(IW,6)I
    6 FORMAT(1H0,' MATRIZ E'I2)
      EU(I,1)=1.
      WRITE(IW,4)(EU(I,J),J=1,NSTAT)
   20 CONTINUE
      CALCULO DAS PROBABILIDADES DOS VETORES DE INFORMACAO
      DO 30 I=1,NVI
      ISOMA=0
      DO 35 J=1,K
      ISOMA=ISOMA+X(I,J)
   35 CONTINUE
      L=K-ISOMA
      Q(I)=(P**ISOMA)*((1.-P)**L)
   30 CONTINUE
      CALCULO DOS COMPONENTES CONTINUO E DISCRETO,NORMALIZADOS,
      DO ESPECTRO DE POTENCIA DO CODIGO
      CALL SPECTR(N,NSTAT,NVI,1.)
      STOP
      END
```

```
      ESTE PROGRAMA DETERMINA OS COMPONENTES CONTINUO E DISCRTO,
      NORMALIZADOS,DO ESPECTRO DE POTENCIA DE CODIGOS DE BLOCO
      NAO LINEARES.
      INTEGER X(32,6)
      REAL AU(96,6),EU(96,6),Q(32)
      COMMON AU,EU,Q
      IW=21
      IR=22
      READ(IR,500)P
500   FORMAT(F5.3)
      READ(IR,501)N,K,NSTAT
      WRITE(IW,511)
511   FORMAT(///,'VETORES DE INFORMACAO')
      CALL SPACE(K,X)
      NVI=2**K
      KA=NSTAT*NVI
      READ(IR,502)((AU(I,J),J=1,N),I=1,KA)
      READ(IR,503)((EU(I,J),J=1,NSTAT),I=1,KA)
501   FORMAT(3I1)
502   FORMAT(12(F3.1))
503   FORMAT(8(F3.1))
      L1=1
      L=1
      M=NSTAT
504   WRITE(IW,506)L1
506   FORMAT(1H0,' MATRIZ A'I2)
      DO 505 I=L,M
      WRITE(IW,507)(AU(I,J),J=1,N)
507   FORMAT(1H0,8(F3.1,2X))
505   CONTINUE
      L1=L1+1
      L=L+NSTAT
      M=M+NSTAT
      IF(L1.LE.NVI)GO TO 504
      L1=1
      L=1
      M=NSTAT
508   WRITE(IW,509)L1
509   FORMAT(1H0,' MATRIZ E'I2)
      DO 510 I=L,M
      WRITE(IW,507)(EU(I,J),J=1,NSTAT)
510   CONTINUE
      L1=L1+1
      L=L+NSTAT
      M=M+NSTAT
      IF(L1.LE.NVI)GO TO 508
      CALCULO DAS PROBABILIDADES DOS VETORES DE INFORMACAO
      DO 515 I=1,NVI
      ISOMA=0
      DO 520 J=1,K
      ISOMA =ISOMA+X(I,J)
520   CONTINUE
      L=K-ISOMA
      Q(I)=(P**ISOMA)*((1.-P)**L)
515   CONTINUE
      CALCULO DOS COMPONENTES CONTINUO E DISCRETO,NORMALIZADOS,DO
      ESPECTRO DE POTENCIA DO CODIGO
      CALL SPECTR(N,NSTAT,NVI)
      STOP
      END
```

BIBLIOGRAFIA

[1] W.R.BENNETT, "Statistics of Regenerative Digital Transmission", Bell Syst. Tech. J., Vol. 37, pgs. 1501-1542, Nov. 1958.

[2] GIAFRANCO L. CARIOLARO and GIUSEPPE P. TRONCA, "Spectra of Block Coded Digital Signals", IEEE Trans. on Comm., Vol. 22, pgs. 1555-1583, Out. 1974.

[3] GIAFRANCO L. CARIOLARO and GIUSEPPE P. TRONCA, "Correlation and Spectral Properties of Multilevel (M,N) Coded Digital Signals with Applications to Pseudoternary (4,3) Codes", Alta Frequenza, Vol. XLIII, pgs. 2-15, Jan. 1974.

[4] F.P.PREPARATA and L.BELLATO, "Error Detection and Synchronization with Pseudoternary Codes for Data Transmission", Alta Frequenza, Jun. 1973.

[5] E.BALICCO and G.JUDICELLO, "Binary Coding in Fiber Optic Digital Transmission with Respect to Timing Extraction and Error Monitoring", TELETTRA S.p.A. Digital Transmission and Wire Communication Division, Mar. 1976.

[6] RADE PETROVIC, "New Transmission Code for Digital Optical Communication", Electronics Letters, Vol. 14, nº 17, Ago.1978.

[7] HISASHI KOBAYASHI, "A Survey of Coding Schemes for Transmission or Recording of Digital Data", IEEE Trans. on Comm., Vol. COM-19, nº 6, Dez. 1971.

[8] PETERSON and WELDON, "Error Correcting Codes", The M.I.T. PRESS, Cambridge, Mass, 1972.

[9] SHU LIN, "An Introduction to Error Correcting Codes", Prentice-Hall, Englewood Cliffs, N.I. 1970.

[10] B.P.LATHI, "An Introduction to Random Signals and Communication Theory", International Textbook, Scranton, Pa, 1969.

[11] F.R.GANTMACHER, "The Theory of Matrices (Vol. I e II) Chelsea Publishing Company, New York, 1971.

[12] Z.KOHAVI, "Switching and Finit Automat Theory", New York, Mc Graw Hill, 1970.

[13] J.L.MASSAY and M.K.SAIN, "Inverses of Linear Sequential Machines", IEEE Trans. on Computers, Abr. 1968.

[14] LAWRENCE P. HUELSMAN, "Digital Computations in Basic Circuit Theory", International Student Edition, Mc Graw-Hill, 1968.

[15] TERCIO PACITTI, CYRIL P. ATKINSON, "Programação e Métodos Computacionais", (Vol. I e II), Rio de Janeiro, Livros Técnicos e Científicos S.A., 1977.