UNIVERSIDADE FEDERAL DE PERNAMBUCO

CENTRO DE TECNOLOGIA

DEPARTAMENTO DE ENGENHARIA ELETRÔNICA E SISTEMAS

CONSTRUÇÃO DE CÓDIGOS LINEARES

Depart.º de Eng.ª Eletrônica e Sistemas
Centro de Tecnologia
Cidade Universitária
50.000 Recife - PE

TESE DE MESTRADO

CONSTRUÇÃO DE CÓDIGOS LINEARES

por

Maicia Cabral Lzal

DEPARTAMENTO DE ELETRÔNICA E SISTEMAS

UNIVERSIDADE FEDERAL DE PERNAMBUCO

Cidade Universitária - Tel. 271.2004

RECIFE - BRASIL

-1985-

UNIVERSIDADE FEDERAL DE PERNAMBUCO

DEPARTAMENTO DE ELETRÔNICA E SISTEMAS

CONSTRUÇÃO DE CÓDIGOS LINEARES

MaAcia Cabial Le.a.1

Tese apresentada à Coordenação do Mestrado em Engenharia Elétrica da Universidade Federal de Pernambuco como parte dos pre-requisitos para a obtenção do título de Mestre em Engenharia Elétrica.

Prof. UALPEMAR C. *ROCHA JA.*

Orientador

AGRADECIMENTOS

A todos que colaboraram durante a elaboração desta tese, minha gratidão, em especial ao meu orientador, *?roíes_* sor VALDEMAR C. ROCHA Jr., cujo incentivo tornou possível a conclusão do meu curso de Mestrado. Agradeço também a CAPES pelo apoio financeiro concedido durante o transcorrer do curso.

R E S U M O

Apesar dos códigos de bloco lineares possuírem uma estrutura matemática bem definida, não existe, em geral, um procedimento prático para a determinação da distância mínima e da distribuição de pesos das palavras código.

0 objetivo deste.trabalho e apresentar procedimentos sistemáticos de construção de códigos lineares.

No primeiro capítulo e apresentada a estrutura básica de um sistema digital de comunicação e conceitos relati-. vos aos códigos corretores de erros. A teoria fundamental dos códigos de bloco lineares e apresentada no segundo capí^ tulo. 0 capítulo 3 apresenta procedimentos sistemáticos de construção de códigos lineares. 0 primeiro procedimento determina a distância mínima e a distribuição de pesos das palavras código, através do programa apresentado no apêndice A. 0 segundo procedimento estabelece uma cota inferior para a distância mínima.

Í N D I C E

Página

Página

CAPÍTULO I

I N T R O D U Ç Ã O

O objetivo fundamental de um sistema de comunicação digital e fazer com que mensagens enviadas por uma determinada fonte cheguem ao seu destinatário com o menor numero de erros possível.

Para que isto aconteça são adicionados, ã mensagem a ser transmitida, dígitos redundantes que permitem a deteção e/ou a correção de possíveis erros ocorridos durante a transmissão. Esta operação é realizada pelo codificador através da utilização de um determinado código corretor de erros.

Atualmente na maioria dos sistemas digitais a informação e codificada em dígitos binários "0" e "1".

Um diagrama de blocos de um sistema digital de comunicação típico e mostrado na figura 1.1.

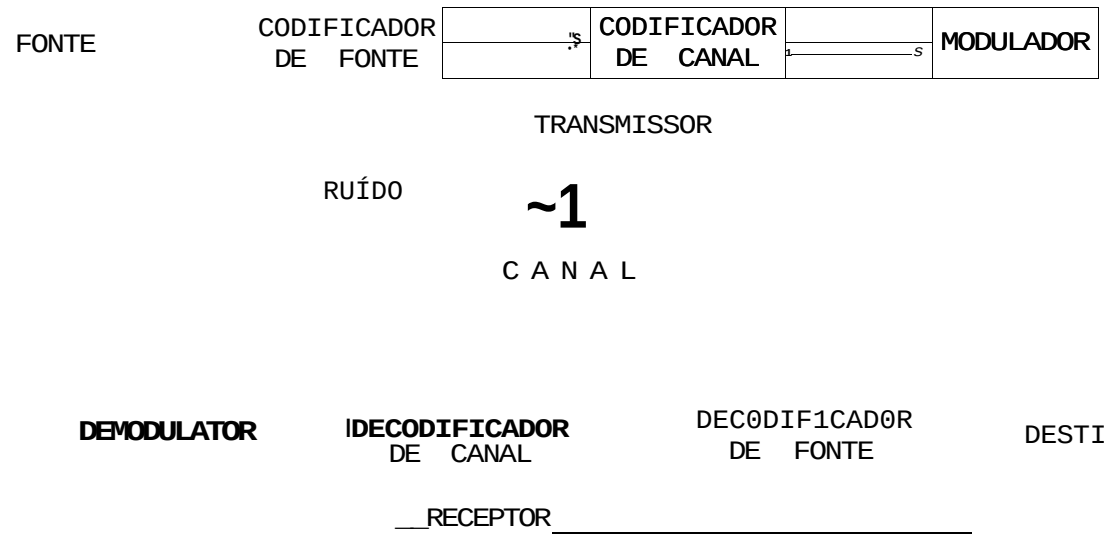


FIG. 1.1 - DIAGRAMA DE BLOCOS DE UM SISTEMA DE COMUNICAÇÃO

FONTE - local de origem da informação a ser transmitida. A saída da fonte pode ser por exemplo uma forma de onda contínua ou uma seqüência de símbolos discretos.

TRANSMISSOR - transforma a saída da fonte em uma forma de onda adequada a transmissão através do canal. H composto dos seguintes dispositivos:

1. *CODIFICADOR DE FONTE* - pode ser por exemplo um conversor analógico/digital que transforma a informação a ser transmitida em uma seqüência de dígitos binários.

2. *CODIFICADOR DE CANAL* - de acordo com determinadas regras, adiciona dígitos redundantes ã informação da fonte para detetar e/ou corrigir erros produzidos pelo ruído do canal.

3. *MODULADOR* - a cada símbolo de saída do codificador, o modulador produz uma forma de onda adequada para ser transmitida através do canal.

CANAL - meio físico através do qual a informação é transmitida. A seqüência de saída do modulador ao passar pelo canal é corrompida pelo ruído fazendo com que ela perca sua forma original. Por exemplo, em uma fita magnética, defeitos na fita podem ser considerados como ruído.

RECEPTOR - estima a forma de onda transmitida a partir da forma de onda recebida da saída do canal, provavelmente cor

rompida pelo ruído. é composto dos seguintes dispositivos:

1. *DEMODULADOR* - estima, a partir da forma de onda recebida do canal, a forma digital correspondente da informação transmitida.

2. *DECODIFICADOR DE CANAL* - a partir da seqüência digital, fornecida pelo demodulador, e .utilizando as regras do codificador de canal, tenta corrigir os erros causados pelo ruído produzindo uma estimativa da informação transmitida.

3. *DECODIFICADOR DE FONTE* - transforma a estimativa da forma digital da informação, na saída do decodificador de canal, na forma original da saída da fonte.

DESTINATÁRIO - recebedor final da mensagem transmitida pode ser por exemplo uma pessoa no extremo da uma linha telefôn_ica ou um computador.

Em geral, qualquer sinal presente no canal que não seja o transmitido pela fonte e denominado ruído. 0 ruído mais comum e o ruído aleatório do tipo mostrado na figura 1.2 (a). Se este ruído e do tipo aditivo, ele irá somar-se a mensagem e o sinal na saída do demodulador pode ser repr£sentado pela figura 1.2 (b). Sendo assim um modo de recuperar o sinal original seria através da amostragem ' do sinal recebido como na figura 1.2 (b). 0 sinal obtido pode ser representado pela figura 1.2 (c).

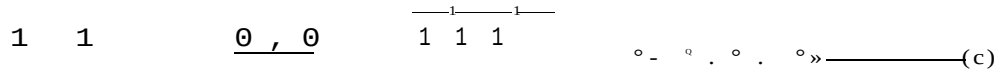


FIG. 1.2 - FORMAS DE ONDAS PRESENTES NO DIAGRAMA DA FIG. 1.1 (RECIiPTOR)

Um dos problemas principais nos sistemas de comunicação e o projeto do par codificador/decodificador de canal de tal modo que a informação seja transmitida o mais rápido possível através do canal e uma reprodução confiável da informação possa ser obtida na saída do decodificador de canal.

Se a seqüência de saída da fonte for segmentada em blocos de mensagem com k dígitos de informação, teremos um total de 2 mensagens distintas. O codificador transforma cada bloco de mensagem com k dígitos em uma seqüência com n dígitos, denominada palavra código. Os n-k dígitos adicionados pelo codificador são denominados dígitos redundantes.

A razão, R=k/n representa a taxa de informação em bits por segundo.

1.1. Canais Discretos sem Memoria

Um canal sem memoria pode ser definido [1] como aque

le cuja saída durante cada intervalo de comprimento T, correspondente a transmissão de um sinal simples, é independente da entrada e da saída do canal durante os intervalos de tempo precedentes.

Para que um canal seja completamente especificado precisamos conhecer os seguintes itens:

(1) - 0 conjunto das possíveis entradas do canal

(2) - 0 conjunto das possíveis saídas do canal

(3) - As probabilidades de transição relacionando os conjuntos acima.

Seja o conjunto M de possíveis entradas constituído dos símbolos $m^{},m^{},...,m^{}$ e o conjunto R de possíveis saidas constituído dos símbolos $r_1, r_2,...,r_p$. Então este canal fica completamente definido a partir da especificação do conjunto de probabilidades de transição PCr^/m^). Para o canal discreto sem memoria a probabilidade condicional P(r/m), onde m é a palavra transmitida e r a palavra recebida, pode ser calculada através da seguinte expressão

$$P(r/m) = \prod_{i=0}^{n-1} P(r_i/m_i)$$

Este canal pode ser representado pela figura 1.3

FIG. 1.3 - CANAL DISCRETO SEM MEMORIA

Se M = R = {0,1} este canal discreto sem memória é denominado canal binário simétrico (BSC) e está representado na figura 1.4.

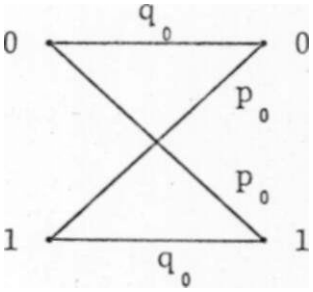


FIG. 1.4 - CANAL BINÁRIO SIMÉTRICO

Sendo q_o a probabilidade de que um determinado símbolo recebido e igual ao transmitido teremos a probabilidade, $p_o = 1 - q_o$, que o símbolo oposto tenha sido recebido. Logo, se a palavra recebida difere da palavra transmitida em d posições teremos,

$$P(\bar{r}^i/m) = q_o^{n-d} \cdot p_o^{d}$$

Em geral $q_o > p_o$ e P(r/m) descresce monotonicamente com o aumento de d. Sendo assim P(r/m) será máxima para a palavra código m que difere no menor número de posições da palavra recebida r.

Se todas as palavras têm a mesma probabilidade de se tem transmitidas então consideraremos a palavra transmitida como sendo aquela que maximiza a probabilidade condicional P(r/m). Este esquema de decodificàção e denominado decodifi^cação por máxima verossimilhança.

1.2. Capacidade do Canal

Considere agora um canal discreto sem memoria com *al*fabetos de entrada e saída representados respectivamente por X e Y que possuem os símbolos particulares x e y com um conjunto de probabilidade condicional P(y/x).

A informação fornecida por um símbolo particular da saída, y, sobre um símbolo particular na.entrada, x, ***ê*** definida como,

$$I(x;y) = \log \frac{P^{\wedge}/tí-}{P(x)} \cdot \log \frac{PCzZxi}{P(y)} \cdot \log \frac{P(x,y)}{P(x).p(y)}$$

Observe que I(x;y) ê uma função simétrica de x e y ou seja a informação fornecida por um y particular a respeito de um x particular e a mesma informação fornecida por x sobre y.

A media, ou valor esperado, da informação mútua entre os símbolos de entrada e saída, e, então

$$I(X;Y) = \sum_{XY} P(x,y) \cdot I(x;y)$$

Esta quantidade depende da distribuição de probabilidade da entrada P(x) e das características do canal representado pelas distribuições de probabilidade condicional P(y/x)« Logo, seu valor para um determinado canal depende somente da distribuição de probabilidade P(x).

A capacidade do canal ê definida como o valor máximo de I(X;Y) com respeito a P(x), isto é,

$$C = \max_{P(x)} I(X;Y)$$

A capacidade do canal, C, representa a informação máxima fornecida por um determinado símbolo na saída do canal, a respeito da entrada.

Sendo assim ao transmitirmos dígitos de informação com uma taxa R>C não devemos esperar que essa transmissão seja confiável.

Entretanto o teorema fundamental de Shannon afirma que [10, pp. 145] para um determinado canal com uma capacidade máxima C, que transmita dígitos de informação a uma taxa R<C, existem códigos de taxa R, os quais, com decodi^ficação de máxima verossimilhança, têm uma probabilidade de erro, Pe, limitada por

$$Pe < e^{-n'^{E}W} \qquad (1.1)$$

onde n ê o comprimento da palavra código e E(R) é uma

função positiva de R para R<C e é especificada pelas probabilidades de transição do canal. A relação entre E(R) e R pode ser representada graficamente pela fig. 1.5

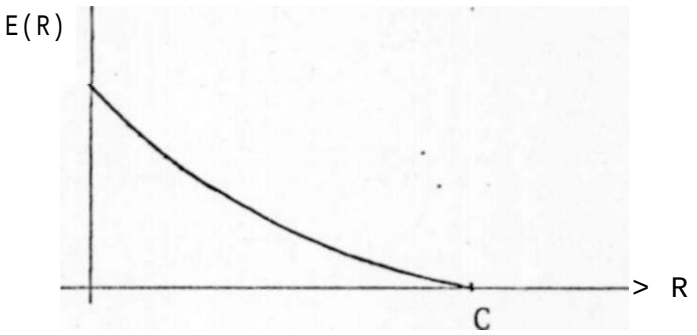


FIG. 1.5 - Relação entre o componente exponencial E(R) e a taxa de transmissão de informação R.

Da expressão (1.1) podemos observar que, para R<C, quanto maior o comprimento da palavra código menor e a probabilidade de erro. Por outro lado isto implica em um equipamento terminal mais complexo. Portanto o problema básico da codificação consiste em encontrar códigos longos e eficientes que possuam métodos práticos de codificação e decodificação.

0 teorema de Shannon apenas prova a existência de códigos com uma probabilidade de erro arbitrariamente pequena mas não indica como estes códigos podem ser construídos.

1.3. Tipos de Erros

Os dois tipos de erros que podem ocorrer durante a transmissão de uma determinada mensagem são:

1. Erros que afetam independentemente cada dígito da mensagem transmitida, denominados erros aleatórios.

2. Erros, que ocorrem em "bursts" de vários erros de cada vez, e, então diz-se que o canal tem memoria. Por exemplo um defeito em uma fita magnética em geral afeta mais de um dígito.

1.4. Códigos Corretores de Erros

Atualmente podemos observar um crescente interesse em relação a pesquisa de códigos corretores de erros. Entre os vários fatores que contribuiram para isto podemos citar:

(1) - a diminuição do custo de dispositivos eletrônicos, o que estimulou o desenvolvimento da tecnologia digital.

(2) - a crescente demanda de sistemas que idealmente não deveriam ter erros como por exemplo o sistema bancário.

(3) - o próprio desenvolvimento da teoria de códigos com a descoberta de novas técnicas de codificação e decodificação.

Idealmente um sistema de transmissão deveria fornecer ao destinatário uma réplica da mensagem gerada pela fonte. Em um sistema real precisamos proteger esta mensagem para que ela chegue ao destinatário com o menor numero de erros possível. Isto é feito através da adição de dígitos redundantes, cuja finalidade é a deteção e/ou correção de er-

ros causados pelo ruído do canal. A adição destes dígitos redundantes e controlada pelo codificador e depende do cod_igo utilizado.

Os dígitos de informação de uma determinada mensagem podem ser transmitidos sem o acréscimo de dígitos redundantes, como por exemplo o seguinte conjunto de todas as poss_íveis palavras de dois bits,

0 0
0 1
1 0
1 1

Considerando as mensagens equiprovãveis, a informação contida em cada uma delas é, $I = \log \mathbf{i} = \log \frac{}{1/4} = 2$ bits e a máxima informação contida em dois dígitos binários é, $I_{MAA} = \log \frac{}{P_1} + \log \frac{}{P_2} = \log \frac{}{1/2} + \log \frac{}{1/2} = 21og2=2$ bits.

Por definição a eficiência de uma mensagem, R, é a relação entre a informação contida na mesma e a máxima informação possível de ser transmitida. Para o exemplo visto teremos,

$$R = \frac{}{2} = \text{í E íocn}$$

e a redundância

$$1 - R \bullet 0 \; S \; 01$$

Neste caso podemos observar que não hã redundância no conjunto de mensagens e qualquer erro ocorrido durante a transmissão de uma determinada mensagem converte-a em uma outra mensagem igualmente valida. Portanto não hã como det£ tar e/ou corrigir erros utilizando este conjunto de mensagens.

Se adicionarmos um dígito extra a este conjunto de mensagens, um zero se o numero de l's é par e um 1 se o numero de l's é ímpar, obteremos o seguinte conjunto de palavras

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

Neste caso um erro ünico em qualquer uma destas palavras pode ser detetado. Por exemplo se a palavra 0 11 for transmitida e ocorrer um erro de tal modo que a palavra recebida é 0 0 1, se recalcularmos o dígito de paridade no receptor baseados nos dígitos de informação concluiremos que deveria ser um zero. Como o dígito reduntante recebido foi um 1, detetamos a presença de erro. Este dígito reduntante é denominado dígito de paridade. Observe que erros duplos não podem ser detetados mas erros triplos podem.

Embora esta redundância proteja a mensagem, reduz a eficiência do código pois agora teremos

$$R = \frac{}{31og_2} = - = 0{,}67 \; E \; 67^{5}c$$

e a redundância

$$1 - R = 0{,}33 \; E \; 33°ô$$

Como podemos observar há um compromisso entre a eficiência do código e a capacidade de deteção e/ou correção de erros, cuja solução ôtima ê função dos objetivos ou das características de cada sistema.

Existem basicamente dois tipos de códigos corretores de erros. Um deles e o código de bloco, cuja seqüência de dígitos de informação e seccionada pelo codificador em blocos com k dígitos de informação fornecendo na sua saída uma seqüência com n símbolos, sendo n>k. Esta seqüência de saída e denominada palavra código e n e o comprimento da palavra código. 0 outro tipo de código e denominado código de árvore. 0 codificador deste código secciona a seqüência de entrada em blocos com k_o dígitos de informação e baseado neste bloco de informação e nos símbolos de informação anteriores, fornece na sua saída uma palavra código com n_o símbolos. Como podemos observar neste tipo de código os bioblocos de informação não são independentes. A subclasse mais importanto destes códigos e a dos códigos convolucionais.

Os códigos de bloco possuem uma estrutura matemática bem definida o que permitiu um maior desenvolvimento da teoria deste tipo de código em relação aos códigos de árvore.

Entre os fatores determinantes na escolha do tipo de código adequado, temos:

- o formato dos dados
- o retardo na decodificação
- a complexidade requerida para um dado desempenho

Os códigos corretores de erros são utilizados essencialmente em sistemas cuja probabilidade de erro tolerável é muito baixa. Apresentaremos a seguir alguns destes sistemas.

(1) - *Transmissão de Dados* - de acordo com a CCITT a taxa de erros em transmissão de dados na rede pública deve ser inferior a 10^{5}, isto é, em média um bit errado em IO^{5} transmitidos. Para que esta taxa de erro seja alcançada sem o uso de um sistema bastante complexo é preciso utilizarmos códigos corretores de erros.

(2) - *Enlaces de* HF - A faixa do espectro de HF, 3a 30 MHz, é normalmente utilizada para transmissão de sinais de voz e sinais telegráficos via rádio. Um sistema de transmissão de dados utilizando a melhor freqüência disponível apresenta taxas de erros da ordem de 10^{-3} ___ 10^{-2} por períodos de 5 a 10 minutos. Utilizando códigos corretores de erros esta taxa poderia atingir o valor ideal de 10^{5}.

(3) - *Comunicação Via Satélite* - algumas das razões para a utilização de códigos corretores de erros em sistemas de comunicação via satélite são:

limitações de largura de faixa e potência fazendo com que seja importante a transmissão de informação a uma taxa tão próxima quanto possível da capacidade do canal.

nas condições normais de operação o canal de comunicação por satélite pode ser modelado por um canal binário simétrico o qual é um dos mais simples para controle de erros.

CAPTTULO I I

TEORIA DE CÓDIGOS DE BLOCO LINEARES

Um código de bloco linear é um conjunto de 2^ palavras código de comprimento n que formam um subspaço k-dimensional do espaço de todas as n-uplas.

As palavras código são formadas a partir da segmentação da mensagem na saída da fonte em blocos de mensagem com k dígitos de informação que são transformados pelo codificador em uma palavra código com n dígitos. Os n-k dígitos redundantes acrescentados pelo codificador são calculados por meio de somadores módulo-q. A função desses dígitos é permitir na*recepção a deteção e/ou correção de possíveis erros ocorridos durante a transmissão da mensagem.

Exemplo 2.1 - Neste exemplo apresentamos um codificador que segmenta a mensagem em blocos de três dígitos e fornece na saída uma palavra código com seis dígitos.

Mensagens	Palavras Código
0 0 0	0 0 0 0 0 0
0 0 1	0 0 1 1 1 0
0 1 0	0 1 0 0 1 1
0 1 1	0 1 1 1 0 1
1 0 0	1 0 0 1 0 1
1 0 1	1 0 1 0 1 1
1 1 0	1 1 0 1 1 0
1 1 1	1 1 1 0 0 0

Observamos que temos 2 ou seja $2^3=8$ mensagens e oito palavras código correspondentes que formam um código linear pois esse conjunto é um subespaço tridimensional do espaço de todas as 6-uplas.

2.1. Matriz Geradora

Uma palavra código pode ser representada por um vetor pois ela ê uma n-upla do espaço vetorial de todas as n-uplas.

Como um código linear ê um subspaço k-dimensional po demos encontrar um conjunto de k vetores,' v ,v ,...,v^, que são linearmente independentes ou seja que formam uma ba se para esse subspaço. Sendo assim qualquer palavra código poderá ser obtida através de combinações lineares desses ve tores.

A matriz construída de tal forma que cada linha e um determinado vetor da base é denominada matriz geradora. Es-sa matriz poderá então ser representada na seguinte forma

$$G = \begin{vmatrix} v_1 \\ v_2 \\ \vdots \\ v_k \end{vmatrix} = \begin{vmatrix} v_{11} & v_{12} & v_{13} & \cdots & v_{1n} \\ v_{21} & v_{22} & v_{23} & \cdots & v_{2n} \\ \vdots & \vdots & \vdots & & \vdots \\ v_{k1} & v_{k2} & v_{k3} & \cdots & v_{kn} \end{vmatrix}$$

Conhecendo a matriz G de um determinado código poderemos obter qualquer palavra código, u, correspondente a um dado bloco de mensagem, m=(m^,m^,...,m^) através do produto desse vetor pela matriz G:

$$u = m \cdot G = (m_1 m_2 \ldots m_k) \quad v$$

$$m_1 v_1 + m_2 v_2 + \ldots + m_k v_k$$

A estrutura matemática associada aos códigos lineares reduz consideravelmente a complexidade do codificador "que não precisa armazenar os 2^ vetores do código mas apenas as k linhas da matriz G.

Em geral os códigos lineares se apresentam em uma forma sistemática onde os dígitos de informação têm uma posição determinada na palavra código bem como os dígitos redundantes e nesse caso a matriz geradora tem a seguinte for

$$G = \left[\begin{array}{ccc|c} 1 & 0 & 0 & \\ 0 & 1 & 0 & \\ 0 & 0 & 1 & \end{array}\right.$$

$$\left.\begin{array}{c} \\ \\ \\ \vdots \\ \vdots \\ \vdots \\ \end{array}\right.$$

$$\left.\begin{array}{c} P_{1,n-k} \\ P_{2,n-k} \\ P_{3,n-k} \\ \vdots \\ \vdots \\ \vdots \\ P_{k,n-k} \end{array}\right]$$

ou G = [I^P], onde 1^ é a matriz identidade k*k e P é uma matriz kx(n-k) tal que p^. e GF(q).

Exemplo 2.2 - Em relação ao código linear (6,3) do Exemplo 2.1 temos a seguinte matriz geradora

$$G = \begin{matrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{matrix}$$

Com a matriz geradora na forma sistemática podemos observar através da multiplicação matricial do bloco de mensagem, m=(m^,m^,...,m^), pela matriz geradora G, que as componentes das palavras código u podem ser determinadas como segue

u^ = nu para i=l,2,...,k

ᵘk₊j⁼Pij'Vᵖ₂j⁻'V---⁺Pkj*ᵐk Para J"¹'²....ⁿ"ᵏ

Desse modo os k primeiros dígitos da palavra código u são os dígitos de informação a serem transmitidos e os n-k dígitos redundantes são obtidos pela equação acima.

Exemplo 2.3 - Neste exemplo utilizaremos a matriz geradora do Exemplo 2.2 para obtermos as equações das palavras código u.

$$u = m*G \quad . \quad . \quad (u_1 u_2 u_3 u_4 u_5 u_6) = (m_1 m_2 m_3) \begin{matrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{matrix}$$

$$. \quad . \quad (u_1 u_2 u_3 u_4 u_5 u_6) = (m_1, m_2, m_3, m_1+m_3, m_2+m_3, m_1+m_2+m_3)$$

Logo:

$$u_1 = m_1 \quad , \quad u_2 = m_2 \quad , \quad u_3 = m_3$$

$$u_4 = m_1 + m_3 \quad , u_5 = m_2 + m_3 \quad , u_6 = m_1 + m_2$$

Observamos também que com a matriz geradora na forma sistemática a complexidade do codificador diminui mais ainda pois agora ele so precisa armazenar a matriz P de dimensão kx(n-k) em vez de armazenar a matriz G de dimen-*são kxn.

A partir da matriz G podemos obter a matriz de paridade ou matriz H que é uma matriz (n-k)xn construída de tal forma que o produto interno de um vetor no espaço linha de G e uma linha de H é zero ou seja o espaço linha de G é ortogonal a H.

Seja,

$$H = \begin{vmatrix} h_1 \\ h_2 \\ \vdots \\ h_{n-k} \end{vmatrix} = \begin{vmatrix} h_{11} & h_{12} & \cdots & h_{1n} \\ h_{21} & h_{22} & & \vdots \\ \vdots & \vdots & & \vdots \\ h_{n-k,1} & h_{n-k,2} & \cdots & h_{n-k,n} \end{vmatrix}$$

Se u=(u^,,...,u$_n$) e um vetor do espaço linha de G então pela definição da matriz H teremos: •

u-H = (O O...O)

Logo, u e uma palavra código se e somente se u-H^T = (O O...O).

A matriz de paridade, correspondente a matriz gerado ra na forma sistemática, e representada do seg-uinte modo:

$$H = \begin{matrix} \overset{11}{} & \overset{21}{} \\ \overset{12}{} & \overset{22}{} \end{matrix} \begin{matrix} P_{k1} & 1\ 0\ 0\ 0 \\ Pk_2 & o\ i\ o\ o \\ & \\ {}^{p}i,n-k & {}^{p}{}_{2},n-k \quad {}^{P}k,n-k \quad O\ O\ O\ O \end{matrix}$$

ou H = [P^T Ijj^J onde P^T e a transposta da matriz P e I$_n$ _ ^ e a matriz identidade (n-k)x(n-k).

2.2. Distancia Mínima

A determinação da-distância mínima de um código e muito importante pois e ela que especifica a capacidade de correção e/ou deteção de erros'. Definiremos a seguir alguns conceitos necessários.

Definição 2.1-0 peso de um vetor v, w(v), é definido como o numero de posições não nulas de v.

Exemplo 2.4 - Seja v=(0,l,l,2), então w(v)=3.

Definição 2.2-0 numero de posições nas quais dois vetores u e v diferem é denominado distância de Hamming.

Seja d(u,v) a distância de Hamming entre dois vetores binários, u e v, então pela definição de adição modulo-2 podemos observar que:

$$d(u,v) = w(u \oplus v)$$

Definição 2.3 - A distância mínima de um código é definida como a menor distância de Hamming entre suas palavras código.

Se um código é linear então a soma de duas palavras quaisquer do código dará origem a uma terceira palavra que também pertence ao código uma vez que o conjunto de palavras código forma um subespaço. Logo, para determinarmos a distância mínima de um código linear é necessário apenas que encontremos a palavra código de peso mínimo.

2.3. Capacidade de Correção e/ou Deteção de Erros

Durante a transmissão de uma determinada mensagem através de um canal, podem ocorrer dois tipos de erros. Um deles é aquele em que cada símbolo transmitido é afetado in

dependentemente pelo ruído e é denominado erro aleatório. 0 outro tipo de erro e aquele que atinge vários símbolos durante um tempo indeterminado e nesse caso dizemos que o canal tem memoria.

Devemos observar que um código que possui uma distância mínima, $d > 2t + 1$, tem uma capacidade de corrigir até t erros ocorridos durante a transmissão pois com essa distancia mínima se a palavra código recebida tiver menos de t erros ela estará mais próxima da palavra código realmente transmitida do que de qualquer outra palavra código.

Exemplo 2.5-0 código do Exemplo 2.1 tem distancia mínima 3, sendo assim sua capacidade de correção de erro e 1, significando que todas as palavras código com um erro podem ser corrigidas, não acontecendo o mesmo no caso de ocorrerem dois ou mais erros. Vamos supor que ao transmitirmos a palavra código (0 1 1 1 0 1) ocorreram dois erros de tal modo que recebemos o vetor (0 1 0 0 0 1). 0 vetor mais próximo do recebido e (0 1 0 0 1 1) que difere deste em apenas uma posição enquanto o vetor realmente transmitido difere em duas posições. Então o decodificador iria determinar que o vetor transmitido foi (0 1 0 0 1 1) quando na realidade o vetor transmitido foi (0 1 1 1 0 1), ocorrendo desse modo uma decodificação incorreta.

No caso da deteção de erros podemos observar que um código com distância mínima d, tem condições de detetar até

d-1 erros pois a ocorrência de d-1 ou menos erros não altera uma determinada palavra código em outra palavra código.

Um código pode ao mesmo tempo corrigir t erros e detetar *l* erros, *l>t*, necessitando para isso, ter uma distância mínima, d>t+£+l. Desse modo podemos limitar a correção de erros a um valor menor que o máximo possível para *-po* dermos detetar um número maior de erros.

2.4. Alguns Métodos de De codificação

2.4.1. Arranjo Padrão

Como já foi visto anteriormente as palavras de um código linear satisfazem a seguinte equação, $u*H^{T}=0.$, onde u e uma palavra código e H^{T} a transposta da matriz de paridade.

Ocorrendo erros durante a transmissão de uma determinada palavra u, o decodificador terá que decidir a partir da palavra recebida r=u+e, onde e e o vetor erro, qual o vetor realmente transmitido. 0 produto $r*H^{T}=s$ e denominado síndrome. Quando s=0 o decodificador decide que não houve erros e que r foi à palavra código realmente transmitiˆda. No caso de s ser diferente de zero. a palavra r não pertence ao conjunto de palavras código e o decodificador fará a correção e/ou deteção de erros utilizando a síndrome de erros.

Considere um código linear (n,k) que possui as seguintes 2^k palavras código: $v_1, v_2, \ldots, v_{2^k}$. Como o vetor recebido pode ser qualquer uma das 2^n n-uplas iremos fazer uma partição das 2^n n-uplas em 2^k subconjuntos disjuntos. Cada subconjunto conterá apenas uma palavra código e se a palavra recebida, r, pertencer ao subconjunto da palavra transmitida, será feita uma decodificação correta.

Um modo de fazer essa partição e através da construção de um arranjo padrão.

A primeira linha desse arranjo e constituída dos 2^k vetores pertencentes ao código, com o vetor $v_1 = (0,0,\ldots,0)$ ocupando a primeira posição a partir da esquerda. Escolhemos uma n-upla e_2 que ainda não tenha sido utilizada e colocamos abaixo do vetor v_1, o resto da linha e completada somando cada vetor v_i a e_2 e colocando a soma $e_2 + v_i$ abaixo de v_i. Depois de completarmos a segunda linha escolhemos um vetor e_3 que ainda não tenha sido utilizado e o colocamos abaixo de e_2, o resto da terceira linha será completada adicionando o vetor e_3 a cada um dos vetores v_i e colocando o resultado $e_3 + v_i$ abaixo de v_i. Esse procedimento e repetido ate que todas as n-uplas tenham sido utilizadas.

O arranjo ficará com a seguinte forma:

v_1	v_2	$\cdots$	v_{2^k}
e_2	e_2+v_2	$\cdots$	$e_2+v_{2^k}$
e_3	e_3+v_2	$\cdots$	$e_3+v_{2^k}$
$\vdots$	$\vdots$		$\vdots$
$e_{2^{n-k}}$	$e_{2^{n-k}}+v_2$	$\cdots$	$e_{2^{n-k}}+v_{2^k}$

As 2^{n-k} linhas desse arranjo sao denominados "cosets" e cada n-upla e_i da coluna um é denominada "coset leader".

Devemos observar que qualquer uma das n-uplas do arranjo padrão pode ser um vetor erro e que somente os 2^{n-k} "coset leaders" são padrões de erro corrigíveis, logo para minimizar a probabilidade de erro esses 2^{n-k} "coset leaders" devem ser escolhidos como os vetores erro mais prováveis de ocorrer em um dado canal.

Considere e_i um vetor erro de peso $w(e_i)$ e e_j um vetor erro de peso $w(e_j)$. Para o canal binários simétrico com probabilidade de transição menor que 0,5, se $w(e_i)<w(e_j)$ então e_i é mais provável de ocorrer que e_j. Nesse caso os coset leaders devem ser escolhidos entre os vetores de menor peso no conjunto de n-uplas disponíveis.

Esse método de decodificação não é muito prático pois precisamos localizar o coset ao qual o vetor recebido per-

tence, o que em geral não é fácil de implementar.

A seguir veremos dois métodos mais práticos de decodificação.

2.4.2. Decodif'icação por Busca Sistemática

Considere um código linear (n,k) com matriz de paridade H.

Teorema 2.1 - (I) Todas as 2^k n-uplas de um coset tem a mesma síndrome.

(II) As síndromes para diferentes cosets são diferentes

Prova: (I) Seja $e_j + v_l$ uma n-upla pertencente ao coset cujo coset leader é e_j para algum i. A síndrome dessa n-upla é, $(e_j+v_l)H^T = e_jH^T + v_lH^T$. Como v_l é uma palavra código, temos $v_l*H^T=0$, logo, $(e_j+v_l)H^T = e_jH^T$. Isto é, a síndrome de qualquer n-upla pertencente ao coset cujo coset leader é e_l será igual a síndrome desse coset leader. Concluimos que todas as n-uplas pertencentes a um mesmo coset têm a mesma síndrome.

(II) Suponha que as síndromes dos coset leaders e_j e e_t são iguais, isto é, $e_jH^T = e_tH^T$. Então $(e_l+e_j)H^T = 0$, o que implica que e_j+e_l deve ser um vetor código v_l. Desse mo-

do e =e.+v., isto é, e está no coset cujo coset leader e e.. o que contraria a regra de construção do arranjo padrão onde o coset leader de cada linha deve ser uma n-upla não utilizada anteriormente. Concluimos que para j ^ t devemos ter $e_j H^T \neq e_t H^T$.

Sendo a síndrome um vetor com (n-k) componentes podemos ter 2^{n-k} síndromes diferentes. Como vimos o arranjo padrão possui 2^{n-k} coset leaders. Observamos então que existe uma correspondência um a um entre os coset leaders e as síndromes.

Utilizando esse fato vamos construir uma tabela para a decodificação do vetor recebido, r, que consiste das seguintes etapas:

(1) Calcular a síndrome de r ou seja $r^* H^T$.

(2) Localizar o coset leader e_i que possui a mesma síndrome de r considerando-o então como o vetor erro causado pelo canal.

(3) O vetor $v_i = r + e_i$ é identificado como o vetor transmitido.

Exemplo 2.6 - A partir da matriz geradora do Exemplo 2.2 obtemos a seguinte matriz de paridade

$$H = \begin{matrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{matrix}$$

Utilizando essa matriz calculamos as síndromes correspondentes aos coset leaders e obtemos assim a seguinte tabela:

SÍNDROME	*COSET LEADER*
0 0 0	0 0 0 0 0 0
0 0 1	0 0 0 0 0 1
0 1 0	0 0 0 0 1 0
1 0 0	0 0 0 1 0 0
1 1 0	0 0 1 0 0 0
0 1 1	0 1 0 0 0 0
1 0 1	1 0 0 0 0 0
1 1 1	1 0 0 0 1 0

Suponha que ao transmitirmos o vetor código (10 0 10 1) ocorreu um erro de tal forma que recebemos o vetor r - (l 0 0 0 0 1). A decodificação do vetor recebido será feita a partir da tabela construída. A síndrome do vetor recebido é:

s = (1 0 0 0 0 1) Q 0 1

0 1 0

1 1 0

1 0 0

0 1 0

0 0 1

s = (1 0 0)

Na tabela observamos que o coset leader correspondei! te a essa síndrome ê (0 0 0 1 0 0), logo, decidiremos que esse foi o vetor erro e então o vetor transmitido foi:

(1 0 0 0 0 1) + (0 0 0 1 0 0) = (1 0 0 1 0 1)

Nesse caso a decodificação foi correta pois o vetor erro era um coset leader.

Agora suponha que ao transmitirmos o mesmo vetor código recebemos o seguinte vetor, r ^ (l 0 1 0 0 1). A síndrome desse vetor é, s=(0 1 0) que na tabela corresponde ao coset leader (0 0 0 0 1 0) sendo este vetor considerado o vetor erro. Desse modo o decodificador decidiria que o vetor transmitido foi

(1 0 1 0 0 1) + (0 0 0 0 1 0) = (1 0 1 0 1 1)

e nesse caso teríamos uma decodificação errônea pois na realidade o vetor erro foi (0 0 1 1 0 0) que não é um coset leader.

Como os coset leaders da tabela incluem todos os padrões de erro de peso um este é um código que corrige um erro por palavra.

Esse tipo de decodificação pode tornar-se bastante complexo a medida que n cresce pois o decodificador precisa armazenar as 2^{n} síndromes correspondentes aos coset leaders.

Além disso para um determinado código corrigir t erros por bloco é necessário gerar, $C^{*}+C^{2}+...+C^{t}$ E $C^{i}<2^{n}$ configurações distintas de erros corrigíveis.

Desse modo podemos concluir que o número de configurações distintas de erros corrigíveis cresce rapidamente com n e t tornando esta técnica de decodificação restrita quanto a utilização prática.

2.4.3. Decodificação por Máxima Verossimilhança

Considere um código cujas palavras cõdigò são equiprováveis e independentes. Nesse caso um procedimento ótimo de decodificação consiste na comparação da palavra recebida com cada uma das 2^{k} palavras código. A palavra código que diferir da palavra recebida no menor número de posições ou seja, que tiver a menor distancia de Hamming é considerada a palavra código transmitida.

Devemos observar que utilizando esse método o tempo de decodificação pode tornar-se excessivamente longo mesmo para valores razoáveis de k, uma vez que cada palavra rece

bida precisa ser comparada com as 2 palavras código.

Essas observações fazem com que esse processo tenha aplicação prática limitada.

2.5. Códigos Simples

Mostraremos a seguir o mecanismo básico de formação de alguns códigos simples e especificaremos seus parâmetros principais.

2.5.1. Códigos de Repetição

Este tipo de código possui os seguintes parâmetros

$$k = 1$$

$$c > 1$$

$$n = k+c = 1+c$$

Para transmitirmos o dígito zero teremos então a palavra código correspondente com n zeros, sendo os n-1 dígitos de paridade repetições do dígito de informação. Analogamente, a transmissão do dígito 1 implica em uma palavra código com n 1's.

Nesse caso teremos apenas duas palavras código, uma com n zeros e outra com n l's.

Um procedimento simples de decodificação, quando n for ímpar, consiste em decidir que foi transmitido o dígito que ocupa a maioria das posições da palavra código. Quando n for par, se ocorrer um empate na quantidade de zeros e

l's apenas detetaremos a ocorrência de erros.

Como podemos observar qualquer padrão de erros $t < [n/2]$, onde $[n/2]$ indica a parte inteira de $n/2$, e corrigível.

A eficiência de um código de repetição é dada por, $R = i$, e a distância de Hamming destes códigos é igual a n.

2.5.2. Códigos de um Onieo Dígito de Paridade

A característica básica destes códigos e a presença de um único dígito de paridade em cada palavra código. 0 ***àí*** gito redundante e acrescentado de tal forma a tornar par o número de l's na palavra código. Sendo assim se a seção de informação possuir um número ímpar de l's, o dígito redundante é feito igual a 1, caso contrário ele e feito igual a zero.

Os parâmetros destes códigos são

$$k > 1$$

$$c = 1$$

$$n = k+c = k+1$$

A distância de Hamming e a eficiência destes códigos são respectivamente

Como podemos observar este tipo de código tem uma *a.1* ta eficiência pois possui um único dígito de paridade. Além disso é capaz de detetar qualquer configuração que tenha um número ímpar de erros. Apesar da sua capacidade de detetar apenas um número ímpar de erros, este código torna-se efi- caz quendo o sistema de comunicação possui um canal de re- torno permitindo que seja solicitada a retransmissão da men sagem.

2.5.3. Códigos de Hamming

A distância mínima e o comprimento de bloco destes códigos são

$$d = 3$$
$$n < 2^{c} - 1$$

onde c é o número de dígitos de paridade.

Sendo d=3, este código pode corrigir um erro por pa. lavra e como o vetor síndrome possui c posições, existin- do portanto $2^{c}-1$ síndromes diferentes de zero, o compri- mento de bloco deste código é tal que permite a verificação de ocorrência de um erro numa palavra porque o número de sín dromes não nulas é sempre maior que o número de posições on de um erro pode ocorrer.

Exemplo 2.7 - Neste exemplo apresentaremos o código de Hamming (7,4,3). Como c=3 iremos escrever todos os núme-

ros binários não nulos que possuem três dígitos, formando assim a seguinte tabela

$$
\begin{array}{lll}
0 & 0 & 1 \quad c_1 \\
0 & 1 & 0 \quad c_2 \\
0 & 1 & 1 \quad k_1 \\
1 & 0 & 0 \quad c_3 \\
1 & 0 & 1 \quad k_2 \\
1 & 1 & 0 \quad k_3 \\
1 & 1 & 1 \quad k_4
\end{array}
$$

Observe que a posição dos dígitos de verificação de paridade está associada aos números da forma 2^i, $i=0,1,2,\ldots$ enquanto os dígitos de informação foram associados aos números binários restantes

Além disso se observarmos as colunas da tabela no sentido descendente poderemos formar as seguintes equações de paridade

$$
\begin{aligned}
c_1 &= k_1 + k_2 + k_t \\
c_2 &= k_1 + k_3 + k_4 \\
c_3 &= k_2 + k_3 + k_4
\end{aligned}
$$

Cada dígito de paridade é obtido através de somas modulo 2 das posições de informação onde um 1 aparece na coluna que está sendo considerada.

A decodificação é feita recalculando os dígitos de

paridade utilizando os dígitos de posição de informação da palavra recebida. Se ocorrer por exemplo um erro em $k^{\wedge}$, os dígitos da síndrome nas posições de c e vão falhar, o que não acontecerá com porque não verifica $k^{\wedge}$. Essa situação particular pode ser representada por

$$\begin{array}{ccc} c_3 & c_2 & c_1 \\ 1 & 0 & 1 \end{array}$$

Na tabela este numero binário corresponde a k_2, o erro então foi localizado e portanto pode ser corrigido.

2.6. Códigos Cíclicos

Os códigos cíclicos têm se destacado entre os diversos tipos de códigos de bloco devido as facilidades encontradas com relação a codificação e decodificação.

A codificação pode ser facilmente implementada utilizando-se registradores de deslocamento. Alem disso a estrutura matemática associada a esses códigos permite que se obtenham vários métodos simples de decodificação.

2.6.1. Definição dos Códigos Cíclicos

Definição 2.4 - Um código linear e um código cíclico se para cada deslocamento cíclico da palavra código $v=(v_o, v_i, \ldots, v_{n-1})$ obtemos uma palavra que pertence ao código. Isto e, deslocando i vezes a palavra código v c i-

clicamente para a direita obtemos a palavra código

$$v^{(i)} = (v_{n-i}, v_{n-i+1}, \ldots, v_{n-1}, \ldots, v_{n-i-1})$$

Nos iremos considerar as componentes da n-upla v como coeficientes de um polinómio de grau máximo (n-1) ou seja

$$v(X) = v_0 + v_1 X + v_2 X^2 + \ldots + v_{n-i} \cdot X^{n-i}$$

Esse polinómio e denominado polinómio código.

A seguir serão citados alguns teoremas relativos a códigos cíclicos. [4, pp.60-63]

Teorema 2.2: Em um código cíclico (n,k) existe um e somente um polinómio g(X) de grau n-k denominado polinómio gerador que e um fator de X^n+1.

Teorema 2.3: Se g(X) e um polinómio de grau n-k e e um fator de X^n+1 então g(X) gera um código cíclico (n k).

Teorema 2.4: Todas as palavras código são múltiplas de g(X) e reciprocamente todo polinómio de grau igual ou menor que n-1 que divide g(X) e um polinómio código.

Exemplo 2.8-0 polinómio X^7+l pode ser fatorado da seguinte forma

$$X^7+l = (1+X+X^3)(l+x+X^{**}x^{*}-J$$

0 polinómio gerador $(1+X+X^2+X^4)$ gera o código cíclico (7,3) mostrado na tabela a seguir onde podemos observar que a distância mínima deste código e quatro e ele não se apresenta na forma sistemática.

MENSAGENS	*POLINÓMIOS CÓDIGO*	*VETORES CÓDIGO*
(0 0 0)	0- (1+X+X²+X") =0	0 0 0 0 0 0 0
(0 0 1)	x². (l + X+X^X") = x²+x³+x⁴+x⁶	0 0 1 1 1 0 1
(0 1 0)	X- (1+X+X²+X*) = x + x²+x³+x⁵	0 1 1 1 0 1 0
(0 1 1)	(X+X²)• (1+X+X²+X⁴) = x+x⁴*+x⁵+x⁶	0 1 0 0 1 1 1
(1 0 0)	1- (1+X+X²+X⁰) = i + x + x² + x⁰	1 1 1 0 1 0 0
(1 0 1)	(i+X²). (1 + X+X²+X³*) = 1 + x + x 3 + x 6	1 1 0 1 0 0 1
(1 1 0)	(1+X). (1+X+X²+X") =1+X³+X⁰+X⁵	1 0 0 1 1 1 0
(1 1 1)	(1+X+X²)• (1+X+X²+X*) =1+X²+X⁵+X⁶	1 0 1 0 0 1 1

A palavra código v(X) pode ser colocada em uma forma sistemática, isto e, na forma $(m_1,m_2,m_3,c_1,c_2,c_3,c_4)$ onde m_1, m_2, m_3 são os dígitos de informação e ocupam uma posição bem definida na palavra código assim como os dígitos de paridade c_1, c_2, c_3, c_4.

A palavra código v(X) pode ser obtida na forma sistemática através do seguinte procedimento

$$v(X) = C(X) + X^{n-k} \cdot m(X)$$

onde C(X) e o resto da divisão de $X^{n-k}*m(X)$ por g(X).

2.6.2. Representação Matricial

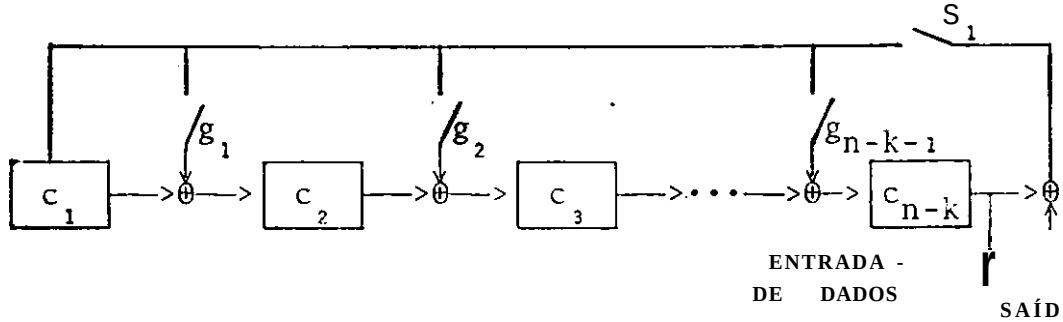
Pelo Teorema 2.4 podemos concluir que os polinómios $g(X), X^{1}g(X), X^{2}g(X), \ldots, X^{k-1}g(X)$ sao palavras código e como são linearmente independentes podem ser utilizados para formar a matriz geratriz de um código cíclico cujo polinómio gerador e $g(X)$ ou seja, a matriz G pode ser representada da seguinte forma

$$G = \begin{matrix} X^{k-1}g(X) \\ X^{k-2}g(X) \\ Xg(X) \\ g(X) \end{matrix}$$

2.6.3. Codificação de Códigos Cíclicos

Pelo que foi visto no item 2.5.1 podemos observar que a codificação de uma mensagem e feita basicamente através do calculo do resto da divisão de $X^{n-k}m(X)$ por $g(X)$.

Esta operação pode ser feita através de um circuito que utiliza n-k estágios de registradores de deslocamento, como o circuito a seguir,



$$g(x) = 1 + g_1 x + g_2 X^2 + \ldots + g_{n-k-1} \cdot x^{n-k-1} + x^{n-k}$$

Neste circuito se $g_i = 1$ a chave correspondente é fechada e $g_i = 0$ indica que a chave deve ficar aberta.

Inicialmente a chave S_1 é fechada e a chave S_2 é colocada na posição 1. Os k dígitos de informação são então enviados simultaneamente à saída do circuito e aos seus registradores de deslocamento. Quando os k dígitos forem todos transmitidos os registradores de deslocamento conterão os dígitos de paridade da palavra código. A seguir a chave S_1 é aberta, a chave S_2 é comutada para a posição 2 e os dígitos de paridade são enviados para o canal.

Pelo Teorema 2.2 temos

$$x^n + 1 \bullet g(X) \bullet h(X)$$

onde h(X) é da forma

$$h(X) = h_0 + h_1 X + h_2 X^2 + \ldots + h_K X^k$$

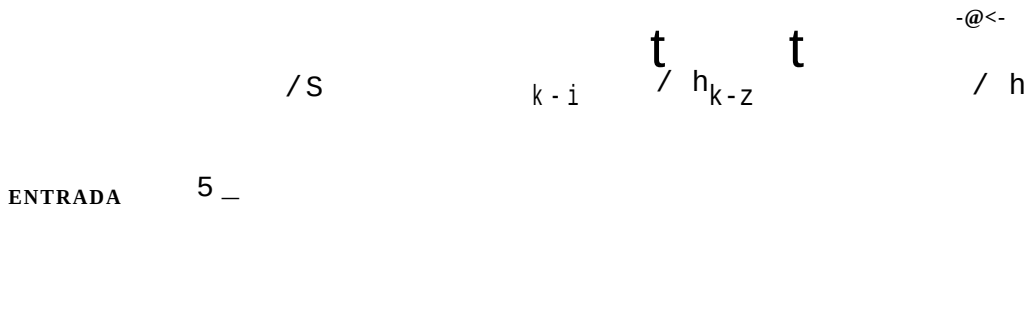
com $h_o = 1$ e $h_k = 1$

Pode ser mostrado que um código cíclico (n>k) é completamente especificado por h(X) [4].

Além disso os n-k dígitos de paridade são determinados da seguinte forma

$$v_j = \sum_{i=0}^{k-1} h_i v_{j-k+i}, \quad k \leq j \leq n-1$$

Um circuito codificador que realiza essa operação e mostrado na figura a seguir



Inicialmente a chave S_1 é fechada e a chave S_2 e aberta. Os k dígitos de informação são então enviados simultaneamente para o canal e para os registradores de deslocamento. Apos os k dígitos de informação terem sido todos transmitidos,'a chave S_1 e aberta, a chave S_2 é fechada e o primeiro dígito de paridade e então calculado e enviado para o canal. Apos ocorrer um deslocamento no conteúdo dos registradores o segundo dígito de paridade e calculado e ajs sim sucessivamente ate que todos os dígitos de paridade tenham sido enviados para o canal.

Como vimos a codificação de códigos cíclicos pode ser feita utilizando um circuito codificador com n-k registradores de deslocamento ou um circuito codificador com k registradores de deslocamento. Obviamente se o código possuir mais dígitos de informação que de paridade o primeiro tipo será mais adequado, do ponto de vista econômico, caso contrário devemos optar pelo segundo tipo.

2.6.4. Decodificação de Códigos Cíclicos

A estrutura algébrica dos códigos cíclicos permite que a deteção de erros na palavra código seja feita de uma maneira bastante simples através do cálculo da síndrome. Ao dividirmos a palavra recebida pelo polinómio g(X) estaremos calculando a síndrome que é o resto desta divisão. Se a síndrome for nula consideraremos que não ocorreram erros durante a transmissão. No caso da síndrome ser diferente de zero decidiremos que ocorreram erros.

Encontramos entretanto uma maior dificuldade no que se refere ã correção de erros na palavra código. Mostraremos a seguir alguns procedimentos utilizados na decodificação de códigos cíclicos:

I - Um dos métodos de decodificação de códigos cíclicos é o algoritmo de Meggit que utiliza um circuito composto de um registrador para armazenar o vetor código recebido, outro registrador que calcula a síndrome deste vetor e um detetor de erros que fornece na sua saída um "1" se o símbolo que se encontra na posição mais a direita do vetor código estiver errado, caso apareça um "0" o símbolo é considerado correto. Se o símbolo estiver errado ele será corrigido na saída do detetor. 0 vetor recebido ê então deslocado e o novo símbolo que ocupa a posição mais ã direita no registrador ê testado do mesmo modo que o anterior. Desse modo todos os díg_itos do vetor recebido são examinados pelo circuito cuja complexidade determina a conveniência de sua utilização.

II - O algoritmo conhecido como "error-trapping" emprega um circuito lógico combinacional bastante simples para detetar e corrigir erros.

Suponha que ao transmitirmos o vetor código, $v(X)$, pertencente ao código cíclico (n,k) com capacidade de corrigir t erros, recebemos o vetor $r(X)$. Sendo assim teremos um padrão de erro $e(X)=r(X).+v(X)$. Pode ser mostrado que [4, pp. 78-79]

$$e(X) = q(X)g(X) + s(X)$$

isto e, a síndrome $s(X)$ é o resto da divisão de $e(X)$ por $g(X)$.

Se os erros de $r(X)$ estão confinados nas $n-k$ posições de paridade, $1,X,..,X^{n-k-1}$ então $e(X)$ é um polinômio de grau $n-k-1$ ou menor.

Observando a equação acima podemos concluir que neste caso teríamos $q(X)=0$ e $e(X)=s(X)$ e a correção seria feita simplesmente adicionando a síndrome aos $n-k$ dígitos de paridade recebidos.

Se os erros não estiverem confinados nas $n-k$ posições de paridade de $r(X)$ mas estiverem confinados a $n-k$ posições consecutivas, ou seja, $X^{*},X^{*+1},...,X^{n+i}$, depois de deslocarmos $n-i$ vezes o vetor recebido $r(X)$, os erros estarão localizados nas $n-k$ posições de paridade do vetor $r^{n}(X)$. Nesse caso a síndrome de $r^{n}(X)$ é i-

gual aos erros nas posições $X^{1}, X^{1+1}, \ldots, X^{n}$ ~ e desse modo os erros podem ser corrigidos.

Caso a síndrome após ter sido deslocada n vezes não apresentar um peso menor ou igual a t o decodificador indica a presença de um padrão de erros que se espalha por um número de bits, ciclicamente consecutivos, maior que n-k.

Este tipo de decodificação e mais adequado para códi^ gos de baixa eficiência pois para sua aplicação a condição ^ > t deve ser satisfeita.

III - A soma $A=b_{o} e_{o} + b_{i} e_{i} + \ldots + b_{n-1} e_{n-1}$, onde b_{i} e igual a 0 ou 1 e denominada soma de verificação de paridade ou simplesmente soma de verificação. 0 dígito e^ e dito ser verificado por A se o coeficiente b^ de e^ em A e 1.

Definição 2.5 - Um conjunto de somas de verificação de paridade $A_{1}, A_{2}, \ldots, A_{J}$ e dito ser ortogonal no dígito e_{p} se $e_{Z.}$ e verificado por cada soma de verificação A_{j} no conjunto e nenhum outro dígito e verificado por mais que uma soma de verificação.

Da definição acima podemos representar cada soma ortogonal em e^ por

$$A_{j} = T_{m} e_{1} + e_{p_{1}}$$

Se for possível formarmos um conjunto $A_1, A_2, \ldots, A_J$ ortogonal no dígito de mais alta ordem e_{n-1} pertencente ao vetor, $e=(e_0, e_1, \ldots, e_{n-1})$ e supondo que apenas J/2 ou menos dígitos podem ser diferentes de zero teremos duas situações. Se $e_{n-1}=0$ então os dígitos não nulos podem estar distribuídos no máximo em J/2 equações portanto pelos menos J-[J/2] equações são iguais a e_{n-1}. Se $e_{n-1}=1$ então os outros dígitos não nulos podem estar distribuídos no máximo em [J/2]-1 equações, portanto pelos menos J-[J/2]+1 ou seja mais da metade das equações são iguais a $e_{n-1}=1$.

Como podemos observar o valor do dígito e_{n-1} será assumido pela maioria das equações de verificação de paridade ortogonais em e_{n-1}. Caso ocorra um empate nos resultados das equações assumiremos que $e_{n-1}=0$.

Sendo assim decidiremos que $e_{n-i}=1$ se a maioria das somas de verificação for igual a 1, caso contrário decidiremos que $e_{n-i}=0$ e dessa forma podemos decodificar o dígito recebido r_{n-i}.

Observe que através do deslocamento cíclico do vetor e, uma vez para a direita, obteremos somas de verificação ortogonais em e_{n-2} ou seja

$$A_1 = e_{n-1} + e_{n-2}$$

Dessa forma podemos repetir o procedimento utilizado na decodificação de e_{n-1} para decodificar os outros dígitos.

0 algoritmo que acabamos de descrever ê denominado decodificação por lógica de maioria. A eficiência deste algoritmo em relação a um determinado código está diretamente ligada ao fato de [J/2] ser ou não igual ou próximo ã capacidade de correção de erro E(d-l)/2] deste código ou seja J deve ser igual ou aproximadamente igual a (d-1).

Definição 2.6.- Um código cíclico com distância mínima d e dito ser completamente ortogonalizável em um passo se e somente se e possível formar J=d-1 somas de verificação ortogonais em todo dígito de informação.

2.7. Códigos B.C.H *(Bose Chaudhuri - Hocquenghem)*

Os códigos cíclicos descobertos independentemente por Hocquenghem (1.959) e por Bose e Chaudhuri (1.960), são denominados códigos B.C.H. ..

Estes códigos possuem os seguintes parâmetros

$$n = 2^m - 1$$

$$n-k = c < mt$$

$$d > 2t + 1$$

onde m e t são quaisquer inteiros positivos, $(t<2^{m-1})$.

Como podemos observar, este código tem uma capacidade de correção de erros igual a t .

0 teorema fundamental dos códigos B.C.H. é a seguir enunciado.

Teorema 2.5 - 0 código B.C.H. cujo polinómio gerador tem d-1 raízes consecutivas $a, a^2, a^3, \ldots, ^1$ tem distância mínima de pelo menos d.

2.8. Códigos de Seqüências-m

Para qualquer inteiro m>2 podemos obter um código cíclico com 2^m palavras código que são as 2^m - l versões deslocadas de uma seqüência-m e mais a seqüência de 2^m - l zeros.

Os parâmetros que caracterizam estes códigos são os seguintes

$$n = 2^m - 1$$

$$k = m$$

$$d - l™'^1$$

A distância de Hamming destes códigos e derivada do fato que o número de dígitos que qualquer par de palavras código difere e 2^1, pois o número de "1's" da palavra código resultante da adição de duas palavras código e 2^{m-1}. [6]

CAPTTULO I I I

ALGUNS PROCEDIMENTOS SISTEMÁTICOS DE CONSTRUÇÃO

Neste capítulo apresentaremos procedimentos sistemáticos para a construção de códigos de bloco lineares. Um destes procedimentos explora as propriedade de grupamentos combinatórios binários e estabelece expressões para o cálculo da distância mínima e da distribuição de pesos. Outro procedimento apresentado dá origem a códigos de bloco lineares a partir de uma técnica de utilização de sub-espaços cíclicos. Estes códigos tem sua distância mínima avaliada indiretamente pela cota BCH apesar de não serem cíclicos.

3.1. Grupamentos Combinatórios

Seja um grupamento de números binários onde as colunas são as combinações de k elementos tomados i de cada vez. Observe que este grupamento tem k linhas e colunas. 0 peso de um vetor linha resultante da adição módulo 2 de quaisquer s linhas deste grupamento e dado por

$$W(s) = \sum_{j\ \text{ímpar}} Z \wedge J \ . \ C^{3} \ , \quad l<s<k \ ; \ j<i$$

A expressão obtida acima e resultante da observação de que W(s) representa o numero de colunas cujas s pos_i

ções envolvidas na adição das linhas contêm um número ímpar de l's.

Estes códigos têm k dígitos de informação. O comprimento e a distância mínima são dados respectivamente por

$$n = C_k^i$$

$$d = \underset{s}{\text{Min}}\ W(s) \quad , \quad l<;s<k$$

A distribuição dos pesos é dada por W(s), l<s<k, existindo para cada valor de s, palavras de peso W(s).

Como $(-)^{k}C_k^i$ existem dois valores de i . que dão o mesmo comprimento de bloco, entretanto de um modo geral as *distâncias* mínimas resultantes são diferentes.

Se construirmos uma matriz de verificação de paridade de tal modo que as k primeiras colunas desta matriz sejam as k linhas do grupamento combinatório e cujas posições vazias restantes são preenchidas com uma matriz unitária de dimensão n, obteremos códigos com k dígitos de informação cujo comprimento e distância mínima são dados respectivamente por

$$d = \text{Min}(W(s)+s)$$

Exemplo 3.1 - Este exemplo mostra a construção do código BCH de parâmetros (15,5,7), a partir de um grupamento combinatório. Seja k=5 e i=3, então

$$d_1 = \min_{s}[s(\tilde{\pounds}^{s}) + (?^{*}) + s] = 7$$

$$n_t = k + (^{k}) = S + (|) = 15$$

Este código tem a seguinte matriz de verificação de paridade:

$$H = \begin{matrix} 0 & & & & \\ 0 & o & & & \\ & o & o & & \\ & & o & o & \\ & & & o & o \end{matrix} \quad {}_{1O}$$

onde $I_{íO}$ é uma matriz unitária 10x10. Para cada valor de s existem ( ) palavras de peso W(s)+s com a seguinte distribuição de pesos

s	1	2	3	4	5
W(s)+s	7	8	7	8	15
numero de palavras			C|3		

A tabela acima pode ser resumida da seguinte forma

Peso	0	7	8	15
numero de paiavras	1	15	15	1

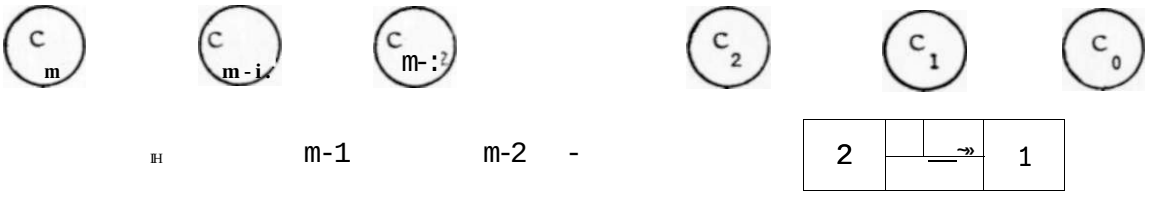
Quando i é par a distancia mínima do código e k. Porem apagando uma linha do grupamento combinatório a distância mínima e aumentada. Por exemplo, para k=9 e i=6 o código (92,9,9) após o apagamento.mencionado torna-se (92, 8,41). No apêndice A e apresentado um programa que fornece o comprimento e a distância mínima destes códigos. Muitos destes códigos são ótimos ou pelo menos excedem a cota inferior publicada por Helgert & Stinaff [5].Como exemplo de códigos que excedem as cotas inferiores mencionadas temos o (92,8,41) obtido com k=9, i=6 e o (120,8,56) para k=9, i«7.

3.2.1. Seqüências-Tü

Um circuito composto de m registradores de deslocamento pode ter somente 2^m estados distintos. A seqüência de saída de comprimento máximo, $n=2^m-1$, e denominada seqüência-m. Devido â realimentação do circuito, o estado zero deve ser excluído pois ele' não leva o circuito a nenhum outro estado.

De um modo geral as seqüências-m são geradas por um circuito do seguinte tipo:

EXCLUSIVE - OR



Para a sequencia-m, $a_o, a_i, a_2, \ldots, a_{2^m-2}$, este circuito implementa a seguinte relação,

$$a_{m+j} = c_m a_{o+j} + c_{m-1} a_{1+j} + \ldots + c_1 a_{m-1+j}, \quad 0 \le j < 2^m - 1$$

onde o símbolo, +, representa a soma modulo-2 e os índices de a são reduzidos modulo 2^m-1.

Como $c_0 = c_m = 1$, a relação acima pode ser expressa por

$$a_{m+j} = a_{o+j} + c_1 a_{1+j} + \ldots + c_{m-1} a_{m-1+j}$$

O polinómio,

$$h(x) = c_m x^m + c_{m-1} x^{m-1} + \ldots + c_1 x + c_0 = x^m + c_{m-1} x^{m-1} + \ldots + c_1 x + 1$$

especifica os valores dos c_i, e as potência de x indicam o registrador de deslocamento correspondente a cada c_i.

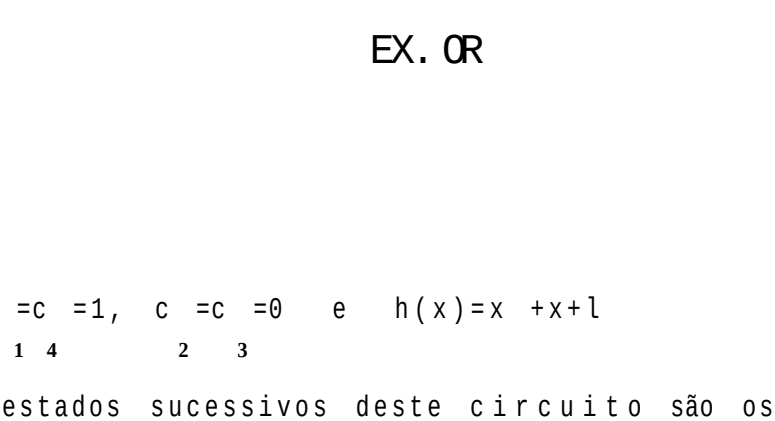
Para o caso binário, os coeficientes do polinómio pertencem ao campo de dois elementos (0 e 1), GF(2).

Se h(X) gera uma seqüência-m então ele e um polinômio primitivo, ou seja, h(x) e um polinómio irredutível [2, pp. 161] e o expoente ao qual ele pertence e $2^{m}-1$, onde m e o grau de h(x).

Exemplo 3.1 - Para m=4, n=15, a seqüência repetitiva,

...0 1 1 1 l 1 0 0 0 1 0 0 1 1 0 1 0 1 1 1 l 1 0 0 0 1.
<——————período 15-——————>

é gerada pelo seguinte circuito



onde $c_0=c_1=c_4=1$, $c_2=c_3=0$ e $h(x)=x^4+x+1$

Os estados sucessivos deste circuito são os seguintes:

	4	3	2	1	(saída)
estado inicial arbitrário	0	0	0	1	
	1	0	0	0	
	0	1	0	0	
	0	0	1	0	
	1	0	0	1	
	1	1	0	0	
	0	1	1	Q	
	1	0	1	1	
	0	1	0	1	
	1	0	1	0	
	1	1	0	1	
	1	1	1	0	
	1	1	1	1	
	0	1	1	1	
	0	0	1	1	
	0	0	0	1	

Como podemos observar esta seqüência-m tem um período, n=15.

Qualquer outra condição inicial resultaria na mesma seqüência deslocada ciclicamente.

Propriedades

1. Uma seqüência-m contém (2^{m-1}) l's e $(2^{m}-1)$ zeros.

2. Se uma "janela" de largura m é deslocada ciclicamente ao longo de uma seqüência-m, então cada uma das $2^m - 1$ m-uplas não nulas ê vista somente uma vez.

3. A adição módulo-2 de duas versões deslocadas de uma seqüência-m é outra versão deslocada desta seqüência-m.

4. Se uma seqüência-m é* "dizimada", selecionando todo k-ésimo dígito, onde k e $2^m - 1$ são primos entre si, então obtemos uma versão deslocada da mesma seqüência-m ou de outra.

3.2.2. Sub-Espaços Cíclicos

Considere a matriz de verificação de paridade construída como segue:

a) As primeiras k colunas são k versões consecutivamente deslocadas de uma seqüencia-m de comprimento $2^k - 1$, isto é, k colunas linearmente independentes.

b) As n-k colunas seguintes são escritas de modo a formar uma matriz identidade (ri-k) * (n-k)

0 código resultante tem os seguintes parâmetros:

Comprimento do bloco: $n = k + 2^k - 1$

Número de dígitos de informação: k

Distancia Mínima: $d = 1 + 2^{k-i}$

A distância mínima é calculada a partir da observação dé que qualquer combinação linear das k primeiras colunas resulta num vetor coluna de peso 2^{k-1} e portanto o número mínimo de colunas da matriz de verificação de parida

de que adicionadas dao uma coluna toda nula e $1+2^{k-1}$

Em relação aos códigos de seqüências-m, estes códigos possuem k dígitos de verificação de paridade a mais, para um aumento de 1 na distância mínima. Embora isto possa parecer desfavorável, é o melhor que pode ser feito com códigos lineares tendo os parâmetros acima [5] e que portanto são ótimos.

Estes códigos são facilmente decodificáveis por lógica de maioria, pelo fato de suas matrizes de verificação de paridade poderem ser vistas como matrizes de verificação de paridade de códigos de seqüências-m com $^{c}\wedge^{=}\wedge\wedge$ » l ^ i ^ k dígitos de verificação de paridade adicionais. Como códigos deseqüências-m são decodificáveis por lógica de maioria [2, pp. 310-317] os códigos resultantes permitem a formação de 2^{k-1} equações de verificação de paridade ortogonais em cada uma das k posições de informação. Portanto estes códigos são decodificáveis em um passo por lógica de maioria.

Considere agora a seguinte matriz de verificação de paridade:

a) As k primeiras colunas são k versões deslocadas consecutivamente de uma seqüencia-m de comprimento $2^{k}-1$.

b) As k colunas seguintes são k versões deslocadas con-

secutivamente de uma outra seqüência-m de comprimento 2^k-1.

c) As colunas restantes são preenchidas de modo a formar uma matriz unitária $(2^k-1)x(2^k-1)$.

A fim de determinar a distância mínima deste código determinaremos inicialmente a distância mínima entre os sub espaços gerados pelas duas seqüências-m dos passos a e b acima. Como estes dois conjuntos de k seqüências-m deslocadas são linearmente independentes, tomados em conjunto eles formam um sub-espaço cíclico de dimensão 2k, i.e, um código cíclico de comprimento 2^k-1 com 2k dígitos de informação. A distância mínima $d^{\wedge}$ deste código cíclico e cotada inferiormente pela cota BCH [2, pp. 269-291] e e a mesma que a distância mínima entre os dois sub-espaços originais de dimensão k. Esta cota ê calculada contando o número máximo de raízes consecutivas de um polinómio que e o máximo divisor comum dos polinómios geradores dos dois códigos de seqüência-m distintos.

Como podemos observar a distância mínima será dada por $d^{ss}2+d$.

Este procedimento e expandido considerando matrizes de verificação de paridade contendo conjuntos de colunas que são bases de sub-espaços cíclicos independentes. A distância do código resultante e pelo menos $i^{\dagger}$ > onde i e o número de sub-espaços cíclicos independentes e d e adis

tância do código cujo polinómio gerador e o maior divisor comum dos polinómios geradores destes sub-espaços independentes.

Exemplo 3.2 - Fazendo uso da seqüência linear de comprimento máximo igual a 7 obtem-se a seguinte matriz de verificação de paridade onde I representa uma matriz unitária 7x7.

H =

0 código resultante tem parâmetros

n = 10

k = 3

d = 5

Um ponto importante é a escolha dos diferentes sub-espaços usados na formação da matriz de verificação de paridade, com o objetivo de maximizar a distancia mínima para valores fixados de n e k.

CAPTTULO IV

C O N C L U S Õ E S

A principal característica dos códigos apresentados neste trabalho, é a forma sistemática como eles foram construídos.

Os códigos lineares construídos a partir de sub-espaços cíclicos possuem propriedades de distância conhecidas. A seleção dos diferentes sub-espaços que formam a matriz de paridade deve ser feita com cuidado-a fim de que se obtenha o melhor código permitido por esta construção. Uma referência para tal seleção e a cota BCH.

Embora os códigos construídos através da utilização do primeiro método apresentado neste trabalho tenham baixa eficiência, e preciso observar que:

1. Apresentam expressões bem definidas para a determinação da distribuição de peso das palavras código e da distância mínima do código.

2. Através de um programa bastante simples, como o apresentado no apêndice A, podemos obter todos os parâmetros destes códigos em função de k e i.

3. Muitos destes códigos são ótimos ou pelo menos excedem a cota inferior publicada por Helgcrt & Stinaff [5], como

por exemplo o (92, 8, 41) obtido com K=9 e i=6.

A Tabela I do Apêndice C apresenta alguns códigos construídos com k e i variando de 1 a 33. O sinal, * , desta tabela indica que, para um determinado código (n,k,d) ao apagarmos uma linha do grupamento combinatório obtivemos um código (n,k-l,d') com d'>d.

Seria interessante encontrarmos métodos práticos que permitissem a codificação bem como a decodificação dos cód.igos apresentados neste trabalho.

Em relação aos códigos derivados de sub-espaços cíclicos observamos que embora seja possível a determinação da distancia mínima destes códigos indiretamente, através da cota B.C.H., poderíamos tentar encontrar uma forma mais exata para o cálculo desta distância.

APÊNDICE A

PROGRAMA DE COMPUTADOR

```
C       CONSTRUÇÃO DE C0DIG3S LINEARES A PARTIR DE GRUPAMENTOS
C       COMBINATÓRIOS
        INTEGER W,S
        DIMENSION W(50),COMB(50,50),N1(50,50),N3(50,50),N2C50,50)
        DIMENSION FAT(0:50),ICOM(50,50),AST(50,50)
        FAT(0)=1
        DO 5 JC=1,33
        M=JC
        FAT(M)=FAT(M-1)*M
5       CONTINUE
        DO 600 L=l,33
        DO 600 JI=1,33
        K=L
        I=JI
        IF(K-I) 10,7,7
7       COMB(K,I)=(FAT(K)/FAT(I))/FAT(K-I)+.5
        IF(COMB(K,I)-127.)30,10,10
10      N1(I,K)=0
        N2(I,K)=0
        AST(I,K)='
        GO TO 600
30      ICOM(K,I)=COMB(K,I)
C       COMPRIMENTO DAS PALAVRAS CÓDIGO COM K DÍGITOS DE INFORMAÇÃO
        Nl(I,K)=K+ICOM(K,I)
        DO 300 FS=1,K
        S=FS
        W(S)=0
        DO 220 FJ=1,I,2
        J=FJ
        IF(S-J)250,50,50
50      IF((K-S)-(I-J))220,200,200
200     COMB(S,J)=(FAT(S)/FAT(J))/FAT(S-J)+.5
        ICOM(S,J)=CQMB(S,J) .
        COMB(K-S,I-J)=(FAT(K-S)/FAT(I-J))/FAT((K-S)-(I-J))+.5
        I COM(K-S, I - J) =COMB (K-S, I - J)
        W(S)=1COM(S,j)*ICOM(K-S,I-J)+W(S)
220     CONTINUE
C       DISTRIBUIÇÃO DE PESO DAS PALAVRAS CÓDIGO
250     VV(S)=W(S)+S
300     CONTINUE
        MIN=W(1)
        DO 500 S=1,K
        IF (MIN-W(S))500,500,400
400     NQN=W(S)
500     CONTINUE
```

```
C       DISTANCIA MÍNIMA DO CÓDIGO COM K DÍGITOS DE INFORMAÇÃO
        N2(I,K)=MIN
        AST(I,K)=' '
        JD=I/2
        JM=2*JD
        IF(I-JM) 1000,510,600
510     NIM=W(1)
        DO 550 S=1,K-1
        IF(NIM-W(S))550,550,520
520     NIM=W(S)
550     CONTINUE
C       DISTANCIA MÍNIMA DO CÓDIGO COM K-l DÍGITOS DE INFORMAÇÃO
        N3(I,K)=NIM
        IF(N2(I,K)-N3(I,K))560,600,600
C       COMPRIMENTO DAS PALAVRAS CÓDIGO COM K-l DÍGITOS DE INFORMAÇÃO
560     N1(I,K)=N1(I,K)-1
        N2(I,K)=N3(I,K)
        AST(I,K)='*'
600     CONTINUE
        DO 900 INI=0,22,11
        DO 900 KSO=INI,22,ll
        JI=INI+1
        JF=INI+11
        WRITE(3,650) (II,II=JI,JF)
650   FORMATCl*,9X,11('1=',12,6X))
        DO 900 K=l,ll
        KK=K+KS0
        WRITE(3,700) KK
700     FORMAT(//,lX,'K=',12)
        WRITE(3,800) (N1(I,KK),N2(I,KK),AST(I,KK),I=JI,JF)
800     FORMAT('+',5X,11(2X,I3,',',I3,A1))
900     CONTINUE
1000    STOP
        END
```

APÊNDICE B

ÁLGEBRA BÁSICA

Sistemas algébricos satisfazem determinadas regras que, em sua maioria, podem ser aplicadas ao nosso sistema numeral comum.

Códigos que apresentam uma estrutura al-gébrica bem definida, podem ser implementados através de métodos mais práticos e suas propriedades são mais facilmente estabelecidas.

Apresentaremos a seguir os conceitos fundamentais necessários a compreensão da teoria de códigos de grupo, apresentados neste trabalho.

B.1 - Grupos

Um grupo G é um conjunto de elementos a,b,c,..., para o qual uma operação está definida, e que satisfaz os axiomas 1 a 4 abaixo. Esta operação é denotada por a+b=c ou ab=c, e chamada adição ou multiplicação embora possa não ser a adição ou multiplicação da aritmética de números ordinários.

Axioma 1 - Para quaisquer elementos a e b £ G, o elemento a+b=c **E** G.

Axioma 2 - Para quaisquer três elementos a,b e c e G, tem

se, (a+b)+c = a+(b+c).

Axioma 3 - Existe um elemento identidade.

Se a operação é denominada adição, o elemento identidade (ou elemento neutro) é chamado zero e representado por 0, e é definido pela equação, 0+a = a+0=0 para todo a e G.

Se a operação ê denominada multiplicação, a identidade é chamada um e representada por 1, e é definida pela equaçao, la=al=a.

Axioma 4 - Todo elemento do grupo tem um inverso.

Se a operação ê adição, o elemento inverso correspondente do elemento a é denotado -a, e é definido pela equação, a+(-a*) = (-a)+a = 0.

Se a operação é multiplicação, o inverso de a é* denotado a^{1} e é definido pela equação, $a«a^{1=}a^{1}*a=l$.

Um grupo que além de satisfazer os axiomas 1 a 4, satisfaz a propriedade comutativa, isto é, a+b=b+a ou ab=ba, ***ê*** denominado grupo Abeliano ou Comutativo.

Teorema B.l - 0 elemento identidade em um grupo é único, e o inverso de cada elemento do grupo ê único.

Prova: Suponha que existem dois elementos identidade, l e i ' então, (1)(1')=1=1^{1}. Se um elemento do grupo, g, tiver dois inversos $g"^{1}$ e $g\sim\backslash$ então $g\sim^{1}=1 \cdot g"^{1}=g\sim^{1} \cdot g \cdot g"^{1}=g\wedge"^{1} \cdot l=g\sim^{1}$, logo, eles são iguais.

Observe que o inverso de um produto e o produto dos inversos na ordem trocada, pois $(ab)(b^{-1}a^{-1})=a(bb^{-1})a^{-1}$ $a\cdot 1\cdot a^{-1}=1$, e portanto $b^{-1}a^{-1}=(ab)^{-1}$.

Exemplo 1-0 conjunto de todos os números reais é um grupo em relação ã operação de adição.

Exemplo 2-0 conjunto de todos os números reais excluindo o zero é um grupo em relação ã operação de multiplicação.

Exemplo 3-0 conjunto de todas as matrizes não-singulares n*n é um grupo não-Abeliano em relação à operação de multiplicação matricial.

Existe um grupo com somente um elemento que pelo Axioma 3 deve ser o elemento neutro.

Podemos também formar um grupo com dois elementos. Um deles deve ser o elemento identidade, 0. Se a for o outro elemento do grupo então a deve ter um inverso, -a. Temos

$$a+0 = a/0$$

então

$$-a \neq 0$$

logo

$$-a = a$$

Este grupo apresenta então a seguinte tabela de adição

+	0	a
0	0	a
a	a	0

Podemos observar que este grupo satisfaz os Axiomas 1 a 4 e â propriedade comutativa sendo portanto um grupo Abeliano.

B.2 - Anéis

Um anel R é um conjunto de elementos para o qual duas operações são definidas. Uma é denominada adição e denotada, a+b , e a outra é denominada multiplicação e denotada, ab, embora essas operações possam não ser a adição ou multiplicação de números ordinários.

Para R ser um anel, os seguintes axiomas devem ser satisfeitos.

Axioma 1-0 conjunto R é um grupo Abeliano em relação a adição.

Axioma 2 - Para quaisquer dois elementos a e b e R, o produto ab é definido e é um elemento de R

Axioma 3 - Para quaisquer três elementos a,b e c e R,

$$a(b+c) = ab + ac$$

e

$$(b+c)a = ba + ca$$

Se a operação multiplicação, de um determinado anel, é comutativa, isto é, para quaisquer elementos a e b, ab=ba, então este anel é denominado anel comutativo.

Exemplo 4-0 conjunto de todos os números reais é um anel comutativo em relação as operações de adição e multiplicação.

Exemplo 5-0 conjunto de todas as matrizes n*n é um anel não-comutativo em relação âs operações de adição e multiplicação matricial.

Um conjunto consistindo somente do elemento zero é um anel com as regras, 0+0=0, (0)(0)=0.

Pode-se formar dois anéis diferentes com dois elementos. Um elemento deve ser a identidade aditiva 0 e o outro elemento deve satisfazer, a+a=0. Temos, (0)(0)=0a=a0=0 e como, aa=0 ou aa=a, satisfazem as leis associativa e distributiva, podemos formar dois anéis de estruturas diferentes.

B.Z - Campos

Um campo é um anel comutativo, com um elemento unitário (identidade multiplicativa), no qual cada elemento não-nulo tem um inverso multiplicativo.

Exemplo 6-0 conjunto de todos os números reais forma um campo em relação ãs operações de adição e multiplicação da

álgebra comum.

Exemplo 7-0 conjunto {0,1} forma um campo em relação às operações de adição modulo 2 e multiplicação modulo 2, apresentando as seguintes tabelas

+	0	1
0	0	1
1	1	0

Adição Mõdulo-2

•	0	1
0	0	0
1	0	1

Multiplicação Mõdulo-2

Este campo e denominado campo de Galois de dois elementos e representado por GF(2).

Pode ser mostrado [2, pp.155] que para todo numero q que é uma potência de um numero primo existe um campo com q elementos.

Definição B.l - Sejam F_1 e F_2 dois campos e suponhamos que F_1 esta contido em F_2, isto ê, F_1 e um subconjunto de F_2. Então diz-se que F_1 e um subcampo de F_2.

Pela definição B.l, o conjunto dos números reais ê um subcampo do conjunto dos números complexos.

B.4 - Espaços Vetoriais

Um conjunto V de elementos e um espaço vetorial sobre um campo F, se ele satisfaz os seguintes axiomas:

Axioma 1-0 conjunto V é um grupo Abeliano em relação â adição.

Axioma 2 - Para todo vetor v e para todo elemento do campo, a, está definida uma operação denominada produto e representada por av.

Axioma 3 - Se u e v são vetores em V e a e F,

$$a(u+v) = au+av$$

Axioma 4 - Se v é um vetor e c e d pertencem a F,

$$(c+d)v = cv+dv$$

Axioma 5 - Se v é um vetor e c e d pertencem a F,

$$(cd)v = c(dv), \text{ e } lv=v$$

Os elementos do campo são denominados escalares e os elementos do conjunto V são denominados vetores.

Exemplo 8 - Seja V_n o conjunto de todas as n-uplas, $[v]=[v_1,v_2,...,v_n]$ onde $v_i=0,l$, $i=l,2,...,n$, isto e, V. **E** GF(2).

A soma de duas n-uplas [u] e [v] é definida como sendo

$$[u]+[v] = [u_1+v_1, u_2+v_2,...,u_n+v_n]$$

onde u_i+v_i representa a soma mõdulo-2 entre u_i e v_i.

Como '''''' ® claro que, [u]+[v]=[v]+[u]

A multiplicação de uma n-upla [v] por um elemento do campo, a, é definida como sendo

$$a[v] = [av_1, av_2, \ldots, av_n]$$

Pode ser facilmente verificado que as operações definidas, satisfazem os axiomas que definem um espaço vetorial sobre um campo F. Sendo assim podemos dizer que o conjunto V_n de todas as n-uplas binarias, juntamente com as operações de adição e multiplicação mõdulo-2, constitui um espaço vetorial sobre o campo GF(2).

Definição B.2 - Um subconjunto V, do espaço vetorial V, que satisfaz a todos os axiomas que definem um espaço vetorial, é denominado um subspaço do espaço vetorial V.

Definição B.3 - Um vetor u é uma combinação linear dos vetores $v_1, v_2, \ldots, v_k$, pertencentes a um espaço vetorial V sobre um campo F, se existem escalares $a_1, a_2, \ldots, a_k$ e F não todos nulos, tais que

$$u = a_1 v_1 + a_2 v_2 + \ldots + a_k v_k$$

onde a_i é denominado o coeficiente de v_i.

Exemplo 9-0 conjunto V de todas as combinações linea-
res de um conjunto de vetores $v_1, \ldots, v_n$ ^e um espaço veto
rial V é um subspaço de V.

Dizemos que V é o subspaço gerado pelos vetores
$v_i, \ldots, v_n$ ^e V'=V, ou seja, se todo elemento de V é uma
combinação linear dos v^, então dizemos que estes vetores
geram o espaço V_n-

Definição B.4 - Um conjunto de vetores $v_1, \ldots, v_n$ é linear
mente dependente se e somente se existem escalares $a_1, \ldots, a_n$
não todos nulos tal que

$$a_1 v_1 + a_2 v_2 + \ldots + a_n v_n = 0$$

Se um conjunto de vetores nao é linearmente dependen
te então ele é' dito linearmente independente.

. *Exemplo 10* - Considere a matriz H de elementos em GF(2).

$$H = \begin{matrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{matrix}$$

Observe que as três linhas desta matriz são linear-
mente independentes. 0 conjunto formado por todas as possí-
veis combinações lineares das linhas desta matriz, consti-
tui um subespaço do espaço vetorial de todas as 6-uplas. Es

te subspaço é chamado o espaço linha de H.

Definição B.5 - Todo espaço vetorial V possui pelo menos um conjunto de vetores linearmente independente que gera V. Este conjunto é denominado uma base de V e o numero de elementos da base e a dimensão de V.

Exemplo 11 - Considere o seguinte conjunto de vetores

$$[v_1] \quad [1\ 0\ 0\ 0]$$
$$[v_2] \quad [0\ 1\ 0\ 0]$$
$$[v_3] \quad [0\ 0\ 1\ 0]$$
$$[v_i] \quad [0\ 0\ 0\ 1]$$

Podemos verificar que estes vetores são linearmente independentes. Qualquer vetor em V é uma combinação linear deste conjunto de vetores e portanto este conjunto constitui uma base para este subespaço de dimensão quatro.

Definição B.6 - Um produto escalar de duas n-uplas, u e v, pertencentes a um espaço vetorial V sobre um campo F, e uma regra que associa a cada par u, v, um escalar em F representado por u*v.

Para $[u]=[u_1,u_2,...,u_n]$ e $[v]\ [v_1,v_2,...,v_n]$, temos $[u]\ [v] = u_i v_i + u_2 v_2 + ... + U_n v_n$.

Da definição B.6 podemos observar que $[u>[v]=[v]'[u]$. Se o produto escalar de dois vetores e igual a zero, eles são ditos ortogonais.

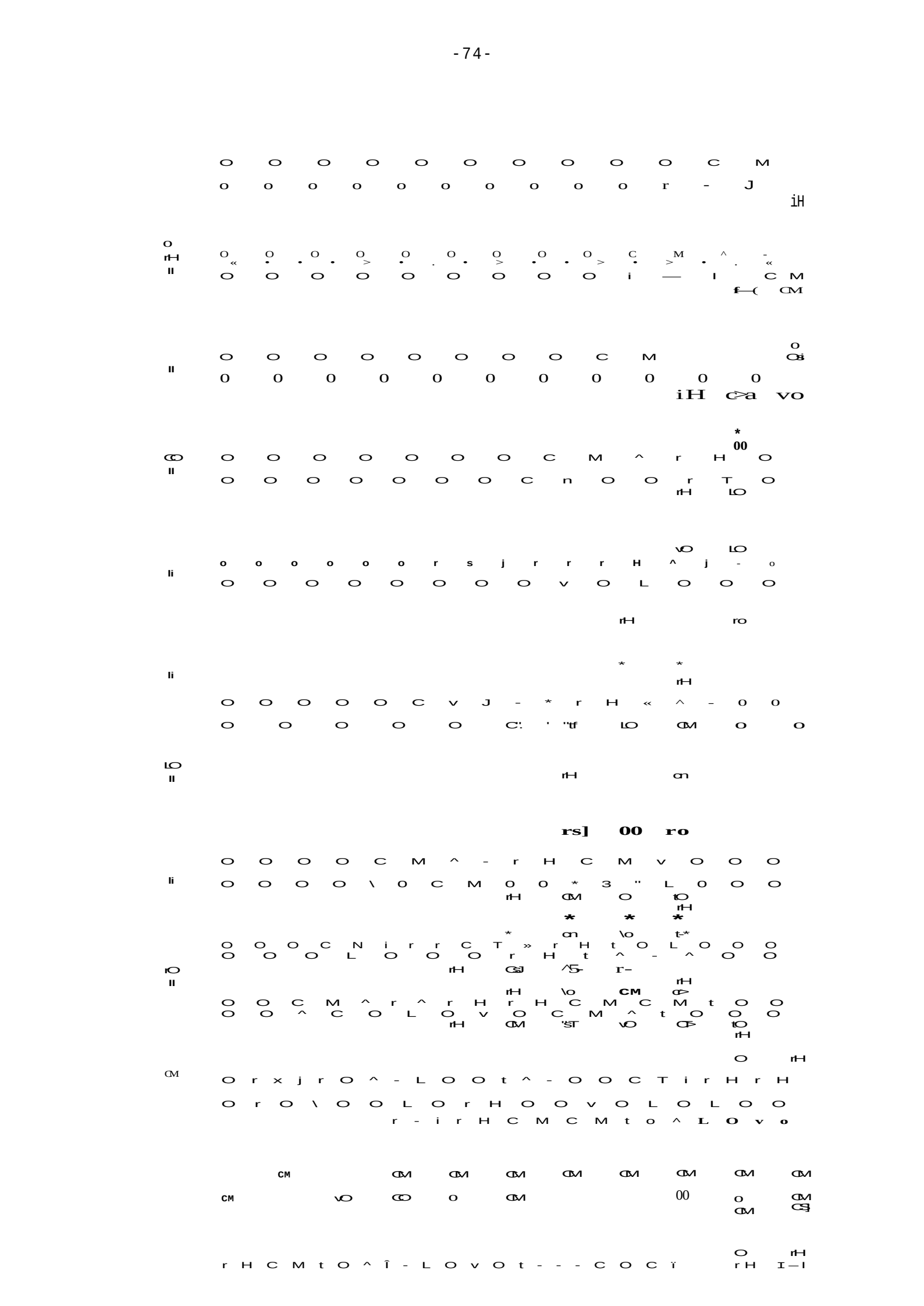
Pode-se provar [4, pp. 29-30] que para toda matriz G, Kxn, com K linhas linearmente independentes, existe uma matriz H, (n-K)xn, com n-K linhas linearmente independentes, tal que todo vetor, v, no espaço linha de G é ortogonal a todos os vetores linha, h.., de H, isto é, o produto interno

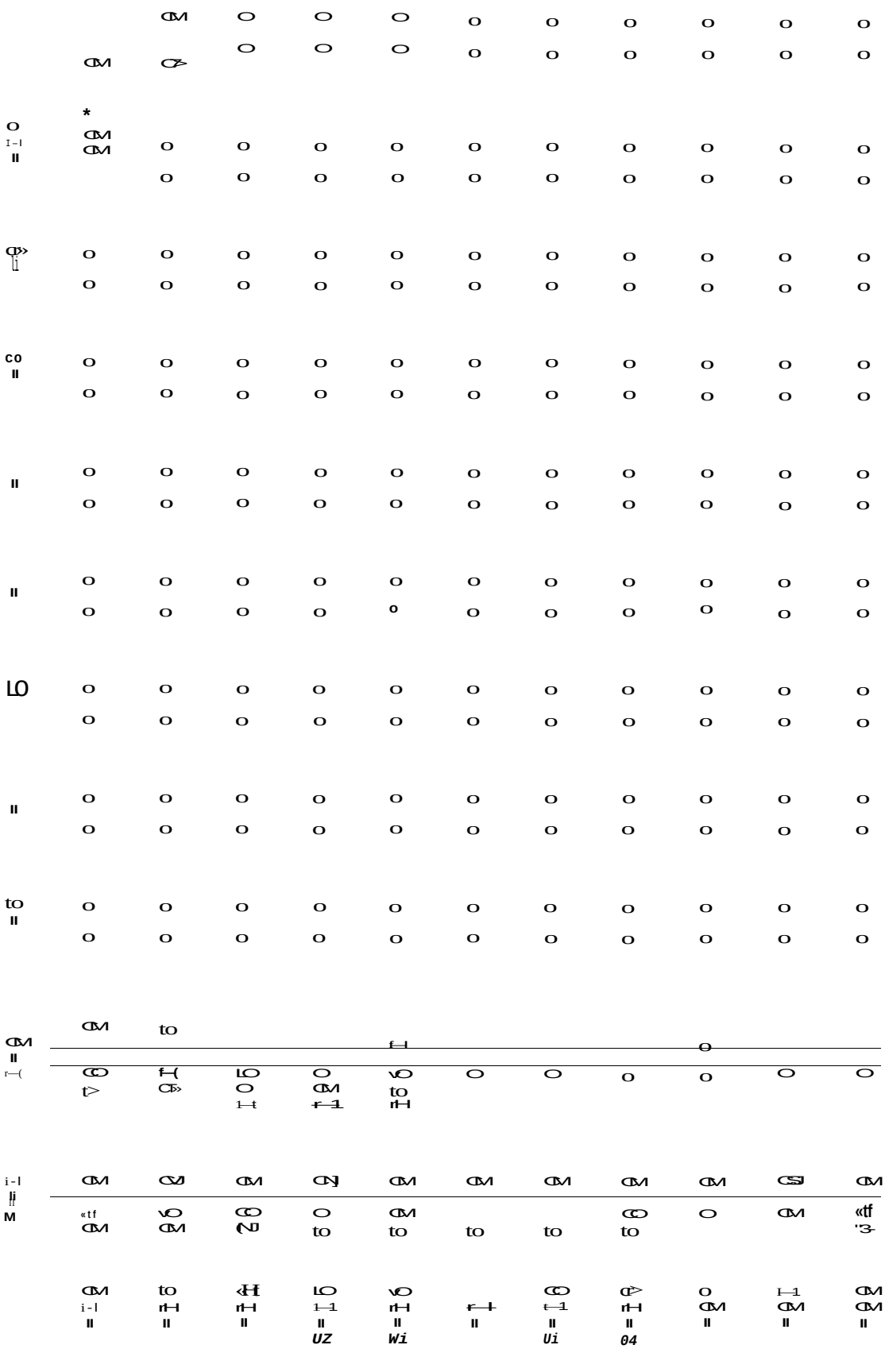
$$v h_j = 0 \text{ , para } 1<j<n-K$$

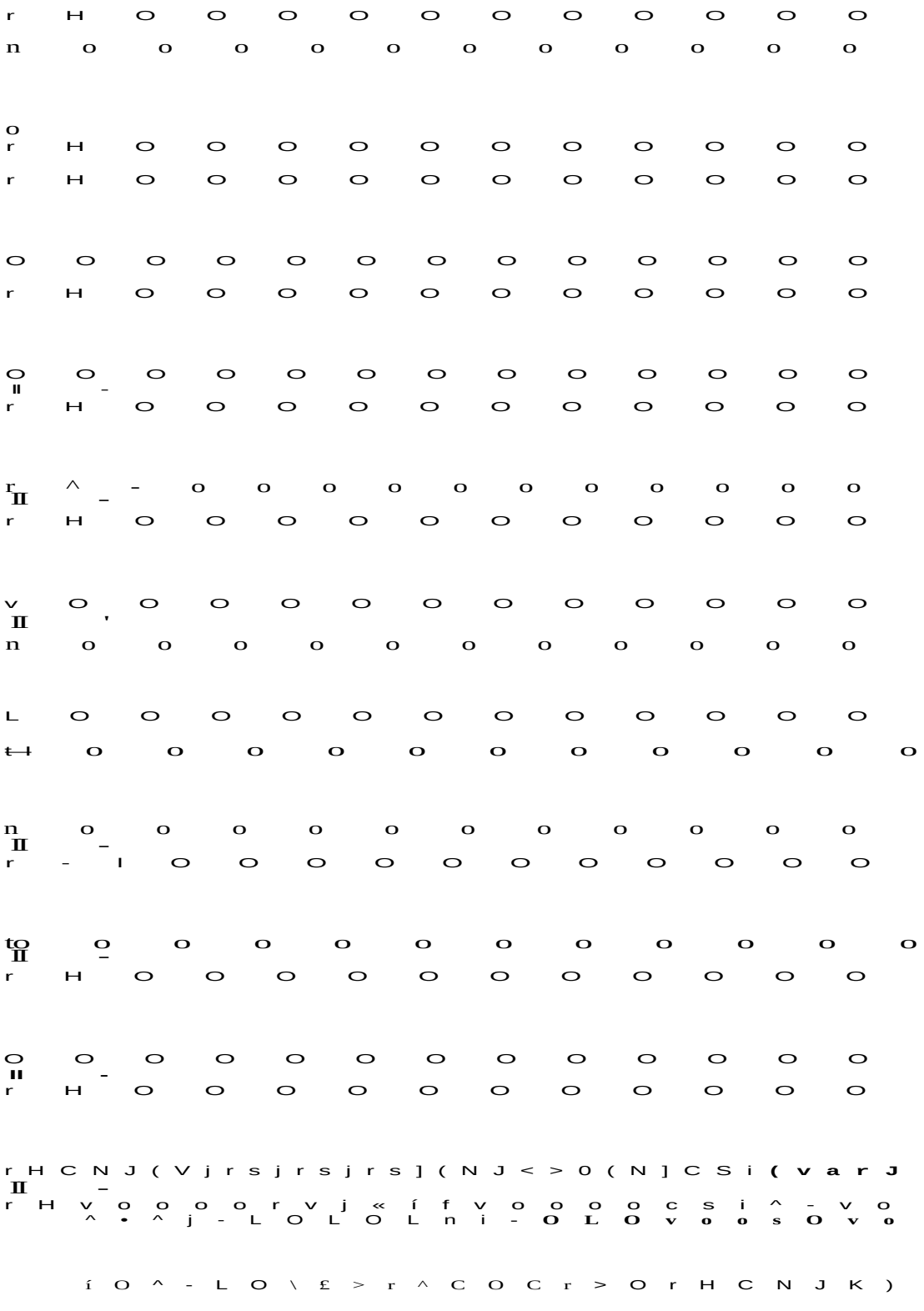
Qualquer vetor y no espaço linha de G é ortogonal a qualquer vetor u no espaço linha de H, isto é, u»v=0. 0 espaço linha de G é denominado o espaço nulo de H ou o espaço linha de H é o espaço nulo de G.

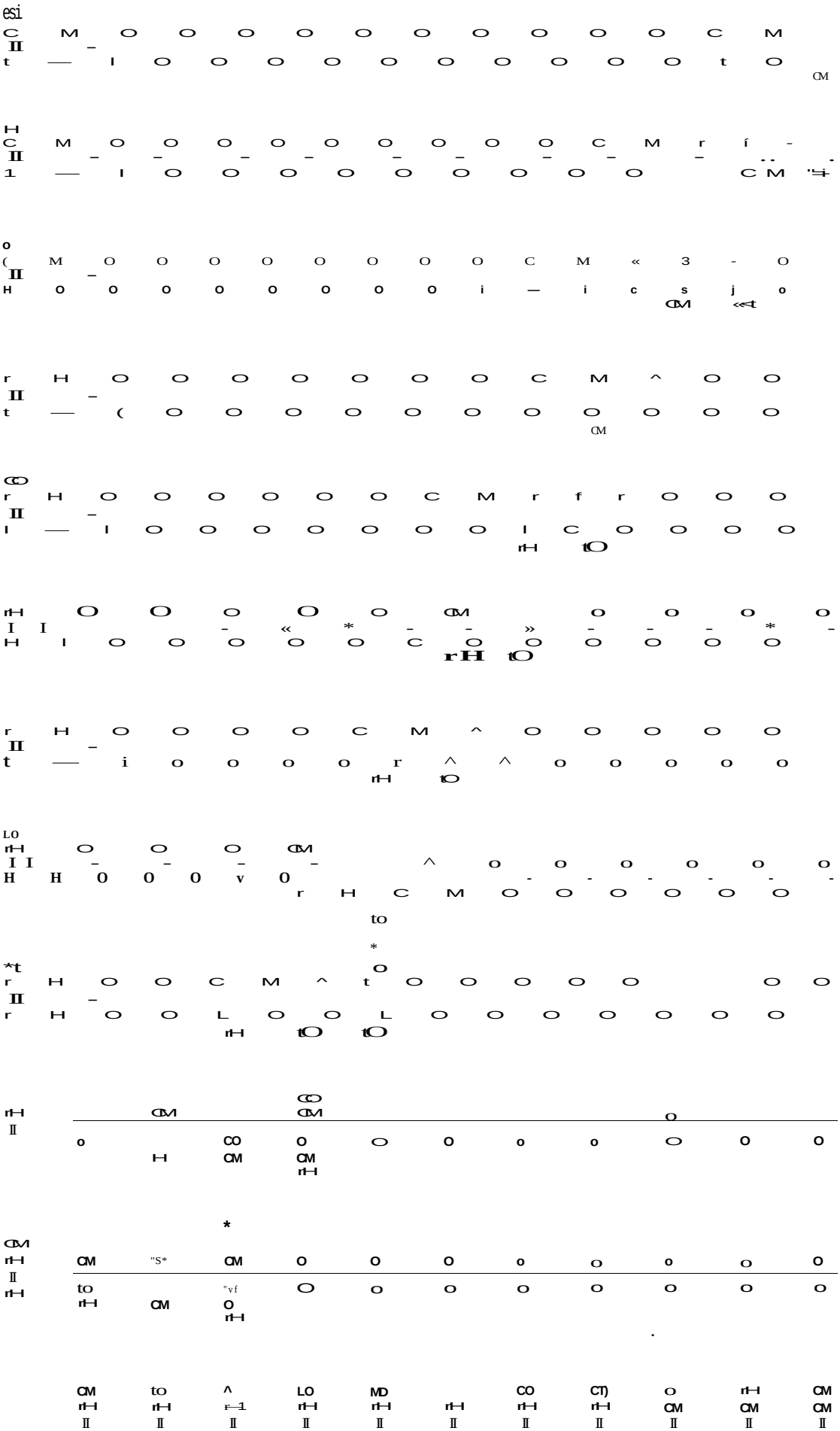
APÊNDICE C

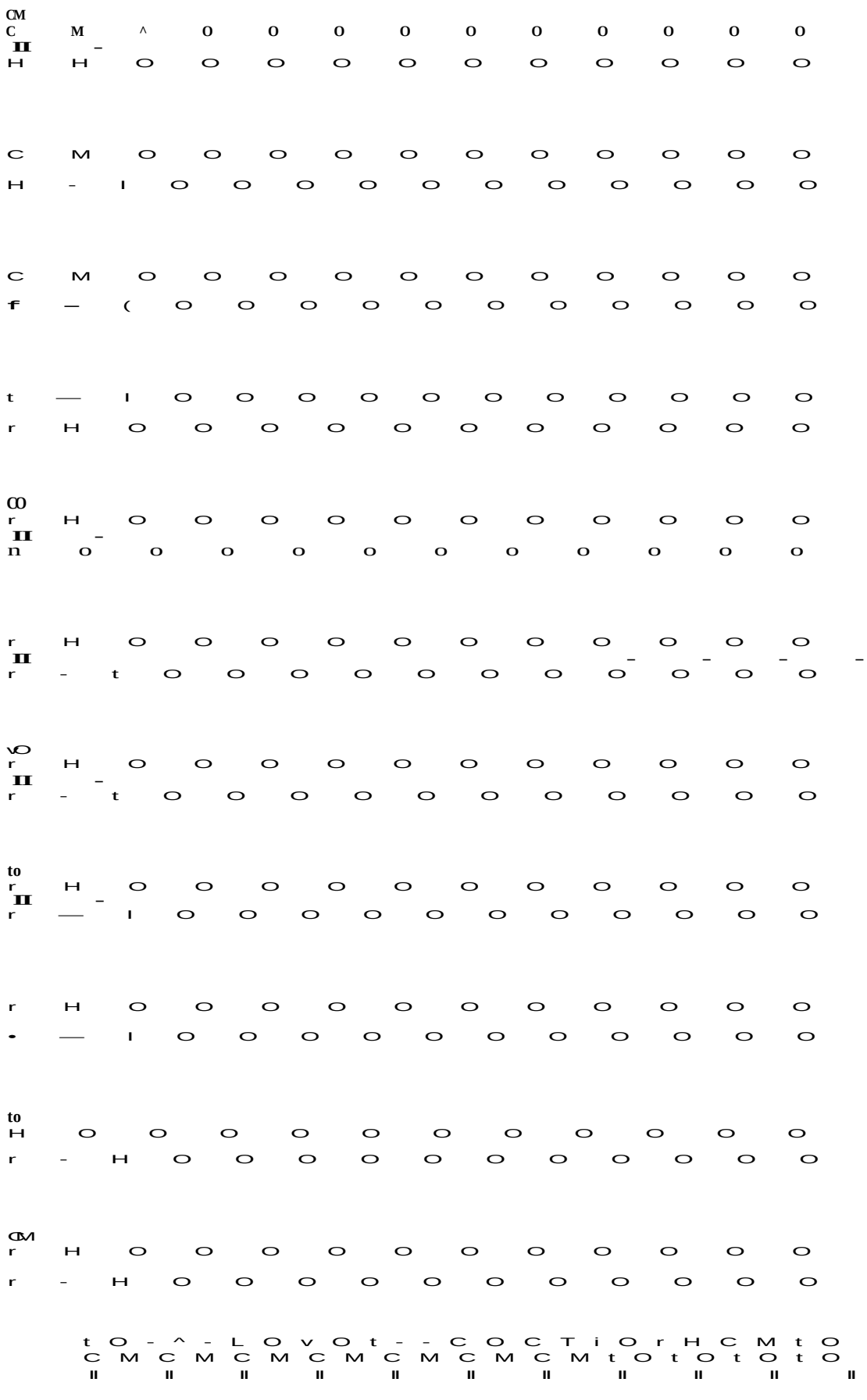
T A B E L A I

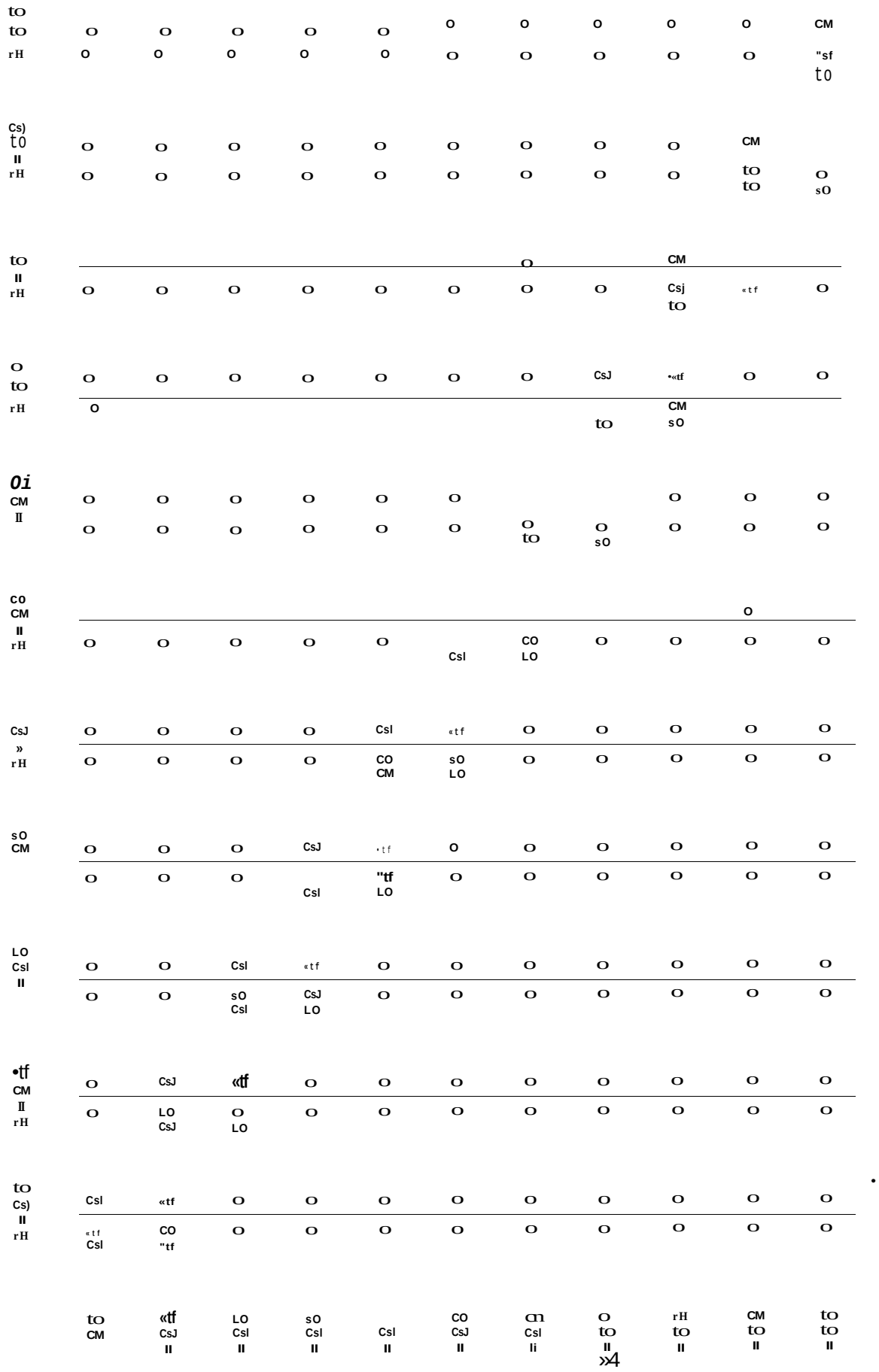












REFERÊNCIAS

[1] R.M.FANO, "A Heuristic Discussion of Probabilistic Decoding", IEEE Trans. Info. Th. Vol. IT-9, abril, 1963

[2] W.W.PETERSON e E.J.WELDON Jr., "Error-Correcting Codes", The Mit Press, 2- ed., Massachusetts, 1972.

[3] V.C.ROCHA Jr., "Easily Decodable Binary Cyclic Codes", INT. J. Eletronics, 1983, Vol. 55, 465-467.

[4] S.LIN, "An Introduction to Error Correcting Codes", Prentice-Hall, New Jersey, 1970.

[5] H.J.HELGERT e R.D.STINAFF, "Minimum-Distance Bounds for Binary Linear Codes", IEEE Trans. Info. Th., Vol. IT-19, maio, 1973.

[6] P. G. FARRELL, "Lectures Notes on Communications Theory", UFPE, junho, 1977, publicação interna.

[7] V.C.ROCHA Jr., "The Burst Error Correcting Power of m-sequence codes", Int. J. Eletronics, 1983, Vol. 55, 469-471.

[8] J.M.WOZENCRAFT e B.REIFFEN, "Sequential Decoding", The Technology Press e John Wiley, New York, 1961.

[9] J.L.MASSEY, D.J.COSTELLO Jr., e J.JUSTESEN, "Polynomial Weights and Code Constructions", IEEE Trans. Info.Th., Vol. IT-19, Janeiro, 1973.

[10] R.G.GALLAGER, "Information Theory and Reliable Communication", John Wiley, New York, 1968.

[11] V.C.ROCHA Jr. e R.M.C.SOUZA, "Cyclic Codes for Random or Burst Error-Correction", IEEE, International Symposium on Information Theory, St. Jovite, Quebec, Canada, 26-30 de setembro de 1983.

[12] V.C.ROCHA Jr., R.M.C.SOUZA e P.G.FARRELL, "Multilevel Pseudocyclic Codes", IEEE Trans. Inf. Th. (submetido)

[13] V.C.ROCHA Jr., M.C.LEAL, "Construção de Códigos Lineares", V Congresso Nacional de Matemática Aplicada e Computacional, 01-05 de agosto de 1982, João Pessoa-PB

[14] V.C.ROCHA Jr., "A Class of Multilevel Error-Correcting Codes", Int. J. Electron. 51, 825-829.

[15] V.C.ROCHA Jr., "Maximum Distance Separable Multilevel Codes", IEEE Trans. Info. Th., Vol. IT-30, n? 3, pp. 547-548, maio 1984.

[16] E.R.BERLEKAMP, "Algebraic Coding Theory", Mc Graw-Hill, New York, 1968.

[17] R.M.CAMPELLO de SOUZA, "Decodificação Probabilística de Códigos Lineares", Tese de Mestrado, UFPE, 1979.

[18] V.C.ROCHA Jr., "Linear Codes Derived from Cyclic Codes", Int. J. Eletronics, 1983, Vol. 54, 345-347.