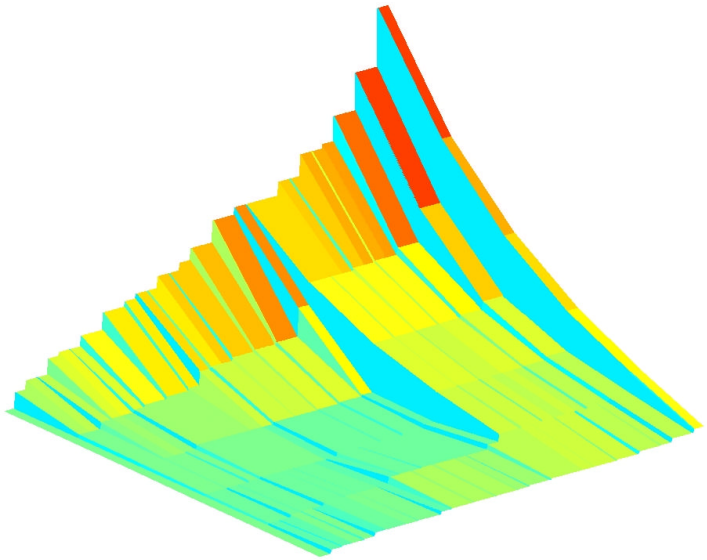
Universidade de Aveiro Departamento de Electrónica e Telecomunicações
2000

José Manuel Neto Vieira

Reconstrução de sinal e codificação



José Manuel Neto Vieira

Reconstrução de sinal e codificação

Reconstrução de sinal com detecção da posição das amostras erradas e códigos de correcção de erros

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Doutor em Engenharia Electrotécnica, realizada sob a orientação científica do Dr. Paulo Jorge dos Santos Gonçalves Ferreira, Professor Associado com Agregação do Departamento de Electrónica e Telecomunicações da Universidade de Aveiro

O júri
Presidente

Prof. Dr. José Abrunheiro da Silva Cavaleiro
Professor Catedrático da Universidade de Aveiro por delegação do Reitor da Universidade de Aveiro

Prof. Dr. José Manuel Nunes Leitão
Professor Catedrático do Instituto Superior Técnico da Universidade Técnica de Lisboa

Prof. Dr. Francisco António Cardoso Vaz
Professor Catedrático do Departamento de Electrónica e Telecomunicações da Universidade de Aveiro

Prof. Dr. Paulo Jorge dos Santos Gonçalves Ferreira
Professor Associado com Agregação do Departamento de Electrónica e Telecomunicações da Universidade de Aveiro

Prof. Dr. Jorge dos Santos Salvador Marques
Professor Auxiliar do Instituto Superior Técnico da Universidade Técnica de Lisboa

Prof. Dr. Aníbal João de Sousa Ferreira
Professor Auxiliar da Faculdade de Engenharia da Universidade do Porto

Prof. Dr. Atílio Manuel da Silva Gameiro
Professor Auxiliar do Departamento de Electrónica e Telecomunicações da Universidade de Aveiro

Agradecimentos

Ao Prof. Francisco Vaz gostava de agradecer o constante estímulo e as condições de trabalho e amizade que me proporcionou.

Ao Prof. Paulo Jorge Ferreira agradeço as longas horas de conversa, as suas explicações que quase sempre tornaram claro o que antes parecia obscuro, a sua prontidão na revisão da tese e a amizade e estímulo dado.

Ao Prof. Tomás Oliveira e Silva agradeço as estimulantes conversas, os artigos que me facultou e que tanto influenciaram o percurso deste trabalho e finalmente a leitura atenta da tese que realizou.

Ao Prof. Armando Pinho pela sua ajuda na formatação do texto desta tese.

Ao meu colega Prof. Dr. José Luís Azevedo agradeço a camaradagem e o ambiente de trabalho que me proporcionou. Gostava também de agradecer a todos os meus colegas do Departamento de Electrónica e Telecomunicações, nomeadamente à Prof. Dr.ª Ana Tomé e ao Mestre João Rodrigues pelo excelente relacionamento humano e profissional que proporcionam.

Resumo

Considere-se um sinal passa-baixo com *N* amostras, *t* das quais foram posteriormente modificadas. Será possível detectar quais as amostras modificadas, e recuperar o seu valor original?

A quase totalidade das técnicas conhecidas sobre reconstrução de sinal não resolve este problema, sendo necessário conhecer a posição das amostras erradas para que seja possível obter a amplitude correcta. No entanto, os códigos de correcção de erros do tipo BCH, entre outros, conseguem resolver este problema, ou seja, determinar a posição das amostras erradas e, numa segunda fase, corrigir a sua amplitude. Estes códigos são normalmente aplicados a sinais digitais recorrendo a "software" com aritmética num corpo finito necessitando, por esse motivo, de processadores dedicados para realizar as operações de codificação e descodificação de forma eficiente. No entanto, e como veremos ao longo deste trabalho, é possível utilizar estas técnicas no corpo dos números complexos, levantando-se no entanto uma série de questões novas como, por exemplo, a da estabilidade da reconstrução.

As técnicas de reconstrução de sinal e os códigos de correcção de erros são usualmente encarados como disciplinas distintas com as suas técnicas próprias não existindo aparentemente qualquer relação entre as duas. No entanto, estas dificuldades resultam em grande medida da diferente aritmética, notação e linguagem utilizadas. Esta dissertação utiliza em simultâneo estas duas disciplinas, transportando técnicas e conceitos de uma área para outra, numa tentativa de enriquecer o conhecimento e compreensão de ambas.

Neste trabalho estudamos várias técnicas para determinar a posição das amostras erradas num sinal discreto limitado em banda e de duração finita, sendo descrito como reconstrução não linear. Comparamos igualmente algumas técnicas de determinação da amplitude das amostras erradas, oriundas dos códigos de correcção de erros, com as que são utilizadas nos algoritmos de reconstrução de sinal. O problema da estabilidade da determinação da posição das amostras erradas é investigado, mostrando-se por exemplo que os efeitos da amplitude e posição dos erros na estabilidade podem ser separados.

Um dos objectivos deste trabalho foi o de encontrar técnicas que permitissem o projecto de códigos de correcção de erros no corpo dos números complexos. A chave para a solução deste problema reside na combinatória dos padrões dos erros e da sua influência na estabilidade dos algoritmos de reconstrução.

Abstract

Consider a low-pass signal with *N* samples where the amplitude of *t* samples was modified. Is it possible to find a method to detect which samples violate the low-pass condition and reconstruct their original amplitude?

Most of the known techniques of signal reconstruction don't solve this problem, being necessary to know the error positions in order to find their correct amplitudes. However, the BCH type error correction codes, for example, can solve this problem in two steps, first, they found the error positions, then, they found the error amplitudes. Those codes are usually implemented on a finite arithmetic field with dedicated processors, which implement the coding and decoding tasks on an efficient way. Nevertheless, and as we will see during this work, it is possible to use those techniques on the complex field, arising some new problems as the numerically stability of the reconstruction process.

The signal reconstruction techniques and the error correction codes are usually faced as distinct disciplines, with their own techniques and results, and apparently with few aspects in common. These difficulties are the result of differences in the arithmetic notation and language used. This thesis deals with both disciplines transporting techniques and concepts from one area to another, trying to enrich the knowledge and comprehension of both.

In this work we have studied several techniques to find the error positions in a band-limited and time-limited signal. Those techniques are described as non-linear signal reconstruction. We also compare some techniques for error amplitude correction imported from error correction codes with some signal reconstruction techniques. The stability of the problem of finding the error positions was also studied. We have achieved some interesting results as for example the separation of the influence of the error amplitude from the error position on the stability.

One of the goals of this work was to find some techniques to design error correction codes on the complex field. The key to this problem is the error pattern combinatory and their influence on the reconstruction stability.

À minha esposa Ci,

aos meus filhos Guilherme, Francisco e Bernardo

Índice

Lista de Figuras

Capítulo 1

Introdução

1.1 Motivação

Quando começámos os estudos preliminares desta tese, conhecíamos um conjunto de métodos de reconstrução de sinal com uma característica comum: pressupunham sempre como conhecidas as posições das amostras erradas. Durante a conferência SampTA95, surgiu-nos a seguinte questão:

Problema 1 *Considere-se um sinal com N amostras e cuja transformada de Fourier discreta possui M componentes nulas. Vamos supor que algumas das amostras deste sinal são modificadas. Será possível determinar o número de amostras modificadas e as suas posições e amplitudes?*

As semelhanças deste enunciado com o problema de correcção de erros levou-nos a procurar na bibliografia [Wakerly 78, Hamming 80, Clark 81, Berlekamp 84, Blahut 83, Blahut 85a, Hill 86, Sweeney 91] uma possível solução. Na realidade verificámos que o problema podia ser resolvido por métodos análogos aos usados na descodificação dos códigos BCH. A diferença consistia apenas no tipo de aritmética utilizada, dado que em vez de operar sobre um corpo numérico finito como GF(2^p), pretendíamos utilizar aritmética sobre o corpo dos reais. No entanto, estes algoritmos tinham a contrapartida de poderem ser instáveis, tendo este problema sido estudado com o objectivo de encontrar limites para os erros de reconstrução.

Nesta introdução começamos por abordar a reconstrução de sinais discretos e limitados em frequência. Faremos uma breve descrição de alguns algoritmos conhecidos para a reconstrução de sinal quando se conhecem as posições das amostras erradas. A seguir a uma breve introdução aos códigos de correcção de erros com aritmética real, enumeramos os resultados originais e finalmente uma descrição da organização da tese.

1.2 Reconstrução de sinal

Em todos os sistemas de aquisição de sinal podem ocorrer erros que corrompem o sinal em apenas algumas das suas amostras. Outras vezes o próprio sistema de aquisição possui limitações que impossibilitam a aquisição de sinal de forma adequada ocorrendo, por exemplo, períodos mais ou menos longos sem qualquer informação. Vamos considerar o caso em que um

sinal foi amostrado e que algumas das suas amostras foram apagadas[1] em posições arbitrárias. Se não for conhecida qualquer característica do sinal adquirido, então, não é possível recuperar a amplitude das amostras apagadas a partir das restantes, mas se o sinal for limitado em banda (por exemplo passa-baixo), então, tal operação de reconstrução já pode ser possível. A este tipo de reconstrução costuma-se chamar de interpolação ou extrapolação consoante as amostras desconhecidas são esparsas ou contíguas, não existindo contudo qualquer outra diferença. O problema de interpolação pode ser equacionado para o caso de sinais contínuos no domínio do tempo e limitados em banda [Marks II 91]. O teorema fundamental da teoria da amostragem[2] estabelece que se um sinal $x(t)$ passa-baixo tiver uma largura de banda B e for amostrado a uma frequência $f_s \geq 2B$, então é possível recuperar o sinal $x(t)$ a partir das amostras. Se o sinal for amostrado à frequência mínima $f_s = 2B$, teremos para a equação de reconstrução

$$x(t) = \sum_{n=-\infty}^{\infty} x\left(\frac{n}{2B}\right) \operatorname{sinc}(2Bt - n),$$

onde a função sinc é dada por

$$\operatorname{sinc}(t) = \frac{\sin(\pi t)}{\pi t}.$$

Se o sinal $x(t)$ for sobre-amostrado a uma frequência $f_s = 2W > 2B$, teremos, por exemplo,

$$x(t) = \sum_{n=-\infty}^{\infty} x\left(\frac{n}{2W}\right) \operatorname{sinc}(2Wt - n),$$

e uma vez que neste caso se tem

$$x(t) = x(t) * 2B \operatorname{sinc}(2Bt)$$

podemos escrever

$$\begin{aligned} x(t) &= \sum_{n=-\infty}^{\infty} x\left(\frac{n}{2W}\right) \operatorname{sinc}(2Wt - n) * 2B \operatorname{sinc}(2Bt) = \\ &= \beta \sum_{n=-\infty}^{\infty} x\left(\frac{n}{2W}\right) \operatorname{sinc}(\beta(2Wt - n)) \end{aligned} \tag{1.1}$$

em que o factor de sobre-amostragem é dado por

$$\beta = \frac{B}{W} < 1,$$

e $*$ representa a operação de convolução (ver o apêndice A para uma descrição completa da notação utilizada). Na figura 1.1 podemos observar a reconstrução dada pela equação

[1] Em alguns sistemas de comunicações existem regeneradores com "soft decision" que indicam que a amostra recebida tem uma amplitude errada.

[2] Este teorema é atribuído normalmente a Shannon [Shannon 48],mas há autores que se lhe referem como o teorema WKS (Whittaker, Kotelnikov e Shannon). As referências [Marks II 91,Higgins 85] fazem uma revisão histórica sobre a autoria deste resultado da teoria da informação.

(1.1) quando $\beta = 1$, verificando-se que o valor do sinal reconstruído num determinado instante de amostragem $t_k = kT_a$ depende apenas do valor da amostra $x(kT_a)$. No caso da sobre-amostragem, nos instantes de amostragem o sinal reconstruído $x(t)$ depende não só da amplitude da amostra nesse instante, mas também de todas as outras (figura 1.1). Por exemplo para $t = 0$ temos

$$x(0) = \beta \sum_{n=-\infty}^{\infty} x\left(\frac{n}{2W}\right) \operatorname{sinc}(\beta n),$$

que evidencia a dependência linear entre amostras. Se isolarmos o termo para $n = 0$ na equação anterior e resolvendo para $x(0)$ obtemos

$$x(0) = \frac{\beta}{1-\beta} \sum_{n\neq 0} x\left(\frac{n}{2W}\right) \operatorname{sinc}(\beta n).$$

A equação anterior especifica completamente o valor da amostra na origem, $x(0)$, em função das restantes amostras.

Nesta dissertação estaremos especialmente interessados numa classe de sinais que se pode designar por polinómios trigonométricos. Estes sinais podem ser descritos como uma soma finita de exponenciais, ou uma série de Fourier truncada,

$$x(t) = \sum_{n=0}^{K-1} c_n e^{-\mathrm{j}2\pi nt/T},$$

em que c_n são os coeficientes da série de Fourier e T o período da série trigonométrica. Estes sinais são completamente especificados por K coeficientes espectrais, ou por K amostras temporais uniformes mais o período T. Neste caso podemos ter discretização nos dois domínios (tempo e frequência) que o sinal fica representado sem ambiguidade. Se tivermos sobre-amostragem, o número de amostras $N > K$ nos dois domínios é superior ao necessário, tendo-se $M = N - K$ coeficientes c_n nulos. O factor de sobre-amostragem β será definido neste caso pela relação

$$\beta = K/N.$$

1.2.1 Reconstrução de polinómios trigonométicos

Tal como foi referido na secção anterior no caso dos polinómios trigonométricos um conjunto finito de amostras temporais é suficiente para representar sem ambiguidade este tipo de sinal. Por esse motivo, daqui para a frente falaremos apenas de sinais discretos representados por vectores da forma $x \in \mathbb{C}^N$, com amostras $x_0, x_1, \ldots, x_{N-1}$. A definição usada para a transformada de Fourier de um sinal x é dada por $\hat{x} = Fx$, em que a matriz de Fourier F é dada por

$$F_{pq} = \frac{1}{\sqrt{N}} e^{-\mathrm{j}\frac{2\pi}{N}pq}. \tag{1.2}$$

Vamos definir agora algumas variáveis que serão utilizadas durante toda a dissertação.

- $t-$Número de erros.

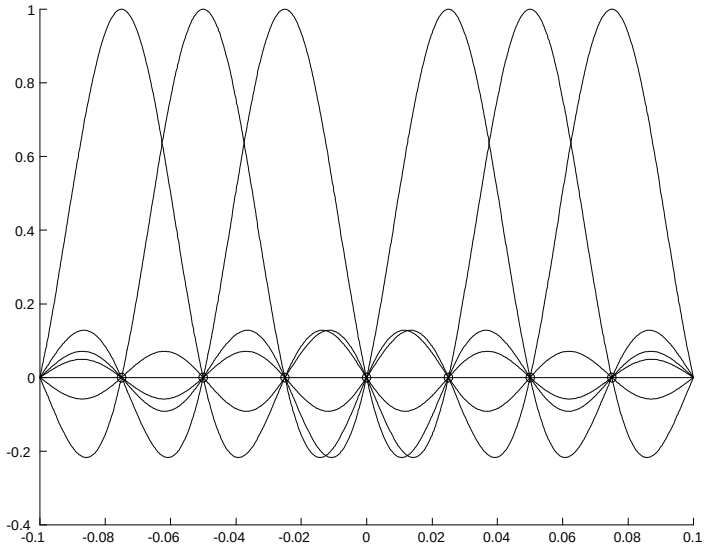


Figura 1.1:
Funções de interpolação sinc para o caso da amostragem crítica com o factor de sobre-amostragem $\beta = 1$. Repare-se que todas as curvas passam por zero nos instantes de amostragem, não existindo dependência entre amostras. Por esse motivo não é possível recuperar $x\,(0)$.

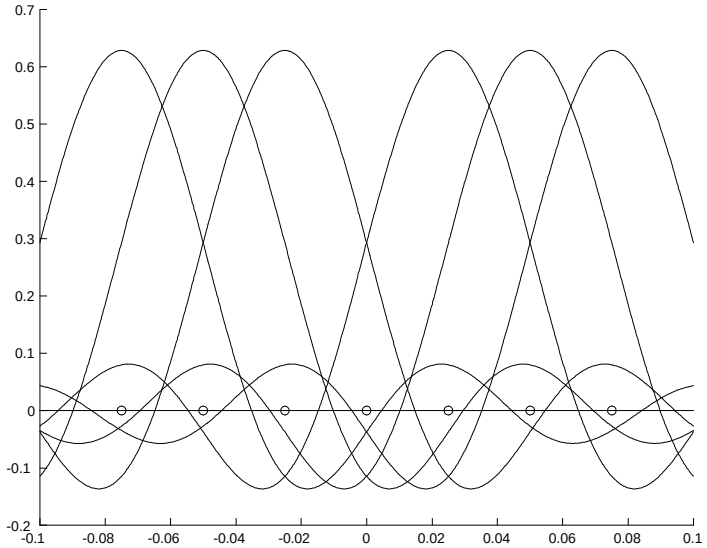


Figura 1.2:
Funções de reconstrução sinc para o caso da sobre-amostragem com $\beta \approx 0.6$. Neste caso as curvas não passam por zero nos instantes de amostragem, e existe interdependência entre as amostras. Utilizando esta informação é possível recuperar a amostra errada $x\,(0)$.

- $K-$Número de componentes espectrais desconhecidas.
- $M-$ Número de componentes espectrais conhecidas (de valor nulo).
- $\bar{S}_t-$ Conjunto dos índices das amostras erradas (desconhecidas) de um sinal x,

$$\bar{S}_t = \{i_0, i_1, i_2, \dots, i_{t-1}\} : 0 \leq i_m \leq N-1. \tag{1.3}$$

- S_t- Conjunto dos índices das $N-t$ amostras conhecidas do sinal x e complementar de $\bar{S}_t$.
- $\bar{S}_f-$ Conjunto dos índices das K amostras desconhecidas do espectro de x.
- S_f- Conjunto dos índices das M amostras nulas (conhecidas) do espectro de x. Se o sinal for passa-baixo, N par e M ímpar, o conjunto S_f é dado por

$$S_f = \left\{ \frac{N}{2} - \frac{M-1}{2}, \dots, \frac{N}{2}, \dots, \frac{N}{2} + \frac{M-1}{2} \right\}. \tag{1.4}$$

Para a formulação do algoritmo de Papoulis e Gerchberg da secção que se segue é necessário definir dois operadores em $\mathbb{C}^N$:

Definição 1 *Um operador de amostragem pode-se definir como uma matriz diagonal D de zeros e uns , com apenas $N-t$ elementos da diagonal principal unitários, em que $t > 0$. Os elementos nulos da diagonal principal estarão nas linhas de D com índices dados por $\bar{S}_t$.*

Definição 2 *Um operador de limitação de banda de um sinal x, pode ser caracterizado pela matriz $B = F^{-1}\Gamma F$, onde Γ, é uma matriz de amostragem constituída por zeros e uns com K elementos unitários na diagonal principal. Os elementos nulos da diagonal principal de Γ estarão nas linhas com índices dados por S_f.*

Acerca da definição anterior, é interessante verificar que esta operação pode ser encarada como uma amostragem no domínio da frequência. No operador de amostragem pode-se definir a densidade d como sendo a relação $(N-t)/N$, e no operador de limitação em banda podemos definir a largura de banda w como a relação K/N.

1.2.2 Algoritmo de Papoulis-Gerchberg

Vamos supor que um sinal $x \in \mathbb{C}^N$ passa-baixo, sofreu t erros em posições conhecidas. Vamos colocar o valor das amostras erradas a zero e calcular a transformada de Fourier deste sinal. Devido aos erros, o espectro obtido terá normalmente um valor diferente de zero nas componentes com índices $i \in S_f$. Se forçarmos essas componentes a zero e calcularmos uma transformada inversa obtemos um sinal em que o valor das t amostras erradas terá tomado um valor diferente de zero e mais próximo do valor correcto. No entanto, com esta operação também se alterou a amplitude das restantes $N-t$ amostras cujo valor correcto é conhecido, podendo-se portanto repô-las. Para continuar este processo, calcula-se a DFT deste sinal e repetem-se os passos anteriores. Este algoritmo pode ser descrito como projecções sucessivas entre os dois domínios, tempo e frequência, repondo-se em cada um deles a informação conhecida. Trata-se também de um caso de projecções alternadas em conjuntos convexos (POCS),

e os primeiros trabalhos conhecidos devem-se a [Papoulis 75] e [Gerchberg 74], podendo-se encontrar em [Ferreira 94a] uma revisão histórica.

Este algoritmo pode ser formulado recorrendo aos operadores de limitação em frequência e de amostragem. Consideremos o sinal x_n como o resultado da iteração n do algoritmo, então, a sequência de operações descrita anteriormente pode ser colocada na forma

$$x^{(n+1)} = \mu D x^{(0)} + (I - \mu D)\, B x^{(n)}, \tag{1.5}$$

onde I é a matriz identidade e μ uma constante. Definindo o operador T_1, como

$$T(\cdot) = \mu D x^{(0)} + (I - \mu D)\, B(\cdot),$$

a equação anterior toma uma forma mais compacta

$$x^{(n+1)} = T\left(x^{(n)}\right)$$

Foi demonstrado [Ferreira 94a] que se a densidade d da operação de amostragem D for superior à largura de banda w e se $0 < \mu < 2$, então T é estritamente não expansivo e a equação (1.5) converge para a solução única

$$x = \lim_{n\to\infty} T^n x^{(0)}.$$

A convergência deste algoritmo depende de vários factores como o padrão das amostras perdidas e o factor de sobre-amostagem. Para um estudo da convergência deste algoritmo e do valor óptimo do parâmetro μ, ver [Ferreira 94a].

1.2.3 Algoritmos não iterativos de dimensão mínima

É possível de forma não iterativa resolver o problema de reconstrução de sinal definido na secção anterior. Os algoritmos que vamos descrever são designados de dimensão mínima porque os sistemas de equações a resolver possuem uma dimensão idêntica ao número de amostras desconhecidas no domínio do tempo ou ao número de amostras desconhecidas no domínio da frequência. Inicialmente será feita uma descrição não matricial do método de reconstrução de dimensão mínima no domínio do tempo, com o objectivo de permitir uma melhor compreeensão do mesmo, seguindo-se uma descrição conjunta dos dois métodos na forma matricial. Esta última descrição será baseada nos trabalhos de Ferreira [Ferreira 96] e Walsh [Walsh 96], e resulta numa formulação compacta e geral do problema.

Reconstrução de dimensão mínima no tempo (descrição não matricial)

Considere-se o sinal $x \in \mathbb{C}^N$ limitado em frequência, com transformada de Fourier

$$\hat{x}_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x_n e^{-\mathrm{j}\frac{2\pi}{N}nk}, \tag{1.6}$$

e transformada inversa

$$x_n = \frac{1}{\sqrt{N}} \sum_{k\in\bar{S}_f} \hat{x}_k e^{\mathrm{j}\frac{2\pi}{N}nk}. \tag{1.7}$$

Separando a equação (1.6) em duas partes

$$\hat{x}_k = \frac{1}{\sqrt{N}} \left[\sum_{p \in S_t} x_p e^{-\mathrm{j}\frac{2\pi}{N}pk} + \sum_{q \in \bar{S}_t} x_q e^{-\mathrm{j}\frac{2\pi}{N}qk} \right],$$

e substituindo na equação (1.7) temos

$$x_n = \frac{1}{N} \sum_{k \in \bar{S}_f} \left[\sum_{p \in S_t} x_p e^{-\mathrm{j}\frac{2\pi}{N}pk} + \sum_{q \in \bar{S}_t} x_q e^{-\mathrm{j}\frac{2\pi}{N}qk} \right] e^{\mathrm{j}\frac{2\pi}{N}nk}, \qquad n \in \bar{S}_t,$$

que pode ainda tomar a forma

$$x_n = \frac{1}{N} \sum_{k \in \bar{S}_f} \left[\sum_{p \in S_t} x_p e^{\mathrm{j}\frac{2\pi}{N}k(n-p)} + \sum_{q \in \bar{S}_t} x_q e^{\mathrm{j}\frac{2\pi}{N}k(n-q)} \right], \qquad n \in \bar{S}_t.$$

Se definirmos o sinal das amostras desconhecidas no tempo $u_k = x_{i_k}$, podemos colocar a equação anterior na forma matricial

$$u = Su + h, \tag{1.8}$$

em que

$$\begin{aligned} S_{ab} &= \frac{1}{N} \sum_{k \in \bar{S}_f} e^{\mathrm{j}\frac{2\pi}{N}k(i_a - i_b)}, \qquad i_a, i_b \in \bar{S}_t \wedge a \neq b, \\ S_{aa} &= \frac{K}{N}, \qquad a = b \end{aligned} \tag{1.9}$$

e

$$h_a = \frac{1}{N} \sum_{p \in S_t} \sum_{k \in \bar{S}_f} x_p e^{\mathrm{j}\frac{2\pi}{N}k(i_a - p)}, \qquad i_a \in S_t. \tag{1.10}$$

Podemos obter u a partir da equação (1.8) resolvendo-a para u

$$u = (I - S)^{-1} h, \tag{1.11}$$

sendo a matriz a inverter de dimensão $t \times t$. Da equação (1.10) é possível verificar que o vector h pode ser calculado de forma eficiente recorrendo à FFT.

Reconstrução de dimensão mínima no tempo (descrição matricial)

Considere-se um vector x com transformada de Fourier $\hat{x} = Fx$, em que em cada um dos domínios existem amostras conhecidas e desconhecidas. Rearranjando as linhas e colunas da matriz de Fourier a equação (1.2) pode escrever-se

$$\begin{bmatrix} \hat{x}_c \\ \hat{x}_d \end{bmatrix} = \begin{bmatrix} \mathfrak{A} & \mathfrak{B} \\ \mathfrak{C} & \mathfrak{D} \end{bmatrix} \begin{bmatrix} x_d \\ x_c \end{bmatrix} \tag{1.12}$$

e

$$\begin{bmatrix} x_d \\ x_c \end{bmatrix} = \begin{bmatrix} \mathfrak{A}^\dagger & \mathfrak{C}^\dagger \\ \mathfrak{B}^\dagger & \mathfrak{D}^\dagger \end{bmatrix} \begin{bmatrix} \hat{x}_c \\ \hat{x}_d \end{bmatrix}, \tag{1.13}$$

onde $x_c = x(S_t) \in \mathbb{C}^{N-t}$, $x_d = x\left(\bar{S}_t\right) \in \mathbb{C}^t$, são as amostras conhecidas e desconhecidas no tempo e $\hat{x}_c = \hat{x}(S_f) \in \mathbb{C}^M$, $\hat{x}_d = \hat{x}\left(\bar{S}_f\right) \in \mathbb{C}^K$, são as amostras conhecidas e desconhecidas na frequência. A partir de (1.13), podemos escrever

$$x_d = \mathfrak{A}^\dagger \hat{x}_c + \mathfrak{C}^\dagger \hat{x}_d, \tag{1.14}$$

mas como a parte conhecida do espectro é constituída por componentes nulas vem

$$x_d = \mathfrak{C}^\dagger \hat{x}_d \tag{1.15}$$

Como

$$\hat{x}_d = \mathfrak{C} x_d + \mathfrak{D} x_c,$$

teremos

$$x_d = \mathfrak{C}^\dagger \mathfrak{C} x_d + \mathfrak{C}^\dagger \mathfrak{D} x_c,$$

que resolvendo para x_d dá

$$\boxed{x_d = \left(I - \mathfrak{C}^\dagger \mathfrak{C}\right)^{-1} \mathfrak{C}^\dagger \mathfrak{D} x_c.} \tag{1.16}$$

A equação anterior é a solução de dimensão mínima no domínio do tempo. Se na equação anterior se substituir

$$\begin{aligned} S &= \mathfrak{C}^\dagger \mathfrak{C}, && (1.17)\\ h &= \mathfrak{C}^\dagger \mathfrak{D} x_c, && (1.18)\\ u &= x_d, && (1.19) \end{aligned}$$

obtém-se a equação (1.11).

Para uma solução não iterativa de dimensão mínima no domínio da frequência, podemos a partir das equações (1.12) e (1.13), obter uma equação que dê o valor das amostras desconhecidas na frequência em função das conhecidas no tempo. Da equação (1.12) resulta

$$\hat{x}_d = \mathfrak{C} x_d + \mathfrak{D} x_c, \tag{1.20}$$

mas se substituirmos x_d de (1.15) em (1.20), e atendendo ao facto de as amostras conhecidas na frequência $\hat{x}_c$ serem nulas, teremos

$$\hat{x}_d = \mathfrak{C}\mathfrak{C}^\dagger \hat{x}_d + \mathfrak{D} x_c,$$

que resolvendo para $\hat{x}_d$ vem

$$\boxed{\hat{x}_d = \left(I - \mathfrak{C}\mathfrak{C}^\dagger\right)^{-1} \mathfrak{D} x_c}. \tag{1.21}$$

Tal como pretendido esta equação calcula o valor das amostras desconhecidas do espectro em função das amostras conhecidas no tempo. A dimensão do sistema de equações a resolver é igual ao número K de amostras desconhecidas na frequência. Este método não iterativo de reconstrução foi estudado em [Grochenig 93] e uma análise sobre a dualidade destes dois métodos de dimensão mínima foi realizada em [Ferreira 96].

Não é difícil verificar que estas duas soluções (1.16) e (1.21) são equivalentes à solução de mínimos quadrados da equação directa obtida de (1.12). Partindo de

$$\hat{x}_c = \mathfrak{A} x_d + \mathfrak{B} x_c$$

e sabendo que $\hat{x}_c = 0$, temos

$$\boxed{x_d = -\mathfrak{A}^{+} \mathfrak{B} x_c}, \tag{1.22}$$

onde $\mathfrak{A}^{+}$é a pseudo-inversa [Golub 83] de $\mathfrak{A}$ definida por

$$\mathfrak{A}^{+} = \left(\mathfrak{A}^{\dagger}\mathfrak{A}\right)^{-1} \mathfrak{A}^{\dagger}.$$

A partir das equações (1.12) e (1.13), podemos escrever

$$\begin{bmatrix} \mathfrak{A}^{\dagger} & \mathfrak{C}^{\dagger} \\ \mathfrak{B}^{\dagger} & \mathfrak{D}^{\dagger} \end{bmatrix} \begin{bmatrix} \mathfrak{A} & \mathfrak{B} \\ \mathfrak{C} & \mathfrak{D} \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}$$

ou

$$\begin{array}{ll} \mathfrak{A}^{\dagger}\mathfrak{A} + \mathfrak{C}^{\dagger}\mathfrak{C} = I & \mathfrak{A}^{\dagger}\mathfrak{B} + \mathfrak{C}^{\dagger}\mathfrak{D} = 0 \\ \mathfrak{B}^{\dagger}\mathfrak{A} + \mathfrak{D}^{\dagger}\mathfrak{C} = 0 & \mathfrak{B}^{\dagger}\mathfrak{B} + \mathfrak{D}^{\dagger}\mathfrak{D} = I \end{array} . \tag{1.23}$$

Então, a equação (1.16) pode ser escrita como

$$x_d = -\left(\mathfrak{A}^{\dagger}\mathfrak{A}\right)^{-1} \mathfrak{A}^{\dagger}\mathfrak{B} x_c = -\mathfrak{A}^{+}\mathfrak{C} x_c,$$

e como de (1.13) se pode relacionar $\hat{x}_d$ com x_d

$$x_d = \mathfrak{C}^{\dagger} \hat{x}_d,$$

então

$$\hat{x}_d = \mathfrak{C}^{-\dagger} \mathfrak{A}^{+} x_c,$$

provando que a solução dos dois métodos de dimensão mínima, é igual à solução de mínimos quadrados do problema de reconstrução [Walsh 96, Walsh 98].

Não é difícil demostrar [Ferreira 94b] que o algoritmo de Papoulis-Gerchberg (PG) também converge para a mesma solução que os algoritmos anteriores. Se colocarmos a equação iterativa (1.5) do algoritmo PG numa forma mais adequada, e se por conveniência fizermos $\mu = 1$ e $y = Dx$, obtém-se

$$x^{(n+1)} = (I - D)\, B x^{(n)} + y.$$

Pressupondo $y\left(\bar{S}_t\right) = 0$, podemos escrever a equação anterior na forma

$$x_k^{(n+1)} = \sum_{j \in \bar{S}_t} B_{kj} x_j^{(n)} + \sum_{j \in S_t} B_{kj} y_j,$$

e se $k \in \bar{S}_t$, a equação anterior pode ser escrita na forma

$$x_d^{(n+1)} = Sx_d^{(n)} + h,$$

e fazendo uso da identidades (1.17), (1.18), (1.23) temos

$$x_d^{(n+1)} = \left(I - \mathfrak{A}^\dagger \mathfrak{A}\right) x_d^{(n)} - \mathfrak{A}^\dagger \mathfrak{B} x_c.$$

Se assumirmos convergência, então,

$$x_d^{(\infty)} = \left(I - \mathfrak{A}^\dagger \mathfrak{A}\right) x_d^{(\infty)} - \mathfrak{A}^\dagger \mathfrak{B} x_c,$$

ou seja

$$\mathfrak{A}^\dagger \mathfrak{A} x_d^{(\infty)} = -\mathfrak{A}^\dagger \mathfrak{B} x_c$$

e como a matriz $\mathfrak{A}^\dagger\mathfrak{A}$ tem inversa se $t \leq M$, obtemos finalmente a solução de mínimos quadrados dada pela pseudo inversa de A tal como na equação (1.22)

$$x_d^{(\infty)} = -\left(\mathfrak{A}^\dagger \mathfrak{A}\right)^{-1} \mathfrak{A}^\dagger \mathfrak{B} x_c,$$

$$\boxed{x_d^{(\infty)} = -\mathfrak{A}^{+} \mathfrak{B} x_c}.$$

1.3 Códigos de correcção de erros

O armazenamento ou transmissão de informação digital está sujeita a erros que degradam o seu conteúdo. A ideia de adicionar informação redundante à mensagem transmitida de modo a que no receptor se possa verificar a sua integridade foi desenvolvida nos anos 50 por Hamming [Hamming 50]e Golay [Golay 49], tendo estes criado os primeiros códigos detectores e correctores de erros. Desde essa altura a teoria associada aos códigos de correcção de erros evoluiu bastante no contexto da disciplina da Teoria da Informação. Um dos avanços mais importantes foi a descoberta dos códigos BCH por Bose, Chaudhuri (1960) e Hocquenghem (1950), e que viriam a ser aplicados com sucesso nos mais variados campos nomeadamente na codificação da informação dos discos compactos, nas comunicações espaciais, em comunicações via satélite, etc. A estrutura destes códigos permitiu a obtenção de resultados teóricos importantes sendo possível projectar um código BCH de forma a garantir a correcção de um número pré-determinado de erros.

Na figura 1.3 podemos observar um diagrama de blocos que descreve um sistema de codificação e descodificação do tipo BCH e que manipula os dados por blocos. Um sinal mensagem com K amostras é codificado obtendo-se um bloco com $N > K$ amostras tendo-se acrescentado $N - K$ amostras de informação redundante.

O descodificador faz uso desssa redundância para conseguir corrigir os erros ocorridos durante a transmissão/armazenamento da informação, que na figura 1.3 se representam por e.

O paralelismo com a reconstrução de sinal é evidente, residindo a principal diferença no facto de na reconstrução de sinal não ser normalmente possível controlar a forma como o sinal foi gerado e a posição dos erros ser desconhecida. Contudo, por diversas razões, estas

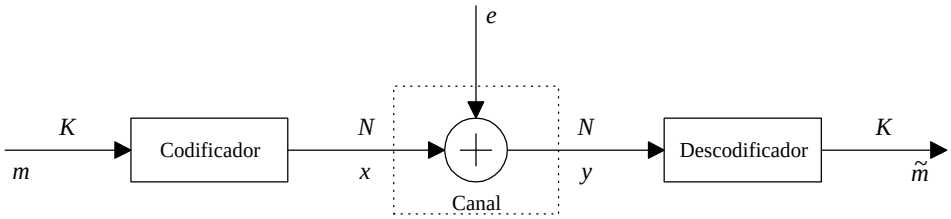


Figura 1.3: Um sinal mensagem m com K amostras é codificado dando origem ao sinal a transmitir/arquivar x com N amostras. A redundância introduzida permitirá ao descodificador corrigir os erros e ocorridos no canal.

duas disciplinas mantiveram-se de algum modo afastadas criando cada uma delas resultados e soluções equivalentes de forma independente. Segundo Blahut [Blahut 83], uma das razões para este afastamento resulta do facto de os "algebristas" dos códigos de correcção de erros trabalharem com aritmética em corpos finitos, enquanto que na área do processamento digital de sinal se trabalha com aritmética real e complexa. São assim razões históricas e de formação académica dos investigadores que levaram a que a linguagem e notação utilizada fossem diferentes, criando a ilusão nos que trabalham em cada uma das disciplinas de não existirem aspectos comuns.

Um dos primeiros autores a verificar a relação existente entre os códigos de correcção de erros e a possibilidade de se utilizar alguns dos resultados e algoritmos no corpo dos números reais foi Richard E. Blahut, que nos seus artigos [Blahut 79, Blahut 85a] chega mesmo a chamar à Teoria dos Códigos de Correcção de erros processamento digital de sinal em corpos finitos. Numa série de artigos [Marshall Jr 79, Marshall Jr 81, Marshall Jr 82a, Marshall Jr 82b, Marshall Jr 82c, Marshall Jr 83, Marshall Jr 84, Marshall Jr 86, Marshall Jr 87a, Marshall Jr 87b], T. G. Marshall aborda o problema da realização prática dos códigos de correcção de erros no corpo dos complexos, sendo de salientar o seu principal trabalho [Marshall Jr 84] em que para além de estudar os códigos por blocos apresenta igualmente alguns resultados para o caso dos códigos convolucionais. Até à actualidade vários autores exploraram as ligações existentes entre estas duas disciplinas, tais como Kumaresan [Kumaresan 85], Ja-Ling Wu [Wu 92, Wu 95, Shiu 95, Shiu 96], Jack Keil Wolf [Wolf 67, Wolf 83], Farokh Marvasti [Marvasti 87, Marvasti 93, Marvasti 97, Marvasti 99, Wong 95] e David L. Sprague [Sprague 82].

Os códigos de correcção de erros no corpo dos números complexos possuem a vantagem de não estarem limitados quanto ao tamanho do bloco, e de poderem ser realizados de forma eficiente em qualquer microprocessador. Têm no entanto a desvantagem de estarem sujeitos a erros de arredondamento, o que levanta o problema da estabilidade numérica dos algoritmos de reconstrução. Como veremos no capítulo seguinte, os algoritmos de detecção da posição das amostras erradas, e de correcção da sua amplitude, requerem a resolução de um sistema de equações, para além de outras etapas. Assim, a repercursão dos erros de arredondamento efectuados durante os cálculos no resultado da descodificação é determinado pelo condicionamento do sistema de equações a resolver. Como veremos existem vários factores que influenciam o condicionamento dos problemas a resolver:

- O número de erros
- O padrão dos erros

- O factor de sobre-amostragem
- O tamanho do bloco de dados

Apesar de um código BCH no corpo dos reais ser teoricamente capaz capaz de corrigir $M/2$ erros, devido aos problemas de estabilidade poderão existir alguns padrões de erro que serão descodificados incorrectamente. Por esta razão focámos a nossa atenção no estudo da estabilidade do problema de reconstrução em função dos parâmetros enumerados anteriormente e também no estudo da combinatória dos padrões de erro.

Para que os códigos de correcção de erros possam ter algum interesse prático é necessário encontrar métodos que permitam projectá-los. Como afirmámos anteriormente, quando projectamos um código BCH no corpo dos reais, este pode não ser capaz de corrigir todas os padrões possíveis com $M/2$ erros, sendo por conseguinte importante determinar quantos e quais é que são descodificados erradamente e no caso de descodificação errada determinar se a solução é "muito diferente" da exacta. Com este fim em vista definimos a distância mínima entre erros como forma de classsificar os padrões de erro e obtivemos (capítulo 5) uma forma de contar o número de padrões de erro que possuem uma dada distância mínima.

Combinando estes resultados com os resultados obtidos no capítulo 3 e em [Ferreira 99], é possível determinar de forma aproximada quantos dos padrões de erro é que se conseguem corrigir.

A realização prática de códigos de correcção de erros em corpos finitos requer a utilização de "hardware" ou "software" específico capaz de realizar de forma eficiente as operações de codificação e descodificação. Em aplicações em que esse "hardware" não existe a utilização de codificação no corpo dos reais pode ser uma alternativa atraente, mesmo que a capacidade de correcção destes seja inferior. Um exemplo de aplicação é na difusão de informação em redes de comunicação por pacotes que não permite a confirmação da correcta recepção. Se se acrescentar redundância aos pacotes transmitidos existe a possibilidade de recuperar a informação de pacotes perdidos [Bolot 95, Bolot 96, Hu 96] melhorando a fiabilidade da comunicação.

Em sistemas de multiprocessamento é importante assegurar de algum modo que as operações numéricas sejam realizadas correctamente e que no caso de ocorrerem erros que estes sejam automaticamente corrigidos. A ideia base consiste em incluir nos próprios algoritmos de processamento um esquema de codificação no corpo dos reais que permita sem recurso a hardware específico corrigir os erros. Esta é uma área interessante mas que requer algum aprofundamento para se encontrarem técnicas que lidem com o problema dos erros numéricos de forma a evitar falsas detecções [Anfinson 88, Agarwal 73, Agarwal 74, Huang 84, Nair 90].

Reconstrução de sinal e os códigos de correcção de erros

Os códigos de correcção de erros podem ser formulados numa forma matricial idêntica à utilizada na secção anterior utilizando-se igualmente a transformada de Fourier. Consideremos então que pretendíamos codificar uma mensagem m com K amostras de modo a que seja possível corrigir t erros ocorridos em posições cujos índices são dados por $\bar{S}_t$. Particionando a matriz de Fourier da seguinte forma

$$F = \left[\begin{array}{cc} G & H \end{array} \right],$$

e se se chamar à matriz G de dimensão $N \times K$ de codificadora e à matriz H de dimensão $N \times M$ de verificadora da paridade, para gerar um sinal codificado a partir de m, basta fazer

$$x = Gm.$$

A equação anterior pode ser expandida para a forma

$$x = \begin{bmatrix} G & H \end{bmatrix} \begin{bmatrix} m \\ \mathbf{0} \end{bmatrix},$$

em que o vector $\mathbf{0}$ é composto por M zeros. Se considerarmos um sinal de erro e, com amostras não nulas nas posições dadas pelo conjunto $\bar{S}_t$, então o sinal corrompido é dado por

$$y = x + e.$$

No receptor, para verificar a ocorrência de erros, multiplica-se o sinal recebido y pela matriz de verificação da paridade

$$s = H^\dagger y = H^\dagger x + H^\dagger e,$$

mas como x é o resultado da codificação $Hx = 0$, então

$$s = H^\dagger e.$$

A equação anterior evidencia o facto de s (o sindroma) ser unicamente uma função do sinal de erro e.

No capítulo seguinte será descrito um método para determinar o número e a posição dos erros a partir do sindroma quando $t \leq M/2$. Este método envolve a resolução de um sistema de equações Toeplitz de ordem igual ao número de erros ocorrido.

1.4 Resultados originais

1. Resolução do problema da correcção de apagamentos recorrendo a técnicas oriundas da área dos códigos de correcção de erros. Nomeadamente através da equação recursiva (2.9), da forma matricial não recursiva da secção 2.3.2 e do algoritmo de Forney da secção 2.3.3.

2. A definição da distância mínima entre erros penso ser uma ideia original, não conhecendo qualquer trabalho que a use. Os resultados de combinatória em que obtivemos expressões para o número de padrões de erro que possuem uma dada distância mínima são também originais e de uma grande importância para o projecto de códigos de correcção de erros no corpo dos números reais.

3. A decomposição da matriz A descrita na secção 3.3.2, em que se conseguiu isolar o efeito da amplitude dos erros da sua posição para efeitos do estudo do condicionamento numérico de A.

4. A técnica proposta para a descodificação dos códigos de Reed-Muller, apesar de não ser completamente nova, vem contudo corrigir alguns aspectos da solução apresentada por J.-L. Wu [Shiu 96].

5. Identificação das ligações existentes entre os bancos de filtros e os códigos convolucionais, nomeadamente o facto de que se o banco de filtros possuir a propriedade de reconstrução perfeita pode-se ter um código convolucional capaz de corrigir erros.

6. Demonstração da equivalência da utilização da transformada DFT com uma base do tipo $e^{j\frac{2\pi}{N}lm}$, (em que l e m são primos relativos), na codificação dos códigos do tipo BCH, com a utilização de um entrelaçamento regular das amostras do sinal transmitido.

1.5 Organização da tese

A tese está organizada em capítulos e estes por sua vez em secções e sub-secções. O primeiro capítulo pretende apresentar de forma breve os métodos de reconstrução de sinal quando se conhece a posição dos erros. Estabelece igualmente a analogia destes métodos com os códigos de correcção de erros, enunciando igualmente resultados que pensamos serem originais.

O capítulo 2 descreve alguns métodos de correcção de erros quando se desconhece a posição dos erros, fazendo também uma descrição dos diferentes algoritmos exitentes para a resolução de sistemas de equações Toeplitz.

O capítulo 3 aborda o problema da estabilidade dos algoritmos para determinação da posição e amplitude dos erros.

O capítulo 4 começa com uma descrição dos códigos lineares de correcção de erros no corpo dos números reais, seguindo-se os códigos cíclicos incluindo os BCH. Este capítulo termina com uma descrição dos códigos convolucionais no corpo dos números reais fazendo analogias com os bancos de filtros com a propriedade da reconstrução perfeita.

No capítulo 5 aborda-se o problema da combinatória dos padrões de erro que satisfazem uma dada distância mínima entre erros e determina-se o número de padrões com uma dada distância mínima.

No final são apresentadas as conclusões e mencionados alguns problemas em aberto e sugestões para trabalho futuro.

Capítulo 2

Problemas não lineares de reconstrução

As técnicas de reconstrução de sinal descritas no capítulo anterior supunham sempre que se conhecia a posição das amostras erradas. Neste capítulo vamos ver que é possível determinar dentro de certos limites a posição das amostras erradas fazendo uso de uma informação não utilizada nos métodos do capítulo anterior: a amplitude das amostras erradas. O problema pode-se formular do seguinte modo

Proposição 1 *Se um sinal com N amostras contiver M componentes espectrais contíguas conhecidas, então é possível encontrar a posição de $t \leq \lfloor M/2 \rfloor$ amostras que tenham sido alteradas resolvendo um sistema de equações Toeplitz de ordem t.*

Os sinais considerados serão sempre de dimensão finita e podem ser encarados como sinais periódicos com período N. A proposição anterior pode ser vista como codificação de um sinal para o tornar robusto a erros, e na realidade existem vários códigos correctores de erros que podem ser interpretados desta forma. Blahut foi dos primeiros a verificar que um código BCH se caracteriza por os vectores pertencentes ao código possuirem M componentes espectrais nulas contíguas e verificou que os algoritmos de correcção de erros podiam ser aplicados não só em corpos finitos mas também no corpo dos complexos. Um sinal que tenha algumas componentes espectrais nulas, vai possuir uma dependência linear entre as suas amostras. Este mesmo problema aparece em predição linear dando origem às equações de Yule-Walker, em estimação espectral em que temos o problema dual, ou seja, determinar a posição das componentes espectrais, etc.

Este problema não linear de reconstrução pode ainda ser resolvido recorrendo a uma generalização do método de Prony [Prony 95]. A escolha deste método como ponto de partida, deveu-se ao facto de na sua versão original se encontrarem formulados todos os passos fundamentais para a solução do problema: determinação da equação recursiva cujos parâmetros podem ser determinados resolvendo um sistema Toeplitz e que permitem determinar a posição e amplitude dos erros resolvendo um sistema de equações Vandermonde.

Quanto à organização deste capítulo, começaremos por formular o problema de reconstrução com determinação das amostras erradas. Descreveremos o método de Prony na sua versão original seguida da sua aplicação ao nosso problema. Sistematizam-se quatro formas diferentes para o sistema de equações, descrevendo em seguida as condições para se obter um sistema de equações com uma matriz hermítica. Uma versão simplificada do algoritmo com codificação na frequência e que recorre a apenas uma transformada, é analisada seguida de

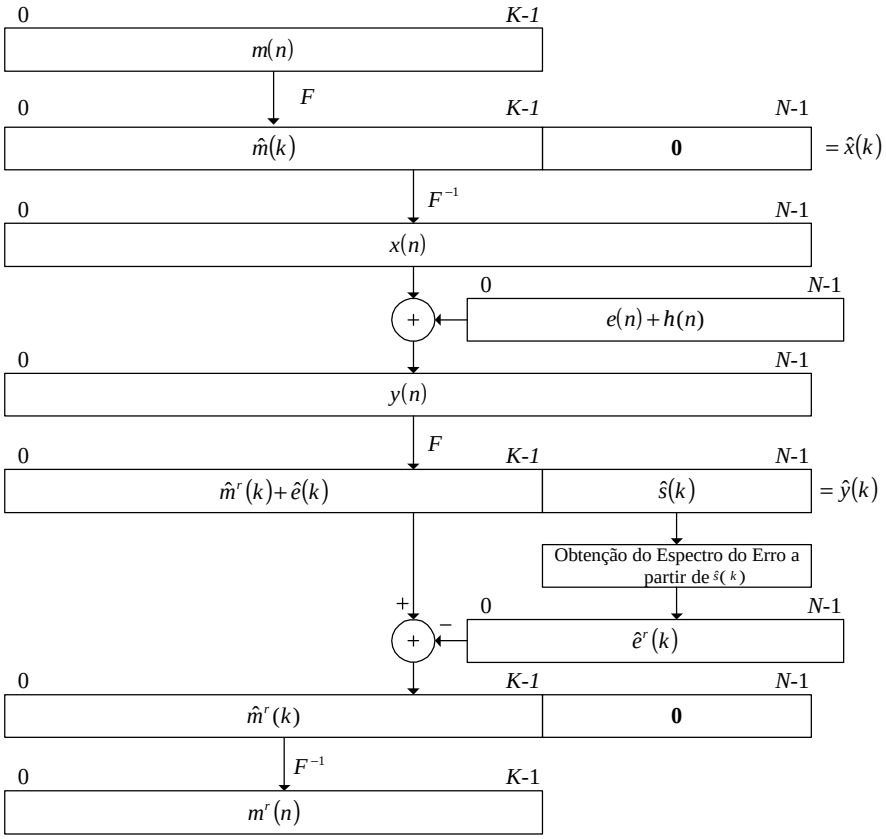


Figura 2.1: Diagrama de blocos com o algoritmo de codificação e descodificação no tempo de um sinal $m \in R^K$. Para M amostras nulas acrescentadas, este código é capaz de corrigir $t = M/2$ erros ocorridos durante a transmissão ou armazenamento. No caso da figura as amostras nulas foram acrescentadas no final do bloco perdendo-se as condições de simetria do espectro que permitem ao sinal x ter parte imaginária nula.

alguns exemplos numéricos. O sinal de erro é sempre determinado recorrendo à solução do sistema de equações Toeplitz realizando uma extrapolação do espectro do sinal de erro de forma directa ou indirecta, sendo descritas e comparadas várias técnicas. Finalmente teremos uma comparação de vários algoritmos para a resolução de sistemas Toeplitz, e a forma como estes apesar de terem sido criados para resolver problemas aparentemente diferentes e em áreas distintas são de facto equivalentes.

2.1 Reconstrução de sinal com detecção das amostras erradas

Vamos agora apresentar um algoritmo de codificação e reconstrução, em que é possível detectar a posição das amostras erradas. Tal como anteriormente vamos considerar apenas sinais discretos de dimensão finita pertencentes a $\mathbb{C}^N$, em que um sinal é um vector de dimensão $N \in \mathbb{N}$ dado por $x = [x_0\ x_1 \dots x_{N-1}]^T$. Os sinais $x \in \mathbb{C}^N$ podem ser encarados como funções complexas definidas no grupo finito $\mathbb{Z}_N$, tendo portanto período N, ou seja, $x(i+N) = x(i)$. Considere-se a transformada de Fourier discreta F definida por (1.2) em que tal como em (1.6) e (1.7), $\hat{x}$ é a transformada de x. Na figura 2.1 pode-se observar um diagrama de blocos

que ilustra o conjunto de operações a efectuar sobre um sinal, de modo a que seja possível corrigir até $M/2$ amostras corrompidas com ruído impulsivo. Entende-se neste caso por ruído impulsivo um sinal de dimensão N em que apenas t amostras são diferentes de zero e em que t é normalmente muito menor que N. Por enquanto abordaremos apenas o caso em que N é par e K ímpar, sendo os restantes casos analisados na secção 2.2.3. Para este caso se pretendermos gerar um sinal codificado com parte imaginária nula temos de garantir as condições de simetria do espectro de x. Com a transformada de Fourier e blocos de N amostras, os M zeros a inserir terão de ser em número ímpar para que tal seja assegurado. Como veremos, nesta situação só é possível corrigir um máximo de $(M-1)/2$ erros não se obtendo por conseguinte um código do tipo CDM.

Considere-se um sinal discreto $m \in \mathbb{R}^K$ a que vamos chamar *mensagem*, com transformada de Fourier dada por $\hat{m}$. Se a $\hat{m}$ se acrescentarem M amostras de valor nulo, vamos ter um novo vector de dimensão $N = K + M$ com a seguinte forma

$$\hat{x} = [\hat{m}(0), \ldots, \hat{m}\left(\frac{K-1}{2}\right), 0, \ldots, 0, \hat{m}\left(\frac{K+1}{2}\right), \ldots, \hat{m}(K-1)],$$

em que os M zeros foram colocados nas posições com os índices dados por (1.4). Tal como anteriormente S_f dá-nos os índices das amostras conhecidas do espectro que neste caso têm valor nulo. O conjunto S_f não necessita de estar centrado em $\frac{N}{2}$ e pode generalizar-se para a forma que será usada mais à frente

$$S_f = \left\{ r - \frac{M-1}{2}, \ldots, r, \ldots, r + \frac{M-1}{2} \right\}, \tag{2.1}$$

com $r \in [0..N-1]$ sendo o deslocamento circular. Aplicando a transformada de Fourier inversa a este sinal obtem-se o sinal x que constitui uma versão sobreamostrada de m por um factor de $\beta = \frac{K}{N}$. Dado que este sinal possui um certo grau de redundância é possível recuperar o sinal original se um número limitado de amostras forem corrompidas. Suponhamos que o sinal x foi corrompido obtendo-se

$$y = x + e + \eta,$$

em que e é o sinal com ruído impulsivo com t amostras diferentes de zero, e cujos índices são dados pelo conjunto $\bar{S}_t$ definido em (1.3). O sinal η é ruído de pequena amplitude que foi introduzido para representar os erros provocados pela quantificação do sinal antes de ser transmitido/armazenado, possui uma amplitude muito menor do que e, afecta todas as amostras de x e depende da implementação do algoritmo.

Considere-se a transformada de Fourier $\hat{y}$ do sinal recebido y. No caso particular de $e_n = 0$, $\hat{y}$ terá M amostras de valor aproximadamente nulo devido a η, nas posições dadas por S_f. Quando ocorrem alguns erros, essas amostras terão um valor diferente de zero e só dependem do sinal de erro e, sendo $\hat{y}(S_f) = \hat{e}(S_f)$, uma "janela" sobre o espectro de e conhecida na terminologia da TCCE como *sindroma*. Se a partir do sindroma for possível extrapolar os restantes valores de $\hat{e}$ então, calculando a transformada inversa de $\hat{e}$, e como $x^r = y^r - e^r$ temos (desprezando o efeito do ruído η) o sinal original reconstruído $m^r = F^\dagger \hat{m}^r$ em que $\hat{m}^r = \hat{x}^r\left(\bar{S}_f\right)$.

Como veremos mais à frente, sob certas condições é possível reconstruir o sinal original m recorrendo ao método de Prony, que descreveremos sucintamente na secção seguinte na sua versão original.

2.2 Método de Prony para determinar a posição dos erros

O método de reconstrução aqui descrito, é conhecido na área da TCCE como uma implementação espectral de um código do tipo BCH e as suas ligações com a área do PDS foram reconhecidas e aprofundadas por Blahut [Blahut 83, Blahut 79, Blahut 85a], e também por Wolf, Marshall, Kumaresan, [Wolf 67, Wolf 83, Marshall Jr 82a, Marshall Jr 82b, Marshall Jr 82c, Marshall Jr 83, Marshall Jr 84, Marshall Jr 86, Marshall Jr 87a, Marshall Jr 87b, Kumaresan 82, Kumaresan 85] e mais recentemente por Marvasti, Shiu e Wu [Marvasti 87, Marvasti 91, Marvasti 92, Marvasti 93, Marvasti 94, Marvasti 97, Shiu 95, Wu 92, Wu 95, Shiu 96]. Um dos aspectos mais relevantes das ligações entre as duas áreas é a possibilidade dos códigos de correcção de erros poderem ser utilizados no corpo dos complexos. Vários autores estudaram este problema, e verificaram que muitos dos códigos existentes num corpo finito têm um análogo em $\mathbb{C}$ [Kumaresan 85, Marshall Jr 81, Marshall Jr 84, Wolf 83, Marvasti 93].

Apesar da existência do código análogo existem algumas diferenças entre as implementações de um mesmo código em cada um dos corpos. Enquanto que nos corpos finitos a dimensão N dos vectores não é arbitrária e está relacionada com o número de bits b das amostras (símbolos em terminologia TCCE), no corpo $\mathbb{C}$ não existe essa limitação. Por outro lado, em $\mathbb{C}$ vamos ter erros de arredondamento que não acontecem nos corpos finitos, com os inevitáveis problemas de condicionamento na resolução de sistemas de equações. Apesar disso os códigos em $\mathbb{C}$ apresentam a grande vantagem de poderem ser implementados em qualquer processador de forma eficiente, não sendo necessário utilizar hardware específico que implemente a aritmética finita.

O método de reconstrução que mais à frente descreveremos, e que permite determinar a posição de erros resolvendo um sistema de equações Toeplitz, foi identificado por Wolf [Wolf 83] como sendo o método de Prony de interpolação descrito no seu artigo original [Prony 95], o qual pode ser encontrado em versões mais actualizadas em [Hildebrand 56] e [Wolf 67]. O método de Prony é igualmente utililizado hoje em dia em estimação espectral em casos em que se sabe que o sinal a estudar possui pouco ruído e é modelizável como uma soma de sinusóides (também conhecido por polinómio trigonométrico).

O método de Prony na sua versão original

O Barão de Prony publicou em 1795 um artigo [Prony 95] em que descrevia um método para modelizar a função da expansão de gases. Este modelo considerava que esta função era uma soma finita de exponenciais reais e a principal contribuição de Prony consistiu em encontrar um método não linear para calcular os parâmetros destas exponenciais conhecendo um certo número de pontos da função. O método de Prony pretende assim aproximar uma função $f(x)$ por uma soma finita de n exponenciais reais do tipo:

$$f(x) \approx \sum_{k=1}^{n} C_k e^{b_k x}, \tag{2.2}$$

onde C_k e b_k são incógnitas reais. Para aplicar o método de Prony é necessário conhecer $2n$ pontos de $f(x)$ para valores de x equidistantes na forma $x_i = i\Delta + \Delta_0$ em que Δ é o passo e Δ_0 o valor inicial que vamos assumir nulo. Vamos admitir que a função $f(x)$ é uma soma de exponenciais como na equação (2.2), tendo-se uma igualdade em vez de uma aproximação.

A equação (2.2) vem então com $z_k = e^{\Delta b_k}$

$$f_i = \sum_{k=1}^{n} C_k z_k^i. \tag{2.3}$$

Se se encontrar um método para calcular os coeficientes de amortecimento b_k, podemos determinar C_k resolvendo o sistema de equações Vandermonde

$$\begin{bmatrix} z_1^0 & z_2^0 & \cdots & z_n^0 \\ z_1^1 & z_2^1 & \cdots & z_n^1 \\ \vdots & \vdots & \ddots & \vdots \\ z_1^{n-1} & z_2^{n-1} & \cdots & z_n^{n-1} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_n \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_{n-1} \end{bmatrix}. \tag{2.4}$$

Prony verificou que a equação (2.3) é a solução de uma equação linear de diferenças, homogénea e de coeficientes constantes. As raízes desta equação são da forma $z_k = e^{\Delta b_k}$ e podemos definir o polinómio cujos zeros são z_k por

$$P(z) = \prod_{k=1}^{n} (z - z_k).$$

Se expandirmos a equação anterior como uma série de potências teremos

$$P(z) = \sum_{i=0}^{n} h_i z^i,$$

e como $P(z_k) = 0$ podemos escrever

$$\sum_{i=0}^{n} h_i z_k^i = 0.$$

Multiplicando a expressão anterior por $C_k e^{-\Delta b_k m}$ e somando em k, teremos

$$\sum_{i=0}^{n} h_i \sum_{k=1}^{n} C_k e^{\Delta b_k (i-m)} = 0.$$

A expressão anterior contém a definição (2.2) de $f(i-m)$ o que nos permite finalmente escrever

$$\sum_{i=0}^{n} h_i f(i-k) = 0.$$

Com a equação anterior e os $2n$ elementos conhecidos de $f(x)$ podemos formar um sistema de equações Toeplitz e determinar os coeficientes h_i do polinómio $P(z)$, por meio de factorização os seus zeros, e finalmente os parâmetros b_k. Para calcular as amplitudes das exponenciais C_k, basta resolver o sistema de equações Vandermonde (2.4).

Aplicação do método de Prony à reconstrução de sinal

O método original de Prony descrito anteriormente, considerava unicamente exponenciais reais. No entanto, o método pode ser facilmente generalizado para considerar sinusóides de amplitude constante fazendo C_i complexo e b_i imaginário puro, que é o caso que nos interessa.

O espectro $\hat{e}$ do sinal de erro pode ser interpretado como um polinómio trigonométrico composto por t exponenciais, e por esta razão o método de Prony não dá uma aproximação de $\hat{e}$ mas o valor exacto. O algoritmo que a seguir se descreve, não é mais do que o método de Prony aplicado ao nosso problema de extrapolação do espectro $\hat{e}$.

Considere-se a transformada de Fourier do sinal de erro

$$\hat{e}(k) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} e(n) e^{-\mathrm{j}\frac{2\pi}{N}nk} = \frac{1}{\sqrt{N}} \sum_{m=0}^{t-1} e(i_m) e^{-\mathrm{j}\frac{2\pi}{N}i_m k}, \tag{2.5}$$

em que $i_m \in \bar{S}_t$ são os índices das amostras erradas. Fazendo a mudança de variável

$$z_m = e^{-\mathrm{j}\frac{2\pi}{N}i_m},$$

vem

$$\hat{e}(k) = \frac{1}{\sqrt{N}} \sum_{m=0}^{t-1} e(i_m) z_m^k. \tag{2.6}$$

O objectivo é determinar a posição dos erros i_m e a sua amplitude $e\,(i_m)$. O método de Prony permite determinar em primeiro lugar a posição dos erros e uma vez estes conhecidos calcular a sua amplitude. A equação (2.6) é a solução de uma equação linear homogénea de diferenças com os zeros dados por z_m, que é conhecida na terminologia da TCCE como o polinómio localizador de erros

$$P(z) = \sum_{i=0}^{t} h_i z^{-i}, \tag{2.7}$$

em que

$$P(e^{-\mathrm{j}\frac{2\pi}{N}i_m}) = 0, \qquad (m = 0, 1, \ldots, t-1).$$

Substituindo em (2.7) $z_m = e^{-\mathrm{j}\frac{2\pi}{N}i_m}$, multiplicando por $\frac{1}{\sqrt{N}} e(i_m) e^{-\mathrm{j}\frac{2\pi}{N}i_m k}$ e somando em m, temos

$$\sum_{i=0}^{t} h_i \frac{1}{\sqrt{N}} \sum_{m=0}^{t-1} e(i_m) e^{-\mathrm{j}\frac{2\pi}{N}i_m(k-i)} = 0.$$

Utilizando a equação (2.5), na equação anterior obtemos

$$\sum_{i=0}^{t} h_k \hat{e}(k-i) = 0. \tag{2.8}$$

Como se conhecem $2t+1$ valores de $\hat{e}$ (mais um do que o necessário) cujos índices são dados por (2.1), pode-se construir um sistema de equações para as t incógnitas h_k, em quatro formas diferentes, que passamos a expor.

Recursão para trás

Neste método cada amostra de índice mais baixo é uma combinação linear das seguintes. Podemos distinguir dois casos, um em que surge uma matriz de Toeplitz, e outro em que surge uma matriz Hankel.

Caso 1 $h_t = 1$ **e** $k = r, r+1, \ldots, r+t-1$ Para este caso, a equação (2.8) pode-se escrever na forma recursiva

$$\hat{e}_{k-t} = -\sum_{i=0}^{t-1} h_i \hat{e}_{k-i}, \tag{2.9}$$

dando origem ao sistema de equações

$$\begin{bmatrix} \hat{e}_r & \hat{e}_{r-1} & \cdots & \hat{e}_{r-t+1} \\ \hat{e}_{r+1} & \hat{e}_r & \cdots & \hat{e}_{r-t+2} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{e}_{r+t-1} & \hat{e}_{r+t-2} & \cdots & \hat{e}_r \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_{t-1} \end{bmatrix} = - \begin{bmatrix} \hat{e}_{r-t} \\ \hat{e}_{r-t+1} \\ \vdots \\ \hat{e}_{r-1} \end{bmatrix}, \tag{2.10}$$

que se pode escrever na forma compacta $Ah = b_1$, em que a matriz A é uma matriz Toeplitz.

Caso 2 $h_t = 1$ **e** $k = r+t-1, \ldots, r+1, r$ Para este caso, aplica-se igualmente a equação (2.9) mas percorrendo as componentes do sindroma de forma inversa dando origem ao sistema de equações

$$\begin{bmatrix} \hat{e}_{r+t-1} & \hat{e}_{r+t-2} & \cdots & \hat{e}_r \\ \hat{e}_{r+t-2} & \hat{e}_{r+t-3} & \cdots & \hat{e}_{r-1} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{e}_r & \hat{e}_{r-1} & \cdots & \hat{e}_{r-t+1} \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_{t-1} \end{bmatrix} = - \begin{bmatrix} \hat{e}_{r-1} \\ \hat{e}_{r-2} \\ \vdots \\ \hat{e}_{r-t} \end{bmatrix}, \tag{2.11}$$

que pode ser escrito na forma compacta $Bh = b_2$, em que a matriz B é Hankel. É possível escrever o sistema de equações nesta forma a partir do obtido no caso anterior com a transformação $B = JA$

$$JAh = Jb_1 \Longleftrightarrow Bh = b_2 \tag{2.12}$$

em que J é a matriz

$$J = \begin{bmatrix} 0 & 0 & 0 & \cdots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 1 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 0 \end{bmatrix}. \tag{2.13}$$

Recursão para a frente

Neste método cada amostra de índice mais elevado é uma combinação linear das anteriores. Tal como na recurção para trás, podemos distinguir igualmente dois casos, um em que surge uma matriz de Toeplitz, e outro em que surge uma matriz Hankel.

Caso 3 $\alpha_0 = 1$ **e** $k = r+1, r+2, \ldots, r+t$ Para este caso vamos escrever a equação (2.8) utilizando outra notação para os parâmetros h, que se passarão a designar por α. Assim, a equação (2.8) resulta em

$$\hat{e}_k = -\sum_{i=1}^{t} \alpha_i \hat{e}_{k-i}, \tag{2.14}$$

dando origem ao sistema de equações

$$\begin{bmatrix} \hat{e}_r & \hat{e}_{r-1} & \cdots & \hat{e}_{r-t+1} \\ \hat{e}_{r+1} & \hat{e}_r & \cdots & \hat{e}_{r-t+2} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{e}_{r+t-1} & \hat{e}_{r+t-2} & \cdots & \hat{e}_r \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_t \end{bmatrix} = - \begin{bmatrix} \hat{e}_{r+1} \\ \hat{e}_{r+2} \\ \vdots \\ \hat{e}_{r+t} \end{bmatrix}. \tag{2.15}$$

Podemos escrever este sistema na sua forma compacta $A\alpha = b_3$ em que a matriz A é Toeplitz e igual à matriz A do caso 1.

Caso 4 $\alpha_0 = 1$ **e** $k = r+t, r+t-1, \ldots, r+1$ Para este caso, aplica-se igualmente a equação (2.14) mas percorrendo as componentes do sindroma de forma inversa dando origem ao sistema de equações

$$\begin{bmatrix} \hat{e}_{r+t-1} & \hat{e}_{r+t-2} & \cdots & \hat{e}_r \\ \hat{e}_{r+t-2} & \hat{e}_{r+t-3} & \cdots & \hat{e}_{r-1} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{e}_r & \hat{e}_{r-1} & \cdots & \hat{e}_{r-t+1} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_t \end{bmatrix} = - \begin{bmatrix} \hat{e}_{r+t} \\ \hat{e}_{r+t-1} \\ \vdots \\ \hat{e}_{r+1} \end{bmatrix}, \tag{2.16}$$

que pode ser escrita na forma $B\alpha = b_4$, em que a matriz B é igual à do caso 2. Tal como se demonstrou ser possível transformar o caso 1 no 2, o mesmo se verifica entre o caso 3 e 4.

Conhecendo o valor das variáveis h_i ou α_i, pode-se determinar a totalidade do espectro de $\hat{e}$ através da equação recursiva (2.9) para os sistemas na forma 1 e 2 e pela equação (2.14) para os casos 3 e 4. Outra possibilidade consiste em resolver o sistema de equações Vandermonde que se constrói a partir da equação (2.6)

$$\frac{1}{\sqrt{N}} \begin{bmatrix} z_0^r & z_1^r & \cdots & z_{t-1}^r \\ z_0^{r+1} & z_1^{r+1} & \cdots & z_{t-1}^{r+1} \\ \vdots & \vdots & \ddots & \vdots \\ z_0^{r+t-1} & z_1^{r+t-1} & \cdots & z_{t-1}^{r+t-1} \end{bmatrix} \begin{bmatrix} e_{i_0} \\ e_{i_1} \\ \vdots \\ e_{i_{t-1}} \end{bmatrix} = \begin{bmatrix} \hat{e}_r \\ \hat{e}_{r+1} \\ \vdots \\ \hat{e}_{r+t-1} \end{bmatrix}. \tag{2.17}$$

Os parâmetos α_i e h_i, estão como é evidente, relacionados entre si. Da equação (2.14) pode-se escrever

$$\begin{array}{c} \sum_{i=0}^{t} \alpha_i \hat{e}_{k-i} = 0 \Leftrightarrow \\ \sum_{i=0}^{t-1} \alpha_i \hat{e}_{k-i} = -\alpha_t \hat{e}_{k-t} \end{array},$$

combinando a equação anterior com a equação (2.9) temos

$$\sum_{i=0}^{t-1} \alpha_i \hat{e}_{k-i} = \alpha_t \sum_{i=0}^{t-1} h_i \hat{e}_{k-i},$$

que se pode escrever na forma matricial

$$\boldsymbol{\alpha} = \alpha_t \mathbf{h}, \tag{2.18}$$

com

$$\boldsymbol{\alpha} = [\alpha_0, \alpha_1, \ldots, \alpha_t] \tag{2.19a}$$
$$\mathbf{h} = [h_0, h_1, \ldots, h_t]\,. \tag{2.19b}$$

De (2.18) resulta a seguinte relação entre as variáveis α_i, e h_i:

$$h_i = \frac{\alpha_i}{\alpha_t}, \quad i = \{0, 1, 2, \ldots, t\} \tag{2.20}$$

com $\alpha_0 = 1$ e $h_t = 1$.

2.2.1 Generalização do sindroma S

Na definição de sindroma dada em (2.1), os índices das suas amostras eram contíguos. No entanto, pode-se considerar um sindroma com posições arbitrárias para as suas amostras. Para obter essa forma considere-se que os índices das componentes espectrais do sindroma são dados por

$$S' = \{\mathbf{j}_0, \mathbf{j}_1, \mathbf{j}_2, \ldots, \mathbf{j}_{t-1}\},$$

em que $\mathbf{j}_l = [j_l, j_l + 1, \ldots, j_l + t - 1]$. Substituindo na equação (2.8) k por j_l obtem-se

$$\sum_{i=0}^{t} h_i \hat{e}\,(j_l - i) = 0.$$

Para o caso em que $h_t = 1$ e em que $l = 0, 1, \ldots, t-1$, obtemos o seguinte sistema de equações

$$\begin{bmatrix} \hat{e}_{j_0} & \hat{e}_{j_0-1} & \cdots & \hat{e}_{j_0-t+1} \\ \hat{e}_{j_1} & \hat{e}_{j_1-1} & \cdots & \hat{e}_{j_1-t+1} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{e}_{j_{t-1}} & \hat{e}_{j_{t-1}-1} & \cdots & \hat{e}_{j_{t-1}-t+1} \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_{t-1} \end{bmatrix} = - \begin{bmatrix} \hat{e}_{j_0-t} \\ \hat{e}_{j_1-t} \\ \vdots \\ \hat{e}_{j_{t-1}-t} \end{bmatrix}. \tag{2.21}$$

Como se pode constatar o sindroma é constituído não apenas por $2t$ amostras, mas por t blocos de $t + 1$ amostras que no caso limite de não existir qualquer sobreposição entre eles origina um sindroma com uma dimensão de $t \times (t + 1)$ amostras. Nos casos anteriores a sobreposição entre cada um destes blocos era de t amostras, resultando num sindroma de apenas $2t$ amostras. Convém também realçar que no caso geral a matriz quadrada do sistema de equações (2.21) não é nem Hankel nem Toeplitz.

2.2.2 Caso em que A é Hermítica

Os casos 1 e 3 referidos anteriormente em que a matriz A é Toeplitz, podem assumir uma forma mais favorável à solução e estudo do sistema de equações se se escolher $r = \frac{N}{2}$. Assim, desde que N seja par e o vector de erro $e \in \mathbb{R}^N$ pode-se escrever

$$\hat{e}_{r+i} = \hat{e}^*_{-r-i} = \hat{e}^*_{N-r-i} = \hat{e}^*_{r-i}. \tag{2.22}$$

Substituindo na equação (2.15) obtém-se o seguinte sistema de equações

$$\begin{bmatrix} \hat{e}_{N/2} & \hat{e}_{N/2-1} & \cdots & \hat{e}_{N/2-t+1} \\ \hat{e}^*_{N/2-1} & \hat{e}_{N/2} & \cdots & \hat{e}_{N/2-t+2} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{e}^*_{N/2-t+1} & \hat{e}^*_{N/2-t+2} & \cdots & \hat{e}_{N/2} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_t \end{bmatrix} = - \begin{bmatrix} \hat{e}^*_{N/2-1} \\ \hat{e}^*_{N/2-2} \\ \vdots \\ \hat{e}^*_{N/2-t} \end{bmatrix}.$$

No ponto 2.4 serão estudados em detalhe alguns métodos para a resolução deste sistema de equações, nomeadamente o de Levinson que permite determinar em simultâneo o número de erros ocorridos com uma complexidade algorítmica $O\left(t^2\right)$ ou mesmo $O\left(t \log^2 t\right)$ [Brent 80, Blahut 85b, Zhang 89, Zhang 92].

Para o caso em que a matriz A é Hermítica, a relação entre os parâmetros h e α (2.20) pode ser simplificada. Utilizando os vectores com dimensão $t+1$, definidos por (2.19a) e (2.19b) podemos demonstrar que estão relacionados por

$$Jh^* = \alpha \;\Leftrightarrow\; h^*_{t-i} = \alpha_i, \tag{2.23}$$

em que J, é a matriz definida por (2.13).

Para o caso 1 temos

$$Ah = b_1, \tag{2.24}$$

e para o caso 3 temos

$$A\alpha = b_3, \tag{2.25}$$

podendo-se utilizar as relações de simetria (2.22) para se obter a relação

$$Jb^*_1 = b_3.$$

Multiplicando por J o conjugado de (2.24) vem

$$\begin{aligned} JA^*h^* &= Jb^*_1, \\ JA^*JJh^* &= b_3 \\ AJh^* &= b_3. \end{aligned}$$

Esta última equação é idêntica a (2.25) com $\alpha = Jh^*$, tal como queríamos demonstrar.

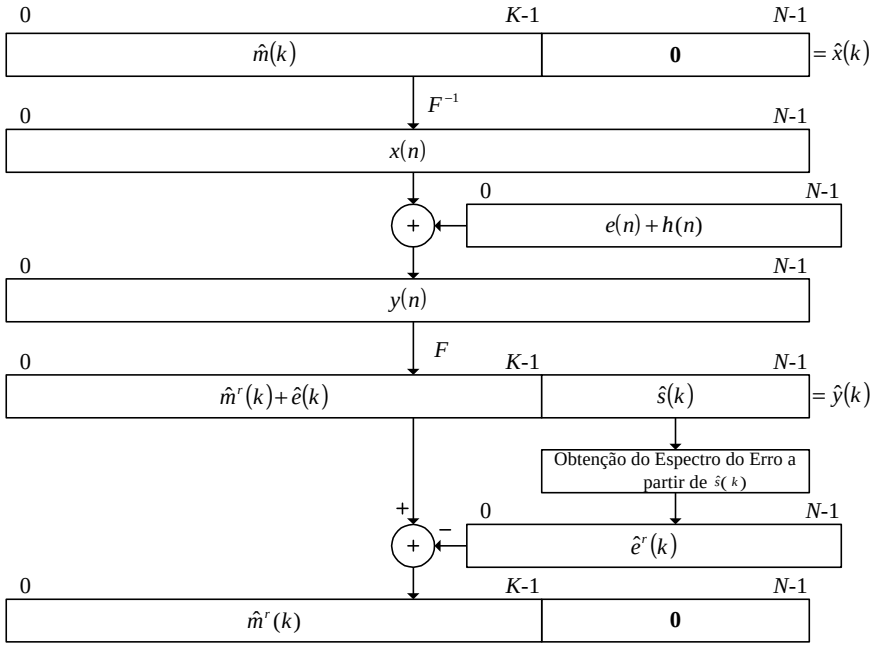


Figura 2.2: Versão do codificador e do descodificador na frequência. Como se pode constatar, são eliminadas duas transformadas em relação ao caso da codificação proposta na figura 2.1

2.2.3 Codificação simplificada no domínio da frequência.

É possível eliminar o cálculo de duas transformadas das operações de codificação / descodificação, se o objectivo for o de codificar o sinal mensagem m de modo a que seja possível corrigir t erros no destino sem nos preocuparmos com o significado físico do sinal transmitido. No algoritmo descrito na figura 2.1, é preciso acrescentar M amostras ao espectro do sinal m, sendo necessário na codificação aplicar duas transformadas, uma directa para obter $\hat{m}$ e outra inversa para se obter o sinal a transmitir x. No entanto, se considerarmos que o sinal mensagem são componentes espectrais de um sinal hipotético, basta acrescentar M amostras conhecidas e calcular a sua transformada inversa para se obter o sinal a transmitir x. Na figura 2.2 podemos ver facilmente que na descodificação basta igualmente calcular uma única transformada (directa) para se obter $\hat{y}$. Aplicando o algoritmo de extrapolação do espectro do erro a partir do sindroma, e como $\hat{m} = \hat{y} - \hat{e}$, temos o sinal mensagem original recuperado.Apesar da simplicidade deste método ele possui a grande desvantagem de o sinal a transmitir/arquivar y, pertencer a $\mathbb{C}$ e ter do dobro das amostras de $\hat{m}$.

Para tornar o sinal real é possível numa abordagem simples, manipular o sinal a m a transmitir, de modo a que este tome a forma dada pela equação (2.26), em que Θ é um vector com $2t+1$ zeros.

$$\begin{aligned} \Re\{\hat{x}\} &= \{0, m_0, \ldots, m_{K/2-1}, \Theta, m_{K/2-1}, \ldots, m_0\} \\ \Im\{\hat{x}\} &= \{0, m_{\frac{K}{2}}, \ldots, m_{K-1}, \Theta, -m_{K-1}, \ldots, m_{\frac{K}{2}}\} \end{aligned} \tag{2.26}$$

O zero extra acrescentado no início destina-se a garantir as condições de simetria em $\hat{x}$ de modo que o sinal a transmitir x seja real. Este código de correcção de erros necessita assim

de acrescentar à mensagem m, $2t+2$ zeros para conseguir corrigir t erros.

Em terminologia de TCCE a codificação anterior não é do tipo "Código de Distância Máxima" CDM, o que por outras palavras significa que acrescenta mais redundâcia do que aquela que é aproveitada. Em [Marshall Jr 84] Marshall demonstra que no corpo dos números reais ou no dos complexos, existe sempre um código CDM para qualquer valor (par ou ímpar) de N e K.

Para construir um código com N e K pares pode-se utilizar a transformada de Fourier ímpar ODFT[1] [Bellanger 89] definida pela equação

$$\hat{x}_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x_n e^{-\mathrm{j}\frac{2\pi}{N}n\left(k+\frac{1}{2}\right)}. \tag{2.27}$$

Com esta transformada consegue-se construir um código em que não é necessário acrescentar o zero extra para se garantir a simetria em $\hat{x}$, de modo a que x seja real. Podemos assim escrever o arranjo de $\hat{x}$ como

$$\begin{aligned} \Re\{\hat{x}\} &= \{m_0, \ldots, m_{\frac{K}{2}-1}, \Theta, m_{\frac{K}{2}-1}, \ldots, m_0\} \\ \Im\{\hat{x}\} &= \{m_{\frac{K}{2}}, \ldots, m_{K-1}, \Theta, -m_{K-1}, \ldots, m_{\frac{K}{2}}\} \end{aligned} . \tag{2.28}$$

Exemplos numéricos

Os exemplos de reconstrução das figuras 2.3 e 2.4 foram gerados com os seguintes valores para os parâmetros

$$N = 128, \quad K = 108, \quad t = 10 \tag{2.29}$$

$$e(i_m) = (-1)^{i_m}, \quad i_m = 5m \tag{2.30}$$

Na figura 2.4 utilizam-se duas transformadas para codificar o sinal (codificação da figura 2.1) enquanto na figura 2.3 se utiliza apenas uma (codificação da figura 2.2). Em ambos os casos foi utilizada a ODFT pois N e K são pares e esta transformada garante uma codificação do tipo CDM.

Determinação dos zeros do polinómio localizador de erros

Vamos apresentar uma técnica de calcular os zeros do polinómio localizador de erros P, utilizando a transformada de Fourier. Considere-se novamente a equação (2.7)

$$P(z_m) = \sum_{n=0}^{t} h_n z_m^{-n} = 0,$$

ou ainda

$$\sum_{n=0}^{t} h_n e^{\mathrm{j}\frac{2\pi}{N}ni_m} = 0.$$

[1] Neste caso referimo-nos à transformada de Fourier ímpar na frequência.

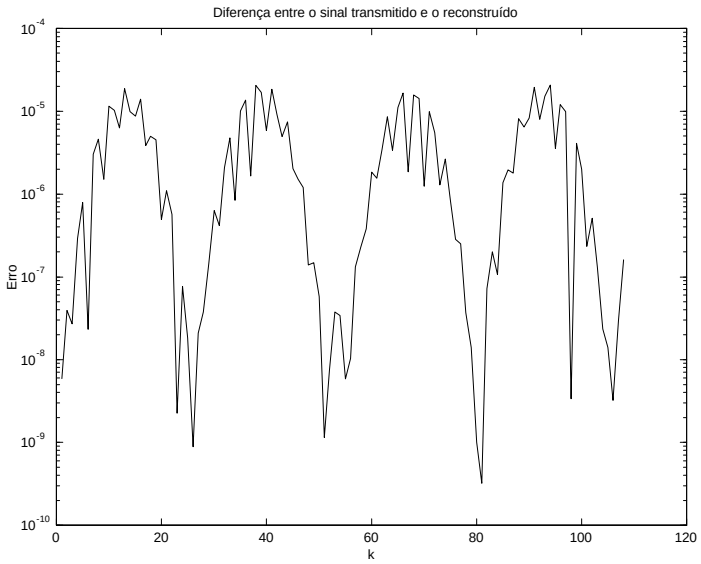


Figura 2.3:
Erro de reconstrução utilizando apenas uma transformada (ODFT), para realizar a codificação. Os parâmetros deste exemplo são os fornecidos pelas equações (2.29) e (2.30)

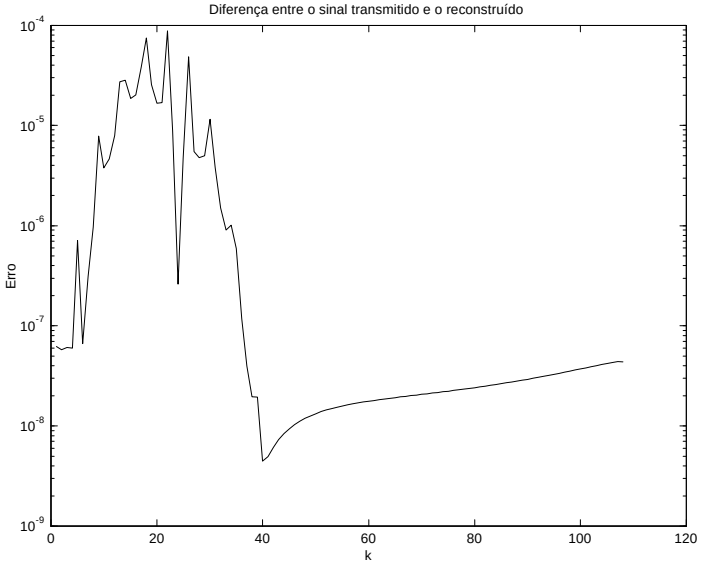


Figura 2.4:
Exemplo de reconstrução em que se utilizam duas transformadas (ODFT) para acrescentar redundância a um sinal nas mesmas condições da figura anterior.

A equação anterior pode ser escrita como a transformada inversa de Fourier $\check{h}$ em que h_n são os coeficientes de Fourier

$$\check{h}_k = \sum_{n=0}^{N-1} h_n e^{\mathrm{j}\frac{2\pi}{N}nk} \tag{2.31}$$

em que

$$\check{h}\left(\bar{S}_t\right) = 0.$$

Para determinar a posição dos zeros de $\check{h}$ basta calcular a IDFT de $h = [h_0\ h_1\ \ldots\ h_t\ \Theta]$, em que Θ representa um vector com $N - t - 1$ zeros. Para determinar a posição dos zeros do polinómio, compara-se cada componente de $\check{h}$ com um dado limiar ε. A amplitude deste limiar, depende da precisão numérica usada nos cálculos e do condicionamento do problema. Este método possui a vantagem de filtrar os erros numéricos cometidos até à determinação da posição dos erros, como aliás aconteceu no algoritmo de reconstrução não recursivo. No entanto o valor do limiar ε é díficil de calcular, sendo obtido normalmente de forma empírica. Um valor incorrecto de ε pode originar uma determinação incorrecta da posição dos erros.

Uma alternativa consiste em considerar que o número de erros ocorrido foi sempre o máximo t permitido pelo código. Neste caso ordenam-se as amostras de $\check{h}$ por ordem crescente do módulo e escolhem-se as t menores. Apesar de se determinarem posições para erros que não ocorreram, posteriormente, o algoritmo de correcção das amplitudes dará para os erros "falsos" uma amplitude muito pequena.

Estabilização numérica do cálculo da posição dos erros

Um código de correcção de erros no corpo dos números reais pode ser sobre-dimensionado de modo a garantir uma melhor estabilidade numérica na determinação da posição dos erros. Para o conseguir basta que a dimensão do sindroma seja superior ao dobro do número máximo de erros que se pretende corrigir. Nesse caso a matriz A do sistema de equações (2.15) do caso 3 (por exemplo), terá uma dimensão $(M/2 \times t_{\max})$ e b uma dimensão $(M/2 \times 1)$ obtendo-se

$$A\alpha = b$$

que tem uma solução de norma mínima [Golub 83] recorrendo à pseudo-inversa de A

$$\alpha = A^{+}b.$$

Podemos ilustrar esta técnica considerando o seguinte exemplo numérico em que o sindroma tem o dobro da dimensão necessária

$$N = 128, \quad K = 108, \quad t = 5$$

$$e\left(i_m\right) = (-1)^{i_m}, \quad i_m = m,$$

utilizando-se igualmente a ODFT para a codificação. Na figura 2.5 podemos observar o erro na determinação dos parâmetros α quando se utiliza todo o sindroma disponível (linha a cheio) e quando se utiliza só o necessário (linha a tracejado).

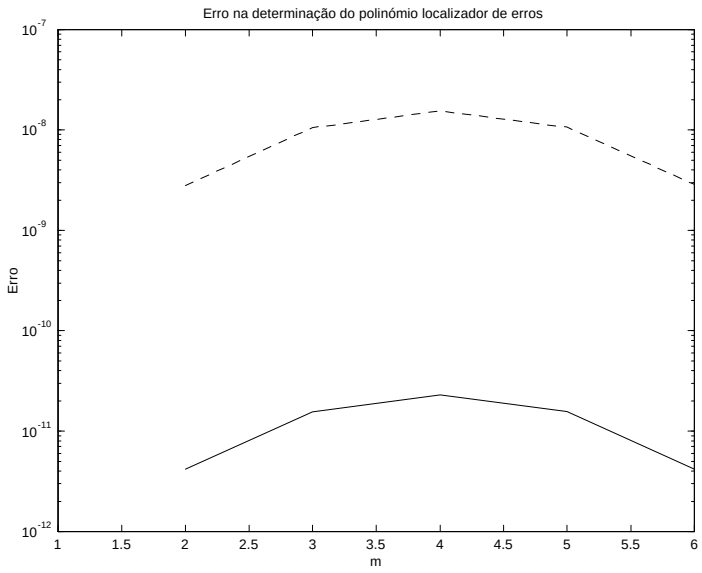


Figura 2.5:
Erro na determinação dos coeficientes do polinómio localizador de erros para o caso em que se utiliza todo o sindroma disponível (linha a cheio) e para o caso em que se utiliza somente o necessário. Como era de esperar o erro é menor no primeiro caso.

2.3 Técnicas para a reconstrução da amplitude do erro

Como se pode observar nos exemplos numéricos anteriores, a utilização da equação recursiva (2.9), para calcular o espectro do erro directamente, apesar da simplicidade de implementação, sofre do problema de propagação de erros numéricos. Por outro lado, o problema de determinar a amplitude dos erros pode resumir-se à resolução do sistema de equações Vandermonde (2.17). Serão apresentadas algumas das técnicas possíveis para a resolução deste problema.

- A primeira técnica a ser descrita consiste na utilização da mesma equação recursiva dividindo a extrapolação em duas partes com metade das amostras. Esta técnica limita apenas em parte a propagação dos erros.

- A segunda técnica consiste em encontrar uma equação matricial que permita obter qualquer componente de $\hat{e}$ a partir do sindroma de forma não recursiva [Zadeh 93a]. Apesar desta técnica ser mais robusta numéricamente, apresenta um esforço computacional mais intenso.

- A terceira técnica consiste em utilizar directamente o algoritmo de Forney [Blahut 83] que permite determinar a amplitude dos erros a partir dos coeficientes do polinómio localizador dos erros e de parte do sindroma.

- Finalmente, pode-se dividir o problema em duas partes, em que primeiro se calcula a posição dos erros a partir dos zeros do polinómio (2.7) e de seguida utiliza-se um dos métodos de reconstrução descritos no capítulo 1 para obter a amplitude dos erros.

2.3.1 Extrapolação directa recursiva bidireccional

Uma vez que com a equação recursiva (2.9) os erros se propagam e acumulam, pode-se dividir a operação de extrapolação do espectro $\hat{e}$ em duas partes utilizando a recursão (2.9) para realizar a extrapolação para trás do sindroma e a recursão (2.14) para realizar a extrapolação para a frente do sindroma. Desta forma consegue-se reduzir ligeiramente o erro de reconstrução tal como se pode vêr nas figuras 2.6 e 2.7. Repare-se que uma vez calculados os parâmtetros h_i, caso A seja Hermítica pode-se facilmente obter os α_i, usando a relação (2.20) ou (2.23). Com este método são necessárias cerca de $K \times t$ multiplicações e somas.

2.3.2 Extrapolação directa não recursiva

É possível encontrar uma equação não recursiva que permita obter directamente qualquer valor de $\hat{e}$ a partir das amostras conhecidas do Sindroma. Zadeh descreve este método em detalhe em [Zadeh 93b, Zadeh 93a], aplicando-o ao seguinte problema:

Dadas P amostras conhecidas de um sinal $v \in \mathbb{C}^N$, e sabendo que v é limitado em frequência com P componentes diferentes de zero, calcular as restantes amostras de v.

O enunciado anterior é clássico e equivalente ao problema de reconstrução de sinais pertencentes a $\mathbb{C}^N$ limitados em frequência apresentado no início do capítulo 1. Nesta secção, iremos descrever a utilização deste algoritmo à estrapolação do espectro do sinal de erro. Considere-se que se utiliza a transformada ODFT para realizar a codificação, e que são dadas $2t$ componentes de $\hat{e}$, (o dobro do necessário para obter unicamente as amplitudes dos erros) e sabe-se que o sinal no domínio dos tempos possui apenas t amostras diferentes de zero, pretende-se determinar as restantes amostras de $\hat{e}$.

Considere-se o vector

$$\hat{\mathbf{e}}_k = [\hat{e}_k, \hat{e}_{k-1}, \ldots, \hat{e}_{k-t+1}]^T,$$

então a regressão (2.14) pode-se escrever na forma matricial

$$\begin{aligned} \hat{\mathbf{e}}_{k+1} &= \begin{bmatrix} -\alpha_1 & -\alpha_2 & \cdots & -\alpha_{t-1} & -\alpha_t \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \hat{\mathbf{e}}_k \\ &= B\hat{\mathbf{e}}_k = W\Lambda W^{-1}\hat{\mathbf{e}}_k \end{aligned}$$

com W dada por

$$W = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ z_0 & z_1 & \cdots & z_{t-1} \\ \vdots & \vdots & \ddots & \vdots \\ z_0^{t-1} & z_1^{t-1} & \cdots & z_{t-1}^{t-1} \end{bmatrix}$$

e Λ dada por

$$\Lambda = \begin{bmatrix} z_0 & 0 & \cdots & 0 \\ 0 & z_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & z_{t-1} \end{bmatrix}.$$

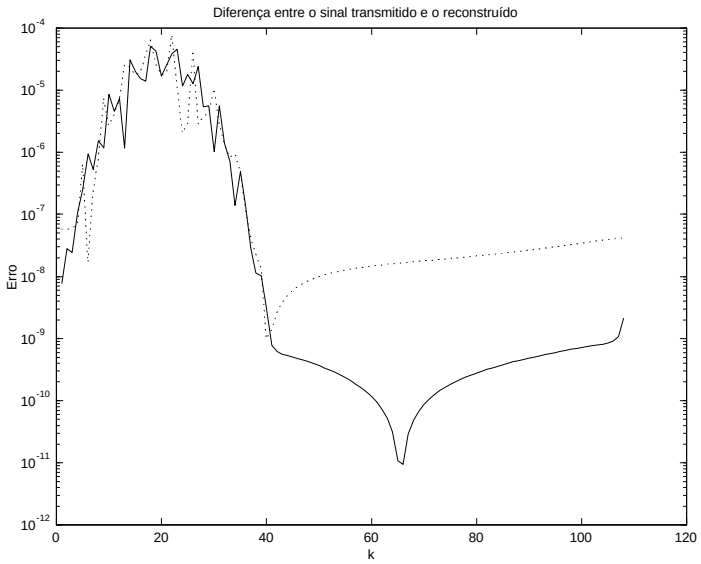


Figura 2.6: Comparação entre a extrapolação bidireccional (a cheio) e a unidirecional (a ponteado). O gráfico mostra a diferença entre o sinal mensagem transmitido m e o recebido $\hat{m}^r$, para o caso da codificação no tempo utilizando duas transformadas para codificar. No caso da extrapolação bidireccional a amplitude da diferença tende a ser menor.

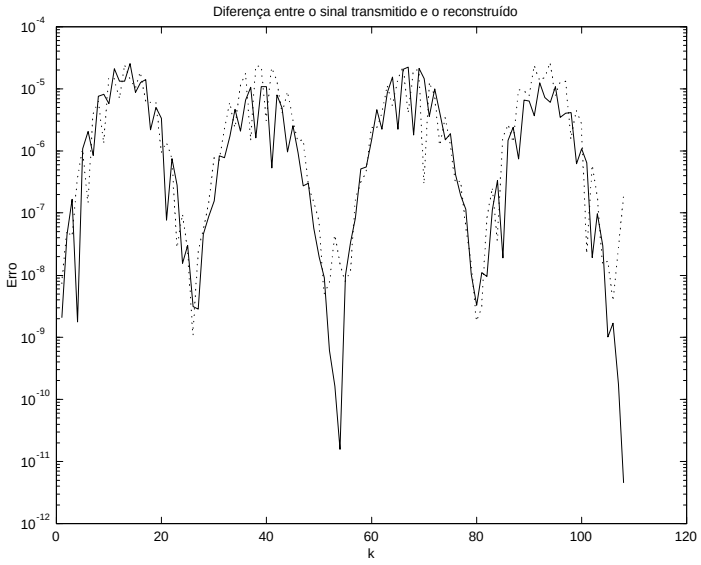


Figura 2.7: Figura idêntica à anterior mas que realiza a codificação no domínio da frequência e em que se utiliza uma só transformada para codificar. A extrapolação bidireccional (a cheio) possui um erro um pouco menor que a unidireccional (a ponteado).

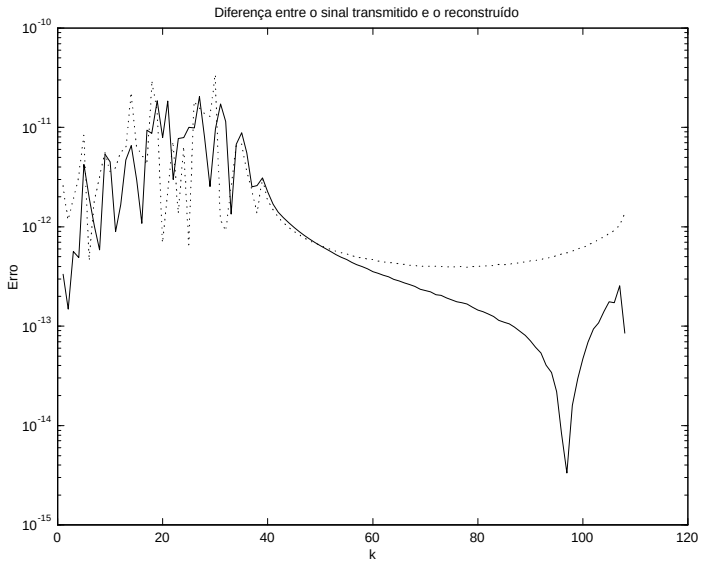


Figura 2.8: Tal como na figura 2.6 podemos observar a diferença entre o sinal original e o reconstruído mas para o caso em que se forçam as posições dos erros a serem números inteiros e positivos. A melhoria no erro de reconstrução é evidente, obtendo-se para este caso particular uma redução do erro máximo de cerca de 10^{-4} para cerca de 10^{-11}.

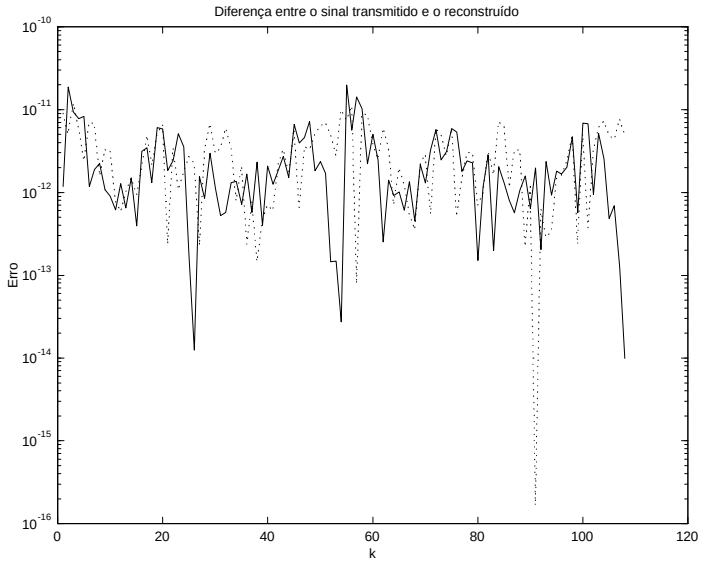


Figura 2.9: Figura idêntica à 2.7 mas tal como a anterior forçam-se as posições das amostras erradas a assumirem valores inteiros positivos. Verifica-se igualmente que a extrapolação bidireccional (a cheio), dá origem a um erro de reconstrução um pouco menor que a versão unidireccional (a ponteado).

Como se pode constatar os valores próprios de B são $z_m = e^{-\mathrm{j}\frac{2\pi}{N}i_m}$, e W é a matriz com os vectores próprios de B, porque B é a matriz companheira na forma canónica controlável [Kuo 90]. Repare-se que a matriz W é a mesma que a da equação (2.17) para $r = 0$. Para obter as componentes espectrais desconhecidas de $\hat{e}$, temos de determinar $K = N - 2t$ componentes espectrais em blocos de dimensão K/t. Para obter cada bloco procede-se do seguinte modo:

- 1. Resolve-se o sistema de equações (2.15) obtendo-se os parâmetros α_i.
  2. Constróiem-se as matrizes W, Λ e W^{-1}, por um de dois métodos:
     (a) Construir primeiro B e determinar algebricamente W, Λ e W^{-1}.
     (b) Obter a posição dos erros usando por exemplo a técnica descrita na página 26, construir as matrizes W e Λ directamente e calcular W^{-1} algebricamente.
  3. Forma-se um vector $\hat{\mathbf{e}}_k$ com dimensão t usando as amostras conhecidas de $\hat{e}$.
  4. Calcula-se cada bloco de t amostras através da equação

$$\hat{\mathbf{e}}_r = W\Lambda^{r-k}W^{-1}\hat{\mathbf{e}}_k \tag{2.32}$$

  5. Aplica-se sucessivamente a equação anterior até se ter completado a extrapolação.

A equação (2.32) resulta do facto de

$$\hat{\mathbf{e}}_{k+2} = B\hat{\mathbf{e}}_{k+1} = B^2\hat{\mathbf{e}}_k,$$

podendo então escrever-se

$$\begin{aligned}\hat{\mathbf{e}}_{k+n} &= B^n\hat{\mathbf{e}}_k = (W\Lambda W^{-1})(W\Lambda W^{-1})\ldots(W\Lambda W^{-1})\hat{\mathbf{e}}_k = \\ &= W\Lambda^n W^{-1}\hat{\mathbf{e}}_k.\end{aligned}$$

Uma vez que Λ é um sub-conjunto das N raízes da unidade, $\hat{\mathbf{e}}_k = \hat{\mathbf{e}}_{k+N}$, fica assegurado um período N para $\hat{\mathbf{e}}$.

Na figura 2.10 podemos ver dois exemplos de reconstrução em que no caso a) se aplicou a equação (2.32) directamente e no caso b) se utilizou um outro método intermédio para a obtenção da posição dos erros. Quanto ao número de cálculos a realizar em cada um dos dois casos descritos temos a operação de extrapolação comum a ambos e que necessita aproximadamente de $K\times(2t+1)$ multiplicações e somas. Para além disso temos a inicialização das matrizes W e W^{-1} e Λ:

- No caso a) o cálculo algébrico das matrizes W e Λ necessita de aproximadamente $O\left(t^3\right)$ operações, mais $O\left(t^3\right)$ para a inversão de W.
- No caso b) a obtenção das matrizes faz-se detectando primeiro a posição dos erros que consome $O\left(N\log_2 N\right)$ operações e a geração das matrizes pode ser realizada com $t + t^2$ operações, mais $O\left(t^3\right)$ para a inversão de W.

Repare-se na maior precisão dos resultados obtidos no caso b). Tal facto fica-se a dever à maior precisão numérica com que as matrizes são calculadas e à "regeneração" introduzida no cálculo da posição dos erros.

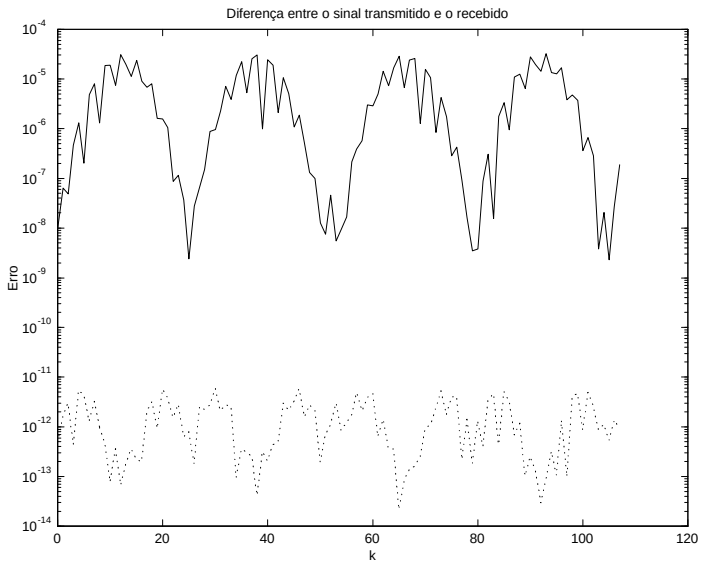


Figura 2.10: Duas versões de implementação do algoritmo de extrapolação do espectro do sinal de erro de forma não recursiva. No caso a) (a cheio) calculam-se as matrizes W, W^{-1} e Λ por métodos algébricos, no caso b) (a ponteado) calcula-se primeiro a posição dos erros e obtêm-se as mesmas raízes de forma directa.

2.3.3 Algoritmo de Forney

O algoritmo de Forney [Forney 65] determina a amplitude das amostras erradas a partir dos coeficientes do polinómio localizador de erros. Constitui quase sempre a última etapa na descodificação de códigos BCH ou Reed-Soloman quando implementados em corpos finitos. Marvasti aplicou-o recentemente [Marvasti 99]para resolver o problema de determinar a amplitude dos erros em códigos BCH sobre o corpo dos reais que é o nosso caso.

O algoritmo de Forney pode ser intrepertado como um método polinomial de resolver sistemas de equações Vandermonde sem necessidade de inversão da matriz do sistema. A exposição que aqui vamos fazer baseia-se na do livro [Blahut 83] que foi adaptada ao nosso caso e notação.

A matriz do sistema de equações (2.17) pode ser convertida para Vandermonde efectuando a seguinte modificação

$$\frac{1}{\sqrt{N}}\begin{bmatrix} z_0^0 & z_1^0 & \cdots & z_{t-1}^0 \\ z_0^1 & z_1^1 & \cdots & z_{t-1}^1 \\ \vdots & \vdots & \ddots & \vdots \\ z_0^{t-1} & z_1^{t-1} & \cdots & z_{t-1}^{t-1} \end{bmatrix}\begin{bmatrix} z_0^r e_{i_0} \\ z_1^r e_{i_1} \\ \vdots \\ z_{t-1}^r e_{i_{t-1}} \end{bmatrix} = \begin{bmatrix} \hat{e}_r \\ \hat{e}_{r+1} \\ \vdots \\ \hat{e}_{r+t-1} \end{bmatrix},$$

e fazendo a mudança de variável

$$e'_{i_m} = \frac{1}{\sqrt{N}} z_m^r e_{i_m}, \tag{2.33}$$

teremos

$$\begin{bmatrix} z_0^0 & z_1^0 & \cdots & z_{t-1}^0 \\ z_0^1 & z_1^1 & \cdots & z_{t-1}^1 \\ \vdots & \vdots & \ddots & \vdots \\ z_0^{t-1} & z_1^{t-1} & \cdots & z_{t-1}^{t-1} \end{bmatrix} \begin{bmatrix} e'_{i_0} \\ e'_{i_1} \\ \vdots \\ e'_{i_{t-1}} \end{bmatrix} = \begin{bmatrix} \hat{e}_r \\ \hat{e}_{r+1} \\ \vdots \\ \hat{e}_{r+t-1} \end{bmatrix}.$$

O polinómio localizador de erros pode ser escrito em função das suas raízes $z_m = e^{-\mathrm{j}\frac{2\pi}{N}i_m}$

$$P(z) = \prod_{i=1}^{t} \left(1 - z_i z^{-1}\right),$$

e podemos definir um polinómio associado ao sindroma como

$$\hat{e}\,(z) = \sum_{j=0}^{t-1} \hat{e}_j z^{-j} = \sum_{j=0}^{t-1} \sum_{m=0}^{t-1} e'_{i_m} z_m^j z^{-j}. \tag{2.34}$$

Defina-se agora o polinómio $\quad(z)$ como

$$(z) = \hat{e}\,(z)\, P(z) \pmod{z^t}, \tag{2.35}$$

substituindo $P(z)$ e $\hat{e}\,(z)$ pelas equações anteriores temos

$$(z) = \left[\sum_{j=0}^{t-1} \sum_{m=0}^{t-1} e'_{i_m} z_m^j z^{-j}\right] \times \left[\prod_{i=1}^{t} \left(1 - z_i z^{-1}\right)\right] \pmod{z^t},$$

e se se colocar em evidência o termo m do produtório, temos

$$(z) = \sum_{m=0}^{t-1} e'_{i_m} \left[\left(1 - z_i z^{-1}\right) \sum_{j=0}^{t-1} \left(z_m z^{-1}\right)^j\right] \times \prod_{i \neq m}^{t} \left(1 - z_i z^{-1}\right) \pmod{z^t}.$$

O termo entre parênteses rectos é uma expansão em série de $\left(1 - \left(z_m z^{-1}\right)^t\right)$, podendo-se assim escrever

$$(z) = \sum_{m=0}^{t-1} e'_{i_m} \left(1 - \left(z_m z^{-1}\right)^t\right) \times \prod_{i \neq m}^{t} \left(1 - z_i z^{-1}\right) \pmod{z^t}$$

e como a expressão anterior é calculada $\bmod z^t$ temos

$$(z) = \sum_{m=0}^{t-1} e'_{i_m} \prod_{i \neq m}^{t} \left(1 - z_i z^{-1}\right).$$

Se calcularmos o valor do polinámio $\quad(z)$ para cada raíz z_l, então, para cada parcela do somatório da equação anterior haverá um termo em que $i = l$ e que anula todo o produtório. Só quando $m = l$ é que $i \neq l$ e o produtório virá diferente de zero,

$$(z_l) = e'_{i_m} \prod_{i \neq l}^{t} \left(1 - z_i z_l^{-1}\right).$$

Da expressão anterior podemos retirar o valor da amplitude dos erros uma vez que

$$e'_{i_l} = \frac{(z_l)}{\prod_{i\neq l}^{t}\left(1 - z_i z_l^{-1}\right)},$$

e utilizando a equação (2.33), obtemos

$$e_{i_l} = \frac{z_l^{-r}\sqrt{N}\quad(z_l)}{\prod_{i\neq l}^{t}\left(1 - z_i z_l^{-1}\right)}.$$

O polinómio $\quad(z)$ pode ser obtido directamente da equação (2.35)

$$(z) = \left[\sum_{j=0}^{t-1}\hat{e}_j z^{-j}\right] \times \left[\prod_{i=1}^{t}\left(1 - z_i z^{-1}\right)\right] \quad \left(\mathrm{mod}\, z^t\right)$$

da qual se podem eliminar os termos com ordem superior a t

$$(z) = \sum_{i=0}^{t-1}\alpha_i z^{-i} \sum_{j=0}^{t-(i+1)}\hat{e}_{(r+j)} z^{-j}.$$

Versão para a ODFT

Como vimos no início deste capítulo, quando N e K são pares consegue-se um código de distância máxima se se usar a ODFT em vez da DFT para realizar a codificação e descodificação. Por esse motivo vamos deduzir as expressões anteriores para esta transformada. A diferença começa na equação (2.34) vindo esta então dada por

$$\hat{e}\,(z) = \sum_{j=0}^{t-1}\hat{e}_j z^{-j} = \sum_{j=0}^{t-1}\sum_{m=0}^{t-1} e'_{i_m} z_m^{(\mathrm{j}+1/2)} z^{-j},$$

onde se utilizou a definição da ODFT dada na equação (2.27). O polinómio $\quad(z)$ virá então dado por

$$(z) = \left[\sum_{j=0}^{t-1}\sum_{m=0}^{t-1} e'_{i_m} z_m^{(\mathrm{j}+1/2)} z^{-j}\right] \times \left[\prod_{i=1}^{t}\left(1 - z_i z^{-1}\right)\right] \quad (\mathrm{mod}\, z^t),$$

e realizando o mesmo raciocínio teremos a seguinte expressão para a amplitude do erro

$$e_{i_l} = \frac{z_l^{-r}\sqrt{N}\quad(z_l)}{z_l^{1/2}\prod_{i\neq l}^{t}\left(1 - z_i z_l^{-1}\right)}.$$

Foram realizadas algumas simulações da realização prática do algoritmo de Forney para a determinação das amplitudes dos erros. O exemplo utilizado é o mesmo que o usado anteriormente (2.29) e assume-se que a posição das amostras erradas foram calculadas sem qualquer erro. O erro de reconstrução pode ser apreciado na figura 2.11.

2.3.4 Extrapolação indirecta do espectro do erro

As raízes do polinómio $P(z)$ da equação (2.7), dão directamente a posição dos erros pois $z_m = e^{-\mathrm{j}\frac{2\pi}{N} i_m}$, ficando assim determinado o conjunto $\bar{S}_t$ (1.3) com os índices das amostras erradas. Deste modo, começando pelo sistema de equações $A\alpha = b$, podemos calcular primeiro a posição dos erros e posteriormente determinar a amplitude do sinal de erro e com um dos métodos de reconstrução descritos no capítulo anterior.

Método directo por eliminação Gaussiana

Este método consiste em aplicar o algoritmo de eliminação Gaussiana ao sistema de equações (2.17), em que a matriz Vandermonde é de ordem $t \times t$, obtendo-se a amplitude dos erros directamente das amostras do sindroma.

Na figura 2.12 podemos observar o erro de reconstrução para duas situações: uma em que a matriz Vandermonde é de ordem $t \times t$ e que portanto não tira partido de toda a informação disponível; a outra situação tira partido de todas as $2t$ amostras do sindroma calculando a pseudo-inversa com dimensão $2t \times t$. Repare-se que em todos os métodos aqui descritos, o sindroma tem o dobro da dimensão da necessária para determinar a amplitude dos erros. Se quiséssemos calcular apenas a amplitude de t erros sabendo as suas posições bastava um sindroma com t amostras.

Método de dimensão mínima no tempo

Este método foi descrito no capítulo anterior [Ferreira 92, Ferreira 94a, Ferreira 94b, Marvasti 91] e determina a amplitude de t erros resolvendo um sistema de equações Toeplitz de ordem t. No capítulo anterior deduzimos as expressões para este método utilizando a DFT e a ODFT. Como os métodos de reconstrução estudados anteriormente utilizam a ODFT, foi esta a versão utilizada para obter os resultados da figura 2.13.

Método de dimensão mínima na frequência

Este método e o anterior são duais (ver [Ferreira 96]) e os resultados obtidos na reconstrução da amplitude do sinal de erro são semelhantes. Tal como no método anterior a versão usada para gerar a figura 2.14 realiza a codificação utilizando a transformada ODFT. Este algoritmo de reconstrução foi estudado no capítulo 1 e permite determinar a amplitude dos t erros resolvendo um sistema de equações Toeplitz de ordem K em que K é o número de amostras não nulas do espectro do sinal recebido.

Análise dos resultados

Para comparar os diferentes métodos de reconstrução da amplitude do erro, considerámos duas situações distintas:

- Na primeira temos um sindroma com 10 amostras com 10 erros para determinar a amplitude e os resultados da reconstrução podem ser observados na figura 2.15.
- No segundo temos um sindroma maior com 20 amostras e igualmente 10 erros para determinar a amplitude. Os resultados da simulação podem ser observados na figura 2.16.

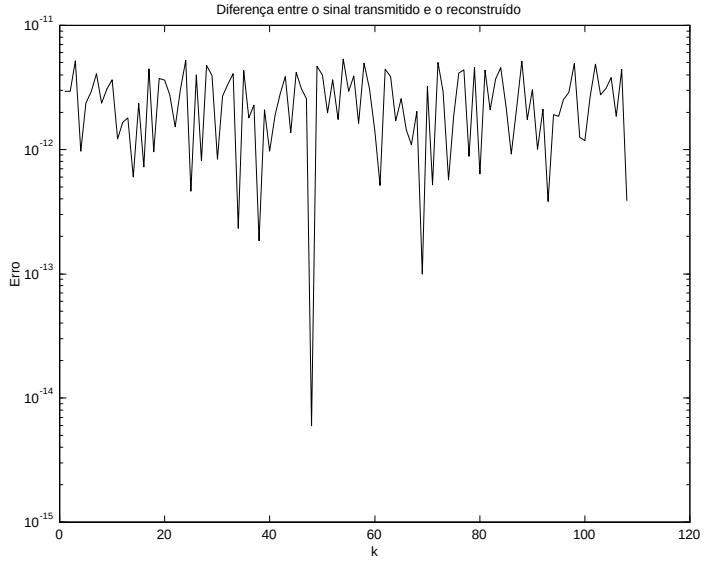


Figura 2.11: Resultado da reconstrução de sinal utilizando o algoritmo de Forney e a ODFT para realizar a codificação e descodificação. Tal como nos métodos anteriores o facto de se forçar as posições dos erros a serem inteiros positivos, permite obter um erro de reconstrução baixo.

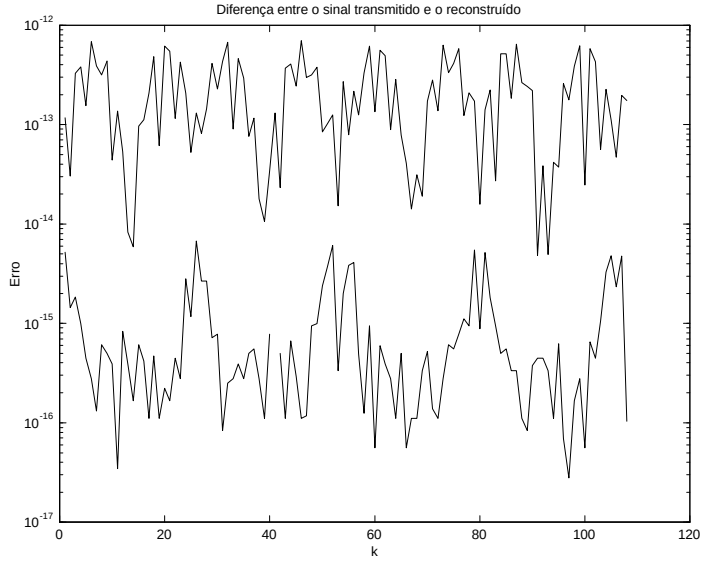


Figura 2.12: Erro na reconstrução de sinal utilizando o algoritmo de eliminação Gaussiana na resolução do sistema de equações (2.17), obtendo-se a amplitude das amostras erradas directamente das amostras do sindroma. A curva superior é o erro de reconstrução obtido com um sistema de equações de dimensão $t \times t$, enquanto que a inferior é o erro obtido com um sistema de equções de dimensão $2t \times t$.

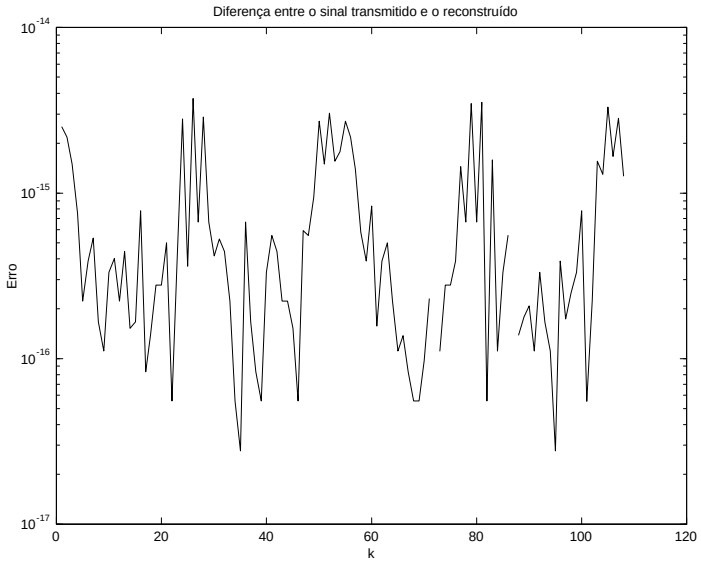


Figura 2.13: Este método de dimensão mínima no tempo para determinação da amplitude dos erros apresenta um erro de reconstrução extremamente baixo, apenas um pouco acima da precisão da máquina utilizada nos cálculos.

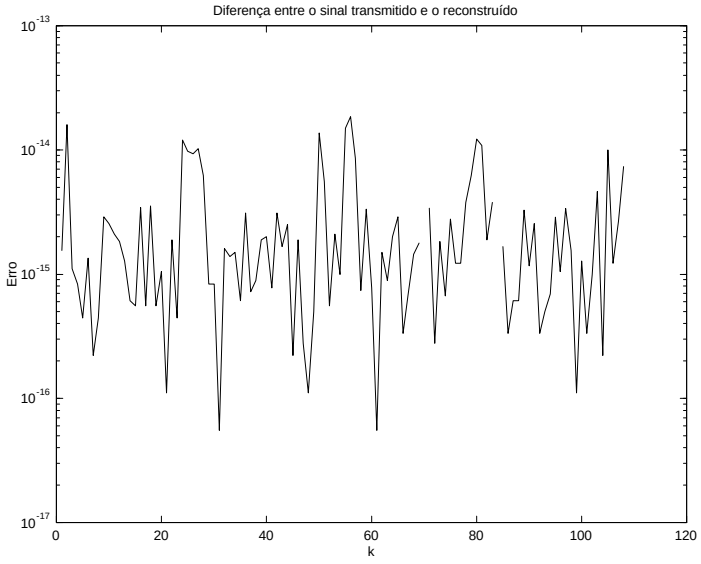


Figura 2.14: Tal como na figura 2.13, o erro de reconstrução do método de dimensão mínima no domínio da frequência é extremamente baixo. Estes métodos permitem melhores resultados na utilização de códigos no corpo dos reais.

Uma breve análise da figura 2.15, revela que quando não existe redundância no sindroma os métodos que pior se comportam são os de dimensão mínima no tempo e na frequência, sendo o melhor o que usa eliminação Gaussiana. Quando temos um sindroma com redundância, os métodos que tiram partido deste facto, conseguem melhores resultados que os apresentados na secção anterior. Na figura 2.16 podemos verificar que o método de eliminação Gaussiana e os dimensão mínima são os que conseguem melhores resultados, sendo o primeiro o que tem um comportamento mehor nas duas situações.

2.4 Algoritmos para a resolução de sistemas Toeplitz

Nesta secção abordaremos o problema da resolução do sistema de equações que permite determinar os coeficientes α. Das quatro formas descritas na secção 2.2 que o sistema de equações pode tomar iremos considerar apenas o caso 3 (2.15) com $r = \frac{N}{2}$, em que a matriz A é Toeplitz e Hermítica.

Este tipo de sistema de equações é dos mais estudados uma vez que aparece numa série muito diversa de problemas como predição linear, estimação espectral, projecto de filtro recursivos, códigos correctores de erros, análise de séries temporais em estatística, etc. Para uma abordagem unificadora recente pode-se ver [Bultheel 97], onde o algoritmo de Euclides aparece como a base unificadora para obter soluções rápidas de sistemas Toeplitz ou Hankel. No entanto a abordagem mais abrangente para os diferentes algoritmos conhecidos para resolver este tipo de sistema: Euclides, Levinson-Durbin, Berlekamp-Massey e Schur, talvez seja a aproximação de Padé [Pad 92, Gragg 72, Zhang 92], pois permite uma compreensão elegante e unificadora do problema.

O algoritmo de Levinson [Levinson 47] foi o primeiro algoritmo rápido para inverter matrizes Toeplitz simétricas e permite resolver sistemas de equações do tipo

$$Ax = b$$

em que a matriz A é Toeplitz e se pode representar por $A_{i,j} = A_{|j-i|}$. Durbin criou igualmente um outra versão do algoritmo [Durbin 60], mas para o caso em que b se encontra relacionado com A por $b_i = -a_{i+1}$, sendo este aliás o nosso caso. Trench [Trench 64] generalizou o algoritmo de Levinson para matrizes não simétricas e Berlekamp [Berlekamp 84] criou um método que resolve de forma indirecta um sistema Toeplitz numa forma semelhante ao algoritmo de Durbin. Mais tarde este algoritmo foi identificado como podendo ser aplicado ao projecto de filtros recursivos de dimensão mínima [Kuijper 97]. A utilização do algoritmo de Euclides para resolver sistemas Toeplitz foi estudada por Brent, Gustavson e Yun [Brent 80]. Kailath, em [Kailath 86] também identificou o algoritmo de Schur [Schur 86a, Schur 86b] como um método de resolver sistemas Toeplitz. Blahut descreve todos estes algoritmos (excepto o de Schur) em [Blahut 85b] tentando unificá-los. Zhang e Duhamel em [Zhang 92] avançaram mais neste sentido alargando o âmbito da unificação.

Uma dedução destes algoritmos será apresentada mais à frente, sendo neste ponto apresentados alguns exemplos numéricos. Começar-se-á pelo algoritmo de Levinson, mostrando a sua relação com os algoritmos de Schur, Berlekamp-Massey e Euclides. Nos resultados das simulações numéricas será utilizado sempre o mesmo exemplo com as condições dadas pelas equações (2.29 e 2.30).

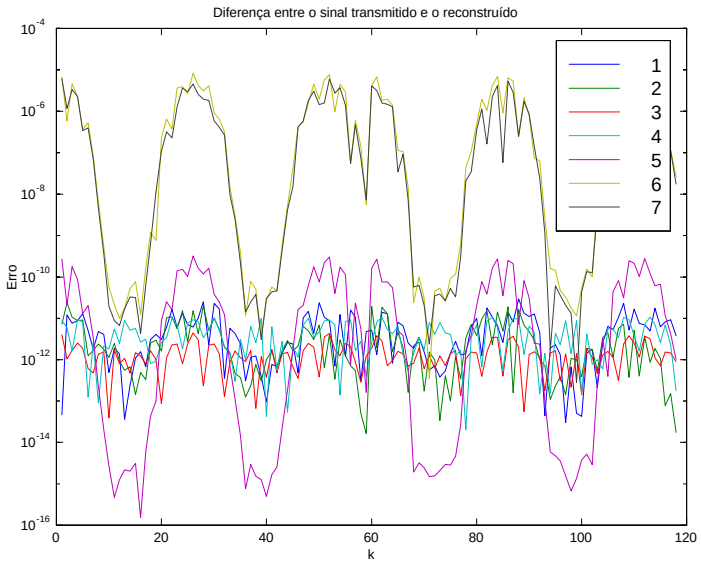


Figura 2.15: Comparação de sete diferentes métodos de reconstrução da amplitude do erro quando são conhecidas as posições dos erros, com $N = 128$, $M = 10$ e $t = 10$. As curvas de 1 a 7 correspondem aos seguintes métodos: 1- Recursão unidireccional, 2- Recursão bidireccional, 3- Zadeh, 4- Forney, 5- Eliminação Gaussiana, 6- Dimesão mínima no tempo e 7- Dimensão mínima na frequência.

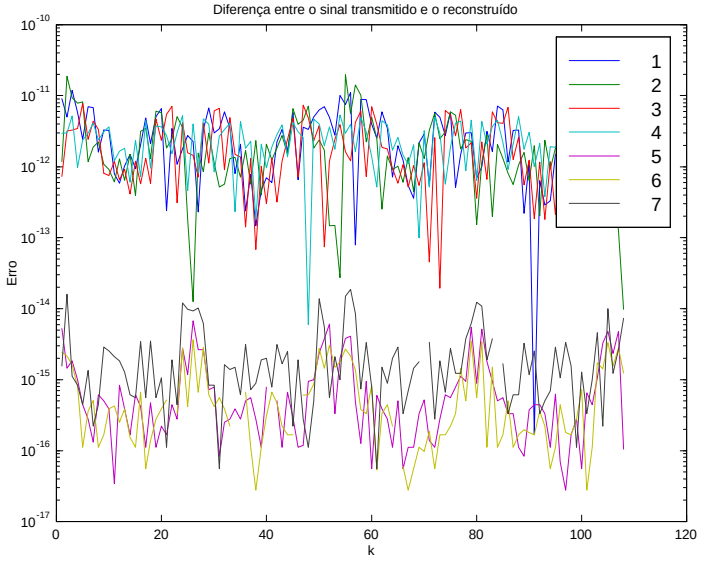


Figura 2.16: Comparação de sete diferentes métodos de reconstrução da amplitude dos erros quando são conhecidas as posições dos erros, com $N = 128$, $M = 20$ e $t = 10$. As curvas de 1 a 7 têm o mesmo significado que na figura 2.15

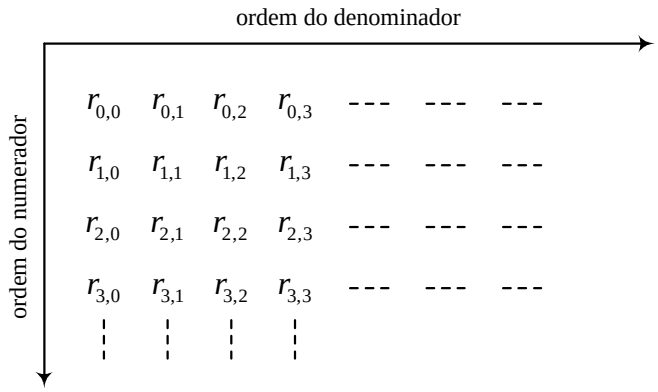


Figura 2.17: Tabela de Padé com as aproximações sucessivas de ordem superior.

2.4.1 Aproximações de Padé

As aproximações de Padé aproximam uma série de potências por uma função racional. Considere-se o polinómio em z:

$$C(z) = c_0 + c_1 z + c_2 z^2 + \ldots$$

uma série formal de potências infinita. Uma função racional

$$\frac{u(z)}{v(z)}$$

é uma aproximação de Padé de ordem (m, n) de $C(z)$ se

$$\begin{aligned} \text{grau}(u(z)) &\leq m & (2.36)\\ \text{grau}(v(z)) &\leq n & (2.37)\\ C(z)\,v(z) - u(z) &= O\left(z^{m+n+1}\right). & (2.38) \end{aligned}$$

Para qualquer série de potências formal, existe uma aproximação de Padé de ordem (m, n) dada pela função racional

$$r_{m,n}(z) = \frac{p_{m,n}(z)}{a_{m,n}(z)}$$

com $p_{m,n}(z)$ e $a_{m,n}(z)$ primos relativos com $p_{m,n}(0) = c_0$ e $a_{m,n}(z) = 1$. Na figura 2.17 podemos ver a tabela de Padé duplamente infinita, em que os termos da primeira coluna $r_{m,0}(z)$, contêm por definição os termos da soma parcial

$$r_{m,0}(z) = \sum_{k=0}^{m} c_k z^k.$$

A expansão em série de Maclaurin de $r_{m,n}(z)$ é exactamente igual a $C(z)$ até pelo menos à potência z^{m+n}.

Em [Zhang 92], é demonstrado que os três algoritmos referidos (Levinson-Durbin/Schur, Euclides e Berlekamp-Massey) são apenas três modos diferentes de percorrer a tabela de Padé para alcançar a solução que não é mais do que o termo (n, n) na tabela de Padé, em que n é a ordem do sistema a resolver. Na figura 2.18 podemos ver o percurso efectuado por cada um dos algoritmos nas suas implementações clássicas.

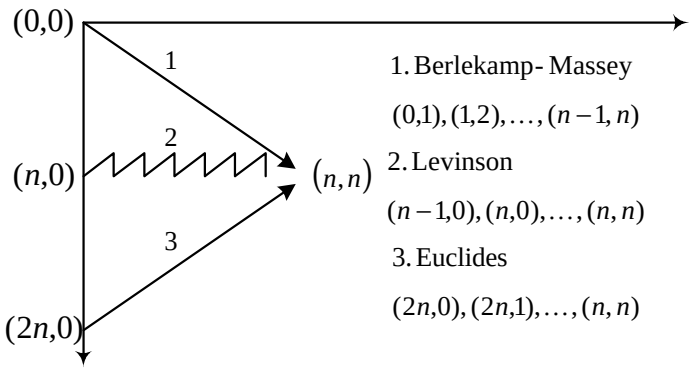


Figura 2.18: Percurso na tabela de Padé, para os algoritmos de Berlekamp-Massey, Levinson e Euclides.

2.4.2 Adaptação de Levinson-Durbin

Este algoritmo para resolver sistemas de equações com matrizes de Toeplitz foi desenvolvido de forma independente por Levinson [Levinson 47] e Durbin [Durbin 60] e descrições da sua implementação podem ser encontradas em [Robinson 80, Golub 83] para o caso simétrico e em [Press 88] para o caso assimétrico. O sistema de equações (2.15) permite aplicar a recursão de Levinson sem se ter que determinar previamente o número de erros ocorridos, calculando a ordem da matriz por um outro método. A ordem do sistema fica automaticamente determinada, quando o algoritmo encontra uma matriz singular. Devido ao erro resultante das operações aritméticas, o teste de singularidade da matriz tem de ser realizado por comparação com um limiar. A determinação deste limiar ainda não foi objecto de estudo e foram utilizados apenas testes empíricos.

A descrição do algoritmo que se segue pode ser encontrada em vários textos [Kailath 86, Marple 87], e aqui ilustra-se apenas o caso em que a matriz Toeplitz é hermítica tendo-se portanto $r = N/2$.

O método começa com um sistema de equações de ordem $n = 1$ e acaba quando encontra um sistema de equações singular com ordem $n = t+1$, em que t é o número de erros ocorridos. Considere-se o seguinte sistema de equações representando a recursão de Levinson de ordem n,

$$\begin{bmatrix} \hat{e}_r & \hat{e}_{r-1} & \cdots & \hat{e}_{r-n+1} \\ \hat{e}^*_{r-1} & \hat{e}_r & \cdots & \hat{e}_{r-n+2} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{e}^*_{r-n+1} & \hat{e}^*_{r-n+2} & \cdots & \hat{e}_r \end{bmatrix} \left[\begin{array}{c|c} 1 & \alpha^*_{n,n} \\ \alpha_{n,1} & \alpha^*_{n,n-1} \\ \vdots & \vdots \\ \alpha_{n,n} & 1 \end{array}\right] = \left[\begin{array}{c|c} \sigma_n^2 & 0 \\ 0 & \vdots \\ \vdots & 0 \\ 0 & \sigma_n^2 \end{array}\right] \tag{2.39}$$

em que as incógnitas α_i foram tornadas dependentes da ordem do sistema de equações passando a designar-se por $\alpha_{n,i}$. A recursão de Levinson permite determinar a solução de ordem $n+1$ conhecendo-se uma solução de ordem inferior n. Suponha-se que se conhece a solução

dada pela equação (2.39), então, pode-se formar a solução de ordem superior com a forma

$$\left[\begin{array}{ccccc} \hat{e}_r & \hat{e}_{r-1} & \cdots & \hat{e}_{r-n+1} & \hat{e}_{r-n} \\ \hat{e}^*_{r-1} & \hat{e}_r & \cdots & \hat{e}_{r-n+2} & \hat{e}_{r-n+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \hat{e}^*_{r-n+1} & \hat{e}^*_{r-n+2} & \cdots & \hat{e}_r & \hat{e}_{r-1} \\ \hat{e}^*_{r-n} & \hat{e}^*_{r-n+1} & \cdots & \hat{e}^*_{r-1} & \hat{e}_r \end{array}\right] \left[\begin{array}{c|c} 1 & 0 \\ \alpha_{n,1} & \alpha^*_{n,n} \\ \vdots & \vdots \\ \alpha_{n,n} & \alpha^*_{n,1} \\ 0 & 1 \end{array}\right] = \left[\begin{array}{c|c} \sigma_n^2 & \Delta_n \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ \Delta^*_n & \sigma_n^2 \end{array}\right], \quad (2.40)$$

em que Δ^*_n é dado pelo producto interno

$$\Delta^*_n = \sum_{i=0}^{n-1} \hat{e}^* (r-n+i)\, \alpha_{n,i}. \quad (2.41)$$

Uma vez que $\alpha^*_{n+1,0} = 1$, para eliminar o Δ^*_n da última equação pode-se realizar uma combinação linear das duas soluções somando à primeira k_{n+1} vezes a segunda de modo a que

$$\Delta^*_n + k_{n+1}\sigma_n^2 = 0,$$

ou

$$k_{n+1} = -\frac{\Delta^*_n}{\sigma_n^2},$$

onde k_n costumam ser designados por coeficientes de reflexão. O valor das incógnitas $\alpha_{n+1,i}$ é dado por

$$\alpha_{n+1,i} = \alpha_{n,i} + k_{p+1} a_{n,n+1-i}, \quad i = 0, 1, \ldots, n+1.$$

Para σ^2_{n+1}, teremos

$$\sigma^2_{n+1} = \sigma_n^2 - k_{n+1}\Delta_n = \left(1 - |k_{n+1}|^2\right)\sigma_n^2, \quad (2.42)$$

ficando completa a recursão que é iniciada para $n = 0$ com os valores

$$a_{0,0} = 1,\ \sigma_0^2 = \hat{e}_r.$$

No caso concreto da resolução do sistema de equações (2.15) não é conhecido à partida quantos erros ocorreram e por esse motivo não se conhece a ordem do sistema a resolver. A iteração do algoritmo de Levinson deve parar quando a matriz é singular, o que corresponde a se obter um coeficiente de reflexão k_{n+1} de módulo unitário, ou como se pode ver por (2.42), um valor nulo para σ^2_{n+1}. No entanto, como estamos a operar no corpo $\mathbb{C}$, é necessário estabelecer um limiar ε em que um número abaixo desse limiar é considerado zero. O problema na determinação desse limiar reside no facto do seu valor variar com a posição dos erros $\bar{S}_t$ com a dimensão do bloco N e com o número máximo de erros t.

2.4.3 Algoritmo de Schur

Tal como no caso do algoritmo de Levinson, o algoritmo de Schur permite resolver um sistema de equações Toeplitz em $O\left(n^2\right)$ operações, com a diferença de que calcula directamente os coeficientes de reflexão k_i sem necessidade de efectuar o producto interno (2.41). Os dois artigos originais de I. Schur, dos quais existe uma tradução para inglês [Schur 86a,Schur 86a],

apesar de não descreverem directamente o algoritmo como permitindo resolver sistemas de equações de Toeplitz, revelaram-se como uma versão paralelizável do algoritmo de Levinson [Kailath 86]. Se se dispuser de n processadores consegue-se resolver o sistema de equações em $O(n)$ operações. A versão do algoritmo de Schur que aqui vamos descrever, aplica-se somente a matrizes Hermíticas. Começaremos por descrever o algoritmo aplicando-o ao nosso sistema de equações e uma vez que se obtêm directamente os coeficientes de reflexão k_i, realizou-se uma implementação da recursão (2.9) na forma de um filtro AR com uma estrutura "lattice" com o objectivo de diminuir o erro de extrapolação do espectro do erro. Contudo, os resultados não foram melhores do que os obtidos directamente com a equação (2.9).

Descrição do algoritmo

Considere-se o sistema Toeplitz (2.40) e a partir da primeira linha vamos formar a matriz geradora G_0 como se pode ver em (2.43). Para obter $\tilde{G}_1$ desloca-se para baixo a primeira coluna de G_0 e de seguida calcula-se o coeficiente k_1 como o quociente entre os elementos da segunda linha obtendo-se

$$k_1 = \frac{\hat{e}_{r-1}}{\hat{e}_r},$$

finalmente multiplica-se a matriz $\tilde{G}_1$ pela matriz

$$\Phi(k_1) = \phi_i \begin{bmatrix} 1 & -k_i \\ -k_i & 1 \end{bmatrix},$$

obtendo-se a matriz G_1. Esta iteração é repetida até se terem determinado todos os t coeficientes de reflexão.

$$G_0 = \begin{bmatrix} \hat{e}_r & 0 \\ \hat{e}_{r-1} & \hat{e}_{r-1} \\ \hat{e}_{r-2} & \hat{e}_{r-2} \\ \vdots & \vdots \\ \hat{e}_{r-t} & \hat{e}_{r-t} \end{bmatrix} \quad \tilde{G}_1 = \begin{bmatrix} 0 & 0 \\ \hat{e}_r & \hat{e}_{r-1} \\ \hat{e}_{r-1} & \hat{e}_{r-2} \\ \vdots & \vdots \\ \hat{e}_{r-t+1} & \hat{e}_{r-t} \end{bmatrix} \quad G_1 = \begin{bmatrix} 0 & 0 \\ \frac{\left(\hat{e}_r^2 - \hat{e}_{r-1}^2\right)}{\hat{e}_r} & 0 \\ ? & ? \\ \vdots & \vdots \\ ? & ? \end{bmatrix}. \tag{2.43}$$

Existem várias possibilidades para o factor de normalização ϕ_i, pode ser apenas a unidade como no caso do exemplo dado, ou por exemplo como vem referido em [Kailath 86]

$$\phi_i = \frac{1}{\sqrt{1 - k_i^2}}.$$

Se se pretender utilizar a recursão (2.14) para extrapolar o espectro do erro é possível obter os coeficientes α_i do polinómio a partir dos coeficientes de reflexão k_i através da recursão

$$\begin{bmatrix} \alpha_m(z) \\ \alpha_m^*(z) \end{bmatrix} = \Phi(k_m) \begin{bmatrix} 1 & 0 \\ 0 & z^{-1} \end{bmatrix} \cdots \Phi(k_1) \begin{bmatrix} 1 & 0 \\ 0 & z^{-1} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \tag{2.44}$$

em que os coeficientes α_m^* são iguais aos coeficientes h_i no caso da matriz ser Hermítica.

Resultados da extrapolação de $\hat{e}_r$

Para averiguar da eficiência e da robustez da extrapolação do espectro do sinal de erro, realizaram-se algumas experiências (e uma vez que o algoritmo calcula directamente os coeficientes de reflexão), implementou-se o filtro recursivo (2.14) com uma estrutura *"lattice"* só com zeros. A estrutura é inicializada com a saída de um filtro inverso em que se coloca na sua entrada os valores conhecidos do espectro e a saída serve para inicializar os atrasos do filtro "lattice". Os resultados do erro de reconstrução desta estrutura são comparados com os realizados com a equação recursiva (2.14). Apesar de se esperarem resultados mais correctos e uma menor sensibilidade à propagação dos erros numéricos, a implementação prática veio provar que tal não acontece, obtendo-se resultados piores ou semelhantes.

2.4.4 Algoritmo de Berlekamp-Massey

Este algoritmo é bastante conhecido e aplicado na área da TCCE [Blahut 83, Berlekamp 84] para resolver o sistema de equações (2.15) e determinar o espectro do erro a partir do sindroma. Embora este algoritmo tenha sido criado para ser utilizado com aritmética finita ele é válido igualmente no corpo dos números complexos. Considere-se que se conhecem os parâmetros α_i da equação (2.14), então, o sistema de equações (2.15) que a seguir se repete

$$\begin{bmatrix} \hat{e}_r & \hat{e}_{r-1} & \cdots & \hat{e}_{r-t+1} \\ \hat{e}_{r+1} & \hat{e}_r & \cdots & \hat{e}_{r-t+2} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{e}_{r+t-1} & \hat{e}_{r+t-2} & \cdots & \hat{e}_r \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_t \end{bmatrix} = - \begin{bmatrix} \hat{e}_{r+1} \\ \hat{e}_{r+2} \\ \vdots \\ \hat{e}_{r+t} \end{bmatrix},$$

mostra que a partir da primeira linha se pode definir $\hat{e}_{r+1}$ em função dos termos $\hat{e}_r \ldots \hat{e}_{r-t+1}$, e que a segunda linha define $\hat{e}_{r+2}$ em função de $\hat{e}_{r+1} \ldots \hat{e}_{r-t+2}$, e assim sucessivamente. Este recorrência não é mais do que a equação (2.14), que por conveniência se repete aqui

$$\hat{e}_k = -\sum_{i=1}^{t} \alpha_i \hat{e}_{k-i}, \qquad k = r+1, r+2, \ldots, r+t.$$

Para um dado $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \ldots, \alpha_t\}$ de dimensão fixa, a equação em cima pode ser vista como um filtro autoregressivo implementável como um registo de deslocamento com realimentação tal como a figura 2.19 mostra. Visto desta forma o método de Berlekamp consiste em desenhar o filtro de dimensão mínima, - ou seja, que utiliza um número mínimo de coeficientes α - que consegue gerar o sindroma recebido.

Qualquer método para desenhar filtros autoregressivos pode assim ser utilizado para resolver o sistema de equações.

2.4.5 Algoritmo de Euclides

O algoritmo de Euclides aplicado a polinómios permite determinar o maior divisor comum (mdc) entre dois polinómios e ainda exprimir o mdc como uma combinação linear desses dois polinómios. Na sua forma original, Euclides descreveu este algoritmo no sétimo volume da sua obra *Elementos* (330 - 275 ac) como um método para determinar a maior régua que conseguia medir o comprimento de outras duas réguas um número inteiro de vezes a medida da primeira. Do ponto de vista da aproximação, o tipo de aproximantes racionais que se obtêm são conhecidos por aproximantes de Padé. Apesar da teoria dos aproximantes de

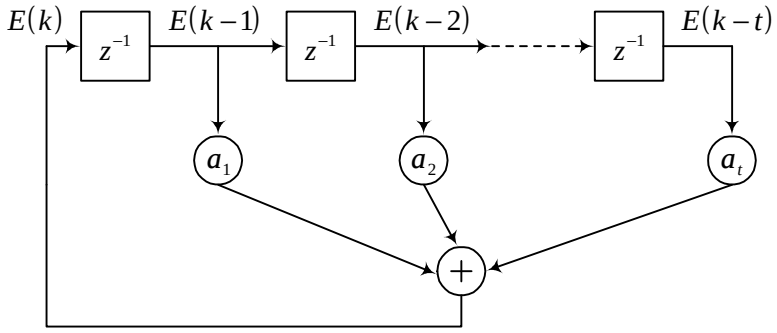


Figura 2.19: O algoritmo de Berlekamp-Massey consiste em projectar um filtro recursivo de dimensão mínima que consiga gerar recursivamente o espectro do erro a partir do sindroma.

Padé ter quase um século só há cerca de vinte anos se reconheceu o algoritmo de Euclides como um dos métodos para os obter [Bultheel 97].

Assim se forem dados os polinómios $s(z)$ e $u(z)$, aplicando o algoritmo de Euclides, podemos escrever

$$\mathrm{mdc}(s,u) = \alpha(z)\, s(z) + \beta(z)\, u(z),$$

em que $\alpha(z)$ e $\beta(z)$, são dois polinómios obtidos a partir do algoritmo. Existem várias referências recentes onde este algoritmo pode ser encontrado como por exemplo [Blahut 85b, Berlekamp 84, Krishna 94]. A descrição aqui utilizada foi inspirada no artigo de Zhang e Duhamel [Zhang 92] em que se faz um estudo comparativo de vários métodos para resolver sistemas Toeplitz.

Para aplicar o algoritmo de Euclides à resolução do sistema de equações Toeplitz (2.15), tem de se colocar primeiro o sistema na seguinte forma equivalente:

$$\begin{bmatrix} \hat{e}_r & \hat{e}_{r-1} & \cdots & \hat{e}_{r-t} \\ \hat{e}_{r+1} & \hat{e}_r & \cdots & \hat{e}_{r-t+1} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{e}_{r+t} & \hat{e}_{r+t-1} & \cdots & \hat{e}_r \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_t \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \tag{2.45}$$

com $\alpha_0 = 1$, e que fica com uma forma mais conveniente se se fizer a mudança de variável $c_i = \hat{e}_{r-t+i}$.

$$\begin{bmatrix} c_t & c_{t-1} & \cdots & c_0 \\ c_{t+1} & c_t & \cdots & c_1 \\ \vdots & \vdots & \ddots & \vdots \\ c_{2t} & c_{2t-1} & \cdots & c_t \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_t \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{2.46}$$

Para aplicar o algoritmo de Euclides à resolução de (2.46), é necessário colocar este sistema

numa forma polinomial realizando a sua expansão como se segue [Zhang 92]:

$$\begin{bmatrix} c_0 & 0 & \cdots & 0 \\ c_1 & c_0 & & \vdots \\ \vdots & & \ddots & 0 \\ c_t & c_{t-1} & \cdots & c_0 \\ c_{t+1} & c_t & & c_1 \\ \vdots & & \cdots & \vdots \\ c_{2t} & c_{2t-1} & \cdots & c_t \\ 0 & c_{2t} & & c_{t+1} \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & c_{2t} \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_t \end{bmatrix} = \begin{bmatrix} p_0 \\ \vdots \\ p_{t-1} \\ 0 \\ \vdots \\ 0 \\ q_0 \\ q_1 \\ \vdots \\ q_t \end{bmatrix}$$

com $q_0 = 0$. Definindo

$$\begin{aligned} c(z) &= c_0 + c_1 z + \ldots + c_{2t} z^{2t}, \\ C(z) &= c(z) + O\left(z^{2t+1}\right), \\ \alpha(z) &= \alpha_0 + \alpha_1 z + \ldots \alpha_t z^t, \\ p(z) &= p_0 + p_1 z + \ldots + p_{t-1} z^t, \\ q(z) &= q_0 + q_1 z + \ldots + q_t z^t. \end{aligned}$$

Podemos finalmente escrever a seguinte equação polinomial

$$C(z)\,\alpha(z) = p(z) + z^{2t} q(z)$$

ou

$$-z^{2t} q(z) + C(z)\,\alpha(z) = p(z),$$

se aplicarmos o algoritmo de Euclides com $s(z) = c(z)$ e $u(z) = -z^{2t}$, o $\mathrm{mdc}\left(c, -z^{2t}\right)$ é encontrado quando $p(z) = 0$. No entanto o que pretendemos obter é a solução para a equação (2.46) e assim, se $c(z)$ for normal o algoritmo deve ser realizado apenas t vezes. Se $c(z)$ não for normal, então a recursão é interrompida antes e realizada apenas $g \leq t$ vezes em que g é igual ao número de erros ocorridos. Um polinómio diz-se normal se todos os seus aproximantes de Padé forem diferentes.

Resultados numéricos

Os algoritmos descritos anteriormente resolvem todos o mesmo problema de forma semelhante e com uma complexidade algorítmica de $O\left(n^2\right)$. Em [Zhang 92] classificam-se os algorimos como sendo de uma passagem ou de duas passagens. O algoritmo de Schur apresentado por exemplo é de uma passagem se apenas se pretender os coeficientes de reflexão k_i, e é fácil compreender que a segunda passagem acontece quando se aplica a recursão (2.44) para calcular os coeficientes do polinómio. Por outro lado o método de Levinson, na versão apresentada anteriormente é uma versão de duas passagens. É possível no entanto formalizar versões de uma e de duas passagens para todos os algoritmos apresentados [Zhang 92].

Quanto à robustez numérica dos quatro algoritmos para resolver o sistema de equações Toeplitz, os testes práticos não permitiram retirar qualquer conclusão sobre qual o melhor

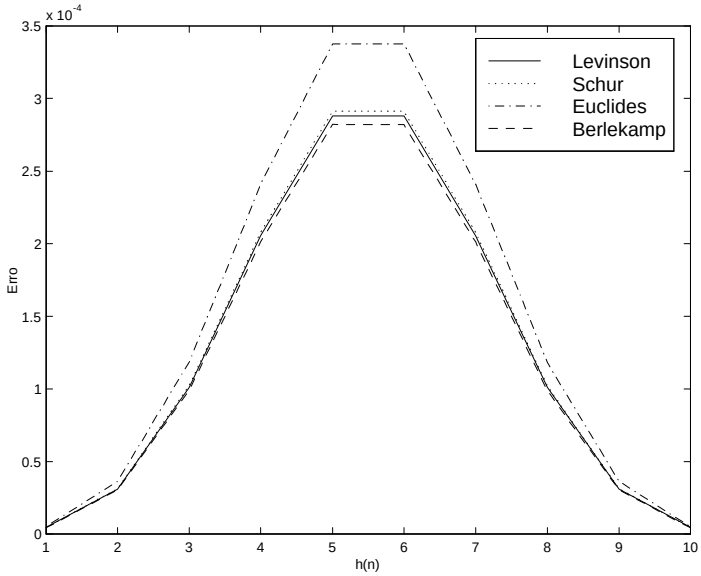


Figura 2.20: Comparação do erro obtido na determinação dos coeficientes α_i para os vários métodos descritos neste capítulo, Levinson, Schur, Euclides e Berlekamp. Considera-se que o sinal mensagem m é nulo.

método. Consoante se variavam os parâmetros do problema ou o sinal de entrada, os resultados para cada algoritmo variavam bastante. Como mera ilustração pode-se ver na figura 2.20 o erro cometido na determinação dos coeficientes α_i, para o caso dado em (2.29 e 2.30) e com sinal de entrada nulo. Como se pode observar, o erro é relativamente elevado (aproximadamente 10^{-4}) para um caso aparentemente fácil de resolver.

Capítulo 3

Estabilidade do problema de reconstrução

Devido aos erros de arredondamento ocorridos durante os cálculos efectuados na codificação e na reconstrução de sinal, qualquer algoritmo de reconstrução pode tornar-se instável, dando origem a resultados errados. Como veremos, existem vários factores que afectam a estabilidade dos algoritmos de reconstrução:

- dimensão N do vector
- posição relativa dos erros
- amplitude dos erros
- número de erros.

O estudo da estabilidade dos algoritmos de reconstrução no corpo dos números complexos $\mathbb{C}$ é essencial para se poder avaliar a fiabilidade dos resultados. É claro que, no caso dos códigos correctores de erros em que se utiliza aritmética num corpo finito (por exemplo o corpo de Galois GF(p^n) [Blahut 83, Berlekamp 84]), não ocorrem erros de arredondamento durante os cálculos aritméticos, e por consequência, o problema da estabilidade não se coloca. Esta característica dos corpos finitos advém do facto do conjunto dos seus números ser finito e assim poderem ser representados exactamente num computador.

O conhecimento dos factores que influenciam a estabilidade dos algoritmos de reconstrução no corpo $\mathbb{C}$ permite determinar os limites dos mesmos e realizar o seu projecto prático.

Para o caso conhecido como correcção de apagamentos[1], em que se conhece a posição dos erros e se pretende determinar apenas a sua amplitude, a sua caracterização já foi feita, existindo alguns resultados importantes nos artigos [Ferreira 94b, Ferreira 94a, Ferreira 94c, Ferreira 95, Ferreira 97] cujos resultados caracterizam a estabilidade do problema, consoante a posição dos erros, o número de erros e a dimensão do bloco. Neste capítulo serão apresentados alguns destes resultados com o objectivo de estabelecer uma comparação com o problema da determinação da posição dos erros.

Para o caso em que se desconhece a posição dos erros, é necessário resolver o sistema de equações (2.15), sendo a análise da estabilidade deste problema mais complexa que a anterior e por isso menos estudada. Neste capítulo serão apresentados alguns resultados analíticos

[1] Traduzido do termo Inglês "Erasure Correction".

para $t = 1$ e $t = 2$, e formulados alguns teoremas inspirados no tratamento feito em [Ferreira 97]. Serão igualmente apresentados alguns testes experimentais, com o objectivo de se obterem resultados empíricos para o dimensionamento deste tipo de algoritmos de reconstrução. Nomeadamente apresentaremos as sequências de erros correspondentes ao maior e menor valor próprio da matriz A, a variação do condicionamento da matriz A em função de $e_{\min}$, N, t e da distância mínima entre erros. Será também demonstrado que o condicionamento deste problema piora quando a dimensão do bloco aumenta.

Numa realização prática, os algoritmos de reconstrução considerados neste trabalho operam sempre sobre sinais armazenados ou transmitidos. Este tipo de sinais são codificados recorrendo a um número finito de bits por amostra e por esse motivo possuem uma amplitude máxima e mínima. Deste modo, o sinal de erro e satisfaz a relação

$$e_{\min} \leq e_k \leq e_{\max}.$$

3.1 A importância do condicionamento na resolução de sistemas de equações

Os dois problemas de reconstrução conseguem-se reduzir à resolução de um sistema de equações do tipo

$$Ax = b. \tag{3.1}$$

Como já foi referido, a ocorrência de pequenos erros nos cálculos pode afectar seriamente a precisão da solução encontrada. Suponhamos que os sistema de equações sofreram pequenas alterações e que vamos resolver o seguinte sistema

$$(A + E)\,\tilde{x} = (b + e)\,, \tag{3.2}$$

em que a matriz E é suficientemente pequena para que $(A + E)$ seja invertível. Neste caso, será possível encontrar um majorante para o erro $x - \tilde{x}$?

Este erro pode ser escrito utilizando as equações (3.2) e (3.1),

$$x - \tilde{x} = A^{-1}b - (A + E)^{-1}(b + e) = \left[A^{-1} - (A + E)^{-1}\right] b - (A + E)^{-1} e. \tag{3.3}$$

Por outro lado o termo $(A + E)^{-1}$ pode se reescrito como $\left(I + A^{-1}E\right)^{-1} A^{-1}$ e se $\rho\left(A^{-1}E\right) < 1$, então, pode-se escrever $\left(I + A^{-1}E\right)$ como uma série de potências na base $\left(A^{-1}E\right)$

$$(A + E)^{-1} = \sum_{k=0}^{\infty} (-1)^k \left(A^{-1}E\right)^k A^{-1}.$$

Utilizando esta expansão para reescrever a equação (3.3), vem

$$x - \tilde{x} = \left[\sum_{k=1}^{\infty} (-1)^k \left(A^{-1}E\right)^k A^{-1}\right] b - \left[\sum_{k=0}^{\infty} (-1)^k \left(A^{-1}E\right)^k A^{-1}\right] e.$$

Se $\|\cdot\|$ for uma norma sobre matrizes e vectores, e desde que $\left\|A^{-1}E\right\| < 1$, então

$$\begin{aligned} \left\|A^{-1} - (A + E)^{-1}\right\| &= \left\|\sum_{k=1}^{\infty} (-1)^{k+1} \left(A^{-1}E\right)^k A^{-1}\right\| \leq \\ &\leq \sum_{k=1}^{\infty} \left\|A^{-1}E\right\|^k \left\|A^{-1}\right\| = \frac{\left\|A^{-1}E\right\|}{1 - \left\|A^{-1}E\right\|} \left\|A^{-1}\right\|, \end{aligned}$$

e

$$\begin{aligned}\left\|(A+E)^{-1}\right\| &= \left\|\sum_{k=0}^{\infty}(-1)^k\left(A^{-1}E\right)^k A^{-1}\right\| \\ &\leq \sum_{k=0}^{\infty}\left\|A^{-1}E\right\|^k\left\|A^{-1}\right\| = \frac{1}{1-\left\|A^{-1}E\right\|}\left\|A^{-1}\right\|.\end{aligned}$$

A norma do erro absoluto será então limitada por

$$\|x-\tilde{x}\| \leq \frac{\left\|A^{-1}E\right\|}{1-\left\|A^{-1}E\right\|}\left\|A^{-1}\right\|\|b\| + \frac{1}{1-\left\|A^{-1}E\right\|}\left\|A^{-1}\right\|\|e\|, \tag{3.4}$$

definindo o condicionamento de A como

$$\kappa(A) = \left\|A^{-1}\right\|\|A\|,$$

substituindo $b = Ax$, e dividindo (3.4) por $\|x\|$, temos a seguinte expressão para o erro relativo

$$\frac{\|x-\tilde{x}\|}{\|x\|} \leq \frac{\kappa(A)}{1-\kappa(A)\left(\|E\|/\|A\|\right)}\frac{\|E\|}{\|A\|} + \frac{\kappa(A)}{1-\kappa(A)\left(\|E\|/\|A\|\right)}\frac{\|e\|}{\|b\|}. \tag{3.5}$$

Se o condicionamento de A for pequeno, o que equivale a ter $\kappa(A) \simeq 1$, então, como $\|E\| \ll \|A\|$,o erro relativo na solução do sistema é da mesma ordem de grandeza que os erros relativos $\|E\|/\|A\|$ e $\|e\|/\|b\|$. Se por outro lado, o condicionamento for muito grande então, os erros relativos virão aumentados pelos factores dados na equação (3.5).

A demonstração aqui descrita pode ser encontrada em [Stewart 73, Horn 85].

3.2 Estudo da estabilidade na reconstrução de apagamentos

No capítulo 1 abordou-se a reconstrução de sinal quando se conhecem as posições das amostras erradas. Nesta secção iremos apresentar alguns resultados já publicados sobre a estabilidade deste problema remetendo a sua demonstração para as referências [Ferreira 94b, Ferreira 97, Ferreira 94a].

Considere-se um vector de dimensão N pertencente a $\mathbb{C}^N$, e o conjunto dos índices das amostras erradas $\bar{S}_t$. O problema de reconstrução consiste em determinar a amplitude das amostras erradas $x_i\left(i \in \bar{S}_t\right)$, e foi demonstrado na secção 1.2.3 que o problema pode ser reduzido à resolução do sistema de equações

$$u = Su + h \Leftrightarrow u = (I-S)^{-1}h \tag{3.6}$$

onde u é o vector de dimensão t das amostras desconhecidas. Vamos agora considerar o caso em que o sinal x é passa-baixo sendo os índices das amostras conhecidas de $\hat{x}$ dados por S_f (1.4). O estudo dos valores próprios da matriz $(I-S)$, com S definida por

$$\begin{aligned}S_{ab} &= \frac{1}{N}\sum_{k=-m}^{m} e^{\mathrm{j}\frac{2\pi}{N}k(i_a-i_b)}, \qquad i_a, i_b \in \bar{S}_t \wedge a \neq b \\ S_{ab} &= K/N = (2m+1)/N, \qquad a = b,\end{aligned} \tag{3.7}$$

permite avaliar a estabilidade numérica do problema de reconstrução.

Pelo teorema de Rayleigh-Ritz [Horn 85], o estudo dos valores próprios de uma matriz pode ser abordado como um problema de optimização da forma quadrática associada à matriz na superfície de uma esfera unitária. A forma quadrática associada a S é dada por

$$\begin{aligned} x^{\dagger}Sx &= \sum_{a=0}^{t-1}\sum_{b=0}^{t-1} x_a^* x_b \left[\frac{1}{N}\sum_{k=-m}^{m} e^{\mathrm{j}\frac{2\pi}{N}(i_a-i_b)k}\right] \\ &= \sum_{k=-m}^{m}\left|\frac{1}{\sqrt{N}}\sum_{a=0}^{t-1} x_a e^{-\mathrm{j}\frac{2\pi}{N}i_a k}\right|^2 . \end{aligned} \tag{3.8}$$

Fazendo

$$\phi(k) = \frac{1}{\sqrt{N}}\sum_{a=0}^{t-1} x_a e^{-\mathrm{j}\frac{2\pi}{N}i_a k},$$

podemos escrever (3.8) na forma mais compacta

$$x^{\dagger}Sx = \sum_{k=-m}^{m} |\phi(k)|^2 . \tag{3.9}$$

A matriz S é Hermítica e positiva definida, pois a partir da equação (3.9) pode-se verificar que todos os valores próprios são positivos. Calculando o somatório na equação (3.7), podemos escrever

$$S_{ab} = e^{j\frac{2\pi}{N}(i_a-i_b)(2m+1)}\frac{\sin\left[\pi\left(2m+1\right)\left(i_a-i_b\right)/N\right]}{N\sin\left(\pi\left(i_a-i_b\right)/N\right)} \qquad 0 \le a,b \le t-1,$$

que é uma sub-matriz de B com elementos dados por

$$B_{ab} = \frac{\sin\left[\pi\left(2m+1\right)\left(a-b\right)/N\right]}{N\sin\left(\pi\left(a-b\right)/N\right)} \qquad 0 \le a,b \le N-1,$$

que tem somente valores próprios $\lambda = 0$ e $\lambda = 1$. Das propriedades de entrelaçamento dos valores próprios de submatrizes de matrizes Hermíticas, referidas em [Horn 85], a matriz S tem os seus valores próprios neste intervalo.

A estabilidade da resolução do sistema de equações (3.6), depende do condicionamento da matriz $(I-S)$. Vamos agora enunciar sem demonstrar alguns teoremas que caracterizam os valores próprios de $(I-S)$. O teorema 1 indica limites para $\lambda_{\min}$ e $\lambda_{\max}$ em função do factor de sobreamostragem β. O teorema 2 indica limites para $\lambda_{\min}$ e $\lambda_{\max}$ mais rigorosos do que o teorema 1 na condição de as posições dos erros serem múltiplos de um número inteiro. O teorema 3 indica limites para os valores próprios de S quando se aumenta a ordem (número de erros) do problema a resolver.

Teorema 1 *O valor próprio mínimo* $\lambda_{\min}$ *de* $(I-S)$ *pertence ao intervalo* $[0\ldots 1-\beta]$*, e o valor próprio máximo* $\lambda_{\max}$ *pertence ao intervalo* $[1-\beta\ldots 1]$*, com* $\beta = (2m+1)/N$.

Teorema 2 *Se* $\bar{S}_t = \{i_0 k, i_1 k, \ldots i_{t-1}k\}$*, forem os índices das amostras erradas, e* N/k *for um inteiro, o menor e o maior valor próprio da matriz* $(I-S)$ *satisfazem a relação*

$$1 - \frac{\lceil k\beta \rceil}{k} \le \lambda_{\min} \le \lambda_{\max} \le 1 - \frac{\lfloor k\beta \rfloor}{k}$$

com $\beta = (2m+1)/N$.

Teorema 3 *Considere-se* $\bar{S}_t^{(t)} = \{i_0, i_1, \ldots i_{t-1}\}$ *e* $\bar{S}_t^{(t-1)} = \{i_0, i_1, \ldots i_{t-2}\}$ *dois conjuntos de índices de amostras erradas e* $S^{(t)}$ *e* $S^{(t-1)}$*, as correspondentes matrizes de interpolação com dimensão* $(t \times t)$ *e* $(t-1) \times (t-1)$ *respectivamente. Então*

$$\begin{aligned} 1 - \lambda'\left(I^{(t-1)} - S^{(t-1)}\right) - \|v\| &\leq \lambda\left(I^{(t)} - S^{(t)}\right), \\ \lambda\left(I^{(t)} - S^{(t)}\right) &\leq 1 - \lambda'\left(I^{(t-1)} - S^{(t-1)}\right) + \|v\|, \end{aligned}$$

onde o vector v *é definido por*

$$v_k = \frac{\sin\left(\pi\left(2m+1\right)\left(i_k - i_{t-1}\right)/N\right)}{N\sin\left(\pi\left(i_k - i_{t-1}\right)/N\right)}$$

para $0 \leq k \leq t-2$*.* $\lambda'\left(I^{(t-1)} - S^{(t-1)}\right)$ *representa a sequência dos valores próprios de* $\left(I^{(t-1)} - S^{(t-1)}\right)$ *ordenados por ordem crescente, e* $\lambda\left(I^{(t)} - S^{(t)}\right)$ *é a sequência dos valores próprios de* $\left(T^{(t)} - S^{(t)}\right)$ *ordenados da mesma forma.*

Corolário 1 *Os limites do teorema 3 podem ser substituidos pelos limites mais fracos*

$$\begin{aligned} 1 - \lambda'\left(I^{(t-1)} - S^{(t-1)}\right) - \sqrt{\beta\left(1-\beta\right)} &\leq \lambda\left(I^{(t)} - S^{(t)}\right), \\ \lambda\left(I^{(t)} - S^{(t)}\right) &\leq 1 - \lambda'\left(I^{(t-1)} - S^{(t-1)}\right) + \sqrt{\beta\left(1-\beta\right)}. \end{aligned}$$

Para o caso de apenas ocorrerem dois erros é possível obter uma expressão analítica para os valores próprios da matriz $(I - S)$, resultando para o condicionamento da matriz a expressão

$$\kappa\left(I - S\right) = \frac{2\left(\beta - 1\right)}{\beta - 1 + \frac{1}{N}\left|\frac{\sin((2m+1)(i_0 - i_1)\pi/N)}{\sin((i_0 - i_1)\pi/N)}\right|} - 1. \tag{3.10}$$

Na figura 3.1, podemos observar a evolução do condicionamento da matriz $(I - S)$ em função da distância entre erros e para vários valores de N. Para cada valor de N, o factor de interpolação β foi mantido aproximadamente constante. Como se pode constatar, o condicionamento da matriz $(I - S)$ é independente da dimensão N do bloco de dados.

3.3 Estudo da estabilidade na reconstrução da posição dos erros

Tal como indicado na figura 2.1, e para ter em conta o ruído devido à quantificação, introduziu-se o sinal η. A presença deste ruído complica bastante a realização prática dos algoritmos levando a que não seja possível atingir a sua capacidade máxima. Vamos estudar o condicionamento do sistema $A\alpha = b$, na forma do caso 3 (2.15) em que a matriz A é dada por:

$$A = \begin{bmatrix} \hat{e}_r & \hat{e}_{r-1} & \cdots & \hat{e}_{r-t+1} \\ \hat{e}_{r+1} & \hat{e}_r & \cdots & \hat{e}_{r-t+2} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{e}_{r+t-1} & \hat{e}_{r+t-2} & \cdots & \hat{e}_r \end{bmatrix}, \tag{3.11}$$

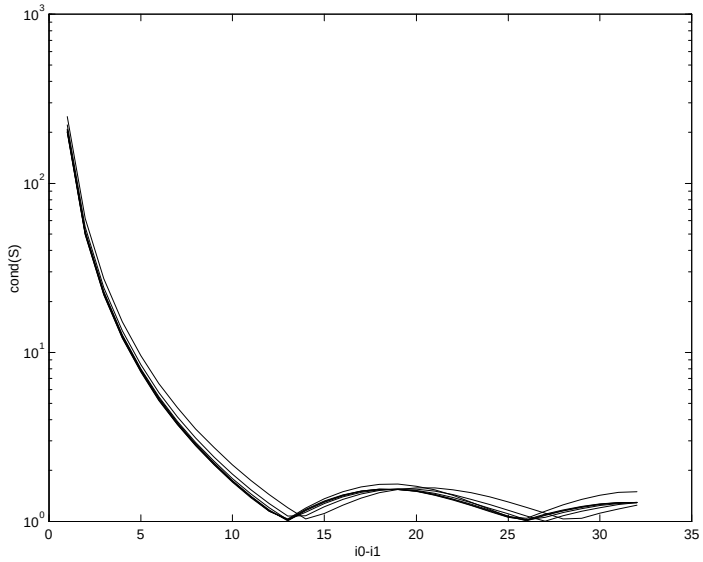


Figura 3.1: Variação do condicionamento da matriz $I - S$ em função da distância entre erros $i_0 - i_1$, mantendo o factor de interpolação β constante. Como se pode observar, o condicionamento não varia quando se mantem β aproximadamente igual a 0.92.

e que pode ser escrita na forma mais compacta

$$A_{pq} = \hat{e}\,(r + p - q)\,, \qquad p, q = 0, 1, \ldots, t-1, \tag{3.12}$$

em que p e q são os índices das linhas e colunas de A respectivamente. Considere-se a forma quadrática associada a A com $x \in \mathbb{C}^t$

$$x^\dagger A x = \sum_{p=0}^{t-1} \sum_{q=0}^{t-1} A_{pq} x_p^\dagger x_q,$$

substituindo nesta equação (3.12), e uma vez que segundo a definição de transformada de Fourier utilizada (1.6) $\hat{e}_k$ é dado por

$$\hat{e}_k = \frac{1}{\sqrt{N}} \sum_{m=0}^{t-1} e(i_m) e^{-\mathrm{j}\frac{2\pi}{N} k i_m},$$

então,

$$x^\dagger A x = \sum_{p=0}^{t-1} \sum_{q=0}^{t-1} \frac{1}{\sqrt{N}} \sum_{m=0}^{t-1} e\,(i_m)\, e^{-\mathrm{j}\frac{2\pi}{N}(r+p-q) i_m} x_p^* x_q.$$

Trocando a ordem dos somatórios obtem-se

$$x^\dagger A x = \frac{1}{\sqrt{N}} \sum_{m=0}^{t-1} e\,(i_m) \left[\sum_{p=0}^{t-1} x_p^* e^{-\mathrm{j}\frac{2\pi}{N} i_m (r+p)} \sum_{q=0}^{t-1} x_q e^{-\mathrm{j}\frac{2\pi}{N} i_m (r-q)} \right],$$

ou

$$x^{\dagger}Ax = \frac{1}{\sqrt{N}} \sum_{m=0}^{t-1} e\left(i_m\right) e^{-\mathrm{j}\frac{2\pi}{N} r i_m} \left[\left(\sum_{p=0}^{t-1} x_p e^{\mathrm{j}\frac{2\pi}{N} i_m p} \right)^{*} \sum_{q=0}^{t-1} x_q e^{\mathrm{j}\frac{2\pi}{N} i_m q} \right],$$

a qual para $r = N/2$ dá

$$x^{\dagger}Ax = \frac{1}{\sqrt{N}} \sum_{m=0}^{t-1} e\left(i_m\right) \left(-1\right)^{i_m} \left|\phi\left(i_m\right)\right|^2, \tag{3.13}$$

em que

$$\phi\left(i_m\right) = \sum_{p=0}^{t-1} x\left(p\right) e^{j\frac{2\pi}{N} i_m p}.$$

Como vimos anteriormente, se A for hermítica, o cálculo dos seus valores próprios pode ser transformado num problema de optimização da relação de Rayleigh-Ritz [Horn 85]

$$\frac{x^{\dagger}Ax}{x^{\dagger}x}.$$

Mais concretamente temos que

$$\lambda_{\max} = \max_{x \neq 0} \frac{x^{\dagger}Ax}{x^{\dagger}x} \tag{3.14}$$

$$\lambda_{\min} = \min_{x \neq 0} \frac{x^{\dagger}Ax}{x^{\dagger}x}, \tag{3.15}$$

em que $\lambda_{\max}$ e $\lambda_{\min}$, são respectivamente o valor próprio máximo e mínimo de A.

3.3.1 Limites para os valores próprios da matriz A

É possível ter uma ideia aproximada dos limites para os valores próprios de A, e o seguinte teorema é uma primeira aproximação.

Teorema 4 *Os valores próprios máximo e mínimo de A satisfazem a desigualdade*

$$\boxed{\lambda_{\min} \leq \frac{1}{\sqrt{N}} \sum_{m=0}^{t-1} e\left(i_m\right) \left(-1\right)^{i_m} \leq \lambda_{\max}.} \tag{3.16}$$

Demonstração. O teorema 4.3.26 de [Horn 85], diz que a soma dos elementos da diagonal principal, de qualquer menor principal, de uma matriz hermítica majoriza a soma dos seus valores próprios. No caso particular da diagonal principal de A temos

$$\sum_{p=0}^{t-1} A_{pp} = \sum_{i=0}^{t-1} \lambda_i. \tag{3.17}$$

Dado que

$$t\lambda_{\min} \leq \sum_{i=0}^{t-1} \lambda_i \leq t\lambda_{\max} \tag{3.18}$$

e atendendo ao facto de

$$A_{pp} = \frac{1}{\sqrt{N}} \sum_{m=0}^{t-1} e\left(i_m\right) (-1)^{i_m}$$

temos de (3.17) que

$$\sum_{i=0}^{t-1} \lambda_i = tA_{pp} = \frac{t}{\sqrt{N}} \sum_{m=0}^{t-1} e\left(i_m\right) (-1)^{i_m},$$

e a desigualdade (3.18) vem

$$\lambda_{\min} \leq \frac{1}{\sqrt{N}} \sum_{m=0}^{t-1} e\left(i_m\right) (-1)^{i_m} \leq \lambda_{\max},$$

ficando provado o teorema. ■

Uma característica importante da matriz A é a de ela poder ser positiva definida, negativa definida ou indefinida consoante os sinais de $e\left(i_m\right)$ e a paridade de i_m. Esta característica é sistematizada no seguinte teorema:

Teorema 5 *A matriz A é (a) positiva definida sse $e\left(i_m\right)(-1)^{i_m} > 0$ para todos os valores de $m = 0, 1, \ldots, t-1$; (b) negativa definida sse $e\left(i_m\right)(-1)^{i_m} < 0$ para $m = 0, 1, \ldots, t-1$; (c) senão, é indefinida.*

Demonstração. O sinal da equação (3.13) depende apenas de $e\left(i_m\right)(-1)^{i_m}$. ■

3.3.2 Decomposição de A

No estudo dos valores próprios de A pode ser separado o efeito das amplitudes dos erros do efeito da posição dos erros, para tal vamos definir as seguintes matrizes

$$W_{kk} = \frac{1}{\sqrt{N}} e\left(i_k\right) (-1)^{i_k} \qquad 0 \leq k \leq t-1 \tag{3.19}$$

e

$$D_{pq} = e^{-\mathrm{j}\frac{2\pi}{N} i_p q} \qquad 0 \leq p, q \leq t-1. \tag{3.20}$$

Lema 1 *A matriz A é congruente com a matriz W. Mais especificamente*

$$A = D^{\dagger} W D, \tag{3.21}$$

em que W e D foram definidas anteriormente.

Demonstração. Para demonstrar este lema basta calcular $D^\dagger W D$ e verificar a igualdade. ■

Como a matriz A é Hermítica os seus valores próprios são reais vamos ordená-los segundo a convenção

$$\lambda_{\min} = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_t = \lambda_{\max}.$$

Teorema 6 *Os valores próprios da matriz A satisfazem a desigualdade*

$$\lambda_{\min}\left(D^\dagger D\right)\lambda_k\left(W\right) \leq \lambda_k\left(A\right) \leq \lambda_{\max}\left(D^\dagger D\right)\lambda_k\left(W\right). \tag{3.22}$$

Demonstração. Pelo teorema de Ostrowski [Horn 85], para cada $k = 1, 2, \ldots, t$ existe um número real positivo θ_k de modo que

$$\lambda_1\left(D^\dagger D\right) \leq \theta_k \leq \lambda_t\left(D^\dagger D\right)$$

e

$$\lambda_k\left(D^\dagger W D\right) = \theta_k \lambda_k\left(W\right)$$

que em conjunto com $A = D^\dagger W D$, prova o teorema. ■

Convem salientar que $\lambda_k\left(W\right)$ são somente os elementos da equação (3.19) ordenados por ordem crescente.

Podemos assim dividir o problema no estudo dos valores próprios das matrizes W e T, definida por

$$T = D^\dagger D. \tag{3.23}$$

Recorrendo à definição da matriz D obtemos a seguinte equação para os elementos de T

$$T_{pq} = \sum_{m=0}^{t-1} e^{\mathrm{j}\frac{2\pi}{N} i_m (p-q)}. \tag{3.24}$$

Uma vez que T é Hermítica, $\lambda_i\left(DD^\dagger\right) = \lambda_i\left(D^\dagger D\right)$, e podemos assim definir a matriz

$$Q = DD^\dagger, \tag{3.25}$$

$$Q_{pq} = \sum_{m=0}^{t-1} e^{\mathrm{j}\frac{2\pi}{N}(i_p - i_q)m} : \qquad 0 \leq p, q \leq t-1. \tag{3.26}$$

Calculando o somatório na equação anterior, os elementos da matriz Q podem ainda ser escritos na forma

$$Q_{pq} = e^{-\mathrm{j}\frac{\pi}{N}(i_p - i_q)(t-1)} \frac{\sin\left(\pi t\left(i_p - i_q\right)/N\right)}{\sin\left(\pi\left(i_p - i_q\right)/N\right)} : \qquad 0 \leq p, q \leq t-1$$

com

$$Q_{pp} = t.$$

Por outro lado a matriz D é Vandermonde e constitui uma sub-matriz da matriz de Fourier (1.2) multiplicada por uma factor de escala, podendo ser definida do seguinte modo

$$D_{pq} = \sqrt{N} F_{i_p q},$$

e se fizermos $z = e^{-\mathrm{j}\frac{2\pi}{N}}$, podemos escrever

$$D = \begin{bmatrix} 1 & z^{i_0} & \dots & z^{(t-1)i_0} \\ 1 & z^{i_1} & \dots & z^{(t-1)i_1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & z^{i_{t-1}} & \dots & z^{(t-1)i_{(t-1)}} \end{bmatrix}. \tag{3.27}$$

As matrizes T e Q possuem semelhanças surpreendentes com a matrizes interpoladoras, que aparecem nos problemas de reconstrução para a correcção de apagamentos. A matriz Q possui uma expressão idêntica à matriz S (1.9) que surge na reconstrução de sinais limitados em frequência, na sua versão de dimensão mínima no domínio do tempo [Ferreira 97], em que a diferença principal consiste nos limites do somatório. No caso da equação (3.26) temos uma soma de t termos, correspondentes ao número de erros e na equação (1.9) teremos $(b - a + 1) = 2m + 1$ termos no somatório, correspondentes ao número de componentes espectrais diferentes de zero.

3.3.3 Limites para os valores próprios de T

Tal como foi referido no ponto anterior o estudo do condicionameto da matriz pode ser dividido em dois problemas mais simples de analisar, recorrendo ao facto de a matriz A ser *-congruente com W. Pode-se verificar facilmente que a matriz D só depende da posição dos erros e que a matriz W depende da sua amplitude. Com base no teorema 6, podemos concluir que o estudo dos valores próprios das matrizes D e W permite compreender o comportamento dos valores próprios de A.

Começaremos por obter o valor exacto para os valores próprios de $\mathbf{T}$ com dimensão um e dois. Tal como foi realizado para a matriz S, do problema de correcção de apagamentos, será igualmente analisada a variação do condicionamento da matriz T com a dimensão do bloco N, o número de erros t e a posição dos erros $\bar{S}_t$. No final daremos alguns exemplos numéricos da variação do condicionamento da matriz T com estes parâmetros.

Para a situação mais simples de apenas ter ocorrido um único erro, o valor próprio mínimo é igual ao máximo e o condicionamento da matriz T é igual à unidade. Os valores próprios são dados por

$$\lambda_{\min} = \lambda_{\max} = 1$$

Quando ocorrem dois erros a matriz T tem dimensão dois, os seus valores próprios são

$$\begin{aligned} \lambda_{\min} &= 2 - 2\left|\cos\left(\frac{(i_0 - i_1)\,\pi}{N}\right)\right| \\ \lambda_{\max} &= 2 + 2\left|\cos\left(\frac{(i_0 - i_1)\,\pi}{N}\right)\right| \end{aligned}$$

e o condicionamento de T é,

$$\kappa(T) = \frac{\lambda_{\max}}{\lambda_{\min}} = \frac{2}{1 - \left|\cos\left(\frac{(i_0 - i_1)\pi}{N}\right)\right|} - 1. \tag{3.28}$$

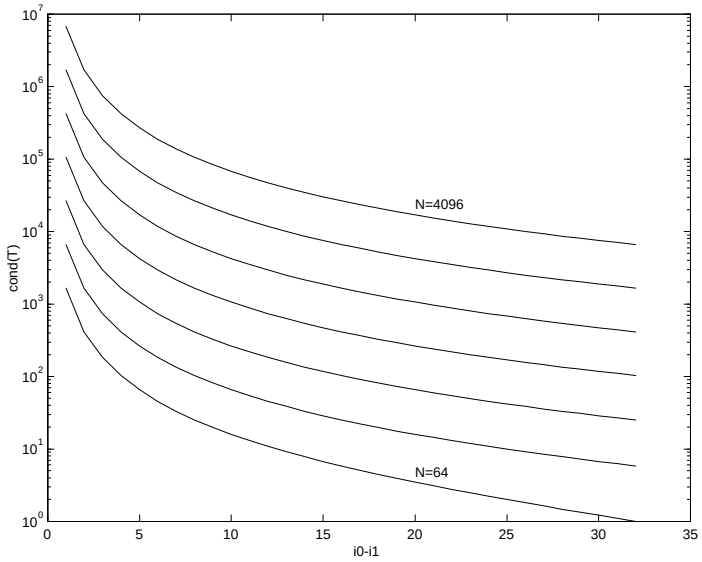


Figura 3.2: Esta figura ilustra a variação do condicionamento da matriz T em função da distância entre os dois erros $i_0 - i_1$, dada pela equação (3.28). As curvas apresentadas são para valores de N da forma para $N = 2^6$ e a superior para $N = 2^{12}$.

Na figura 3.2 podemos apreciar um gráfico da equação anterior que mostra a variação do condicionamento da matriz T em função da distância entre erros i_0 e i_1. A curva inferior é para $N = 64$ e a superior para $N = 4096$. Como se pode ver, o condicionamento degrada-se bastante quando se aumenta a dimensão do bloco N. Para um dado N, a situação de erros contiguos representa o pior caso para o condicionamento de T.

Comparando o condicionamento da matriz T dado pela equação anterior com o condicionamento da matriz $(I - S)$ dado por (3.10), facilmente se verifica que $\kappa(T)$, ao contrário de $\kappa(I - S)$, não depende do factor de interpolação. Este facto leva a que quando a dimensão N do bloco aumenta o problema se vá tornando cada vez mais mal condicionado. No entanto, se o sindroma for maior do que o necessário, pode-se construir um sistema de equações sobredeterminado e utilizar a pseudo-inversa, obtendo-se maior estabilidade numérica e passando o condicionamento da matriz T a ser uma função de β e não de N.

Variação do condicionamento de T com a dimensão N do bloco

O comportamento do condicionamento da matriz T para $t = 2$, ou seja dois erros, mostra que pelo menos para este caso, o condicionamento do problema piora bastante quando se aumenta a dimensão N do bloco de dados. Iremos demonstrar que se passa o mesmo para qualquer outro valor de t.

Teorema 7 *Para um dado t e $\bar{S}_t$, o condicionamento da matriz T aumenta quando N aumenta.*

Demonstração. O condicionamento de uma matriz é definido como

$$\kappa(T) = \frac{\lambda_{\max}}{\lambda_{\min}},$$

e cada elemento da matriz T é dado pela equação (3.24). Se considerarmos fixos $\bar{S}_t = \{i_0, i_1, \ldots, i_{t-1}\}$ e t, então, temos que quando $N \to \infty$, $e^{-\mathrm{j}\frac{2\pi}{N} i_m (q-p)} \to 1$, e os elementos da matriz T convergem para t. Nesta situação, o máximo e o mínimo dos valores próprios de T convergem para os seguintes valores

$$\lambda_{\min} \to 0, \qquad \lambda_{\max} \to t^2,$$

e por conseguinte $\kappa(t) \to \infty$. ■

Variação do condicionamento da matriz T com o número de erros t

Apesar de em geral o condicionamento de T piorar quando o número de erros aumenta, existem situações em que tal não se verifica. Consideremos o seguinte exemplo com $N = 32$, $t = 3$, $K = 17$ e $\bar{S}_t = \{0, 1, 2\}$, em que a respectiva matriz T tem um condicionamento de 53000. Se acrescentarmos ao problema anterior um erro na posição 18, verifica-se que o condicionamento de T melhora ficando com um valor de 11500.

Variação do condicionamento da matriz T com a posição dos erros $\bar{S}_t$.

Como a matriz T é hermítica podemos considerar a forma quadrática associada

$$x^\dagger T x,$$

que se pode escrever na forma

$$x^\dagger T x = x^\dagger D^\dagger D x = \|Dx\|^2, \tag{3.29}$$

fazendo uso da equação (3.23). Se utilizarmos a equação (3.29), os valores próprios de T podem ser obtidos a partir das seguintes equações

$$\begin{aligned} \lambda_{\max} &= \max_{x \neq 0} \frac{\|Dx\|^2}{x^\dagger x}, \\ \lambda_{\min} &= \min_{x \neq 0} \frac{\|Dx\|^2}{x^\dagger x}. \end{aligned}$$

Se $i_p = p$ (erros contíguos) as linhas da matriz D (3.27) são quase linearmente dependentes, fazendo com que o condicionamento de T dado pela relação $\lambda_{\max}/\lambda_{\min}$ tenha um valor elevado. Por outro lado, se o espaçamento entre erros aumentar, as linhas da matriz D vão-se aproximar da situação em que são ortogonais. Esta última situação é verificada quando N é da forma $N = kt$ e $i_p = kp$, pois neste caso temos

$$D_{pq} = e^{-\mathrm{j}\frac{2\pi}{N} i_p q} = e^{-\mathrm{j}\frac{2\pi}{kt} kpq} = e^{-\mathrm{j}\frac{2\pi}{t} pq} = \sqrt{N} F_{pq},$$

em que F_{pq} é a matriz de Fourier de dimensão t. Nesta caso, corrigir t erros é equivalente em termos de condicionamento numérico a corrigir apenas 1.

Este resultado é semelhante ao obtido por Ferreira em [Ferreira 94a] mas no contexto da determinação das sequências de amostragem óptimas.

3.3.4 Limites para os valores próprios de W

A matriz W definida em (3.19) depende simultaneamente da amplitude dos erros $e(i_m)$ e da sua posição i_m. Considere-se o caso mais realista do ponto de vista de uma possível implementação, que todos os sinais envolvidos estão quantificados com um número finito b de bits, e que por conseguinte, se pode dizer que o sinal de erro respeita a desigualdade

$$e_{\min} \leq |e(i_m)| \leq e_{\max}. \tag{3.30}$$

Como a matriz W é diagonal, se a desigualdade anterior se verificar, os seus valores próprios satisfazem a desigualdade

$$\frac{e_{\min}}{\sqrt{N}} \leq |\lambda_k(W)| \leq \frac{e_{\max}}{\sqrt{N}}.$$

Podemos assim reescrever o resultado do teorema 6 para o pior caso dos valores próprios de W, que como facilmente se pode constatar, acontece quando temos em simultâneo erros de amplitude mínima ($e_{\min}$) e máxima ($e_{\max}$)

$$\lambda_{\min}(T)\frac{e_{\min}}{\sqrt{N}} \leq \lambda_k(A) \leq \lambda_{\max}(T)\frac{e_{\max}}{\sqrt{N}}.$$

O condicionamento de A vem então

$$\kappa(A) = \frac{e_{\max}}{e_{\min}}\frac{\lambda_{\max}(T)}{\lambda_{\min}(T)}$$

que indica que o condicionamento de A é inversamente proporcional ao factor

$$\epsilon = \frac{e_{\min}}{e_{\max}}.$$

3.3.5 Limite superior para o condicionamento de A

Na equação (3.13), o termo $|\phi(i_m)|^2$ é sempre positivo, e portanto para uma dada sequência de erros $\tilde{S}_t$, esta equação atinge valores extremos para

$$\begin{aligned} \tilde{\lambda}_{\max}(A) &: \quad e(i_m) = e_{\max}(-1)^{i_m} \\ \tilde{\lambda}_{\min}(A) &: \quad e(i_m) = e_{\min}(-1)^{i_m}. \end{aligned} \tag{3.31}$$

Como foi indicado na secção 3.3.3, a pior sequência para os erros corresponde a $i_m = m$, e para esta situação $\tilde{\lambda}_{\max}(A)$ atinge o seu maior valor e $\tilde{\lambda}_{\min}(A)$ o seu menor. Vamos chamar a cada um destes valores $\check{\lambda}_{\max}(A)$ e $\check{\lambda}_{\min}(A)$ respectivamente. Podemos então afirmar que para um problema com um dado N e t, $\check{\lambda}_{\max}(A)$ é o maior valor possível para $\lambda(A)$ e $\check{\lambda}_{\min}(A)$ o menor valor possível para $\lambda(A)$. Não é difícil de verificar que o condicionamento de A é majorado pela relação

$$\kappa(A) \leq \frac{\check{\lambda}_{\max}(A)}{\check{\lambda}_{\min}(A)}. \tag{3.32}$$

Apesar de dar uma estimativa muito por cima do limite para o condicionamento de A, esta equação tem algum interesse prático pois permite ter uma ideia do pior condicionamento possível para A. Para calcular esta estimativa, basta para um dado problema em que se conhece t, N e ϵ, construir a matriz A para a situação de erros consecutivos e amplitudes dadas por (3.31) e calcular o condicionamento dessa matriz. Na secção seguinte serão dados alguns exemplos numéricos deste limite e do seu desvio em relação ao valor correcto. Por último convém notar que $T = A$ quando a condição (3.31) se verifica.

3.3.6 Estudo da variação do condicionamento de A

Nesta secção iremos estudar a variação do condicionamento da matriz A em função de vários factores, nomeadamente:

- A dimensão N do problema de reconstrução
- O número de erros ocorridos t
- A posição dos erros.

Os primeiros testes que se realizaram ao modo como os factores mencionados em cima influenciavam o condicionamento do problema de determinação da posição dos erros, revelaram que com o aumento do número de erros t ou do número de amostras N do bloco, o condicionamento piorava. Verificou-se, igualmente, que quando os erros aconteciam de forma contígua em vez de esparsa, o condicionamento piorava.

No caso da matriz S, o seu condicionamento piora quando se baixa o factor de interpolação β, e tal como foi demonstrado, quando se aumenta a dimensão N do bloco, se se aumentar a dimensão do sindroma de modo a que β permaneça constante, então, o condicionamento de S não aumenta.

Este comportamento fica-se a dever à informação que cada um dos algoritmos utiliza. O primeiro recorre apenas à amplitude das amostras conhecidas para determinar a amplitude de parte do espectro das amostras erradas (sindroma), enquanto que o segundo faz apenas uso da amplitude das amostras erradas. Um dos problemas com o algoritmo de determinação da posição dos erros, resulta do facto que quando ocorre um número $t < M/2$ de erros, este utiliza apenas $2t$ em vez das M componentes disponíveis do sindroma para determinar a posição dos erros. Como foi descrito no capítulo 2 é possível utilizar as restantes $M - 2t$ componentes do sindroma para melhorar a estabilidade na determinação da posição e amplitude das amostras erradas calculando a solução que minimiza o erro quadrático.

3.3.7 Resultados experimentais

Numa tentativa de ganhar algum conhecimento sobre o condicionamento do problema e dada a dificuldade de calcular limites rigorosos para o condicionamento da matriz A, optou-se por utilizar métodos numéricos para analisar o problema para casos de pequena dimensão. Existem várias questões que não têm resposta fácil, tal como por exemplo:

- Qual a pior sequência para o sinal de erro em termos do condicionamento de A?
- Qual a variação do condicionamento de A com o padrão dos erros $\bar{S}_t$?
- Qual a variação do condicionamento de A com o número de erros ocorridos?

Pior sequência para a amplitude do sinal de erro

Nesta experiência determina-se qual a pior sequência para as amplitudes dos erros considerando o caso particular dos erros ocorrerem em posições contíguas. Considera-se que o sinal de erro satisfaz a condição (3.30).

O tipo de testes que foram realizados são exaustivos, ou seja, para um dado problema com bloco de dimensão N, o valor de $\kappa(A)$ é calculado para todas as combinações de posições de erros e para cada uma destas, testam-se todas as combinações de amplitudes de

t	$e\left(\bar{S}_t\right)/\lambda_{\max}\left(A\right)$	$e\left(\bar{S}_t\right)/\lambda_{\min}\left(A\right)$
1	{-1}/0.177	{-ϵ}/0.0884
2	{-1,1}/0.705	{-ϵ,ϵ}/8.5E-4
3	{-1,1,-1}/1.56	{-ϵ,ϵ,-ϵ}/1.47E-5
4	{-1,1,-1,1}/2.67	{-ϵ,ϵ,-ϵ,ϵ}/4.7E-7
5	{-1,1,-1,1,-1}/3.82	{-ϵ,ϵ,-ϵ,ϵ,-ϵ}/2.44E-8
6	{-1,1,-1,1,-1,1}/4.76	{-ϵ,ϵ,-ϵ,ϵ-ϵ,ϵ}/1.91E-9
7	{-1,1,-1,1,-1,1,-1}/5.33	{-ϵ,ϵ,-ϵ,ϵ,-ϵ,ϵ,-ϵ}/2.11E-10

Tabela 3.1: Nesta tabela temos as piores sequências para o sinal de erro em termos dos valores próprios de A. Os resultados foram obtidos com $N = 32$, $e_{max} = 1$, $e_{min} = \epsilon = 0.5$, com t a variar de 1 a 7 e $e\left(\bar{S}_t\right)$=$\{0, \ldots, t-1\}$. Tal como afirmado anteriormente, os piores casos em termos de valores próprios são atingidos quando os erros são do tipo $e\left(i_m\right) = \epsilon\left(-1\right)^{i_m}$.

t	$e\left(\bar{S}_t\right)/\kappa_{\max}\left(A\right)$	$e\left(\bar{S}_t\right)/\kappa_{\min}\left(A\right)$
1	{-1}/1	{-1}/1
2	{-ϵ,1}/466	{-1,-1}/1
3	{-1,ϵ,-1}/7.31E4	{-ϵ,-1,ϵ}/9.74
4	{-1,ϵ,-ϵ,1}/3.97E6	{-ϵ,-1,-1,-ϵ}/355
5	{-1,ϵ,-ϵ,1,-1}/1.05E8	{-ϵ,1,1,1,-1}/1.76E5
6	{-1,1,-ϵ,ϵ,-1,1}/1.75E9	{-ϵ,-1,-1,-1,-1,-ϵ}/9.97E4
7	{-1,1,-ϵ,ϵ,-ϵ,1,-1}/1.69E10	{-ϵ,-1,-1,-1,-1,-1,-ϵ}/4.84E6

Tabela 3.2: Nesta tabela temos as piores sequências para o sinal de erro em termos do condicionamento da matriz A, para a situação $N = 32$, $e_{min} = 1$, e $e_{max} = \epsilon = 0.5$ e com t a variar de 1 a 7.

erros. O número de combinações cresce muito rapidamente tornando impraticável realizar simulações para sequências com valores elevados de t e factor $\epsilon = e_{\min}/e_{\max}$ muito pequeno. Nomeadamente, o número de hipóteses é dado por

$$\left(\left\lceil\frac{2}{\epsilon}\right\rceil + 1\right)^t,$$

sendo a amplitude dos erros incrementada em passos iguais a ϵ. Na tabela 3.1 podemos observar alguns resultados em que se mostram quais as sequências que dão o menor e o maior valor próprio, e na tabela 3.2 o maior e o menor condicionamento de A.

Apesar de na tabela 3.2 se ser tentado a deduzir uma regra para construir a sequência para o pior caso do condicionamento, existem contra exemplos, ou seja, se se variar a amplitude do erro mínimo, a pior sequência varia de forma, ver a tabela 3.4.

Variação do condicionamento de A com N

Neste estudo pretende-se averiguar a variação do condicionamento da matriz A em função da dimensão do bloco N, mantendo fixos todos os outros parâmetros. No caso particular de $t = 2$, verificamos pela equação (3.28) que o condicionamento de A piora quando se aumenta N. Como seria de esperar, para valores superiores de t verifica-se um comportamento idêntico.

t	$e\left(\bar{S}_t\right)/\lambda_{\max}\left(A\right)$	Seq./$\lambda_{\min}\left(A\right)$
1	{-1}/0.177	{-ϵ}/0.0442
2	{-1,1}/0.705	{-ϵ,ϵ}/4.26E-4
3	{-1,1,-1}/1.56	{-ϵ,ϵ,-ϵ}/7.37E-6
4	{-1,1,-1,1}/2.67	{-ϵ,ϵ,-ϵ,ϵ}/2.35E-7
5	{-1,1,-1,1,-1}/3.82	{-ϵ,ϵ,-ϵ,ϵ,-ϵ}/1.22E-8
6	{-1,1,-1,1,-1,1}/4.76	{-ϵ,ϵ,-ϵ,ϵ-ϵ,ϵ}/9.53E-10
7	{-1,1,-1,1,-1,1,-1}/5.33	{-ϵ,ϵ,-ϵ,ϵ,-ϵ,ϵ,-ϵ}/1.05E-10

Tabela 3.3: Nesta tabela temos as piores sequências para o sinal de erro em termos dos valores próprios de A. Os resultados foram obtidos com $N = 32$, e $\epsilon = 0.25$, com t a variar de 1 a 7 e $e\left(\bar{S}_t\right)= \{0, \ldots, t-1\}$. Tal como era esperado os padrões de amplitudes são iguais aos da tabela 3.1.

t	$e\left(\bar{S}_t\right)/\kappa_{\max}\left(A\right)$	$e\left(\bar{S}_t\right)/\kappa_{\min}\left(A\right)$
1	{-1}/1	{-1}/1
2	{-ϵ,1}/648	{-1,-1}/1
3	{-1,ϵ,-1}/1.18E5	{-ϵ,-1,-1}/948
4	{-1,ϵ,-ϵ,1}/6.35E6	{-ϵ,-1,-1,-ϵ}/425
5	{-1,ϵ,-ϵ,1,-1}/1.68E8	{-1,-1,-1,-1,ϵ}/1.11E5
6	{-1,1,-ϵ,ϵ,-1,1}/2.85E9	{-ϵ,1,1,1,1,-ϵ}/1.57E5
7	{-ϵ,ϵ,-ϵ,ϵ,-1,1,-1}/2.7E10	{-ϵ,-1,-1,-1,-1,-1,-ϵ}/5.26E6

Tabela 3.4: Nesta tabela temos as piores sequências para o sinal de erro em termos do condicionamento da matriz A, para a situação $N = 32$, e $\epsilon = 0.25$ e com t a variar de 1 a 7. No caso do condicionamento de A, e alterando somente ϵ, os padrões de amplitude mudam em relação aos da tabela 3.2.

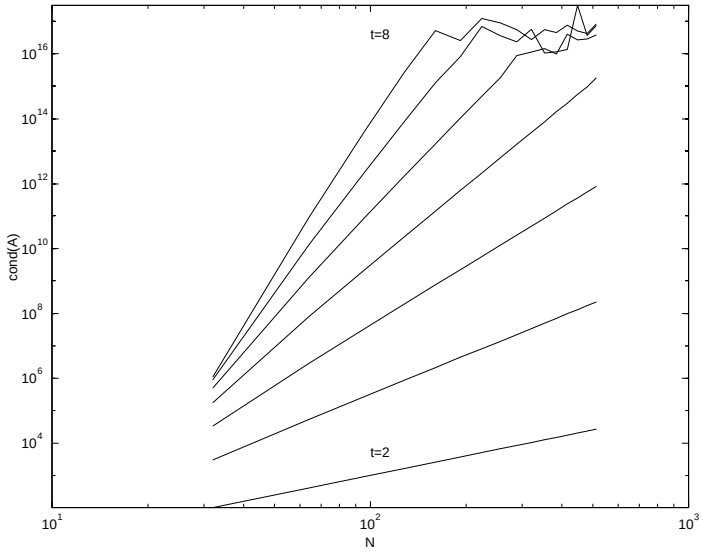


Figura 3.3: Variação do condicionamento da matriz A em função de N. A curva inferior é para $t = 2$ (número de erros) e a superior para $t = 8$. As irregularidades nas curvas superiores devem-se à precisão finita com que os cálculos foram efectuados.

Na figura 3.3 podemos ver um gráfico da variação referida para $N = 32 \ldots 512$, $t = 2 \ldots 8$ e erros de amplitude $+1$ e espaçamento 2.

Variação do condicionamento de A com o número de erros t

O gráfico da figura 3.4 mostra a variação do condicionamento de A com o número de erros ocorridos, nas situações em que a posição dos erros é dada por

$$i_m = km, \qquad m = 0, 1, \ldots, t-1 \qquad k = 1, 2, \ldots, n$$

com $n \leq \left\lfloor \frac{N-1}{t-1} \right\rfloor$. A amplitude dos erros usada foi de $e(i_m) = (-1)^{i_m}$, e neste situação a matriz A é igual a T. Como se pode verificar, quando os erros são consecutivos o condicionamento de A degrada-se rapidamente.

3.3.8 Conclusões e resultados

Pensamos que o resultado mais importante deste capítulo consiste na separação da matriz A, que veio permitir a análise do efeito da determinação da posição dos erros, de forma independente do efeito da sua amplitude no valor de $\kappa(A)$. A demonstração de que a estabilidade do problema de determinação das posições das amostras erradas piora quando se aumenta a dimensão do bloco é igualmente um resultado importante. Apesar da semelhança entre a matriz interpoladora S e a matriz Q, pensamos que conseguimos demonstrar que os problemas de estabilidade são bastante mais acentuados no caso do segundo problema. Foi igualmente encontrado um limite superior para o condicionamento de T, mas que peca por ser demasiado conservador. Finalmente, as tabelas dadas no final do capítulo, permitem uma observação empírica do comportamento do condicionamento do problema para situações extremas.

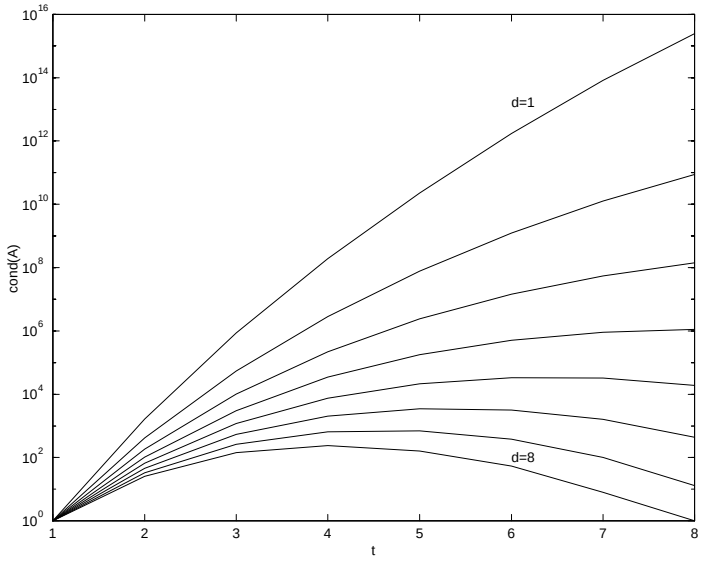


Figura 3.4: Variação do condicionamento da matriz A em função do número de erros t. A curva inferior é para um espaçamento d entre erros de 8 e a curva superior para um espaçamento de 1 para um bloco de dados com $N = 64$.

Capítulo 4

Correcção de erros com aritmética real

A correcção de erros em sistemas digitais costuma ser realizada recorrendo a aritmética finita. A álgebra envolvida no estudo dos códigos correctores de erros requer um conjunto de conhecimentos que não faz parte da formação base em matemática dos cursos de engenharia electrónica, que abordam unicamente álgebra real e complexa. Este facto constitui uma barreira para a compreensão das ligações existentes entre a teoria dos códigos de correcção de erros e disciplinas como a teoria do controlo ou o processamento digital de sinal. Um dos primeiros autores a identificar a equivalência de conceitos e técnicas foi R. Blahut [Blahut 83], chegando mesmo a chamar à teoria dos códigos de correcção de erros, "processamento digital de sinal em corpos finitos". Esta perspectiva foi seguida por outros autores como Marshall, Kumaresan, Wolf e Wu, que transportaram vários códigos e algoritmos utilizados na correcção de erros para o corpo dos reais. Marshall, demonstra em [Marshall Jr 84] que para qualquer código de correcção de apenas um erro definido num corpo finito existe um código equivalente no corpo dos reais ou complexos. A interpretação realizada por Blahut dos códigos BCH no domínio da frequência e a sua transposição imediata para o corpo dos complexos constitui um marco na aproximação entre a teoria dos códigos de correcção de erros e a reconstrução na teoria do sinal.

Neste capítulo faz-se uma transposição de algumas definições, propriedades e conceitos relativos aos códigos de correcção de erros dos corpos finitos para o corpo dos complexos. Definiremos os códigos lineares e a classe dos códigos cíclicos no corpo dos reais. Será dada uma especial ênfase aos códigos de Reed-Muller, dada a sua facilidade de realização prática e a estimativa rigorosa que permitem do erro numérico. Será igualmente descrito um código por blocos sistemático, baseado na tranformada de Fourier. Descreveremos os códigos do tipo BCH, demonstrando a ligação com alguns dos métodos de reconstrução de sinal do capítulo 2.

No final abordaremos de forma breve os códigos convolucionais com o objectivo de demonstrar a sua ligação com os bancos de filtros. Demonstraremos que se um banco de filtros possui a propriedade de reconstrução perfeita, então é possível gerar no receptor um sinal que é função apenas do sinal de erro e.

4.1 Códigos por blocos no corpo dos complexos

4.1.1 Estrutura dos códigos lineares

Considere-se o espaço vectorial $\mathbb{C}^N$ em que cada vector é composto por N números complexos designados por amostras ou símbolos[1]. Vamos considerar que a soma de dois vectores e o producto de uma constante por um vector são definidos da forma habitual. Um código linear pode então ser descrito como sendo um subespaço ϑ de $\mathbb{C}^N$.

Nesta secção iremos apenas falar de códigos por blocos que se definem do seguinte modo:

Definição 3 *Um código por blocos em que o vector ou sinal a codificar tem K elementos e o vector codificado tem uma dimensão $N > K$ é designado por código do tipo (N, K).*

Vamos agora definir a seguinte métrica que mede a distância entre dois vectores de um espaço e que costuma ser designada por distância de Hamming.

Definição 4 *A distância de Hamming $d(a, b)$ entre duas palavras a e b de um código no corpo dos reais é dada pelo número de amostras em que estas diferem.*

A definição anterior permite definir a noção de peso de um dado código.

Definição 5 *O peso de um código linear é igual ao valor mínimo da distância de Hamming de qualquer vector desse código em relação ao vector nulo.*

A definição 4 para a distância de Hamming foi criada originalmente para o corpo finito $\mathrm{GF}(2^p)$ mas é perfeitamente generalizável para o corpo dos reais ou complexos. Contudo, no corpo dos reais deve-se ter em conta o erro cometido nos cálculos. Assim, a definição mais correcta e que foi utilizada nas simulações é a seguinte:

Definição 6 *A distância de Hamming $d(a, b, \epsilon)$ entre duas palavras a e b de um código no corpo dos reais é dada pelo número de amostras em que a diferença $|a_i - b_i|$ é superior a ϵ, sendo ϵ um número positivo maior que zero.*

Note-se que para qualquer código linear o resultado da diferença entre dois vectores pertencentes ao código pertence igualmente ao subespaço do código, e por isso, o vector nulo pertence sempre ao subespaço. Assim, a distribuição dos vectores pertencentes a ϑ, com uma distância d em torno de cada um dos vectores é idêntica à distribuição em torno do vector nulo.

4.1.2 Descrição matricial dos códigos lineares

Os códigos lineares podem ser descritos na forma matricial, com evidentes vantagens para a sua compreensão e estudo. Considere-se um código linear (N, K), em que o vector $m \in \mathbb{R}^K$ a codificar resulta no vector $c \in \mathbb{R}^N$. Então, a operação de codificação pode ser descrita pela equação matricial

$$c = Gm$$

[1] As componentes de um vector são designadas em PDS por amostras e na TCCE por símbolos. Neste documento usaremos qualquer um dos termos.

em que a matriz G tem dimensão $N \times K$. As colunas de G são linearmente independentes e o seu número K é a dimensão do subespaço ϑ do código, uma vez que qualquer palavra do código é uma combinação linear das colunas de G.

Dado que ϑ é um subespaço de dimensão K de $\mathbb{C}^N$, existe um subespaço ortogonal $\vartheta^{\perp}$ de dimensão $N-K$ que define o código dual de G. Se este código dual for definido pela matriz H, então as palavras do código pertencentes ao subespaço ϑ são ortogonais a H, ou seja

$$H^{\dagger}c = 0, \tag{4.1}$$

sendo esta equação um teste de verificação para todas as palavras do código. Por este motivo, H é conhecida como a matriz verificação de paridade. Uma vez que c é uma combinação linear das colunas de G tem-se

$$H^{\dagger}G = 0.$$

4.1.3 Códigos sistemáticos

Este tipo de código caracteriza-se por as suas palavras serem da forma

$$c = \begin{bmatrix} m \\ p \end{bmatrix},$$

sendo m o vector original ao qual se acrescentou o vector de paridade p com $N-K$ amostras que poderão permitir detectar e corrigir eventuais erros. Qualquer código linear tem um equivalente sistemático em que a matriz de codificação é dada por

$$G = \begin{bmatrix} I_K \\ P \end{bmatrix}, \tag{4.2}$$

sendo I_K a matriz identidade de dimensão $K \times K$ e P uma matriz de dimensão $(N-K) \times K$. Quando a matriz de codificação G toma a forma (4.2) vem

$$H = \begin{bmatrix} -P^{\dagger} \\ I_{N-K} \end{bmatrix}$$

porque

$$H^T G = \begin{bmatrix} -P & I_{N-K} \end{bmatrix} \begin{bmatrix} I_K \\ P \end{bmatrix} = 0.$$

Teorema 8 (Limite de Singleton) *A distância de Hamming mínima entre duas palavras de qualquer código linear satisfaz*

$$d \leq 1 + N - K. \tag{4.3}$$

Demonstração. Considere-se um dado código (N, K) com distância mínima d. Quando colocado na forma sistemática existem palavras do código só com um símbolo diferente de zero e $N-K$ símbolos de paridade. Uma tal palavra do código não pode ter um peso superior a $1 + (N-K)$ e assim a distância mínima do código não pode ser superior a $1 + N - K$. ■

Qualquer código em que a distância mínima seja dada por $1 + N - K$ é designado por CDM (Código de Distância Máxima) [Blahut 83].

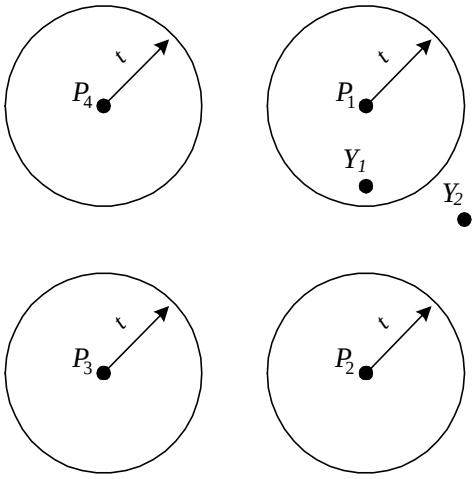


Figura 4.1: Interpretação gráfica do limite de Singleton para códigos lineares em que P_i representa uma palavra do código e Y_t uma palavra recebida com erros.

Uma conclusão imediata do limite de Singleton é que para um código consiguir corrigir t erros é necessário que este tenha pelo menos $2t$ símbolos de paridade, apresentado-se na figura 4.1 uma descrição gráfica do problema. Imaginem-se palavras válidas de um código linear dispostas num plano à distância $1 + N - K = 1 + 2t$ umas das outras. Consideremos agora as hiper-esferas de diâmetro t que indicam o conjunto de palavras erradas com t ou menos erros. Se for transmitida a palavra P_1 e à chegada tivermos Y_1, como esta palavra está dentro da hiper-esfera é possível recuperar a palavra correcta P_1. Se por exemplo a palavra recebida fosse Y_2 com mais do que t erros, o código não seria capaz de a corrigir porque esta se encontra fora da hiper-esfera. Nesta situação o descodificador pode optar por devolver uma mensagem indicando erro na descodificação ou então tentar associar o símbolo recebido ao código P_i que estiver mais próximo.

4.1.4 Exemplos de códigos lineares em $\mathbb{C}$

Nesta secção iremos descrever alguns exemplos de códigos lineares no corpo dos complexos que por não serem cíclicos não podem tirar partido da estrutura acrescida da matriz de codificação e assim requerem um tratamento diferenciado do utilizado nos códigos do tipo BCH. Apesar do tema ser bastante tratado em livros de texto clássicos [Blahut 83, Sweeney 91, Berlekamp 84] estes referem-se sempre ao caso dos corpos finitos. Para o caso dos reais e complexos a informação encontra-se dispersa por vários artigos.

Código sistemático com a DFT

Em [Marshall Jr 84], é abordado o tema dos códigos sistemáticos no corpo dos complexos sem se considerar os aspectos práticos da codificação e descodificação. Como vimos anteriormente, a equação (4.2) implementa a codificação gerando $2t$ símbolos de paridade. Considere-se a matriz de Fourier tal como definida em (1.2) e vamos particionar a matriz F tal como

em [Marshall Jr 84]

$$F = \begin{bmatrix} F_a & F_b \\ F_c & F_d \end{bmatrix}. \tag{4.4}$$

Com base na matriz anterior e uma vez que F_a e F_d são invertíveis, podemos definir a matriz

$$T = \begin{bmatrix} F_a^{-1} & 0 \\ 0 & F_d^{-1} \end{bmatrix}$$

e escrever

$$F_s = TF = \begin{bmatrix} I & F_a^{-1}F_b \\ F_d^{-1}F_c & I \end{bmatrix} = \begin{bmatrix} G_s & H_s \end{bmatrix},$$

em que a matriz P da equação (4.2) é dada por

$$P = F_d^{-1}F_c = -F_b^{\dagger}F_a^{-\dagger}$$

com $F_d^{-\dagger} = \left(F_d^{\dagger}\right)^{-1}$.

Vamos considerar que um sinal m com K amostras foi codificado com o código sistemático definido, tendo-se obtido o vector c_s

$$c_s = G_s m = \begin{bmatrix} m \\ Pm \end{bmatrix}. \tag{4.5}$$

Se algumas das amostras deste sinal forem corrompidas teremos

$$y_s = c_s + e_s.$$

Para realizar a codificação basta efectuar o producto Pm tendo-se assim $(N-K) \times K$ multiplicações e somas.

A descodificação deste código sistemático pode ser realizada recorrendo à descodificação de um código BCH, tendo em conta que as matrizes G_s e H_s podem ser escritas em função de T

$$\begin{aligned} H_s &= TH, \\ G_s &= TG, \end{aligned}$$

podendo-se então escrever o sindroma S em função de H

$$S = H_s^{\dagger} y_s = H^{\dagger}T^{\dagger}y_s. \tag{4.6}$$

Se transformarmos o sinal recebido y_s de um codificador sistemático em

$$y = T^{\dagger}y_s,$$

teremos

$$\begin{aligned} S &= H^{\dagger}y = H^{\dagger}T^{\dagger}y_s = H^{\dagger}T^{\dagger}\left(c_s + e_s\right), \\ S &= H^{\dagger}T^{\dagger}TGm + H^{\dagger}T^{\dagger}e_s = H^{\dagger}T^{\dagger}e_s = H^{\dagger}e, \end{aligned}$$

uma vez que $H^\dagger T^\dagger TG = 0$ e com $e = T^\dagger e_s$. Então, com um descodificador BCH calcula-se o valor do sinal de erro e obtendo-se finalmente o sinal de erro ocorrido no código sistemático

$$e_s = T^{-\dagger} e.$$

Apesar da grande vantagem de neste tipo de códigos os dados originais estarem disponíveis no vector recebido, estes possuem algumas desvantagens em relação aos códigos BCH. As operações de codificação (4.5) e de descodificação (4.6) não podem ser realizadas recorrendo à FFT e têm de ser implementados como multiplicação de matrizes. No primeiro caso temos $O(N \log_2 N)$ e no segundo $O(2Nt)$, sendo portanto os códigos sistemáticos compensadores apenas para valores reduzidos de t.

Códigos de Reed-Muller

Os códigos de Reed-Muller (RM) apesar de não possuirem uma grande eficiência, são no entanto muito simples e podem ser descodificados por uma simples técnica de voto (maioritário). Este tipo de código é normalmente definido em corpos finitos, sendo contudo possível modificar a matriz de codificação de modo a que funcionem no corpo dos reais ou dos complexos. A descrição aqui realizada baseia-se no artigo de Jiun Sien e J. Wu [Shiu 96], que adapta ao corpo dos reais $\mathbb{R}$ o trabalho de Blahut [Blahut 83] sobre os códigos RM em corpos finitos. Nesta secção será apresentada uma técnica de codificação e descodificação diferente da apresentada em [Shiu 96], que pensamos possuir um melhor desempenho. Será igualmente demonstrado que a formação das equações para votação por maioria difere da realizada para o caso da praticada num corpo finito.

Comecemos por lembrar como se constrói um código RM para corpos finitos. Para cada inteiro p existe um código de Reed-Muller de dimensão 2^p. A sua construção é definida pela matriz de codificação

$$G = \left[\begin{array}{cccc} G_0 & G_1 & \cdots & G_r \end{array} \right], \tag{4.7}$$

em que G_0 é um vector coluna de dimensão $N = 2^p$ composto unicamente por elementos $G_0(i) = 1$, G_1 é uma matriz de dimensão $2^p \times p$ em que aparecem todas as combinações possíveis dos números binários formados com p bits. A matriz G_l é construída realizando todos os productos de Hadamard possíveis com l colunas de G_{l-1}, sendo o número total de matrizes limitado a $r \leq p$.

O producto entre as colunas da matriz referido em cima é o producto de Hadamard, definido como se segue:

Definição 7 *O producto de Hadamard entre duas matrizes A e B com elementos a_{ij} e b_{ij}, é dado pelo producto elemento a elemento*

$$C = A \circ B \ : \ c_{ij} = a_{ij} b_{ij}.$$

De seguida mostramos um exemplo para a matriz de codificação de um código de Reed-

Muller sobre $GF(2)$, para $p = 4$, $N = 16$ e $r = 2$.

$$
\begin{aligned}
G_0 &= \left[\begin{array}{cccccccccccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right]^T = [a_0]^T, \\
G_1 &= \left[\begin{array}{cccccccccccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{array}\right]^T = \left[\begin{array}{c} a_1 \\ a_2 \\ a_3 \\ a_4 \end{array}\right]^T, \\
G_2 &= \left[\begin{array}{cccccccccccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right]^T = \left[\begin{array}{c} a_1 \circ a_2 \\ a_1 \circ a_3 \\ a_1 \circ a_4 \\ a_2 \circ a_3 \\ a_2 \circ a_4 \\ a_3 \circ a_4 \end{array}\right]^T.
\end{aligned}
$$

O código anterior é do tipo (16,11), dado que acrescenta 5 símbolos de "paridade" a cada bloco com 11 símbolos do sinal a codificar. Repare-se que nos códigos Reed-Muller a dimensão do vector codificado é dada por

$$K = 1 + \binom{p}{1} + \cdots + \binom{p}{r}.$$

A codificação é realizada multiplicando a mensagem m pela matriz de codificação G. Atendendo à estrutura de G podemos escrever a equação de codificação como

$$c = \left[\begin{array}{cccc} G_0 & G_1 & \cdots & G_r \end{array}\right] \left[\begin{array}{c} M_0 \\ M_1 \\ \vdots \\ M_r \end{array}\right], \tag{4.8}$$

em que a mensagem m foi dividida em parcelas M_j, com dimensão $\binom{p}{j}$. A palavra recebida y virá corrompida com o sinal de erro e, ou seja

$$y = \left[\begin{array}{cccc} G_0 & G_1 & \cdots & G_r \end{array}\right] \left[\begin{array}{c} M_0 \\ M_1 \\ \vdots \\ M_r \end{array}\right] + e.$$

O algoritmo de descodificação começa por tentar reconstruir apenas M_r, subtraindo depois o seu efeito do vector recebido y pela equação

$$y^{(r-1)} = y^{(r)} - G_r M_r, \tag{4.9}$$

com $y = y^{(r)}$, reduzindo assim a ordem do código de r para $r - 1$, sendo possível então aplicar recursivamente o mesmo algoritmo até se ter obtido todos os símbolos da mensagem transmitida m. No exemplo apresentado os vectores M_k são definidos por

$$
\begin{aligned}
M_0 &= [m_0]^T \\
M_1 &= [m_1\ m_2\ m_3\ m_4]^T \\
M_2 &= [m_5\ m_6\ m_7\ m_8\ m_9\ m_{10}]^T.
\end{aligned}
$$

Em cada uma das etapas determina-se o valor de cada um dos símbolos de M_r, formando 2^{p-r} equações sobre os 2^p símbolos de paridade recebidos, envolvendo cada equação 2^r símbolos. As equações são formadas de modo a que cada símbolo a descodificar apareça uma única vez e os outros um número par de vezes o que conduz ao seu cancelamento. Se não ocorrerem erros as equações para cada um dos símbolos dão o mesmo resultado, e de valor igual ao símbolo transmitido. Caso ocorram erros, as equações vão dar resultados diferentes devendo-se utilizar um voto maioritário. Em caso de empate só se pode dizer que ocorreram mais erros do que os que se conseguem corrigir. O código de Reed-Muller é capaz de corrigir $\left(\frac{1}{2}2^{p-r}-1\right)$ erros e recuperar os K símbolos de informação. Do que foi descrito, um código RM de ordem r pode ser construído a partir de um de ordem $r-1$ e um código RM de ordem $r-1$ pode ser "extraído" de um de ordem r. Por este motivo se um código RM de ordem r contém o código de ordem $r-1$ e sua distância mínima não pode ser maior.

Vamos mostrar todos os sistemas de equações para descodificar o exemplo dado, começando por escrever o resultado da codificação (4.8) em função dos símbolos da mensagem m

$$\begin{aligned}
c_0 &= m_0\\
c_1 &= m_0+m_4\\
c_2 &= m_0+m_3\\
c_3 &= m_0+m_3+m_4+m_{10}\\
c_4 &= m_0+m_2+m_9\\
c_5 &= m_0+m_2+m_4\\
c_6 &= m_0+m_2+m_3+m_8+m_9\\
c_7 &= m_0+m_2+m_3+m_4+m_8+m_{10}
\end{aligned}$$

$$\begin{aligned}
c_8 &= m_0+m_1\\
c_9 &= m_0+m_1+m_4+m_7\\
c_{10} &= m_0+m_1+m_3+m_6\\
c_{11} &= m_0+m_1+m_3+m_4+m_6+m_7+m_{10}\\
c_{12} &= m_0+m_1+m_2+m_5\\
c_{13} &= m_0+m_1+m_2+m_4+m_5+m_7+m_9\\
c_{14} &= m_0+m_1+m_2+m_3+m_5+m_6+m_8\\
c_{15} &= m_0+m_1+\ldots+m_8+m_9+m_{10}.
\end{aligned}$$

Com alguma manipulação simples das equações anteriores, verifica-se que é possível escrever $m_{10}, m_9, m_8, m_7, m_6$ e m_5, em função de $c_k, k=0\ldots15$. Como $y=c+e$, temos então as seguintes equações para o símbolo m_{10}

$$\begin{aligned}
\tilde{m}_{10} &= y_0^{(2)}+y_1^{(2)}+y_2^{(2)}+y_3^{(2)}\\
\tilde{m}_{10} &= y_4^{(2)}+y_5^{(2)}+y_6^{(2)}+y_7^{(2)}\\
\tilde{m}_{10} &= y_8^{(2)}+y_9^{(2)}+y_{10}^{(2)}+y_{11}^{(2)}\\
\tilde{m}_{10} &= y_{12}^{(2)}+y_{13}^{(2)}+y_{14}^{(2)}+y_{15}^{(2)}
\end{aligned}$$

Se não ocorrerem erros teremos $y_j=c_j$, e no caso de símbolos binários, a primeira estimativa para m_{10} é dada pela soma módulo 2 dos termos $c_0=m_0, c_1=m_0+m_4, c_2=$

$m_0 + m_3$ e $c_3 = m_0 + m_3 + m_4 + m_{10}$. As restantes estimativas podem ser verificadas de forma idêntica. Repare-se que se apenas tiver ocorrido um erro, somente uma das estimativas anteriores estará errada, sendo possível por uma técnica de voto maioritário recuperar o valor correcto de m_{10}. O índice (2) associado à variável y, serve unicamente para lembrar a iteração em que se está e que o seu valor muda de iteração para iteração. No entanto para simplificar a notação poderemos omiti-lo desde que não haja risco de confusão. De seguida iremos enumerar todas as equações para os restantes símbolos.

$$\begin{array}{ll}
\tilde{m}_9 = y_0 + y_1 + y_4 + y_5 & \tilde{m}_8 = y_0 + y_2 + y_4 + y_6 \\
\tilde{m}_9 = y_2 + y_3 + y_6 + y_7 & \tilde{m}_8 = y_8 + y_{10} + y_{12} + y_{14} \\
\tilde{m}_9 = y_8 + y_9 + y_{12} + y_{13} & \tilde{m}_8 = y_1 + y_3 + y_5 + y_7 \\
\tilde{m}_9 = y_{10} + y_{11} + y_{14} + y_{15} & \tilde{m}_8 = y_9 + y_{11} + y_{13} + y_{15} \\
\tilde{m}_7 = y_0 + y_1 + y_8 + y_9 & \tilde{m}_6 = y_0 + y_2 + y_8 + y_{10} \\
\tilde{m}_7 = y_2 + y_3 + y_{10} + y_{11} & \tilde{m}_6 = y_1 + y_3 + y_9 + y_{11} \\
\tilde{m}_7 = y_4 + y_5 + y_{12} + y_{13} & \tilde{m}_6 = y_4 + y_6 + y_{12} + y_{14} \\
\tilde{m}_7 = y_6 + y_7 + y_{14} + y_{15} & \tilde{m}_6 = y_5 + y_7 + y_{13} + y_{15} \\
\tilde{m}_5 = y_1 + y_5 + y_8 + y_{12} & \\
\tilde{m}_5 = y_0 + y_4 + y_9 + y_{13} & \\
\tilde{m}_5 = y_2 + y_6 + y_{11} + y_{15} & \\
\tilde{m}_5 = y_3 + y_7 + y_{10} + y_{14} &
\end{array} \tag{4.10}$$

Se se obtiver sucesso a corrigir os símbolos dados nas equações anteriores, podemos reduzir a ordem do código recorrendo à equação (4.9) com $r = 2$,

$$y^{(1)} = y^{(2)} - G_2 M_2.$$

Os símbolos m_1, m_2, m_3 e m_4, podem ser agora recuperados por voto maioritário construindo equações de forma idêntica à realizada anteriormente, utilizando $y^{(1)}$

$$\begin{array}{ll}
\tilde{m}_4 = y_0 + y_1 & \tilde{m}_3 = y_0 + y_2 \\
\tilde{m}_4 = y_2 + y_3 & \tilde{m}_3 = y_1 + y_3 \\
\tilde{m}_4 = y_4 + y_5 & \tilde{m}_3 = y_4 + y_6 \\
\tilde{m}_4 = y_6 + y_7 & \tilde{m}_3 = y_5 + y_7 \\
\tilde{m}_2 = y_0 + y_4 & \tilde{m}_1 = y_0 + y_8 \\
\tilde{m}_2 = y_1 + y_5 & \tilde{m}_1 = y_1 + y_9 \\
\tilde{m}_2 = y_2 + y_6 & \tilde{m}_1 = y_2 + y_{10} \\
\tilde{m}_2 = y_3 + y_7 & \tilde{m}_1 = y_3 + y_{11}
\end{array} ,$$

e reduzindo novamente o código

$$y^{(0)} = y^{(1)} - G_1 M_1,$$

teremos finalmente, utilizando $y^{(0)}$

$$\begin{aligned}
\tilde{m}_0 &= y_0 \\
\tilde{m}_0 &= y_1 \\
\tilde{m}_0 &= y_2 \\
\tilde{m}_0 &= y_3.
\end{aligned}$$

Vamos agora construir o código RM sobre o corpo dos reais. Como o cálculo numérico em $\mathbb{R}$ é realizado com uma representação finita para os números, ocorrem sempre erros de arredondamento, pelo que não podemos pretender que a diferença entre as diferentes estimativas das equações anteriores seja exactamente zero quando não ocorrem erros. É assim necessário fixar um limiar apropriado ϵ, considerando-se que duas estimativas para o valor verdadeiro de um símbolo são válidas se a diferença entre elas for inferior a ϵ. Uma vez que estão apenas envolvidas somas e subtrações, estes códigos são fáceis de implementar e não apresentam grandes problemas com os erros de arredondamento. Possuem no entanto a desvantagem de não existirem para todos os N. No artigo [Shiu 96] é proposta uma alteração da matriz de codificação G de modo a que o mesmo algoritmo de descodificação utilizado para $GF(p)$ possa ser utilizado sem alterações. No entanto, constatou-se que existem alguns problemas com este processo sendo sugerida uma nova forma de calcular a matriz G. Serão igualmente apresentadas todas as equações de descodificação assim como os resultados de algumas simulações.

A regra proposta para a construção da matriz G é a seguinte:

Proposição 2 *Considere-se uma matriz de codificação G de um dado código de Reed-Muller em $GF(p)$. Altere-se a primeira linha de modo a que os sinais apareçam alternados. De seguida propague-se os sinais dos elementos da primeira linha em cada uma das colunas da matriz.*

Vamos dar um exemplo de um código Reed-Muller real do tipo (16,11). Seguindo a regra da definição anterior teremos para a matriz de codificação

$$G_0 = \begin{bmatrix} 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}^T$$

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

$$G_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 & -1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 \end{bmatrix}^T$$

Relembrando que a codificação se faz por meio da equação (4.8), podemos escrever o seguinte conjunto de equações

$$
\begin{aligned}
c_0 &= m_0 \\
c_1 &= -m_0 + m_4 \\
c_2 &= m_0 + m_3 \\
c_3 &= -m_0 - m_3 - m_4 - m_{10} \\
c_4 &= m_0 + m_2 + m_9 \\
c_5 &= -m_0 - m_2 - m_4 \\
c_6 &= m_0 + m_2 + m_3 + m_8 + m_9 \\
c_7 &= -m_0 - m_2 - m_3 - m_4 - m_8 - m_{10} \\
c_8 &= m_0 + m_1 \\
c_9 &= -m_0 - m_1 - m_4 - m_7 \\
c_{10} &= m_0 + m_1 + m_3 + m_6 \\
c_{11} &= -m_0 - m_1 - m_3 - m_4 - m_6 - m_7 - m_{10} \\
c_{12} &= m_0 + m_1 + m_2 + m_5 \\
c_{13} &= -m_0 - m_1 - m_2 - m_4 - m_5 - m_7 - m_9 \\
c_{14} &= m_0 + m_1 + m_2 + m_3 + m_5 + m_6 + m_8 \\
c_{15} &= -m_0 - m_1 - \ldots - m_8 - m_9 - m_{10}.
\end{aligned}
\tag{4.11}
$$

O sinal recebido é dado por $y = c + e$, em que e é o sinal de erro. A partir das expressões anteriores podemos obter conjuntos de 4 equações que dão uma estimativa para os símbolos do sinal mensagem:

$$
\begin{aligned}
&\tilde{m}_{10} = y_0 + y_1 - y_2 - y_3 \qquad \tilde{m}_9 = -y_0 - y_1 + y_4 + y_5 \\
&\tilde{m}_{10} = y_4 + y_5 - y_6 - y_7 \qquad \tilde{m}_9 = y_8 + y_9 - y_{12} - y_{13} \\
&\tilde{m}_{10} = y_8 + y_9 - y_{10} - y_{11} \qquad \tilde{m}_9 = -y_2 - y_3 + y_6 + y_7 \\
&\tilde{m}_{10} = y_{12} + y_{13} - y_{14} - y_{15} \qquad \tilde{m}_9 = y_{10} + y_{11} - y_{14} - y_{15} \\
&\tilde{m}_8 = y_0 - y_2 - y_4 + y_6 \qquad \tilde{m}_7 = y_0 + y_1 - y_8 - y_9 \\
&\tilde{m}_8 = y_8 - y_{10} - y_{12} + y_{14} \qquad \tilde{m}_7 = y_2 + y_3 - y_{10} - y_{11} \\
&\tilde{m}_8 = -y_1 + y_3 + y_5 - y_7 \qquad \tilde{m}_7 = y_4 + y_5 - y_{12} - y_{13} - 2\tilde{m}_9 \\
&\tilde{m}_8 = -y_9 + y_{11} + y_{13} - y_{15} \qquad \tilde{m}_7 = y_6 + y_7 - y_{14} - y_{15} - 2\tilde{m}_9 \\
&\tilde{m}_6 = y_0 - y_2 - y_8 + y_{10} \qquad \tilde{m}_5 = -y_1 + y_5 - y_8 + y_{12} \\
&\tilde{m}_6 = -y_1 + y_3 + y_9 - y_{11} \qquad \tilde{m}_5 = y_0 - y_4 + y_9 - y_{13} \\
&\tilde{m}_6 = y_4 - y_6 - y_{12} + y_{14} \qquad \tilde{m}_5 = y_2 - y_6 + y_{11} - y_{15} \\
&\tilde{m}_6 = -y_5 + y_7 + y_{13} - y_{15} \qquad \tilde{m}_5 = -y_3 + y_7 - y_{10} + y_{14}
\end{aligned}
$$

Repare-se na semelhança entre o conjunto de equações anteriores e as dadas em (4.10) em que apenas os sinais mudam. Além disso duas das estimativas para os símbolos m_7 dependem da correcta descodificação de m_9. Esta diferença não é problemática uma vez que se ocorrerem mais erros do que o código consegue descodificar o método pode não conseguir descodificar qualquer um dos símbolos de m_{10} a m_5. Reduzindo a ordem do código e utilizando $y^{(1)}$ podemos obter as estimativas para os símbolos m_4, m_3, m_2 e m_1

$$\begin{array}{ll}\tilde{m}_4 = -y_0 - y_1 & \tilde{m}_3 = -y_0 + y_2 \\ \tilde{m}_4 = -y_2 - y_3 & \tilde{m}_3 = y_1 - y_3 \\ \tilde{m}_4 = -y_4 - y_5 & \tilde{m}_3 = -y_4 + y_6 \\ \tilde{m}_4 = -y_6 - y_7 & \tilde{m}_3 = y_5 - y_7 \\ \tilde{m}_2 = -y_0 + y_4 & \tilde{m}_1 = -y_0 + y_8 \\ \tilde{m}_2 = y_1 - y_5 & \tilde{m}_1 = y_1 - y_9 \\ \tilde{m}_2 = -y_2 + y_6 & \tilde{m}_1 = -y_2 + y_{10} \\ \tilde{m}_2 = y_3 - y_7 & \tilde{m}_1 = y_3 - y_{11}\end{array}$$

e finalmente voltando a reduzir uma vez mais a ordem do código teremos a estimativas para m_0 utilizando $y^{(0)}$

$$\begin{array}{c}\tilde{m}_0 = y_0 \\ \tilde{m}_0 = -y_1 \\ \tilde{m}_0 = y_2 \\ \tilde{m}_0 = -y_3\end{array}.$$

Repare-se que o número de equações utilizadas para estimar os símbolos m_4, m_3, m_2 e m_1, podia ter sido de 8, e no caso de m_0, de 16. Como a distância mínima dos códigos RM resultantes da redução de outros de ordem superior não podem ter uma distância mínima superior, não faz sentido utilizar mais do que quatro equações para realizar o voto maioritário.

Como qualquer implementação prática deste código passa por uma representação binária dos simbolos vamos realizar uma breve análise dos erros. Considerando que cada símbolo é representado com b bits, o pior caso acontece para o elemento c_{N-1} do código, pois como se pode ver na última equação de (4.11) o seu valor é igual á soma dos elementos de m. Para evitar "overflow" e permitir uma recuperação sem erros de arredondamento serão necessários $\log_2 K + b$ bits para representar c_{N-1}. Se utilizarmos esta representação para os restantes elementos de c, teremos uma eficiência $\mathcal{E}$ do código de

$$\mathcal{E} = \frac{bK}{(\log_2 K + b)\, N}.$$

Uma observação mais atenta das equações (4.11) permite concluir que cada um dos elementos de c é a soma de um subconjunto de elementos de m, e por esse motivo pode ter um número inferior de bits para a sua representação. No caso concreto do código real $RM\,(16, 11)$ apresentado podemos ter o seguinte número de bits extra para cada elemento de c

Elem.	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_{11}	c_{12}	c_{13}	c_{14}	c_{15}
nbits	0	1	1	2	2	2	3	3	1	2	2	3	2	3	3	4

em que o número de bits dado na segunda linha pode ser calculado pela expressão

$$nbits\,(c_i) = \left\lceil \log_2 \left(\sum_{j=0}^{K-1} |G_{ji}| \right) \right\rceil.$$

Esta redução do número de bits por símbolo permite melhorar a eficiência dos códigos reais de RM e no caso concreto do código descrito teremos para uma representação com 8 bits dos elementos de c uma eficiência de

$$\mathcal{E} = \frac{8 \times 11}{(\log_2 11 + 8)\, 16} = 0,46$$

No caso de termos um número de bits diferente para cada c_i a eficiência do código é

$$\mathcal{E} = 0,54.$$

Código linear com a DCT

É possível construir um código linear utilizando como matriz G de codificação uma submatriz de uma matriz unitária. Um desses casos é a DCT (Discrete Cosine Transform) bastante utilizada em codificadores de áudio e vídeo devido às suas propriedades de compactação da energia de um sinal. No entanto, esta transformada não permite a construção de um código cíclico e por esse motivo não se podem utilizar directamente os algoritmos rápidos disponíveis para os códigos cíclicos. A razão para esta dificuldade deve-se ao facto de a matriz de descodificação H não ser Vandermonde. Ja-Ling Wu propõe em [Wu 95] um algoritmo de descodificação que converte a matriz de descodificação H do código com a DCT numa matriz Vandermonde podendo-se depois aplicar a esta matriz as técnicas utilizadas no capítulo 2 para os códigos cíclicos reais do tipo BCH.

Definição 8 *A transformada DCT de um sinal* $x = [x_0, x_1, \cdots, x_{N-1}]^T$ *é dada por*

$$\hat{x} = Fx$$

em que os elementos da matriz são dados por

$$F_{kn} = \begin{cases} \sqrt{\frac{1}{N}} & k = 0, \\ \sqrt{\frac{2}{N}} \cos \frac{(2n+1)k\pi}{2N} & k = 1, 2, \ldots, N-1. \end{cases}$$

Tal como foi feito no caso dos códigos BCH (subsecção 4.1.6), obtem-se a matriz de codificação $G_{(N\times K)}$ a partir de K colunas de F, resultando numa matriz de paridade $H_{(N\times(N-K))}$ com as restantes $N-K$ colunas de F. Como F é unitária, as seguintes equações são verificadas

$$\begin{aligned} H^\dagger G &= 0 \\ G^\dagger G &= I_K. \end{aligned}$$

O sindroma do sinal recebido y é dado por

$$\begin{aligned} s &= H^\dagger y \\ &= H^\dagger (c + e) \\ &= H^\dagger e. \end{aligned}$$

Definindo

$$s'_i = \frac{1}{f_i} s_i = \sum_{r=0}^{N-1} e_r \cos \frac{(2r+1)\, i\pi}{2N} \quad i = 0, 1, \ldots (N-K) - 1 \tag{4.12}$$

com $f_i = \sqrt{1/N}$ para $i = 0$, e $f_i = \sqrt{2/N}$ para $i = 1, 2, \ldots, N-1$, pode-se escrever a equação (4.12) como

$$s' = Ps,$$

em que a matriz P é diagonal com com elementos f_i.

Antes de prosseguir, consideremos a seguinte expansão de $\cos(k\omega)$

$$\cos kw = \sum_{n=0}^{k} C_{k,n} (\cos w)^n$$

onde

$$C_{k,n} = \begin{cases} 0 & k-n \quad \text{ímpar} \\ (-1)^{(k-n)/2} \sum_{m=0}^{\lfloor \frac{n}{2} \rfloor} \binom{k}{n-2m} \binom{\frac{k-n}{2}+m}{m} & k-n \quad \text{par.} \end{cases}$$

Se ocorrerem t erros em posições $\bar{S}_t = \{i_0, i_1, \ldots, i_{t-1}\}$, então, utilizando a expressão anterior na equação (4.12), teremos

$$\begin{aligned} s' &= e\left(\bar{S}_t\right) \begin{bmatrix} 1 & X_1 & X_1^2 & \cdots & X_1^{d-1} \\ 1 & X_2 & X_2^2 & \cdots & X_2^{d-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & X_t & X_t^2 & \cdots & X_t^{d-1} \end{bmatrix} \times \\ &\times \begin{bmatrix} C_{0,0} & 0 & C_{2,0} & 0 & \cdots & . \\ 0 & C_{1,1} & 0 & C_{3,1} & \cdots & . \\ 0 & 0 & C_{2,2} & 0 & \cdots & . \\ 0 & 0 & 0 & C_{3,3} & \cdots & . \\ \vdots & \vdots & \vdots & \vdots & \ddots & . \\ 0 & 0 & 0 & 0 & \cdots & C_{d-1,d-1} \end{bmatrix} \end{aligned}$$

ou

$$s' = uXC,$$

com

$$u = e\left(\bar{S}_t\right).$$

Como a matriz C é sempre invertível podemos obter o seguinte sindroma

$$s'' = s'C^{-1} = uX$$

e como a matriz X é Vandermonde podemos aplicar a técnica de descodificação dos códigos BCH e obter o vector com os erros u.

4.1.5 Códigos cíclicos em $\mathbb{C}$

Os códigos cíclicos são uma subclasse dos códigos lineares e podem-se definir do seguinte modo

Definição 9 *Se num código linear em $\mathbb{C}^N$, $C(z) = (c_0, c_1, \ldots, c_{N-1})$ é uma palavra pertencente ao subespaço ϑ código, e se a palavra $C' = (c_{N-1}, c_0, c_1, \ldots, c_{N-2})$, pertence igualmente a ϑ, então o código diz-se cíclico.*

A maior estrutura destes códigos permite evitar as técnicas tabulares utilizadas nos códigos lineares, possibilitando a utilização de algoritmos mais eficientes.

Descrição polinomial dos códigos cíclicos em $\mathbb{C}$

Por razões históricas os códigos de correcção de erros foram desenvolvidos recorrendo a uma descrição polinomial. Nesta secção pretende-se enunciar alguns resultados dos códigos cíclicos no corpo $\mathbb{C}$ utilizando esta descrição.

Pode-se descrever uma palavra de um código cíclico como um polinómio em z

$$C(z) = \sum_{k=0}^{N-1} c_k z^k, \tag{4.13}$$

em que $C(z)$ satisfaz a definição 9. Cada palavra do código é um vector com N componentes que são os coeficientes do polinómio e por isso os polinómios não têm potências superiores a $N-1$. O conjunto destes polinómios forma um anel com o producto definido por

$$C_1(z) \cdot C_2(z) = R_{z^N-1}\left[C_1(z) C_2(z)\right]$$

em que

$$R_{z^N-1}\left[C(z)\right]$$

indica o resto da divisão de $C(z)$ por $z^N - 1$. O teorema 5.2.3 de [Blahut 83], estabelece:

Teorema 9 *Existe um código cíclico de dimensão N com polinómio gerador $G(z)$ se e só se $G(z)$ divive $z^N - 1$.*

Este teorema implica que para qualquer código cíclico com polinómio gerador $G(z)$ temos

$$z^N - 1 = G(z) H(z)$$

em que o polinómio $H(z)$ é o polinómio verificador de paridade em terminologia de TCCE.

Se a aritmética utilizada for limitada ao corpo dos reais, $z^N - 1$ tem apenas dois zeros, $+1$ e -1, e por conseguinte a dimensão máxima do bloco é de $N = 2$. Por outro lado se se trabalhar no corpo dos complexos, N pode ser qualquer inteiro positivo e as raízes complexas vão ser do tipo

$$z_m = e^{\mathrm{j}\frac{2\pi}{N}m}. \tag{4.14}$$

Estas são as N raízes da unidade de ordem N, e por definição vamos ter K em $G(z)$ e $M = N - K$ em $H(z)$. Deste modo podemos escrever o polinómio $z^N - 1$ como um produtório

$$z^N - 1 = \prod_{m=0}^{N-1} (z - z_m) = \prod_{m=0}^{N-1} \left(z - e^{\mathrm{j}\frac{2\pi}{N}m}\right).$$

Consideremos agora o polinómio $M(z)$, que representa a mensagem a codificar e a operação de codificação

$$C(z) = M(z) \cdot G(z)$$

o resultado da codificação. Se $E(z)$ for o polinómio associado ao sinal de erro com t coeficientes diferentes de zero, temos

$$V(z) = C(z) + E(z),$$

e calculando o resto da divisão polinimial de $V(z)$ por $G(z)$ obtemos o *polinómio sindroma*

$$\begin{aligned} S(z) &= R_{G(Z)}[V(z)] = R_{G(Z)}[C(z) + E(z)] \\ &= R_{G(Z)}[E(z)], \end{aligned}$$

que como se pode constatar pela última equação só depende do polinómio associado ao sinal de erro $E(z)$.

Repare-se que qualquer polinómio $C(z)$ pertence ao código ϑ se $C(z_k) = 0$ sendo z_k os zeros do polinómio $G(z)$, tal como estabelece o seguinte teorema.

Teorema 10 *Seja $G(z)$ o polinómio gerador de um código de correcção de erros em $\mathbb{C}$ com zeros $z_p = e^{\mathrm{j}\frac{2\pi}{N}p}$, em que $p \in \{0, \ldots, K-1\}$. Então, um polinómio $C(z)$ é uma palavra pertencente ao código se e só se*

$$C(z_0) = C(z_1) = \cdots = C(z_{N-1}) = 0.$$

Demonstração. *Como $C(z) = M(z) \cdot G(z)$, então $C(z_p) = M(z_p) \cdot G(z_p) = 0$. Por outro lado se $C(z_p) = 0$ e escrevermos*

$$C(z) = Q(z) \cdot (z - z_p) + S(z),$$

então

$$0 = C(z_p) = Q(z_p) \cdot (z_p - z_p) + S(z_p)$$

leva a que $S(z_p) = 0$. Pode-se então concluir que $C(z)$ é divisível por qualquer dos K factores $(z - z_p)$, de $G(z)$. ■

Descrição matricial dos códigos cíclicos em $\mathbb{C}$

Os códigos cíclicos no corpo dos complexos podem ser descritos na forma matricial recorrendo à equação (4.13) e ao teorema 10, que juntos conduzem a

$$\sum_{k=0}^{N-1} c_k z_p^k = \sum_{k=0}^{N-1} c_k e^{\mathrm{j}\frac{2\pi}{N}kp} = 0 \qquad p \in S_f.$$

Esta última equação pode ser colocada na forma matricial

$$\begin{bmatrix} z_{p_0}^0 & z_{p_0}^1 & \cdots & z_{p_0}^{N-1} \\ z_{p_1}^0 & z_{p1}^1 & \cdots & z_{p_1}^{N-1} \\ z_{p_2}^0 & z_{p_2}^1 & \cdots & z_{p_2}^{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ z_{p_{M-1}}^0 & z_{p_1}^1 & \cdots & z_{p_{M-1}}^{N-1} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{N-1} \end{bmatrix} = 0,$$

$$H^T c = 0.$$

Podemos portanto a partir de M raízes da unidade de ordem N, z_{p_m}, construir um código cíclico com a matriz H dada pela equação anterior. A matriz geradora do código será construída com as restantes K raízes que não foram utilizadas para construir a matriz H.

Os códigos cíclicos podem ter as raízes do polinómio gerador G dispostos de forma arbitrária não dando em geral códigos com distância de Hamming máxima. Se atendermos à subsecção 2.2.1, podemos verificar facilmente que se podem precisar de $t \times (t+1)$ elementos conhecidos (nulos) do espectro do sinal (sindroma) para corrigir t erros.

4.1.6 Códigos cíclicos do tipo Bose-Chaudhuri-Hocquenghem e Reed-Solomon

Os códigos BCH representam uma classe de códigos lineares que permitem corrigir vários erros. No capítulo 2 é descrita a utilização dos códigos BCH no corpo dos números complexos $\mathbb{C}$, indicando as técnicas de codificação e descodificação e algoritmos rápidos para a sua realização prática. Nesta secção iremos indicar apenas as condições a que um código deve obedecer para ser considerado do tipo BCH e demonstrar a equivalência entre a utilização de raízes não contíguas para o polinómio gerador e o entrelaçamento regular. Os códigos BCH são lineares e cíclicos e para serem classificados como BCH é necessário que as raízes do polinómio codificador sejam do tipo

$$z_m = e^{\mathrm{j}\frac{2\pi}{N}lm}, \tag{4.15}$$

em que l e N são primos relativos [Blahut 83, Kumaresan 85]. A possibilidade mais simples é tomar $l = 1$, correspondendo a raízes contíguas. Esta condição para a escolha das raízes do polinómio codificador equivale a escolher para matriz de codificação K linhas contíguas da matriz de Fourier. Com esta escolha para as raízes, garante-se que o código é do tipo CDM, ou seja, para $M = 2t$ conseguem-se corrigir t erros. Por vezes os códigos BCH permitem corrigir mais do que t erros,e por esse motivo a distância $d = 2t + 1$ entre símbolos é designada por distância de projecto do código, podendo a distância efectiva ser maior. Em [Wolf 83] são utilizadas algumas técnicas de voto que descreveremos na subsecção seguinte e que permitem em certas circunstâncias descodificar correctamente mais do que t erros.

Como foi afirmado no início desta subsecção, pode-se demonstrar que quando $l \neq 1$, a codificação é equivalente à realizada com $l = 1$ seguida de entrelaçamento regular do sinal codificado. A operação de entrelaçamento pode ser definida do seguinte modo

Definição 10 *O entrelaçamento de um vector $x \in \mathbb{C}^N$ é uma permutação $y = Px$, com P uma matriz de permutação.*

O entrelaçamento regular pode por sua vez ser definido como

Definição 11 *Se $c(i)$ for a coluna que possui o elemento unitário correspondente à linha i numa matriz de permutação P, então, um entrelaçamento diz-se regular se $c(i) = c(li \bmod N)$, com l e N primos relativos e $i \in \{0, \ldots, N-1\}$.*

Considere-se P uma matriz de permutação [Horn 85], e F a matriz de Fourier tal como foi definida em (1.2). Se compararmos os elementos da matriz anterior com as raízes em (4.15), verificamos que correspondem ao caso mais simples em que $l = 1$. Quando $l \neq 1$ e l e N são primos relativos, as colunas da matriz de Fourier são as mesmas mas dispostas por ordem diferente. Tal deriva do facto de $\phi = e^{\mathrm{j}\frac{2\pi}{N}kl}$, ser uma raíz da unidade se l e N forem primos relativos e de as potências sucessivas de ϕ, $\left\{\phi^0, \phi^1, \phi^2, \ldots, \phi^{N-1}\right\}$, gerarem todas as raízes de ordem N da unidade. Se quando $l \neq 1$ chamarmos à matriz de Fourier correspondente $\breve{F}$ teremos

$$\breve{F} = PF,$$

em que P é uma matriz de permutação.

Por outro lado, a matriz de codificação G $(N \times K)$ e a matriz de paridade H $(N \times M)$, são submatrizes da matriz de Fourier F

$$F = \left[\begin{array}{cc} G & H \end{array}\right],$$

podendo-se escrever

$$\breve{F} == \left[\begin{array}{cc} \breve{G} & \breve{H} \end{array}\right] = P \left[\begin{array}{cc} G & H \end{array}\right] = \left[\begin{array}{cc} PG & PH \end{array}\right].$$

O sinal codificado por esta transformada é dado por

$$\breve{c} = \breve{G}m = PGm,$$

que depois de sofrer alguns erros virá dado por

$$\breve{y} = \breve{c} + \breve{e},$$

e que se pode descodificar como

$$\breve{H}^{\dagger}\breve{y} = (PH)^{\dagger}(\breve{c} + \breve{e}) = H^{\dagger}P^{\dagger}(PGm + \breve{e}) \tag{4.16}$$

$$\breve{H}^{\dagger}\breve{y} = H^{\dagger}P^{T}PGm + H^{\dagger}P^{T}\breve{e}. \tag{4.17}$$

Se atendermos ao facto de que $P^T P = I$ e que $H^{\dagger}G = 0$, a diferença principal entre as duas codificações consiste na permutação por $P^{\dagger}$ da posição dos erros do vector $\breve{e}$. Assim, se ocorrem erros em posições contíguas durante a transmissão, a utilização de uma transformada com $l \neq 1$ faz com que o sistema veja estes erros como esparsos. Um sistema de codificação com a transformada $\breve{F}$ é equivalente a um sistema que utilize a transformada F realizando entrelaçamento P antes de transmitir a palavra codificada e desentrelaçamento por P^T na palavra recebida

$$\begin{aligned} \breve{c} &= Pc \\ y &= P^{T}\breve{y}. \end{aligned}$$

4.1.7 Correcção de mais do que $2t$ erros

Os códigos BCH podem corrigir mais do que t erros quando $M = 2t$, sendo o limite máximo dado pelo limite de Singleton

$$t \leq N - K - 1. \tag{4.18}$$

No artigo [Wolf 83], Wolf apresenta uma técnica de voto que permite descodificar mais do que t erros quando $M = 2t$. Para cada uma das $\binom{N}{K}$maneiras de escolher K amostras do sinal recebido reconstrói-se uma sequência com K amostras que é candidata a mensagem, desde que o sindroma seja inferior a um dado limiar $\epsilon \approx 0$. Admitindo que o ruído de arredondamento é pequeno, podemos comparar as sequências recebidas escolhendo as que obtiverem mais votos. No caso de se verificar a condição (4.18), então a sequência original receberá pelo menos $K + 1$ votos.

Exemplo

Consideremos um código cíclico com $N = 8$ e $K = 4$, caso em que (4.18) conduz a $t \leq 3$. Vamos supor que os erros aconteceram nas amostras de índice $0, 1$ e 2, então vamos ter tal como previsto $K + 1 = 5$ padrões

$$\{3, 4, 5, 6\}$$
$$\{3, 4, 5, 7\}$$
$$\{3, 4, 6, 7\}$$
$$\{3, 5, 6, 7\}$$
$$\{4, 5, 6, 7\}$$

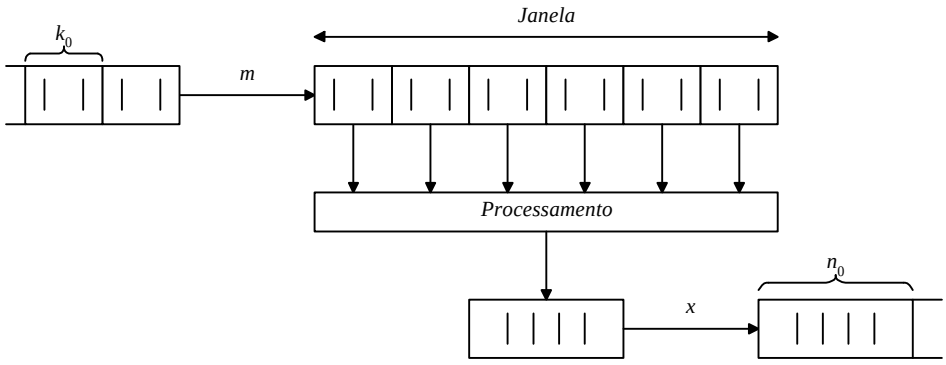


Figura 4.2: Codificador convolucional com uma relação $r = \frac{k_0}{n_0}$. A entrada é dividida em pequenos segmentos de k_0 amostras sendo guardados v desses segmentos. O bloco de processamento realiza uma combinação linear utilizando os segmentos guardados e gera uma saída com n_0 amostras em que $n_0 > k_0$.

que resultam na mesma palavra descodificada, dado que não incluem amostras com erro.

Este método de descodificação, apesar de funcionar e ser de fácil realização prática, só é aplicável para blocos de pequena dimensão, pois o número $\binom{N}{K}$ de combinações a testar cresce rapidamente. Outra situação em que pode ser útil é para os casos em que o tempo de processamento é menos importante do que a correcta descodificação dos dados. Neste caso, podemos combinar este algoritmo com a descodificação utilizada no capítulo 2 dos códigos BCH e só usar a técnica de voto quando a primeira falha.

A robustez da técnica de voto ao ruído de quantificação originado quando se transmite o sinal deve ser objecto de estudo para averiguar com rigor a real vantagem da sua utilização.

4.2 Códigos convolucionais no corpo dos complexos

Nesta secção relacionamos a teoria dos códigos convolucionais com a teoria desenvolvida recentemente sobre bancos de filtros e "wavelets". Existem poucos trabalhos sobre a utilização de códigos convolucionais com aritmética real. Marshall apresenta em [Marshall Jr 84] alguns algoritmos de codificação e descodificação em códigos convolucionais mas sem contudo oferecer uma forma sistemática de projecto e análise.

Nesta secção faremos a ligação entre alguns resultados obtidos na teoria dos bancos de filtros e os códigos convolucionais. No final será apresentado um exemplo de um descodificador que utiliza um banco de filtros com duas bandas.

4.2.1 Códigos convolucionais

Os códigos convolucionais realizam a operação de codificação sem dividir o sinal a codificar em blocos. Na figura 4.2 podemos observar como este tipo de código realiza a operação de codificação.

Como se pode observar o sistema possui uma linha de atraso onde armazena amostras passadas do sinal de entrada. Este é dividido em pequenos segmentos com uma ou mais amostras (k_0) e realiza uma combinação linear de v segmentos do passado do sinal com o actual e gera um segmento de saída com n_0 amostras. Para que haja redundância é necessário

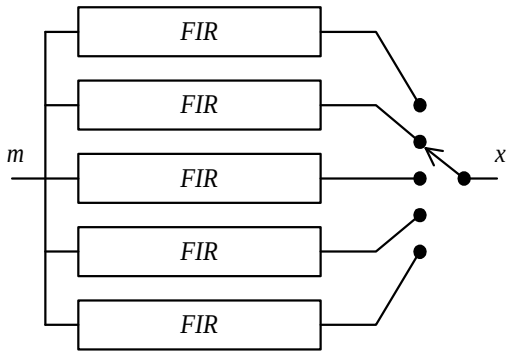


Figura 4.3: Código convolucional do tipo $(5, 1)$ em que para cada amostra colocada são geradas 5 na saída.

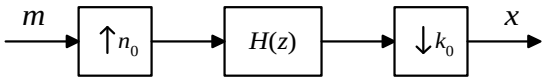


Figura 4.4: Sistema de alteração da frequência de amostragem por um factor de $r = \frac{k_0}{n_0}$. Este tipo de sistema é equivalente ao descrito na figura 4.2.

que $n_0 > k_0$, e na terminologia da TCCE costuma-se chamar a v "constraint length". Este parâmetro é de grande importância na descrição de um código convolucional, sendo um múltiplo de k_0, ou seja $v = mk_0$. Os códigos convolucionais com um dado k_0 e n_0 são designados de forma compacta por códigos (n_0, k_0).

Na figura 4.3 podemos observar um exemplo de um código convolucional do tipo $(5, 1)$, em que para cada amostra na entrada são geradas 5 na saída. A combinação linear das amostras passadas do sinal de entrada é realizada por um filtro FIR.

Convém realçar que as técnicas de descodificação de Viterbi não podem ser utilizadas no caso dos códigos com aritmética real, uma vez que elas se baseiam no facto de o alfabeto da aritmética utilizada ser finito.

4.2.2 Bancos de filtros

Os códigos covolucionais descritos são facilmente identificados como sistemas de processamento de sinal para alteração da frequência de amostragem, sendo a relação da mudança dada por

$$r = \frac{n_0}{k_0}.$$

Existem duas formas equivalentes de encarar estes sistemas: como um sistema composto por um filtro FIR com um interpolador na entrada e um decimador na saída ou como um banco de filtros. Um sistema de alteração da frequência de amostragem baseado no filtro FIR pode ser realizado tal como se pode ver na figura 4.4.

Este sistema acrescenta redundância de forma controlada, sendo possível a sua utilização para a detecção de ruído impulsivo. No entanto vamos focar a nossa atenção nos bancos filtros e suas propriedades.

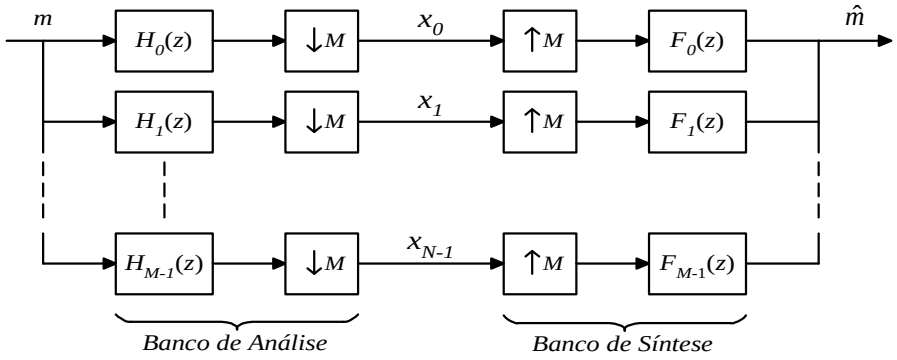


Figura 4.5: Banco de filtros com M bandas e decimação máxima igual ao número de bandas. Se os filtros $H_i(z)$ e $F_i(z)$ possuirem a propriedade de reconstrução perfeita a saída $\hat{m}$ será igual à entrada m.

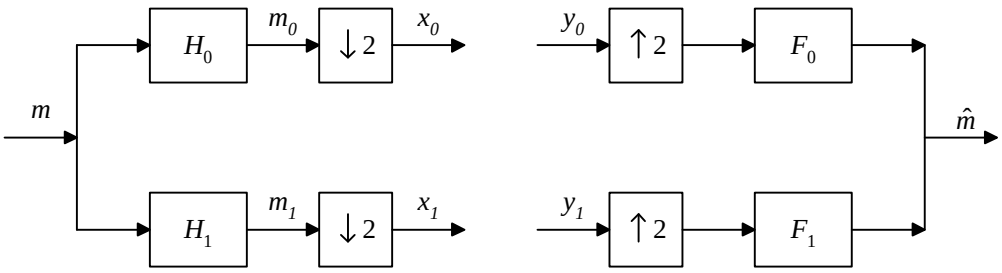


Figura 4.6: Banco de filtros com apenas duas bandas. Neste caso os sinais de cada banda são decimados à taxa máxima, não se registando alteração na taxa de amostragem total ao longo do sistema. Se os filtros satisfizerem as condições de reconstrução perfeita, o sinal de saída $\hat{m}$ será um versão atrasada do sinal de entrada m.

Os bancos de filtros apresentam normalmente uma estrutura semelhante à apresentada na figura 4.5. São constituídos por duas partes: uma de análise e outra de síntese. No banco de análise o sinal de entrada é dividido em sub-bandas e com decimação máxima em cada uma delas. Deste modo não é acrescentada qualquer redundância ao sinal de entrada mas irão aparecer componentes de "aliasing". Estas componentes de "aliasing" podem ser anuladas no banco de síntese se o sistema possuir a propriedade de reconstrução perfeita [Vaidyanathan 93, Strang 96].

Parece evidente que se a decimação introduzida não for máxima se estará a acrescentar algum grau de redundânica ao sinal, sendo o número de amostras por unidade de tempo à saída do banco de análise maior que à entrada, conseguindo-se assim o objectivo de aumentar a frequência de amostragem do sinal de entrada.

Para simplificar a análise deste tipo de bancos de filtros vamos considerar o caso mais simples de apenas duas bandas tal como se pode ver na figura 4.6.

Para que o sistema possua a propriedade de cancelamento do "aliasing", a seguinte equação deve ser satisfeita [Vaidyanathan 93, Strang 96]

$$H_0(-z)\,F_0(z) + H_1(-z)\,F_1(z) = 0. \tag{4.19}$$

Para conseguir este objectivo basta que os filtros do banco de síntese obedeçam às seguintes condições

$$F_0(z) = H_1(-z) \tag{4.20}$$
$$F_1(z) = -H_0(-z). \tag{4.21}$$

Por outro lado para não haja distorção do sinal de entrada o sistema deve satisfazer a seguinte equação

$$H_0(z) F_0(z) + H_1(z) F_1(z) = 2z^{-L}. \tag{4.22}$$

Uma vez que os filtros $F_0(z)$ e $F_1(z)$ podem ser obtidos a partir de $H_0(z)$ e $H_1(z)$, se se encontrar um filtro $H_0(z)$ adequado e uma forma de deduzir $H_1(z)$ de $H_0(z)$ de modo a que a equação (4.22) seja observada, temos o problema solucionado. Vamos definir o filtro

$$P(z) = F_0(z) H_0(z), \tag{4.23}$$

assim, quando as equações (4.20, 4.21) são satisfeitas e fazendo uso da equação anterior, podemos escrever (4.22) na forma mais compacta

$$P(z) - P(-z) = 2z^{-L}.$$

Para que o sistema não introduza distorção é necessário que o filtro $P(z)$, seja do tipo "meia banda". Uma das formas de o conseguir é escolher o filtro $H_1(z)$ do seguinte modo

$$H_1(z) = -z^{-N} H_0\left(-z^{-1}\right),$$

dando origem a um banco de filtros ortogonal em que $P(z)$ será uma aproximação de Butterword se $H_0(z)$ for um filtro de Daubechies [Strang 96].

Com esta escolha para os filtros teremos um banco de filtros com a propriedade de reconstrução perfeita. As equações (4.19) e (4.22) podem ser colocadas na forma matricial mais compacta

$$\begin{bmatrix} H_0(z) & H_1(z) \\ H_0(-z) & H_1(-z) \end{bmatrix} \begin{bmatrix} F_0(z) \\ F_1(z) \end{bmatrix} = \begin{bmatrix} 2z^{-L} \\ 0 \end{bmatrix},$$

e que pode ser expandida para

$$\begin{bmatrix} H_0(z) & H_1(z) \\ H_0(-z) & H_1(-z) \end{bmatrix} \begin{bmatrix} F_0(z) & F_0(-z) \\ F_1(z) & F_1(-z) \end{bmatrix} = \begin{bmatrix} 2z^{-L} & 0 \\ 0 & 2(-z)^{-L} \end{bmatrix}. \tag{4.24}$$

Se no sistema da figura 4.6 retirarmos os decimadores e interpoladores, na saída do codificador teremos o dobro das amostras do sinal original. Contudo, uma vez que este sistema satisfaz a propriedade da reconstrução perfeita basta que cheguem ao banco de síntese metade das amostras para se conseguir reconstruir o sinal original. Com base na equação (4.24) construímos o sistema de codificação e descodificação da figura 4.7 em que se inclui o sinal de erro e um novo banco de filtros de síntese que calcula o sindroma s.

O sinal de saída r será igual ao sinal mensagem m desde que $e_i = 0$. Por outro lado, o sinal s será sempre nulo sendo portanto apenas função do sinal de erro e. Por esta razão,

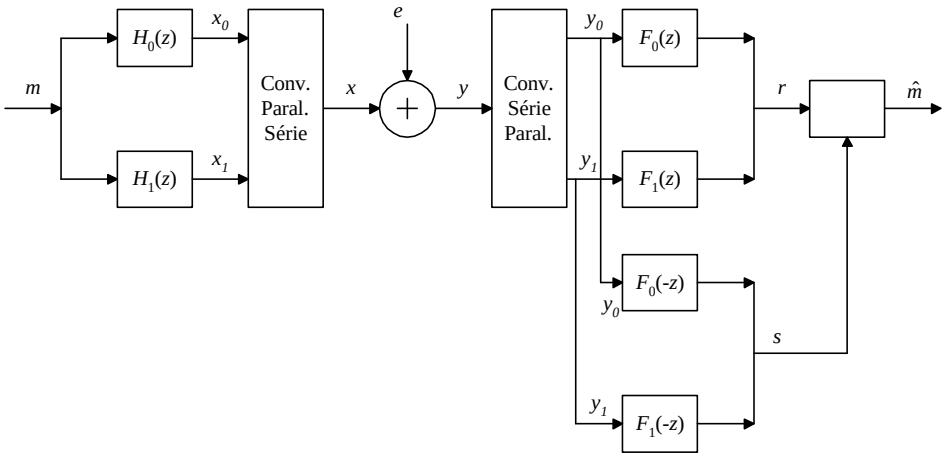


Figura 4.7: Diagrama de blocos de um sistema de correcção de erros utilizando bancos de filtros. O sistema possui a propriedade de reconstrução perfeita para a saída r, e $y_i = x_i$ se $e = 0$. A saída s é nula desde que não ocorram erros. Se ocorrerem erros esta saída é função apenas do sinal de erro, podendo-se chamar de forma apropriada de sindroma.

pode-se detectar a ocorrência de um erro quando o sinal s é maior que um dado limiar ε próximo zero, de modo a entrar em conta com o ruído de quantificação.

No actual momento não possuímos uma técnica eficiente para corrigir os erros ocorridos, nem métricas que permitam avaliar da capacidade de correcção do código. Uma das possibilidades será a de tentar subtrair o efeito dos erros à saída y. Vamos supor que ocorria um único erro com amplitude unitária e que só afectava o sinal y_0. Nesse caso, teríamos como sindroma a resposta impulsional do filtro $F_0(-z)$. O sistema de correcção ao detectar a primeira amostra de s diferente de zero pode esperar e guardar um número igual ao comprimento da resposta impulsional do filtro $F_0(-z)$, realizando a desconvolução desse pedaço de sinal com os filtros $F_0(-z)$ e $F_1(-z)$. De seguida pode-se subtrair estas estimativas para o sinal de erro aos sinais y_0 e y_1 e calcular neste caso duas novas versões do sindroma. A estimativa de erro que tiver anulado o sindroma é então utilizada para corrigir a saída r. Com este método só é possível corrigir erros que distem de um número de amostras igual à resposta a impulso dos filtros $F_i(z)$.

Capítulo 5

Combinatória dos padrões de erro

5.1 Introdução

No capítulo 3, foi reconhecido que a posição dos erros $\bar{S}_t$ influencia o condicionamento numérico do problema de reconstrução de sinal corrompido com ruído impulsivo. No entanto, para realizar o projecto de um sistema de correcção de erros no corpo dos reais seria interessante saber quantos dos $\binom{N}{t}$ padrões de erro é que se conseguem corrigir. Calcular o condicionamento do problema para todos os padrões de erro possíveis e verificar a correcta reconstrução só é viável para valores de N pequenos. Como alternativa, pode-se procurar uma métrica para os padrões de erro que possua alguma correlação com o condicionamento do problema a resolver e para a qual seja possível contar o número dos padrões de erro. Neste capítulo começaremos por uma descrição de alguns resultados conhecidos [Ferreira 97, Grochenig 93] sobre este problema, mas que utilizam outras métricas para classificar os padrões de erro em termos de condicionamento do problema a resolver. No entanto, nenhum destes estudos abordou o problema da contagem do número de padrões de erros que satisfazem uma dada métrica.

A métrica proposta neste capítulo é a da distância mínima entre erros que será definida na secção seguinte, sendo igualmente apresentadas as expressões que dão o número de combinações para cada um dos valores da distância mínima. Para cada um destes valores e no caso de um sindroma contíguo, a sequência de erros mais compacta é a que origina um condicionamento mais elevado para o sistema de equações a resolver. Se para um dado sistema de reconstrução é possível estimar este condicionamento, demonstraremos que se consegue calcular o erro cometido quer na determinação da posição dos erros quer na sua amplitude. No final deste capítulo serão descritos alguns exemplos de aplicação bem como um método para projectar códigos reais baseado na métrica da distância mínima.

5.2 Métricas para os padrões de erro

Para o problema de correcção de erros em posições conhecidas (apagamentos), se a sequência de erros i_m puder ser escrita como ki_r em que k é um inteiro positivo maior que zero, N/k é um inteiro, e i_r é outra sequência apropriada, podem ser dados limites precisos para o condicionamento do problema [Ferreira 97] pela expressão

$$\kappa\left(I-S\right)=\frac{\lambda_{\max}}{\lambda_{\min}}\leq\frac{k-\lfloor k\beta\rfloor}{k-\lceil k\beta\rceil}. \tag{5.1}$$

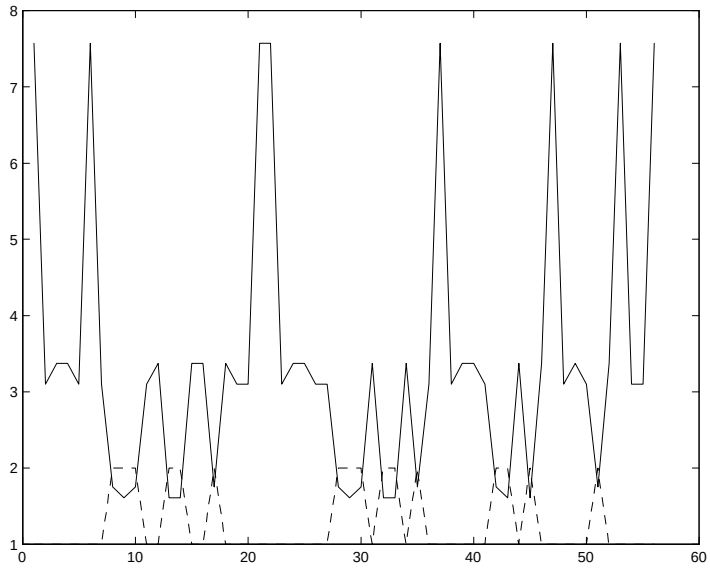


Figura 5.1: Nesta figura temos na linha a cheio o valor do condicionamento de $(I - S)$ para todas as combinações possíveis das posições dos erros com $N = 8$ e $t = 3$. A linha a tracejado indica o valor da distância mínima d para cada caso. Como se pode ver os máximos de d coincidem com os mínimos do condicionamento.

O parâmetro β é o factor de interpolação, definido por K/N, em que K é o número de componentes não nulas do espectro e as operações $\lceil\cdot\rceil$ e $\lfloor\cdot\rfloor$ representam a parte inteira de um real depois de arredondado para cima e para baixo respectivamente. A condição ki_r para os índices dos erros é equivalente a calcular o maior divisor comum k do conjunto $\bar{S}_t$ e utilizar esse valor para caracterizar o condicionamento do problema. Note-se que para as sequências em que $k = 1$ o limite dado pela equação 5.1 é trivial ($\kappa\,(I - S) \leq \infty$).

Esta métrica pode ser aplicada ao caso de uma transmissão de dados por pacotes, em que se introduziu entrelaçamento regular. No caso de se perder um pacote os erros serão da forma ki_r e a desigualdade 5.1, dá-nos um limite para o condicionamento para o problema de reconstrução.

Repare-se que o número de padrões de erro que satisfazem a condição $k \cdot i_r$ é relativamente reduzida, sendo o seu número dado pela expressão:

$$\bar{C}_t^N = \sum_{k=2}^{\lfloor \frac{N-1}{t-1} \rfloor} \sum_{i=0}^{k-1} \binom{\lceil (N-i)/k \rceil}{t}.$$

A métrica utilizada neste trabalho é a da distância mínima d entre as posições dos erros, que passamos a definir:

Definição 12 *Um padrão de erros $\bar{S}_t = \{i_0, i_1, \ldots, i_{t-1}\}$ tem distância mínima d se $|i_a - i_b| \geq d\,|a - b|$ para qualquer $0 \leq a, b \leq t$, onde $|\cdot|$ representa a distância circular com $i_t \triangleq i_0$.*

Esta métrica surgiu da constatação de que em qualquer problema de reconstrução de sinal com o sindroma contíguo, o condicionamento numérico piora quando os erros se agrupam. No caso da correcção de apagamentos podemos constatar pela figura 5.1 que quando a distância

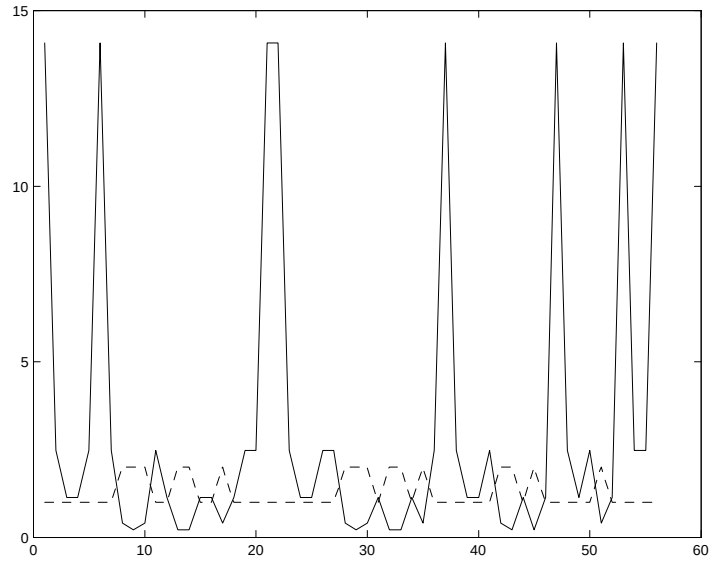


Figura 5.2: Nesta figura temos na linha a cheio o valor do condicionamento de T para todas as combinações possíveis das posições dos erros com $N = 8$ e $t = 3$. A linha a tracejado indica o valor da distância mínima d para cada caso. Como se pode ver os máximos de d coincidem com os mínimos do condicionamento.

mínima atinge o seu valor máximo, o condicionamento de $(I - S)$ atinge um mínimo. Este comportamento do condicionamento acontece também para o caso do problema da determinação da posição dos erros em que a matriz de Toeplitz (3.23) possui um comportamento idêntico tal como podemos ver na figura 5.2.

5.3 Combinatória dos padrões de erro

Pode-se verificar facilmente que num vector com N amostras e t erros se tem $\binom{N}{t}$ combinações possíveis para os padrões de erro. O nosso objectivo é contar o número de padrões para cada um dos valores da distância mínima d. Convém notar que a codificação é invariante a deslocamentos circulares e por conseguinte um padrão com dois erros nas posições 0 e $N - 1$ possui uma distância mínima $d = 1$. No entanto, dada a complexidade de contar as combinações de erros para este caso, vamos começar por analisar a versão mais simples em que não se consideram blocos circulares.

Para um dado N e t, o valor máximo que a distância mínima entre erros pode atingir ocorre quando o espaçamento entre os erros é igual, mas sem que o índice do último erro ultrapasse a dimensão N do bloco. Colocando esta condição na forma de equação temos

$$i_{t-1} \leq N - 1$$

e como

$$i_{t-1} = (t - 1)\, d_{\max} \tag{5.2}$$

tem-se

$$\begin{aligned}(t-1)\,d_{\max} &\leq N-1 \Leftrightarrow \\ d_{\max} &\leq \frac{N-1}{t-1}\end{aligned}$$

e dado que $d_{\max}$ é um inteiro vem

$$d_{\max} = \left\lfloor \frac{N-1}{t-1} \right\rfloor . \tag{5.3}$$

A equação (5.3) dá-nos o valor máximo que a distância entre erros pode ter em função da dimensão do bloco N e do número de erros t.

A partir da equação (5.2) pode-se concluir que para a sequência mencionada ficam livres no final do bloco

$$N-(t-1)\,d_{\max}-1 \tag{5.4}$$

posições para formar combinações mantendo o valor de $d = d_{\max}$. Como a distância entre erros neste caso é sempre maior ou igual que $d_{\max}$, o número de posições possíveis para os t erros são as t iniciais mais as livres no final do bloco, chegando-se assim ao número de combinações para este primeiro caso,

$$\begin{aligned}\mathcal{C}\,(N,t,d_{\max}) &= \binom{N-(t-1)\,d_{\max}-1+t}{t} = \\ &= \binom{N-(t-1)\,(d_{\max}-1)}{t}.\end{aligned} \tag{5.5}$$

Antes de prosseguir com um pequeno exemplo ilustrativo vamos definir as funções $\mathcal{S}$, S, $\mathcal{C}$ e C.

Definição 13 *A função $\mathcal{S}\,(N,t,d) \equiv \mathcal{S}_d$ dá o conjunto das combinações cuja distância mínima entre erros é **maior ou igual** a d, para um bloco de dimensão N e considerando t erros. O número destas combinações é dado por $\mathcal{C}\,(N,t,d) \equiv \mathcal{C}_d$.*

Definição 14 *A função $S\,(N,t,d) \equiv S_d$ dá o conjunto das combinações cuja distância mínima é **igual** a d. O número destas combinações é dado por $C\,(N,t,d) \equiv C_d$.*

Para clarificar ideias vamos verificar a expressão (5.5) para um pequeno exemplo com $N = 8$ e $t = 3$. Com a equação (5.3) obtemos o valor de $d_{\max} = 3$ para o valor máximo que a distância mínima pode atingir. Considere-se o caso inicial $U = \{0,3,6\}$

$$\boxed{0}\ 1\ 2\ \boxed{3}\ 4\ 5\ \boxed{6}\ 7$$

em que resta uma amostra livre no final do bloco tal como é dado pela expressão (5.4). Temos assim um total de 3 erros mais uma posição livre para gerar por ordem lexicográfica [Brualdi 77] as 4 combinações possíveis:

$$S_3\,(8,3,3) = \mathcal{S}_3\,(8,3,3) = \{0,3,6\}\,;\{0,3,7\}\,;\{0,4,7\}\,;\{1,4,7\}$$

confirmando-se o resultado da equação (5.5)

$$\mathcal{C}\,(8,3,3) = C\,(8,3,3) = \binom{8-(3-1)\,(3-1)}{3} = \binom{4}{3} = 4.$$

Se o mesmo raciocínio for realizado para o caso de $d = d_{\max} - 1$, a expressão (5.5), dá o número total de combinações em que $d \geq d_{\max} - 1$.

Em termos dos conjuntos $\mathcal{S}_d$ e S_d definidos anteriormente podemos escrever

$$\mathcal{S}_{k+1} \subset \mathcal{S}_k \tag{5.6}$$

e

$$\mathcal{S}_m = S_m \cup S_{m+1} \cup \cdots \cup S_{d_{\max}}. \tag{5.7}$$

Relações idênticas podem ser dadas para o número de combinações

$$C_d = \mathcal{C}_d - \mathcal{C}_{d+1} \tag{5.8}$$

e

$$\mathcal{C}_m = \sum_{k=m}^{d_{\max}} C_k$$

5.4 Combinatória dos padrões de erro para blocos circulares

Este problema é idêntico ao anterior mas considera-se agora o caso de blocos circulares. A solução encontrada baseia-se numa análise em que para cada valor de d se calculam as combinações possíveis para os diferentes valores que a posição do primeiro erro i_0 pode tomar, somando-se no final os resultados parcelares.

No cálculo da distância Δ_k entre erros dada por

$$\Delta_k = i_{k+1} - i_k,$$

o facto do bloco ser circular só constitui problema na determinação da distância entre o primeiro e o último erro. Neste caso temos de entrar em conta com a diferença $(i_0 + N) - i_{t-1}$, que pode ser incorporada no problema acrescentando ao conjunto $\bar{S}_t$ a posição $i_t = i_0 + N$, obtendo-se

$$\bar{S}_t = \{i_0, i_1, \ldots, i_{t-1}, i_t\}.$$

Pode-se verificar que o máximo da distância mínima entre erros $d_{\max}$ vai ser ligeiramente inferior no caso dos blocos circulares dado que algumas das últimas amostras do bloco não podem ser usadas para gerar combinações. Tal como anteriormente, vamos deduzir o valor de $d_{\max}$ em função de N e t. Para garantir que $d \leq d_{\max}$, é necessário que $i_k - i_{k-1} \geq d_{\max}$, e em particular que

$$i_t - i_{t-1} \geq d_{\max} \Leftrightarrow (i_0 + N) - i_{t-1} \geq d_{\max}. \tag{5.9}$$

Considere-se a seguinte situação particular em que os erros satisfazem a condição anterior

$$\begin{aligned} i_0 &= 0 \\ i_1 &= i_0 + d_{\max} = d_{\max} \\ &\vdots \\ i_{t-1} &= i_0 + (t-1)d_{\max} = (t-1)d_{\max}. \end{aligned}$$

Como $i_t = i_0 + N$, a condição (5.9) conduz a

$$\begin{aligned}(i_0 + N) - (i_0 + (t-1)\, d_{\max}) &\geq d_{\max} \Leftrightarrow \\ (1+N) - (t-1)\, d_{\max} &\geq d_{\max} \Leftrightarrow \\ N \geq t d_{\max} \Leftrightarrow d_{\max} \leq \frac{N}{t}.&\end{aligned}$$

Como $d_{\max}$ é um inteiro chega-se finalmente à condição

$$\boxed{d_{\max} = \left\lfloor \frac{N}{t} \right\rfloor}. \tag{5.10}$$

Este é o valor máximo que a distância mínima pode tomar, dados N e t. Estamos agora em condições de formular a seguinte proposição.

Proposição 3 *O número de padrões de erro com distância mínima entre erros maior ou igual a d é dado por*

$$\mathcal{C}_d = \sum_{i_0=0}^{N-(t-1)d-1} \mathcal{C}\left(N, t, d, i_0\right), \tag{5.11}$$

com

$$\mathcal{C}\left(N, t, d, i_0\right) = \binom{N - (d-1)\,(t-1) - (i_0+1) - \langle d - i_0 - 1 \rangle}{t-1}$$

e onde $\langle \cdot \rangle$ tem o seguinte significado

$$\begin{aligned}\langle n \rangle &= 0, \qquad n < 0 \\ \langle n \rangle &= n, \qquad n \geq 0.\end{aligned}$$

Demonstração. Vamos começar por considerar o caso em que $d = d_{\max}$ e $i_0 = 0$. Neste caso, para calcular o número de posições livres para gerar as combinações, temos de subtrair a N as $(t-1) \times (d_{\max} - 1)$ posições entre os erros e a posição ocupada por i_0 que é fixa. Por outro lado como o bloco é circular, se $i_0 = 0$ e $i_{t-1} = N - 1$, teremos $d = 1$. Assim, de modo a garantir que i_{t-1} não viola a distância mínima, existem $d_{\max} - 1$ posições contíguas no final do bloco que não poderemos utilizar. O número de posições possíveis para os erros é dado por $N - (t-1) \times (d_{\max} - 1) - 1 - (d_{\max} - 1)$ e como consideramos i_0 fixo, teremos apenas $t - 1$ erros para gerar combinações. Logo o número total de combinações é dado por

$$\mathcal{C}\left(N, t, d_{\max}, i_0 = 0\right) = \binom{N - (d_{\max} - 1)\,(t-1) - d_{\max}}{t-1}. \tag{5.12}$$

Incrementando a posição do primeiro erro i_0, o número de posições possíveis para os erros mantém-se constante uma vez que teremos menos uma posição no início para gerar as combinações que é compensada por uma a mais no final do bloco. Este efeito desaparece quando $i_0 \geq d - 1$, sendo o número de combinações reduzida até i_0 atingir o seu valor máximo dado por

$$\max\left(i_0\right) = N - (t-1)\, d_{\max} - 1.$$

A equação (5.12) é igualmente válida para valores $d < d_{\max}$, e somando os termos correspondentes a todos os valores possíveis de i_0, teremos finalmente

$$\mathcal{C}_d = \sum_{i_0=0}^{N-(t-1)d-1} \mathcal{C}\left(N, t, d, i_0\right),$$

com

$$\mathcal{C}\left(N, t, d, i_0\right) = \binom{N-(d-1)\left(t-1\right)-\left(i_0+1\right)-\langle d-i_0-1\rangle}{t-1}.$$

■

Uma vez que no caso de blocos circulares a relação (5.7) também se verifica, podemos utilizar a equação (5.8) para obter o número de padrões de erros que têm uma dada distância mínima d.

Vejamos agora um pequeno exemplo para o caso de $N = 8$ e $t = 3$, tendo-se neste caso $d_{\max} = \left\lfloor \frac{8}{3} \right\rfloor = 2$. Para garantir este valor da distância mínima com $i_0 = 0$ a sequência de erros mais à esquerda é

$$\boxed{0}\ 1\ \boxed{2}\ 3\ \boxed{4}\ 5\ 6\ 7 \vdots \boxed{8},$$

ficando apenas livres para gerar combinações as posições 5 e 6. Para $i_0 = 0$, teremos 16 combinações com distância mínima $d = 2$:

$$\begin{array}{ll} i_0 = 0 & \{0,2,4\}, \{0,2,5\}, \{0,2,6\}, \{0,3,5\}, \{0,3,6\}, \{0,4,6\} \\ i_0 = 1 & \{1,3,5\}, \{1,3,6\}, \{1,3,7\}, \{1,4,6\}, \{1,4,7\}, \{1,5,7\} \\ i_0 = 2 & \{2,4,6\}, \{2,4,7\}, \{2,5,7\} \\ i_0 = 3 & \{3,5,7\}, \end{array}$$

que dá o mesmo resultado que a equação (5.11)

$$\mathcal{C}_d = \sum_{i_0=0}^{3} \mathcal{C}\left(8, 3, 2, i_0\right) = \binom{4}{2} + \binom{4}{2} + \binom{3}{2} + \binom{2}{2} = 16.$$

Nas tabelas que se seguem temos alguns exemplos do número de combinações para vários valores de N e t, tendo todos estes resultados sido confirmados experimentalmente.

Para realizar o projecto dos códigos reais correctores de apagamentos interessa saber o valor do condicionamento mínimo $\kappa_{\min}$ e máximo $\kappa_{\max}$ da matriz $(I - S)$ para cada valor da distância mínima entre erros. A definição usada para o condicionamento foi [Horn 85]

$$\kappa\left(I - S\right) = \left\| (I-S)^{-1} \right\|_{\infty} \left\| (I-S) \right\|_{\infty}.$$

A utilização da norma $\|\cdot\|_{\infty}$ permitirá mais tarde calcular o erro máximo na reconstrução de sinal em função de $\kappa\left(I - S\right)$. Os valores $\tilde{\kappa}_{\min}$ e $\tilde{\kappa}_{\max}$ são estimativas de $\kappa_{\min}$ e $\kappa_{\max}$, respectivamente, e foram obtidos considerando as seguintes sequências de erro:

- Apenas dois erros à distância d, e os restantes com o espaçamento máximo possível entre eles, ou seja, a sequência mais esparsa com distância mínima d.
- Todos os erros à distância d uns dos outros, ou seja, a sequência mais compacta com distância mínima d.

Vejamos alguns exemplos para estas duas situações:

Exemplo 1 *Para um bloco de dimensão* $N = 10$*,* $t = 4$ *e* $d = 1$

- *Sequência mais esparsa* $\{0, 1, 4, 7\}$
- *Sequência mais compacta* $\{0, 1, 2, 3\}$

Exemplo 2 *Para um bloco de dimensão* $N = 10$*,* $t = 4$ *e* $d = 2$

- *Sequência mais esparsa* $\{0, 2, 4, 7\}$
- *Sequência mais compacta* $\{0, 2, 4, 6\}$

Note-se que as sequências de cada tipo não são únicas. Da observação das tabelas pode-se concluir que o número de combinações com distância $d = 1$ é relativamente elevado e que os limites inferiores e superiores do condicionamento de $(I - S)$ para cada d se sobrepõem bastante para valores de d adjacentes. Estes dois factos limitam de algum modo a aplicação prática deste método uma vez que se a classificação fosse ideal não existiria sobreposição entre os limites para os casos com d adjacente.

As tabelas mencionadas foram geradas mantendo o número de componentes espectrais nulas no seu valor mínimo, ou dito de outro modo, mantendo o factor de sobre-amostragem no seu valor máximo (que corresponde a uma frequência de amostragem o mais próximo possível da frequência de Nyquist). Contudo, se o sistema de descodificação conseguir corrigir sem erro o padrão de erros com o maior κ para uma dada distância mínima d, então será capaz de corrigir todos os padrões com distância mínima superior a d. Na figura 5.4 podemos apreciar o condicionamento normalizado de $(I - S)$ para todos os padrões de erro. Da figura podemos concluir que o factor de sobre-amostragem é determinante no valor do condicionamento de $(I - S)$, e que influencia o número de padrões de erro com o mesmo valor de κ.

Na figura 5.3 podemos observar um histograma comulativo do número de combinações que têm um condicionamento menor que um dado valor. Apesar de ser apenas um caso particular, e de o condicionamento máximo ter um valor elevado (3200), cerca de 90% dos padrões de erro possuem um condicionamento abaixo de 40. Se o algoritmo de correcção for capaz de resolver o problema com este valor do condicionamento, então será possível corrigir 90% dos padrões de erro.

$t = 3$	$N = 16$	$K = 13$	$\beta \approx 0.81$		
d	$\kappa_{\max}$	$\tilde{\kappa}_{\max}$	$\kappa_{\min}$	$\tilde{\kappa}_{\min}$	$Comb$
1	3.05e+003	3.05e+003	4.30e+001	4.30e+001	208
2	1.41e+002	1.41e+002	1.01e+001	1.01e+001	160
3	1.84e+001	1.84e+001	4.37e+000	4.37e+000	112
4	4.00e+000	4.00e+000	2.14e+000	2.14e+000	64
5	1.42e+000	1.42e+000	1.42e+000	1.42e+000	16

$t = 3$	$N = 32$	$K = 29$	$\beta \approx 0.91$		
d	$\kappa_{\max}$	$\tilde{\kappa}_{\max}$	$\kappa_{\min}$	$\tilde{\kappa}_{\min}$	$Comb$
1	5.30e+004	5.30e+004	1.71e+002	1.71e+002	928
2	3.05e+003	3.05e+003	4.20e+001	4.20e+001	832
3	5.29e+002	5.29e+002	1.85e+001	1.85e+001	736
4	1.41e+002	1.41e+002	1.01e+001	1.01e+001	640
5	4.72e+001	4.72e+001	6.32e+000	6.32e+000	544
6	1.84e+001	1.84e+001	4.19e+000	4.19e+000	448
7	8.12e+000	8.12e+000	2.98e+000	2.98e+000	352
8	4.00e+000	4.00e+000	2.14e+000	2.14e+000	256
9	2.38e+000	2.38e+000	1.64e+000	1.64e+000	160
10	1.42e+000	1.42e+000	1.20e+000	1.20e+000	64

$t = 3$	$N = 64$	$K = 61$	$\beta \approx 0.95$		
d	$\kappa_{\max}$	$\tilde{\kappa}_{\max}$	$\kappa_{\min}$	$\tilde{\kappa}_{\min}$	$Comb$
1	8.66e+005	8.66e+005	6.81e+002	6.81e+002	3904
2	5.30e+004	5.30e+004	1.70e+002	1.70e+002	3712
3	1.01e+004	1.01e+004	7.52e+001	7.52e+001	3520
4	3.05e+003	3.05e+003	4.20e+001	4.20e+001	3328
5	1.18e+003	1.18e+003	2.67e+001	2.67e+001	3136
6	5.29e+002	5.29e+002	1.84e+001	1.84e+001	2944
7	2.63e+002	2.63e+002	1.34e+001	1.34e+001	2752
8	1.41e+002	1.41e+002	1.01e+001	1.01e+001	2560
9	7.97e+001	7.97e+001	7.87e+000	7.87e+000	2368
10	4.72e+001	4.72e+001	6.26e+000	6.26e+000	2176
11	2.90e+001	2.90e+001	5.10e+000	5.10e+000	1984
12	1.84e+001	1.84e+001	4.19e+000	4.19e+000	1792
13	1.21e+001	1.21e+001	3.50e+000	3.50e+000	1600
14	8.12e+000	8.12e+000	2.94e+000	2.94e+000	1408
15	5.62e+000	5.62e+000	2.51e+000	2.51e+000	1216
16	4.00e+000	4.00e+000	2.14e+000	2.14e+000	1024
17	3.08e+000	3.08e+000	1.85e+000	1.85e+000	832
18	2.38e+000	2.38e+000	1.59e+000	1.59e+000	640
19	1.84e+000	1.84e+000	1.39e+000	1.39e+000	448
20	1.42e+000	1.42e+000	1.20e+000	1.20e+000	256
21	1.09e+000	1.09e+000	1.09e+000	1.09e+000	64

$t = 4$	$N = 16$	$K = 12$	$\beta \approx 0.75$		
d	$\kappa_{\max}$	$\tilde{\kappa}_{\max}$	$\kappa_{\min}$	$\tilde{\kappa}_{\min}$	$Comb$
1	4.58e+003	4.58e+003	1.40e+001	1.53e+001	1160
2	2.63e+001	2.63e+001	3.16e+000	3.78e+000	520
3	2.77e+000	1.48e+000	1.48e+000	2.26e+000	136
4	2.00e+000	2.00e+000	2.00e+000	2.00e+000	4

$t = 4$	$N = 32$	$K = 28$	$\beta \approx 0.88$		
d	$\kappa_{\max}$	$\tilde{\kappa}_{\max}$	$\kappa_{\min}$	$\tilde{\kappa}_{\min}$	$Comb$
1	4.24e+005	4.24e+005	5.61e+001	6.25e+001	12560
2	4.58e+003	4.58e+003	1.38e+001	1.53e+001	9232
3	2.44e+002	2.44e+002	5.83e+000	6.68e+000	6416
4	2.63e+001	2.63e+001	3.15e+000	3.84e+000	4112
5	4.76e+000	4.76e+000	1.86e+000	2.63e+000	2320
6	2.77e+000	1.48e+000	1.29e+000	2.26e+000	1040
7	2.39e+000	1.64e+000	1.64e+000	2.17e+000	272
8	2.00e+000	2.00e+000	2.00e+000	2.00e+000	8

$t = 4$	$N = 64$	$K = 60$	$\beta \approx 0.94$		
d	$\kappa_{\max}$	$\tilde{\kappa}_{\max}$	$\kappa_{\min}$	$\tilde{\kappa}_{\min}$	$Comb$
1	3.00e+007	3.00e+007	2.26e+002	2.46e+002	115232
2	4.24e+005	4.24e+005	5.61e+001	6.12e+001	100384
3	3.17e+004	3.17e+004	2.47e+001	2.75e+001	86560
4	4.58e+003	4.58e+003	1.37e+001	1.53e+001	73760
5	9.43e+002	9.43e+002	8.58e+000	9.71e+000	61984
6	2.44e+002	2.44e+002	5.82e+000	6.81e+000	51232
7	7.47e+001	7.47e+001	4.17e+000	4.93e+000	41504
8	2.63e+001	2.63e+001	3.09e+000	3.73e+000	32800
9	1.05e+001	1.05e+001	2.37e+000	3.01e+000	25120
10	4.76e+000	4.76e+000	1.84e+000	2.63e+000	18464
11	2.94e+000	2.50e+000	1.47e+000	2.40e+000	12832
12	2.77e+000	1.48e+000	1.18e+000	2.33e+000	8224
13	2.58e+000	1.10e+000	1.10e+000	2.21e+000	4640
14	2.39e+000	1.64e+000	1.64e+000	2.13e+000	2080
15	2.20e+000	2.06e+000	2.06e+000	2.10e+000	544
16	2.00e+000	2.00e+000	2.00e+000	2.00e+000	16

$t = 5$	$N = 16$	$K = 11$	$\beta \approx 0.69$		
d	$\kappa_{\max}$	$\tilde{\kappa}_{\max}$	$\kappa_{\min}$	$\tilde{\kappa}_{\min}$	$Comb$
1	1.78e+005	1.78e+005	1.51e+001	2.45e+001	3696
2	1.62e+002	1.62e+002	3.28e+000	4.55e+000	656
3	2.00e+000	2.00e+000	2.00e+000	2.00e+000	16

$t = 5$	$N = 32$	$K = 27$	$\beta \approx 0.84$		
d	$\kappa_{\max}$	$\tilde{\kappa}_{\max}$	$\kappa_{\min}$	$\tilde{\kappa}_{\min}$	$Comb$
1	7.82e+007	7.82e+007	5.88e+001	7.26e+001	105696
2	1.78e+005	1.78e+005	1.45e+001	1.67e+001	57376
3	3.48e+003	3.48e+003	6.10e+000	6.47e+000	26656
4	1.62e+002	1.62e+002	3.28e+000	3.37e+000	9536
5	1.27e+001	1.27e+001	1.94e+000	2.61e+000	2016
6	2.00e+000	2.00e+000	1.38e+000	1.60e+000	96

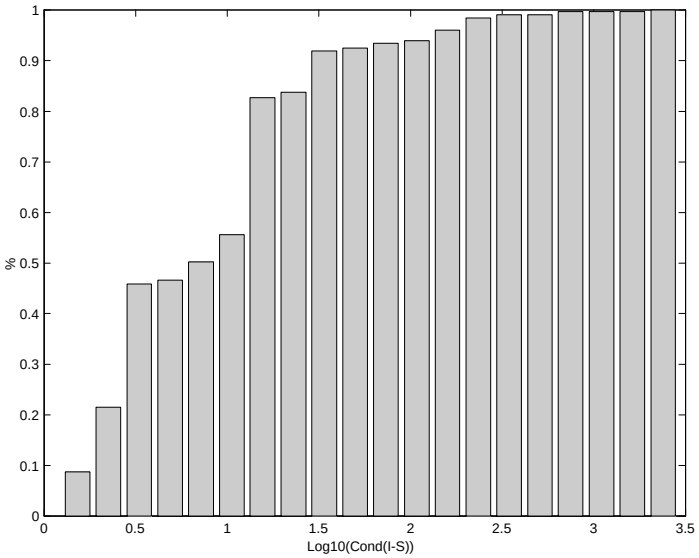


Figura 5.3: Histograma cumulativo do condicionamento de $(I - S)$ para $N = 22$ e $t = 4$. Nas abcissas temos o logaritmo de base 10 do condicionamento.

$t = 5$	$N = 64$	$K = 59$	$\beta \approx 0.92$		
d	$\kappa_{\max}$	$\tilde{\kappa}_{\max}$	$\kappa_{\min}$	$\tilde{\kappa}_{\min}$	$Comb$
1	2.34e+010	2.34e+010	2.37e+002	2.81e+002	219386
2	7.82e+007	7.82e+007	5.85e+001	6.89e+001	168250
3	2.40e+006	2.40e+006	2.58e+001	2.85e+001	125754
4	1.78e+005	1.78e+005	1.43e+001	1.59e+001	910976
5	2.14e+004	2.14e+004	8.99e+000	9.91e+000	634816
6	3.48e+003	3.48e+003	6.06e+000	6.64e+000	421056
7	6.97e+002	6.97e+002	4.36e+000	4.54e+000	261696
8	1.62e+002	1.62e+002	3.21e+000	3.37e+000	148736
9	4.24e+001	4.24e+001	2.46e+000	2.68e+000	74176
10	1.27e+001	1.27e+001	1.89e+000	2.03e+000	30016
11	4.58e+000	4.58e+000	1.51e+000	1.55e+000	8256
12	2.00e+000	2.00e+000	1.20e+000	1.20e+000	896

5.5 Projecto de códigos reais

Os códigos de correcção de erros em corpos finitos não são afectados por problemas de estabilidade numérica, devido ao facto de nestes corpos não existirem erros de arredondamento durante as operações aritméticas. Tal como foi mencionado no capítulo 4, qualquer código de correcção de erros definido no corpo dos complexos $\mathbb{C}$, sofre de problemas de estabilidade inerentes à própria aritmética. No capítulo 4 descrevemos um código de Reed-Muller realizado no corpo dos reais e que consegue assegurar b digitos binários correctos na descodificação.

No caso dos métodos de correcção de apagamentos descritos no capítulo 1 e dos métodos de correcção de erros do capítulo 2, também é possível determinar de forma aproximada quantos bits correctos se podem garantir na descodificação. Nesta secção iremos estudar este problema sendo apresentados no final os resultados de algumas simulações.

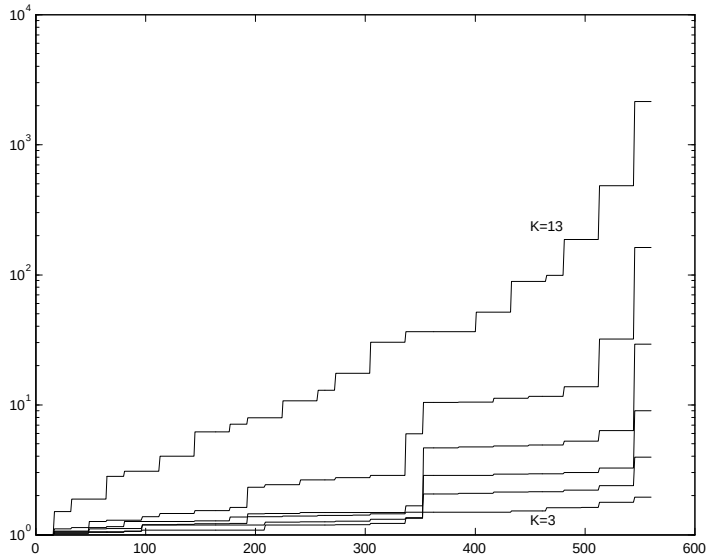


Figura 5.4: Nesta figura podemos observar várias curvas do condicionamento de $(I - S)$ para todos os possíveis padrões de erro, para o caso em que $N = 16$ e $t = 3$. Na curva superior temos $K = 13$ e na inferior $K = 3$, em que K é o número de componentes espectrais não nulas da codificação. Cada uma das curvas foi normalizada pelo condicionamento mínimo de cada curva e desenhada com escala logarítmica nas ordenadas. O valor do condicionamento em cada uma das curvas foi ordenado por ordem crescente.

5.5.1 Definição do problema

Na figura 5.5 podemos observar o sistema de codificação/descodificação para correcção de erros em que se especifica o número de bits de cada sinal envolvido.

- m - Sinal mensagem com K amostras representadas com b bits.
- x - Sinal transmitido com N amostras representadas com $b + r$ bits.
- y - Sinal recebido com N amostras representadas com $b + r$ bits.
- $\tilde{m}$ - Sinal mensagem recebido com K amostras representadas com b bits.

Codificador

O codificador manipula o sinal mensagem tal como na equação (2.28) de modo a conseguir que o sinal x transmitido seja real e que a codificação seja realizada com apenas uma transformada. A utilização da ODFT garante por seu lado que o código é do tipo CDM quando N e K são pares. Dito de outro modo, acrescentam-se somente $2t$ zeros de redundância para se conseguirem corrigir t erros. O sinal x à saída do codificador real tem de ser quantificado para $b+r$ bits antes de ser transmitido. Repare-se que b é o número de bits do sinal mensagem e r o número extra de bits acrescentado de modo a garantir um ruído η de quantificação baixo. Teremos então que

$$x_q = x + \eta.$$

O ruído de quantificação propaga-se pelo descodificador, constituindo a principal fonte de erro numérico.

Descodificador

O descodificador é realizado em duas fases: detecção da posição dos erros e correcção da sua amplitude. Depois de calcular a ODFT do sinal recebido utilizam-se as amostras do sindroma para formar a matriz A tal como na equação (2.15). Resolvendo este sistema de equações obtemos os coeficientes do polinómio localizador de erros e para calcular os zeros deste polinómio e determinar a partir deles as posições dos erros utilizamos o método descrito no final da sub-secção 2.2.3. Obtidas as posições dos erros, a amplitude destes é determinada utilizando o método de dimensão mínima descrito no capítulo 1, e do qual é dado um exemplo na subsecção 2.3.4.

Depois de corrigidos os erros, o sinal mensagem é novamente quantificado com b bits, obtendo-se o sinal $\tilde{m}$, sendo o principal objectivo deste sistema que este sinal seja igual ao sinal m transmitido em todos os bits. Quando tal não acontece diremos que ocorreu um erro de descodificação.

Projecto do código

Se considerarmos um dado esquema de codificação / descodificação, sendo N a dimensão de bloco, $M/2$ o número máximo de erros e b o número de bits do sinal de entrada, então, o projecto de um código real consiste unicamente em determinar o número de bits r que assegura uma comunicação com $\tilde{m} = m$ quando $t \leq M/2$. Se o número de bits r não puder ser tão elevado, o método de projecto deve ainda poder fornecer a percentagem de padrões de erro que é possível corrigir.

No capítulo 3 foi justificada de forma rigorosa a importância do condicionamento $\kappa(A)$ na resolução de um sistema de equações da forma

$$Ax = b \tag{5.13}$$

tendo sido fornecido um limite (3.5) para o erro relativo na solução x quando a matriz A e o vector b, são conhecidos de forma aproximada e $|x_k| \leq 1$. Estes resultados serão utilizados para determinar a influência do ruído η na determinação das posições dos erros. O sinal recebido y_q, é dado por

$$y_q = e + x_q = e + x + \eta,$$

e virá após a ODFT

$$\hat{y}_q = Fe + Fx + F\eta,$$

em que F é a matriz da transformada ODFT (2.27). Se definirmos

$$\hat{\eta} = F\eta,$$

então

$$\|\hat{\eta}\| = \|F\eta\| \leq \|F\| \, \|\eta\| \,,$$

é válida para qualquer norma vectorial $\|\cdot\|$, e norma matricial por si induzida. Considerando daqui em diante a norma $\|\cdot\|_\infty$, a equação anterior virá

$$\|\hat{\eta}\|_\infty \leq \sqrt{N} \, \|\eta\|_\infty \,,$$

e como o sinal antes de ser transmitido é arredondado para $b+r$ bits, a norma do ruído de arredondamento $\|\eta\|_\infty$ será dada por

$$\|\eta\|_\infty = \frac{2^{-(b+r)}}{2},$$

e então teremos o limite

$$\|\hat{\eta}\|_\infty \leq \sqrt{N}\frac{2^{-(b+r)}}{2}. \tag{5.14}$$

Se atendermos à forma como a matriz A é construída (2.15), verificamos que esta é constituída por elementos do sindroma que só deviam ser função do sinal de erro e. Contudo, devido ao arredondamento do sinal transmitido, para as componentes pertencentes ao sindroma, cujos índices são dados por S_f, teremos

$$\hat{y}_q = Fe + F\eta.$$

Podemos assim estimar a influência do erro no sistema de equações $A\alpha = b$, por

$$(A+\Sigma)\,\alpha = (b+\epsilon)$$

em que Σ representa a influência do erro de arredondamento em A e ϵ a influência do mesmo erro em b. Para estes sinais de erro obtidos a partir de $\hat{\eta}$, e utilizando a equação (5.14), podem-se definir os seguintes limites para o erro relativo em A e b na norma $\|\cdot\|_\infty$

$$\frac{\|\Sigma\|_\infty}{\|A\|_\infty} \leq \sqrt{N}\frac{2^{-(b+r)}}{2} \tag{5.15}$$

$$\frac{\|\epsilon\|_\infty}{\|b\|_\infty} \leq \frac{2^{-(b+r)}}{2}. \tag{5.16}$$

Utilizando os limites obtidos na equação (3.5) teremos

$$\frac{\|\alpha-\tilde{\alpha}\|}{\|\alpha\|} \leq \frac{\kappa(A)}{1-\kappa(A)\left(\|\Sigma\|/\|A\|\right)}\frac{\|\Sigma\|}{\|A\|} + \frac{\kappa(A)}{1-\kappa(A)\left(\|\Sigma\|/\|A\|\right)}\frac{\|\epsilon\|}{\|b\|}, \tag{5.17}$$

sendo todas as normas $\|\cdot\|_\infty$, e o condicionamento $\kappa(A)$ definido por

$$\kappa(A) = \|A\|_\infty \left\|A^{-1}\right\|_\infty. \tag{5.18}$$

A desigualdade (5.17) pode ser simplificada se atendermos a que $\kappa(A)\left(\|\Sigma\|/\|A\|\right) \ll 1$ se a pertubação for suficientemente pequena para garantir que A possui inversa. Teremos então

$$\frac{\|\alpha-\tilde{\alpha}\|}{\|\alpha\|} \leq \kappa(A)\left[\frac{\|\Sigma\|}{\|A\|} + \frac{\|\epsilon\|}{\|b\|}\right],$$

que combinada com as equações (5.15) resulta em

$$\frac{\|\alpha-\tilde{\alpha}\|}{\|\alpha\|} \leq \kappa(A)\,\frac{2^{-(b+r)}}{2}\left(\sqrt{N}+1\right). \tag{5.19}$$

Para estimar o valor de $\kappa(A)$, podemos utilizar duas abordagens: uma mais empírica consiste em considerar como uma boa aproximação o valor que o condicionamento de A toma para o seguinte padrão de erros

$$e\left(\bar{S}_t\right) = \{-1, 1-1, \ldots, -2^{-b}, 2^{-b}, \cdots\},$$

em que temos metade dos erros com amplitude máxima e a outra com amplitude mínima. O valor de $\kappa(A)$ para este padrão de erros não é muito diferente do valor máximo para um dado problema, tal como se pode constatar por exemplo na tabela 3.4.

Outra abordagem consiste em utilizar os limites obtidos para o condicionamento da matriz A na norma l_2 da equação (3.32) e que têm a vantagem de não ser necessário inverter a matriz A mas que são sempre mais conservadores.

Como já foi mencionado, para determinar a posição dos erros, podemos utilizar o método descrito no final da sub-secção 2.2.3, que calcula a transformada de Fourier de α em N pontos

$$\hat{\alpha} = F\alpha,$$

onde F designa agora uma submatriz de Fourier de dimensão $N \times (t+1)$, sendo o erro relativo dado por

$$\frac{\|\Delta_{\hat{\alpha}}\|}{\|\hat{\alpha}\|} \leq \sqrt{N}\frac{\|\Delta_{\alpha}\|}{\|\alpha\|},$$

obtendo-se finalmente a expressão para a influência do erro de arredondamento do sinal x na determinação da posição dos erros

$$\frac{\|\Delta_{\hat{\alpha}}\|}{\|\hat{\alpha}\|} \leq \kappa(A)\,\frac{2^{-(b+r)}}{2}\left(N + \sqrt{N}\right). \tag{5.20}$$

Se pretendermos um erro relativo em $\hat{\alpha}$ menor do que 2^{-p}, então a seguinte desiguldade terá que ser satisfeita

$$2^{-p} \leq \kappa(A)\,\frac{2^{-(b+r)}}{2}\left(N + \sqrt{N}\right). \tag{5.21}$$

5.5.2 Utilização dos resultados de combinatória

Como veremos nos exemplos dados a seguir, o condicionamento da matriz A para o pior caso de padrão de erros, atinge valores extremamente elevados mesmo para valores pequenos de N e t. No entanto, sabemos que são relativamente poucos os padrões de erro com um condicionamento tão elevado e que a maioria possui um $\kappa(A)$ bastante inferior (ver o histograma da figura 5.3). O objectivo a atingir em termos de projecto de códigos no corpo dos reais, consiste em encontrar uma técnica que responda à seguinte questão:

Problema 2 *Considerando um sistema de codificação no corpo dos reais em que o número de bits $r+b$ do sinal transmitido não é suficiente para assegurar a correção de t erros, quantos dos padrões com t erros é que o sistema é capaz de corrigir correctamente?*

Os resultados de combinatória obtidos anteriormente, permitem responder a esta questão de forma aproximada. Se determinarmos com base na expressão (5.20) que o sistema consegue descodificar correctamente todos os padrões com $\kappa(A) < L$, e se L for o condicionamento mais elevado no conjunto de padrões de erro com distância mínima d, então o sistema será capaz de corrigir todos os padrões com uma distância mínima igual ou inferior a d. Como podemos saber o número de padrões para cada valor da distância mínima d, podemos igualmente determinar o número de padrões de erro que temos garantias de conseguir corrigir.

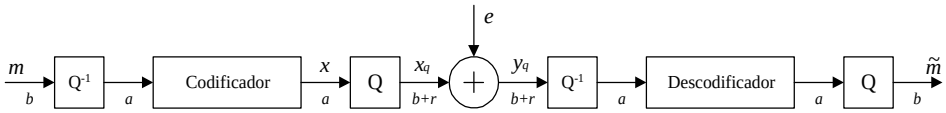


Figura 5.5: Sistema de correcção de erros com números reais. O sinal mensagem m possui apenas b bits significativos sendo a codificação realizada com uma precisão bastante maior. Para além da redundância acrescentada pelo codificador, o sinal transmitido terá $b+r$ bits de modo a garantir um ruído pequeno no sindroma do sinal recebido y_q. Depois de descodificado o sinal recebido é novamente quantificado para b bits.

5.5.3 Exemplo

Vamos agora considerar um exemplo de um código real com $N = 32$, $t = 4$ e $K = 26$. O sinal mensagem m possui $b = 8$ bits por amostra e pretende-se calcular o número $(b + r)$ de bits com que se deve transmitir o sinal codificado para as seguintes situações:

- Ocorreram 4 erros e o sinal descodificado $\tilde{m}$ deve ser igual a m em todos os seus bits para todos os padrões de erro possíveis;
- Ocorreram 4 erros e o sinal m deve ser descodificado com sucesso para a maioria dos padrões de erro.

O valor do condicionamento da matriz A para o padrão de erros da equação (5.21), é $\kappa_\infty(A) = 2.27 \times 10^8 \approx 2^{28}$, e se pretendemos conseguir corrigir todos os possíveis padrões de erro, então terá de existir alguma margem na distinção dos zeros do espectro do polinómio localizador de erros, tendo sido escolhido o valor de 2^{-8}. Substituindo os valores anteriores na equação (5.19)

$$\begin{aligned} \frac{\|\alpha - \tilde{\alpha}\|}{\|\alpha\|} &= 2^{-8} \leq 2^{28} \frac{2^{-(8+r)}}{2} \left(32 + \sqrt{32}\right) \\ 2^{-8} &\leq 2^{19} 2^5 2^{-r} \\ r &\geq 32. \end{aligned}$$

Se utilizarmos 40 bits para representar o sinal transmitido x_q garantimos uma transmissão sem erros na representação binária do sinal mensagem se ocorrerem 4 ou menos erros. O factor de sobre-amostragem efectivo é demasiado pequeno para atingir o objectivo de conseguir corrigir apenas 4 erros. O sinal m utiliza $K \times b = 26 \times 8 = 208$ bits, enquanto o sinal transmitido x_q utiliza $N \times (b + r) = 32 \times (8 + 32) = 1280$ bits, obtendo-se um factor de sobre-amostragem

$$\beta = \frac{208}{1280} \approx 0.16.$$

Podemos utilizar os resultados do estudo da combinatória dos padrões de erro para determinar o número de padrões de erro que se continua a conseguir corrigir se diminuirmos o número de bits de transmissão $(b + r)$. Se por exemplo pretendermos corrigir apenas os padrões de erro com distância mínima $d \geq 2$, temos para pior caso do condicionamento $\kappa_\infty(A) = 1.2 \times 10^3 \approx 2^{10}$, obtido para as sequências de erros mais compactas com $d = 2$. O número de bits r para esta situação será igualmente dado pela equação (5.19)

$$\begin{aligned}\frac{\|\alpha-\tilde{\alpha}\|}{\|\alpha\|} &= 2^{-8} \leq 2^{10}\frac{2^{-(8+r)}}{2}\left(32+\sqrt{32}\right)\\ 2^{-8} &\leq 2^1 2^5 2^{-r}\\ r &\geq 14,\end{aligned}$$

que corresponde a um factor de sobre-amostragem efectivo de

$$\beta = \frac{208}{704} \approx 0.3.$$

A percentagem de padrões de erro que possuem uma distância mínima maior ou igual a 2 é de 23400 de um total de 35960, ou seja garantimos que o código corrige cerca de 65% dos padrões de erro possíveis. O número efectivo deve ser ligeiramente superior uma vez que existem padrões de erro com $d = 1$ a que corresponde a condicionamento para a matriz A bastante inferior ao pior caso com $d = 2$.

5.5.4 Simulações

Com o objectivo de validar resultados realizámos algumas simulações dos algoritmos de reconstrução descritos. Estas simulações foram realizadas em Matlab que utiliza o algoritmo de eliminação Gaussiana com "pivoting" parcial para resolver os sistemas de equações. Nas figuras 5.6 e 5.7 podemos ver os resultados para dois valores diferentes do número de bits $r + b$ utilizados na transmissão. Os gráficos apresentados são do módulo da transformada de Fourier dos coeficientes do polinómio localizador de erros obtidos a partir da solução do sistema de equações (2.15), e cujos zeros dão a posição dos erros ocorridos. Para cada valor de $b + r$ realizaram-se várias transmissões de sinal corrompido por ruído impulsivo de amplitude aleatória nas quatro primeiras amostras, tendo sido sobrepostos os resultados de 100 simulações. Estes gráficos funcionam um pouco como o diagrama de olho pois permitem determinar a margem conseguida na determinação dos zeros do polinómio localizador de erros.

5.5.5 Correcção de apagamentos - simulações

Como vimos no capítulo 3, o condicionamento da matriz $(I - S)$ que aparece no método de dimensão mínima no domínio do tempo, diminui quando se aumenta a sobre-amostragem. Por outro lado a redundância é acrescentada ao sinal transmitido de duas formas: no número de componentes espectrais nulas $N - K$ acrescentadas e no número de bits r acrescentados para representar o sinal transmitido. Esta distribuição dupla da redundância leva naturalmente à formulação da seguinte questão:

Se o factor de sobre-amostragem efectivo for fixo, qual será a melhor distribuição dos bits disponíveis?

Vamos ilustrar este problema com dois exemplos de distribuição dos bits.

Exemplo 3 *Considere-se um sistema de correcção de apagamentos com as seguintes especificações:* $N = 32$, $t = 4$, $K = 24$, $b = 8$, $r = 12$. *O número total de bits a transmitir por bloco é de* $32 \times (8 + 12) = 640$.

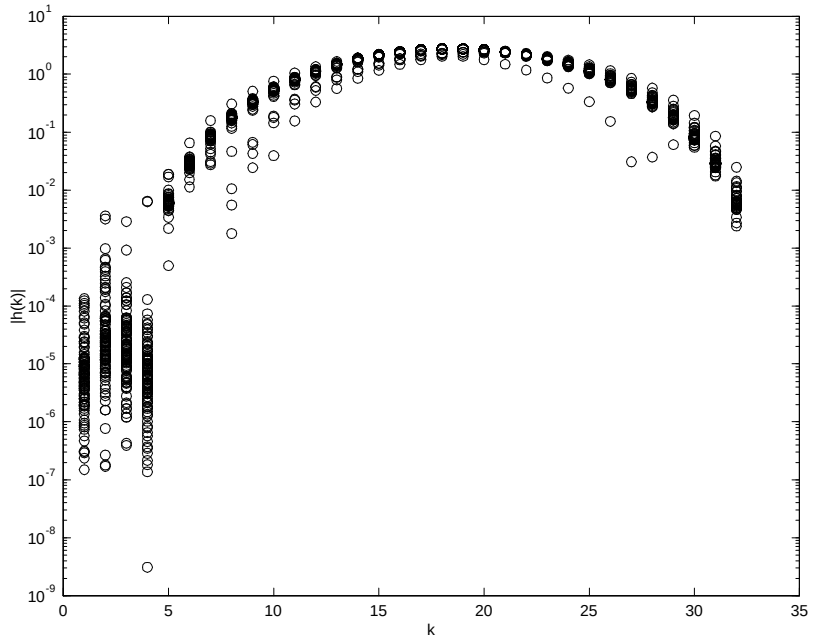


Figura 5.6: Para construir este gráfico sobrepusemos 100 realizações do sinal $\left|\check{h}_k\right|$ (2.31) variando a amplitude do sinal de erro e. A configuração utilizada para o problema de reconstrução foi a seguinte: $N = 32$, $t = 4$, $b = 8$, $r = 20$, $\bar{S}_t = \{1, 2, 3, 4\}$ e $e(\bar{S}_t)$ de amplitude aleatória. Repare-se na fraca separação entre os "zeros" da transformada e as restantes componentes para algumas situações.

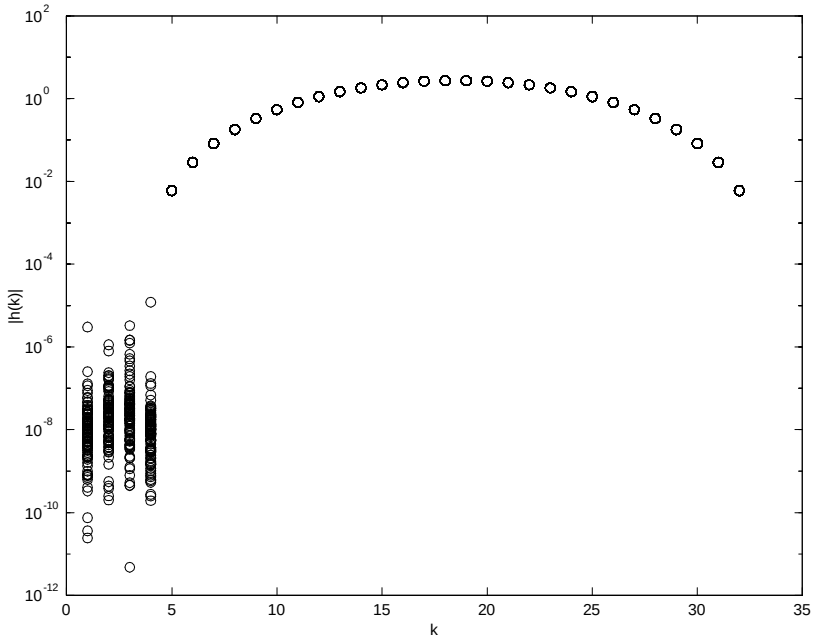


Figura 5.7: Para construir este gráfico sobrepusemos 100 realizações do sinal $\left|\check{h}_k\right|$ (2.31) variando a amplitude do sinal de erro e. A configuração utilizada para o problema de reconstrução é idêntica à da figura anterior à excepção do número de bits $r = 30$.

Exemplo 4 *Considere-se um sistema de correcção de apagamentos com as seguintes especificações:* $N = 40$, $t = 4$, $K = 24$, $b = 8$, $r = 8$. *O número total de bits a transmitir por bloco é de* $40 \times (8 + 8) = 640$.

Repare-se que a única diferença entre estes dois códigos consiste na forma como se distribuíram os bits disponíveis e que para $K = 24$ e $b = 8$, o código gera 640 bits na saída sendo factor de sobre-amostragem efectivo de

$$\beta_{ef} = \frac{256}{640} = 0.4.$$

No entanto, o número de componentes espectrais nulas é maior no segundo caso, levando a que para o mesmo padrão de erro $\bar{S}_t$, o condicionamento $\kappa_\infty (I - S)$ seja menor nesse caso. Os valores obtidos para a situação $\bar{S}_t = \{0, 1, 2, 3\}$, são de $\kappa_\infty (I - S) = 17800$ e $\kappa_\infty (I - S) = 640$, para o primeiro e segundo exemplos respectivamente. Depois de realizadas 1000 simulações de cada um dos exemplos, obtivemos 52 blocos descodificados incorrectamente no primeiro caso e nenhum no segundo.

Esta diferença substancial nos resultados levanta a questão de saber se existirá uma repartição óptima dos bits disponíveis.

Capítulo 6

Conclusões e trabalho futuro

O objectivo inicial deste trabalho, era encontrar um método para detectar as posições das amostras erradas em sinais limitados em frequência. Julgamos que este objectivo foi plenamente alcançado tendo-se encontrado relações com métodos utilizados noutras áreas como é o caso da teoria dos códigos. Pensamos que não seria demais afirmar que o trabalho de síntese efectuado e as relações encontradas entre resultados de diferentes disciplinas como a reconstrução de sinal, teorias dos códigos, entre outros constitui uma panorâmica enriquecedora e por si só um resultado importante deste trabalho.

O problema dos erros de arredondamento que aparece pelo facto de se trabalhar com aritmética real, obrigou a um estudo da estabilidade do problema de reconstrução tendo-se obtido alguns resultados relevantes.

Estes resultados em combinação com os que foram obtidos no capítulo 5, abrem caminho para o projecto prático de códigos de correcção de erros no corpo dos reais, possibilitando a suas utilização eficiente num grande número de aplicações.

6.1 Trabalho futuro

Durante a execução desta tese foram identificados alguns problemas em aberto que pensamos serem merecedores de um estudo futuro.

6.1.1 Códigos convolucionais

A ênfase deste trabalho foi sobre os sinais de dimensão finita tendo-se dado maior atenção aos códigos por blocos. Para os códigos convolucionais conseguimos demonstrar que é possível com um banco de filtros que satisfaça a propriedade de reconstrução perfeita gerar um sindroma que permite detectar e corrigir erros. Pensamos que neste domínio se encontram em aberto os seguintes problemas:

- Continuar o estudo dos códigos convolucionais no corpo dos reais determinando a distância mínima dos códigos em função dos filtros e factor de sobre-amostragem utilizados.
- Pesquisar métodos de descodificação eficientes para os códigos convolucionais no corpo dos reais.
- Os bancos de filtros como códigos convolucionais de correcção de erros podem ser importantes para construir bancos de filtros para codificação com bandas de guarda na frequência que permitam a correcção de erros.

6.1.2 Estabilidade da reconstrução

- Os códigos de correcção de erros designados por "Turbo-Codes" [Divsalar 95, Rothweiler 99], utilizam entrelaçamento das amostras do sinal mensagem na codificação. Propomos o estudo da estabilidade numérica deste tipo de códigos no corpo dos números reais para avaliar o seu desempanho.

- Estudo dos códigos sistemáticos no corpo dos números reais, determinando a estabilidade da solução e limites para o condicionamento numérico das matrizes de reconstrução.

- Estudo da estabilidade dos códigos realizados com a transformada DCT e outras transformadas.

6.1.3 Projecto de códigos com aritmética real

- Comparação da eficiência da codificação dos códigos de correcção de erros no corpo dos reais com os códigos em corpos finitos.

- Teste dos códigos de correcção de erros no corpo dos reais em realizações práticas com diferentes tipos de canais de transmissão.

- Teste dos códigos com aritmética real para protecção contra erros dos ficheiros em computadores. Uma vez que a simples alteração de um bit num ficheiro de um computador pode tornar este inutilizável, esta aplicação dos códigos com aritmética real, coloca problemas de avaliação do erro numérico máximo introduzido pelo algoritmo muito delicados.

- Simplificação das expressões de combinatória obtidas no capítulo 5 para contar o número de padrões de erro com uma dada distância mínima.

- O conjunto dos padrões de erro que possuem uma dada distância mínima é normalmente bastante grande e a gama de valores para o condicionamento do problema a resolver de grande amplitude. Uma das soluções consiste em sub-dividir cada uma destes conjuntos classificando os padrões de erro pelo número de vezes que a distância mínima é satisfeita. Uma expressão de combinatória que conte o número de padrões de erro destes sub-conjuntos, permitirá aperfeiçoar o projecto dos códigos de correcção de erros no corpo dos reais.

Apêndice A

Notação utilizada

Tentou-se definir uma notação única a utilizar durante toda a tese. Praticamente em todo o texto apenas se consideram sinais discretos e com um número finito de amostras, ou seja, vectores, daí a opção pela utilização da notação algébrica uma vez que é mais compacta e de fácil compreensão. Na lista que se segue estão a maior parte das convenções utilizadas.

- x Sinal temporal ou vector.
- x_n Elemento de um sinal. Em determinadas situações, por uma questão de legibelidade do texto poderemos usar igualmente $x\,(n)$ para o mesmo efeito
- $\hat{x}$ Transformada de Fourier do sinal x.
- A matriz;
- A^T Transposta da matriz A;
- A^* Conjugada da matriz A;
- $A^\dagger$ Conjugada da transposta da matriz A;
- F Matriz da transformada de Fourier;
- $x\,(S)$ Elementos de x, cujos índices são dados pelo conjunto de inteiros S.
- $x * y$ Convolução entre os sinais x e y.
- N dimensão do bloco de dados
- K número de amostras conhecidas do sinal.
- $M = N - K$ número de zeros do espectro de um código de correcção de erros
- t número de erros a corrigir.
- $\mathrm{j} = \sqrt{-1}$.

A.1 Siglas utilizadas durante a tese

- BCH- Bose, Chaudhury and Hocquenghem
- CDM- Código de distância máxima
- TCCE- Teoria dos Códigos de Correcção de Erros
- PDS- Processamento Digital de Sinal
- DFT- Discrete Fourier Transform
- FFT- Fast Fourier Transform
- DCT- Discrete Cosine Transform
- ODFT- Odd Discrete Fourier Transform

Bibliografia

[Agarwal 73] R. C. Agarwal e C. S. Burrus, "Fast digital convolution using Fermat transforms". In *Southwest IEEE Conference in Rec.*, pp. 538–543, IEEE, Houston, Texas, Abril 1973.

[Agarwal 74] R. C. Agarwal e C. S. Burrus, "Fast convolution using Fermat number transforms with applications to digital filtering", *IEEE Trans. on Acoustics, Speech and Signal Processing*, 22(2):87–97, Abril 1974.

[Anfinson 88] C. J. Anfinson e F. T. Luk, "A linear algebraic model of algorithm-based fault tolerance", *IEEE Transactions on Computers*, 37(12):1599–1604, Dezembro 1988.

[Bellanger 89] M. Bellanger, *Digital Processing of Signals*. John Wiley & Sons. Inc., 1989.

[Berlekamp 84] E. R. Berlekamp, *Algebraic Coding Theory*. Aegean Park Press, CA - U.S.A., 1984.

[Blahut 79] R. E. Blahut, "Transform techniques for error control codes", *IBM Journal of Research and Development*, 23, Maio 1979.

[Blahut 83] R. E. Blahut, *Theory and Practice of Error Control Codes*. Addison-Wesley, NY, 1983.

[Blahut 85a] R. E. Blahut, "Algebraic fields, signal processing, and error control", *Proceedings of IEEE*, (5):874–893, Maio 1985.

[Blahut 85b] R. E. Blahut, *Fast Algorithms for Digital Signal Processing*. Addison-Wesley, 1985.

[Bolot 95] C. H. Bolot, J. C. e A. U. Garcia, "Analysis of audio packet loss in the internet". In *Proceedings of the 5th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, 1995.

[Bolot 96] J. C. Bolot e A. Vega-Garcia, "Control mechanisms for packet audio in the internet". In *Proceedings INFOCOM 96*, 1996.

[Brent 80] G. F. G. Brent, Richard P. e D. Y. Y. Yun, "Fast solution of Toeplitz systems of equations and computation of Padé approximations", *Journal Algorithms*, 1:259–295, 1980.

[Brualdi 77] R. A. Brualdi, *Introductory Combinatorics*. Elsevier North Holland, Inc., 1977.

[Bultheel 97] A. Bultheel e M. V. Barel, *Linear Algebra, Rational Approximation and Orthogonal Polynomials. Studies in computational mathematics*, Elsevier, Amsterdam, 1997.

[Clark 81] G. C. Clark e J. B. Cain, *Error-Correcting Coding for Digital Communications*. Plenum Press, New York, 1981.

[Divsalar 95] D. Divsalar e F. Pollara, "Turbo codes for PCS applications", *Proceedings of IEEE*, 1995.

[Durbin 60] J. Durbin, "The fitting of time series models", *Rev. Inst. Int. de Stat.*, 28(3):233–244, 1960.

[Ferreira 92] P. J. S. G. Ferreira, *Estudo e Unificação de uma Classe de Problemas de Amostragem, Interpolação e Extrapolação*. PhD thesis, Dep. de Electrónica e Telecomunicações da Universidade de Aveiro, Aveiro, Portugal, 1992.

[Ferreira 94a] P. J. S. G. Ferreira, "Interpolation and discrete Papoulis-Gerchberg algorithm", *IEEE Transactions on Signal Processing*, 42(10):2596–2606, Outubro 1994.

[Ferreira 94b] P. J. S. G. Ferreira, "Non-iterative and fast iterative methods for interpolation and extrapolation", *IEEE Transactions on Signal Processing*, 42(11):3278–3282, Novembro 1994.

[Ferreira 94c] P. J. S. G. Ferreira, "The stability of a procedure for the recovery of lost samples in band-limited signals", *Signal Processing*, 40(3), 1994.

[Ferreira 95] P. J. S. G. Ferreira, A. M. P. Tomé, e J. M. N. Vieira, "On two recent approaches to the reconstruction of signals from nonuniform samples". In *Sampta, Workshop on Sampling Theory & Applications*, pp. 263–267, Jurmala, Latvia, Setembro 1995.

[Ferreira 96] P. J. S. G. Ferreira, "Interpolation in the time and frequency domains", *IEEE Signal Processing Letters*, 3(6):176–178, Junho 1996.

[Ferreira 97] P. J. S. G. Ferreira, "The eigenvalues of matrices which occur in certain interpolation problems", *IEEE Transactions on Signal Processing*, 45(8):2115–2120, Agosto 1997.

[Ferreira 99] P. J. S. G. Ferreira, "The condition number of certain matrices and applications". In *ICASSP 99*, IEEE, Phoenix, Arizona, U.S.A., Maio 1999.

[Forney 65] J. Forney, G. D.., "On decoding BCH codes", *IEEE Transactions on Information Theory*, 11:549–557, 1965.

[Gerchberg 74] R. W. Gerchberg, "Super resolution through error energy reduction", *Opt. Acta*, 21(9):709–720, 1974.

[Golay 49] M. J. E. Golay, "Notes on digital coding", *Proceedings of IRE*, 37:657, 1949.

[Golub 83] G. H. Golub e C. F. V. Loan, *Matrix Computations*. John Hopkins University Press, USA, 1983.

[Gragg 72] W. B. Gragg, "The Padé table and its relation to certain algorithms of numerical analysis", *SIAM Rev.*, 14(1):1–62, 1972.

[Grochenig 93] K. Grochenig, "A discrete theory of irregular sampling", *Linear Algebra and its Applications*, 193:129–150, 1993.

[Hamming 50] R. S. Hamming, "Error detecting and error correcting codes", *Bell Systems Technical Journal*, 29:147–160, 1950.

[Hamming 80] R. W. Hamming, *Coding and Information Theory*. Prentice-Hall, Englewood Cliffs, N. J., 1980.

[Higgins 85] J. R. Higgins, "Five short stories about the cardinal series", *Bull American Math. Soc.*, 12(1):45–89, 1985.

[Hildebrand 56] F. B. Hildebrand, *Introduction to Numerical Analysis*. Dover Publications Inc., New York, 2 edition, 1956.

[Hill 86] R. Hill, *A First Course in Coding Thory*. Oxford University Press Inc., New York, 1986.

[Horn 85] R. A. Horn e C. R. Johnson, *Matrix Analysis*. Cambridge Press, New York, 1985.

[Hu 96] L. C. Hu e K.-J. Lin, "A simulation study of real-time MPEG traffic using forward error control scheme". In *Proceedings of WORDS'96 - Workshop Object-Oriented Real-Time Dependable Systems*, 1996.

[Huang 84] K.-H. Huang e J. A. Abrham, "Algorithm-based fault tolerance for matrix operations", *IEEE Transactions on Computers*, 33(6):518–528, Junho 1984.

[Kailath 86] T. Kailath, "A theorem of I. Schur and its impact on modern signal processing". In I. Gohberg, ed., *I. Schur Methods in Operator Theory and Signal Processing*, pp. 9–90, Birkhäuser Verlag, Germany, 1986.

[Krishna 94] K. B. L. K.-Y. Krishna, H. e J. D. Sun, *Computational Number Theory and Digital Signal Processing*. CRC Press, Florida, 1994.

[Kuijper 97] M. Kuijper e J. C. Willems, "On constructing a shortest linear recurrence relation", *IEEE Trans. on Automatic Control*, 42(11):1554–1558, Novembro 1997.

[Kumaresan 82] R. Kumaresan, "Accurate frequency estimation using an all-pole filter with mostly zero coefficients", *Proceedings of the IEEE*, 70(8):873–875, Agosto 1982.

[Kumaresan 85] R. Kumaresan, "Rank reduction techniques and burst error-correction decoding in real/complex fields". In *Proceedings 19th Asilomar Conference Circuits and Systems*, pp. 457–461, Pacific Grove, CA, Novembro 1985.

[Kuo 90] B. C. Kuo, *Digital Control Systems*. Holt, Rinehart & Winston, 1990.

[Levinson 47] N. Levinson, "The Wiener (root mean square) error criterion in filter design and prediction", *Journal Math. Phys.*, 25:261–278, 1947.

[Marks II 91] R. J. Marks II, *Introduction to Shannon Sampling and Interpolation Theory*. Springer-Verlag, New York, 1991.

[Marple 87] S. L. Marple, Jr., *Digital Spectral Analysis with Applications*. Prentice-Hall, 1987.

[Marshall Jr 79] T. G. Marshall Jr., "Image coders with semidefinite and definite decoders". In *Proc. Soc. Photo-Optic. Instr. Eng.*, pp. 299–307, Agosto 1979.

[Marshall Jr 81] T. G. Marshall Jr., "Real number transform and convolutional codes". In *Proc. 24th Midwest Symp. on Circuits and Systems*, pp. 650–653, Albuquerque, NM, Junho 1981.

[Marshall Jr 82a] T. G. Marshall Jr., "Codes and algorithms for simultaneous error correction and rate reduction implementable with standard digital signal processors". In *Proceedings of IEEE Global Telecommunication Conference*, pp. 936–940, Miami, FL, Novembro 1982.

[Marshall Jr 82b] T. G. Marshall Jr., "Methods for error correction with digital signal processors". In *Proceedings 25th Midwest Symposium on Circuits and Systems*, pp. 1–5, IEEE, Agosto 1982.

[Marshall Jr 82c] T. G. Marshall Jr., "Structures for digital filter banks". In *ICASSP 82*, pp. 315–318, IEEE, Paris, Abril 1982.

[Marshall Jr 83] T. G. Marshall Jr., "Decoding of real-number error-correcting codes". In *Proceedings IEEE Global Telecommunications Conference*, pp. 1249–1303, IEEE, San Diego, Agosto 1983.

[Marshall Jr 84] T. G. Marshall Jr., "Coding of real-number sequences for error correction: A digital signal processing problem", *IEEE Journal on Selected Areas of Communication*, 2(2):381–391, Março 1984.

[Marshall Jr 86] T. G. Marshall Jr., "Codes for error correction based upon interpolation of real number sequences". In *Proceedings 19th Asilomar Conference on Circuits and Systems*, pp. 202–206, Pacific Grove, CA, Novembro 1986.

[Marshall Jr 87a] T. G. Marshall Jr., "Removing noise pulses from frequency constrained signals". In *Proceedings IEEE Int. Conf. Communications*, pp. 997–1000, IEEE, Junho 1987.

[Marshall Jr 87b] T. G. Marshall Jr., "Signal restoration viewpoints for estimating errors in discrete time signals". In *Proceedings 20th Asilomar Conference Circuits, Systems and Computers*, Pacific Grove, California, 1987.

[Marvasti 87] F. A. Marvasti, *A Unified Approach to Zero-Crossings and Nonuniform Sampling of Single and Multidimensional Signals and Systems*. Oak Park, 1 edition, 1987.

[Marvasti 91] F. A. Marvasti, M. Analoui, e M. Gamshadzahi, "Recovery of signals from nonuniform samples using iterative methods", *IEEE Transactions on Signal Processing*, 39(4):872–878, Abril 1991.

[Marvasti 92] F. A. Marvasti e P. M. Clarkson, "Reconstruction of speech signals with lost samples", *IEEE Transactions on Signal Processing*, 40(12):2897–2903, Dezembro 1992.

[Marvasti 93] F. A. Marvasti e M. Nafie, "Sampling theorem: A unified outlook on information theory, block and convolutional codes", *IEICE Tran. on Fundamentals of Electronics Communications and Computer Sciences*, 76(9):1383–1391, Setembro 1993.

[Marvasti 94] F. A. Marvasti, C. Liu, e G. Adams, "Analysis and recovery of multidimensional signals from irregular samples using nonlinear and iterative techniques", *Signal Processing*, 36:13–30, 1994.

[Marvasti 97] F. A. Marvasti e M. Echhart, "Burst correction for missing samples". In *SAMPTA97*, Aveiro, Portugal, Junho 1997.

[Marvasti 99] H. M. Marvasti, Farokh e S. Talebi, "Efficient algorithms for burst error recovery using FFT and other transform kernels", *IEEE Transactions on Signal Processing*, 47(4), Abril 1999.

[Nair 90] V. S. S. Nair e J. A. Abraham, "Real-number codes for fault-tolerant matrix operations on processor arrays", *IEEE Transactions on Computers*, 39(4):426–435, Abril 1990.

[Pad 92] H. Padé, "Sur la representation approché d'une fonction par des fractions rationnelles", *Thesis, Ann. Ecole Normale - supplement*, 3(9):1–93, 1892.

[Papoulis 75] A. Papoulis, "A new algorithm in spectral analysis and band-limited extrapolation", *IEEE Transactions on Circuits and Systems*, 22(9):735–742, Setembro 1975.

[Press 88] a. F. B. P. e. Press, William H., *Numerical Recipes in C*. Cambridge, Cambridge, 1988.

[Prony 95] B. d. Prony, "Essai expérimental et analytique: sur les lois de la dilatabilité de fluides élastiques et sur celles de la force expansive da la vapeurde l'eau et de la vapeur de lalkool, à différentes températures", *J. L'Ecole Polytechnique de Paris*, 1(2):24–76, 1795.

[Robinson 80] E. A. Robinson e S. Treitel, *Geophysical Signal Analysis*. Prentice-Hall, N.J., 1980.

[Rothweiler 99] J. Rothweiler, "Turbo codes - making communications more efficient", *IEEE Potencials*, 23–25, Fevereiro 1999.

[Schur 86a] I. Schur, "On power series which are bounded in the interior of the unit circle - I". In I. Gohberg, ed., *I. Schur Methods in Operator Theory and Signal Processing*, pp. 9–90, Birkhäuser Verlag, Germany, 1986.

[Schur 86b] I. Schur, "On power series which are bounded in the interior of the unit circle - II". In I. Gohberg, ed., *I. Schur Methods in Operator Theory and Signal Processing*, pp. 9–90, Birkhäuser Verlag, Germany, 1986.

[Shannon 48] C. E. Shannon, "A mathematical theory of communication", *Bell Systems Technical Journal*, 27:379–423, 1948.

[Shiu 95] J. Shiu e J.-L. Wu, "Real-number error-control coding as a new technique for nonlinear filtering". In *1995 IEEE Workshop on Nonlinear Signal and Image Processing*, IEEE, Neos Marmaras, Greece, Junho 1995.

[Shiu 96] J. Shiu e J.-L. Wu, "Class of majority decodable real-number codes", *IEEE Transactions on Communications*, 44(3):281–283, Março 1996.

[Sprague 82] D. L. Sprague e T. G. Marshall Jr., "A Hadamard transform code for error correction over the real numbers". In *Proceedings 25th Midwest Symposium on Circuits and Systems*, pp. 9–13, IEEE, Agosto 1982.

[Stewart 73] G. W. Stewart, *Introduction to Matrix Computations*. Academic Press, London, 1973.

[Strang 96] G. Strang e T. Nguyen, *Wavelets and Filter Banks*. Wellesley - Cambridge Press, USA, 1996.

[Sweeney 91] P. Sweeney, *Error Control Coding - An Introduction*. Prentice-Hall, 1991.

[Trench 64] W. F. Trench, "An algorithm for the inversion of finite Toeplitz matrices", *Journal of Society of Industrial and Applied Mathematics*, 12(3):512–522, 1964.

[Vaidyanathan 93] P. P. Vaidyanathan, *Multirate Systems and Filter Banks. Signal Processing*, Prentice-Hall, New Jersey, 1 edition, 1993.

[Wakerly 78] J. Wakerly, *Error-Detecting Codes, Self-Checking Circuits and Applications*. Elsevier, New York, 1978.

[Walsh 96] D. P. A. Walsh, D. O. e M. W. Marcellin, "Non-iterative implementation of a class of iterative signal restoration algorithms". In *ICASSP 96*, pp. 1672–1675, IEEE, 1996.

[Walsh 98] B. K. G. M. Walsh, D. O., "Non-iterative implementation of the gerchberg algorithm", 1998.

[Wolf 67] J. K. Wolf, "Decoding of Bose-Choudhuri-Hocquenhem codes and Prony's method of curve fitting", *IEEE Transactions on Communications*, 13:608, Outubro 1967.

[Wolf 83] J. K. Wolf, "Redundancy, the discrete Fourier transform, and impulse noise cancellation", *IEEE Transactions on Communications*, 31(3):458–461, Março 1983.

[Wong 95] C. K. W. Wong, F. Marvasti, e W. G. Chambers, "Implementation of recovery of speech with impulsive noise on a DSP chip", *Electronic Letters*, 31(17):1412–1413, Agosto 1995.

[Wu 92] J.-L. Wu e J. Shiu, "Real-valued error control coding by using DCT", *Proceedings of IEE*, 139(2):133–139, Abril 1992.

[Wu 95] J.-L. Wu e J. Shiu, "Discrete cosine transform in error control coding", *IEEE Transactions on Communications*, 43(5):1857–1861, Maio 1995.

[Zadeh 93a] H. S. Zadeh e A. E. Yagle, "A fast algorithm for extrapolation of discrete band-limited signals". In *ICASSP*, pp. 591–594, IEEE, Abril 1993.

[Zadeh 93b] H. S. Zadeh e A. E. Yagle, "A fast algorithm for extrapolation of discrete-time periodic band-limited signals", *Signal Processing*, 33(2):183–196, Agosto 1993.

[Zhang 89] H.-M. Zhang e P. Duhamel, "Doubling Levinson/Schur algorithm and its implementation". In *Proceedings ICASSP 89*, pp. 1115–1118, 1989.

[Zhang 92] H.-M. Zhang e P. Duhamel, "On the methods for solving Yule-Walker equations", *IEEE Transactions on Signal Processing*, 40(12):2987–3000, Dezembro 1992.