CRISTHOF JOHANN ROOSEN RUNGE

DECODIFICAÇÃO TURBO DE CÓDIGOS DE REED-SOLOMON EM SISTEMAS DE MODULAÇÃO QAM: UMA ABORDAGEM GEOMÉTRICA

São Paulo

2012

CRISTHOF JOHANN ROOSEN RUNGE

DECODIFICAÇÃO TURBO DE CÓDIGOS DE REED-SOLOMON EM SISTEMAS DE MODULAÇÃO QAM: UMA ABORDAGEM GEOMÉTRICA

São Paulo

2012

CRISTHOF JOHANN ROOSEN RUNGE

DECODIFICAÇÃO TURBO DE CÓDIGOS DE REED-SOLOMON EM SISTEMAS DE MODULAÇÃO QAM: UMA ABORDAGEM GEOMÉTRICA

Tese apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do título de Doutor em Ciências.

Área de concentração:
Engenharia de Sistemas

Orientador: Prof. Dr. José Roberto Castilho Piqueira

São Paulo

2012

FICHA CATALOGRÁFICA

Runge, Cristhof Johann Roosen
Decodificação turbo de códigos de Reed-Solomon em sistemas de modulação QAM: uma abordagem geométrica / C.J.R. Runge. -- São Paulo, 2012.
130 p.

Tese (Doutorado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Telecomunicações e Controle.

1. Teoria da informação e comunicação 2. Codificação 3. Algoritmos 4. Modulação digital I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Telecomunicações e Controle II. t.

DEDICATÓRIA

Dedico este trabalho a minha esposa
Marisa e a minha filhota Eduarda.

AGRADECIMENTOS

Ao Professor Dr. José Roberto Castilho Pereira, pela orientação, paciência, e constante apoio para que eu continue seguindo em frente.

Ao Professor Dr. Ricardo Paulino Marques, pelas correções e valiosas sugestões para a melhoria deste trabalho.

Ao Professor Dr. Fuad Kassab Junior, pelas correções e valiosas sugestões para a melhoria deste trabalho.

A todos aqueles que direta ou indiretamente contribuíram para a elaboração deste trabalho.

A mente que se abre a uma nova ideia
jamais voltará ao seu tamanho original.
(Albert Einstein)

RESUMO

Este trabalho investiga a decodificação turbo de códigos produto construídos a partir de códigos de Reed-Solomon sobre constelações QAM. Por meio da geometria euclidiana da constelação utilizada e das relações dos elementos de campo de Galois utilizados na codificação Reed-Solomon com os símbolos pertencentes ao alfabeto de modulação, é proposto um algoritmo de decodificação que utiliza a localização geométrica dos símbolos recebidos no processo de decodificação turbo.
Tanto a primeira etapa da decodificação SISO baseada no algoritmo de Chase, como a extração da informação extrínseca baseada no algoritmo proposto por Pyndiah, são tratadas pelo uso de uma abordagem geométrica sobre o espaço euclidiano. Os resultados de simulação utilizando o algoritmo proposto coincidem com aqueles usando a decomposição e análise pragmática binária, sendo que a abordagem apresentada conduz a simplificações e otimizações em relação à metodologia binária no que ser refere à implementação dos possíveis esquemas de decodificação.

Palavras-chave: Decodificação Turbo. Reed-Solomon. QAM. BTC. Código produto. Espaço euclidiano.

ABSTRACT

This work investigates the turbo decoding of product codes built using Reed-Solomon codes in QAM constellations. Using the euclidian geometry of the constellation and the relations of the Galois field elements used in the Reed-Solomon code and the modulation alphabet, a decoding algorithm for the turbo decoding process is proposed using the geometric localization of the received symbol.

Both the first stage of the SISO decoding based on the Chase algorithm, as the calculation of the extrinsic information based on the Pyndiah algorithm, are treated using the geometric approach in the euclidian space. The simulations show that this algorithm gives similar performance results as the pragmatic binary decomposition approach, and leads to simplifications and optimizations in decoding implementations schemes compared to the former.

Keywords: Turbo Decoding. Reed-Solomon. QAM. BTC. Product Code. Euclidian space.

LISTA DE SÍMBOLOS

$\oplus$	operação soma módulo 2
$\exp(\ )$	função exponencial.
$\log(\ )$	função logaritmo.
$\cosh(\ )$	função cosseno hiperbólico
e	algarismo de Euler.
$erfc(\ )$	função erro complementar.
$E[x]$	função esperança da variável x
No	densidade espectral de potência associada ao ruído branco.
$p(y \mid x)$	probabilidade condicional da variável y em relação a variável x.
dE	distância euclidiana associada a menor distância entre dois símbolos de modulação em um espaço euclidiano.
$\lfloor x \rfloor$	maior inteiro menor que x.
$\binom{n}{k}$	combinação de n elementos tomados k a k.

d_{min} distância mínima de Hamming

$Sa(x)$ função *Sampling*

$GF(p^m)$ Campo de Galois sobre p^m, sendo p um número primo e m um número inteiro.

LISTA DE ABREVIATURAS E SIGLAS

QAM	Quadrature Amplitude Modulation
SISO	Soft Input Soft Output
BTC	Block Turbo Codes
FSK	Frequency Shift Keying
PSK	Phase Shift Keying
AWGN	Additive White Gaussian Channel
BCH	Bose, Chaudhuri and Hocquenghem
RS	Reed-Solomon
OWEF	Output Weight Enumerating Function
CWEF	Conditional Weight Enumerating Function
IOWEF	Input Output Weight Enumerating Function
MAP	Maximum a Posteriori
ML	Maximum Likelihood
LLR	Logarithm Likelihood Ratio
PML	Pseudo Maximum Likelihood
BER	Bit Error Rate

LISTA DE FIGURAS

Figura 1 – Diagrama em blocos de um sistema de comunicação digital 24
Figura 2 - O canal de transmissão. 26
Figura 3- Processo gaussiano branco. 27
Figura 4 – Ruído gaussiano de faixa limitada. 28
Figura 5- O canal gaussiano discreto. 29
Figura 6 - Representação de uma constelação 16-QAM no espaço euclidiano. . . . 31
Figura 7 – O receptor de correlação. 32
Figura 8 – Arranjo matricial do código produto $C_1 x C_2$. 34
Figura 9 – Efeito de embaralhamento na matriz produto. 36
Figura 10 - O canal de transmissão. 37
Figura 11 – Demodulador soft. 37
Figura 12- Localização da k -ésima posição na matriz produto. 38
Figura 13 - Diagrama em blocos do sistema de comunicação. 44
Figura 14- Representação geométrica para o decodificador com medida de informação de canal. 47
Figura 15 – Decodificação *soft*, desempenho código Galois (24,12) em um canal gaussiano.. 51
Figura 16 – Decodificador elementar. 58
Figura 17- Esquema para ciclo completo de decodificação turbo. 59
Figura 18- Matriz produto associada ao símbolo não binário. 62
Figura 19 - Decodificação *soft* utilizando abordagem clássica e abordagem geométrica para um mesmo número de vetores de teste. 65
Figura 20 - Decodificação *soft* utilizando abordagem clássica e geométrica para um mesmo resultado de desempenho. 66
Figura 21- Constelação 16-QAM. Escolhas possíveis de símbolos para formar as sequências candidatas. 67
Figura 22- Constelação 16-QAM. Limites do espaço euclidiano utilizado no processo de decodificação. 69
Figura 23- Eventos de erro no espaço euclidiano. 73
Figura 24 – Um passo de decodificação SISO. 78

Figura 25 – Decodificação turbo RS(15,13)^2 – 16-QAM. 81
Figura 26 – Decodificação turbo RS(255,253)^2 – 256-QAM. 82
Figura 27- Evolução no espaço euclidiano para seis passos de iteração utilizando abordagem geométrica. 84
Figura 28– Resultados comparativos do uso das métricas euclidiana e euclidiana quadrática – 16-QAM. 86
Figura 29– Resultados comparativos do uso das métricas euclidiana e euclidiana quadrática - 256-QAM. 87
Figura 30 - Degradação causada pela quantização no cálculo da distância euclidiana. 89
Figura 31 – Influência de β no desempenho do algoritmo de decodificação. 91
Figura 32 – Medida empírica dos eventos de erro para um esquema RS(15,13)^2 16QAM. 93
Figura 33 – Resultados de desempenho para diferente número de iterações. 94

SUMÁRIO

1 INTRODUÇÃO 17

1.1 INTRODUÇÃO. 17

1.2 REVISÃO DA LITERATURA. 18

1.3 O SISTEMA DE COMUNICAÇÃO. 23

1.3.1 O canal de transmissão. 25

1.3.1.1 O canal gaussiano discreto. 27

1.3.2 O sistema de modulação QAM. 29

1.3.2.1 Interpretação geométrica dos sinais. 30

1.3.2.2 O receptor de correlação. 31

2 A CODIFICAÇÃO TURBO. 34

2.1 INTRODUÇÃO. 34

2.2 CODIGOS PRODUTO. 34

2.2.1 Concatenação utilizando códigos produto. 35

2.3 ALGORITMOS DE DECODIFICAÇÃO TURBO. 36

2.3.1 O demodulador *soft*. 37

2.3.2 A decisão utilizando a máxima probabilidade a *posteriori*. 38

2.3.3 A decodificação Turbo. 41

2.4 O ALGORITMO DE CHASE. 43

2.4.1 Classe de algoritmos utilizando decisão *Soft*. 45

2.4.1.1 Algoritmo I. 48

2.4.1.2 Algoritmo II. 49

2.4.1.3 Algoritmo III. 50

2.5 O ALGORITMO DE DECODIFICAÇÃO PROPOSTO POR PYNDIAH. . 51

2.5.1 A implementação da decodificação turbo utilizando o algoritmo proposto por Pyndiah. 55

2.5.2 O processo iterativo de decodificação turbo. 57

3 A DECODIFICAÇÃO TURBO UTILIZANDO A ABORDAGEM GEOMÉTRICA. 60

3.1 INTRODUÇÃO. 60

3.2 OS CÓDIGOS DE REED-SOLOMON E O ARRANJO NA MATRIZ PRODUTO. 61

3.3 O DECODIFICADOR SISO. 62

3.3.1 O algoritmo de Chase utilizando a abordagem geométrica. 63

3.3.2 O cálculo da informação extrínseca. 67

3.3.2.1 A justificativa do fator γ. 71

3.4 O ALGORITMO DE DECODIFICAÇÃO SISO. 74

3.5 O PROCESSO DE DECODIFICAÇÃO TURBO. 77

4 RESULTADOS DE SIMULAÇÃO. 79

4.1 INTRODUÇÃO. 79

4.2 RESULTADOS DE SIMULAÇÃO. 79

4.2.1 Resultados de simulação utilizando o algoritmo para modulação 16-QAM. 79

4.2.2 Resultados de simulação utilizando o algoritmo para modulação 256-QAM. 81

4.2.3 Resultados gráficos dos passos de iteração. 83

4.2.4 A utilização da distância euclidiana e da distância euclidiana quadrática no cálculo da informação extrínseca. 85

4.2.4.1 Considerações de implementação quanto à utilização da distância euclidiana. 88

4.2.5 A influência da variação do vetor β no algoritmo de decodificação. 90

4.2.6 Considerações sobre a determinação de γ 92

4.2.7 Considerações sobre o número de iterações 94

4.3 CONSIDERAÇÕES SOBRE A IMPLEMENTAÇÃO DO ALGORITMO TURBO GEOMÉTRICO. 95

CONCLUSÃO. 97

REFERÊNCIAS. 98

APÊNDICE A. 102

INTRODUÇÃO

1.1 INTRODUÇÃO

Nas últimas décadas, a indústria de telecomunicações tem observado um salto significativo na eficiência e na capacidade de seus sistemas, impulsionada por uma crescente demanda de consumo que por sua vez é realimentada pelos novos serviços que são disponibilizados pelas novas tecnologias.

A transmissão da informação por meio de sistemas de telecomunicações pode envolver diversos estágios, tanto de natureza física, para a conformação dos sinais utilizados, como de natureza algorítmica para o estabelecimento de características desejáveis, sendo a maximização da eficiência no estabelecimento da comunicação para um dado canal de comunicação, o principal critério no projeto e dimensionamento dos diversos estágios que compõem um sistema.

O projeto e fabricação de sistemas reais e comerciais podem sofrer restrições na sua implementação, tanto no aspecto físico como algorítmico, devido às limitações impostas pelas tecnologias disponíveis.

A constante evolução tecnológica na fabricação de circuitos integrados tem proporcionado ganhos vertiginosos, tanto na velocidade, quanto na capacidade de processamento, permitindo a implementação de sistemas que utilizam algoritmos antes considerados proibitivos. Dessa forma, a evolução tecnológica, ao mesmo tempo em que proporciona ganhos de velocidade e capacidade para circuitos integrados, proporciona uma potencialização adicional desses ganhos, uma vez que viabiliza a utilização de algoritmos mais eficientes.

Sendo os códigos corretores de erro uma das principais ferramentas para o aumento da eficiência em sistemas de transmissão, e sendo sua implementação possível com circuitos digitais, a utilização de sistemas codificados com algoritmos cada vez mais complexos e eficientes tem sido uma realidade.

Este trabalho está focado em um estudo sobre esquemas de codificação de canal, que podem ser vistos como um dos estágios componentes da maioria dos sistemas de telecomunicações atuais.

Para a obtenção de esquemas de codificação eficientes, um estudo matemático do sistema proposto é apresentado e analisado. O objetivo é a construção de esquemas de codificação e decodificação utilizando técnicas turbo.

Nesse trabalho, inicialmente são apresentados alguns conceitos e fundamentos necessários à caracterização do sistema de transmissão e à caracterização do canal de ruído aditivo branco *Additive White Gaussian Noise* (AWGN), utilizado para transmissão dos símbolos de modulação, a seguir é realizada a descrição do sistema.

É utilizado um sistema de modulação *Quadrature Amplitude Modulation* (QAM), com códigos produto e Reed-Solomon (RS), para a formação da matriz de codificação. São apresentados os algoritmos de Chase e Pyndiah que são base para a construção do algoritmo de decodificação geométrico.

A partir de considerações geométricas do alfabeto de modulação utilizado, o algoritmo de decodificação geométrico, é proposto e analisado matematicamente. Os resultados obtidos com o uso do novo algoritmo são apresentados, mostrando a eficiência desses esquemas de codificação. O trabalho termina com as considerações de implementação física e as conclusões.

1.2 REVISÃO DA LITERATURA

Em 1993 Claude Berrou, Alain Glavieux e Punya Thitimajshima publicaram o trabalho intitulado "*Near Shannon limit error-correcting coding and decoding; Turbo-Codes*" [1], no qual os Códigos Turbo foram apresentados pela primeira vez como uma nova classe de códigos convolucionais, referenciados como *Convolutional Turbo Codes* (CTC), capazes de apresentar um desempenho próximo ao limite de capacidade de canal formulado por Claude E. Shannon em seu trabalho de 1948, "*A Mathematical Theory of Communication*" [2]. O desempenho apresentado para essa nova classe de códigos, jamais havia sido atingido pelos esquemas de codificação conhecidos.

O esquema de codificação apresentado em [1], consistia na utilização de dois códigos convolucionais recursivos de taxa de codificação r = ½, concatenados de forma paralela e utilizando um entrelaçamento (*interleaving*) de comprimento 65536

em um dos códigos constituintes. Por meio de um algoritmo iterativo, e após 18 iterações, esse esquema apresentou desempenho, para uma taxa de erro de bits de $1x10^{-5}$, distante apenas 0,7dB do limite de capacidade de canal de Shannon.

O conceito central apresentado e utilizado pelos códigos Turbo consiste no uso da informação extrínseca, obtida a cada iteração, e baseada no cálculo das estatísticas da informação. Essa informação extrínseca é repassada como informação a priori para utilização a cada novo passo de iteração.

Apesar dos resultados de desempenho apresentados em [1], uma série de questões relativas aos códigos turbo ficaram em aberto. Isso motivou pesquisadores de todo o mundo para a investigação de diferentes aspectos referentes a esta nova classe de códigos.

Em 1998 Ramesh Mahendra Pyndiah apresentou em [3] o detalhamento da extensão da aplicação do conceito de informação extrínseca utilizada nos códigos turbo convolucionais para utilização em códigos de bloco, utilizando o termo *Block Turbo Codes* (BTC) para referenciar esses esquemas. Nesse trabalho foi proposta a concatenação serial de dois códigos de bloco, pela construção de uma matriz produto, formando assim um código produto.

O algoritmo de decodificação proposto em [3] divide-se em duas etapas principais. A primeira etapa consiste em um decodificador suave (*soft*) subótimo. Foi proposto a utilização do algoritmo II de Chase [4], para implementação do decodificador como forma de simplificar e limitar o número de palavras candidatas, conforme justificado em [4]. Como resultado, esse algoritmo de decodificação produz um conjunto de palavras código candidatas a partir do vetor recebido, e uma delas é eleita vencedora utilizando-se o critério de máxima verossimilhança.

A palavra eleita e o conjunto de palavras candidatas formado na primeira etapa são utilizados no cálculo da informação extrínseca, o processo de decodificação inicia então sua segunda etapa.

Na segunda etapa, o cálculo da informação extrínseca é realizado bit a bit, e uma medida de confiabilidade é atribuída a cada bit da palavra código eleita na primeira etapa, baseada nas estatísticas associadas ao conjunto das palavras candidatas. Esse processo é aplicado a todos os vetores recebidos da matriz produto, primeiro no sentido das linhas/ colunas, e uma vez finalizado em um dos sentidos, a informação extrínseca gerada é utilizada como informação a priori para o processo de decodificação, no sentido das colunas/linhas no próximo passo de iteração.

O processo descrito é aplicado sucessivamente, até que algum critério de desempenho definido seja atingido.

A proposta apresentada por Ramamesh Pyndiah em [3] também abriu caminho para uma série de investigações de outros pesquisadores, no que se refere à verificação de desempenho para diferentes códigos, para variantes do algoritmo proposto, para diferentes esquemas de modulação, para diferentes critérios de parada e de atualização/correção de coeficientes, para diferentes abordagens de decodificação *soft*, além de investigações de estratégias de simplificação dos esquemas de decodificação para sistemas reais.

Em sistemas de comunicação que possuem restrições quanto à largura de banda e que necessitam de elevada eficiência espectral, o uso de códigos de alta taxa se justifica fortemente, sendo mandatório em alguns casos. Um dos grandes apelos na utilização dos BTC está na eficiência de sua aplicação em sistemas que necessitam de altas taxas de codificação, nos quais esse nível de eficiência não é conseguido com os CTC, conforme apresentado em [3] e [5]. Em [5] há uma interessante consideração sobre o papel da matriz produto como entrelaçador (*interleaving*) natural no caso dos BTC, em contra partida à necessidade da utilização de longos entrelaçadores no caso dos CTC, isso simplifica o processo de decodificação. Além disso, os códigos de bloco de alta taxa possuem a vantagem de, em geral, demandarem decodificadores relativamente simples.

No que tange à questão dos códigos constituintes utilizados, no trabalho apresentado por Ramamesh Pyndiah em [3], foram utilizados códigos de Bose Chaudhuri Hocquenghem (BCH), como códigos constituintes para um esquema de modulação *Quaternary Phase Shift Keying* (QPSK). Em [6] e [7] são apresentadas soluções comerciais utilizando códigos de Hamming estendidos ou ainda códigos de paridade como códigos constituintes. Em especial, grande atenção tem sido dada à utilização de códigos RS como códigos constituintes, conforme observado no trabalho publicado posteriormente pelo próprio Ramamesh Pyndiah em [8] e [9], e nos trabalhos [10], [11], [12].

Em [10] há uma análise das vantagens da utilização dos códigos de RS, devidas às suas propriedades algébricas e à sua natureza não binária, que conduzem a uma simplificação nos algoritmos de decodificação. Além disso, considerações sobre a complexidade requerida para esquemas que utilizam RS com capacidade de correção de mais de um erro ($t>1$) também são apresentadas.

Nesse trabalho, as vantagens em relação à utilização de códigos BCH com taxas de codificação equivalentes são discutidas, do ponto de vista de implementação. Em [11], há um estudo de caso para diferentes esquemas de modulação em um sistema *Very high speed digital subscriber line* (VDSL), utilizando códigos RS como códigos constituintes, mostrando suas vantagens em relação à abordagem clássica.

Outras aplicações práticas de BTC utilizando códigos RS como códigos constituintes podem ser vistas em [9], onde é relatado um experimento real de um enlace de comunicação utilizando ondas acústicas embaixo da água, conduzido com sucesso. Em [8] foi implementado um protótipo para estabelecimento de transmissão óptica a 10Gbps, em uma rede *Passive Optical Network* (PON), utilizando RS como códigos constituintes em um esquema de codificação BTC. São apresentadas, também, considerações para implementação de um sistema real operando a 40Gbps para uma *Optical Transmission Network* (OTN), mostrando a viabilidade desse tipo de implementação para sistemas de alta taxa de transmissão.

Os códigos RS apresentam inúmeros atrativos para sua utilização em esquemas BTC, por serem códigos já amplamente estudados e mapeados, possuindo algoritmos de decodificação conhecidos, eficientes e simples. Por serem códigos não binários apresentam diversas propriedades que podem ser exploradas, como por exemplo, a concatenação simbólica ao invés da binária, conforme apresentado em [10], a manipulação direta sobre o elemento de campo conforme sugerido em [13], ou ainda a sua associação aos símbolos de modulação como sugerido em [14] e [15].

Adicionalmente, diversos estudos e trabalhos sobre algoritmos de decodificação *soft* de códigos RS e de códigos produto utilizando RS como códigos constituintes já foram publicados, tais como os algoritmos de Reddy-Robinson, Guruswami-Sudan, Kotter-Vardy e Jiang - Narayanan. Discussões sobre a aplicabilidade, restrições e propostas de esquemas de decodificação *soft* baseados em alguns destes algoritmos são apresentadas em [16], [17] e [18]. Implementações práticas de códigos produto, utilizando RS em sistemas de armazenagem de dados para DVDs, são apresentados [19]. A decodificação *soft* constitui a primeira etapa da decodificação de BTC, sendo, portanto, os trabalhos descritos de relevância na escolha do algoritmo de decodificação a ser utilizado. Sua escolha está associada à facilidade de implementação e aos requisitos impostos pela natureza da aplicação, tais como tempo e capacidade de processamento.

Outra questão relevante na escolha do método de decodificação, quando códigos RS são utilizados, é quanto à abordagem binária ou simbólica, tanto no processo de concatenação, quanto no processo de decodificação.

Em [10], é sugerido que a concatenação binária apresenta um desempenho levemente superior à concatenação simbólica. Entretanto, a complexidade associada àquela não justifica a sua utilização na maioria dos casos práticos, em especial quando a capacidade de correção associada à decisão abrupta (*hard)* é maior que 1. Resultados obtidos através de decomposição binária também são apresentados em [20].

Em [21] um estudo comparativo entre a distância mínima "d_{min}" da imagem binária e a distância mínima simbólica "D" para alguns códigos RS, é mostrado. Para taxas de codificação $r<1/2$, é possível ter $d_{min}>D$, o que pode levar a ganhos de desempenho. No entanto, esse fato não se aplica igualmente para RS de altas taxas. A validade desses resultados está vinculada à escolha de uma base apropriada que mapeie o código no seu equivalente binário, o que constitui um problema em aberto, sendo que os resultados de simulação apresentados em [20] para códigos RS produto não mostram ganhos apreciáveis que justifiquem tal abordagem.

Em [13] é mostrado que a decodificação utilizando a abordagem simbólica pode apresentar ganhos em relação à abordagem binária, sendo sugerido que a não observação de ganhos em trabalhos anteriores [22] e [23], se deve à métrica *Log Likelihood* (LLR) utilizada.

Observa-se que a grande maioria dos trabalhos apresentados aborda a aplicação da decodificação turbo dos BTC, utilizando modulação *Binary Phase shift Keying* (BPSK), ou ainda quaternária (QPSK), [3], [4], [10]. Para o cálculo da informação extrínseca, correspondente ao segundo passo da decodificação, independente da modulação utilizada, a abordagem adotada é a binária pragmática. Assim, a informação referente à constelação e à geometria dos símbolos recebidos não é considerada.

Em trabalhos nos quais são verificados os resultados de desempenho para sistemas de modulação de mais alta cardinalidade como em [11] e [24], a abordagem também é a pragmática binária. Alguns trabalhos apresentam considerações em relação à modulação utilizada, como em [25] e [26], mas são trabalhos dedicados ao estudo de CTC. Nesses estudos, entretanto, são apresentados ganhos de desempenho

quando as informações referentes à constelação utilizada é levada em consideração como parte do esquema de codificação/decodificação.

Em [27] foi apresentada uma tentativa de mapeamento, porém limitada para sistemas 16 QAM e 64 QAM, mas ganhos de desempenho também foram relatados quando a geometria é levada em consideração.

Com base nas considerações e estudos como os anteriores mencionados, este trabalho aborda a proposta de codificação/decodificação turbo de códigos produto, tendo como códigos constituintes códigos de Reed-Solomon com t=1, e adotando uma abordagem geométrica que busca levar em consideração a geometria do espaço euclidiano dos símbolos de modulação em sistemas QAM, na construção do esquema de codificação/decodificação.

Fazendo uso da abordagem vetorial dos símbolos recebidos do canal de comunicação dentro do espaço euclidiano, a abordagem geométrica é utilizada tanto na primeira etapa do algoritmo para a implementação do decodificador *soft*, quanto na segunda etapa do algoritmo, na qual é realizado o cálculo da informação extrínseca.

Diferentemente da abordagem convencional pragmática binária, este trabalho faz uso da abordagem geométrica. Tal tratativa possibilita o uso da informação adicional fornecida pela posição do vetor recebido dentro do espaço euclidiano, permitindo a otimização dos algoritmos de decodificação e a simplificação da implementação, conforme será mostrado.

1.3 O SISTEMA DE COMUNICAÇÃO

Embora várias definições e arquiteturas possam ser utilizadas na caracterização e no estabelecimento do transporte da informação, para o presente estudo é definido um sistema de comunicação como aquele capaz de estabelecer a comunicação entre dois pontos geograficamente separados.

Para se discutir os principais estágios que compõem um sistema de comunicação digital, é considerado inicialmente o diagrama em blocos da figura 1, que constitui uma das possíveis representações de um sistema digital de comunicação.

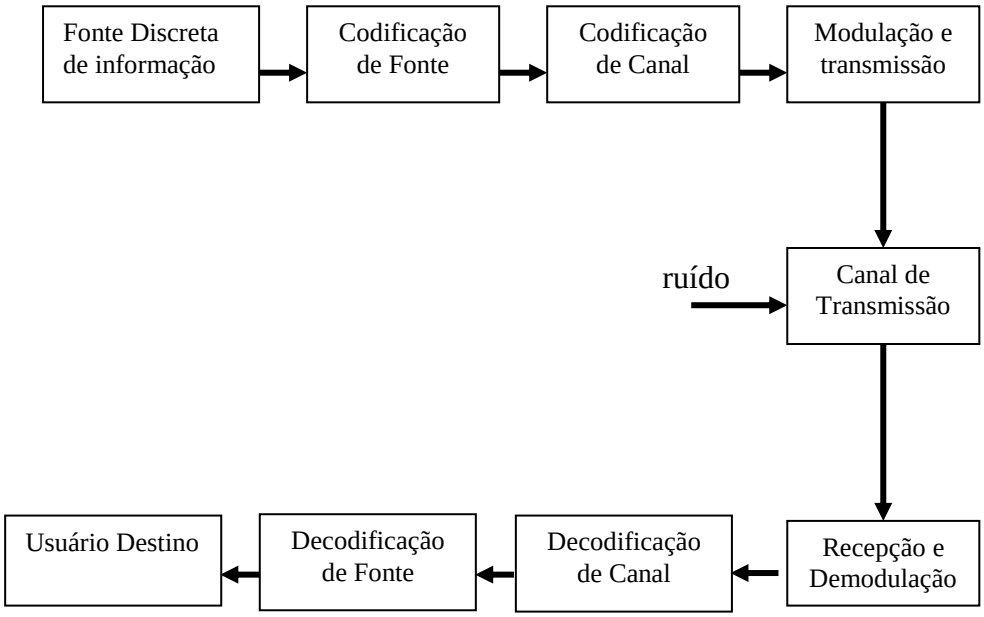


Figura 1 – Diagrama em blocos de um sistema de comunicação digital. Fonte: Autor

Durante a análise, é considerado que a informação da fonte é disponibilizada na forma binária, sendo que a informação a ser enviada no canal passa por três grandes estágios de tratamento e conformação. Primeiramente a informação é codificada por um codificador de fonte. O objetivo na utilização do codificador de fonte é eliminar ou, ao menos reduzir, a redundância presente na representação binária da fonte de informação, melhorando a eficiência da representação da informação. Isso representa uma redução na banda necessária para a transmissão da informação.

O segundo bloco no sistema de transmissão aqui representado é o codificador de canal. Esse bloco, ao contrário do codificador de fonte, irá introduzir redundância na informação fornecida na saída do codificador de fonte. Nesse caso, a contra partida ao aumento na banda requerida é a proteção contra erros causados durante o processo de transmissão. Os códigos corretores de erro utilizados nos codificadores de canal têm seu dimensionamento normalmente atrelado ao tipo de canal através do qual a informação é transmitida, permitindo diversas parametrizações.

O terceiro bloco na transmissão é o modulador, que tem como objetivo conformar o sinal de forma eficiente para a transmissão no canal. Neste trabalho é considerado que este bloco desempenha a função de conformar a sequência binária na saída do codificador de canal em um sinal analógico em banda base, e então modular o sinal usando a técnica de modulação especificada.

Diversas técnicas de modulação digital são possíveis, pelo chaveamento em amplitude, fase ou frequência do sinal a ser transmitido, dando origem a diversas modalidades conhecidas de modulação, tais como *Frequency Shift Keying* (FSK), *Phase Shift Keying* (PSK), QAM, entre outras. A escolha de uma ou outra técnica de modulação é função de diversos critérios, dentre eles complexidade de implementação, características do canal, e frequência de transmissão.

Uma vez codificado e modulado, o sinal é enviado através do canal de transmissão, que pode ser normalmente modelado como um filtro, podendo provocar distorções no sinal recebido, além de adicionar ruído à informação.

Na recepção temos então os blocos que desempenham as funções inversas às operações realizadas pelo sistema digital na transmissão. Ou seja, o sinal recebido é demodulado, operação em que a forma de onda é reconvertida para a banda base e a informação na forma binária correspondente é fornecida ao decodificador de canal, que tem como função eliminar ou minimizar os erros introduzidos durante a transmissão. Por fim no decodificador de fonte é realizado o mapeamento da sequência decodificada na sequencia original inicialmente fornecida pela fonte discreta de informação, finalizando o processo de comunicação.

Um sistema digital de comunicação pode não conter um ou mais dos blocos descritos, ou ainda apresentar outros blocos que não os relacionados.

A seguir serão apresentados alguns conceitos importantes que serão utilizados durante o desenvolvimento deste trabalho.

1.3.1-O canal de transmissão

O canal de transmissão constitui o meio pelo qual a informação é transmitida. Do ponto de vista matemático, os modelos dos canais de transmissão têm como objetivo representar os fenômenos aos quais o sinal físico a ser transmitido estará sujeito durante o envio da informação. Dentre esses fenômenos podemos citar efeitos de multi-percurso, interferência, ruídos originados por diferentes fontes tais como ruído térmico e impulsivo.

Dessa forma, de acordo com as características do canal e do sistema de codificação e modulação, é realizada a construção de um modelo matemático, a fim de permitir a descrição e análise do sistema.

Aqui o canal de transmissão é modelado como um sistema de banda limitada de B Hertz, e perturbado pela presença de ruído gaussiano aditivo, que se soma ao sinal transmitido. O modelo de canal utilizado é também conhecido como canal AWGN. Esse modelo de canal descreve de forma satisfatória grande parte dos sistemas de comunicação (e.g. ópticos, rádio com visada, etc.), representando uma boa aproximação para grande parte dos sistemas reais de transmissão.

A figura 2 é uma representação deste tipo de canal, no qual $s(t)$ representa o sinal a ser transmitido, $n(t)$ o processo estocástico que caracteriza o ruído aditivo, e $r(t)$ o sinal na entrada do receptor.

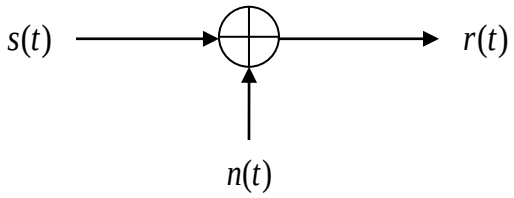


Figura 2 - O canal de transmissão. Fonte: Autor

dessa forma tem-se que:

$$r(t) = s(t) + n(t) \quad (1)$$

O ruído é um processo estocástico, tendo uma função densidade de probabilidade associada, sendo o ruído gaussiano branco um processo estocástico estacionário no sentido estrito. Este é totalmente caracterizado pelos seus momentos de primeira e segunda ordem, ou seja, pela sua média e variância. O ruído gaussiano branco apresenta ainda média μ nula, e suas amostras são totalmente descorrelacionadas para $\tau \neq 0$, sendo τ o intervalo de tempo entre duas amostras.

Este processo é modelado como de natureza impulsiva, sendo sua função de autocorrelação $R(\tau)$ modelada como um impulso com valor $No/2$, para $\tau = 0$, uma vez que suas amostras são totalmente descorrelacionadas para $\tau \neq 0$. Dessa forma,

a sua densidade espectral de potência $S(f)$ é constante. A figura 3 ilustra essas propriedades do processo gaussiano branco.

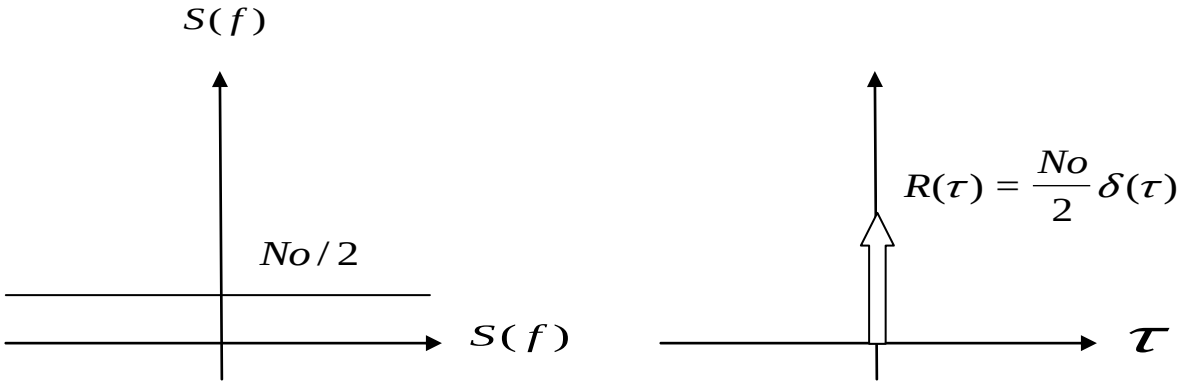


Figura 3- Processo gaussiano branco. Fonte: Autor

A função densidade de probabilidade de uma variável aleatória gaussiana x, de média μ e variância σ, é descrita através da eq.(2) abaixo. Sendo o ruído AWGN caracterizado por possuir $\mu = 0$ e $\sigma = No/2$ como será visto a seguir.

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{(x-\mu)^2}{2\sigma^2} \right) \quad (2)$$

1.3.1.1 O canal gaussiano discreto

Em um sistema de comunicações, o sinal no receptor é filtrado e posteriormente amostrado. Assim, para modelarmos o ruído na saída do amostrador e após a filtragem, faremos a caracterização desse ruído.

O processo de filtragem irá limitar idealmente a banda do ruído gaussiano à banda do filtro aqui considerada de B Hertz. Dessa forma, na saída do filtro o ruído será como ilustrado na figura 4.

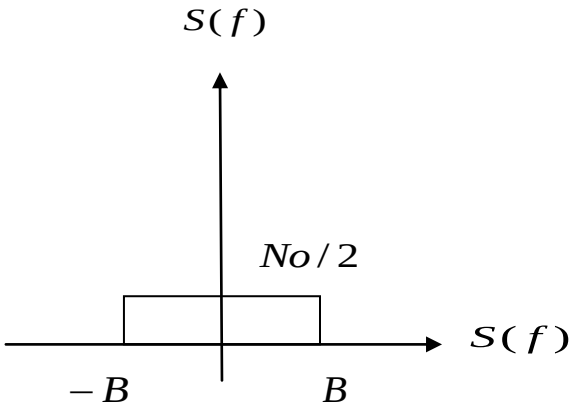


Figura 4 – Ruído gaussiano de faixa limitada. Fonte: Autor

Na saída do filtro, a função densidade espectral é dada por:

$$S_F(f) = \frac{No}{2} G_{2B}(f) \qquad (3).$$

$G_{2B}(f)$, representa a função *gate* definida conforme a seguir:

$$G_{2B}(f) = \begin{cases} 1; & B \leq f \leq B \\ 0; & \text{caso contrário} \end{cases}$$

A função de auto-correlação é dada por:

$$R_F(\tau) = \Im^{-1}\{S_F(f))\} = NoB\,\mathrm{Sa}(2\pi B\tau)$$

$Sa(x)$ representa a função *sampling*, definida como $Sa(x) = sen(x)/x$. Neste caso, as amostras tenderão a ser não correlacionadas somente quando $\tau \to \infty$, ou para os valores de τ para os quais a função *sampling* se anula, ou seja, para:

$$\mathrm{Sa}(2\pi B\tau) = 0 \Rightarrow \tau = \frac{\mathrm{k}}{2\mathrm{B}}; k \in \mathrm{Z}$$

Se o sinal for amostrado nos instantes $t_k = \frac{k}{2B}$, ou seja, na taxa de Nyquist, as amostras de ruído também serão gaussianas de média nula e variância NoB.
Então, podemos representar o canal através de sua forma discreta na saída do amostrador na recepção, conforme a figura 5.

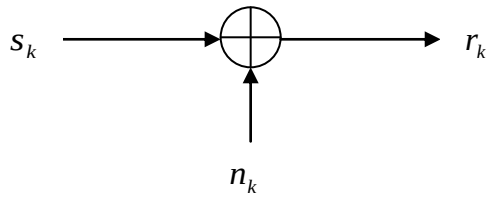


Figura 5- O Canal gaussiano discreto. Fonte: Autor

Uma última consideração a ser feita no cálculo da contribuição do ruído para o sistema, é que caso o modelo trate da representação complexa equivalente em banda base do sistema, o valor para a densidade espectral a ser considerado para o ruído gaussiano branco é No ao invés de $No/2$.

1.3.2-Sistemas de modulação QAM

Seja $s(t)$ o sinal enviado no intervalo de modulação $k\mathrm{T} \leq t \leq (k+1)T$, sendo $k \in Z^*$ e T representando o intervalo de duração de um símbolo de modulação.
O transmissor envia a informação usando M formas de onda $\{s(t) = s_m(t); m = 1,2,...,M\}$ de um alfabeto de modulação QAM. Então $s(t)$ pode ser escrito como,[29]:

$$s(t) = \sqrt{\frac{2E_0}{T}} a_i \cos(2\pi f_c t) + \sqrt{\frac{2E_0}{T}} b_i sen(2\pi f_c t); \quad k\mathrm{T} \leq t \leq (k+1)T \quad (4)$$

Sendo que E_0 é a energia do sinal de menor amplitude na constelação, e a_i e b_i são um par de valores inteiros, que serão mapeados de acordo com o ponto na

constelação a ser enviado, e de acordo com a t-upla de bits de informação correspondente para o envio, e f_c é a frequência da portadora modulada.

Se L é a cardinalidade do alfabeto de modulação, então a quantidade de bits que cada símbolo carrega é dada por:

$$m = \log_2 L \quad (5)$$

1.3.2.1 Interpretação geométrica dos sinais

Por decomposição, o sinal pode ser tratado de forma vetorial, " amostrada" pelo uso de suas componentes sobre uma base ortonormal. Neste processo, se a base é completa, o sinal pode ser completamente caracterizado por sua representação vetorial.

Seja $S_k = [s_{k1}, s_{k2}, ..., s_{kN}]$ o vetor formado pelas projeções do sinal $s(t)$ sobre as funções da base $f_i(t);\ i = 1,.., N$. Então $s(t)$ pode ser reconstruído pela combinação linear das funções da base pré-multiplicadas pelas correspondentes componentes do vetor S_k, ou seja,

$$s(t) = s_{k1} f_1(t) + s_{k2} f_2(t) + ... + s_{kN} f_N(t) \quad (6)$$

sendo:

$$s_{ki} = \int_0^T s(t) f_i(t) dt \quad (7)$$

A figura 6 ilustra a representação geométrica de uma constelação 16 QAM, em um espaço euclidiano formado pelas funções $f_1(t)$ e $f_2(t)$.

sendo:

$$f_1(t) = \sqrt{\frac{2}{T}}\cos(2\pi f_c t)$$

$$f_2(t) = \sqrt{\frac{2}{T}} sen(2\pi f_c t)$$

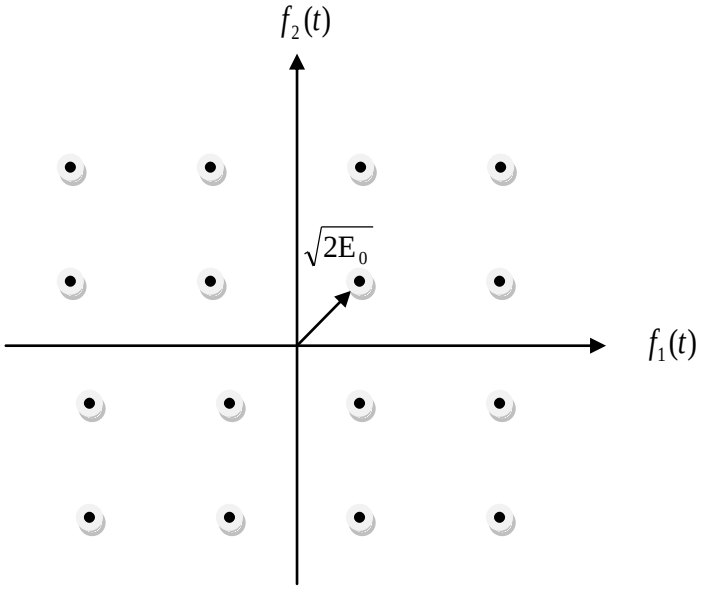


Figura 6 - Representação de uma constelação 16-QAM no espaço euclidiano. Fonte: Autor

1.3.2.2 – O receptor de correlação

O receptor de correlação irá decompor o conjunto de sinais $\{s_m(t)\}$ usando suas componentes sobre uma base de funções $\{f_n(t); n = 1,2,..,N\}$ que é ortonormal e completa para este conjunto [28], [29].

Na prática esse processo equivale a pré-multiplicar o sinal pelas funções da base, integrar o produto no intervalo de existência do sinal e amostrar o valor da integral

ao final do intervalo. O valor das amostras em cada um dos correlacionadores corresponde à projeção do sinal na respectiva base.

Se representarmos o sinal $s(t)$ no intervalo $k\Gamma \leq t \leq (k+1)T$ por eq.(4), então se pode mostrar que o receptor equivalente pode ser implementado conforme a figura 7.

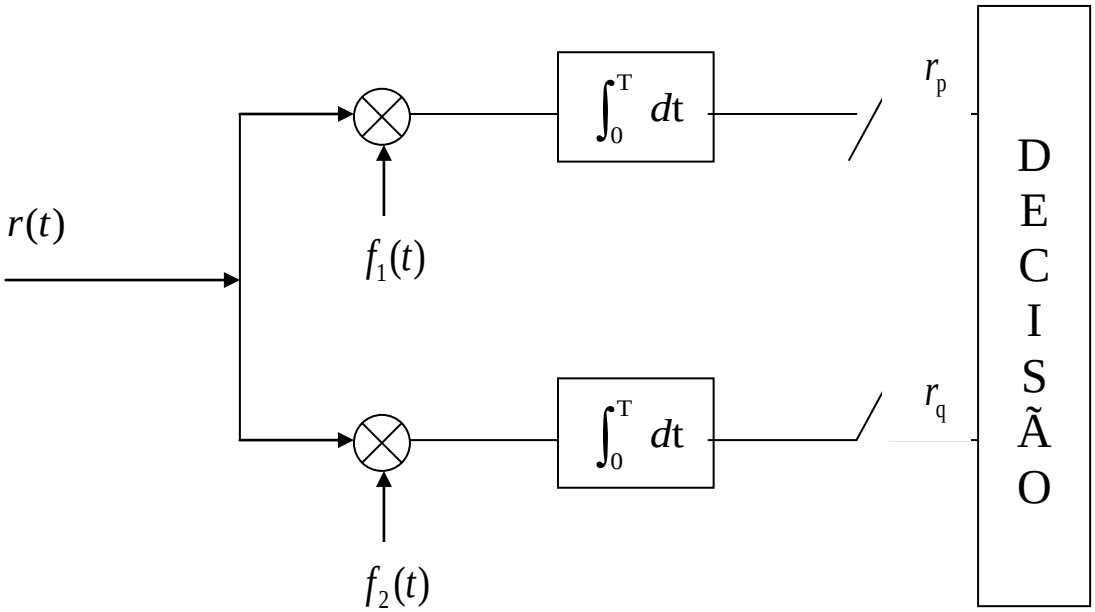


Figura 7 – O receptor de Correlação. Fonte: Autor

Considerando perfeita recuperação de portadora e de relógio na recepção, o valor na saída do amostrador é dado por:

$$r_p = \sqrt{E_0}a_i + \frac{No}{2} \quad (8)$$

$$r_q = \sqrt{E_0}b_i + \frac{No}{2} \quad (9)$$

Supondo que as coordenadas a_i e b_i para uma constelação com cardinalidade M, sejam fornecidas pela matriz:

$$\{a_i, b_i\} = \begin{bmatrix} \left(-\sqrt{M}+1, \sqrt{M}-1\right) & \left(-\sqrt{M}+3, \sqrt{M}-1\right) & . \; . \; . & \left(-\sqrt{M}-1, \sqrt{M}-1\right) \\ \left(-\sqrt{M}+1, \sqrt{M}-3\right) & \left(-\sqrt{M}+3, \sqrt{M}-3\right) & . \; . \; . & \left(-\sqrt{M}-1, \sqrt{M}-3\right) \\ . & . & & . \\ . & . & & . \\ . & . & & . \\ \left(-\sqrt{M}+1, -\sqrt{M}+1\right) & \left(-\sqrt{M}+3, -\sqrt{M}+1\right) & . \; . \; . & \left(-\sqrt{M}-1, -\sqrt{M}+1\right) \end{bmatrix}$$

Então, a probabilidade de erro para uma constelação QAM pode ser aproximada pela expressão:

$$Pe \cong 2\left(1 - \frac{1}{\sqrt{M}}\right) erfc\left(\sqrt{\frac{E_0}{N_0}}\right) \quad (10)$$

Para constelações QAM simétricas em torno da origem, e considerando os símbolos equiprováveis, a probabilidade de erro expressa em termos da energia média da constelação E_{av}, ao invés de E_0, pode escrita conforme eq.(11).

Sendo:

$$E_{av} = \frac{2(M-1)E_0}{3}$$

$$Pe \cong 2(1 - \frac{1}{\sqrt{M}}) erfc\left(\sqrt{\frac{3E_{av}}{2(M \quad 1)N_0}}\right) \quad (11)$$

2 A CODIFICAÇÃO TURBO

2.1 INTRODUÇÃO

Neste capítulo serão apresentados os conceitos envolvidos na codificação turbo. Os tópicos que serão apresentados estão focados nos BTCs e nos algoritmos de Chase e Pyndiah, que são a base teórica para a formulação das propostas que serão apresentadas.

2.2 CÓDIGOS PRODUTO

Seja $\boldsymbol{C}_1$ um código linear sistemático de parâmetros $(\boldsymbol{n}_1, \boldsymbol{k}_1)$ e $\boldsymbol{C}_2$ também um código linear sistemático de parâmetros $(\boldsymbol{n}_2, \boldsymbol{k}_2)$, então se pode formar um código linear de parâmetros $(\boldsymbol{n}_1\boldsymbol{n}_2, \boldsymbol{k}_1\boldsymbol{k}_2)$, usando uma matriz retangular com $\boldsymbol{n}_1$ linhas e $\boldsymbol{n}_2$ colunas. Cada linha é uma palavra código do código $\boldsymbol{C}_1$ e cada coluna é uma palavra código do código $\boldsymbol{C}_2$, conforme a figura 8.

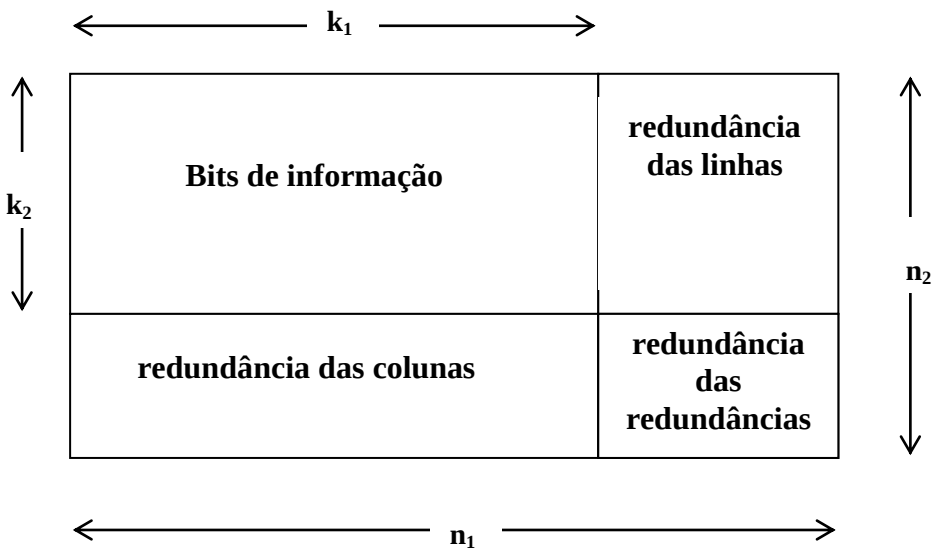


Figura 8 – Arranjo matricial do código produto $\boldsymbol{C}_1 \boldsymbol{x} \boldsymbol{C}_2$ .Fonte: Autor

O arranjo formado da forma descrita acima é chamado código produto, e apresentará propriedades derivadas das propriedades dos códigos formadores $\boldsymbol{C}_1$ e $\boldsymbol{C}_2$.

Se a distância mínima do código $\boldsymbol{C}_1$ for $\boldsymbol{d}_1$ e o peso mínimo do código $\boldsymbol{C}_2$ for $\boldsymbol{d}_2$, então o peso mínimo do código produto é exatamente $\boldsymbol{d}_1\boldsymbol{d}_2$.

A palavra código de menor peso do código produto formado é formada tomando-se a palavra de menor peso do código $\boldsymbol{C}_1$ como linha na matriz, e mantendo nulas todas as colunas para as quais o valor do bit correspondente é zero, e preenchendo com a palavra de menor peso do código $\boldsymbol{C}_2$ as colunas para as quais a posição a palavra do código $\boldsymbol{C}_1$ for igual a um.

Um código produto é capaz de corrigir qualquer vetor de erro de peso até o maior inteiro menor que $\left\lfloor \frac{d_1 d_2 - 1}{2} \right\rfloor$, embora os métodos utilizados e propostos normalmente para a correção de códigos produto dificilmente consigam corrigir essa quantidade de erros [30].

2.2.1 Concatenação utilizando códigos produto

Códigos produto são uma forma de concatenação em série entre dois códigos [31]. Esse esquema de codificação foi proposto por Elias em 1954 [32], e embora não se tenha um *interleaving* fisicamente implementado, o efeito provocado por este arranjo, é o de um *interleaving*. Observa-se também que a regra de permutação, aplicando-se os códigos à matriz de informação e assumindo que a informação é escrita primeiramente no sentido das linhas da matriz contendo os bits de informação e que o segundo código é aplicado no sentido das colunas da matriz, é naturalmente formada pela sequência (0, k, 2K,...(K-1)K,1,K+1,...,(K-1)K+1,....,K-1,2K-1,...KK). A figura 9, representa a matriz formada pelos bits de informação, e codificada pelos códigos produto $\boldsymbol{C}_1$ e $\boldsymbol{C}_2$, no sentido horizontal, e vertical da matriz.

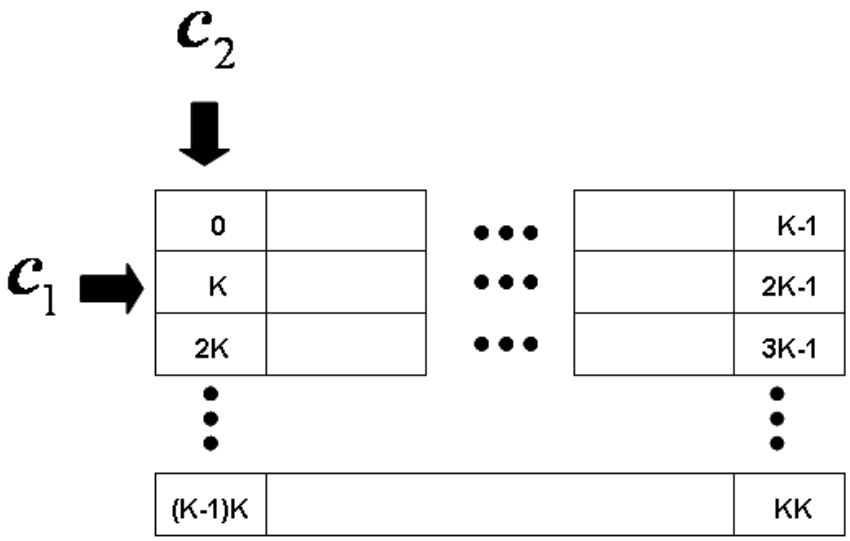


Figura 9 – Efeito de embaralhamento na matriz produto. Fonte: Autor

2.3 ALGORITMOS DE DECODIFICAÇÃO TURBO

Os algoritmos de decodificação turbo fazem uso de várias etapas de iteração, de forma a atingir o elevado desempenho observado nos sistemas que utilizam esses esquemas de decodificação.

A fim de se apresentar uma justificativa para a eficácia desses algoritmos, serão assumidas algumas premissas e introduzidos alguns conceitos apresentados em [33] e [34].

Seja um dado canal de transmissão caracterizado por uma função densidade de probabilidade condicional conforme a figura 10.

Assumindo que esse canal é sem memória, ou seja, a probabilidade associada à observação de um símbolo na recepção do canal só possui correlação com o respectivo símbolo enviado na transmissão, sendo totalmente descorrelacionada dos demais símbolos para uma dada sequência enviada.

Dessa forma, seja a transmissão de uma dada sequência $\boldsymbol{C}$, cuja observação na saída do canal seja a sequencia $\boldsymbol{Y}$, e seja $\boldsymbol{P(Y/C)}$, a probabilidade condicional

associada à função densidade de probabilidade condicional que caracteriza o canal de transmissão.

Figura 10- O canal de transmissão. Fonte: Autor

.

Então, dado que o canal é sem memória, a probabilidade de ocorrência de uma sequência $\boldsymbol{Y}$ observada na saída, condicionada a uma dada sequência transmissão $\boldsymbol{C}$, pode ser escrita como o produto das probabilidades condicionais associadas individualmente a cada símbolo que compõem as sequências, conforme abaixo :

$$\boldsymbol{P}(\boldsymbol{Y}/\boldsymbol{C}) = \prod_{\boldsymbol{k}} \boldsymbol{P}(\boldsymbol{y_k}/\boldsymbol{c_k}) \quad (12)$$

2.3.1- O demodulador *soft*

Assumindo que a função densidade de probabilidade condicional do canal é conhecida, o demodulador *soft* é o estagio da decodificação que fornecerá, por uma função de verossimilhança $L_k(\ )$, associada à função densidade de probabilidade condicional do canal, os valores associadas às probabilidades condicionais dos símbolos recebidos do canal, conforme representado na figura 11.

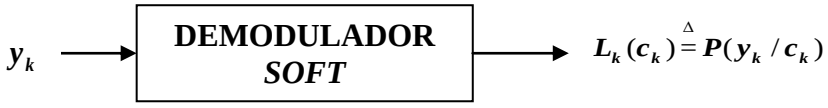


Figura 11 – Demodulador *Soft*

então, para uma dada sequência $\boldsymbol{Y}$ recebida:

$$L(C) = P(Y/C) = \prod_k P(y_k/c_k) = \prod_k L_k(c_k) \quad (13)$$

2.3.2- A decisão utilizando a máxima probabilidade a *posteriori* (MAP)

Lembrando que no sistema de codificação proposto de códigos produto, cada símbolo é codificado por dois codificadores, para cada símbolo recebido existem duas sequências (palavras-código) recebidas. Uma referente ao primeiro código (sentido das linhas/colunas) e outra associada ao segundo código (sentido das colunas/linhas).

Dessa forma uma decisão MAP associada a um valor $\boldsymbol{i}$ (dentro de um alfabeto de símbolos $u \in U$ de informação), mapeado pela codificação da matriz produto em uma posição $\boldsymbol{k}$ da matriz, é aquela que maximiza a probabilidade conjunta a *posteriori* dada pela eq.(14).

$$\boldsymbol{APP_k(i) = P(u_k = i, Y_1, Y_2)} \quad (14)$$

A figura 12 ilustra o $\boldsymbol{k}$-ésimo elemento dentro da matriz produto formada na recepção, além das sequencias $\boldsymbol{Y_1}$ e $\boldsymbol{Y_2}$ associadas às códigos $\boldsymbol{C_1}$ e $\boldsymbol{C_2}$ respectivamente.

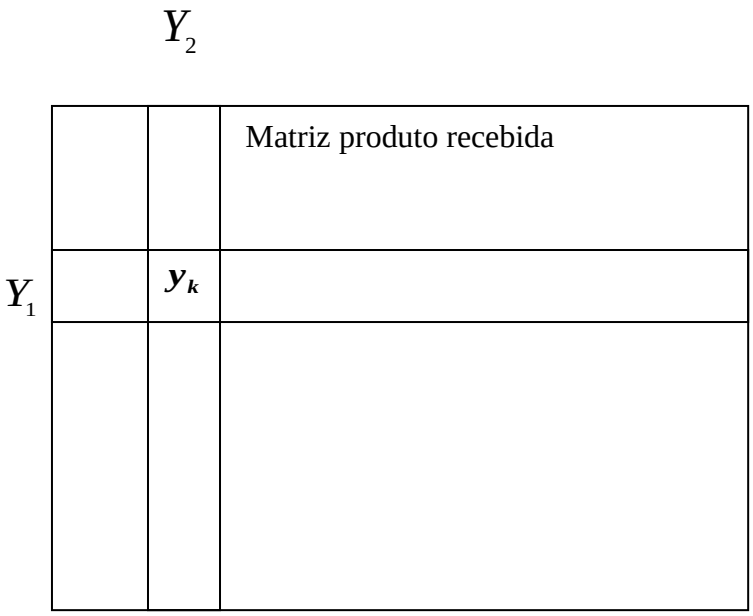


Figura 12- Localização da $\boldsymbol{k}$ -ésima posição na matriz produto. Fonte: Autor

A probabilidade a *posteriori* $\boldsymbol{APP_k(i)}$, para a $\boldsymbol{k}$-ésima posição da matriz formada e associada ao valor $\boldsymbol{i}$ dentro do alfabeto dos símbolos de informação U, pode ser escrita como a somatória das probabilidades condicionais de todas as palavras pertencentes aos códigos $\boldsymbol{C_1}$ e $\boldsymbol{C_2}$, associadas a sequencias de informação $\boldsymbol{U}$ para as quais $\boldsymbol{u = i}$, associada à posição $\boldsymbol{k}$ da matriz produto, ou seja:

$$APP_k(i) = P(u_k = i, Y_1, Y_2) = \sum_{U:u_k=i} P(Y_1 / C_1(U)) x P(Y_2 / C_2(U)) x P(U) \quad (15)$$

Como o canal considerado é sem memória, e aplicando a função de máxima verossimilhança (*Maximum-Likelihood*) $\boldsymbol{L}$ associada às sequências recebidas, temos:

$$P(Y_1 / U) = P(Y_1 / C_1(U)) = L_1(C_1(U)) = \prod_l L_{1l}(c_{1l}(U)) \quad (16)$$

$$P(Y_2 / U) = P(Y_2 / C_2(U)) = L_2(C2(U)) = \prod_l L_{2l}(c_{2l}(U)) \quad (17)$$

Pela análise das equações acima se observa que os valores observados na saída do canal apresentam dependência apenas com o correspondente símbolo enviado, devido ao canal ser sem memória. No entanto, cada símbolo codificado para a transmissão no canal, possuirá correlação com toda a sequencia de informação utilizada para a geração da palavra código correspondente.

Dessa forma, a fim de se calcular a máxima probabilidade a *posteriori* para um dado valor $\boldsymbol{u = i}$, na posição $\boldsymbol{k}$ da matriz produto, é necessário realizar o somatório das probabilidades sobre todas as sequências $\boldsymbol{U}$ para as quais se tem o valor de $\boldsymbol{u_k = i}$ associado às palavras dos códigos $\boldsymbol{C_1}$ e $\boldsymbol{C_2}$ enviadas, conforme (15).

Observa-se que tal operação terá um grau de complexidade que aumenta exponencialmente com o comprimento do código, tornando a decodificação MAP impraticável para a maioria dos casos de códigos utilizados. No entanto, os algoritmos recursivos propostos por Berrou in 1993 para decodificação de códigos convolucionais, e mais tarde os algoritmos recursivos propostos por Pyndiah para a decodificação de códigos produto, funcionam bem na pratica, embora não exista

prova de que estes algoritmos convergem para uma decodificação de máxima probabilidade a *posteriori*.

Benedetto propôs em [34] uma explicação heurística, baseada na suposição de que em determinadas condições, se pode assumir que os símbolos codificados possuem dependência apenas do símbolo de informação correspondente, e não com toda a sequência de informação associada. Tal premissa se justificaria pelo *interleaving* que teria um efeito de diminuir a dependência entre os símbolos enviados e a sequência de informação correspondente.

Embora a análise tenha sido realizada para o caso de códigos convolucionais utilizando *interleaving*, a construção de códigos produto apresenta um efeito equivalente ao de um *interleaving*. Ainda para o caso de códigos produto, utilizando códigos sistemáticos e com alta taxa de codificação, esta suposição passa a ser válida para todos os símbolos correspondentes aos símbolos de informação da sequencia codificada enviada, que representam quase que a totalidade dos símbolos enviados.

Assumindo essa premissa de independência proposta na justificativa heurística, podem-se reescrever as probabilidades a *posteriori* representadas nas equações (16) e (17), por uma nova função de máxima verossimilhança $\tilde{\boldsymbol{L}}$, conforme abaixo:

$$P(Y_1/U) = L_1(C_1(U) = \prod_j L_{1j}(c_{1j}(U)) \approx \prod_l \tilde{L}_{1l}(u_l) \quad (18)$$

$$P(Y_2/U) = L_2(C_2(U) = \prod_j L_{2j}(c_{2j}(U)) \approx \prod_l \tilde{L}_{2l}(u_l) \quad (19)$$

Reescrevendo (15) e utilizando (18) e (19) a $\boldsymbol{APP_k(i)}$ satisfaz o sistema de equações abaixo:

$$(\text{A})\quad APP_k(i) = \tilde{L}_{1k}(i) x Pa(i) x \sum_{U:u_k=i} L_2(C_2(U) \prod_{l \neq k} \tilde{L}_{1l}(u_l) Pa(u_l) \quad (20)$$

$$(\text{B})\quad APP_k(i) = \tilde{L}_{2k}(i) x Pa(i) x \sum_{U:u_k=i} L_1(C_1(U) \prod_{l \neq k} \tilde{L}_{2l}(u_l) Pa(u_l) \quad (21)$$

e, além disso:

$$APP_k(i) = \tilde{L}_{1k}(i) x \tilde{L}_{2k} x Pa(i) \quad (22)$$

pela análise de (20), (21) e (22) tem-se que:

$$\tilde{L}_{1k}(i) = \sum_{U:u_k=i} L_1(C_1(U) \prod_{l \neq k} \tilde{L}_{2l}(u_l) Pa(u_l) \quad (23)$$

$$\tilde{L}_{2k}(i) = \sum_{U:u_k=i} L_2(C_2(U) \prod_{l \neq k} \tilde{L}_{1l}(u_l) Pa(u_l) \quad (24)$$

O segundo termo nas equações acima é denominado informação extrínseca, pois carrega a informação relativa ao símbolo que se está estimando, a partir de todos os demais símbolos da sequencia recebida, menos o próprio símbolo enviado. Esta nova quantidade é a base da formulação para os algoritmos de decodificação turbo.

2.3.3 A decodificação turbo

Para se calcular a quantidade $\boldsymbol{APP_k(i)}$, conforme a eq.(22), é necessário calcular as quantidades $\tilde{\boldsymbol{L}}_{1k}(\boldsymbol{i})$ e $\tilde{\boldsymbol{L}}_{2k}(\boldsymbol{i})$, o que é conseguido pela resolução do sistemas de equações não lineares dado pelas eq.(21) e eq.(20).

Uma maneira de se tentar resolver esse sistema é fazendo uma escolha inicial para os valores referentes à informação extrínseca , e então por um processo de iterações, tentar convergir para os valores corretos de $\tilde{\boldsymbol{L}}_{1k}(\boldsymbol{i})$ e $\tilde{\boldsymbol{L}}_{2k}(\boldsymbol{i})$, a fim de se calcular $\boldsymbol{APP_k(i)}$. De fato, os algoritmos de decodificação turbo podem ser vistos como uma forma iterativa de solução do sistema de equações descrito acima.

Assim, para o cálculo, no m-ésimo passo de iteração da quantidade $\tilde{L}_{1k}^{(m)}(i)$, a informação extrínseca $\tilde{L}_{2k}^{(m-1)}(i)$ calculada no passo anterior é utilizada como informação a priori, e assim sucessivamente, ou seja:

$$\tilde{L}_{1k}^{(m)}(i) = \sum_{U:u_k=i} L_1(C_1(U)\prod_{l\neq k} \tilde{L}_{2l}^{(m-1)}(u_l)Pa(u_l), \forall k \quad (25)$$

$$\tilde{L}_{2k}^{(m)}(i) = \sum_{U:u_k=i} L_2(C_2(U)\prod_{l\neq k} \tilde{L}_{1l}^{(m-1)}(u_l)Pa(u_l), \forall k \quad (26)$$

Existem diferentes algoritmos e abordagens utilizadas para a realização do processo de iteração turbo, que buscam, usando algoritmos sub-óptimos, não só a convergência para uma decodificação MAP ou ML, como também a simplificação dos esquemas de decodificação de forma a possibilitar a sua implementação prática para sistemas reais.

O algoritmo de decodificação que é proposto neste trabalho se baseia no algoritmo proposto por Pyndiah em [3], que consiste em um processo de decodificação iterativa, que utiliza, como parte do processo de decodificação, o algoritmo *soft* proposto por Chase em [4], de forma a limitar a varredura durante a decodificação a um subgrupo de palavras código mais prováveis, ao invés de realizar uma varredura em todas as palavras código associadas a um determinado valor observado na saída do demodulador.

Dessa forma, o conjunto $\{U / u_k = i\}$, é limitado a um subconjunto $U' \subset U$ formado a partir do algoritmo de Chase, e que representa o subconjunto das sequências de maior probabilidade tal que $u_k = i$, limitado às condições de contorno impostas pelo algoritmo. Os critérios utilizados para a formação do subconjunto U', podem ser escolhidos de forma a ter-se um maior ou menor conjunto de sequencias candidatas. O aumento desse subconjunto aumenta a probabilidade do conjunto conter a sequência correta mas aumenta a complexidade do algoritmo de decodificação.

Os critérios para formação desse subconjunto tem sido objeto de estudos, e diferentes algoritmos já foram utilizados com esse propósito, tais como o algoritmo de Chase-II apresentado em [4] e usado em [3], o PML usado em [35], ou o algoritmo de Fang-Battail-Buda (FBBA) usado em [36].

A proposta é utilizar um algoritmo subótimo que limitará a função de máxima verossimilhança às sequencias com maior probabilidade, e não a todas as sequencias que satisfaçam $\boldsymbol{u_k} = \boldsymbol{i}$,ou seja:

$$\sum_{U:u_k=i} L(C(U)) \approx \sum_{U':u_k=i} L(C(U)), \forall k \quad (27)$$

A seguir será apresentando a descrição do algoritmo de Chase, a fim fornecer alguns conceitos necessários para dar continuidade na descrição do algoritmo de decodificação turbo.

2.4 O ALGORITMO DE CHASE

Em 1972 David Chase propôs em [4], uma classe de algoritmos de decodificação, que utilizavam a medida da informação de canal em adição à pura utilização das propriedades algébricas dos códigos. Foi mostrado nesse trabalho que com uma elevada probabilidade, pela utilização da medida da informação de canal, a capacidade de correção de erros pode ser estendida até d -1, sendo d a distância mínima do código. Essa extensão na capacidade de correção do código em relação ao esquema convencional de decodificação binária é devida à utilização da informação de canal (*soft information*), que fornece uma medida da confiabilidade de cada valor binário recebido.

A seguir, a justificativa bem como a classe de algoritmos propostos por Chase serão descritos.

O diagrama em blocos que descreve um sistema genérico para o qual o algoritmo é proposto está ilustrado na figura 13.

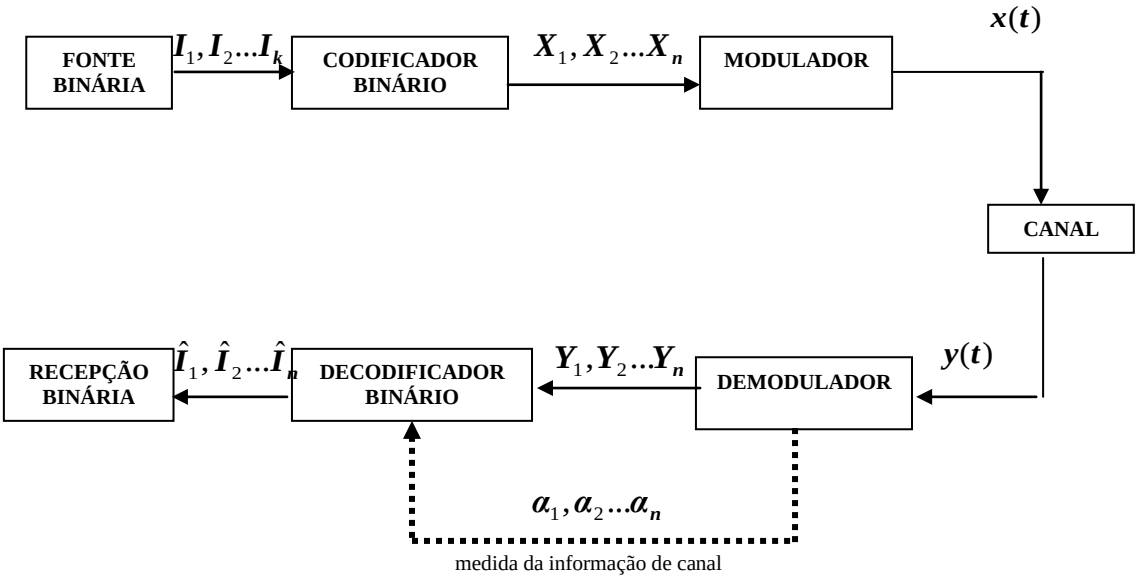


Figura 13 - Diagrama em blocos do sistema de comunicação.

Para cada sequência de k bits de informação, é assumido que o codificador binário irá gerar um bloco de n bits codificados representados por $X_1, X_2 ... X_n$. Essa sequência é fornecida à entrada de um modulador, que fará o mapeamento para transmissão nas formas de onda $x(t)$. Se um canal binário for considerado, o demodulador irá gerar uma sequencia de n bits, $Y_1, Y_2 ... Y_n$, baseada na forma de onda recebida $y(t)$. No modelo proposto, além da sequência Y fornecida na saída do demodulador, também a sequência $\alpha_1, \alpha_2 ... \alpha_n$, é fornecida.

Essa sequência de valores positivos, correspondente à medida da informação de canal é utilizada como uma medida da confiabilidade dos dígitos binários da sequência Y recebida. Dessa forma, se $\alpha_i > \alpha_j$ é assumido que o valor de Y_i tem maior probabilidade de estar correto que o valor de Y_j. Assim cada valor α_i pode ser visto como uma medida da confiabilidade de cada bit recebido.

Em muitas aplicações práticas, a medida de informação de canal pode ser obtida de forma praticamente direta. Por exemplo, se considerarmos que a magnitude do símbolo recebido está relacionada de forma monotônica à estatística do canal e consequentemente à probabilidade do símbolo recebido estar correto, a medida de informação de canal relativa a um dado símbolo de canal recebido pode ser obtida através da utilização de um conversor analógico digital de j bits, no qual o bit mais

significativo corresponderá ao valor binário associado a cada bit Y_i da sequência binária Y, e os demais j-1 bits determinarão cada um dos α_i associados.

Essa informação é comumente referida como *soft-decision information*, uma vez que o valor associado está representado em mais de um bit, em contraste a *hard-decision information*, no qual a informação é obtida pela decisão sobre um único bit.

2.4.1-Classe de algoritmos utilizando decisão *soft*

A seguir será descrita uma classe de algoritmos que utilizam a informação fornecida pela sequência α. Os algoritmos propostos podem utilizar qualquer decodificador binário, capaz de corrigir até $\lfloor (d\text{-}1)/2 \rfloor$ erros.

O decodificador binário irá determinar a palavra $X_m = X_{m1}, X_{m2}, ..., X_{mN}$, que irá diferir no menor número de posições da sequencia recebida $Y = Y_1, Y_2 ... Y_N$, sempre que a diferença não for maior que $\lfloor (d\text{-}1)/2 \rfloor$. A sequência que conterá $1's$ nas posições onde Y difere de X_m, ou seja a sequência de erros, é dada por:

$$Z_m = Y \oplus X_m = Y_1 \oplus X_{m1}, Y_2 \oplus X_{m2}, \ldots, Y_N \oplus X_{mN}$$

A notação $\oplus$ representa a operação de soma modulo-2. Definido o peso binário da sequência Z_m como:

$$W(Z_m) = \sum_{i=1}^{N} Z_{mi} \quad (28)$$

A operação realizada pelo decodificador binário é a determinação da palavra código que satisfaz:

$$W(Z_m) \leq \left\lfloor \frac{(d\text{-}1)}{2} \right\rfloor \quad (29)$$

O decodificador binário irá encontrar uma única palavra código que satisfaça eq.(29), ou então nenhuma palavra código é escolhida.
Um decodificador binário completo pode então ser definido como aquele capaz de encontrar a palavra código que satisfaça:

$$\min_m W(Y \oplus X_m) \quad (30)$$

De forma similar, se pode definir o decodificar completo, com medida de informação de canal, como aquele capaz de determinar a palavra código que satisfaça:

$$\min_m W_\alpha(Y \oplus X_m) \quad (31)$$

Nesse caso, se está buscando pelo padrão de erro $Z_m = Y \bigoplus X_m$ de menor peso analógico, com o peso analógico é definido como:

$$W_\alpha(Z_m) = \sum_{i=1}^{N} \alpha_i Z_{mi} \quad (32)$$

Da mesma forma que, com decodificador binário se busca atingir um desempenho próximo do decodificador binário completo, com o decodificar utilizando medida de informação de canal, se busca atingir um desempenho próximo ao decodificador que satisfaz a eq.(31).

O conceito básico contido no algoritmo de decodificação utilizando a medida de informação de canal pode ser ilustrado pela figura 14.

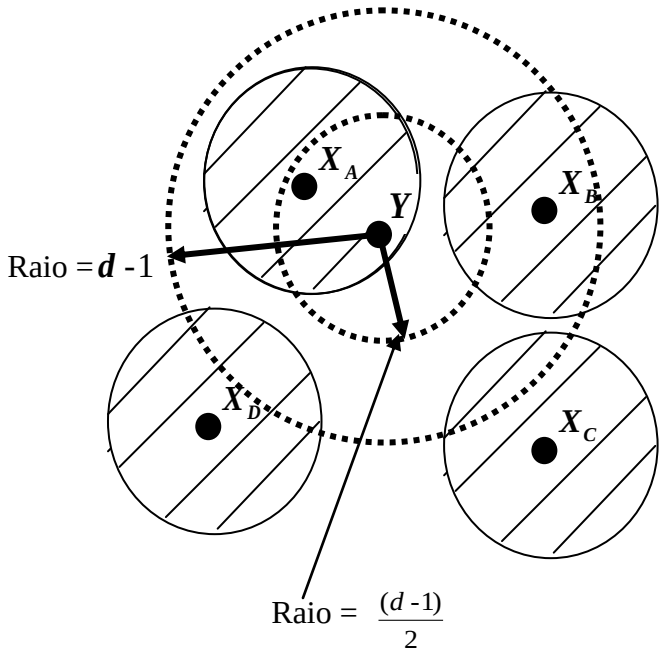


Figura 14- Representação geométrica para o decodificador com medida de informação de canal

Essa representação geométrica representa a distância binária de quatro palavras código X_A, X_B, X_c, X_D e a sequência recebida Y. Cada palavra código está circundada por uma esfera de raio $\lfloor (d-1)/2 \rfloor$. Dessa forma, uma única palavra código, ou equivalentemente um único vetor de erro, é obtido pelo decodificador binário se a sequência recebida estiver dentro de alguma dessas esferas. No caso da figura, existe um único padrão de erro $Z = Y \oplus X_A$, dentro da esfera de raio $\lfloor (d-1)/2 \rfloor$, que envolve a sequência recebida Y. O objetivo do decodificador utilizando medida de informação de canal é utilizar o decodificador binário para auxiliar na formação de um conjunto relativamente pequeno de possíveis padrões de erro associados à sequência recebida Y, ao invés de um único padrão de erro. E determinar dentro desse conjunto, aquele padrão que possuí o menor peso analógico conforme definido em eq.(31).

O conjunto de padrões de erro é obtido perturbando a sequência recebida Y, com o vetores de teste T, que consiste no conjunto formado pelas sequências binárias que possuem 1's nas posições onde os bits da sequência Y serão invertidos. Pela

adição módulo-2 dos vetores de teste com a sequência recebida, uma nova sequência

$$Y' = Y \oplus T \quad (33)$$

é obtida, e pela decodificação binária um novo padrão de erro $\boldsymbol{Z}'$ é obtido. O padrão de erro associado à sequência $\boldsymbol{Y}$, é dado por:

$$Z_T = T \oplus Z', \quad (34)$$

que poderá ou não coincidir com o padrão de erro original dependendo se $\boldsymbol{Y}'$ cair ou não dentro da esfera associada a uma nova palavra código.

Para a classe de algoritmos propostos, a sequência recebida é sempre perturbada dentro da esfera de raio $\boldsymbol{d}$ -1 que envolve a sequência $\boldsymbol{Y}$.

Em [4], três algoritmos são propostos, nos quais os critérios para formação do conjunto de possíveis padrões são estabelecidos de forma a diminuir progressivamente esse conjunto, para estabelecer uma relação de compromisso entre desempenho e complexidade. Em todos os casos, como o decodificador binário é utilizado, e dado o fato de que este é capaz de encontrar uma palavra código apenas quando a eq.(29) é satisfeita, é possível que nenhuma palavra seja encontrada em algumas situações. Nesse caso, a estimativa da palavra código é dada pela sequência recebida, apesar do fato que certamente essa sequência conterá erros.

A seguir serão apresentados os três algoritmos propostos por Chase e os conceitos associados a cada um deles.

2.4.1.1 Algoritmo I

Neste primeiro algoritmo, um grupo com um elevado número de padrões de erro é formado. Nesse algoritmo, todos os padrões de erro dentro da esfera de raio $\boldsymbol{d}$ -1 ao redor da sequência $\boldsymbol{Y}$ ilustrada na figura 3.16 são gerados. Ou seja, todos os padrões de erro de peso binário menor ou igual a $\boldsymbol{d}$ -1 são considerados. Uma vez

que o padrão de erro selecionado é determinado pelo seu peso analógico e não o seu peso binário, é possível que seja selecionado um padrão de erro com mais de $\lfloor (d-1)/2 \rfloor$ 1's ,e dessa forma, estender a capacidade de correção do código. Essa é a ideia básica da decodificação *soft* proposta pelo algoritmo de Chase.

Um conjunto de vetores de testes que é suficiente, (porém não necessário) para gerar todos os padrões de erro de peso menor que $\boldsymbol{d}-1$ é dado pelo conjunto dos vetores de teste $\boldsymbol{T}$, que contenham todas as combinações de $\boldsymbol{d}/2$, "1's". Uma vez que o decodificador binário é capaz de obter padrões de erro até $\lfloor (d-1)/2 \rfloor$, então quando combinados com o vetor de teste apropriado, podemos gerar qualquer padrão de erro de peso binário até $\boldsymbol{d}-1$.

Uma vez que existem $\binom{n}{\lfloor d/2 \rfloor}$ diferentes padrões, este algoritmo só se aplica a códigos cuja distância mínima seja pequena. Na verdade, uma primeira redução no número de vetores de teste já pode ser conseguida eliminando-se aqueles vetores que conduzem ao mesmo padrão de erro. Em termos práticos é possível reduzir significativamente o número de vetores de teste, sem comprometer o desempenho do decodificador, o que será mostrado pelos algoritmos II e III.

2.4.1.2- Algoritmo II

Neste algoritmo, um número consideravelmente menor de vetores de teste é utilizado. Apenas padrões de erro com não mais do que $\lfloor (d-1)/2 \rfloor$ erros localizados fora do conjunto das $\lfloor d/2 \rfloor$ posições de menor confiança de medida de informação de canal serão testados. Os padrões testados continuarão a conter não mais do que $\boldsymbol{d}-1$ erros. No entanto, nem todos os padrões com $\boldsymbol{d}-1$ ou menos erros serão testados.

De forma a gerar o conjunto dos padrões de erro acima, o conjunto dos padrões de teste $\boldsymbol{T}$ é constituído pelo conjunto dos vetores que possuem todas as combinações de 1's nas $\lfloor d/2 \rfloor$ posições de menor confiabilidade, ou seja as $\lfloor d/2 \rfloor$ posições cuja medida de informação de canal possua a menor confiabilidade. Uma vez que

existem $2^{\lfloor d/2 \rfloor}$ vetores de teste possíveis, incluindo o vetor todo nulo, então serão encontrados no máximo $2^{\lfloor d/2 \rfloor}$ vetores de erro candidatos, e na pratica, bem menos.

2.4.1.3 Algoritmo III

Este algoritmo é basicamente igual ao algoritmo II, com a diferença que neste caso, apenas $\lfloor (d-1)/2+1 \rfloor$ padrões de teste $\boldsymbol{T}$ serão considerados, ao invés de $2^{\lfloor d/2 \rfloor}$. Cada padrão de teste $\boldsymbol{T}$ terá $\boldsymbol{i}$ 1's localizados na $\boldsymbol{i}$ posições de menor confiabilidade.

Em [4] foi mostrado por simulações computacionais que este algoritmo apresentará uma desempenho um pouco inferior aos algoritmos II e III. No entanto, devido a sua simplicidade de implementação em relação aos outros algoritmos, pelo fato de possuir o menor conjunto de padrões de teste, tem um forte apelo prático, principalmente quando da utilização de códigos que apresentem distancia mínima elevada, uma vez que o número de padrões de teste neste caso cresce linearmente com a distância do código.

A figura 15 mostra o resultado comparativo publicado em [4] entre o desempenho desses algoritmos em um sistema codificado utilizando um código de Galois (24,12), em canal gaussiano.

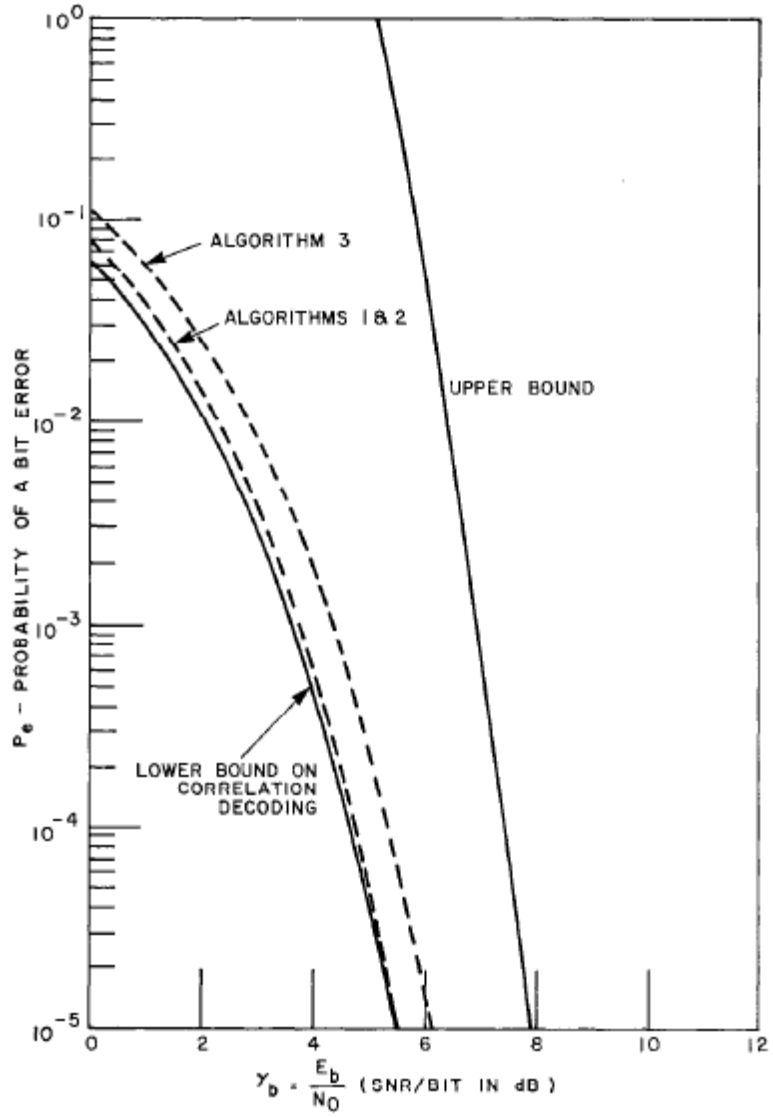


Figura 15 – Decodificação *soft*, desempenho código Galois (24,12) em um canal gaussiano

2.5 O ALGORITMO DE DECODIFICAÇÃO PROPOSTO POR PYNDIAH

O algoritmo proposto por Pyndiah em [3] assume um processo de decodificação binária, a partir do envio de uma sequência de elementos binários $\{0,1\}$, que são mapeados utilizando-se símbolos binários $\{-1,+1\}$, ou seja $0 \rightarrow -1$, e $1 \rightarrow +1$.

Seja uma sequência $\boldsymbol{E}$ transmitida no canal, para a qual foi observada na saída do demodulador a sequência $\boldsymbol{Y}$, e eleita a palavra $\boldsymbol{D}$ como a vencedora utilizando-se o algoritmo de Chase.

Para se calcular a informação extrínseca associada a cada símbolo $\boldsymbol{d_k}$ da palavra $\boldsymbol{D}$ eleita utilizando-se o algoritmo de Chase, dentro do subconjunto de todas as palavras código encontradas pelo algoritmo de Chase na etapa anterior, defini-se a LLR, ou função logarítmica da razão de verossimilhança associada ao $\boldsymbol{k}$-ésimo símbolo $\boldsymbol{e_k}$ transmitido no canal como:

$$\Lambda(d_k) = \ln\left(\frac{\Pr\{e_k = d_k / Y\}}{\Pr\{e_k \neq d_k / Y\}}\right) \quad (35)$$

tem-se que:

$$\Pr\{e_k = d_k / Y\} = \sum_{C^i \in S^{d_k}} \Pr\{E = C^i / Y\} \qquad (36)$$

sendo S^{d_k} o conjunto das palavras código $\{\boldsymbol{C^i}\}$ tal que $\boldsymbol{c_k^i} = \boldsymbol{d_k}$. Da mesma forma, o denominador pode ser escrito como:

$$\Pr\{e_k \neq d_k / Y\} = \sum_{C^i \in S^{\neq d_k}} \Pr\{E = C^i / Y\} \quad (37),$$

sendo $S^{\neq d_k}$ o conjunto das palavras código $\{\boldsymbol{C^i}\}$ tal que $\boldsymbol{c_k^i} \neq \boldsymbol{d_k}$

substituindo a eq.(36) e a eq.(37) na eq.(35), temos:

$$\Lambda(d_k) = \ln\left(\frac{\sum_{C^i \in S^{d_k}} \Pr\{E = C^i / Y\}}{\sum_{C^i \in S^{\neq d_k}} \Pr\{E = C^i / Y\}}\right) \quad (39)$$

Assumindo que as palavras código são uniformemente distribuídas, e que o canal é gaussiano:

$$\Pr\{Y / E = C^i\} = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^n \exp\left(-\frac{\left|\mathrm{Y} - \mathrm{C}^{\mathrm{i}}\right|^2}{2\sigma^2}\right) \quad (40)$$

Seja $C_{\min}^{\neq d_k}$ e $C_{\min}^{=d_k}$, as duas palavras código mais próximas da sequência $\boldsymbol{Y}$ recebida encontradas pelo algoritmo de Chase, tal que $c_k \neq d_k$ e $c_k = d_k$ respectivamente. Então substituindo a eq.(40) na eq.(39), após algumas manipulações, pode-se mostrar que:

$$\Lambda(d_k) = \frac{1}{2\sigma^2}\left(\left|Y - C_{\min}^{\neq d_k}\right|^2 - \left|Y - C_{\min}^{d_k}\right|^2\right) + \ln\left(\frac{\sum_i A_i}{\sum_i B_i}\right) \quad (41),$$

sendo:

$$A_i = \exp\left(\frac{\left|Y - C_{\min}^{d_k}\right|^2 - \left|Y - C^i\right|^2}{2\sigma^2}\right) \leq 1, \text{ onde } C^i \in S^{d_k} \quad (42)$$

e

$$B_i = \exp\left(\frac{\left|Y - C_{\min}^{\neq d_k}\right|^2 - \left|Y - C^i\right|^2}{2\sigma^2}\right) \leq 1 \text{ onde } C^i \in S^{\neq d_k} \quad (43)$$

Para relações sinal-ruído altas, isto é, quando $\sigma \to 0$, $\sum_i A_i \approx \sum_i B_i \to 1$, e o segundo termo em (41) tende a zero.

Desprezando o segundo termo em (41), obtemos a seguinte aproximação para o calculo da LLR de $\boldsymbol{d_k}$:

$$\Lambda'(d_k) = \frac{1}{2\sigma^2}\left(\left|Y - C_{\min}^{\neq d_k}\right|^2 - \left|Y - C_{\min}^{d_k}\right|^2\right) \quad (44).$$

Expandindo (44), e lembrando que algoritmo assume um sistema de transmissão e decodificação binários ($0 \rightarrow -1, \text{e}\, 1 \rightarrow +1$), pode-se escrever:

$$\Lambda'(d_k) = \left(\frac{2}{\sigma^2} \left(y_k + \sum_{l=1, l \neq l}^{n} y_l c_l^{d_k} p_l \right) \right) \quad (45)$$

sendo:

$$p_l = \begin{cases} 0, \text{ se } c_l^{d_k} = c_l^{\neq d_k} \\ 1, \text{se } c_l^{d_k} \neq c_l^{\neq d_k} \end{cases}$$

Supondo que σ é constante e normalizando $\Lambda'(\boldsymbol{d_k})$ com relação a $\frac{2}{\boldsymbol{\sigma}^2}$, se obtém a seguinte equação:

$$y'_k = y_k + w_k \quad (46)$$

sendo

$$w_k = \sum_{l=1, l \neq k}^{n} y_l c_l^{d_k} p_l \quad (47)$$

A LLR normalizada $\boldsymbol{y'_k}$ corresponderá à saída *soft* do decodificador. Ela terá o mesmo sinal que $\boldsymbol{d_k}$ e o seu valor absoluto indica a confiabilidade associada a essa decisão. A equação (46) indica que $\boldsymbol{y'_k}$ é obtido pela entrada soft y_k mais o termo w_k, que é função das duas palavras código dentro do subconjunto U', que se encontram mais próximas da sequencia recebida $\boldsymbol{Y}$. É importante observar que o termo $\boldsymbol{w_k}$ é formado com a informação de todos os símbolos $\{y_l\}$, porém com $\boldsymbol{l \neq k}$. O termo $\boldsymbol{w_k}$ é uma correção aplicada à entrada de dados, corresponde à informação extrínseca associada ao símbolo $\boldsymbol{y_k}$ na entrada do decodificador *soft*. A informação extrínseca é uma variável aleatória com distribuição gaussiana, uma vez que é formada a partir da combinação linear de variáveis aleatórias identicamente distribuídas. Além disto, é uma variável descorrelacionada com $\boldsymbol{y_k}$.

2.5.1- A implementação da decodificação turbo utilizando o algoritmo proposto por Pyndiah

Conforme descrito na seção anterior, dado que a palavra eleita $\boldsymbol{D} = \boldsymbol{C}_{\min}$ e a palavra competidora $C_{\min}^{\neq d_k}$ tenham sido determinadas, então, a partir de (44), pode-se mostrar que $\boldsymbol{y_k'}$ pode ser determinado pela equação:

$$y_k' = \left(\frac{\left|Y - C_{\min}^{\neq d_k}\right|^2 - \left|Y - D\right|^2}{4}\right) d_k \quad (48)$$

No entanto não há garantia de que exista $C_{\min}^{\neq d_k} \subset U'$. Pode-se sempre aumentar a quantidade de palavras candidatas em $\boldsymbol{U}'$, de forma a aumentar a probabilidade de $C_{\min}^{\neq d_k}$ estar contida em $\boldsymbol{U}'$, pelo aumento do número de bits $\boldsymbol{p}$ considerados menos confiáveis na formação dos vetores de teste do algoritmo de Chase. Porém, a complexidade de decodificação do algoritmo irá aumentar exponencialmente com o aumento de $\boldsymbol{p}$.

De forma a estabelecer uma relação de compromisso entre desempenho e complexidade, o algoritmo de Pyndiah propõe a seguinte solução quando a palavra competidora não é encontrada dentro do subconjunto $\boldsymbol{U}'$:

$$y_k' = \beta \text{ x } d_k \text{ , com } \beta \geq 0 \quad (49)$$

Embora essa solução seja bem simples, foi demonstrado que ela é eficiente quando a palavra competidora não for encontrada.

As justificativas apresentadas para esta proposta são conforme abaixo:

- O sinal da saída $\boldsymbol{y_k'}$ é igual a de $\boldsymbol{d_k}$, enquanto apenas o valor absoluto da confiabilidade é função de $C_{\min}^{\neq d_k}$.
- Se a palavra competidora $C_{\min}^{\neq d_k}$, não foi encontrada, então com grande probabilidade $C_{\min}^{\neq d_k}$ está distante da sequência recebida $\boldsymbol{Y}$.

• Se $C_{\min}^{\neq d_k}$ está distante da sequência recebida $\boldsymbol{Y}$, então a probabilidade que $\boldsymbol{d_k}$ esteja correto é relativamente alta e, portanto, a confiabilidade de $\boldsymbol{d_k}$ é também alta.

O valor de $\boldsymbol{\beta}$ proposto inicialmente em [3] foi otimizado por tentativa e erro. Em [37], foi fornecida uma equação para o calculo de $\boldsymbol{\beta}$,

$$\beta \approx \left| \ln\left(\frac{\Pr\{d_k = e_k\}}{\Pr\{d_k \neq e_k\}}\right) \right| \quad (50)$$

sendo $\Pr\{\boldsymbol{d_k} = \boldsymbol{e_k}\}$ a probabilidade que o decodificador tenha feito uma decisão correta, assumindo os valores dentro do intervalo $[0.5,1]$. Quando $\Pr\{\boldsymbol{d_k} = \boldsymbol{e_k}\} \rightarrow 1$, então $\boldsymbol{\beta} \rightarrow \infty$, e quando $\Pr\{\boldsymbol{d_k} = \boldsymbol{e_k}\} \rightarrow 0.5$, então $\boldsymbol{\beta} \rightarrow 0$.

De fato a equação (50) é consistente com conceito de confiabilidade de $\boldsymbol{d_k}$. Quando a probabilidade de $\boldsymbol{d_k}$ ter sido decidido corretamente tende a 1, a confiabilidade de $\boldsymbol{d_k}$ tende a infinito, e quando ela tende a 0.5, então a confiabilidade tende a zero.

De fato $\boldsymbol{\beta}$ pode ser visto como um valor médio da confiabilidade de $\boldsymbol{d_k}$ nas situações em que não foi encontrada uma palavra competidora no subconjunto $\boldsymbol{U}'$, enquanto que quando a palavra competidora é encontrada, então a eq.(48) fornece uma estimação bit-a-bit da confiabilidade. Claramente a eq.(48) é mais precisa que a eq.(49). No entanto, a situação em que um cálculo mais preciso se justifica mais fortemente é quando $C_{\min}^{\neq d_k}$ está a uma distância levemente superior da sequencia recebida $\boldsymbol{Y}$ em relação a $\boldsymbol{D}$. Nesse caso, o cálculo da confiabilidade se dará através da eq.(48). Por outro lado, quando a palavra competidora se encontra distante da sequência recebida $\boldsymbol{Y}$, o valor médio da confiabilidade se mostra como uma boa aproximação.

O valor ótimo de $\boldsymbol{\beta}$ a ser utilizado durante o processo de decodificação iterativa deve ser função da probabilidade de erro de bit (BER). Nos resultados apresentados em [3], $\boldsymbol{\beta}$ é fixado como um vetor cujos valores evoluem com os passos de iteração da decodificação, sendo uma porcentagem do valor médio normalizado da informação extrínseca que é obtida a partir da eq.(48) em cada passo, sendo que esta porcentagem evolui a cada iteração até chegar a 100%.

Os valores do vetor $\boldsymbol{\beta}$ apresentados em [3] foram obtidos de maneira experimental. Foi relatado ainda que os resultados indicaram que não houve degradações significativas no desempenho quando os valores de $\boldsymbol{\beta}$ foram modificados em $\pm 10\%$.

2.5.2-O processo iterativo de decodificação turbo

Considerando a decodificação de um código produto, transmitido através de um canal gaussiano. Seja $[\boldsymbol{R}]$ a matriz na recepção correspondente à matriz de transmissão $[\boldsymbol{E}]$. O primeiro decodificador realiza a decodificação *soft* nas linhas (ou colunas) da matriz recebida $[\boldsymbol{R}]$. Essa decodificação é realizada utilizando-se o algoritmo de Chase e as saídas do decodificador são calculadas utilizando a eq.(48) ou a eq.(49). Subtraindo a entrada da saída é obtida a matriz de informação extrínseca $[\boldsymbol{W}(2)]$, sendo que o índice 2 indica que esta será a informação extrínseca utilizada para a segunda decodificação do código produto.
Dessa forma, a entrada *soft* para a decodificação no sentido das colunas (linhas) da segunda decodificação do código produto é dada por:

$$[\boldsymbol{R}(2)] = [\boldsymbol{R}] + \boldsymbol{\alpha}(2)[\boldsymbol{W}(2)] \quad (51),$$

Sendo $\boldsymbol{\alpha}(2)$ um fator de escala que leva em consideração o fato de que o desvio padrão das amostras da matriz $[\boldsymbol{R}]$ e da matriz $[\boldsymbol{W}]$ são diferentes, [37] e [1]. A justificativa é que o desvio padrão da informação extrínseca é elevado nos primeiros passos de iteração, e decresce ao longo dos passos. O uso desse fator é justificado para reduzir o efeito da informação extrínseca nos primeiros passos de decodificação quando a BER é relativamente elevada. Dessa forma, α é ajustado de maneira a possuir baixos valores nos primeiros passos e ter seu valor aumentado à medida que a BER tende a zero. Nos resultados apresentados em [3], o vetor α utilizado foi $\alpha(m) = [0.0, 0.2, 0.3, 0.5, 0.7, 0.9, 1.0, 1.0]$.

O algoritmo descrito acima pode ser generalizado pelo cascateamento de decodificadores elementares, conforme ilustrado na figura 16.

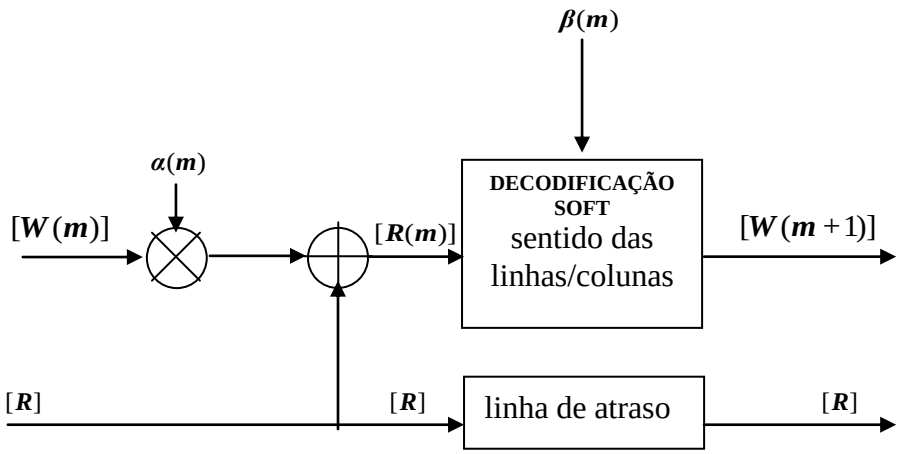


Figura 16 – Decodificador elementar.

A figura 17 ilustra o processo iterativo no sentido das linhas e colunas no processo de decodificação turbo em um ciclo completo.

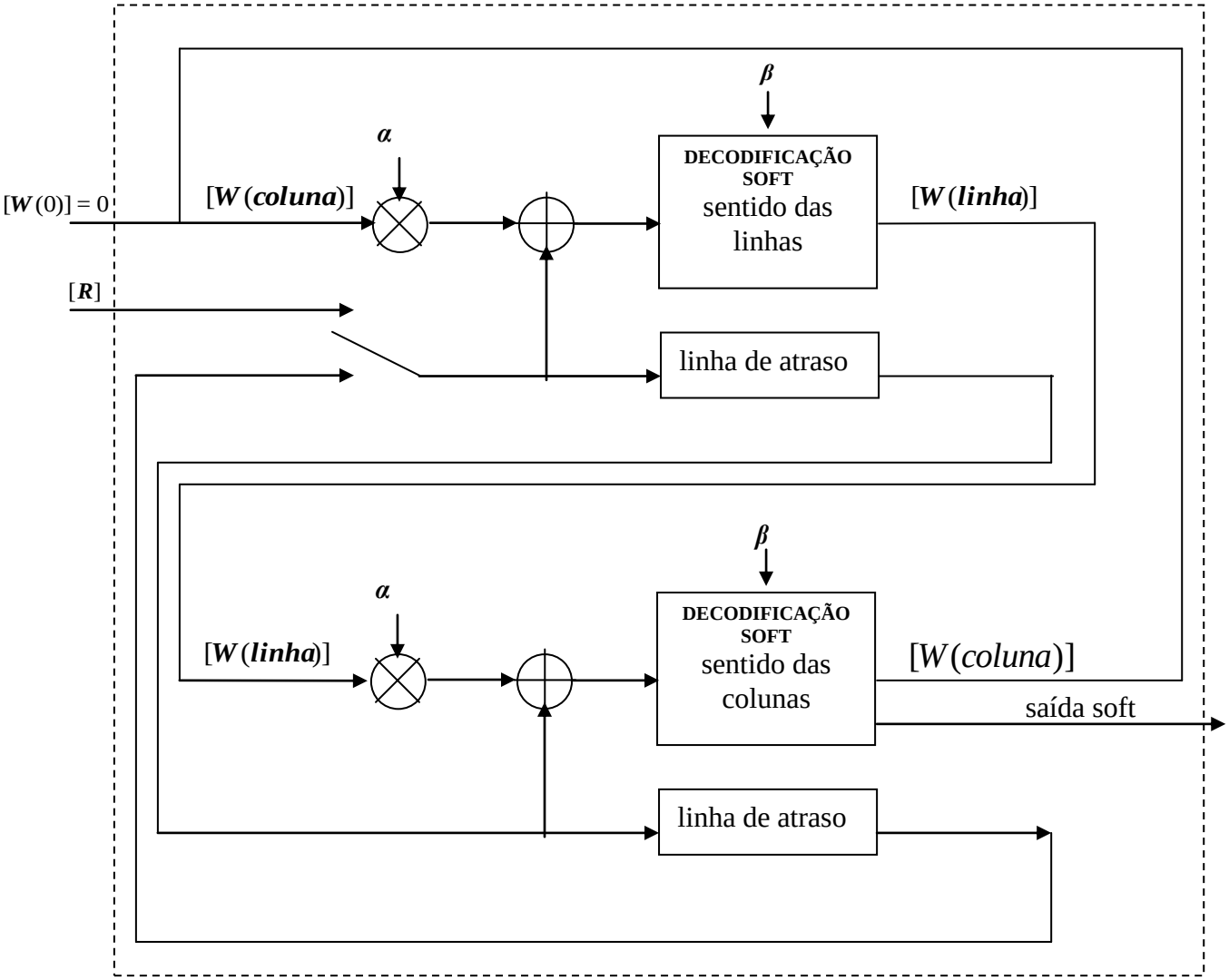


Figura 17- Esquema para ciclo completo de decodificação turbo. Fonte: Autor

Dessa forma, a matriz de recepção $[\boldsymbol{R}]$ é decodificada por sucessivas iterações no sentido das linhas/colunas. A cada nova iteração há uma redução na BER, mas o ganho de desempenho a cada nova iteração vai sendo reduzido, e torna-se pouco significativo, a partir de um certo número de iterações. Desta forma o número de iterações que serão realizadas, torna-se uma relação de compromisso em termos da aplicação e do desempenho desejado.

3 A DECODIFICAÇÃO TURBO UTILIZANDO A ABORDAGEM GEOMÉTRICA

3.1 INTRODUÇÃO

Neste capitulo é apresentado um algoritmo de decodificação baseado nos algoritmos de Chase e Pyndiah, porém sob uma ótica geométrica. Tanto na primeira etapa de decodificação do decodificador SISO (*Soft Input Soft Output*) que utiliza o algoritmo de Chase, quanto na segunda etapa quando é realizado o cálculo da informação extrínseca, a abordagem geométrica é utilizada como ferramenta de otimização e simplificação do processo de decodificação.

Antes da apresentação dos algoritmos utilizando a abordagem geométrica, é importante salientar que o sistema em análise neste trabalho sempre será caracterizado por um esquema de modulação QAM com cardinalidade $L = 2^m$; $m \in \mathbb{Z}^*$, utilizando um código produto, tendo como código constituinte um código corretor de erro Reed-Solomon de comprimento (n,k), com capacidade de correção $t = 1$, e $d = 3$. Para fins de análise e por questão de simplicidade, será considerado ainda $n = 2^{m-1}$, ou seja uma relação de 1 para 1 entre os elementos de campo do código de Reed-Solomon e os símbolos do alfabeto de modulação. A aplicação do algoritmo geométrico para códigos de Reed-Solomon com $t > 1$ ou que utilizem elementos de campo que possuam uma relação de multiplicidade diferente de 1 em relação a cardinalidade do alfabeto de modulação são possíveis, sendo uma extensão dos conceitos apresentados.

3.2 OS CÓDIGOS DE REED-SOLOMON E O ARRANJO NA MATRIZ PRODUTO

Códigos de Reed-Solomon são códigos BCH não binários com elementos de campo pertencentes a $GF(2^m)$, sendo m um inteiro positivo. Os principais parâmetros que caracterizam o código podem ser descritos como (n,k,d), sendo n o comprimento da palavra código, k é o número de símbolos de informação e d é a distancia de Hamming mínima do código.

Dados dois códigos de bloco lineares $\boldsymbol{C}_1$ e $\boldsymbol{C}_2$ com os parâmetros (n_1,k_1,d_1) e (n_2,k_2,d_2), o código produto associado conforme visto no capítulo 3, terá parâmetros (n_1n_2,k_1k_2,d_1d_2).

Um alfabeto de modulação QAM é formado por $\boldsymbol{L}$ símbolos onde $\boldsymbol{L}$ = cardinalidade do alfabeto e normalmente é uma potencia de dois, de tal forma que cada símbolo é um elemento que carrega $\log_2(L)$ bits.

No processo de montagem da matriz de codificação a ser enviada no canal, uma escolha em que o código de Reed-Solomon com elementos de campos pertencentes a $\boldsymbol{GF}(2^{\boldsymbol{m}})$ e a cardinalidade do alfabeto dos símbolos de modulação estabeleçam uma relação de multiplicidade, favorece o mapeamento geométrico tanto na codificação quanto no processo de decodificação turbo que será implementado posteriormente, embora tal relação não seja estritamente necessária.

Sem perda de generalidade, será assumido a utilização de um código de Reed-Solomon associado de tal forma que os elementos sobre $\boldsymbol{GF}(2^{\boldsymbol{m}})$ sejam tais que $2^{\boldsymbol{m}} = \boldsymbol{L}$, e $\boldsymbol{C}_1 = \boldsymbol{C}_2$. Então temos uma relação direta unitária entre o símbolo codificado e o símbolo transmitido no canal. Essa associação poderia ter sido feita também se estabelecendo uma relação de multiplicidade em relação a cardinalidade da constelação ou ainda por dimensão ($\boldsymbol{L}/2$).

A figura 18 é uma representação da matriz produto associada a tal esquema de codificação.

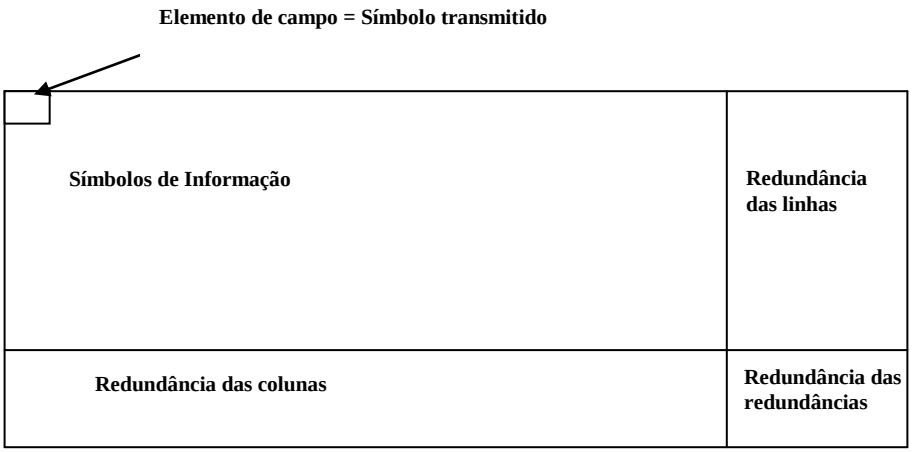


Figura 18- Matriz produto associada ao símbolo não binário. Fonte: Autor

Considerando um canal AWGN e tendo a matriz sido codificada, cada símbolo na recepção corresponderá a um elemento de campo associado. Dessa forma, a confiabilidade do símbolo de informação recebido pode ser medida diretamente sobre o módulo do vetor recebido.

3.3-O DECODIFICADOR SISO

O decodificador SISO (*Soft Input Soft Output*) consiste no bloco principal no processo iterativo de decodificação turbo. A cada iteração este bloco irá gerar a informação extrínseca para o próximo passo de iteração a partir da matriz formada pelos símbolos de modulação recebidos do canal e da informação extrínseca gerada no passo de decodificação anterior. Dessa forma, esse bloco tem como entradas a matriz contendo a informação extrínseca do passo anterior e a matriz contendo os símbolos recebidos do demodulador. A saída é um novo conjunto de valores fornecidos na forma da matriz de informação extrínseca para a próxima iteração.

A primeira etapa de decodificação do decodificador SISO faz uso do algoritmo de Chase, porém utilizando uma abordagem geométrica para a geração dos vetores de teste, conforme será descrito a seguir.

3.3.1 O Algoritmo de Chase utilizando a abordagem geométrica

O algoritmo de Chase conforme descrito na seção 2.4, tem por função gerar um subconjunto de palavras código candidatas, dentre todas as palavras pertencentes ao código corretor utilizado, localizadas na vizinhança da sequência recebida, e que dessa forma possuam maior probabilidade de corresponder à sequência enviada segundo o critério de máxima verossimilhança. Mais precisamente, o algoritmo de Chase irá realizar uma varredura das possíveis palavras código que distem de até $d-1$ posições da sequência recebida, sendo d a distancia mínima do código utilizado.

Os três algoritmos propostos por Chase em [4] para realização dessa varredura diferem entre si basicamente em relação ao tamanho do conjunto de palavras candidatas gerado, representando uma relação de compromisso entre a probabilidade de o conjunto conter a palavra enviada, e a complexidade de decodificação.

O algoritmo de Chase estabelece os critérios de formação dos vetores de teste para realização da varredura a partir da confiabilidade medida de forma analógica, porém sob uma ótica binária, ou seja, analisa-se a confiabilidade associada ao bit.

Essa relação é direta para um sistema que utiliza sinalização binária. De fato em [4], é realizada uma análise sobre um sistema binário antipodal. E conforme sugerido como procedimento para a geração dos vetores de teste, para sistemas puramente binários, a combinação de 0's e 1's associada à análise das posições de menor confiança são a estratégia adequada para a formação dos vetores de teste, variando-se apenas o tamanho do conjunto de vetores gerados para a definição dos algoritmos I, II e III proposto em [4].

O algoritmo proposto neste trabalho leva em consideração um parâmetro adicional que não é considerado no algoritmo proposto por Chase para a formação dos vetores candidatos. A posição geométrica dentro do espaço euclidiano do símbolo recebido.

O algoritmo que será descrito além do critério de confiabilidade dos símbolos recebidos, utiliza a distância euclidiana dos símbolos de menor confiabilidade em relação aos demais símbolos do espaço formado pelo alfabeto de modulação, para a

determinação dos vetores de teste. O que consiste no critério de máxima verossimilhança dentro do espaço euclidiano.

A utilização da abordagem geométrica permite reduzir a quantidade de palavras candidatas em relação a uma abordagem pragmática binária para uma probabilidade equivalente de se encontrar a palavra enviada dentro do conjunto gerado, ou ainda aumentar a probabilidade do conjunto formado conter a palavra enviada em relação à abordagem pragmática binária, para uma mesma quantidade de palavras.

A justificativa é direta, pois uma vez que a abordagem binária não leva em consideração o posicionamento geométrico dos símbolos recebidos, à exceção do algoritmo I de Chase que considera todas os possíveis vetores de peso até $d-1$ e que já para o caso binário não constitui a melhor solução. Os demais algoritmos gerarão vetores de teste que corresponderão a símbolos que não estarão associados diretamente ao critério de máxima verossimilhança. Os gráficos a seguir ilustram estas situações.

Os gráficos 19 e 20 representam o resultado para decodificação *soft* em um canal AWGN, para um esquema de decodificação utilizando um código RS(15,13) com $\left\lfloor \frac{d-1}{2} \right\rfloor = 1$, aplicado sobre uma modulação 16QAM.

Os resultados para as curvas de BER (*Bit Error Rate*) mostrados gráfico da figura 19, demonstram o ganho de desempenho entre a utilização do algoritmo II de Chase (CHASE II), e a aplicação do algoritmo geométrico (G16P4), para o mesmo número de vetores de teste gerados.

Na abordagem clássica utilizando algoritmo II de Chase, foram geradas todas as 16 possíveis combinações na posição correspondente ao símbolo de menor confiança recebido.

Na abordagem geométrica, o algoritmo determina até 4 símbolos vizinhos para cada símbolo substituído na geração dos vetores de teste. Dessa forma, os quatro símbolos de menor confiança são substituídos pelos quatro símbolos mais próximos no espaço euclidiano, segundo o critério de máxima verossimilhança.

Os resultados de simulação mostram um ganho em torno de 0.5dB do esquema de decodificação *soft*, utilizando a abordagem geométrica para um mesmo esforço computacional, comparado com a abordagem clássica.

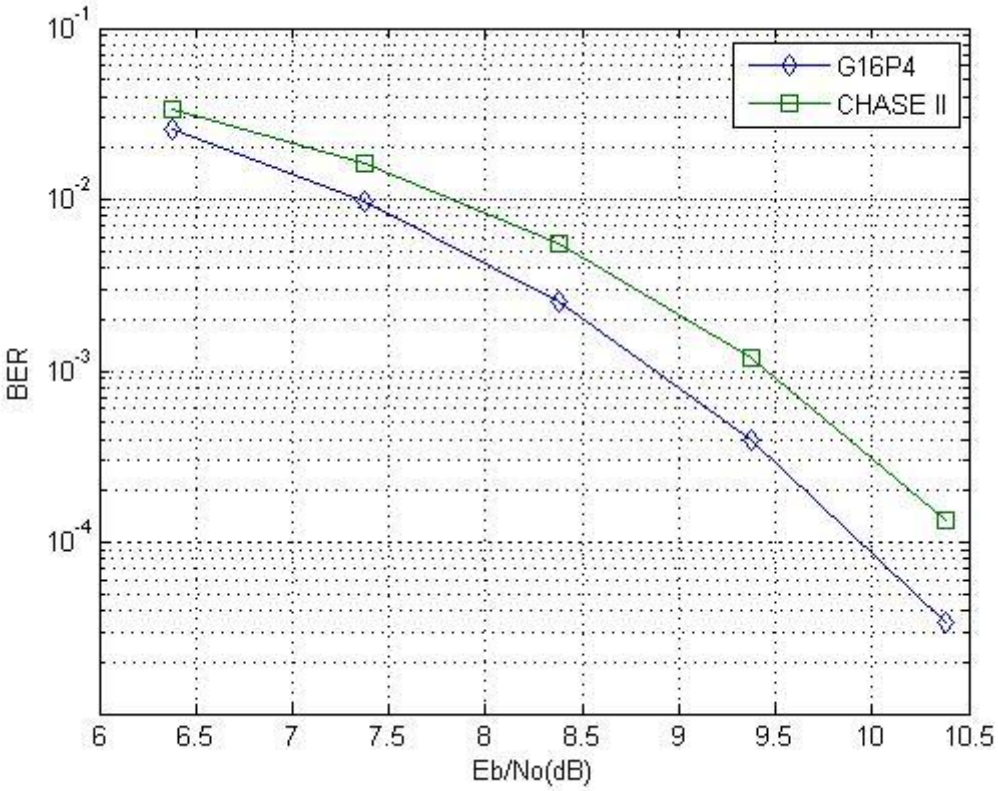


Figura 19 -Decodificação *soft* utilizando abordagem clássica e abordagem geométrica para um mesmo número de vetores de teste. Fonte: Autor

No gráfico da figura 20, o algoritmo geométrico foi restringido para substituição somente do símbolo de menor confiabilidade conforme a abordagem clássica do algoritmo II de Chase, a escolha dos símbolos para gerar os vetores de teste, seguiu o critério de máxima verossimilhança, limitando-se a escolha apenas aos 4 símbolos vizinhos ao símbolo substituído. Nesse caso, os resultados de desempenho são praticamente os mesmos conforme pode ser observado na figura 20. No entanto o conjunto dos vetores de teste passou de 16 para 4.

Para altas relações sinal ruído a redução no número de vetores de teste para uma modulação com cardinalidade $\mathrm{L} = 2^{\mathrm{m}}$, pode chegar a $2^{\mathrm{m}-2}$, o que representa uma redução significativa no esforço de processamento.

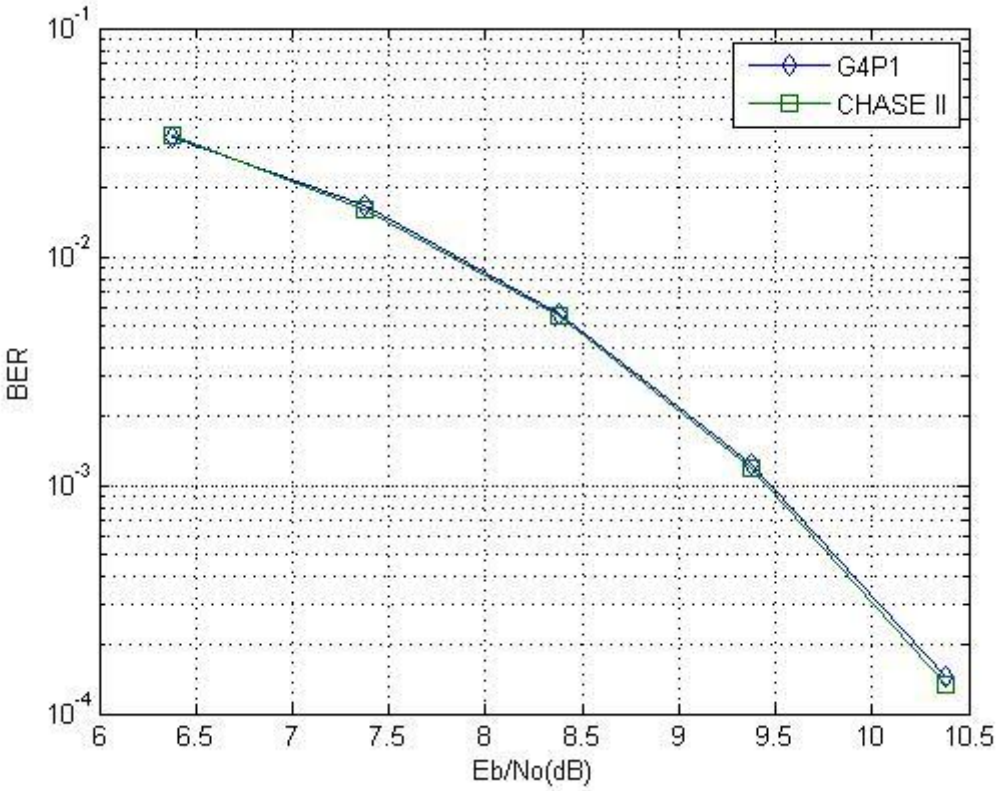


Figura 20 - Decodificação *soft* utilizando abordagem clássica e geométrica para um mesmo resultado de desempenho. Fonte: Autor

O critério de seleção dos símbolos a serem substituídos e a quantidade de símbolos que é utilizada para cada substituição, pode levar em consideração adicionalmente, informações tais como a direção do vetor de ruído associado ao símbolo de menor confiança dentro do espaço euclidiano e a relação sinal ruído em que se está operando como fatores para refinamento e otimização para a formação do conjunto de vetores de teste. Essa análise, no entanto, não será realizada neste trabalho, podendo ser objeto de estudos futuros.

A figura 21 ilustra possíveis escolhas baseadas na geometria da constelação utilizada. No caso de uma constelação retangular como da figura 21, os símbolos mais próximos distam de valores fixos dE, $\sqrt{2}dE$, etc. Sendo que dE representa a menor distância entre dois símbolos no espaço euclidiano. Em algumas situações quando o símbolo a ser substituído se encontra na região mais externa da constelação, o algoritmo poderá buscar candidatos distando de $\sqrt{2}dE$ do símbolo a ser substituído, enquanto que nos casos nos quais os símbolos a serem substituídos

estão na região interna da constelação, o algoritmo buscará símbolos vizinhos que distem de dE como resultado do critério de máxima verossimilhança.

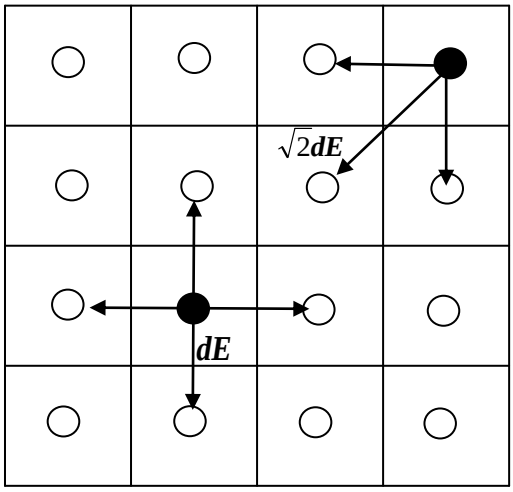


Figura 21- Constelação 16-QAM. Escolhas possíveis de símbolos para formar as sequências candidatas. Fonte: Autor

3.3.2 O Cálculo da informação extrínseca

A segunda etapa de decodificação do decodificador SISO faz uso do algoritmo de Pyndiah, conforme será descrito a seguir.

Nesta etapa do processo de decodificação, é utilizado um processo similar ao proposto por Pyndiah, porém a manipulação se dá no nível do símbolo recebido do canal, ou do elemento de campo de Galois.

Para cada símbolo r_k que compõe a sequência recebida associada a uma linha/coluna da matriz produto, é realizado o cálculo da informação extrínseca w_k correspondente.

Existem duas situações possíveis no cálculo da informação extrínseca associada a um símbolo d_k da palavra vencedora D. A primeira situação, é quando existe uma palavra concorrente no conjunto de palavras candidatas geradas pelo algoritmo de Chase, tal que $c_k^i \neq d_k$, e a segunda situação é quando não existe palavra

concorrente que satisfaça, isto é, $c_k^i \neq d_k$. Cada uma das situações será tratada a seguir.

Existe palavra concorrente

Conforme visto, no algoritmo de Pyndiah o cálculo da informação extrínseca quando existe uma palavra competidora $C_{\min}^{\neq d_k}$ pertencente a $S^{\neq d_k}$, sendo $S^{\neq d_k}$ o conjunto das palavras código $\{C^i\}$ tal que $c_k^i \neq d_k$, é dado pela eq.(48).

A eq.(48), é baseada na distância euclidiana quadrática, aplicada sobre uma sequência de símbolos binários $\{-1,+1\}$, mapeados a partir do envio de uma de elementos binários $\{0,1\}$, ou seja $0 \rightarrow -1, \text{e}\, 1 \rightarrow +1$.

No caso do algoritmo geométrico, os símbolos consistem em vetores dentro do espaço euclidiano, e serão representados por meio de vetores dentro do espaço euclidiano formado e limitado pelos limites gerados pelos conversores A/D dos receptores utilizados.

Para fins de análise, é considerado como limites externos do espaço euclidiano, o espaço formado pelos valores dos símbolos de modulação mais externos acrescidos de $dE/2$, sendo dE a menor distância entre dois símbolos dentro do espaço euclidiano formado pelo alfabeto de modulação utilizado. Essa escolha não representa limitação à aplicação do algoritmo de decodificação, uma vez que pontos que estejam além desses limites serão limitados às proximidades dos pontos mais externos, e do ponto de vista do critério de máxima verossimilhança, isso não representará alteração na escolha do símbolo recebido.

A figura abaixo representa o espaço formado para uma modulação 16QAM.

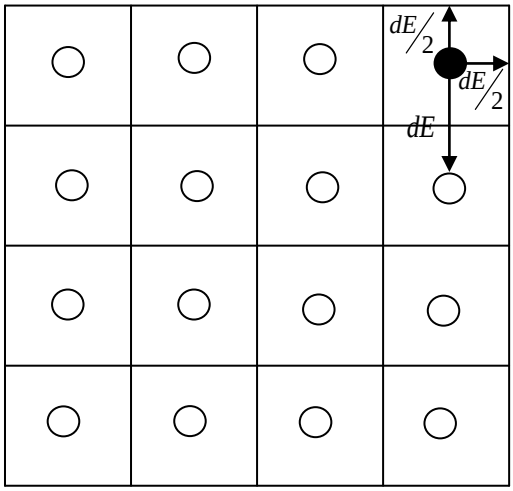


Figura 22- Constelação 16-QAM: Limites do espaço euclidiano utilizado no processo de decodificação. Fonte: Autor

Do ponto de vista geométrico, a informação extrínseca pode ser interpretada como uma quantidade que atua sobre o vetor de ruído do vetor recebido, deslocando-o ao longo das etapas de iteração para a região final de decisão pertencente a um dos símbolos de modulação do espaço euclidiano.

A distância euclidiana quadrática utilizada no algoritmo de Pyndiah, derivada a partir de uma aproximação da função densidade de probabilidade condicional do canal. Apesar de não representar uma métrica do ponto de vista de medida de distâncias, é uma medida bastante utilizada e apropriada quando as distâncias tem apenas que serem comparadas, como no caso de um alfabeto binário, sendo bastante conveniente devido ao fato de não necessitar da operação de extração da raiz quadrada ao final do cálculo, sendo necessárias apenas operações de adição e multiplicação entre as componentes dos vetores envolvido.

Para operações com distâncias, a distância euclidiana mostra-se como a métrica mais apropriada para uso. Conforme descrito, o algoritmo geométrico está baseado no conceito de operações sobre um espaço de vetores. Dessa forma, o cálculo proposto para a informação extrínseca se baseia na distância euclidiana entre os vetores.

A utilização da distância euclidiana no lugar da distancia euclidiana quadrática, implica na operação adicional de raiz após a realização das operações de adição e produto interno entre os vetores. Do ponto de vista de implementação será mostrado que, sem degradação significativa, é possível a realização de aproximações que conduzem a implementações que eliminam essa complexidade.

Para o algoritmo geométrico, o cálculo da informação extrínseca associada a d_k na i-ésima iteração linha/coluna, quando existe uma palavra competidora $C_{\min}^{\neq d_k}$, é feito pela expressão:

$$\Delta w_k^i = \left(\frac{\sqrt{\left|Y - C_{\min}^{\neq d_k}\right|^2} - \sqrt{\left|Y - D\right|^2}}{\gamma}\right)(d_k - r'^{i-1}_k) \quad (52)$$

sendo:

$$w_k^i = w_k^{i-1} + \Delta w_k^i \quad (53)$$

$$r'^i_k = r_k + w_k^{i-1} + \Delta w_k^i \quad (54)$$

$$r'^i_k = r_k + w_k^i \quad (55)$$

Como as operações são sobre vetores no espaço euclidiano, e sendo que a informação extrínseca atua como uma correção sobre o vetor de ruído inicial a cada passo de interação, a fim de se garantir a convergência para o símbolo final de decisão, se faz necessário que a informação extrínseca associada a cada iteração, seja aplicada ao último vetor formado pelo vetor inicialmente recebido mais o valor da informação extrínseca acumulada até a iteração anterior.

A eq.(52) é coerente com o conceito de informação extrínseca quando há uma palavra competidora. A quantidade $\sqrt{\left|Y - C_{\min}^{\neq d_k}\right|^2}$ que representa a distância da sequência recebida em relação a palavra competidora, é uma medida direta da confiabilidade do símbolo d_k, pois quanto mais distante a palavra competidora estiver da sequência recebida, maior é a confiabilidade associada a d_k. De maneira

equivalente a quantidade $\sqrt{|Y - D|^2}$ representa uma contribuição na medida inversa, pois diminui o valor da confiabilidade do símbolo na medida da distancia da palavra eleita em relação à sequência rebebida.

3.3.2.1 A Justificativa do fator γ

O fator γ tem como função normalizar o valor do cálculo da contribuição da informação extrínseca em relação ao maior evento de erro possível. Dessa forma, a determinação dessa quantidade está associada à modulação utilizada e à capacidade de correção do código.

A seguir é feita a análise para determinação dessa quantidade para o caso de utilização de um código de Reed-Solomon e uso de uma modulação 16-QAM. A análise para demais esquemas de codificação e modulação é direta sendo uma extensão dos princípios que serão analisados.

Como este trabalho está concentrado no uso de códigos de Reed-Solomon com capacidade de correção t=1, tem-se que:

Lembrando que distância de Hamming mínima do código é $d_{min} = 3$, e que o algoritmo de Chase permite a correção de erros até $d_{min} - 1 = 2$. Seja D a palavra código vencedora para uma dada sequência recebida R e seja d_D a distância de Hamming da palavra código D em relação a sequência recebida R , e seja d_C a distância entre a sequência recebida R e a palavra concorrente $C_{\min}^{\neq d_k}$, caso seja encontrada uma palavra concorrente. Então três casos serão possíveis:

1º Caso

A distância de Hamming entre a palavra código vencedora e a sequência recebida é zero,($d_D = 0$). Neste caso obrigatoriamente não existe palavra concorrente, pois $d_{min} = 3$ e a capacidade de correção é $d_{min} - 1 = 2$. Nessa situação conforme proposto pelo algoritmo de Pyndiah o cálculo é realizado pela utilização do fator β.

2º Caso

A distância de Hamming entre a palavra código e a sequência recebida é um,($d_D = 1$). Nesse caso, se existir $C_{\min}^{\neq d_k}$, então $d_C = 2$, pois se $d_C = 1$, implica $d_C - d_D \leq 2$, mas $d_{min} = 3$.

3ºCaso

A distância de Hamming entre a palavra código e a sequência recebida é dois ,($d_D = 2$). Nesse caso, se existir $C_{\min}^{\neq d_k}$,então $d_C = 2$.

Resumindo:

$Se\ d_D = 0 \rightarrow \nexists C_{min}^{\neq d_k}$: $1°\ CASO\ \rightarrow \beta\ \ (d_{min} = 3, Chase \rightarrow 2)$

$Se\ d_D = 1, e\ \exists\ C_{min}^{\neq d_k} \rightarrow d_C = 2$ $2°\ CASO\ \ (d_c = 1 \rightarrow d_C - d_D \leq 2, d_{min} = 3)$

$Se\ d_D = 2, e\ \exists\ C_{min}^{\neq d_k} \rightarrow d_C = 2$ $3°\ CASO\ \ (\ d_D < d_C\ )$

A análise das três situações possíveis mostra que a situação que fornece o maior evento de erro possível é o 2º caso, pois no primeiro caso ocorre a utilização do fator β, e pela análise de $\left(\sqrt{\left|Y - C_{\min}^{\neq d_k}\right|^2} - \sqrt{\left|Y - D\right|^2}\right)$, esta expressão é maximizada quando d_C é máximo e d_d é mínimo, o que ocorre no segundo caso analisado.

Fazendo agora a análise do ponto de vista da geometria do espaço euclidiano sob o alfabeto de modulação utilizado, o maior evento de erro simples possível, possui valor igual a $e_{max} = 3.5\sqrt{2}dE$. Ainda analisando o segundo caso, a palavra concorrente obrigatoriamente tem uma de suas posições determinada por uma substituição realizada pelo algoritmo de Chase, ou seja pela substituição por algum símbolo vizinho, sendo que neste caso a distância máxima possível é dada por uma substituição na diagonal, com um evento de erro associado de valor $e_{chase} = 1.5\sqrt{2}d$. Para a palavra vencedora e para os fins do cálculo em questão, a análise é sobre o menor evento de erro possível para o qual $d_D = 1$, que é dado por $e_{min} = 0.5dE$.

Finalmente, γ para uma constelação 16QAM e um código de Reed-Solomon com t=1, é dado por:

$$\gamma_{16QAM} = \max\left(\sqrt{\left|Y - C_{\min}^{\neq d_k}\right|^2} - \sqrt{\left|Y - D\right|^2}\right) = \left(\sqrt{29} - 0{,}5\right)dE \quad (56)$$

A figura 23 ilustra os eventos de erro descritos.

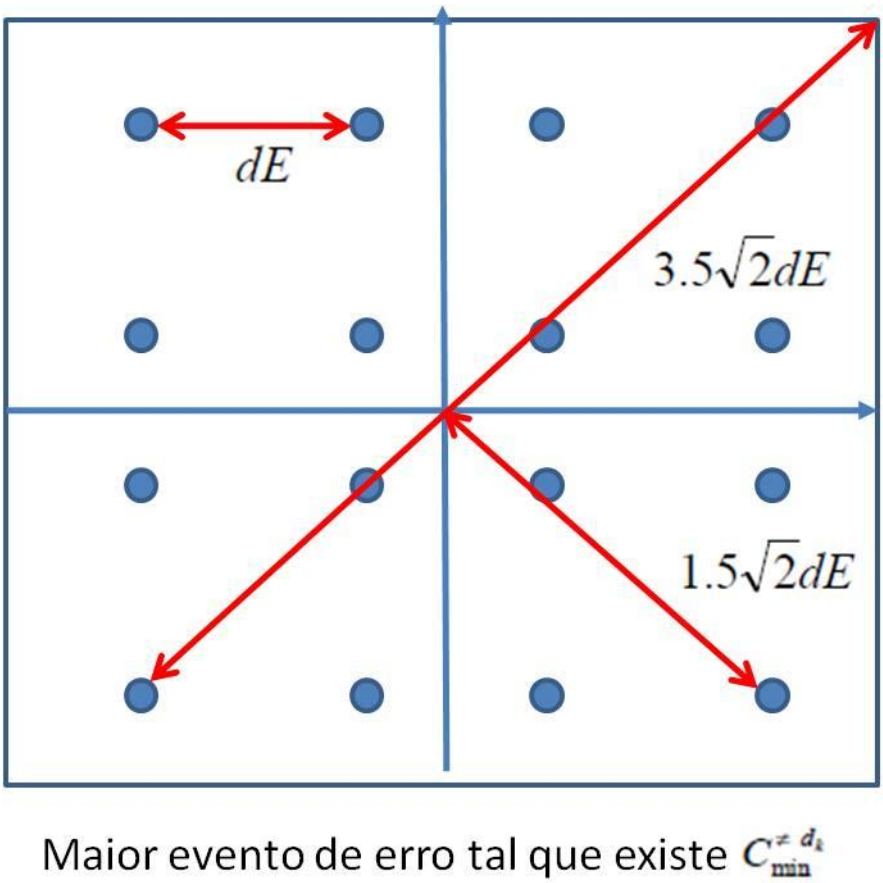


Figura 23- Eventos de erro no espaço euclidiano. Fonte: Autor

Não existe palavra concorrente

De maneira similar à justificativa apresentada para algoritmo proposto por Pyndiah descrito na seção 2.5.1, haverá situações em que no conjunto de palavras código candidatas geradas na primeira etapa do processo de decodificação utilizando o algoritmo de Chase, poderá não ser encontrada uma palavra competidora para o cálculo da informação extrínseca utilizando a eq.(52). Dessa forma, se não for encontrada palavra competidora $C_{\min}^{\neq d_k}$ pertencente a $S^{\neq d_k}$, sendo $S^{\neq d_k}$ o conjunto das palavras código $\{\boldsymbol{C}^i\}$ tal que $\boldsymbol{c}_k^i \neq \boldsymbol{d}_k$, então a informação extrínseca associada é calculada pela expressão :

$$\Delta w_k^i = \beta(d_k - r'^{i-1}_k) \quad (57)$$

O valor de β representa um valor médio da confiabilidade associada ao símbolo d_k toda vez que uma palavra competidora não é encontrada. Conforme justificado por Pyndiah em [3], embora seja uma estimativa menos precisa da informação extrínseca associada ao símbolo, representa uma aproximação razoável quando não for encontrada uma palavra competidora.

O valor de β utilizado nas simulações realizadas neste trabalho seguiram o principio descrito acima de valor médio de confiabilidade, tendo seus valores ajustados por tentativa e erro. Uma descrição mais detalhada das estratégias para definição dos valores de β utilizados neste trabalho é realizada no capítulo que trata dos resultados de simulação.

3.4 O ALGORITMO DE DECODIFICAÇÃO SISO

Conforme descrito anteriormente, no processo decodificação, o decodificador SISO utiliza o algoritmo de Chase e Pyndiah com abordagem geométrica para sequência de operações que será descrita a seguir:

Seja E o vetor associado a uma linha/coluna da matriz enviada no lado de transmissão, com $E = (e_1, e_2, ..., e_n)$, e seja cada símbolo transmitido representado pelo vetor complexo $e_i = a_i + jb_i$, tal que $\boldsymbol{a_i}$ e $\boldsymbol{b_i}$, sejam as componentes em fase e em quadratura associadas aos símbolos que compõe o alfabeto de modulação.

Seja R o vetor correspondendo uma linha/coluna da matriz produto recebida, tal que $R = E + Z$, sendo $\boldsymbol{Z} = (\boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z_n})$ o vetor de ruído complexo gaussiano associado ao vetor recebido do canal.

.

Seja $R^{'i}$ o vetor correspondendo a uma das linhas/colunas da matriz produto no i-ésimo passo de iteração[1], tal que $R^{'i} = R + W^{i}$, onde W^{i} é a correspondente linha/coluna da matriz de informação extrínseca na entrada do decodificador SISO[2] no i -ésimo passo de iteração, e seja $r^{'i}{}_{j} = x + jv$ o valor da j-éssima posição do vetor $R^{'i}$, e $\hat{e}_{j}$ o valor teórico do símbolo do alfabeto de modulação associado, obtido através da decodificação hard.

Então a confiança associada a cada símbolo recebido no i –ésimo passo de iteração é obtida simplesmente através da medida do módulo vetor de ruído associado dada pela eq.(58) conforme abaixo.

$$\left| r^{'i}{}_{\mathrm{j}} - \hat{e}_{j} \right| \quad (58)$$

Na primeira etapa da decodificação SISO é utilizado o algoritmo de Chase e formado o conjunto das sequências candidatas.

O algoritmo é descrito abaixo:

1. Seja $R^{''i-1}$ o vetor correspondendo a uma das linhas/colunas da matriz produto no $(\mathrm{i}-1)$ – ésimo passo de iteração. Encontre os símbolos de menor confiança, através do cálculo de (58).

2. Substitua estes símbolos na sequencia $R^{''i-1}$ recebida pelos símbolos de modulação mais próximos, segundo o critério de menor distância euclidiana, formando o conjunto de vetores de teste.

3. Aplique a decodificação hard segundo as regras do código corretor de erros utilizado e forme o conjunto S das palavras código candidatas.

4. Encontre a palavra código candidata D, mais próxima segundo o critério de máxima verossimilhança.

---

[1] No passo zero R corresponde ao vetor recebido do canal

[2] Este valor na prática corresponde a matriz de informação extrínseca ponderada pelo fator $\propto$ e será discutido posteriormente

A segunda etapa da decodificação SISO consiste no cálculo da informação extrínseca.

O algoritmo é descrito abaixo:

Para cada símbolo d_k da palavra D encontrada na etapa anterior, verifique se existe no conjunto S formado, uma palavra competidora $C_{\min}^{\neq d_k}$ tal que $c_k^i \neq d_k$ e $C_{\min}^{\neq d_k}$ seja a palavra mais próxima da R'^{i-1} dentro do conjunto formado que satisfaça $c_k^i \neq d_k$.

1- Se existe $C_{\min}^{\neq d_k}$, então a informação extrínseca associada é dada por:

$$\Delta w_k^i = \left(\frac{\sqrt{\left|Y - C_{\min}^{\neq d_k}\right|^2} - \sqrt{\left|Y - D\right|^2}}{\gamma}\right)(d_k - r'^{i-1}_k) \quad (59)$$

2- Se não existir $C_{\min}^{\neq d_k}$ então a informação extrínseca é dada por:

$$\Delta w_k^i = \beta(d_k - r'^{i-1}_k) \quad (60)$$

3- Depois de realizada toda a varredura linha/coluna, a matriz de informação extrínseca que é fornecida na saída do decodificador SISO para o próximo passo de iteração é formada através de:

$$W_k^i = W_k^{i-1} + \Delta W_k^i \quad (61)$$

3.5 O PROCESSO DE DECODIFICAÇÃO TURBO

Seja o processo iterativo de decodificação linha/coluna que é realizado durante o processo de decodificação turbo, utilizando o decodificador SISO descrito na seção anterior.

Seja $[R]$ a matriz produto corresponte aos símbolos recebidos do canal, e $\left[W^{i-1}\right]$ a matriz correspondente da informação extrínseca no passo anterior. No i-ésimo passo de iteração a entrada *soft* do decodificador é dada por:

$$\left[R'^{i-1}\right] = [R] + \alpha(i-1) \times \left[W^{i-1}\right] \quad (62)$$

E ao final do i-ésimo passo de iteração o decodificador SISO irá fornecer na sua saida o novo valor para informação extrínsica dado pela matriz W^{i}.

Note que na entrada do decodificador SISO, é utilizado um fator de ponderação multiplicativo $\boldsymbol{\alpha}$ aplicado à matriz de informação extrínseca. A justificativa é a mesma apresentada em [3] e [38]. O fator $\boldsymbol{\alpha}$ tem por função diminuir a contribuição da informação extrínseca nos primeiros passos de iteração, quando o desvio padrão da informação extrínseca é elevado.

A figura 24 representa um desenho simplificado correspondendo a operação do decodificador SISO em um passo de iteração.

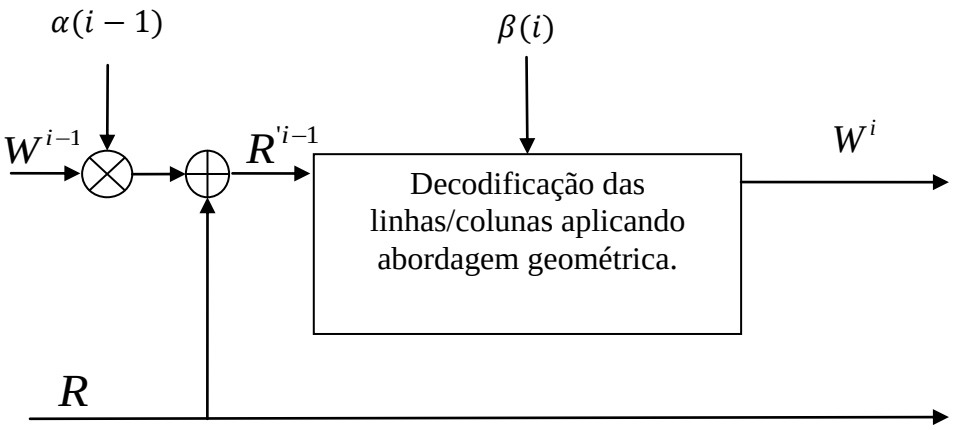


Figura 24 – Um passo de decodificação SISO. Fonte: Autor

Este processo se repete através de sucessivas varreduras no sentido das linhas/colunas da matriz produto, até que algum critério de parada estabelecido seja atingido, normalmente associado a algum critério de desempenho.

4 RESULTADOS DE SIMULAÇÃO

4.1 INTRODUÇÃO

A fim de se verificar os resultados obtidos com a utilização do algoritmo geométrico, foram realizadas simulações em diferentes cenários, buscando analisar a influência dos diversos parâmetros nos resultados de desempenho e na complexidade de implementação.

4.2 RESULTADOS DE SIMULAÇÃO

A seguir serão apresentados os resultados de simulação e a análise dos resultados em cada caso.

4.2.1 Resultados de Simulação utilizando o algoritmo para modulação 16QAM

A figura 25 apresenta os resultados obtidos na aplicação do algoritmo geométrico com parametrizações similar às apresentadas em [9], [10], [11] e [39] para fins de comparação. Existe uma pequena divergência de nomenclatura em relação [39], optou-se por mantê-la, de forma a preservar a conformidade com a nomenclatura apresentada no restante do texto.

De forma a validar a utilização do algoritmo geométrico, foi estabelecido um esquema de transmissão através de um canal AWGN utilizando modulação 16QAM, e um código produto tendo como código constituinte um código RS(15,13).

Os parâmetros a seguir foram utilizados para obtenção dos resultados de simulação mostrados na figura 25.

Número de iterações: 6

$\alpha = [0.06, 0.15, 0.24, 0.35, 0.48, 0.63, 0.9, 1, 1, 1, 1, 1\ ]$

$\beta = 0.15$

$\gamma = 4.9\ dE$

Na figura 25, NCT corresponde a curva teórica de um sistema 16QAM não codificado, RST corresponde a curva teórica de um sistema 16QAM utilizando RS(15,13) e algoritmo de decodificação de Berlekamp, SDG corresponde a curva com os resultados de simulação obtidos utilizando-se o algoritmo *soft* geométrico para RS(15,13), e TDG corresponde a curva com os resultados obtidos utilizando o algoritmo turbo geométrico para o código produto $RS(15,13)^2$.

Nos resultados apresentados em [39], foi utilizada a métrica euclidiana quadrática. Os ganhos adicionais obtidos através do uso da distância euclidiana serão mostrados mais adiante.

Os resultados de simulação mostram um ganho de 5dBs para uma BER $=1x10^{-5}$, e próximo a 6dBs para BER $= 1x10^{-6}$, comparado a um esquema 16QAM não codificado.

Os resultados de simulação mostram ainda os ganhos relativos somente a decodificação *soft* utilizando o algoritmo geométrico. Nesse caso é obtido um ganho próximo a 1,5dB para BER$=1x10^{-6}$, em relação ao decodificação hard, similar aos resultados obtidos em [40], onde é proposto um algoritmo utilizando um esquema de treliças para decodificação *soft* utilizando códigos RS.

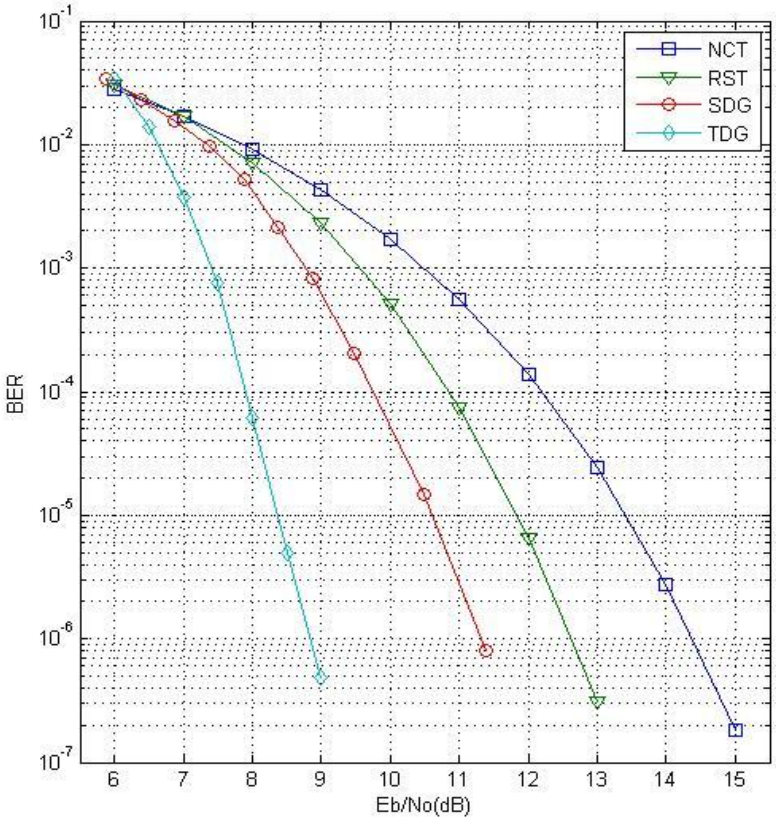


Figura 25 – Decodificação turbo RS(15,13)^2 - 16QAM. Fonte: Autor

4.2.2 Resultados de Simulação utilizando o algoritmo para modulação 256QAM

A figura 26, apresenta os resultados de simulação obtidos para um esquema codificado utilizando um código produto RS(255,253)^2 em um esquema de modulação 256QAM.

Os parâmetros a seguir foram utilizados para obtenção dos resultados de simulação mostrados na figura 26.

Número de iterações: 6

$\alpha =$ [0.06, 0.15, 0.24, 0.35, 0.48 0.63, 0.9, 1, 1, 1, 1, 1]

$\beta = 0.15$

$\gamma = 21.53\ dE$

Os resultados de simulação mostram um ganho em torno de 3.8dBs para uma BER=10^{-6} em relação a um esquema não codificado, e um ganho em torno de 1dB quando é utilizado o algoritmo geométrico somente na decodificação *soft* em relação a uma decodificação *hard* utilizando-se o algoritmo de Berlekamp.

Juntamente com os resultados de simulação estão apresentadas as curvas de BER para decodificação de um código de Reed-Solomon (255,251) com taxa próxima (levemente inferior, r = 0,984313) ao esquema turbo proposto que possui taxa r = 0,984375. Neste caso o esquema turbo apresenta um ganho de 1.5dBs para BER=10^{-6} , e com uma eficiência espectral levemente superior. Note que a decodificação direta *soft* geométrica já apresenta desempenho superior a este esquema codificado. É mostrada também a curva de BER para um código RS(255,247), que possui a mesma distância mínima do esquema turbo, possuindo no entanto uma taxa bem inferior, r = 0,968627, com menor eficiência espectral. Neste é observado ainda um ganho do esquema turbo geométrico da ordem de 0.8dB para BER=10^{-7}.

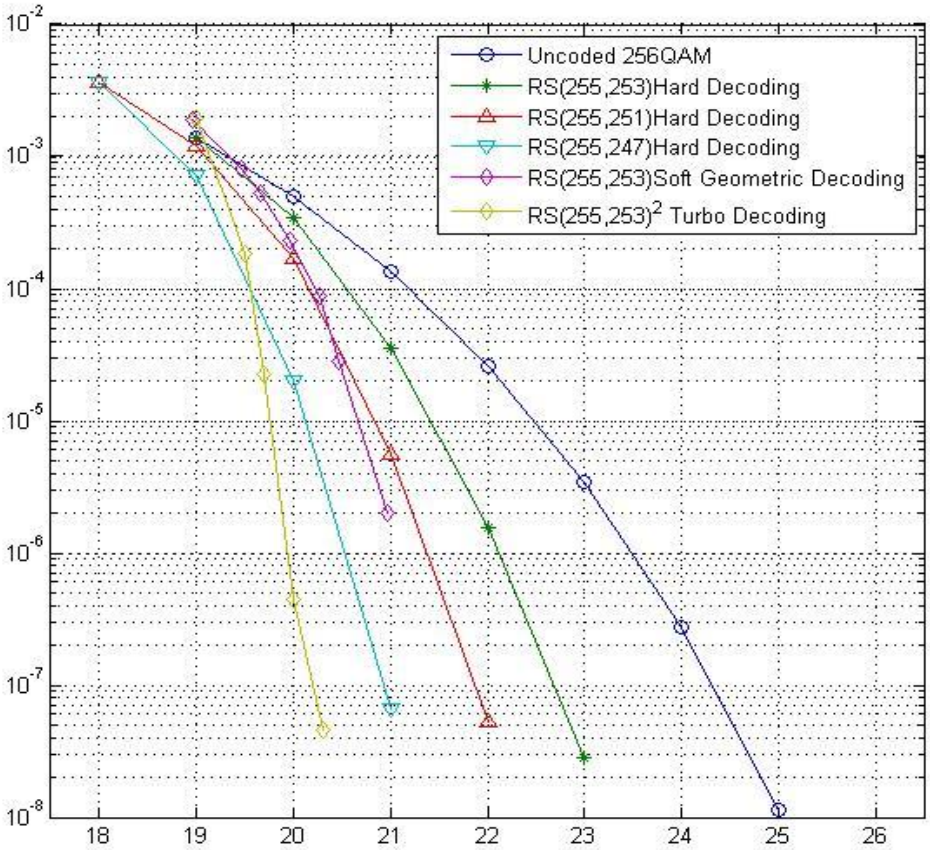
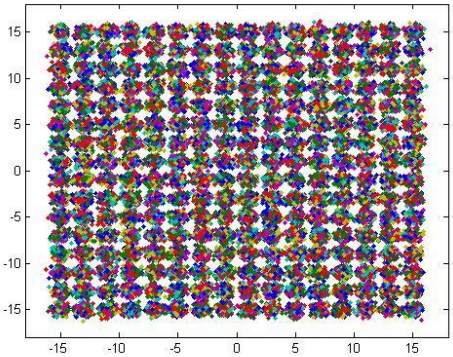


Figura 26 – Decodificação turbo RS(255,253)^2 - 256QAM. Fonte: Autor

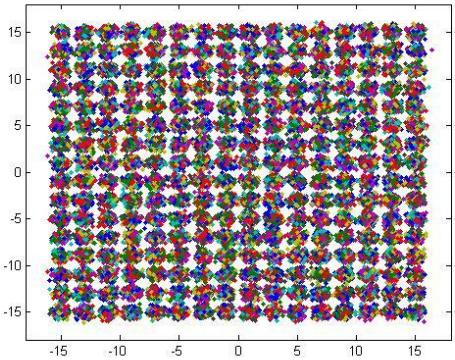
4.2.3 Resultados gráficos dos passos de iteração

A sequência mostrada na figura 27 representa a evolução dentro do espaço euclidiano dos passos de iteração do algoritmo turbo geométrico, obtidos através dos resultados de simulação para uma matriz produto recebida durante o processo de decodificação.

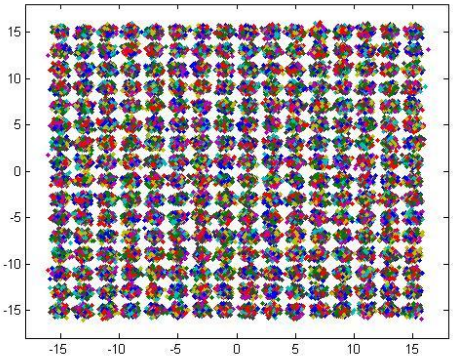
A informação extrínseca atua de forma a reconduzir o símbolo ruidoso na direção da posição geométrica do símbolo correto, conforme ilustrado na figura 27, que mostra a evolução geométrica no espaço euclidiano em 6 passos de iterações sobre a matriz produto para o esquema $RS(255,253)^2$ - 256QAM,e para uma relação sinal ruído inicial de $Eb/No = 20dB$.



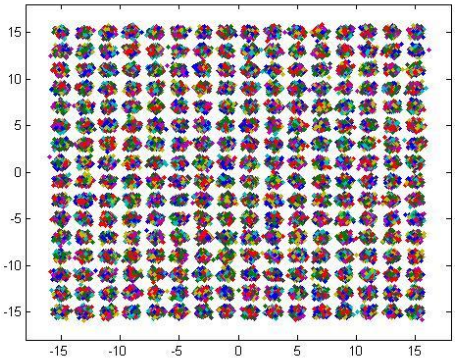
Iteração 1



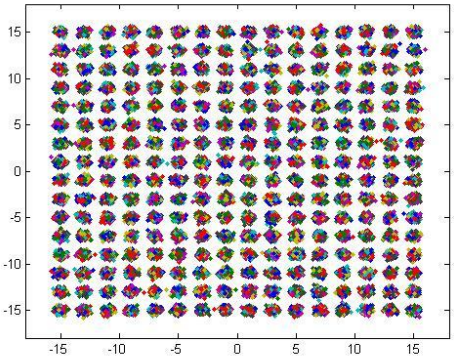
Iteração2



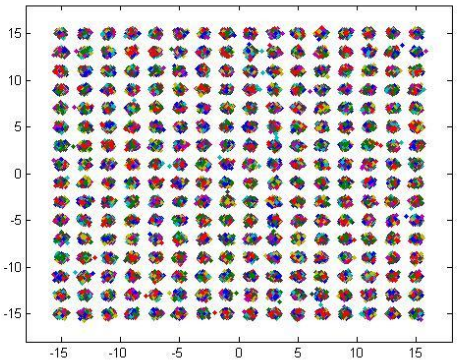
Iteração 3



Iteração4



Iteração 5



Iteração6

Figura 27- Evolução no espaço euclidiano para 6 passos de iteração utilizando abordagem geométrica. Fonte: Autor

4.2.4 A utilização da distância euclidiana e da distância euclidiana quadrática no calculo da informação extrínseca

Os resultados inicialmente obtidos com a aplicação do algoritmo geométrico apresentados em [39], fazem uso da distancia euclidiana quadrática para o cálculo da informação extrínseca, conforme sugerido pelo algoritmo de Pyndiah. Verificou-se posteriormente através de simulações e pelas justificativas já apresentadas na seção 3.3.2, que a distância euclidiana se mostra como métrica mais apropriada para o algoritmo geométrico.

Os resultados de simulação utilizando os mesmos esquemas codificados e mesmos valores de $\alpha, \beta\ e\ \gamma$, mostram ganhos adicionais da ordem de 0.3dB quando o cálculo da informação extrínseca é realizado utilizando a distância euclidiana. As figuras 28 e 29 mostram os resultados comparativos de desempenho entre a utilização da métrica euclidiana e euclidiana quadrática, para o esquema 16QAM, $RS(15,13)^2$ e o esquema 256QAM, $RS(255,253)^2$. MQ denota a curva obtida através do uso da distância euclidiana quadrática, e ME a curva obtida através do uso da distancia euclidiana.

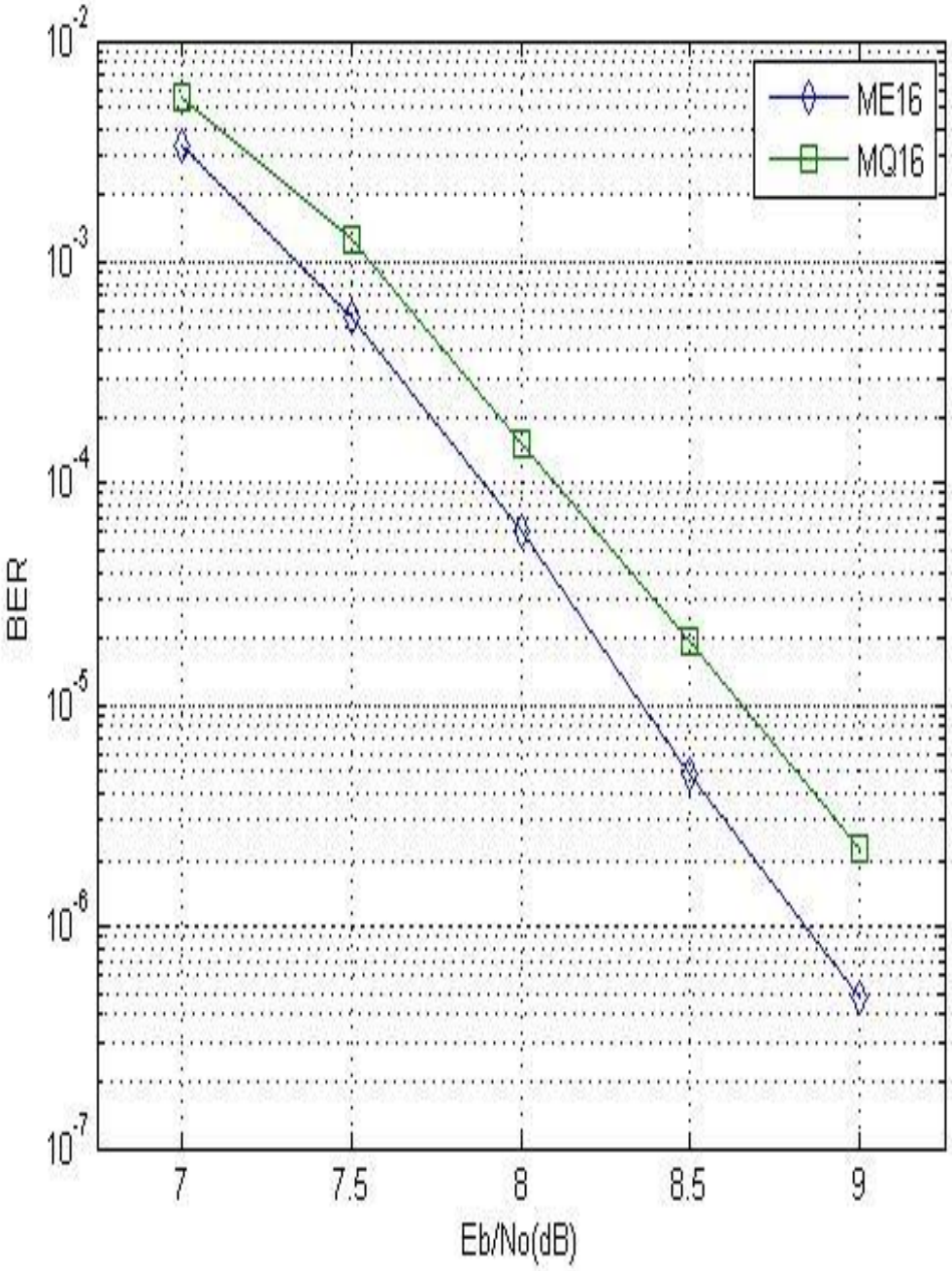


Figura 28 – Resultados comparativos do uso das métricas euclidiana e euclidiana quadrática - 16QAM. Fonte: Autor

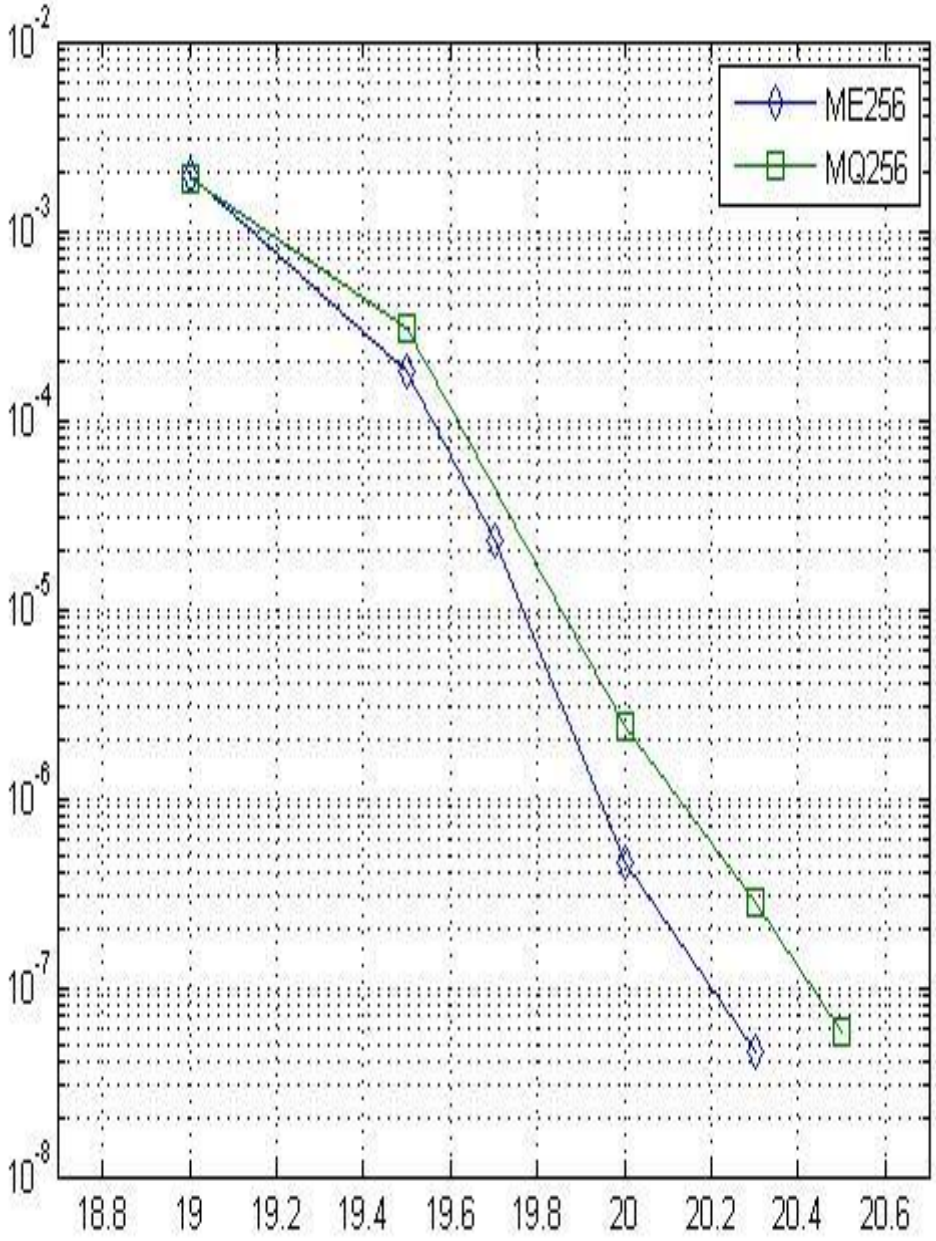


Figura 29 – Resultados comparativos do uso das métricas euclidiana e euclidiana quadrática - 256QAM. Fonte: Autor

4.2.4.1 Considerações de implementação quanto a utilização da distância euclidiana

Foram realizadas simulações de forma a verificar a degradação causada pela utilização de uma tabela com valores quantizados no lugar do cálculo direto da distância euclidiana, como forma de suprimir o cálculo adicional causado pelo uso da distância euclidiana no cálculo da informação extrínseca.

Utilizou-se o esquema 16QAM, $RS(15,13)^2$, para realização destas verificações.

O algoritmo foi modificado de forma que os valores obtidos a partir do cálculo da distância euclidiana quadrática são fornecidos como entrada em uma tabela, que os quantizam em valores fixos de saída de acordo com a faixa de valores a que pertencem, correspondendo a um valor aproximado da distância euclidiana correspondente.

Os resultados de simulação para o esquema testado mostraram resultados satisfatórios já para 8 níveis de quantização, o que corresponde a uma implementação simples sem aumento significativo de processamento.

A figura 30 mostra os resultados de simulação obtidos utilizando esta abordagem. A curva MEgran corresponde aos resultados obtidos quando a quantização é utilizada, enquanto que a curva ME representa os resultados sem aplicação da quantização. Observa-se que a degradação para uma BER em torno de $1x10^{-5}$ é praticamente desprezível, sendo inferior a 0,1dB. Desta forma o uso da métrica euclidiana comparativamente ao uso da métrica euclidiana quadrática, apresenta ganhos.

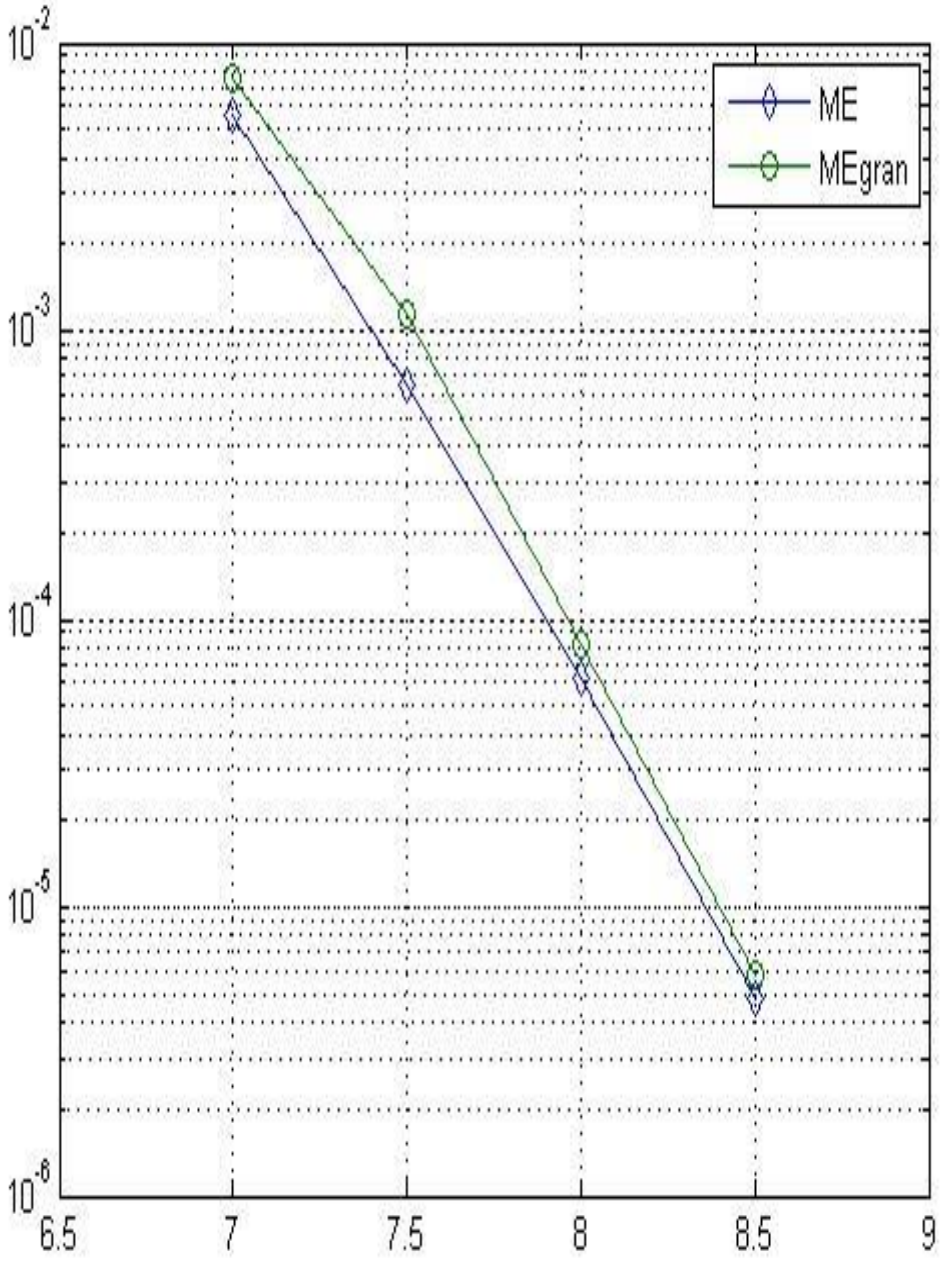


Figura 30 - Degradação causada pela quantização no cálculo da distância euclidiana. Fonte: Autor

4.2.5 A influência da variação do vetor β no desempenho do algoritmo de decodificação

O valor de β representa um valor médio da confiabilidade associada ao símbolo d_k toda vez que uma palavra competidora não é encontrada. Foram realizadas diversas simulações com intuito de verificar a influência dos valores associados ao vetor β nos resultados de desempenho da decodificação turbo geométrica.

Em [3], na parte que trata dos resultados de simulação, é observado que variações da ordem de $\pm 10\%$ no valor de β não produziram diferenças significativas nos resultados de desempenho observados

Diferentemente do caso binário onde $\beta \in \{0, \infty\}$, para o caso geométrico, no cálculo da informação extrínseca, β corresponde a um valor multiplicativo aplicado ao vetor de ruído associado ao passo de iteração. Desta forma $\beta \in \{0,1\}$, estando "1" associado a máxima confiabilidade ou seja $\Pr\{d_k = \hat{e}_k\} \to 1; \beta \to 1$, similarmente $\Pr\{d_k \neq \hat{e}_k\} \to 0.5; \beta \to 0$.

Duas abordagens foram utilizadas para definição dos valores de β utilizados nas simulações. Na primeira abordagem, conforme sugerido em [3], β foi obtido através do valor médio calculado a cada passo de iteração a partir dos valores da informação extrínseca obtida para os símbolos d_k para os quais foi encontrada palavra competidora. Desta forma, β representa o valor médio da confiabilidade dos símbolos para os quais a informação extrínseca foi calculada utilizando a eq.(52).

Na segunda abordagem foram fixados valores de β durante todo o processo de decodificação, sendo que os valores variaram dentro do interlavo ($0,1$).

Observou-se que também no caso geométrico, os resultados de desempenho se mostram pouco sensíveis a variação do valor de β, sendo ainda encontrados valores fixos de β que mostraram melhores resultados de desempenho, que os obtidos na primeira abordagem que exige um esforço de processamento maior devido a necessidade cálculo dos valores médios da informação extrínseca a cada passo de iteração.

Para as simulações realizadas, a abordagem utilizando valor fixo para β se mostrou como solução mais eficiente em termos de implementação para o mesmo desempenho, sendo necessário apenas o ajuste inicial deste parâmetro, o que pode ser conseguido facilmente em um ambiente real através das medidas de desempenho obtidas através da variação deste parâmetro.

A figura 31 mostra os resultados de desempenho para o esquema 16QAM-RS(15,13)[2] na situação em que β é obtido a cada passo de iteração através do cálculo do valor médio da informação extrínseca(Bmean), e para os valores fixo de β, iguais a: 0.05(B05), 0.15 (B15), 0.3 (B30) e 0.5 (B50).

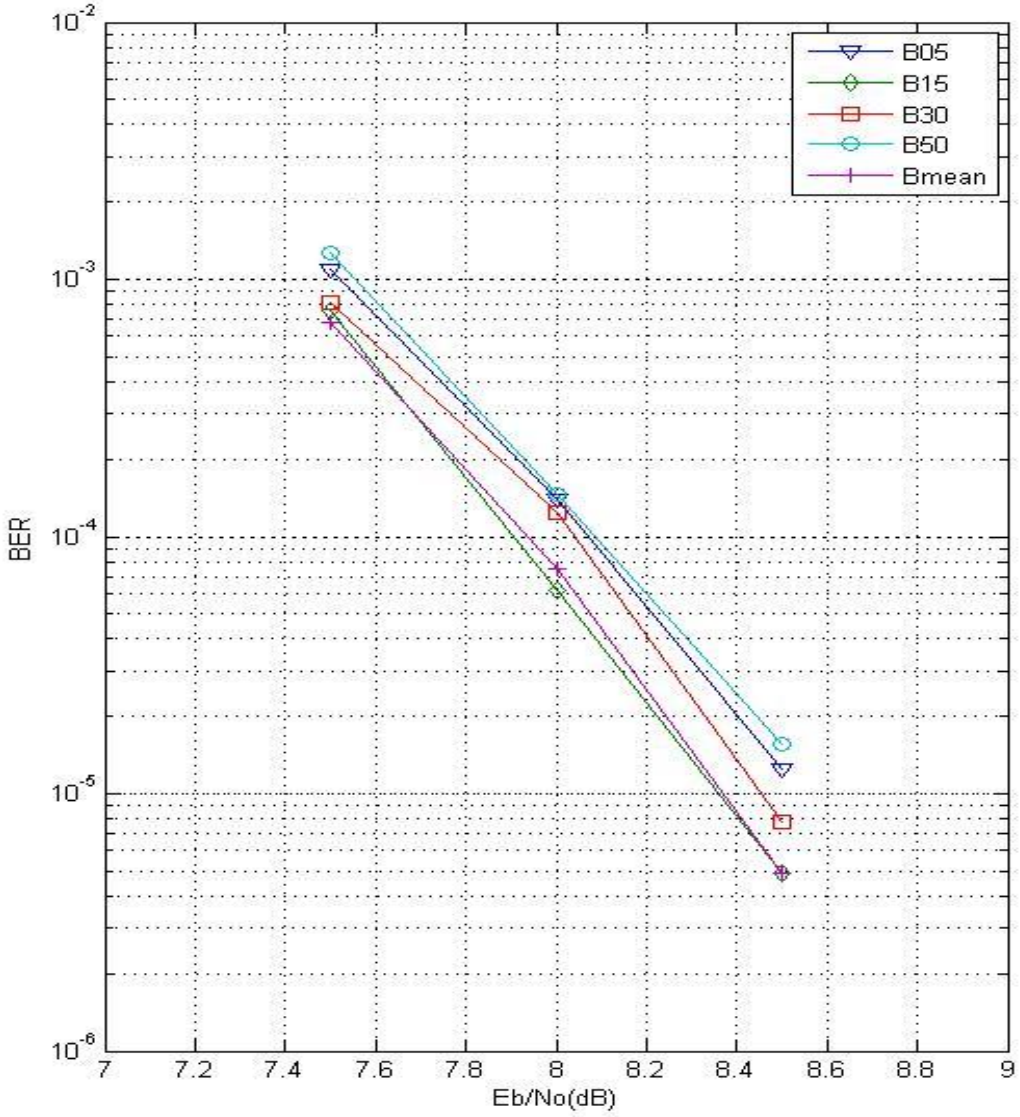


Figura 31 – Influência de β no desempenho do algoritmo de decodificação. Fonte: Autor

4.2.6 Considerações sobre a determinação de γ

Conforme discutido anteriormente na seção 3.3.2.1 a determinação do valor de γ, depende basicamente de dois fatores: O código utilizado, pois γ é função da distância de Hamming do código utilizado, e da cardinalidade do alfabeto utilizado, pois γ também é função da geometria do espaço eucliano utilizado.

O valor γ corresponderá ao maior evento de erro possível dentro do esquema de modulação e codificação utilizado.

Para códigos com d_{min} pequena e constelações de cardinalidade baixa, a obtenção de γ pode ser realizado facilmente de forma algébrica conforme mostrado na seção 3.3.2.1. Para esquemas mais complexo, no entanto, um maneira possível é através de um algoritmo de força bruta, obtendo-se o valor de γ a partir da geração estatística de sequências e acumulando o valor máximo do evento de erro observados. Uma aplicação prática desta estratégia, é através do próprio algoritmo de decodificação implementado. De maneira similar ao que foi sugerido anteriormente como estratégia para determinação do valor de β. Em um ambiente real é possível excitar a entrada do decodificador e observar os valores na saída, ou ainda obtê-los facilmente através do ambiente de simulação.

Os valores obtidos de maneira indireta, obviamente correspondem a ocorrências estatísticas, dependentes da relação sinal ruído, da distribuição de probabilidade e do comprimento das sequências enviadas, e a princípio não há garantia de ocorrência do maior evento de erro durante o ensaio realizado. Porém com grande probabilidade em termos práticos para um vetor de teste apropriado haverá ocorrência deste evento, fornecendo o valor apropriado para ser estabelecido o valor de γ a ser utilizado inicialmente.

O valor de γ utilizado, também não tem obrigatoriedade de constituir-se no valor exato do maior evento de erro, podendo em termos práticos estar ajustado de forma possuir valor próximo ao valor deste evento. Resultados de simulação demonstram que variações em torno de 5% do valor de γ não apresentaram variações significativas de desempenho nos resultados obtidos.

A figura 32, mostra os valores medidos dos eventos de erro observados no decodificador SISO de 1000 símbolos recebidos, obtidos para o esquema 16QAM,

RS(15,13)^2 e uma relação sinal ruído de entrada de 7.0dB. Os valores apresentados estão normalizados pela distância euclidiana do demodulador. O valor teórico máximo esperado para este esquema de modulação/codificação, conforme demonstrado anteriormente é $\left(\sqrt{29}-0{,}5\right)dE \sim 4{,}885dE$ .Note que através da observação dos máximos medidos é possível uma boa aproximação para o valor teórico.

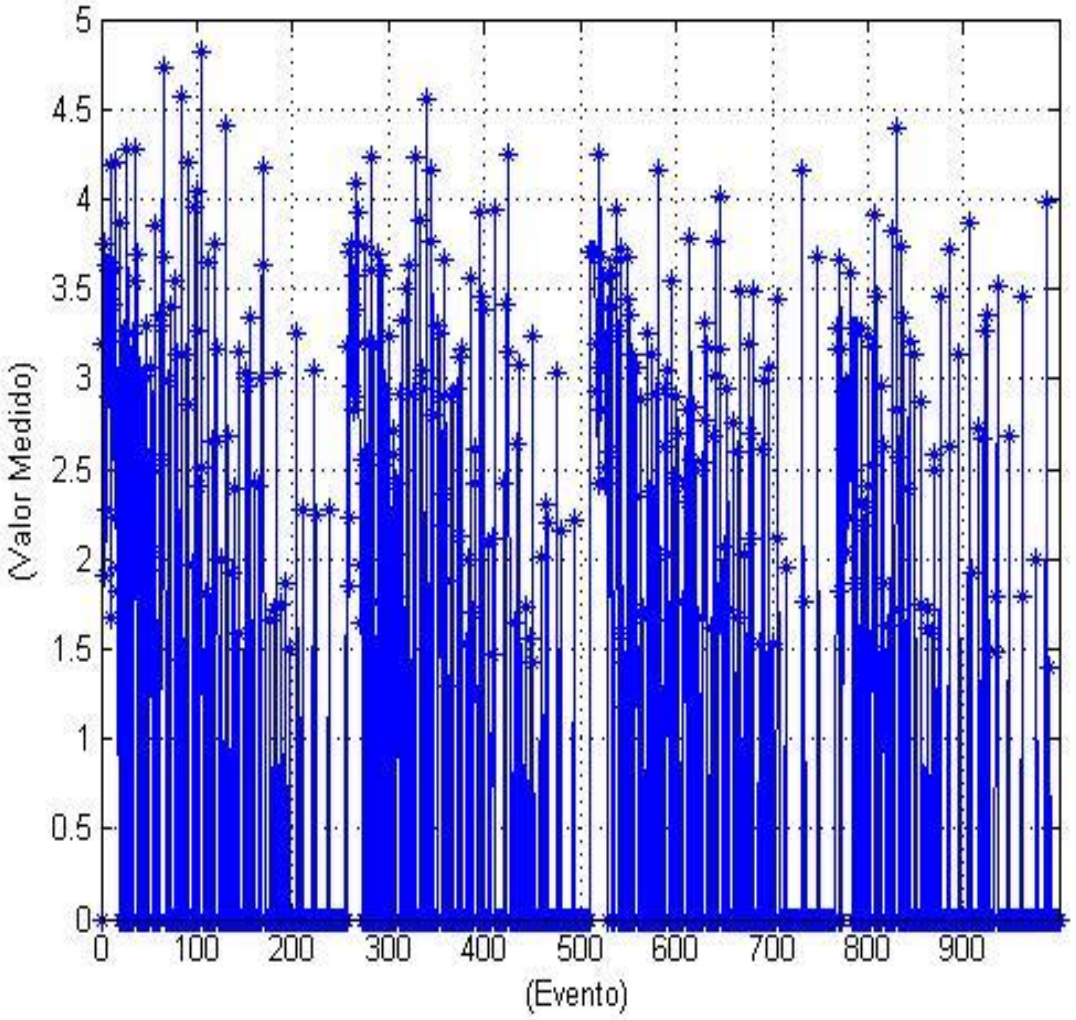


Figura 32 – Medida empírica dos eventos de erro para um esquema RS(15,13)^2 16QAM. Fonte: Autor

4.2.7 Considerações sobre o número de iterações

A figura 33 apresenta os ganhos de desempenho obtidos a cada passo de iteração para o esquema 16QAM-RS(15,13)^2. Observa-se que de maneira similar aos resultados obtidos em [3], não são observados ganhos significativos a partir do sexto passo de iteração.

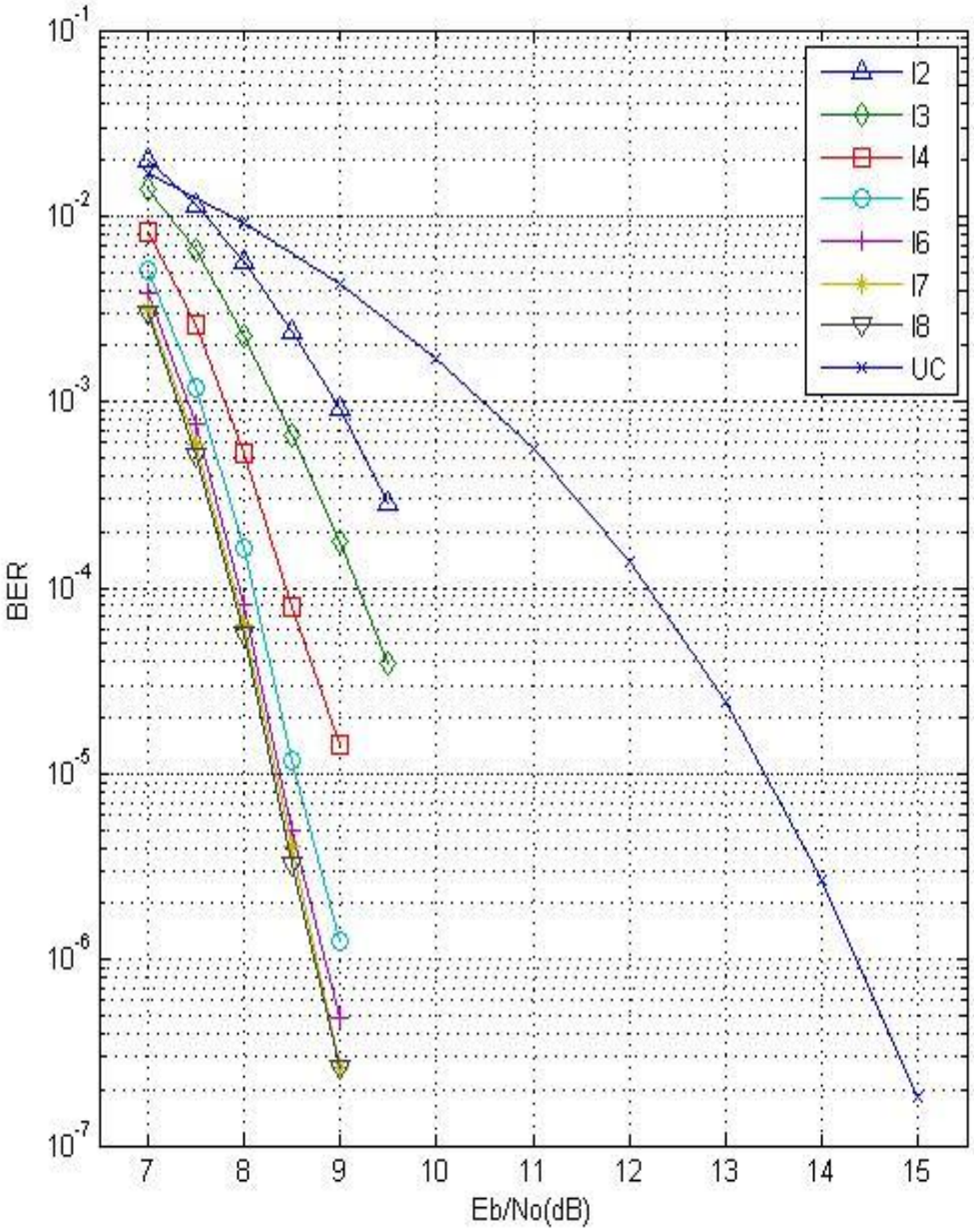


Figura 33 – Resultados de desempenho para diferente número de iterações. Fonte: Autor

4.3 CONSIDERAÇÕES SOBRE A IMPLEMENTAÇÃO DO ALGORITMO TURBO GEOMÉTRICO

Nesta seção é realizada uma análise dos aspectos de implementação do algoritmo proposto.

O algoritmo proposto, diferentemente da abordagem clássica binária, possui uma abordagem vetorial, o que irá trazer benefícios e também limitações em relação à primeira. Do ponto de vista de implementações em circuitos reais, no entanto a abordagem vetorial traz diversas vantagens em relação a abordagem binária.

Os sistemas reais QAM que são o objeto deste estudo, trabalham naturalmente com implementações na forma vetorial, sendo os sinais modulados e demodulados tratados de forma vetorial tanto no modulador como no demodulador. O sinal recebido do canal e o ruído associado, estão presentes na saída do demodulador na forma vetorial. Desta forma a informação pode ser processada de maneira direta quando a abordagem geométrica é utilizada, o que não ocorre quando a abordagem é binária, conforme observado em [27], onde algum mapeamento será necessário, gerando aumento de processamento além de possível perda de parte a informação neste processo.

Considerando a primeira parte da decodificação SISO utilizando o algoritmo II de Chase, tem-se uma redução do número de vetores de teste, e consequentemente da complexidade do decodificador para esta etapa que pode chegar a $2^{\log_2(L)-2}$, para um mesmo desempenho em relação ao a abordagem binária, conforme mostrado na seção 3.3.1.

Considerando a segunda etapa da decodificação SISO quando ocorre o cálculo da informação extrínseca, a abordagem pragmática binária requer operações sobre valores não binários para cada bit recebido. Desta forma o esforço computacional por elemento é praticamente equivalente se comparado ao algoritmo geométrico. No entanto o número de elementos é reduzido quando a abordagem geométrica é utilizada, uma vez que estes cálculos são aplicados aos símbolos de modulação ou elementos de campo formados pelas tuplas binárias.

Para um alfabeto de cardinalidade L, e com relação de multiplicidade de 1 para 1 em relação aos elementos de campo, a redução da complexidade é da ordem de

$\log_2(L-4)$, pois o cálculo da informação extrínseca é realizado para os $\log_2(L)xL^2$ bits da matriz produto, quando a abordagem pragmática binária é utilizada. No caso geométrico o cálculo da informação extrínseca é realizado sobre os L^2 símbolos recebidos. Na abordagem binária, no entanto o cálculo se dá em uma dimensão, enquanto que na abordagem geométrica cada elemento terá duas componentes. Tem-se em média portando o dobro de operações por elemento para o caso geométrico.

Do ponto de vista de implementação observa-se que o algoritmo geométrico se torna uma escolha tão mais vantajosa quanto maior a cardinalidade do alfabeto utilizado, sendo também a relação de multiplicidade outro fator de facilitação da implementação. Obviamente estas condições de contorno conduzem a algumas limitações quanto as definições de esquemas de codificação para os quais esta abordagem se torna interessante em relação ao caso binário.

CONCLUSÃO

Este trabalho mostrou uma nova abordagem para a decodificação turbo de códigos produto em sistemas de modulação QAM utilizando códigos de Reed-Solomon como códigos constituintes. A abordagem geométrica mostrar-se como uma boa alternativa tanto do ponto de vista de desempenho quanto de implementação para vários casos práticos, em especial quando há esquemas de modulação com cardinalidade mais elevada e para taxas de codificação próximas da unidade ($r \to 1$), o que corresponde a várias situações práticas onde se tem sistemas de banda limitada e necessidade de operação em tempo real.

Foi mostrado ainda que embora na abordagem geométrica a operação seja realizada para duas dimensões, ela conduz a outras simplificações que levam a uma redução da complexidade de implementação, se comparado ao método pragmático binário.

Do ponto de vista prático, foi mostrado que é possível a utilização de aproximações que reduzem significativamente a complexidade da implementação sem que isto cause degradação significativa de desempenho.

Investigações adicionais de ganhos possíveis através da abordagem geométrica estão em aberto, como por exemplo, critérios adicionais de formação dos vetores de teste na primeira etapa da decodificação SISO, critérios de parada associados a evolução dos vetores dentro do espaço euclidiano ou ainda parametrizações que possam proporcionar uma redução no tempo de convergência.

A métrica utilizada, apesar de apresentar resultados de desempenho superiores a métrica euclidiana quadrática, pode ainda não ser a melhor métrica para esta abordagem.

Estudos mais aprofundados em trabalhos futuros podem conduzir a ganhos adicionais tanto de implementação como de desempenho, além dos apresentados neste trabalho.

REFERÊNCIAS

[1] BERROU,C.; GLAVIEUX, A.; THITIMAJSHIMA, P. Near Shannon limit error-correcting coding and decoding; Turbo-Codes(1). In: **IEEE International Conference on Communications ICC'93,** v. 2/3, p. 1064-1070, May 1993.

[2] SHANNON,C.E. A mathematical theory of communication. **The Bell System Technical Journal**. v. 27, p. 379–423, 623–656, July, October, 1948.

[3] PYNDIAH, R M. Near-Optimum decoding of product codes: block turbo codes. **IEEE Transactions on Communication**, p. 1003-1010, v. 46, n. 8, August 1998.

[4] CHASE, D. A class of algorithms for decoding block codes with channel measurement information. **IEEE Transactions on Information Theory**, v. IT-18, n. 1, January 1972.

[5] DAVE, S. KIM, J.; KWATRA, S.C. An efficient decoding algorithm for block turbo codes. **IEEE Transactions on Communications**. v.49, n1, p.41-46, January 2001.

[6] DAVID, W. Turbo product code tutorial, Document Number: IEEE 802.16t-00/01. **Advanced Hardware Architectures AHA Presentation**, Pullman WA, May 2000.

[7] XILINX IP-IEEE 802.16-compatible turbo product code decoder v1.1 DS212. **Product Specification**, january, 2003.

[8] PYNDIAH, R, M; ADDE, P. ;JEGO, C. ; BIDAN, R.; LEROUX, C. Reed-Solomon turbo product codes for optical communications: From code optimization to decoder design. Hindawi Publishing Corporation EURASIP **Journal on Wireless Communications and Networking**, Article ID 658042, 2008.

[9] AITSAB, O.; PYNDIAH, R, M. Performance of Reed-Solomon block turbo code. In: **Global Telecommunications Conference**, v. 1, p. 121-125, November 1996.

[10] ZHOU, R.; BIDIAN, R.L.; PYNDIAH, R.M. GOALIC. Low-Complexity high-rate Reed–Solomon block turbo codes. **IEEE Transactions on Communication**, v. 55, n. 9, Sept 2007.

[11] DIATTA,D.; GEEST,D.; GELLER.B. Reed-Solomon turbo codes for high data rate transmission. In: **IEEE 59th Vehicular Technology Conference**, v. 2, p.1023-1027, May 2004.

[12] GOALIC, A.; TRUBUIL, J.; LAOT, C.; BEUZELIN, N. Improvement of underwater acoustic communication using Reed-Solomon block turbo codes. In: **6th International Symposium on Turbo Codes & Iterative Information Processing**, p151-155, 2010.

[13] CUNHA, C,D.; PORTUGHEIS, J; NASCIMENTO, V. Symbol-Wise Chase Decoding of Q-ary Block Codes over BI-AWGN. **IEEE Communications Letters**, v. 15, n. 2, February 2011.

[14] SPEIDEL,J.; RAN-HONG, Y.; BRINK, S. Iterative demapping and deconding for multilevel modulation, In: **IEEE Global Telecommunication Conference**, v. 1, p.579-584, 1998.

[15] GROSS, W.J.; KSCHISCHANG, F.R.; KOETTER, R.; GULAK, P.G. Applications of algebraic soft-decision decoding of Reed–Solomon codes, **IEEE Transactions on Communications**, v. 54, n. 7, July 2006

[16] SARWATE, A. Soft decision decoding of Reed-Solomon product codes. **Final Project EECS 229B**, UC Berkeley, May 2005

[17] KOETTER, R.; VARDY, A . Algebraic soft-decision decoding of Reed-Solomon codes. **IEEE Transactions on Information Theory**, v.49, n.11, p. 2809-2825, 2003.

[18] EL-KHAMY, M.; McELIECE,R.J. Iterative algebraic soft-decision list decoding of Reed-Solomon codes. **IEEE Journal on Selected Areas in Communications**, v.24, n.3, p.481-490, 2006.

[19] CHANG, H.C.; SHUNG, C.B.; LEE, C.Y. A Reed–Solomon product-code (RS-PC), decoder chip for DVD applications. **IEEE Journal of Solid-State Circuits**. v. 36, n. 2, February.

[20] LIU, C.Y.; LIN, S. Encoding and decoding of Reed–Solomon,Codes through binary decomposition and self-concatenation. **IEEE Transactions on Communications**, v. 52, n. 9, September 2004.

[21] PYNDIAH, R,M.; ADDE, P.; JEGO, C.; BIDAN, R. Some results on the binary minimum distance of Reed-Solomon codes and block turbo code. In: **IEEE proceedings ICC**, 2007.

[22] LEE, S.W. Soft-decision decoding of Reed-Solomon codes using pattern information over partial response channels. **Ph.D. dissertation**, Carnegie Mellon Univ., Pittsburgh, PA, 2009.

[23] Lee, S.W.; KUMAR, B. V. Pattern-flipping chase-type decoders with error pattern extracting Viterbi algorithm over partial response channels. **IEEE Journal on Selected Areas in Communications**, v. 28, n. 2, p.181-187, February. 2010.

[24] GELLER,B.; DIATTA, J.P.; BARTOT,C. ;VANSTRACEELET,C.; RAMBEAU, S. Block Turbo Codes: From Architecture to Application. In: **ISIT 2006**, Seattle, USA, July 2006

[25] CHANG,Y.; WEI, V,K.; Q-ary turbo codes with QAM modulations. In: **IEEE International Conference on Universal Personal Communications**, v.2, p. 814-817, Oct 1996.

[26] SEONG, R.K.; FAYZIYEV, A.; CHA, J.; HÁ, J. A New Efficient 16-QAM mapping approach for Iterative receiver using Turbo codes over SISO Channel. **10th International Conference on Advanced Communication Technology**,v.1, p. 421-423, February 2008.

[27] PYNDIAH, R,M.; PICART, A.; GLAVIEUX, A. Performance of block turbo coded 16-QAM and 64-QAM modulations. In: **Global Telecommunications Conference**, v.2, p.1039-1043, 1995

[28] LATHI, B.P. An introduction to random signals and communication theory. Scranton: International Textbook Company, 1968.

[29] HAYKIN, S. Digital communications.NewYork: Wiley, 1988.

[30] LIN, S. COSTELLO, D.J. Error Control Coding fundamentals and applications, New Jersey: Printice-Hall,1983.

[31] MACWILLIAMS, F.J. SLOANE, N.J.A. The theory of error correcting codes. Amsterdam: North-Holland Mathematical Library, p.567-580, 1978.

[32] ELIAS, P. Error-free coding. **IRE Professional Group on Information Theory**. v4, n.4, p. 29-37, September 1954.

[33] BENEDETTO, S. Turbo Codes: Performance analysis, design, iterative decoding and applications – **IEEE COMMUNICATIONS –Tutorials Now.**

[34] BENEDETTO, S.; MONTORSI G. Unveiling turbo codes: some results on parallel concatenated coding schemes. **IEEE Transactions on Information Theory**, September 1995.

[35] AHA. PS4501: Astro 36 Mbits/s Turbo Product Code Encoder/Decoder. Product Specification, Pullman, WA.

[36] FANG, J.; BUDA,F.; LEMOIS, E. Turbo product codes: A well suitable solution to wireless packet transmission for very low error rates. In: **International Symposium on Turbo Codes and Related Topics**, p. 101-111,Brest, France September 2000.

[37] PYNDIAH. R. Iterative decoding of product codes: Block turbo codes, In: **International Symposium on Turbo Codes and Related Topics**. v. 1/1, p. 71-79, September 1997.

[38] BERROU, C.; GLAVIEUX, A C. Near optimum error correcting coding and decoding: Turbo-codes. **IEEE Transaction on Communications**, v.44, p. 1261–1271,
October 1996.

[39] RUNGE, C.J.R. PIQUEIRA, J.R.C. A geometric approach for turbo decoding of Reed-Solomon codes in QAM modulation schemes, **IEEE Communications Letters**, v. 13, n. 5, May 2009

[40] VUCETIC, B.; PONAMPALAM,V.; VUCKOVIC, J. Low complexity soft decision decoding algorithms for Reed-Solomon codes. **IEICE Transaction on Communications** , v. E84-B, n. 3, p. 392-399, March 2001.

[41] CHRISTIAN, S. On error bound and Turbo-codes. **IEEE Communications Letters**, v.3, n. 7, July 1999.

APÊNDICE

APÊNDICE – TEORIA DE INFORMAÇÃO

O texto a seguir foi elaborado a partir do material apresentado em [29] , [33] e [34], bem como as figuras apresentadas. Tem como objetivo, complementar as ideias e conceitos apresentados nesse trabalho.

1 Quantidade de Informação, Entropia e o Limite de Capacidade de Canal.

A quantidade de informação contida em um evento, esta associada ao grau de incerteza quanto à ocorrência desse evento.

Seja $s_k; k \in \{1,..N\}$ um símbolo pertencente ao alfabeto de símbolos de uma fonte de informação, com probabilidade de ocorrência p_k. A emissão do símbolo s_k, é descrita pela ocorrência do evento $S = s_k$.

Suponha que p_k=1, ou seja s_k é o evento "certo". Nesta situação não existe informação associada à ocorrência de s_k, sendo que a quantidade de informação nula, uma vez que já se sabe previamente qual informação a fonte vai emitir.

Matematicamente, a quantidade de informação associada a observação de um evento $S = s_k$, que ocorre com uma probabilidade p_k, é dada pela eq.(1).

$$I(s_k) = \log_2\left(\frac{1}{p_k}\right) \quad (1)$$

Através desta definição, alguns resultados interessantes são observados. Por exemplo, se p_k = 1, então $I(s_k) = 0$, com seria de se esperar. Se $p_k = 1/2$, então a quantidade de informação associada é $I(s_k) = 1$ bit. Ou seja, um é a quantidade de informação associada à ocorrência de um entre dois eventos equiprováveis.

2 Entropia

A quantidade de informação produzida por uma fonte de informação em um intervalo arbitrário de tempo dependerá dos símbolos emitidos no intervalo.

Suponha uma fonte discreta que emite símbolos estatisticamente independentes através de um alfabeto Ψ. Nesta situação $I(s_k)$, é uma variável aleatória discreta, que assume os valores $I(s_0)$, $I(s_1)$,... $I(s_{k\text{-}1})$, com probabilidades p_0, p_1,... $p_{k\text{-}1}$.

O valor médio de $I(s_k)$, sobre o alfabeto Ψ, da fonte é definido como a entropia da fonte, $H(\Psi)$. Desta forma, a entropia é uma medida da quantidade média de informação que cada símbolo da fonte carrega.

A entropia pode ser escrita como:

$$H(\Psi) = E\left[I(s_l)\right] \quad (2)$$

$$H(\Psi) = \sum_{l=0}^{k\text{-}1} p_l I(s_l) \quad (3)$$

Pela definição da entropia, é possível mostrar que a mesma tem o seu valor limitado inferior e superiormente pelos valores.

$0 \leq H(\Psi) \leq \log_2(K)$ (4); onde K é a cardinalidade do alfabeto da fonte.

Seja por exemplo uma fonte binária, que possui os símbolos 0 e 1, ocorrendo com probabilidades p_0 e $1\text{-}p_0$ respectivamente. Neste caso a entropia pode ser escrita como:

$$H(\Psi) = -p_0 \log_2(p_0) - (1 - p_0)\log_2(1 - p_0). \quad (5)$$

3 O Teorema de Codificação de Fonte

São questões de grande importância no projeto e concepção de um sistema digital de comunicação, a eficiência com que a informação de uma dada fonte pode ser representada e a taxa com a qual é possível se estabelecer uma transmissão eficiente através de um canal ruidoso.

A função do codificador de fonte é mapear as informações emitidas por uma dada fonte em um alfabeto que constitua uma representação eficiente dessa informação.

O princípio da codificação de fonte é, a partir da estatística associadas aos diversos símbolos emitidos pela fonte, obter uma representação em um alfabeto mais eficiente. Os símbolos que possuem grande probabilidade de ocorrência são representados mais eficientemente através de palavras codificadas mais curtas, já símbolos que possuem baixa probabilidade de ocorrência podem ser representados por palavras codificadas mais longas. Desta forma as palavras codificadas pelo codificador de fonte tem seu comprimento associado à probabilidade da sua ocorrência.

Suponha uma fonte discreta que produza K símbolos, e que o que i-ésimo símbolo s_i possua probabilidade de ocorrência p_i.

Assumindo que o codificador mapeia cada símbolo da fonte em uma palavra binária correspondente, e que o comprimento de uma palavra codificada associada a um símbolo s_i é l_i.

Definindo o comprimento médio das palavras codificadas $\overline{L}$ através da expressão:

$$\overline{L} = \sum_{l=0}^{k\text{-}1} p_i l_i \quad (6)$$

Seja $L_{\min}$ o menor valor possível de l. Então se define como eficiência da fonte como:

$$\eta = \frac{L_{\min}}{\overline{L}} \quad (7)$$

Como $\overline{L} \geq L_{\min}$, então $\eta \leq 1$. Ou seja a máxima eficiência seria atingida na situação onde se tem $\overline{L} = L_{\min}$.

O primeiro teorema de Shannon estabelece que dado uma fonte discreta e sem memória. O comprimento médio das palavras código do codificador de fonte, é limitado inferiormente pela entropia da fonte. Ou seja:

$$\overline{L} \geq H(\Psi) \quad (8)$$

Substituindo $L_{\min}$, pela entropia $H(\Psi)$ na eq.(7), tem-se:

$$\eta = \frac{H(\Psi)}{\overline{L}} \qquad (9)$$

Concluí-se que a entropia representa o limite para a representação eficiente da informação da fonte. Ou seja, as palavras código podem ter seu valor médio feito tão curto quanto o valor da entropia da fonte, mas não mais.

4 O Canal Discreto sem Memória

A partir deste ponto são investigadas formas eficientes de transmissão da informação em um meio ruidoso. Para tanto, será apresentado o modelamento do canal discreto sem memória.

É denominado canal discreto sem memória, ao modelo estatístico que define uma conjunto de entradas x pertencentes a um alfabeto X, e de saídas y pertencentes a um alfabeto Y, e que constituem a versão ruidosa das entradas x.

A cada instante de tempo, o canal aceita uma entrada x e emite uma saída y correspondente.

O canal é dito discreto devido aos alfabetos terem tamanhos finitos, e sem memória devido ao fato de que uma cada saída apresenta uma relação de dependência

apenas com a correspondente entrada, sendo a estatística associada descrita através de uma probabilidade de transição $p(y_k \mid x_j) = p(Y = y_k \mid X = x_j)$.

A figura abaixo é uma representação do canal:

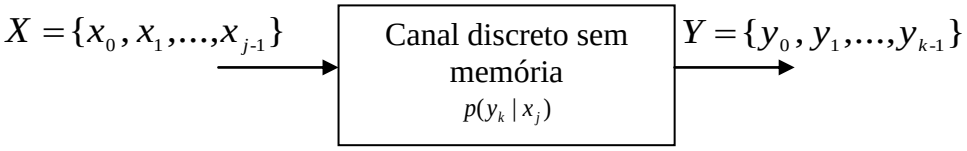


Figura 1 – Canal discreto sem memória. Fonte: Autor

5 Entropia Condicional e a Informação Mútua

Define-se como entropia condicional de um alfabeto de entrada X em relação a saída y_k, a medida da incerteza sobre este alfabeto, dado que foi observada a saída y_k. A entropia condicional é obtida a partir das probabilidades condicionais associada a todo o alfabeto X em relação ao valor observado y_k. Utilizando a eq.(3), aplicada às probabilidades condicionais correspondentes, tem-se:

$$H(X \mid Y = y_k) = \sum_{j=0}^{J\text{-}1} p(x_j \mid y_k) \log_2 \left(\frac{1}{p(x_j \mid y_k)} \right) \quad (10)$$

De maneira similar. Seja $Y = \{y_0, y_1, \ldots, y_{K\text{-}1}\}$, e supondo que as saídas y_k, ocorrem com probabilidade $p(y_0), p(y_1), \ldots p(y_{K\text{-}1})$. Então a incerteza da entrada X, dado que a saída Y do canal é observada, é dada por:

$$H(X \mid Y) = \sum_{k=0}^{K\text{-}1} H(X \mid Y = y_k) p(y_k) \quad (11)$$

E reescrevendo em função das probabilidades associadas a cada um dos símbolos dos alfabetos:

$$H(X \mid Y) = \sum_{k=0}^{K\text{-}1} \sum_{j=0}^{J\text{-}1} p(y_k)\, p(x_j \mid y_k) \log_2 \left(\frac{1}{p(x_j \mid y_k)} \right) \quad (12)$$

Como $H(X)$ representa a incerteza sobre a entrada, e $H(X \mid Y)$ representa a incerteza sobre a entrada depois de observada a saída. Então a diferença $H(X)$ - $H(X \mid Y)$, representa a quantidade de informação, ou a diminuição da incerteza após observar a saída do canal. A informação mútua é definida como a diferença entre a entropia da fonte e a entropia condicional associada ao canal:

$$I(X;Y) = H(X) - H(X \mid Y) \quad (13)$$

6 O Limite de Capacidade de Canal

Através da propriedade da probabilidade conjunta, é sabido que:

$$p(x \mid y)\, p(y) = p(x; y) = p(y \mid x)\, p(x) \quad (14)$$

Desta forma, a informação mútua pode ser escrita conforme abaixo:

$$I(X;Y) = \sum_{j=0}^{J\text{-}1} \sum_{k=0}^{K\text{-}1} p(y_k \mid x_j)\, p(x_j) \log_2 \left(\frac{p(y_k \mid x_j)}{p(y_k)} \right) \quad (15)$$

Observa-se através de (15), que a informação mutua não depende apenas da estatística do canal, mas também da distribuição de probabilidade da entrada $\{p(x_j),\, j = 0,1,...J\text{-}1\}$. Ou seja, a informação mútua pode ser maximizada em função da distribuição de probabilidade da entrada.

Define-se como capacidade de canal, como o máximo valor de $I(X;Y)$, em função de todas as possíveis distribuições de entrada.

$$C = \max_{\{p(x_j)\}} I(X;Y) \quad (16)$$

A quantidade C, é medida em bits por uso do canal, para uma canal discreto.
Para ilustrar este conceito, seja o canal binário simétrico com probabilidade de transição p. E considerando a máxima entropia da fonte $H(X) = 1$, utilizando (12) e (11), a capacidade de canal pode ser escrita como:

$$C = 1 + p\log_2(p) + (1 - p)\log_2(1 - p) \quad (17)$$

Através da expressão acima, fazendo a probabilidade de transição tender a zero, situação em que se tem o canal sem ruído, a capacidade de canal tende a 1, o que corresponde a própria informação contida na entrada. Por outro lado, se a probabilidade tender a 0.5, situação onde ocorre a máxima quantidade de ruído, a capacidade de canal tende a zero.

7 O Teorema da Codificação de Canal

Seja uma fonte com entropia $H(X)$ bits/símbolo, que emite símbolos uma vez a cada T_s segundos. Seja um canal com capacidade C bits/uso do canal, utilizado uma vez a cada T_c segundos. A capacidade de canal por unidade de tempo é dada então por C/T_c.

Se:

$$\frac{H(X)}{T_s} \leq \frac{C}{T_c} \quad (18)$$

Então existe um esquema de codificação para o qual a saída da fonte pode ser codificada e transmitida através do canal, com uma probabilidade de erro p_e, tão pequena quanto se queira. A igualdade na expressão acima, representa chamada taxa crítica, e se $\frac{H(X)}{T_s} > \frac{C}{T_c}$, então não é possível transmitir com uma p_e arbitrariamente baixa.

Seja como exemplo o canal binário simétrico, e seja a fonte binária emitindo símbolos com probabilidade $p_{0,1} = 1/2$, então $H(X) = 1$ bit/símbolo.

Suponha que os símbolos são codificados para transmissão com uma taxa $r = k/n$.

O codificador produz um símbolo a cada T_c segundos, e a fonte um símbolo a cada T_s segundos. Então:

$$\frac{1}{T_s} \leq \frac{C}{T_c}; \quad (19)$$

mas

$$\frac{T_c}{T_s} = r \quad (20)$$

Então se $r \leq C$, existe um esquema de codificação capaz de fazer p_e arbitrariamente baixa.

Suponha por exemplo um canal binário simétrico com probabilidade de transição $p = 10^{-2}$, usando eq.(17) tem-se que $C = 0.9192$. Desta forma, existe um código com taxa $r = 0.9192$, tal que a p_e pode ser feita tão pequena quanto se queira.

8 O Teorema da Capacidade de Canal

Estendendo o conceito de entropia para o caso onde a variável aleatória é contínua.

Seja X uma variável aleatória continua, com função densidade de probabilidade $f(x)$.

Define-se com entropia diferencial a quantidade

$$h(x) = \int_{-\infty}^{\infty} f_x(x)\left(\frac{1}{f_x(x)}\right)dx \quad (21)$$

Para um valor fixo de variância σ^2, a função densidade de probabilidade que maximiza $h(x)$, é dada pela expressão:

$$f_x(x) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right) \quad (22)$$

E substituindo a eq.(22) em eq.(21):

$$h_{\max}(x) = \frac{1}{2}\log_2(2\pi e\sigma^2) \quad (23)$$

Através da eq.(22), se observa que variável aleatória que apresenta maior entropia diferencial é a que possui função densidade de probabilidade gaussiana. Verifica-se ainda que $h(x)$ é totalmente determinada pela variância σ^2.

Suponha a transmissão através de um canal limitado em banda em B Hertz. Um processo aleatório $X(t)$ limitado nesta banda pode ser totalmente reconstruído se o mesmo for amostrado a uma taxa igual ou superior a 2B amostras por segundo.

Considerando a transmissão de um sinal $X(t)$ através de um canal, que é perturbado por um ruído gaussiano aditivo de variância $\sigma^2 = No/2$.

Se $Y_k = X_k + N_k$; k = 1,2,...,2B , as amostras na saída do canal amostradas a uma taxa 2B. Então as amostras N_k terão variância dada por $\sigma^2 = NoB$. Supondo ainda que o sinal transmitido possui potência limitada com valor médio P, ou seja, $E[X_k^2] = P$. Então a capacidade de canal pode ser definida como:

$$C = \max_{f_{X_k}(x)}\{I(X_k;Y_k) : E[X_k^2] = P\} \quad (24)$$

tem-se que:

$$I(X_k;Y_k) = h(Y_k) - h(Y_k / X_k) \quad (25)$$

Como X_k e N_k, são variáveis aleatórias independentes e $Y_k = X_k + N_k$, então $h(Y_k / X_k) = h(N_k)$. E a maximização de $I(X_k;Y_k)$, recai na maximização de $h(Y_k)$, que conforme mostrado em (23), é máxima se a mesma for gaussiana, o que implica em X_k ser gaussiano também.

Reescrevendo:

$$C = I(X_k;Y_k) : X_k Gaussiano \,;\, E[X_k{}^2] = P \quad (26)$$

como :

$$Y_k = X_k + N_k \Rightarrow \sigma_Y^2 = P + \sigma_N^2 \quad (27)$$

então:

$$h(Y_k) = \frac{1}{2}\log_2[2\pi e(\sigma_N^2 + P)] \quad (28)$$

$$h(N_k) = \frac{1}{2}\log_2(2\pi e\sigma_N^2) \quad (29)$$

então:

$$C = h(Y_k) - h(N_k) \quad (30)$$

$$C = \frac{1}{2}\log_2\left(1 + \frac{P}{\sigma_N^2}\right) \text{ bits/uso de canal (31)}$$

$$C = B\log_2\left(1 + \frac{P}{NoB}\right) \text{ bits/segundo (32)}$$

Ou seja, C é a capacidade de um canal limitado em B Hertz e perturbado por um ruído gaussiano de densidade espectral $No/2$.

9 O Limite Inferior para a Probabilidade de Erro de Palavra Codificada

Dentre os vários resultados oriundos dos trabalhos apresentados por Shannon na sua formulação da Teoria da Informação, está o limite inferior para a probabilidade de erro de palavra associado a um esquema codificado de taxa r , onde:

$$0 \leq r \leq C \quad (33)$$

E C é o limite de capacidade de canal.

Este resultado estabelece que a probabilidade de erro de palavra codificada $P_w(e)$, é limitada inferiormente pela expressão:

$$P_w(e) < 2^{-n\mathrm{E}(r)} \quad (34)$$

Onde n é o comprimento da palavra codificada, e $\mathrm{E}(r)$ é uma função convexa, decrescente positiva de r, e associada a capacidade de canal C . $\mathrm{E}(r)$ é também conhecida como expoente de Gallager, [41].

Note que a probabilidade de erro de bit $Pb(e)$ é limitada superiormente por $P_w(e)$. Ou seja:

$$P_b(e) \leq P_w(e) \quad (35)$$

Então o resultado acima, também é valido para probabilidade de erro de bit.

A partir da expressão (34) é possível observar que a fim de se minimizar a probabilidade de erro de um dado esquema codificado, três abordagens são possíveis:

1- Com $\mathrm{E}(r)$, é uma função decrescente de r, a fim de minimizar $Pb(e)$, pode-se diminuir r, o que acarretaria em no aumento da banda necessária para a transmissão da informação.

2 – Como $0 \leq r \leq C$, então uma segunda abordagem é aumentar a capacidade de canal, provocando um aumento nos valores da função $E(r)$, para um mesmo valor de r. Pela eq.(32), observa-se que tal abordagem é possível, através do aumento de P, que é a potência transmitida.

Estas duas abordagens, são soluções clássicas e já conhecidas de sistemas de comunicação. Ou seja, sempre é possível melhorar o desempenho do sistema ao custo de uma aumento da banda utilizada, ou através do aumento da potência do sinal enviado no canal.

O que é revolucionário nos resultados apresentados na teoria formulada por Shannon se refere ao termo associado a n. Através da análise da eq.(34) observa-se que uma terceira estratégia é possível. É possível diminuir a probabilidade de erro através do aumento do comprimento das palavras código.

A eq.(34) foi deduzida através de um teorema existencial. O que a ela mostra, é que se mantendo a mesma potência, e a mesma taxa de codificação, porém aumentando o comprimento das palavras codificadas, existem esquemas de codificação onde é possível fazer a probabilidade de erro de palavra, tão pequena quanto se queira.

De fato, se forem utilizadas sequências muito longas escolhidas aleatoriamente como palavras código e utilizado um sistema de decodificação de máxima verossimilhança (ML), é possível criar bons códigos com uma alta probabilidade de sucesso.

O problema, no entanto reside no fato de que a complexidade de decodificação é função do número de palavras código, que por sua fez são função do comprimento das palavras código, pois para uma da taxa de codificação r e um comprimento de palavra n, o alfabeto do código correspondente terá 2^{nr} palavras código.

O que este resultado mostra é que, é possível fazer $Pb(e)$, tão pequena quanto se queira ao custo de um aumento na complexidade de decodificação, que cresce exponencialmente com o comprimento das palavras código. Assim, a aplicação direta da terceira abordagem do teorema é inviável na prática devido à complexidade de decodificação associada.

Reescrevendo as equações relacionadas a capacidade de canal em termos dos parâmetros relacionados a taxa de codificação r, ao tipo de modulação utilizado e em termos da energia gasta para o envio de cada bit de informação, é possível a obtenção de alguns resultados importantes.

Para a expressão do limite de capacidade de canal para um canal gaussiano AWGN, tem-se:

$$C = \frac{1}{2}\log_2\left(1 + \frac{P}{N_0 B}\right) \text{ bits/uso do canal (símbolo de modulação)} \quad (36)$$

Sendo :

R_s Taxa de bits de informação por segundo

E_b Energia de bit de informação

R_m Taxa de símbolos de modulação por segundo

N Dimensão do espaço euclidiano utilizado.

então:

$$B = NR_m/2 \quad (37)$$

$$P = R_s E_b . \quad (38)$$

e:

$$C = \frac{1}{2}\log_2\left(1 + \frac{2rE_b}{N_0 N}\right) \quad (39)$$

Fazendo $r \rightarrow C$, uma vez que o teorema é valido para $0 \leq r \leq C$, tem-se :

$$r = \frac{1}{2}\log_2\left(1 + \frac{2rE_b}{N_0 N}\right) \quad (40).$$

Isolando o a relação sinal ruído na expressão acima:

$$\frac{E_b}{N_0} = 10\log_{10}\left(N\frac{(2^{2r}\text{ -1})}{2\text{r}} \right) \quad (41)$$

A expressão acima representa a menor relação sinal ruído possível para uma dada taxa de codificação, para qual é possível a transmissão com $P_b(e) \to 0$.

No caso de transmissão em um espaço unidimensional por exemplo, e fazendo-se $r \to 0$, ou seja na situação de máximo excesso de banda possível, a relação sinal ruído possível, de forma a haver transmissão eficiente ($P_b(e) \to 0$) é de -1.59dB. Ou seja, nenhum sistema de comunicação operando em um espaço de dimensão 1 é capaz de transmitir informação de forma eficiente, com uma relação sinal ruído inferior a -1.59dB.

O resultado acima, é valido para um canal gaussiano sem limitação nos símbolos do alfabeto de entrada, nem nos símbolos do alfabeto de saída.

Se o alfabeto de entrada for limitado ao caso binário, é possível mostrar que e a capacidade de canal passa a ser dada pela expressão:

$$C_{2I} = \frac{2rE_b}{N_0} - \frac{1}{\sqrt{2\pi}} \int e^{-y^2/2} \log\left[\cosh\left(\frac{2rE_b}{NN_0} + y\sqrt{\frac{2rE_b}{NN_0}} \right) \right] dy \quad (42)$$

Se tanto a entrada como a saída forem limitadas a um alfabeto binário, o que corresponderia ao modelo de canal AWGN de entrada binária e saída binária com decisão abrupta (*hard decision*). Então é possível mostrar que a capacidade de canal passa a ser dada pela expressão:

$$C_2 = 1 + p\log_2 p + (1 \quad p)\log_2(1 \quad p) \quad (43)$$

sendo:

$$p = \frac{1}{2} erfc\left(\sqrt{\frac{rE_b}{N_0}} \right) \quad (44\ )$$

Utilizando-se as equações 41, 42 e 43, e fazendo $r = C$, o que corresponde a situação com máxima utilização eficiente do canal, e $P_b(e) \rightarrow 0$. O gráfico abaixo estabelece a relação entre a taxa de codificação r , e a relação sinal ruído para os três tipos de canal descritos.

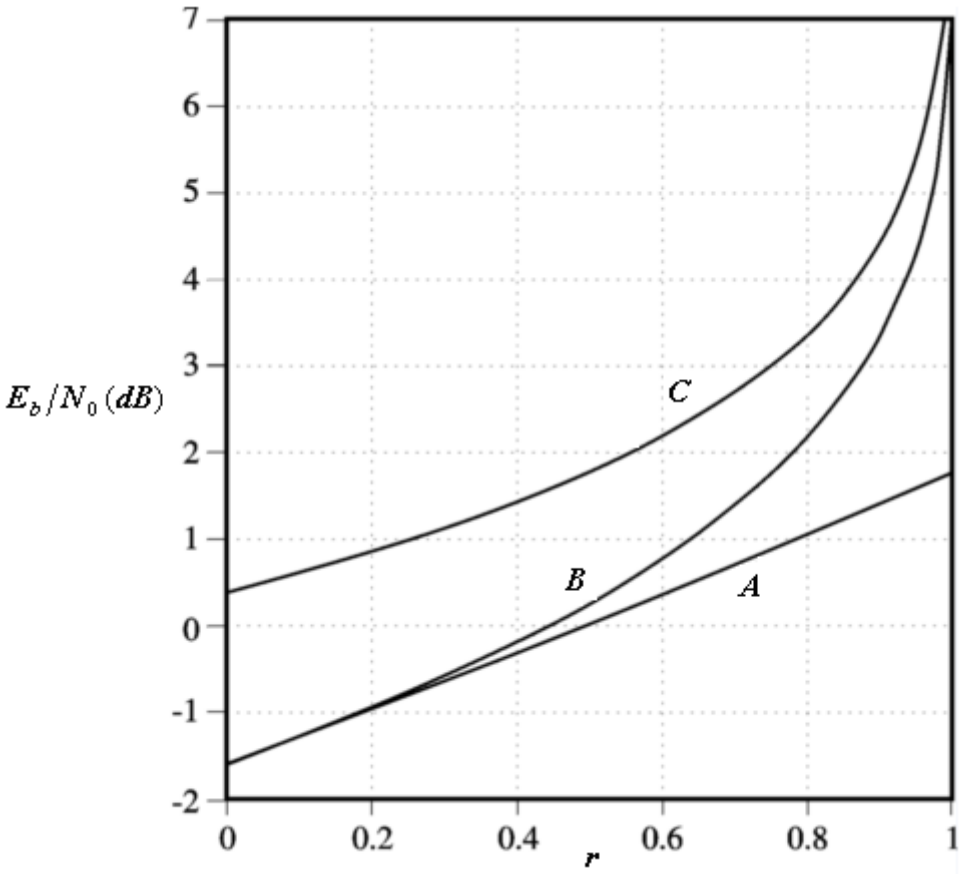


Figura 2 – Capacidade de canal para diferentes restrições de entrada/saída

No gráfico acima, a curva “A”,se refere ao canal gaussiano sem restrição de entrada ou saída, a curva “B”, se refere ao canal cujo alfabeto de modulação na entrada é limitado ao caso binário, e ao curva “C” se refere ao caso em que a entrada e a saída são limitadas ao caso binário.

É possível extrair algumas conclusões interessantes a partir da figura 2. Primeiramente comparando-se as curvas “A” e “B”, observa-se que para o caso de baixas relações sinal ruído, não existe diferença de desempenho entre os dois casos, o que mostra que nesta situação onde o nível de ruído é elevado, um aumento da cardinalidade do canal não traria benefícios devido a dificuldade do demodulador em distinguir um alfabeto de mais alta cardinalidade. Já à medida que

se caminha na direção de relações sinal ruído mais elevadas, são observados os benéficos na utilização de uma cardinalidade de ordem mais elevada.

N curva “C”, tem-se a situação onde a saída é binária, que na prática, como dito anteriormente, corresponde a situação de “*hard decison*” onde o valor analógico presente na saída do filtro casado no demodulador não é utilizado de forma quantitativa, sendo simplesmente quantizado em um bit ‘0’ ou ‘1’ correspondendo a um valor de decisão entre dois símbolos possíveis. Neste caso, para baixas relações sinal ruído há um perda da ordem de 2dBs quando esta abordagem é utilizada.

Se a eq.(41) for reescrita em termos da eficiência espectral, que representa a quantidades de bits por segundo por Hertz (bits/s/Hz), que o sistema é capaz de transportar.

Denotando a eficiência espectral por ε_b:

$$\varepsilon_b = \frac{R_s}{B} \quad (45)$$

e como,

$$B = \frac{NR_m}{2} \quad (46)$$

$$\varepsilon_b = \frac{2r}{N} \quad (47)$$

finalmente:

$$\frac{E_b}{N_0} = \frac{2^{\varepsilon_b} - 1}{\varepsilon_b} \quad (48)$$

Esta expressão nos permite determinar a menor relação sinal ruído possível para uma dada eficiência espectral desejada.

A Figura 3 mostra esta relação para o caso em que a dimensão do esquema de modulação é 2 (N=2).

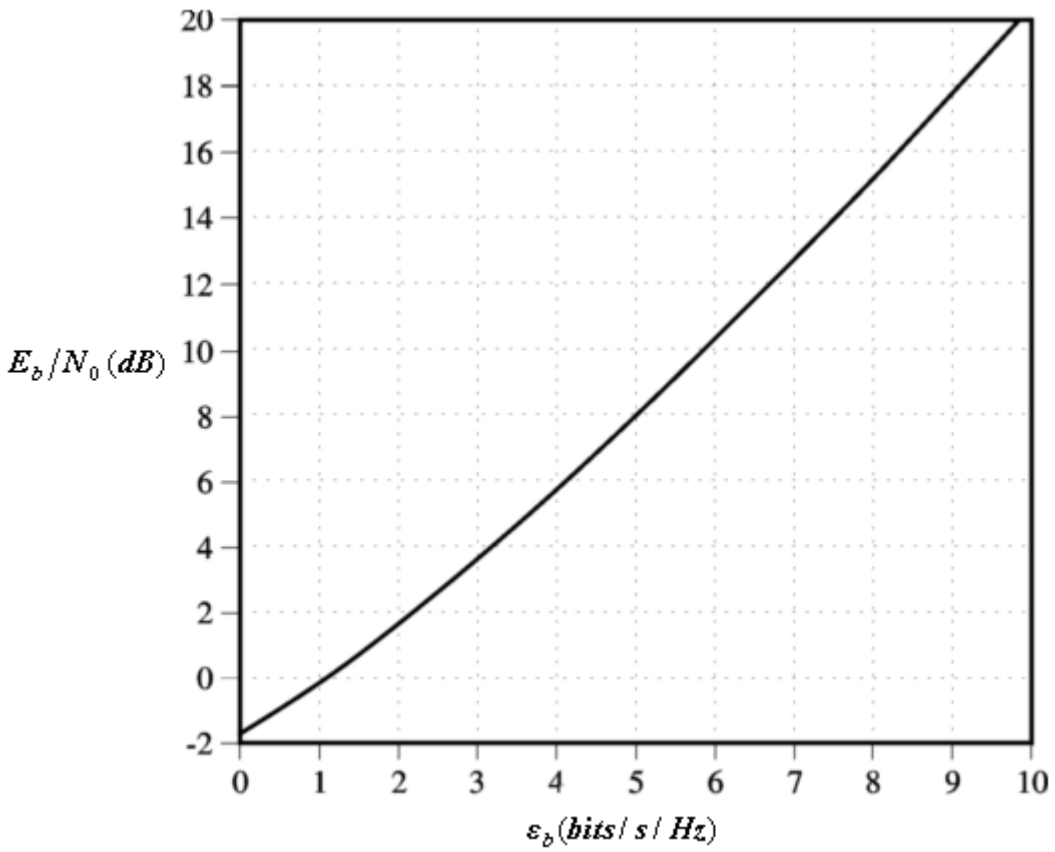


Figura 3- Capacidade de canal em função da eficiência espectral (N=2)

Através da figura 3 observa-se, por exemplo, que a menor relação sinal ruído possível de um esquema codificado que apresente uma eficiência espectral de 5bits/s/Hz, é de 8 dBs. Ou seja, nenhum sistema que possua eficiência espectral de 5 bits/s/Hz, é capaz de operar em uma condição de $P_b(e) \rightarrow 0$, para uma relação sinal ruído inferior a 8 dBs.

A figura 4 mostra o máximo desempenho possível que um sistema pode alcançar para uma dada eficiência espectral e para diversos esquema de modulação em uma dimensão (N=1) como PAM, e em duas dimensões (N=2) como QAM.

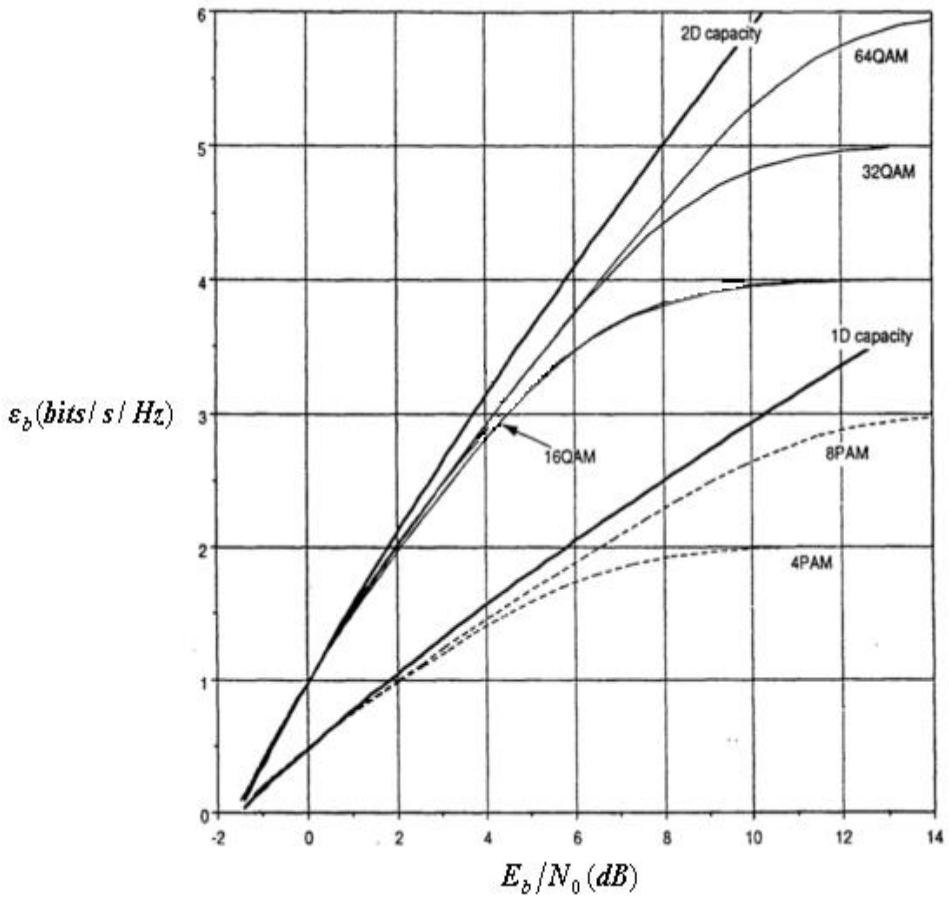


Figura 4 –Capacidade de canal para diferentes esquema de modulação (N= 1 e N=2)

Através da figura 4 observa-se que a capacidade de canal para sistemas de duas dimensões é maior que para unidimensionais, como era de se esperar. Observa-se, por exemplo, que para uma eficiência espectral da ordem de 5 bits/s/Hz, uma modulação 32QAM possui maior eficiência espectral que uma modulação 16QAM, ao passo que nenhum ganho significativo é observado quando se passa para uma modulação 64QAM.

Todos os resultados anteriores, assumem que o comprimento da palavra código pode ser feita tão longo quanto se queira como condição para atingir os valores de relação sinal ruído mínimos fornecidos para uma dada capacidade de canal. Na prática, porém tal premissa não é aplicável para sistemas reais devido a complexidade associada e eventualmente ao tempo de latência associado, fato ainda mais relevante em sistema de transmissão contínuos de informação em tempo real.

Shannon derivou um limitante superior da probabilidade de erro de palavra utilizando um "*sphere packing bound*", através do qual, é possível estabelecer uma relação entre a perda em decibéis em relação aos valores de limite de capacidade de canal para uma dada taxa de codificação, quando o comprimento da palavra código é limitado. Este limitante superior esta descrito de forma compacta pela expressão abaixo:

$$P_w(e) \geq Q(E_b / N_0, r, n) \quad (49)$$

A parte direita da equação acima representa a probabilidade de um vetor gaussiano de dimensão n, correspondendo a palavra recebida, situar-se fora de uma região geométrica em um espaço euclidiano n-dimensional, para a qual a decisão seria correta.

Abaixo é apresentado o resultado gráfico para alguns valores de taxa de codificação, e para um valor da probabilidade de erro de palavra de $1x10^{-4}$.

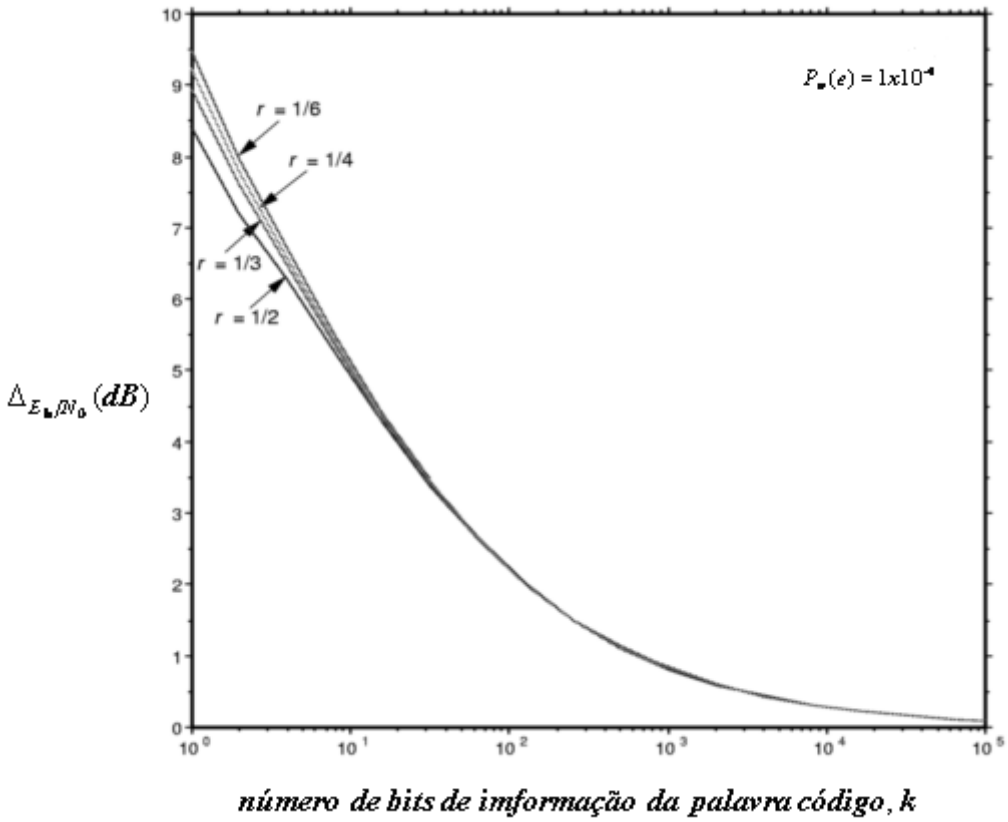


Figura 5- Perda de desempenho devido a restrição no comprimento da palavra código .

Através da analise do gráfico da figura 5 acima se observa que para se atingir valores de relação sinal ruído próximos aos valores de capacidade de canal, o comprimento das palavras código estão na casa de dezenas de milhares, o que na prática tornaria o esquema de decodificação demasiadamente complexo, conforme discutido anteriormente.

Outro fato que pode ser observado, é que a taxa de codificação possui uma influência relevante apenas quando o comprimento dos blocos são muito curtos, a medida que o comprimento aumenta, as curvas convergem.

Os resultados descritos no teorema da codificação de canal de Shannon e analisados até o momento foram estabelecidos para o erro de probabilidade de palavra codificada conforme visto na eq.(34), e que por ser o limite superior para a probabilidade de erro de bit, também é valido para esta probabilidade conforme mostrado na eq.(35).

O teorema mostra em essência, como reduzir a probabilidade de erro de palavra e não de bit.

Neste ponto, a fim de se discutir os conceitos e estratégias associados à codificação turbo, a introdução de um limitante superior para a probabilidade de erro de palavra e um limitante superior para a probabilidade de erro de bit, se tornam necessários .

Seja um código binário linear $\boldsymbol{C}$, definindo como evento de erro E_h para uma palavra $\boldsymbol{X}_0 \in \boldsymbol{C}$ enviada no canal, a ocorrência da decisão por uma palavra $X_k \in X_h$, sendo X_h o conjunto de todas as palavras que distam de $\boldsymbol{h}$ posições de $\boldsymbol{X}_0$.

Definindo a função "*output weight enurating function*" ou OWEF, como o polinômio na indeterminada $\boldsymbol{H}$:

$$B(H) = \sum_{h=0}^{n} B_h H^h \quad (50)$$

E sendo que os coeficientes $\boldsymbol{B_h}$,representam o número de palavras código que distam $\boldsymbol{h}$ posições de $\boldsymbol{X}_0$.

Para um código binário linear C de taxa $r = k/n$, qualquer palavra código pode ser escrita como $X_i(x) = U_i(x) * G(x)$. Onde $G(x)$ é polinômio gerador do código, e $U_i(x)$, representa uma das 2^k possíveis sequencias do vetor de entrada de comprimento $k \in \{U\}$.

Seja agora, $X_i(x)$ e $X_j(x)$ duas palavras código, $X_i(x) + X_j(x) = (U_i(x) + U_j(x)) * G(x)$. Mas $U_i(x) + U_j(x) = U_k(x) \in \{U\}$, e $X_i(x) + X_j(x) = U_k(x) * G(x) = X_k(x) \in C$. Então, o número de palavras que distam de h ($h = d_{\min},..n$) posições de uma palavra X_0 transmitida, dentro do universo de todas as palavras código pertencente ao código, é o mesmo independente de quem seja X_0, uma vez que varrendo todos os vetores de entrada possíveis para um dado vetor de entrada U_0, todas as palavras do código serão geradas independente de quem seja U_0 .

Conforme visto, a distribuição de distâncias h existentes no código, é independente de X_0. Então sem perda de generalidade, seja X_0 o vetor nulo (dado que todo código linear possuía a palavra nula), então $B_h(H)$, é a própria distribuição de pesos das palavras do código em consideração.

Usando então o conceito da OWEF, e o limitante de união, pode-se definir um limitante superior para a probabilidade de erro de palavra conforme abaixo:

$$P_w \leq \sum_{h=d_{\min}}^{n} B_h P(E_h) \quad (51)$$

para um canal Gaussiano:

dado que :

$$\frac{1}{2} erfc(x) < e^{-x^2} \quad (52)$$

então:

$$P_w(e) \leq \frac{1}{2} \sum_{h=d_{\min}}^{n} B_h erfc\left(\sqrt{\frac{hr_c E_b}{N_0}}\right) < \left[B(H) - 1\right]_{H=\exp(-r_c E_b/N_0)} \quad (53)$$

A fim de formular um limitante para a probabilidade de erro de bit, $\boldsymbol{E_{h,m}}$ é definido como o evento de erro associado à escolha errônea de uma código palavra $\boldsymbol{X_i} \in \boldsymbol{X_h}$, que dista de $\boldsymbol{h}$ posições de $\boldsymbol{X}_0$, e que provoca a ocorrência de um erro na $\boldsymbol{m}$-ésima posição do vetor de informação enviado de comprimento $\boldsymbol{k}$.

Então utilizando o limitante de união:

$$P_b(e) \leq \frac{1}{\mathrm{k}} \sum_{m=1}^{k} \sum_{h=d\min}^{h} P\left\{E_{h,m} \middle| X_0\right\} \quad (54)$$

definindo o evento de erro:

$$E_{x_i,x_0} = \left\{X_i \in X_h \text{ seja escolhido em preferencialmente a } X_0\right\}$$

e

$$\begin{aligned} \Pr\left\{E_{h,m}, y \middle| X_0\right\} &\leq \sum_{x_i \in x_h : i \neq 0} \Pr\left\{u_{i,m} \neq u_{0,m}, E_{x_i,x_0}, y \middle| X_0\right\} \\ &= \sum_{x_i \in x_h : i \neq 0} \Pr\left\{u_{i,m} \neq u_{0,m} \middle| E_{x_i,x_0} X_0\right\} \mathrm{X} \Pr\left\{E_{x_i,x_0}, y \middle| X_0\right\} \end{aligned} \quad (55)$$

definindo a função indicador como abaixo:

$$I_{u_{i,m}, u_{o,m}} = \begin{cases} 1, se\ u_{i,m} \neq u_{o,m} \\ 0, caso\ contrário \end{cases} \quad (56)$$

então:

$$\Pr\left\{u_{i,m} \neq u_{0,m} \middle| E_{x_i,x_0} X_0\right\} = I_{u_{i,m} u_{0,m}} \quad (57)$$

Estendendo o resultado para todos os $\boldsymbol{k}$ bits do vetor de informação, resulta na própria distancia de Hamming entre os dois vetores em questão:

$$\sum_{m=1}^{k} \Pr\left\{u_{i,m} \neq u_{0,m} \middle| E_{x_i, x_0} X_0\right\} = \sum_{m=1}^{k} I_{u_{i,m}, u_{o,m}} = w \quad (58)$$

Mais uma vez como tanto a distribuição de $\boldsymbol{w}$ como a distribuição de $\boldsymbol{h}$ são independentes do para $\boldsymbol{X}_0, \boldsymbol{X}_i$, e considerando agora todas as possíveis palavras código. Aplicando-se o limitante superior de união, obtém-se a expressão abaixo para a probabilidade de erro de bit.

$$P_b(e) \leq \sum_{w=1}^{k} \sum_{h=d\min}^{h} \frac{w}{k} A_{w,h} P(E_h) \quad (59)$$

Onde $\boldsymbol{A_{w,h}}$ representa o número de palavras código cujo somatório do número de erros causados no vetor de entrada é dado por $\boldsymbol{w}$, e cuja palavra código decidida erroneamente dista do vetor de entrada em $\boldsymbol{h}$ posições.

Define-se a seguir, a função "*input-output weight enumerating function*" ou IOWEF, como:

$$A(W, H) = \sum_{w=0}^{k} \sum_{h=0}^{n} A_{w,h} W^w H^h \quad (60)$$

Onde $\boldsymbol{A_{w,h}}$, representa o numero de palavras código com peso de Hamming $\boldsymbol{h}$ associadas aos vetores de informação com peso $\boldsymbol{w}$. Utilizando eq.(48) e o limitante de união, é possível estabelecer um limitante superior para a probabilidade de erro de bit para um canal gaussiano, dado pela expressão abaixo:

$$P_b(e) \leq \frac{1}{2} \sum_{h=d_{\min}}^{n} B_h^{(b)} erfc\left(\sqrt{\frac{h r_c E_b}{N_0}}\right) \quad (61)$$

definindo a multiplicidade de bit como:

$$B_h^{(b)} = \sum_{w=1}^{k} \frac{w}{k} A_{w,h} \quad (62)$$

e função CWEF como:

$$A(w,H) = \sum_{h=0}^{n} A_{w,h} H^h \quad (63)$$

Que representa a distribuição de palavras código associadas aos vetores de entrada de peso w. Definindo um segundo limitante superior para a probabilidade de erro de bit dado por:

$$P_b(e) \leq \frac{1}{2} \sum_{w=1}^{k} \sum_{h=d_{\min}}^{n} \frac{w}{k} A_{w,h} erfc\left(\sqrt{\frac{hr_c E_b}{N_0}}\right) < \sum_{w=1}^{k} \frac{w}{k} \left[A(w,H)\right]\Big|_{H=\exp(-r_c E_b/N_0)} \quad (64)$$

Desta forma a fim de se diminuir a probabilidade de erro de bits, duas estratégias são possíveis:

1-Reduzir a probabilidade de erro de palavra aumentando a distância de Hamming código através do aumento do comprimento “n” das palavras. Esta estratégia, corresponde a que foi antecipado pelo Teorema de Shannon

2-Diminuir a multiplicidade de erro de bit, (ao menos dos termos mais significativos). Ou seja, através de esquemas de codificação que maximizem o número de palavras com peso elevado para o universo de vetores de entrada de comprimento k, isto é, muitos vetores de entrada levando a palavras de peso elevado, e poucos vetores de entrada levando a palavras de baixo peso.

10 Estratégias de Minimização da Multiplicidade de Bit

No esquema proposto em [1], onde os códigos turbo foram apresentados pela primeira vez, foi obtido, através de um esquema de codificação utilizando códigos

convolucionais concatenados com um *interleaving* de comprimento 65536 e através de um esquema iterativo de decodificação, atingiu uma probabilidade de erro de 10^{-5} para uma relação sinal ruído de 0.7dB do limite de capacidade de canal , após 18 iterações.

No entanto, neste trabalho inicial uma série de questões em relação a quais fatores e a maneira como eles contribuíam para o desempenho observado, ficaram em aberto. Em [34], uma análise dos diversos fatores que influenciam no desempenho dos códigos turbo e uma proposta para justificativa dos ganhos observados nos esquemas de decodificação turbo foi proposta.

Um dos principais fatores observados que influenciam os resultados de desempenho nos esquemas de concatenação é o chamado "ganho de *interleaving*", que será descrito a seguir.

11 O ganho de interleaving

O ganho de *interleaving* diz respeito ao ganho de desempenho observado através da utilização e do aumento do entrelaçamento (*interleaving)* utilizado.

Este fenômeno pode ser observado através da análise da "CWEF", utilizada para o cálculo do limitante superior para a probabilidade de erro de bit.

Suponha um esquema de códigos de bloco concatenados conforme a figura abaixo:

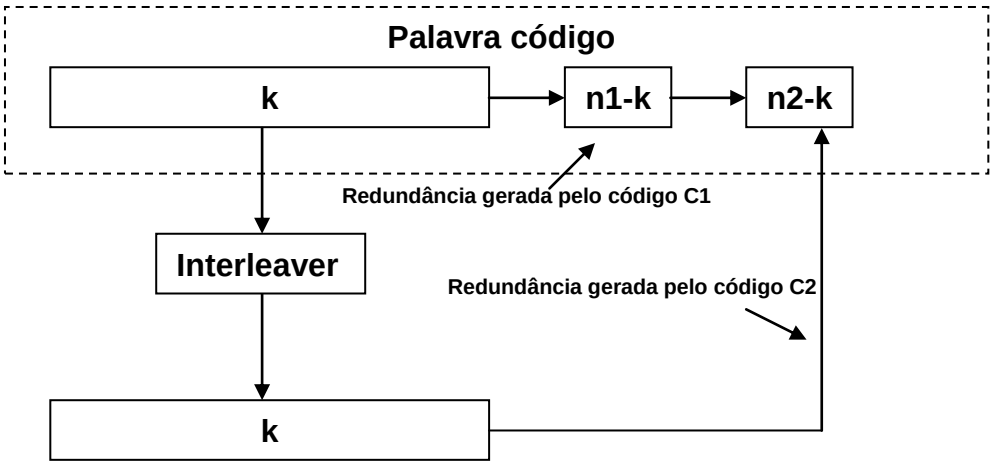


Figura 6- Concatenação de códigos de bloco. Fonte: Autor

A fim de se determinar qual é a CWEF do código concatenado final, é necessário se ter conhecimento de qual regra de permutação o *interleaving* esta aplicando aos bits que serão fornecidos ao segundo código. Uma vez que para o segundo código, devido a permutação causada pelo *interleaving*, a CWEF passará a depender não apenas do peso da palavra de entrada, mas também da posição dos bits de informação. Este fato torna o cálculo da CWEF proibitivo a medida que é aumentado o comprimento do *intervealing*/código.

A fim de se evitar limitações impostas ao calculo da CWEF para a avaliação dos ganhos proporcionados pela utilização do *interleaving*, é utilizado o conceito de *interleaving* uniforme.

O *interleaving* uniforme, representa a média de todos os possíveis *interleavings* (produzindo todas as permutações possíveis) para um dado comprimento de palavra código. Desta forma, o efeito causado por este *interleaving* é de um embaralhamento aleatório dos dados, quebrando a dependência em relação a posição dos bits de informação que serão enviados ao segundo codificador, além de representar o desempenho da média de todos os possíveis *interleavings*. Desta forma se pode dizer que para cada valor de relação sinal ruído existe ao menos um *interleaving* que possui desempenho igual ao conseguido pelo *interleaving* uniforme.

Como este *interleaving* representa a média de todas as combinações de posição para uma entrada de peso $\boldsymbol{w}$ e um comprimento de palavra $\boldsymbol{k}$, a IOWEF deverá ser ponderada por $\binom{k}{w}$, que representa todas as combinações do vetor de comprimento $\boldsymbol{k}$ e com $\boldsymbol{w}$ posições diferentes de 0.

Assim a CWEF pode finalmente ser escrita como o produto das CWEF individuais (e normaliza pela combinações do *interleaving* uniforme), conforme abaixo:

$$A^{C_p}(w,H) = \frac{A^{C_1}(w,H)\mathrm{x}A^{C_2}(w,H)}{\binom{k}{w}} \quad (65)$$

Utilizando a IOWEF para o no cálculo do limitante de erro de bit, é apresentado a seguir um exemplo comparativo dos ganhos devidos ao *interleaving* utilizando um

código concatenado (10, 4) gerado a partir de dois códigos constituintes de Hamming (7, 4).

O resultado mostrado na figura 7 representa o resultado para três casos distintos: O resultado conseguido pela permutação 0123 dos quatro bits de informação, que representa a permutação identidade, ou seja o desempenho do próprio código, depois o caso em que a operação de permutação realizada pelo interleaving é 0132, representa a operação de permutação entre a terceira e quarta posição do vetor de entrada de 4 bits, e finalmente tem-se a operação correspondente ao *interleaving* uniforme.

A operação identidade, que representa nenhum *interleaving* ,é a que apresenta o pior desempenho, a operação de permutação 0132 representa o melhor resultado neste caso. O desempenho do interleaving uniforme, utilizando a CWEF conforme eq.(65) no cálculo do limitante de erro de bit, representa um desempenho intermediário entre estes dois resultados.

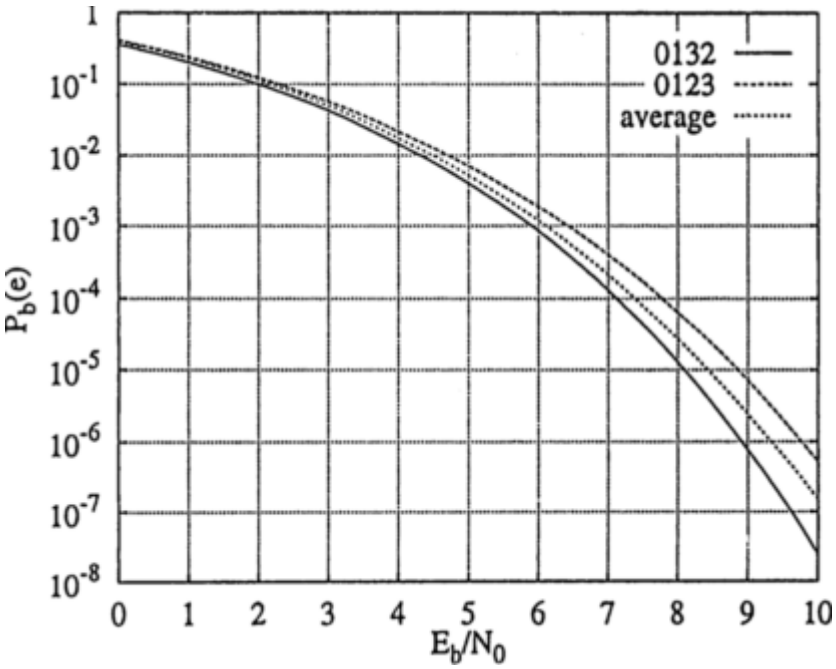


Figura 7- Efeito do Interleaving na $P_b(e)$ para um código de Hamming (7,4) concatenado

Se o interleaving não for restrito pelo comprimento de do vetor de entrada $\boldsymbol{k}$, mas a um número múltiplo de vetores de entrada $\boldsymbol{m}$, então se pode escrever a IOWEF do código concatenado resultante, como a função do código estendido ($\boldsymbol{mn}, \boldsymbol{mk}$) :

$$A^{C_i^m}(W,H) = [A^{C_i}(W,Z)]^m \quad (66)$$

e a CWEF como:

$$A^{C_P^m}(w,H) = \frac{A^{C_1^m}(w,H) \mathrm{x} A^{C_2^m}(w,H)}{\binom{mk}{w}} \quad (67)$$

A figura abaixo mostra os resultados de desempenho para o mesmo código concatenado do exemplo anterior, utilizando código de Hamming (7,4) como código constituinte, porem utilizando comprimentos de *interleaving* de 4, 8, 40, 80, 400 correspondendo a $\boldsymbol{m}$ = 1,2,10,20 e 100.

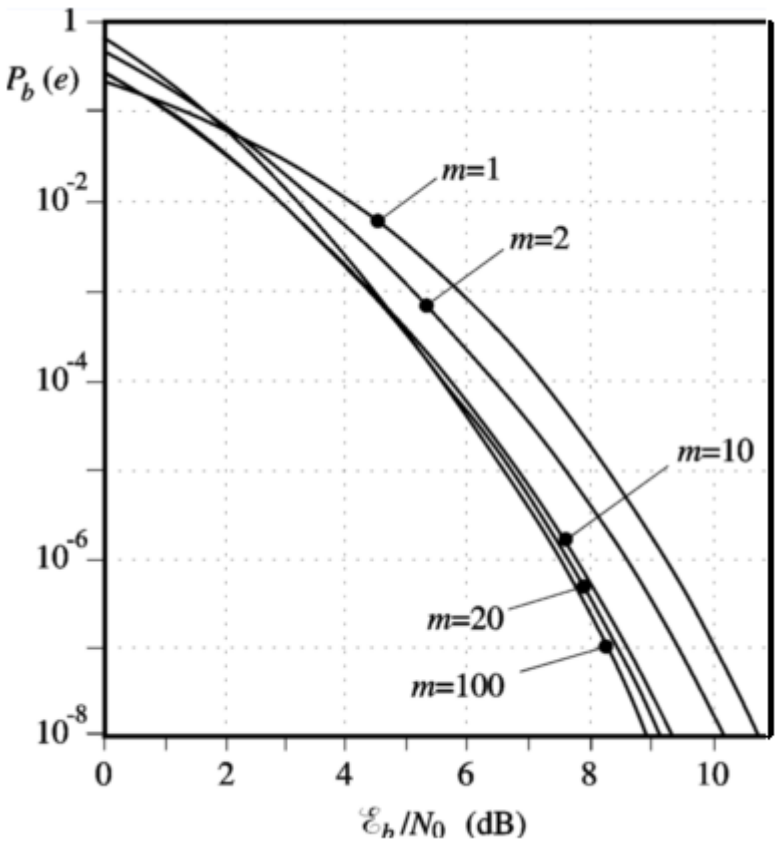


Figura 8- Influência do comprimento do interleaving na $P_b(e)$

Através da análise da figura, é possível verificar que ocorre um efeito de saturação a medida que $\boldsymbol{m}$ aumenta, sendo que para $\boldsymbol{m}$ maior que 10 os ganhos são praticamente desprezíveis a medida que se aumenta o comprimento do *interleaving.*