Universidade Federal de Pernambuco

Centro de Informática

Graduação em Ciência da Computação

ALGORITMOS DE DECODIFICAÇÃO POR DECISÃO SUAVE APLICADOS A CÓDIGOS BCH

Lucas Minoru Ferreira Harada

TRABALHO DE GRADUAÇÃO

Recife, Agosto de 2014

Universidade Federal de Pernambuco

Centro de Informática

Graduação em Ciência da Computação

ALGORITMOS DE DECODIFICAÇÃO POR DECISÃO SUAVE APLICADOS A CÓDIGOS BCH

Trabalho apresentado ao Programa de Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Daniel Carvalho da Cunha (dcunha@cin.ufpe.br)

Aluno: Lucas Minoru Ferreira Harada (lmfh@cin.ufpe.br)

Recife, Agosto de 2014

Agradecimentos

Aos meus pais, pelo apoio, confiança e oportunidades que me foram dadas. Vocês serão meus eternos professores.

À Fernanda Castro, pelo amor, companheirismo e gerência de projetos durante toda esta graduação. Com você ficou tudo mais fácil.

A toda minha família pelo apoio e conselhos recebidos.

A todos os meus amigos, em especial aqueles que viraram noites na faculdade, pela companhia e amizade.

Ao prof. Cecilio Pimentel pelas inúmeras ajudas durante o desenvolvimento deste trabalho.

Ao meu orientador Daniel Cunha pela oportunidade e direcionamento, que tornaram este trabalho possível.

Lista de Siglas

BPSK – *Binary Phase Shift Keying*

AWGN – *Additive White Gaussian Noise*

BCH – Bose-Chaudhuri-Hocquenghem

GF – *Galois Field*

BER – *Bit Error Rate*

FER – *Frame Error Rate*

CA – *Chase Algorithm*

SCA – *Stochastic Chase Algorithm*

RS – *Reed-Solomon*

SNR – *Signal-to-Noise Ratio*

Índice de Figuras

Índice de Tabelas

Resumo

A busca por velocidades de transmissão cada vez maiores com um custo cada vez menor tem sido um dos pontos mais discutidos, quando são projetados sistemas de comunicação digital. Neste ponto, além da criação de códigos ótimos, está sendo muito estudada a utilização de algoritmos novos a serem aplicados a códigos corretores de erros já consagrados, visando melhorar seu desempenho. Atualmente, os algoritmos em estudo são os que propõem a utilização de decodificação por decisão suave. Este trabalho visa explicar o funcionamento de dois algoritmos de decodificação por decisão suave, o Algoritmo de Chase e o Algoritmo de Chase Estocástico, e apresentar os resultados da utilização de cada um dos algoritmos em um canal com ruído aditivo gaussiano branco utilizando modulação binária por chaveamento de fase.

Palavras-Chave: codificação de canal, ruído aditivo gaussiano branco, modulação binária por chaveamento de fase, códigos corretores de erros, código de Golay, código BCH, decodificação por decisão suave, algoritmo de Chase, algoritmo de Chase estocástico.

Abstract

The demand for higher data transmission speed with lower costs has been a trending topic in the digital communications systems area. In this area of study, beyond the creation of perfect codes, there have been a lot of studies around the use of new algorithms being used on established codes, aiming to improve its performance. Nowadays, the algorithms being studied are the ones that propose the use of soft decision decoding. This study aims to explain how two soft decision decoding algorithms work, the Chase Algorithm and the Stochastic Chase Algorithm, and show the results about the uses of each one of them over an Additive White Gaussian Noise channel using Binary Phase Shift Keying modulation.

Keywords: channel coding, additive white Gaussian noise, binary phase shift keying modulation, error-correcting code, Golay code, BCH code, soft decoding, soft decision decoding, Chase algorithm, stochastic Chase algorithm.

Sumário

1. INTRODUÇÃO

A codificação de canal, também conhecida por codificação para controle de erros (códigos corretores de erros), constitui uma das principais técnicas de processamento da informação utilizada em sistemas de comunicação digital. O princípio básico da codificação de canal é a adição de redundância à informação com o objetivo de corrigir erros que possam ocorrer no processo de gravação ou transmissão de dados. Desta maneira, a utilização de códigos corretores de erros permite a recuperação da informação original por parte do usuário final com uma probabilidade de erro arbitrariamente pequena.

A contribuição mais importante da área de codificação de canal foi descoberta por Shannon na década de 1940 [1]. Ele concebeu o teorema de codificação de canal que fornece um limitante superior, denominado capacidade de canal, na taxa com que a informação pode ser transmitida de maneira confiável através de um canal ruidoso. Desde então, atingir a capacidade de Shannon para diversos modelos de canais tem sido a meta da comunidade científica que estuda codificação de canal.

Uma das principais técnicas clássicas de codificação de canal se baseia em códigos de bloco lineares, como, por exemplo, os códigos de Hamming [2]. Os códigos de Hamming são largamente utilizados em controle de erros em comunicação digital e sistemas de armazenamento de dados. Seus algoritmos de decodificação são bastante simples de se implementar e são denominados algoritmos de decodificação por decisão abrupta.

Os algoritmos de decodificação por decisão abrupta são baseados na quantização das saídas dos filtros casados localizados nos estágios de demodulação do receptor digital. Devido à quantização, os sinais recebidos são representados por sequências de *bits*, que são utilizadas em um método específico de decodificação chamado decodificação por decisão abrupta (do inglês, Hard Decision Decoding), conforme mencionado anteriormente. A decodificação por decisão abrupta utiliza a estrutura algébrica do código e, por esta razão, também é chamada de decodificação algébrica. O objetivo deste tipo de decodificação é decodificar a sequência binária recebida mais próxima da palavra-código transmitida em termos de distância de Hamming. Apesar de sua simplicidade, a decodificação por decisão abrupta resulta em perda de informação, o que afeta diretamente o desempenho do sistema.

Para evitar a perda de desempenho provocada pela decodificação por decisão abrupta, existe uma técnica na qual as saídas dos filtros casados não são quantizadas ou são quantizadas em mais de dois níveis. Neste caso, diz-se que o decodificador opera com decisões suaves. Portanto, a sequência observada na saída do demodulador é denominada sequência de decisão suave e o algoritmo de decodificação que a emprega recebe o nome de

algoritmo de decodificação por decisão suave [3-5]. A decodificação por decisão suave permite a obtenção de um melhor desempenho quando comparada à decodificação por decisão abrupta, porém exige uma maior complexidade computacional em sua implementação.

1.1. Objetivo

Este trabalho de graduação visa estudar a codificação de canal, explicar e mostrar os efeitos da utilização de algoritmos de decodificação por decisão suave em sistemas de comunicação digital com canal de comunicação com ruído aditivo gaussiano branco sob modulação binária por chaveamento de fase e comparar os resultados obtidos entre os dois algoritmos implementados.

Os objetivos específicos são:

i. Desenvolver estudos sobre algoritmos de decodificação por decisão suave (do inglês, Soft Decision Decoding) de códigos de bloco lineares, como, por exemplo, o algoritmo de Chase e o algoritmo de Chase Estocástico (do inglês, Stochastic Chase) [5] em canais com ruído aditivo Gaussiano.
ii. Avaliar a relação entre a complexidade de implementação e o desempenho dos algoritmos de decodificação a serem estudados por meio de curvas de probabilidade de erro de *bit* versus relação sinal-ruído.

1.2. Estrutura do Documento

Visando o melhor meio de apresentar o conteúdo deste trabalho foram definidos 5 capítulos.

O primeiro capítulo apresenta a fundamentação teórica em que se insere o tema do trabalho, bem como explicita os objetivos propostos.

No segundo capítulo, encontram-se os conceitos básicos sobre sistemas de comunicação digital codificado, isto é, será apresentado o modelo teórico, o modelo prático em que foi estruturado o ambiente de simulação, os códigos que foram utilizados neste trabalho e quais os parâmetros foram utilizados para comparar o desempenho.

No terceiro capítulo, são descritos os dois algoritmos de decodificação por decisão suave. Ainda contém as análises sobre cada algoritmo e apresenta as limitações do Algoritmo de Chase e o motivo do Algoritmo de Chase Estocástico ser capaz de melhorar o desempenho.

No quarto capítulo, estão presentes os resultados das simulações, juntamente com a análise e explicação para cada resultado apresentado.

Por fim, o quinto e último capítulo traz as conclusões do trabalho, assim como ideias para trabalhos futuros.

2. CONCEITOS BÁSICOS SOBRE SISTEMAS DE COMUNICAÇÃO DIGITAL CODIFICADOS

2.1. Modelo Teórico

Um sistema de comunicação digital tem como objetivo transferir dados de uma fonte de informação para um determinado destino de maneira confiável, permitindo que a mensagem seja recebida de forma fidedigna à informação original.

A fonte de informação gera os símbolos a serem transmitidos, pertencentes a um alfabeto finito que na maioria dos casos é binário. Em várias aplicações, a sequência de informação contém muita redundância, e com isso o uso de um codificador de fonte se torna necessário para melhorar a eficiência do sistema. O codificador de fonte extrai o excesso de redundância da informação e como consequência a saída do codificador de fonte produz uma sequência de informação com o mínimo de redundância possível. Esta sequência passa por um codificador de canal que adiciona redundância controlada a esta informação com o objetivo de deixar a transmissão mais confiável. O codificador de canal transforma uma sequência de k símbolos de informação em uma sequência codificada de n símbolos, em que $n > k$. A quantidade de símbolos redundantes introduzida na sequência de informação é, portanto, $n - k$. A taxa de codificação é definida como $R = n/k$. A capacidade de correção t, isto é, quantos símbolos podem ser corrigidos caso o ruído inverta algum símbolo depende de cada codificador de canal utilizado. Em geral, esse processo provoca a diminuição da taxa de transmissão de dados ou o aumento da largura de faixa de frequências do canal em relação ao sistema não codificado, no entanto, há um ganho de codificação que justifica e compensa essa "perda" espectral.

Os símbolos digitais na saída do codificador entram em um modulador que tem a função de convertê-los em formas de onda analógicas para a transmissão. No caso da modulação binária, os *bits* 0 e 1 são mapeados diretamente em duas formas de onda diferentes. Quando a modulação não é binária, os *bits* são agrupados e mapeados em símbolos complexos que variam a amplitude, fase ou frequência de uma senoide.

O canal de comunicação é o meio físico usado por onde o sinal do transmissor é enviado ao receptor. Alguns exemplos de canais de comunicação são: o canal sem fio, o par de fios trançados (fios telefônicos), a rede elétrica (baixa ou média tensão), a fibra ótica e o cabo coaxial. Os sinais que se propagam por esses canais são corrompidos de forma aleatória por alguns tipos de ruído, tais como o ruído aditivo térmico, gerado pela agitação de elétrons na linha e em componentes eletrônicos, e o desvanecimento, gerado pela propagação por multipercursos.

Chegando ao receptor, o demodulador converte a sequência de formas de ondas corrompidas pelo ruído e desvanecimento do canal em uma sequência de símbolos, num processo chamado de demodulação ou detecção. O processo de detecção pode produzir símbolos errôneos, devido à perturbação ruidosa.

Essa sequência (possivelmente errônea) chega ao decodificador de canal e este, conhecendo a redundância introduzida pelo codificador no transmissor e as características estatísticas do canal, decodifica a sequência recebida de maneira mais confiável (com a menor probabilidade de erro possível).

Baseado nas regras de codificação da fonte, o decodificador de fonte procura reproduzir, fielmente, os dados emitidos pela fonte de informação. Devido ao fato do foco deste trabalho ser o codificador de canal, será usado um modelo simplificado, baseado no modelo apresentado em [6], mostrado na Figura 1.

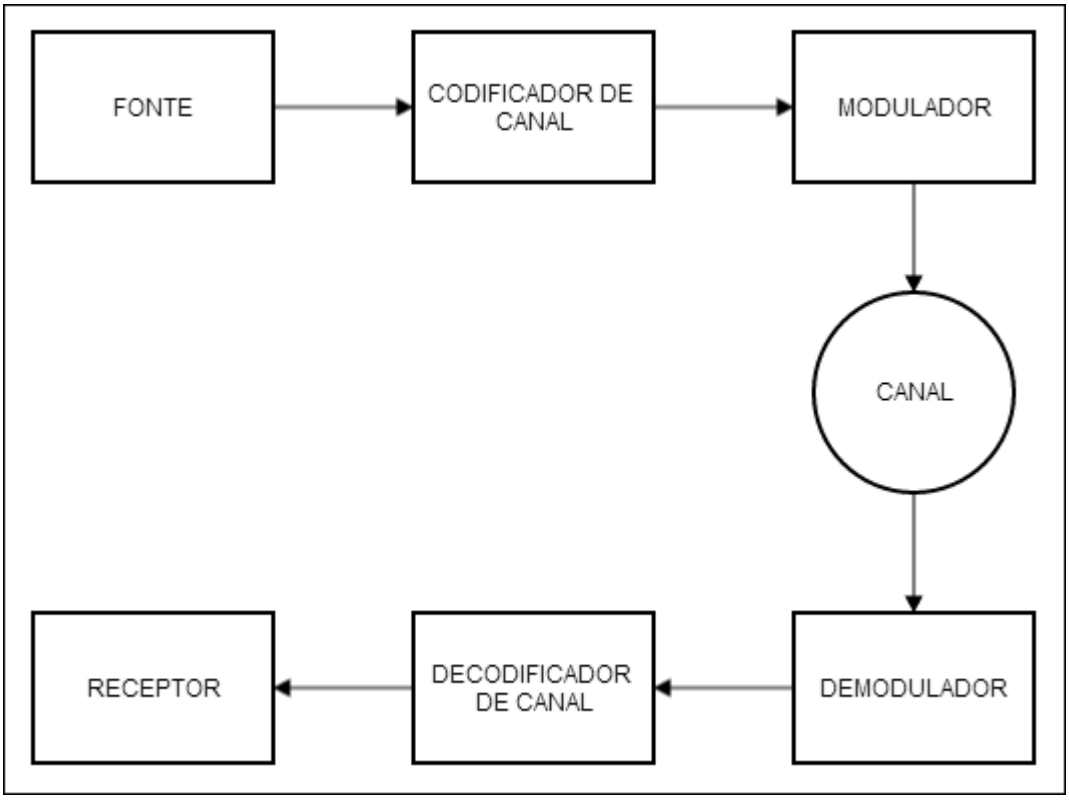

Figura 1 - Modelo simplificado de um sistema de comunicação digital.

Deste modo, a fonte de informação e o codificador de fonte passam a ser chamados apenas de fonte. O decodificador de fonte e o receptor serão considerados um único bloco chamado receptor. O modulador e o demodulador representarão apenas o mapeamento dos dígitos de saída do codificador em símbolos de uma constelação e vice-versa. A parte analógica do modulador e do demodulador ficam embutidas no canal.

A partir deste ponto, só serão tratados sistemas de comunicação binários.

2.2. Modelo do Sistema de Comunicação Codificado

Seja $C(n, k, d)$um código de bloco linear de tamanho n, com mensagem de tamanho k e distância mínima d. A capacidade de correção do código t é definida a partir da distância mínima do código. Um vetor de mensagem $u = (u_0, u_1, \ldots, u_{k-1})$ é codificado em uma palavra-código $c = (c_0, c_1, \ldots, c_{n-1}) \in C$, onde C é o conjunto de palavras-código válidas. Cada c_i é modulado segundo a modulação BPSK (Modulação Binária por Chaveamento de Fase, *Binary Phase Shift Keying*) seguindo a regra

$$x_i = 1 - 2c_i$$

e, em seguida, é transmitido por um canal com ruído aditivo Gaussiano branco (AWGN, *Additive White Gaussian Noise*).

Do lado do receptor, é recebido um vetor com valores reais $r = (r_0, r_1, \ldots, r_{n-1})$. A demodulação é feita com a operação inversa à operação feita na modulação:

$$y_i = \{0, se\ r_i \geq 0; 1, se\ r_i < 0\}$$

A decodificação é feita sobre o vetor demodulado $y = (y_0, y_1, \ldots, y_{n-1})$ e, dependendo da capacidade de correção do código, erros inseridos pelo canal de transmissão são corrigidos. Este procedimento caracteriza a decodificação por decisão abrupta.

2.3. Códigos Corretores de Erros

Como descrito anteriormente, códigos corretores de erros buscam inserir redundância às mensagens para diminuir a taxa de erros após a decodificação, diminuindo, porém, a taxa de dados da transmissão.

Existem duas classes de códigos corretores de erros, quando se trata de codificação de canal: códigos de bloco e códigos convolucionais.

- Códigos de bloco trabalham com conjuntos de *bits* ou símbolos de tamanho fixo pré-determinado.
- Códigos convolucionais trabalham com fluxos de *bits* ou símbolos de tamanhos arbitrários.

Neste trabalho, serão utilizados dois códigos de bloco: códigos de Golay estendido [7], inicialmente, e a família dos códigos BCH (Bose-Chaudhuri-Hocquenghem) [8].

2.3.1. Código de Golay Estendido

O código de Golay estendido (24,12,8) é um código de bloco linear, ou seja, os *bits* de paridade (*bits* adicionados à mensagem para verificar sua integridade no receptor) dependem linearmente dos *bits* da mensagem, com capacidade de correção de 3 *bits*.

Para o código em questão, é definida uma matriz P_{12x12}, a matriz de paridade, em que são derivadas duas outras matrizes $G_{24x12} = [I_{12x12}\ P]$, a matriz geradora, e $H_{24x12} = [P^T\ I_{12x12}]$, a matriz de verificação de paridade, verificando-se que $HG^T = 0$, onde P^T e G^T são as matrizes transpostas da matriz P e da matriz G, respectivamente.

Desta forma, o código de Golay estendido possibilita de forma simples a verificação se a decodificação abrupta é feita com sucesso ou não.

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

Figura 2 - Exemplo da matriz binária P.

A codificação por meio do código de Golay estendido é a simples multiplicação do vetor $u = (u_0, u_1, \ldots, u_{11})$ pela matriz G, tendo como resultado o vetor $c = (c_0, c_1, \ldots, c_{23})$. Já a decodificação abrupta é feita por um método conhecido como decodificação por síndrome. A decodificação por síndrome considera que o vetor demodulado, $y = (y_0, y_1, \ldots, y_{23})$, é resultado de uma soma binária do vetor c e um vetor erro, $e = (e_0, e_1, \ldots, e_{23})$. A síndrome, $s = (s_0, s_1, \ldots, s_{11})$, é o vetor resultado da multiplicação do vetor y pela matriz H, de forma que uma condição necessária e suficiente para que um vetor y seja uma palavra-código válida é que a síndrome desse vetor y seja igual ao vetor nulo.

Cada síndrome está associada a exclusivamente um determinado padrão de erro corrigível, de modo que é possível saber qual o vetor erro que deve ser somada binariamente à saída do demodulador para que a sequência seja decodificada corretamente, desde que a quantidade total de *bits* invertidos não ultrapasse 3 *bits*.

Algebricamente, o código de Golay estendido pode ser descrito a seguir. A partir do vetor mensagem u, obtemos c:

$$c = uG \tag{1}$$

O vetor demodulado y é o vetor c enviado pelo canal somado a um vetor erro e:

$$y = c + e \tag{2}$$

Substituindo (1) em (2), obtemos a equação (2.1):

$$y = uG + e \tag{2.1}$$

A equação (3) mostra a definição de síndrome dada anteriormente:

$$s = yH^T \tag{3}$$

Substituindo (2) em (3), chegamos a equação (3.1):

$$s = (uG + e)H^T \tag{3.1}$$

Fazendo a multiplicação dos fatores de dentro dos parênteses, temos a equação (3.2):

$$s = uGH^T + eH^T \tag{3.2}$$

Por definição, sabemos que:

$$GH^T = 0 \tag{3.3}$$

Logo, substituindo (3.3) em (3.2), concluímos a prova de que cada síndrome está associada unicamente a um padrão de erro corrigível:

$$s = eH^T \tag{4}$$

2.3.2. Códigos BCH

Os códigos BCH (Bose-Chaudhur-Hocquenghem) formam uma subclasse de códigos corretores de erros, dentro da classe de códigos de blocos lineares, chamados de códigos cíclicos que são construídos a partir de Campos de Galois (GF, do inglês, *Galois Field*), onde a cardinalidade de um GF define o comprimento dos códigos definidos a partir dele, por exemplo: um GF de cardinalidade 16 define um código BCH de comprimento 15. Sua principal característica está relacionada à facilidade de geração e decodificação, o que torna os códigos BCH potenciais candidatos a serem utilizados em aplicações práticas. A real importância dos códigos BCH vem do fato que esses códigos são capazes de corrigir todos os padrões contendo até t erros, por meio de um algoritmo simples (do ponto de vista prático) e facilmente implementável. Tal algoritmo é conhecido por Algoritmo de Berlekamp-Massey.

Considere um exemplo de construção de um código BCH binário e primitivo. Seja α um elemento primitivo do campo de Galois $GF(2^4)$ indicado na Tabela 1, tal que $\alpha^4 + \alpha + 1 = 0$. A partir daí, é possível obter um código BCH com distância igual a 5 e capacidade corretora igual a 2, gerado pelo polinômio $g(x) = mmc(m_1(x), m_3(x)) = mmc(x^4 + x + 1, x^4 + x^3 + x^2 + x + 1) = x^8 + x^7 + x^6 + x^4 + 1$, em que $m_1(x)$ é o polinômio minimal de α e $m_3(x)$, o polinômio minimal de α^3, conforme indicado na Tabela 2. Portanto, o código construído é um código BCH (15,7,5), com distância mínima maior ou igual a 5. Como o peso do polinômio gerador vale 5, temos que a distância mínima do código é exatamente 5.

Tabela 1 - Campos de Galois $GF(2^4)$.

Elemento	Representação polinomial
0	0
1	1
α	α
α^2	α^2
α^3	α^3
α^4	$1 + \alpha$
α^5	$\alpha + \alpha^2$
α^6	$\alpha^2 + \alpha^3$
α^7	$1 + \alpha + \alpha^3$
α^8	$1 + \alpha^2$
α^9	$\alpha + \alpha^3$
α^{10}	$1 + \alpha + \alpha^2$
α^{11}	$\alpha + \alpha^2 + \alpha^3$
α^{12}	$1 + \alpha + \alpha^2 + \alpha^3$
α^{13}	$1 + \alpha^2 + \alpha^3$
α^{14}	$1 + \alpha^3$

Tabela 2 - Polinômios minimais.

Elemento	Polinômio minimal
0	x
1	$x + 1$
$\alpha, \alpha^2, \alpha^4, \alpha^8$	$x^4 + x + 1$
$\alpha^3, \alpha^6, \alpha^9, \alpha^{12}$	$x^4 + x^3 + x^2 + x + 1$
α^5, α^{10}	$x^2 + x + 1$
$\alpha^7, \alpha^{11}, \alpha^{13}, \alpha^{14}$	$x^4 + x^3 + 1$

A codificação por meio do código BCH é feita pela multiplicação de um polinômio mensagem (que pode se tratar como um vetor mensagem, onde cada posição do vetor representa o coeficiente de uma potência), $u(x)$, e o polinômio gerador, $g(x)$. A decodificação abrupta de um código BCH é feita pelo Algoritmo de Berlekamp-Massey citado anteriormente.

Para este trabalho, foram utilizados os códigos BCH definidos a partir dos Campos de Galois $GF(2^5)$, $GF(2^6)$ e $GF(2^7)$ mostrados na Tabela 3 [8].

Tabela 3 - Relação de Campos de Galois e Códigos BCH utilizados.

Campos de Galois	Código
$GF(2^5)$	BCH (31,16,7)
$GF(2^6)$	BCH (63,45,7)
	BCH (63,30,13)
$GF(2^7)$	BCH (127,106,7)

2.4. Avaliação de Simulações

As simulações feitas foram avaliadas tendo como medidas a complexidade de cada algoritmo, as suas probabilidades de erro por *bit* enviado (BER, *Bit Error Rate*) e probabilidades de erro por mensagem enviada (FER, *Frame Error Rate*) para cada nível de sinal ruído, onde quanto menor estas probabilidades, melhor a eficiência do código. Outro meio de se avaliar as mesmas medidas é em relação ao nível de energia utilizado para se obter uma dada probabilidade de erro.

As probabilidades são tomadas comparando o vetor entregue ao receptor, após passar pelo canal ruidoso, e o vetor enviado pelo transmissor. Para o cálculo do BER, é contabilizado cada *bit* errado e dividido pela quantidade total de *bits* enviados, enquanto para o cálculo do FER, é contabilizado somente a mensagem inteira. Por exemplo, se apenas 1 *bit* for invertido em uma mensagem de 10 *bits*, a BER é 10% e a FER é 100%, pois 1 *bit* invertido configura uma mensagem inteira errada.

Já a complexidade de um algoritmo é calculada em relação à quantidade de padrões de testes utilizados, um fator comum a ambos os algoritmos. Quanto maior a quantidade de padrões de testes, maior será a complexidade do algoritmo. Por exemplo, se ambos os algoritmos conseguem um desempenho de 10^{-5} em relação à FER, mas o primeiro utiliza 200 padrões de testes e o segundo utiliza 1000 padrões de testes, temos que o primeiro é um algoritmo com menor complexidade.

3. ALGORITMOS DE DECODIFICAÇÃO POR DECISÃO SUAVE

No capítulo anterior, foram discutidas as descrições de dois códigos algébricos e mostradas as limitações da decodificação abrupta em ambos os códigos, isto é, para o código BCH (127,106,7), por exemplo, se ao passar pelo canal de comunicação houver a inversão de mais de 3 *bits*, o decodificador não consegue corrigir os erros inseridos e a decodificação falha.

Entretanto, um sistema de comunicação digital possui informações, como a confiabilidade de *bit*, acerca da transmissão que, ao ser usada de uma forma conveniente, pode melhorar o desempenho sem que seja necessário um grande incremento no custo do sistema. Um sistema que emprega algoritmos de decodificação por decisão suave é dito aquele que utiliza toda a informação disponível ou parte dela de modo que melhore o desempenho do sistema ao preço de um aumento de complexidade.

O aumento da complexidade na estrutura do lado do receptor (toda a estrutura após o canal de comunicação) se dá pela necessidade do decodificador ter conhecimento dos valores reais do sinal da saída do canal de comunicação, pois o demodulador por receber um sinal analógico e transformar em uma saída digital acaba por descartar qualquer informação a mais, além do mais provável *bit* correto.

Neste capítulo, serão discutidos dois algoritmos de decodificação por decisão suave. O primeiro deles, Algoritmo de Chase (CA, do inglês, *Chase Algorithm*), é um algoritmo bem consolidado proposto em 1972 por David Chase. Já o segundo, Algoritmo de Chase Estocástico (SCA, do inglês, *Stochastic Chase Algorithm*), é um algoritmo proposto em 2010 [5] para a classe de códigos Reed-Solomon (RS) com o intuito de melhorar o desempenho do Algoritmo de Chase; neste trabalho, o Algoritmo de Chase Estocástico será implementado em códigos BCH para serem estudadas a melhora em relação ao Algoritmo de Chase e as limitações.

3.1. Algoritmo de Chase

O Algoritmo de Chase utiliza a informação da confiabilidade de *bit* para ampliar o conjunto de sinais possíveis ao procurar a palavra codificada que foi enviada, logo é necessário que o decodificador tenha conhecimento dos valores reais da saída do canal de comunicação. A confiabilidade de *bit* é descrita como o módulo do valor real recebido pelo decodificador.

A ideia geral do algoritmo é, ao invés de decodificar somente a sequência demodulada, criar um conjunto de sequências possíveis, invertendo os *bits* das posições menos confiáveis e verificando se são válidas, pois tendo elas os menores módulos, têm uma maior probabilidade de terem sido invertidas pela ação do ruído do canal. Ao final da criação do conjunto, é selecionada como a sequência correta aquela que minimiza a distância Euclidiana entre ela e a sequência real da saída do canal de comunicação. Minimizar a distância Euclidiana entre a sequência válida e a sequência real da saída do canal de comunicação significa maximizar a correlação entre as duas sequências.

Formalmente, o algoritmo é definido a seguir. Os termos usados são referentes ao modelo apresentado no capítulo anterior.

i. A partir de r obtenha a palavra y gerada pela demodulação de r.

ii. Determine as P posições menos confiáveis de r e crie um conjunto B com 2^P padrões de testes em que cada padrão $b_j, j = 0,1,\ldots,2^{P-1}$ é uma sequência binária de tamanho n com todas as combinações de *bits* 0 e *bits* 1 nas P posições menos confiáveis.

iii. Crie uma lista de palavras-código candidatas Λ obtidas a partir da decodificação abrupta de $y_j = y \oplus b_j$, em que $b_j \in B$. A cardinalidade de Λ pode ser menor que 2^P, caso a decodificação relativa a alguma sequência y_j não seja realizada com sucesso.

iv. A palavra-código decodificada será a palavra de Λ que minimiza a distância Euclidiana entre a sua versão decodificada (Λ_j) e r. Partindo da distância Euclidiana original e considerando o modelo demonstrado na Seção 2.2 deste trabalho, obtemos uma forma mais simples para o cálculo da menor distância Euclidiana:

$$\sum_{i=1}^{n} r_i c_i^j$$

Em outras palavras, a palavra-código escolhida será aquela entre as palavras-código válidas de Λ que maximiza a correlação com o vetor r.

A partir dessas definições, algumas limitações do algoritmo se tornam evidentes:

1. A primeira e mais evidente limitação é caso um *bit* invertido que tenha sido avaliado como mais confiável que outros P *bits*, este erro inserido na mensagem não será tratado, independente da diferença de confiabilidade. Por exemplo, se o P-ésimo *bit* menos confiável tiver confiabilidade de 0,02 e o $(P+1)$-ésimo *bit* menos confiável tiver 0,021 e tiver sido invertido pelo ruído do canal de comunicação, o erro inserido no $(P+1)$-ésimo *bit* menos confiável não será tratado, ainda que o diferença de confiabilidade para o P-ésimo *bit* menos confiável seja de somente 0,001.
2. A complexidade do algoritmo cresce exponencialmente com o valor de P. Um valor razoável de P é necessário para que a palavra-código enviada esteja inclusa no conjunto Λ, entretanto um alto valor de P prejudica o desempenho do sistema no tempo de execução.

3. Alguns padrões de testes (y_j) ao serem decodificados acabam resultando em uma palavra-código válida (Λ_j) repetida. Este caso acaba sendo um aumento de complexidade sem nenhum ganho associado.

3.2. Algoritmo de Chase Estocástico

Em 2010, foi proposta uma abordagem diferente durante a criação dos padrões de testes visando ultrapassar as limitações apresentadas na seção 3.1 deste capítulo. O artigo [5] propõe uma abordagem estocástica, isto é, de natureza probabilística, na criação dos padrões de testes baseado na confiabilidade de cada *bit* para códigos Reed-Solomon.

Para introduzir a abordagem estocástica no modelo de sistema de comunicação digital, é criada uma função P_r para representar os valores reais da saída do canal de comunicação no domínio das probabilidades. Para cada posição do vetor de valores reais, temos um valor p associado:

$$p_i = P_r(r|x_i = 1) = \frac{1}{1 + e^{\frac{2r_i}{\sigma^2}}}$$

No domínio probabilístico, a confiabilidade do *bit* é definida como $|\, p_i - 0{,}5 \,|$ também podendo ainda ser descrito como $|\, r_i \,|$.

O Algoritmo de Chase Estocástico utilizado neste trabalho será descrito logo abaixo. Ele foi modificado em relação ao algoritmo original descrito em [5] devido a problemas de funcionamento. Em seguida, cada etapa do algoritmo será detalhada.

i. Para $1 \leq i \leq n$,
   - Se $p_i \leq 0{,}5 - \theta$, faça $p_i = 0$, onde $0 < \theta < 0{,}5$,
   - Caso contrário se $p_i \geq 0{,}5 + \theta$, faça $p_i = 1$,
   - Caso contrário, faça $p_i = 1/(1 + e^{\beta r_i})$.

ii. Para $1 \leq m \leq \tau$,
   - Para $1 \leq i \leq n$,
     - Gere um valor aleatório uniformemente distribuído $s_i \in [0{,}1]$,
     - Gere $y_i^m = \{0, se\ s_i \geq p_i;\ 1, caso\ contrário\}$.
   - Faça a decodificação abrupta pelo algoritmo de Berlekamp-Massey no y^m para obter y, se válido, e insira y no conjunto Λ.

iii. A palavra-código decodificada será a palavra de Λ que minimiza a distância Euclidiana entre a sua versão decodificada (Λ_j) e r, análoga ao passo (iv) do CA. A

palavra-código escolhida será aquela entre as palavras-código válidas de Λ que maximiza a correlação com o vetor r.

Primeiramente, é necessário fazer uma análise acerca do comportamento da função P_r. A imagem da função opera no intervalo de 0 a 1, dependendo do valor r_i de entrada. Isto é, caso $r_i > 0$, p_i varia de 0 a 0,5; caso contrário, p_i varia de 0,5 a 1. É possível provar também que quando $r_i \to \infty \Rightarrow p_i \to 0$ e quando $r_i \to -\infty \Rightarrow p_i \to 1$.

O passo (i) previne os *bits* mais confiáveis de serem invertidos de acordo com um limiar θ, definido no intervalo aberto entre 0 e 0,5. Uma vez que quanto mais próximo de 0 for o valor de r_i, seja positivo ou negativo, mais próximo de 0,5 será o valor de p_i, isto é, quanto menos confiável for o *bit*, maior a diferença do p_i deste *bit* para 0 e 1. Um valor de θ de maior módulo representa uma maior quantidade de *bits* que não serão invertidos, enquanto um valor menor de θ representa uma maior quantidade de *bits* que podem ser invertidos. O parâmetro β é uma constante positiva que precisa ser otimizada para cada código corretor de erros. O impacto de β no algoritmo tem relação quanto ao intervalo em que a diferença entre dois valores de r_i realmente gera uma diferença entre dois valores de p_i. A Tabela 4 mostra o impacto de diferentes valores de β na geração de p_i.

Tabela 4 - Valores de p_i em relação à variação de β para vários valores de r.

		β				
		2	4	6	10	20
r	2	0,01798621	0,00033535	6,14417E-06	2,06115E-09	4,24835E-18
	1	0,119202922	0,01798621	0,002472623	4,53979E-05	2,06115E-09
	0,5	0,268941421	0,119202922	0,047425873	0,006692851	4,53979E-05
	0,1	0,450166003	0,40131234	0,354343694	0,268941421	0,119202922
	0,05	0,475020813	0,450166003	0,425557483	0,377540669	0,268941421
	-0,05	0,524979187	0,549833997	0,574442517	0,622459331	0,731058579
	-0,1	0,549833997	0,59868766	0,645656306	0,731058579	0,880797078
	-0,5	0,731058579	0,880797078	0,952574127	0,993307149	0,999954602
	-1	0,880797078	0,98201379	0,997527377	0,999954602	0,999999998
	-2	0,98201379	0,99966465	0,999993856	0,999999998	1

O aumento de β torna a região em torno de 0 mais sensível à variação do valor real, ao mesmo tempo em que restringe essa mesma região. Isto é, enquanto a diferença entre $r_i = 0{,}1$ e $r_i = 0{,}05$ é em torno de $0{,}024$ para $\beta = 2$, para $\beta = 4$ a diferença se torna $0{,}048$. Já para os valores reais $r_i = 1$ e $r_i = 0{,}5$, as diferenças são em torno de $0{,}149$ e $0{,}101$ para

$\beta = 2$ e $\beta = 4$, respectivamente. Note que a confiabilidade de valores simétricos de r_i é igual para cada método de se calcular.

No passo (ii), τ palavras-códigos candidatas são geradas de acordo com a probabilidade p_i de cada *bit*, de modo que os *bits* menos confiáveis têm maior chance de serem invertidos. Os únicos *bits* que não tem possibilidade de serem invertidos são os *bits* quer tiveram p_i igual a 0 ou igual a 1. Todos os outros terão uma probabilidade, alta ou baixa, de serem invertidos, ultrapassando a limitação apresentada pelo Algoritmo de Chase, onde apenas um pré-determinado número de posições tinham seus *bits* invertidos. O passo (iii) finaliza o algoritmo selecionando a palavra-código que maximiza a correlação entre o vetor r e a palavra-código candidata.

Os resultados do Algoritmo de Chase Estocástico apresentados em [5] para códigos Reed-Solomon mostram que o SCA é capaz de ultrapassar os limites do CA, diminuindo tanto as taxas de erros quanto a complexidade. Segundo [5], para fazer uma comparação justa acerca do desempenho dos dois algoritmos é necessário que a condição $\tau = 2^P = 1024$ seja seguida. A Figura 3 mostra os resultados disponibilizados em [5]. Este é um aprimoramento em relação à condição descrita na limitação (1) do CA.

A Figura 3 apresenta também os resultados para o SCA com $\tau = 512$. Cada algoritmo mostrado na figura a seguir é segue o modelo ***Algoritmo(quantidade de padrões de testes)***.

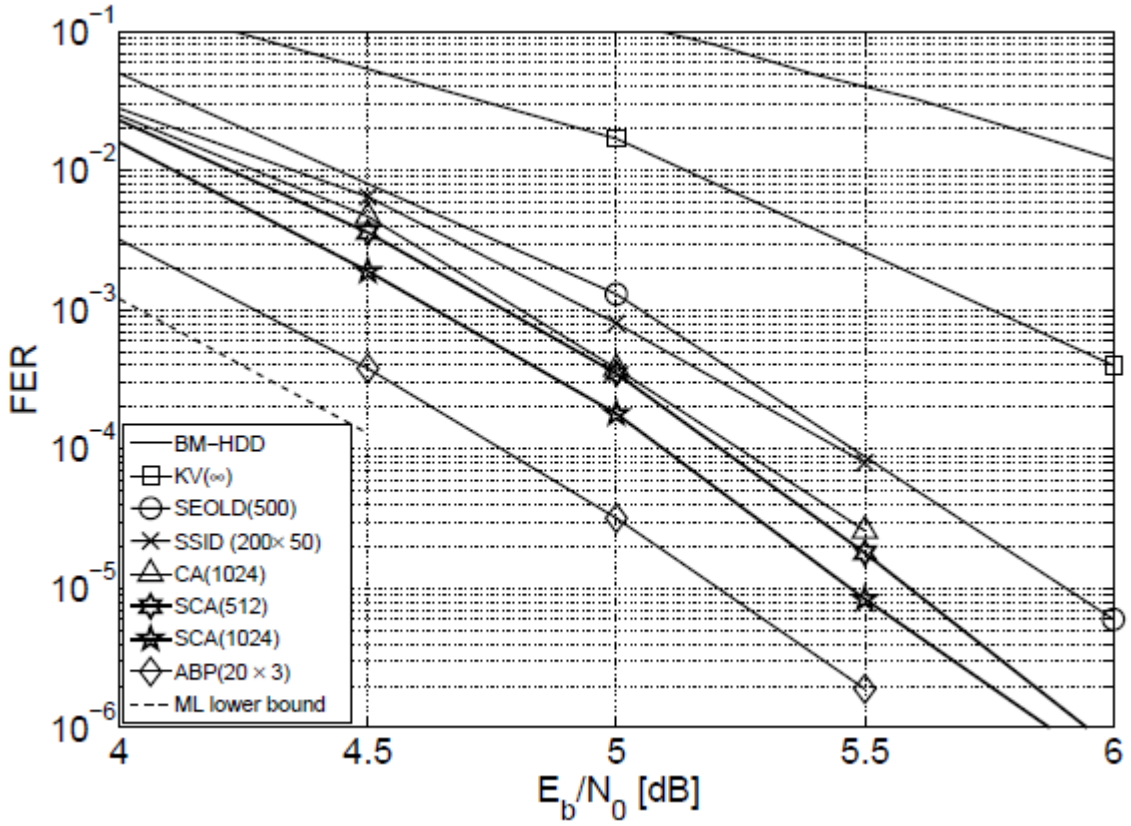

Figura 3 - Gráfico comparativo entre o Algoritmo de Chase e o Algoritmo de Chase Estocástico para códigos RS(31,25) com β = 6 e θ = 0,45. Retirado de [5].

Outra comparação foi feita para verificar quantos padrões de testes são realmente necessários para que os desempenhos se igualem. Os resultados de [5] são mostrados na Tabela 5. Os códigos RS são análogos aos códigos BCH, com a exceção de não serem binários.

Tabela 5 - Número de padrões de teste usados para atingir o mesmo desempenho. Retirado de [5].

Código	CA	SCA	SNR (dB) @ FER = 10^{-4}
RS(31,25)	1024	402	5,2
RS(63,55)	1024	538	5,65
RS(255,239)	1024	549	6,4

Os resultados mostram que para o código RS(31,25) enquanto o CA atinge o desempenho de $5{,}2\ 10^{-4}$ em um dado nível de relação sinal-ruído utilizando 1024 padrões de testes, o SCA atinge o mesmo desempenho no mesmo nível de relação sinal-ruído com 402 padrões de testes. Esta é uma melhora na limitação relativa à complexidade apresentada no item (2).

4. RESULTADOS

Neste capítulo serão apresentados os resultados e análises das simulações do Algoritmo de Chase no código de Golay Estendido e dos Algoritmos de Chase e Chase Estocástico em diferentes códigos BCH.

4.1. Código de Golay Estendido

Nesta seção, serão mostrados os resultados da aplicação do Algoritmo de Chase na decodificação do código de Golay estendido, analisando sua melhora com três valores distintos de P (parâmetro do Algoritmo de Chase).

As simulações foram feitas realizando a transmissão de 80.000 mensagens, o que totaliza 960.000 *bits* de mensagem. Foram realizadas 3 simulações com a utilização do Algoritmo de Chase variando o parâmetro P com valores P = 3, P = 4 e P = 5, além de uma simulação do código de Golay estendido sem o Algoritmo de Chase, isto é, utilizando somente a decodificações abrupta. Os resultados das simulações estão apresentadas na Figura 4.

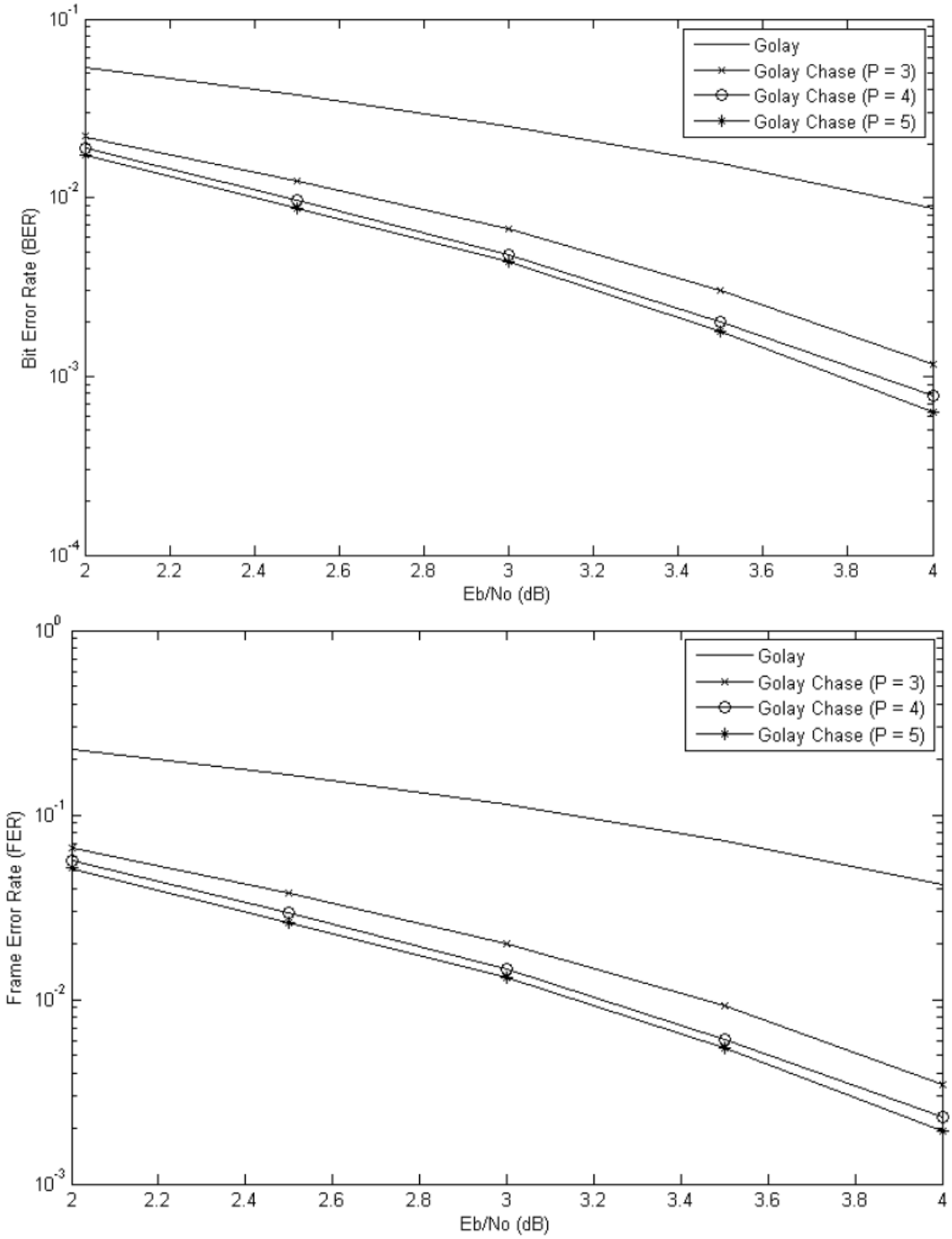

Figura 4 - BER e FER versus relação sinal-ruído para código de Golay estendido (24,12,8).

Como mostrado na Figura 4, a utilização do Algoritmo de Chase com P = 3 mostra um ganho de aproximadamente 1.7 dB, tomando como base uma FER de 7%. Tomando como base o valor de 2 dB, a mesma figura mostra uma melhora de desempenho de cerca de

3% em relação ao BER e cerca de 16% em relação ao FER. O aumento do valor da relação sinal-ruído diminui a diferença entre a presença e a ausência do CA.

Outro ponto mostrado na Figura 4 é que o aumento do parâmetro P resulta em melhora do desempenho, entretanto a complexidade computacional cresce exponencialmente em relação a P. Dependendo da aplicação, o custo adicional de equipamentos para se utilizar um valor maior de P não seja pago com o ganho extra de desempenho.

4.2. Código BCH (31,16,7), Código BCH (63,45,7) e Código BCH (63,30,13)

Nesta seção, serão mostrados os resultados da aplicação do Algoritmo de Chase em três códigos BCH distintos: BCH (31,16,7), BCH (63,45,7) e BCH (63,30,13). Vamos analisar a melhora da utilização do Algoritmo de Chase com dois valores distintos de P (parâmetro do Algoritmo de Chase), além de comparar com os códigos sem utilizar tal algoritmo. Serão feitas também duas comparações acerca do desempenho entre códigos de capacidade corretora iguais [BCH (31,16,7) e BCH (63,45,7)] e entre códigos de mesmo tamanho com capacidade corretora diferentes [BCH (63,45,7) e BCH (63,30,13)].

Inicialmente, para haver justiça na comparação entre códigos de comprimentos diferentes, alguns detalhes devem ser abordados. Foram feitas duas simulações distintas para mostrar os resultados. Na primeira, foram realizadas as transmissões de aproximadamente 800.000 *bits* de mensagem para que possamos comparar a BER dos três códigos BCH de maneira justa. Na segunda, foram realizadas as transmissões de 40.000 mensagens para que se possa ser feita uma comparação justa entre a FER dos dois códigos. Os resultados são mostrados na Figura 5.

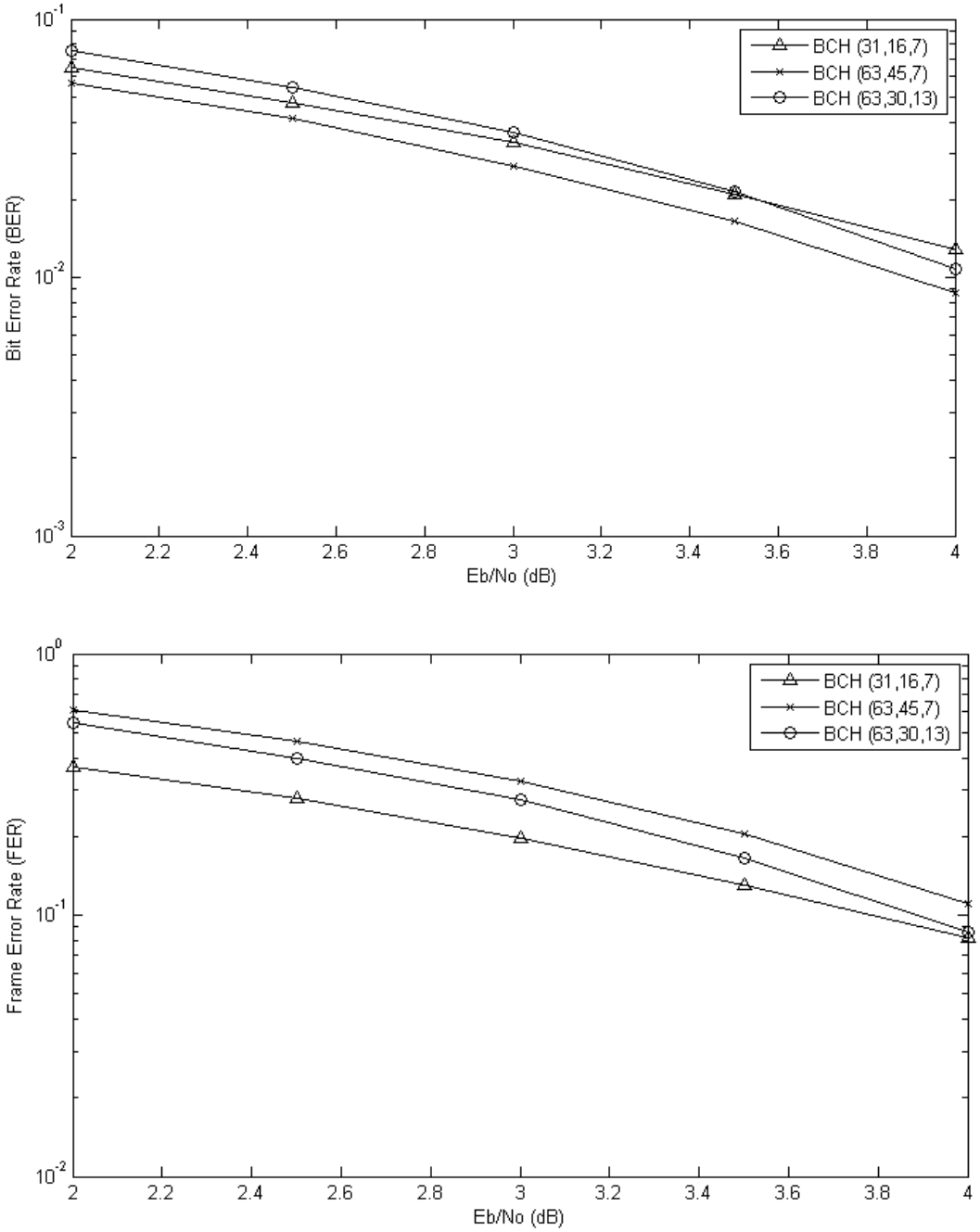

Figura 5 - Comparação entre BER e FER versus relação sinal-ruído para códigos BCH (31,16,7), BCH (63,45,7) e BCH (63,30,13).

A Figura 5 mostra um resultado interessante, pois, ao compararmos os códigos BCH (31,16,7) e BCH (63,45,7), enquanto o primeiro tem um desempenho melhor quando comparado a FER, o código BCH (63,45,7) tem desempenho melhor em relação à BER. Isto se deve ao fato de que como a capacidade corretora do código é a mesma ($t = 3$), uma palavra de comprimento maior, tem uma probabilidade maior de ter 4 ou mais *bits* invertidos.

Em relação ao BER, ainda que a palavra decodificada seja incorreta, a maior parte dos *bits* se mantém correta (maior proporção entre *bits* corretos e *bits* totais) e por ter o tamanho da mensagem na palavra-código maior, o BCH (63,45,7) apresenta melhor desempenho.

Ao compararmos os códigos BCH (63,45,7) e BCH (63,30,13) percebemos que a FER do código BCH (63,45,7) é maior que a do código BCH (63,30,13). Isto ocorre pelo simples fato de a probabilidade de se ter 4 erros ou mais inseridos pelo canal de comunicação na palavra-código enviada é maior que a probabilidade de terem sido inseridos 6 erros ou mais. Entretanto, observa-se que o código BCH (63,45,7) continua com a menor taxa de erros de *bits*, pois tendo 45 *bits* de mensagem, ainda que ocorram erros, a maior parte da mensagem continuará correta, resultando em uma taxa alta de acertos (e, consequentemente, uma taxa baixa de erros). É factível, porém, que um código com uma taxa de código menor consiga um desempenho melhor em relação ao código BCH (63,45,7) desde que a capacidade corretora do código compense os *bits* de mensagem a menos.

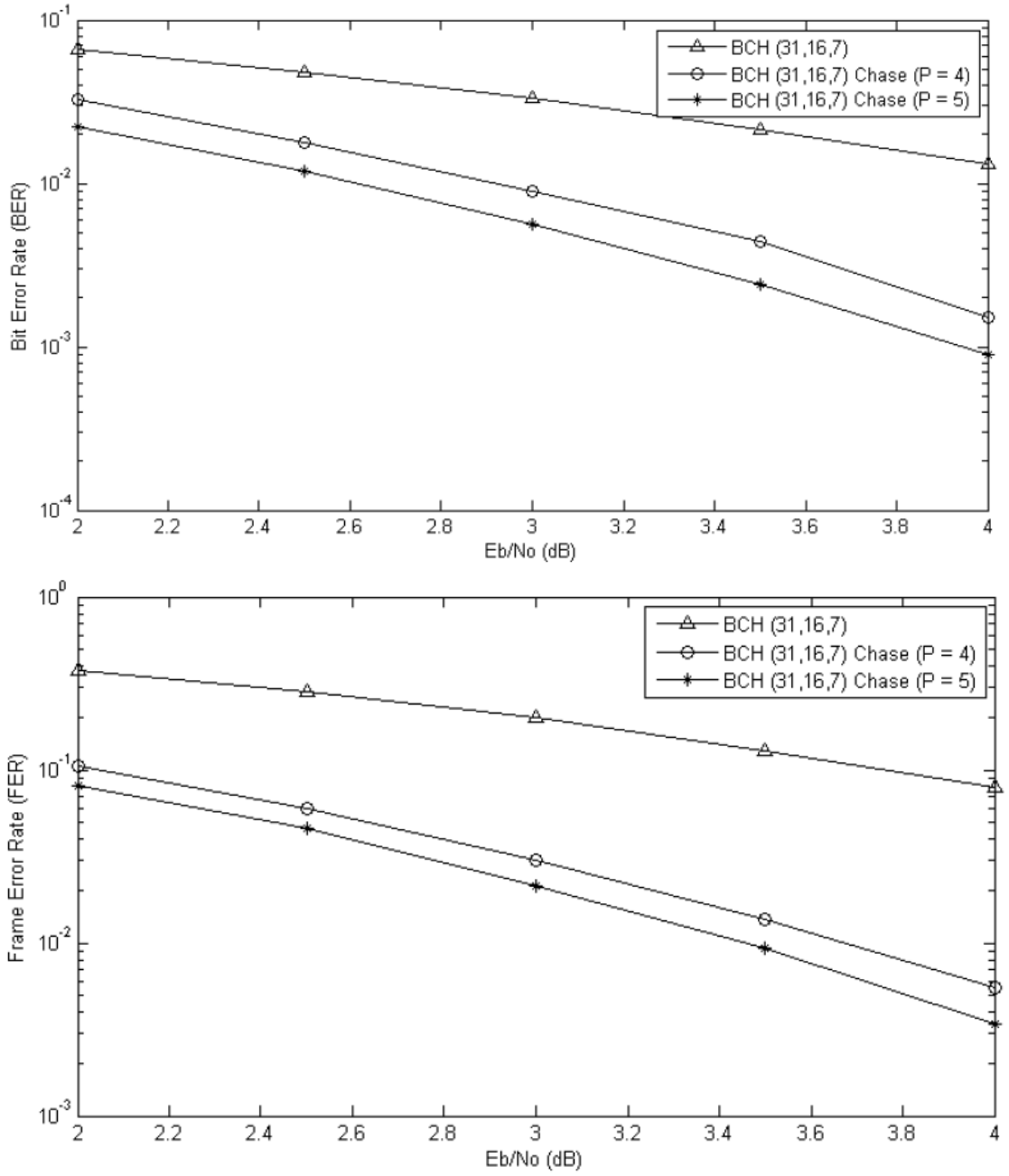

Figura 6 - BER e FER versus relação sinal-ruído para código BCH (31,16,7).

Para as comparações entre o código com decodificação abrupta e com a utilização do Algoritmo de Chase foram feitas 40.000 transmissões para cada caso. Os resultados mostram que assim como para o código de Golay estendido apresentado na seção 4.1, a utilização do Algoritmo de Chase apresenta melhora significativa ao ser aplicado na decodificação do código BCH (31,16,7), segundo a Figura 6. A utilização do CA com $P = 4$ apresenta um ganho de aproximadamente 1.1 dB para uma BER de 3.2%. Em relação à melhora de desempenho para uma dada relação sinal-ruído, em 2 dB, temos uma melhora de desempenho

de 3.3% a menos de erros. Novamente, é mostrado que o aumento do valor de P resulta em melhora de desempenho.

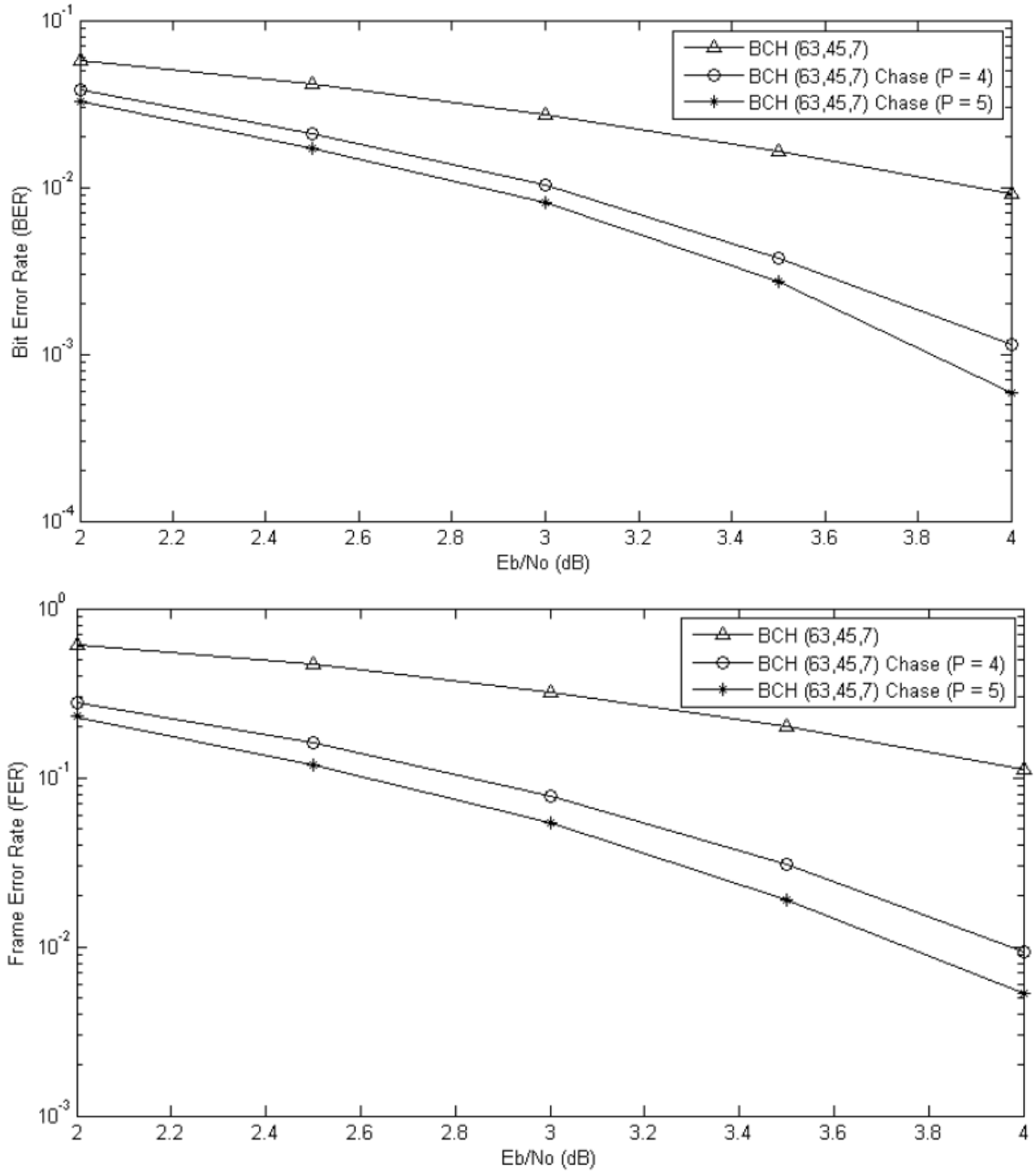

Figura 7 - BER e FER versus relação sinal-ruído para código BCH (63,45,7).

Na Figura 7, observa-se a melhora de desempenho ao introduzir o Algoritmo de Chase em um código BCH (63,45,7). A introdução do CA com $P = 4$ resulta em uma melhora de, aproximadamente, 0.6 dB para uma BER de 3.8%. Fixando a relação sinal-ruído em 2 dB, a melhoria é de 1.8% em relação à taxa de erros de *bits*.

Os resultados apresentados na Figura 8 mostram que a inserção do Algoritmo de Chase no código BCH (63,30,13) apresenta uma melhora significativa no sistema. A utilização do CA com $P = 4$ incrementa o desempenho do código em, aproximadamente, 3% quando fixo uma relação sinal-ruído de 2.5 dB. Para uma BER de 2.3%, a utilização do CA com $P = 4$ se traduz em uma economia de, aproximadamente, 0.8 dB na energia do sinal.

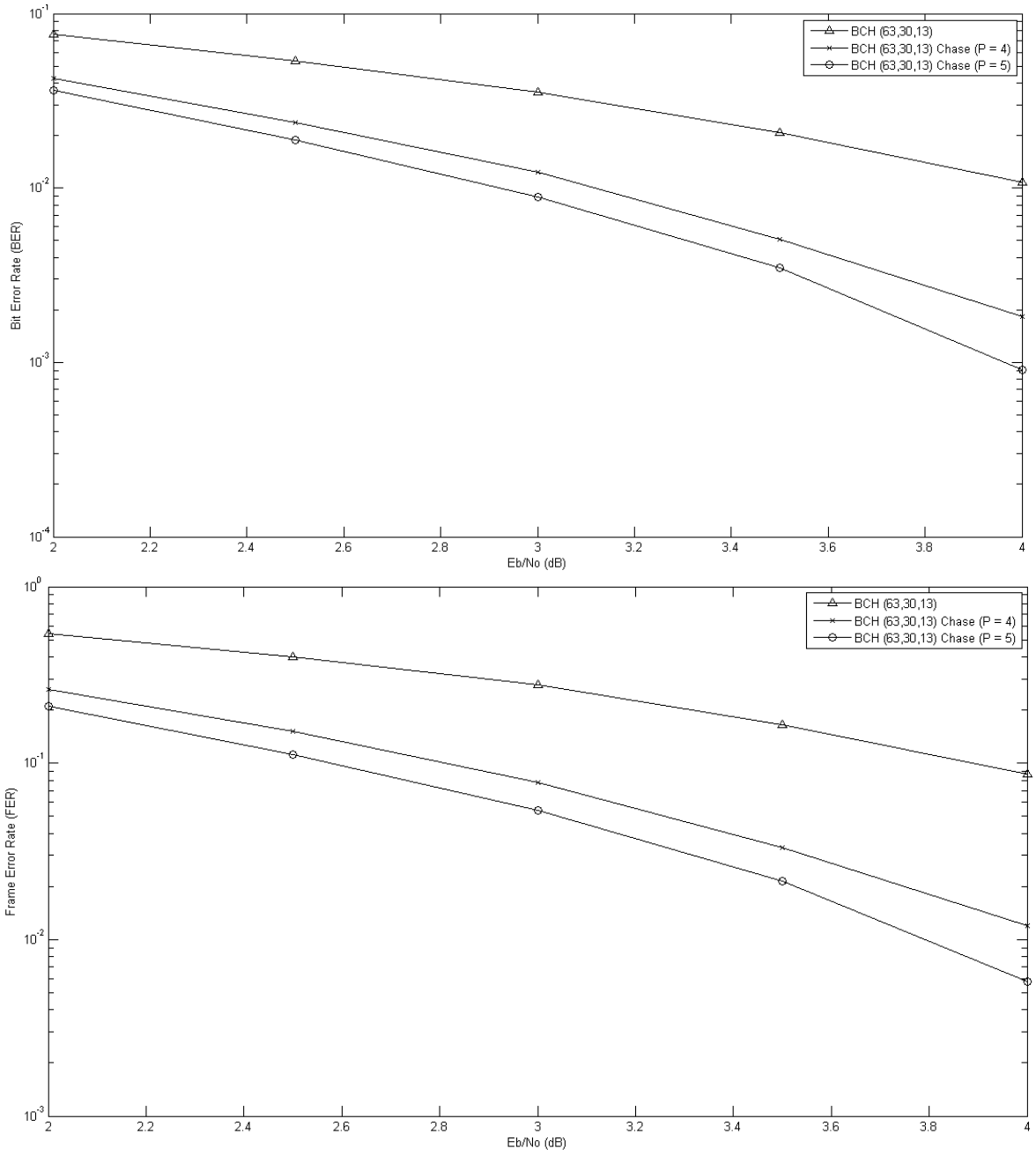

Figura 8 - BER e FER versus relação sinal-ruído para código BCH (63,30,13).

Nota-se que em todos os códigos, o maior ganho é em relação à FER: no código BCH (31,16,7), há um ganho de quase 2 dB para uma FER de 10%; no código BCH (63,45,7), o ganho é de, aproximadamente, 1,2 dB em relação a uma FER de 28%; no código BCH (63,30,13), o ganho é de 1,1 dB em relação a uma FER de 21%.

Comparando os gráficos da Figura 6 e Figura 7, nota-se que a melhora de desempenho do Algoritmo de Chase no código BCH (63,45,7) é menor que o melhoramento apresentado no código BCH (31,16,7). Isto se deve ao fato, já citado, de que códigos maiores tendem a inverter mais *bits* e, portanto, para um valor de P fixo, o CA tem um efeito positivo mais significativo em códigos de comprimentos menores.

Ao comparar os gráficos da Figura 7 e Figura 8, vemos que a utilização do Algoritmo de Chase tem um impacto maior no código BCH (63,30,13), ou seja, no código de maior capacidade corretora. Enquanto há uma melhora de 3.3% em 2 dB ao utilizar o CA com $P = 4$ no BCH (63,30,13), a melhora na mesma faixa de energia para o BCH (63,45,7) é de apenas 1.8%. Isto se deve ao fato de o código com maior capacidade corretora conseguir identificar e tratar mais erros, consequentemente, é capaz de ampliar o conjunto de palavras-códigos possíveis.

4.3. Código BCH (127,106,7)

Nesta seção, serão mostrados os resultados da aplicação do Algoritmo de Chase e do Algoritmo de Chase Estocástico, além do desempenho com a decodificação abrupta para o código BCH (127,106,7). Vamos analisar a melhora da utilização do Algoritmo de Chase e do Algoritmo de Chase Estocástico, comparando o desempenho de ambos algoritmos. Todos os resultados apresentados utilizaram o valor de $\beta = 6$ e $\theta = 0.45$ como padrão, não significando que estes foram os parâmetros ótimos.

Inicialmente foram feitos testes da utilização do Algoritmo de Chase Estocástico em códigos de comprimentos menores, como o código de Golay estendido e códigos BCH com comprimentos iguais a 31 e 63 *bits*. Entretanto, os resultados obtidos não foram favoráveis, mostrando uma limitação não descrita em [5]. Outra observação importante é que os resultados que mostraram valores comparáveis do SCA em relação ao CA só foram obtidas no código BCH (127,106,7) quando comparados o SCA e CA com valores altos para o número de padrões de testes. Quando utilizados valores baixos, o CA se mostrou mais eficiente devido à pequena chance dada ao SCA de gerar padrões de testes "ótimos".

Para haver justiça nas comparações entre o CA e o SCA, foram utilizados, primeiramente, quantidade iguais de padrões de testes, isto é, $2^P = \tau = 1024$. A Figura 9 mostra os resultados das simulações.

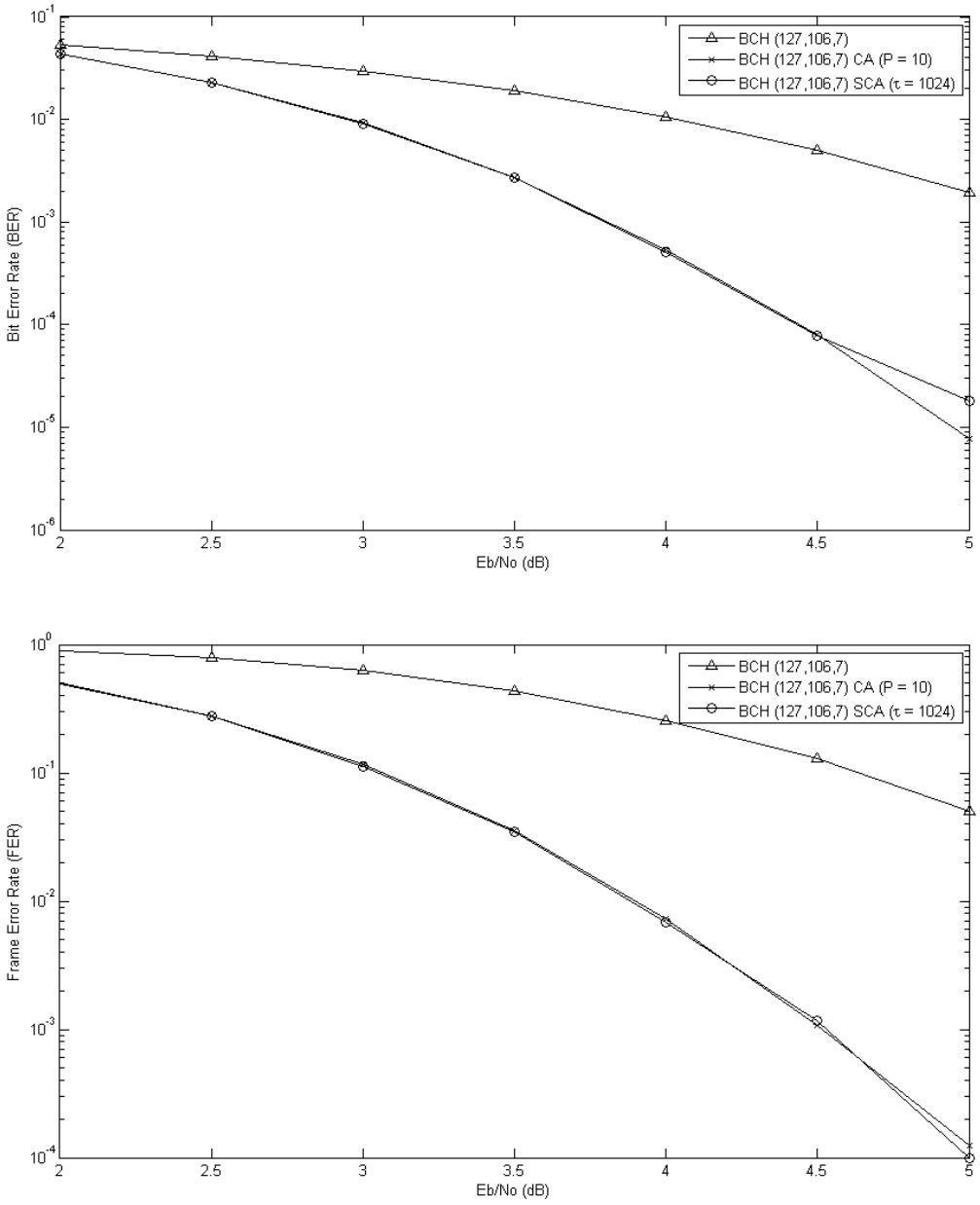

Figura 9 - BER e FER versus relação sinal-ruído para código BCH (127,106,7).

A Figura 9 mostra que o desempenho de ambos os códigos são similares quando a condição de número total de padrões de testes iguais é satisfeita. Este fato pode levar a crer que a utilização do SCA é desnecessária, pois não há ganho de desempenho.

A Figura 10 apresenta os resultados das simulações do SCA para $\tau = 512$ e para $\tau = 1024$ com o código BCH (127,106,7).

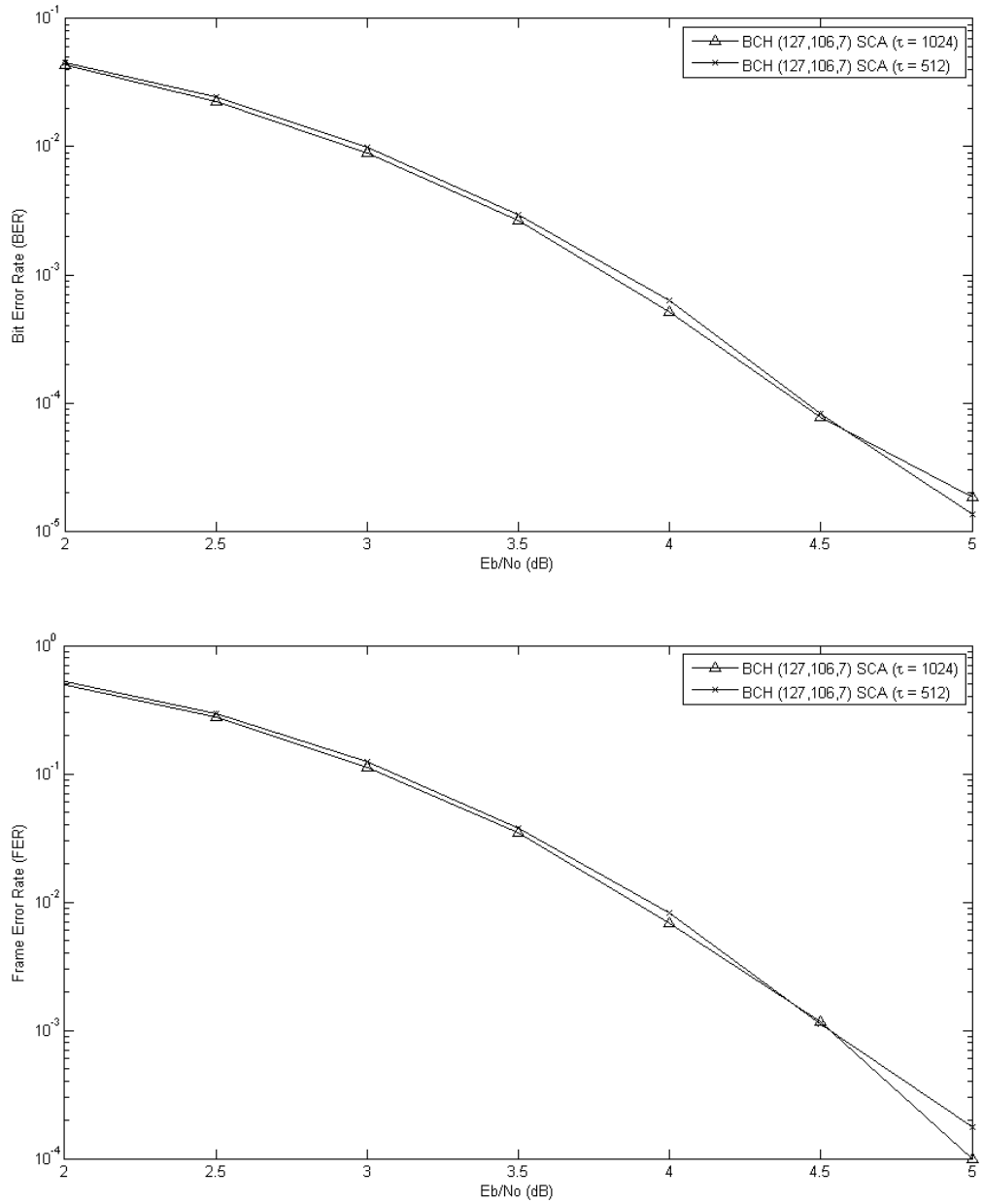

Figura 10 - BER e FER versus relação sinal-ruído para Algoritmo de Chase Estocástico com 512 e 1024 padrões de testes para código BCH(127,106,7).

Os resultados da Figura 10 mostram que para este código com o SCA, os desempenhos com $\tau = 512$ e $\tau = 1024$ são semelhantes. Em relação à complexidade de cada algoritmo, o SCA com $\tau = 512$ tem resultado mais favorável que o SCA com $\tau = 1024$, pois apresenta uma complexidade menor. A Figura 11 apresenta os valores de padrões de testes repetidos gerados pelo SCA (512) e pelo SCA (1024).

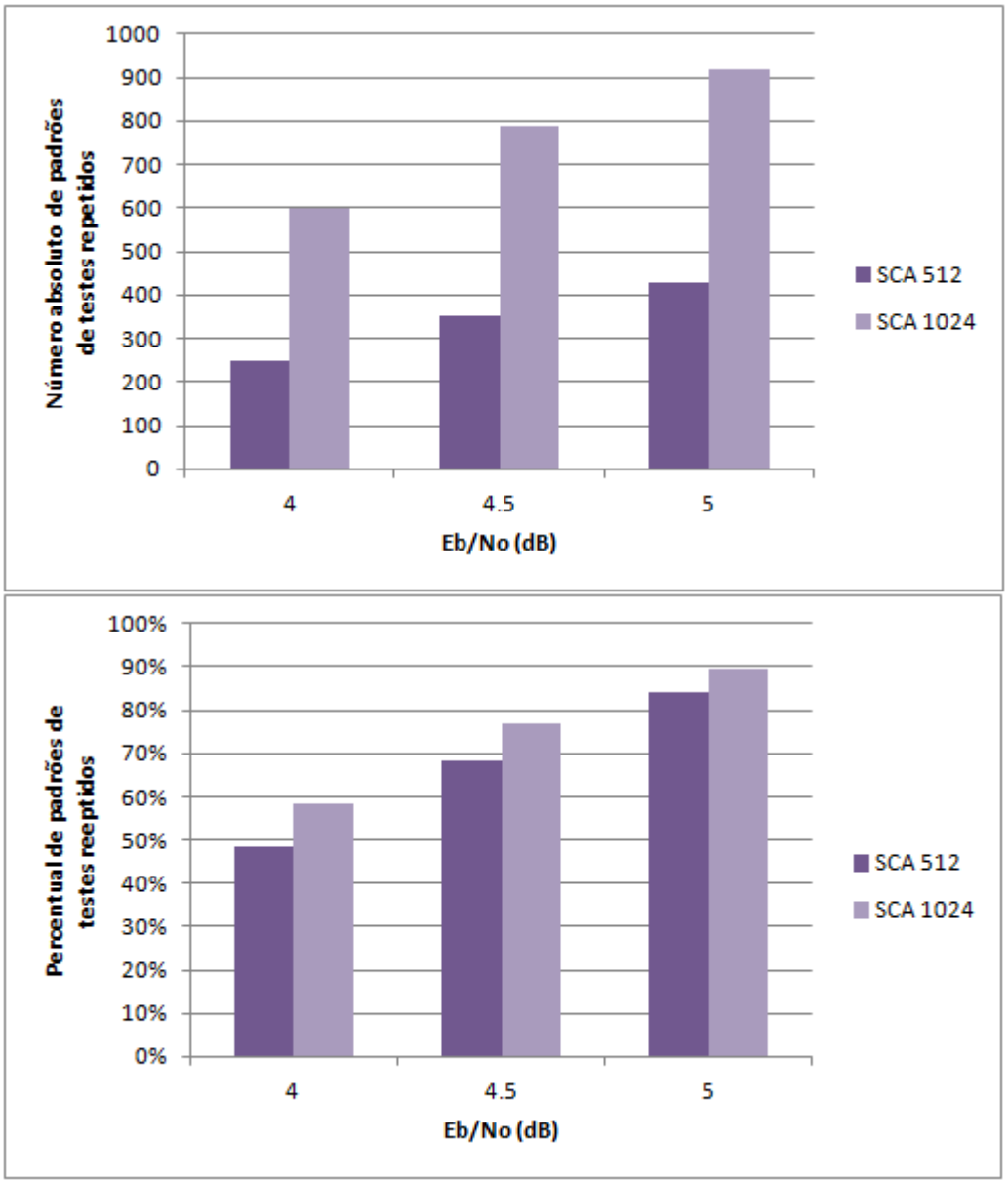

Figura 11 - Quantidade de padrões de testes repetidos do Algoritmo de Chase Estocástico para BCH (127,106,7).

A Figura 11 mostra que existe uma boa parcela de padrões de testes repetidos. Em um nível alto de relação sinal-ruído, os padrões de testes do SCA (512) é composto por pouco mais de 80% de padrões repetidos, enquanto o SCA (1024) chega a ser composto de quase 90% de padrões de testes iguais. O Algoritmo de Chase Estocástico descrito em [5] não menciona como tratar essa parcela de padrões de testes repetidos.

Apesar dos desempenhos em relação às taxas de erros serem próximas, a menor quantidade de padrões de testes distintos presente no Algoritmo de Chase Estocástico mostra que o mesmo tem uma complexidade menor. Ou seja, obrigando o SCA a gerar somente

padrões de testes distintos com 1024 iterações, teremos um conjunto de padrões de testes quase 90% menor para chegarmos ao mesmo desempenho em relação aos erros apresentados pelo CA.

5. CONCLUSÕES E TRABALHOS FUTUROS

A área de códigos corretores de erros tem ganho muito destaque no meio acadêmico recentemente devido à sua importância para os sistemas de comunicação digital. Neste cenário, o campo de aplicação de algoritmos de decodificação por decisão suave tem estado em um nível de grande importância pelos resultados apresentados até o momento, pois enquanto o mercado procura diminuir custos energéticos, também se busca aumentar o desempenho dos sistemas de comunicação.

Este trabalho teve por objetivo mostrar que a aplicação de algoritmos de decodificação por decisão suave tem um enorme impacto positivo em códigos BCH, antes só implementado juntamente a códigos RS, diminuindo a energia gasta para atingir uma dada porcentagem de erros ou melhorando o desempenho para um dado valor de energia gasto.

Diferentemente do Algoritmo de Chase, um algoritmo bem fundamentado e estudado, o Algoritmo de Chase Estocástico é relativamente novo e possui vários pontos que podem ser melhorados, além das modificações já feitas neste trabalho, como um método de identificar previamente quais os melhores valores de β e θ para um dado código, por exemplo. Outros pontos do SCA ainda precisam ser melhores definidos, como, por exemplo, otimizar o conjunto de padrões de testes quando há repetições, podendo ignorar o padrão repetido, diminuindo a complexidade computacional do algoritmo.

Ainda assim, os resultados mostraram que, embora o desempenho do SCA em relação aos erros de *bits* e mensagens se mantenha muito próximo ao desempenho do CA, sua complexidade é consideravelmente menor, o que já se mostra uma grande diferencial.

Como trabalho futuro, pretende-se estudar otimizações do Algoritmo de Chase Estocástico para os problemas citados anteriormente diminuindo a complexidade, além de analisar o impacto deste algoritmo em códigos de utilização real na indústria.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] C. E. Shannon, “A Mathematical Theory of Communication,” Bell Syst. Tech. J., v. 27, 379-423, 1948.

[2] R.W. Hamming, “Error Detecting and Error Correcting Codes,” Bell Syst. Tech. J., v. 29, 147-160, 1950.

[3] D. Chase, “A Class of Algorithms for Decoding Block Codes with Channel Measurement Information,” IEEE Trans on Inf. Theory, v. IT-18, 170-182, 1972.

[4] M. Fossorier and S. Lin, “Soft-Decision Decoding of Linear Block Codes Based on Ordered Estatistics,” IEEE Trans. on Inf. Theory, v. IT-41, 1379-1396, Set 1995.

[5] C. Leroux et. al., “Stochastic Chase Decoding of Reed-Solomon Codes,” IEEE Commun. Letters, v. 14, n. 9, 863-865, 2010.

[6] Renato Machado. Disponível em: < http://coral.ufsm.br/gpscom/professores/Renato%20Machado/TopicosAvancados/TopAvanTelecom02.pdf>. Acesso em: 26 jun. 2014.

[7] Golay, Marcel J. E. (1949). "Notes on Digital Coding". *Proc. IRE* 37: 657.

[8] D. Castello and S. Lin, “Error Control Coding: Fundamentals and Applications: 1st Edition”, Prentice Hall Professional Technical Reference, 1983.