

MICHELLE FOLTRAN MIRANDA

**AVALIAÇÃO DE CÓDIGOS CORRETORES DE ERRO PARA
COMUNICAÇÃO DE DADOS EM OFDM NA REDE ELÉTRICA**

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre, no Programa de Pós-Graduação em Engenharia Elétrica, Setor de Tecnologia, Universidade Federal do Paraná UFPR.–

Orientador: Eduardo Parente Ribeiro, Dr.

Co-orientador: Walter Godoy Jr., Dr.

CURITIBA

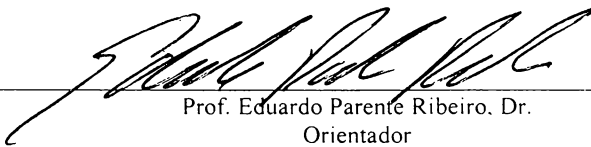
2004

AVALIAÇÃO DE CÓDIGOS CORRETORES DE ERRO PARA COMUNICAÇÃO DE DADOS EM OFDM NA REDE ELÉTRICA

MICHELLE FOLTRAN MIRANDA

Dissertação aprovada como requisito parcial para obtenção do grau de Mestre no
Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do

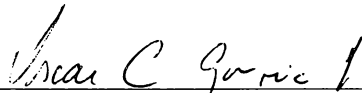
Paraná



Prof. Eduardo Parente Ribeiro, Dr.
Orientador

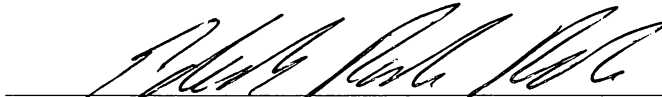


Prof. Walter Godoy Jr., Dr.
Co-orientador

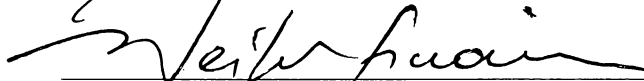


Prof. Oscar da Costa Gouveia Filho, Dr.
Coordenador do Programa de Pós-Graduação em Engenharia Elétrica

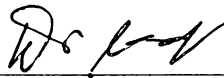
Banca Examinadora



Prof. Eduardo Parente Ribeiro, Dr., Presidente



Prof. Weiler Finamore, Dr. (PUC-Rio)



Prof. Walter Godoy Jr., Dr. (CEFET-PR)



Prof. Marcus Vinicius Lamar, Dr. (UFPR)

Curitiba, 30 de abril de 2004

Aos meus pais, Sonia e Sérgio, e ao Robson pelo amor, compreensão, apoio e incentivo na conquista de meus objetivos.

AGRADECIMENTOS

Ao Prof. Dr. Eduardo Parente Ribeiro por seus ensinamentos e conselhos.

Ao Prof. Dr. Walter Godoy Jr. que contribuiu para a elaboração desse trabalho através de seu conhecimento e de suas sugestões.

A todos aqueles que me ajudaram e forneceram discussões e dicas valiosas. Em especial, aos amigos: Vilson Rodrigo Mognon, Alberto Yassufumi Natori, Salmo Pustilnick, Marcelo Naegele e Igor Gavriloff.

À minha família, meus pais, irmãs, avós e tio, pelo apoio e pelas palavras de incentivo.

Aos professores e funcionários do Departamento de Engenharia Elétrica da Universidade Federal do Paraná.

SUMÁRIO

| | |
|---|-------------|
| AGRADECIMENTOS | III |
| LISTA DE ACRÔNIMOS | V |
| LISTA DE FIGURAS | VI |
| RESUMO | VII |
| ABSTRACT | VIII |
| 1 INTRODUÇÃO | 9 |
| 2 O SISTEMA OFDM | 12 |
| 2.1 HISTÓRICO | 13 |
| 2.2 VANTAGENS E DESVANTAGENS DO SISTEMA OFDM..... | 15 |
| 2.3 PRINCÍPIOS DO SISTEMA OFDM..... | 15 |
| 3 CÓDIGOS CORRETORES DE ERRO | 18 |
| 3.1 TIPOS DE CÓDIGOS DE CANAL..... | 19 |
| 3.2 HISTÓRIA DOS CÓDIGOS CONTROLADORES DE ERRO..... | 20 |
| 3.3 APLICAÇÕES DOS CÓDIGOS CORRETORES DE ERRO | 21 |
| 3.4 CÓDIGOS DE BLOCO..... | 22 |
| 3.4.1 DEFINIÇÕES BÁSICAS..... | 22 |
| 3.4.2 CAPACIDADE DE DETECÇÃO E CORREÇÃO DE ERRO..... | 23 |
| 3.5 CÓDIGOS DE REED-SOLOMON..... | 24 |
| 3.6 CÓDIGOS CONVOLUCIONAIS..... | 25 |
| 3.7 CÓDIGOS TURBO..... | 27 |
| 3.7.1 CODIFICADOR TURBO PADRÃO | 29 |
| 3.7.2 PRINCÍPIO DO DECODIFICADOR TURBO PADRÃO | 30 |
| 3.7.3 DECODIFICAÇÃO ÓTIMA COM DECISÃO SUAVE..... | 32 |
| 3.7.4 A INFORMAÇÃO EXTRÍNSECA DO DECODIFICADOR RSC..... | 36 |
| 3.7.5 DECODIFICADOR TURBO PADRÃO COM REALIMENTAÇÃO..... | 37 |
| 3.8 SIMULAÇÃO DE SISTEMAS COM CONTROLE DE ERRO..... | 40 |
| 4 METODOLOGIA | 43 |
| 4.1 SISTEMA DE TRANSMISSÃO OFDM SEM CODIFICAÇÃO | 43 |
| 4.2 CÓDIGOS DE REED-SOLOMON EM SISTEMAS OFDM | 44 |
| 4.3 CÓDIGOS CONVOLUCIONAIS EM SISTEMAS OFDM | 45 |
| 4.4 CÓDIGOS TURBO EM SISTEMAS OFDM | 48 |
| 4.5 SIMULAÇÕES | 50 |
| 4.6 FLUXOGRAMA DA METODOLOGIA..... | 53 |
| 5 RESULTADOS | 56 |
| 6 CONSIDERAÇÕES FINAIS E CONCLUSÕES | 64 |
| 6.1 CONSIDERAÇÕES FINAIS..... | 64 |
| 6.2 CONCLUSÕES..... | 65 |
| 6.3 TRABALHOS FUTUROS..... | 66 |
| REFERÊNCIAS BIBLIOGRÁFICAS | 67 |
| ANEXO I | 70 |
| ANEXO II | 73 |
| ANEXO III | 76 |

LISTA DE ACRÔNIMOS

| Sigla | Definição | Significado |
|--------------|---|--|
| ADSL | <i>Asymmetric Digital Subscriber Lines</i> | Linhas de Assinantes Digitais e Assimétricas |
| APP | <i>A Posteriori Probability</i> | Probabilidade a Posteriori |
| ARQ | <i>Automatic Repeat Request</i> | Solicitação de Repetição Automática |
| AWGN | <i>Additive White Gaussian Noise</i> | Ruído Gaussiano Branco Aditivo |
| BCH | <i>Bose, Chaudhuri and Hocquenghem Codes</i> | Códigos de Bose, Chaudhuri e Hocquenghem |
| BPSK | <i>Binary Phase Shift Keying</i> | Chaveamento por Deslocamento de Fase |
| DAB | <i>Digital Audio Broadcasting</i> | Transmissão de Áudio Digital |
| DFT | <i>Discrete Fourier Transform</i> | Transformada de Fourier Discreta |
| DMT | <i>Discrete Multitone</i> | Multitom Discreto |
| FDM | <i>Frequency Division Multiplexing</i> | Multiplexação por Divisão em Frequências |
| FFT | <i>Fast Fourier Transform</i> | Transformada Rápida de Fourier |
| FSK | <i>Frequency Shift Keying</i> | Chaveamento por Deslocamento em Frequência |
| GF | <i>Galois Field</i> | Campo de Galois |
| HDSL | <i>High-bit-rate Digital Subscriber Lines</i> | Linhas de Assinantes Digitais de Alta Taxa de Bit |
| HDTV | <i>High-definition Television</i> | Televisão de Alta Definição |
| IFFT | <i>Inverse Fast Fourier Transform</i> | Transformada Rápida Inversa de Fourier |
| LLR | <i>Log Likelihood Ratio</i> | Razão de Verossimilhança Logarítmica |
| MAP | <i>Maximum a Posteriori</i> | Máximo a Posteriori |
| MSK | <i>Minimum Shift Keying</i> | Chaveamento por Deslocamento Mínimo |
| OFDM | <i>Orthogonal Frequency Division Multiplexing</i> | Multiplexação por Divisão em Frequências Ortogonais |
| PAM | <i>Phase Amplitude Modulation</i> | Modulação em Amplitude de Fase |
| PLC | <i>Power Line Communication</i> | Comunicação na Rede Elétrica |
| PSK | <i>Phase Shift Keying</i> | Chaveamento por Deslocamento de Fase |
| QAM | <i>Quadrature Amplitude Modulation</i> | Modulação de Amplitude em Quadratura |
| RS | <i>Reed-Solomon Codes</i> | Códigos de Reed-Solomon |
| RSC | <i>Recursive Systematic Convolutional</i> | Convolucional Sistemático e Recursivo |
| TCM | <i>Trellis Coded Modulation</i> | Modulação Codificada em Treliza |
| TTCM | <i>Turbo Trellis Coded Modulation</i> | Modulação Codificada Turbo em Treliza |
| VDSL | <i>Very-high-speed Digital Subscriber Lines</i> | Linhas de Assinantes Digitais de Velocidade Muito Alta |
| VLSI | <i>Very Large Scale Integration</i> | Integração em Escala Muito Larga |

LISTA DE FIGURAS

| | |
|---|----|
| FIGURA 1: CONCEITO DO SINAL OFDM. (A) SEM SUPERPOSIÇÃO (B) COM SUPERPOSIÇÃO . | 13 |
| FIGURA 2: (A) ESPECTRO DE UM SUBCANAL OFDM; (B) ESPECTRO DO SINAL OFDM..... | 14 |
| FIGURA 3: MODULADOR OFDM | 16 |
| FIGURA 4: MODELO SIMPLIFICADO DE UM SISTEMA DE COMUNICAÇÃO DIGITAL..... | 18 |
| FIGURA 5: DIAGRAMA EM BLOCO DE UM CODIFICADOR CONVOLUCIONAL..... | 26 |
| FIGURA 6: CÓDIGO SISTEMÁTICO RECURSIVO | 28 |
| FIGURA 7: CODIFICADOR TURBO PADRÃO | 29 |
| FIGURA 8: CODIFICAÇÃO TURBO COM FUNÇÃO DE <i>PUNCTURING</i> | 30 |
| FIGURA 9: PRINCÍPIO DO DECODIFICADOR TURBO PADRÃO | 31 |
| FIGURA 10: DECODIFICADOR TURBO PADRÃO..... | 38 |
| FIGURA 11: SISTEMA DE TRANSMISSÃO OFDM SEM CODIFICAÇÃO..... | 43 |
| FIGURA 12: SISTEMA DE RECEPÇÃO OFDM SEM CODIFICAÇÃO | 44 |
| FIGURA 13: CODIFICADOR RS NO SISTEMA OFDM | 45 |
| FIGURA 14: DECODIFICADOR RS NO SISTEMA OFDM | 45 |
| FIGURA 15: MAPEAMENTO GRAY DE DOIS BITS EM PAM DE 4 NÍVEIS | 46 |
| FIGURA 16: RE-MAPEAMENTO DE 4 NÍVEIS PAM EM 2 MÉTRICAS | 46 |
| FIGURA 17: CODIFICADOR CONVOLUCIONAL NO SISTEMA OFDM | 47 |
| FIGURA 18: DECODIFICADOR CONVOLUCIONAL NO SISTEMA OFDM..... | 47 |
| FIGURA 19: CODIFICADOR TURBO NO SISTEMA OFDM..... | 49 |
| FIGURA 20: DECODIFICADOR TURBO NO SISTEMA OFDM | 49 |
| FIGURA 21: DENSIDADE ESPECTRAL DE POTÊNCIA DOS RUÍDOS COLORIDOS E AWGN..... | 52 |
| FIGURA 22: DENSIDADE ESPECTRAL DE POTÊNCIA DO RUÍDO MEDIDO PLC E DO AWGN..... | 52 |
| FIGURA 23: FLUXOGRAMA DA METODOLOGIA..... | 55 |
| FIGURA 24: DESEMPENHOS TEÓRICO E SIMULADO NO CANAL AWGN | 56 |
| FIGURA 25: DESEMPENHO DOS SISTEMAS PARA O CANAL COM RUÍDO AWGN..... | 57 |
| FIGURA 26: DESEMPENHO DOS SISTEMAS COM RUÍDO COLORIDO (12KHZ A 16KHZ)..... | 58 |
| FIGURA 27: DESEMPENHO DOS SISTEMAS COM RUÍDO COLORIDO (15KHZ A 16KHZ)..... | 59 |
| FIGURA 28: DESEMPENHO DOS SISTEMAS COM RUÍDO MEDIDO DA REDE ELÉTRICA | 60 |
| FIGURA 29: COMPARAÇÃO DO DESEMPENHO DOS SISTEMAS COM CONVOLUCIONAL | 61 |
| FIGURA 30: COMPARAÇÃO DO DESEMPENHO DOS SISTEMAS COM CÓDIGO TURBO | 62 |
| FIGURA 31: COMPARAÇÃO DO DESEMPENHO DOS SISTEMAS COM CÓDIGO RS..... | 63 |

RESUMO

A rede elétrica, a qual está presente em praticamente todas as construções e residências, é um meio de comunicação de dados conveniente e que não exige grandes gastos complementares quanto à infraestrutura. Todavia, devido às características variantes no tempo desse canal hostil, projetar um sistema para a transmissão de dados torna-se um problema desafiador. Um único modelo preciso para o canal da rede elétrica ainda não está completamente estabelecido, portanto avaliá-la como meio de comunicação é uma tarefa difícil. A principal contribuição desse trabalho é verificar o desempenho dos códigos corretores de erro em um ambiente de simulação utilizando amostras reais de ruído medido da rede elétrica. A multiplexação por divisão em frequência (*OFDM*) é considerada, pois é o principal método empregado nos sistemas atuais. O método de simulação de Monte Carlo foi utilizado para a obtenção das curvas de desempenho dos códigos convolucionais, de Reed-Solomon e turbo para uma transmissão por um canal com ruído medido da rede elétrica. Os desempenhos para o canal com ruído aditivo branco gaussiano (*AWGN*) também foram traçados com fins comparativos. Os resultados obtidos mostram que os códigos apresentam desempenhos variáveis quando submetidos ao ruído da rede se comparados ao desempenho em canais com ruído *AWGN*, o que confirma a necessidade de uma escolha cuidadosa do código para melhores resultados.

Palavras-chave: Sistema *OFDM*, códigos corretores de erro, códigos convolucionais, códigos de Reed-Solomon, códigos turbo, comunicação de dados, rede elétrica.

ABSTRACT

Power lines, which can be found in essentially all buildings and residences, form a convenient and inexpensive medium for data communication. However due to the time-varying characteristics of this hostile channel, designing a system for data transmission over the power lines becomes a challenging problem. A precise model for power line communication (PLC) is not available yet, so evaluate the power line as a communication channel is a difficult task. The main contribution of this work is to verify the performance of error correction codes in a simulated environment using real samples of measured power line noise. Orthogonal Frequency Division Multiplexing (OFDM) is considered because it is the main method employed on most systems. Monte Carlo method is used to estimate the performances of convolutional codes, the Reed-Solomon codes and the turbo codes on a measured power line noise. Their performances for additive white gaussian noise (AWGN) have also been plotted for comparison purposes. Results show that the codes perform differently on real noise compared to additive white gaussian noise. It confirms that a careful choice of code is necessary for best performances.

Keywords: OFDM System, error correction codes, convolutional codes, Reed-Solomon codes, turbo codes, data communication, power line network

1 INTRODUÇÃO

A utilização crescente da Internet vem criando um aumento de demanda por serviços de banda larga nas residências. Atualmente, a rede de acesso por linhas de par trançado a partir de uma unidade de rede para o cliente, a chamada última milha, é um fator limitante do tráfego da Internet. A crescente demanda por redes que estejam presentes em residências e escritórios também traz a necessidade de acessos de rede mais flexíveis.

Atualmente existem duas principais soluções para a transmissão de dados pela última milha. A primeira seria a transmissão digital por linhas telefônicas através da tecnologia de linha de assinante digital (DSL) ou por redes de cabo e a segunda seria o acesso sem fio.

A rede elétrica está presente em praticamente todas as construções e residências e está sendo considerada um meio conveniente e barato para a comunicação de dados. Em áreas onde as soluções de telefonia ou via cabo não podem ser encontradas, a comunicação através da rede elétrica torna-se uma solução vantajosa e comparável às soluções sem fio.

A tecnologia PLC (*Power Line Communication*) pode então ser utilizada para a transmissão de dados, oferecendo um barramento para formar uma grande rede de dados. Não há necessidade de instalações especiais e o sistema poderia ser acessado em qualquer ponto onde houver uma tomada de parede. Em comparação com as demais soluções, os cabos ou linhas telefônicas sofrem com os poucos pontos de conexão e as soluções sem-fio sofrem com o congestionamento e com interferências nas bandas de frequência.

Embora a rede elétrica seja um meio muito atrativo, existem algumas dificuldades na implementação de um sistema para a transmissão de dados. Originalmente designada para o fornecimento de energia, a rede elétrica possui características hostis como meio de comunicação. Várias cargas conectadas e desconectadas randomicamente na rede elétrica produzem alterações ainda mais imprevisíveis neste canal. Ao contrário de muitos outros canais de comunicação, o ruído presente na rede elétrica não pode ser descrito como um ruído gaussiano branco aditivo (AWGN).

Um outro problema encontrado pela tecnologia PLC é a quantidade excessiva de interferência eletromagnética radiada para outros sistemas. Isso poderia ser contornado reduzindo-se a potência do sinal transmitido, causando uma diminuição da relação sinal-ruído (SNR), o que leva a um aumento da taxa de erro de bit (BER). Faz-se necessária a escolha de bons códigos corretores de erro, cujos desempenhos são bem conhecidos para canais com ruído gaussiano branco aditivo (AWGN), mas não para a rede elétrica. Devido às características variantes no tempo desse canal hostil, um único modelo preciso para a rede elétrica ainda não está disponível. Dessa forma, a implementação de um sistema para transmissão de dados pela rede elétrica torna-se um problema desafiador.

Esse trabalho procurou comparar os desempenhos de alguns códigos corretores de erro em um ambiente de simulação utilizando-se ruídos sintéticos bem como amostras de ruído medido da rede elétrica. A técnica OFDM (*Orthogonal Frequency Division Multiplexing*) [1] é considerada, pois é o principal método utilizado pelos atuais modems [2] e o método de Monte Carlo é usado para estimar a curva BER *versus* SNR de um determinado código corretor de erro para o canal PLC. Os códigos convolucionais, de Reed-Solomon e turbo são investigados. As curvas de desempenho também são traçadas para o canal AWGN com fins comparativos.

Esses códigos corretores de erro podem ser utilizados em várias aplicações. Para a modulação DMT (*Discrete Multitone*) [3], [4] utilizada no padrão ADSL (*Asymmetric Digital Subscriber Lines*) [5] é adotada a codificação de Reed-Solomon.

Os códigos convolucionais também podem ser aplicados ao sistema OFDM [1]. Um código convolucional mapeia k bits de uma seqüência contínua de entrada em n bits de saída, onde o mapeamento é realizado pela convolução dos bits de entrada com uma resposta ao impulso binária.

Os códigos turbo [6-9] são uma nova classe de códigos convolucionais cujo desempenho em termos da taxa de erro de bit está próximo ao limite de Shannon. O codificador é construído utilizando uma concatenação em paralelo de dois códigos convolucionais sistemáticos recursivos e um método iterativo é adotado para a decodificação associada.

No Capítulo 2 é apresentado um histórico sobre o OFDM e são mostradas as características e vantagens mais importantes desse sistema.

O Capítulo 3 descreve os principais códigos corretores de erro: os códigos de Reed-Solomon, os códigos convolucionais e os códigos turbo.

Em seguida, no Capítulo 4 os códigos corretores de erro são associados ao sistema OFDM e a metodologia para a obtenção dos resultados é descrita.

As curvas de desempenho dos sistemas para o canal com ruído gaussiano branco aditivo (AWGN), para o canal com ruído colorido e para o canal com ruído medido da rede elétrica são mostradas no Capítulo 5.

O Capítulo 6 apresenta as conclusões obtidas e as considerações finais.

2 O SISTEMA OFDM

A multiplexação por divisão em freqüências ortogonais (OFDM) é um caso especial de transmissão com múltiplas portadoras em que uma única seqüência de dados é transmitida por um número determinado de subportadoras. O OFDM pode ser visto tanto como uma técnica de modulação como de multiplexação. Uma das principais razões para seu uso é aumentar a robustez do sistema. Em um sistema com uma única portadora, uma interferência pode causar falha em todo o enlace de dados. Já em um sistema com múltiplas portadoras, somente uma pequena porcentagem delas será afetada. Os códigos corretores de erro podem então ser utilizados para corrigir os erros nas subportadoras. O conceito de utilizar uma transmissão de dados em paralelo e a multiplexação por divisão de freqüência foi publicado em meados dos anos 60 [10,11].

Em um sistema clássico de dados em paralelo, a banda total do sinal é dividida em N subportadoras sem superposição em freqüência. Cada subcanal é modulado separadamente e então os subcanais são multiplexados em freqüência. O uso do espectro sem superposição dos canais elimina a interferência intercanal, todavia tem-se uma utilização ineficiente do meio. Como alternativa para esse problema, foi proposta a utilização de dados paralelos e do FDM em canais com superposição.

A Figura 1 mostra as diferenças entre a técnica convencional de múltiplas portadoras sem superposição (Figura 1.a) e a com superposição (Figura 1.b). Utilizando-se as múltiplas portadoras com superposição, pode-se economizar 50% da largura de banda. Todavia, para implementar essa técnica, deve-se reduzir o *crosstalk* entre as subportadoras através da propriedade de ortogonalidade.

A palavra ortogonal indica que existe uma relação matemática precisa entre as freqüências das portadoras no sistema. Em um sistema normal de multiplexação por divisão em freqüência, várias portadoras são espaçadas de tal maneira que os sinais podem ser recebidos utilizando-se filtros e demoduladores convencionais. Em tais receptores, bandas de guarda são introduzidas entre diferentes portadoras no domínio da freqüência, resultando em uma baixa eficiência espectral.

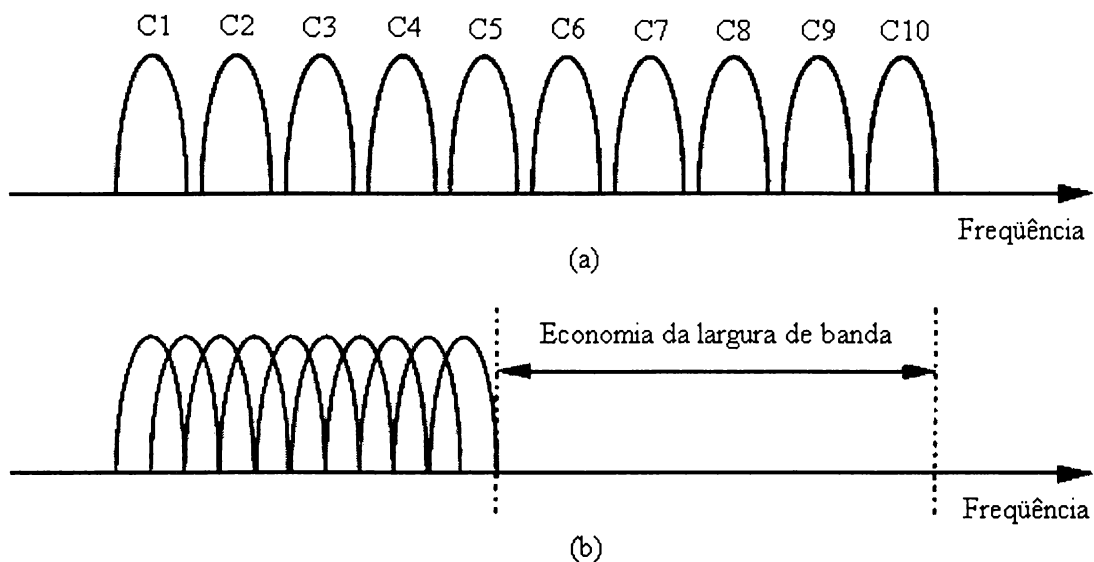


Figura 1: Conceito do sinal OFDM. (a) Sem superposição; (b) Com superposição

Pode-se ainda arranjar as portadoras em um sinal OFDM de modo que as bandas laterais de cada uma delas se sobreponham e os sinais sejam ainda recebidos sem o problema da interferência adjacente. Para tanto, as portadoras devem ser matematicamente ortogonais. O receptor atua como um banco de demoduladores, trasladando cada portadora para o nível DC, de modo que o sinal resultante é integrado sobre um período de símbolo a fim de recuperar os dados. Pode-se dizer, ainda que as portadoras são linearmente independentes, ou seja ortogonais, se o espaçamento entre elas é múltiplo de $1/T$, onde T é o período do símbolo.

2.1 HISTÓRICO

A maioria das pesquisas realizadas visavam um esquema de transmissão com múltiplas portadoras de alta eficiência baseado em portadoras ortogonais em frequência. Em 1971, Weinstein e Ebert [12] aplicaram a transformada discreta de Fourier (DFT) a sistemas de transmissão de dados paralelos como parte de um processo de modulação e demodulação. A Figura 2 (a) mostra o espectro dos dados em uma portadora. O sinal OFDM multiplexado com um espaçamento em frequência constante e igual à velocidade de transmissão de cada subportadora está mostrado

na Figura 2 (b). Nota-se que na frequência central de cada subportadora não há *crosstalk* de outros canais. Dessa forma, utilizando-se a DFT no receptor e calculando-se valores de correlação na frequência central de cada subportadora, pode-se recuperar os dados transmitidos sem *crosstalk*.

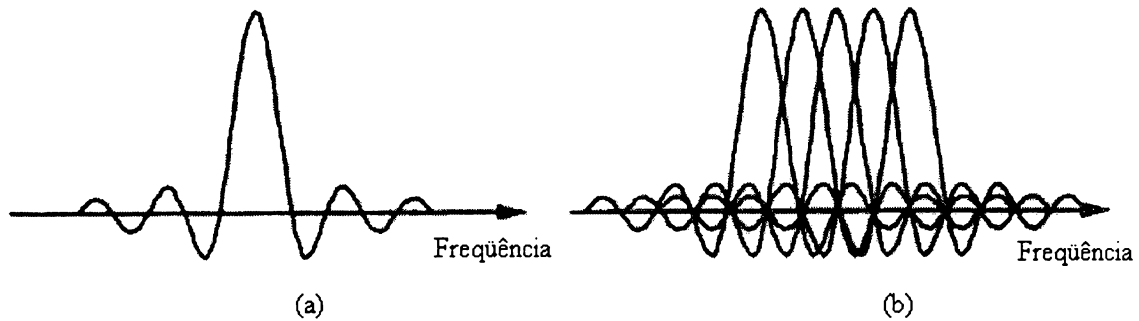


Figura 2: (a) Espectro de um subcanal OFDM; (b) Espectro do sinal OFDM

Para eliminar os bancos de osciladores e demoduladores requisitados pela multiplexação por divisão em frequência, implementações digitais podem ser realizadas através da Transformada Rápida de Fourier (FFT), a qual é uma implementação eficiente da DFT. Avanços na tecnologia VLSI (*very-large-scale integration*) tornaram os *chips* de FFT de alta velocidade comercialmente disponíveis. Utilizando esse método, o transmissor e o receptor são implementados usando técnicas de FFT, que reduz o número de operações da ordem de N^2 para a ordem de $N \log N$ [13].

Nos anos 60, as técnicas de OFDM foram usadas em vários sistemas militares de alta frequência como KINEPLEX [14], ANDEFT [15] e KATHRYN [16].

Nos anos 80, OFDM foi aplicado em modems de alta velocidade, comunicações digitais móveis e gravação de alta densidade. Modems de várias velocidades também foram desenvolvidos para redes de telefonia [17].

A partir de 1990, o OFDM foi aplicado a comunicações de dados de banda larga sobre canais FM de rádio, HDSL (*high-bit-rate digital subscriber lines*), ADSL (*asymmetric digital subscriber lines*), VDSL (*very-high-speed digital subscriber lines*), DAB (*digital audio broadcasting*) e HDTV (*high-definition television*) [1,18,19].

2.2 VANTAGENS E DESVANTAGENS DO SISTEMA OFDM

O esquema de transmissão OFDM possui as seguintes vantagens:

- OFDM é um sistema robusto à interferência multipercurso;
- é possível melhorar a capacidade do sistema adaptando a taxa de dados por subportadora de acordo com a relação sinal-ruído de determinada subportadora;
- OFDM é um sistema eficiente contra interferência de banda estreita, pois tal interferência afeta somente uma pequena porcentagem de subportadoras.

Por outro lado, o OFDM também possui desvantagens quando comparado com a modulação de uma única portadora:

- OFDM é sensível ao ruído de fase na portadora;
- implementação do sistema OFDM requer maior complexidade computacional (*hardware* e *software*).

2.3 PRINCÍPIOS DO SISTEMA OFDM

O princípio básico do sistema OFDM é dividir uma seqüência de dados em seqüências menores que serão transmitidas por um determinado número de subportadoras.

Um sinal OFDM consiste da soma de subportadoras que são moduladas através de PSK ou QAM. Se d_i são os símbolos complexos QAM, N_s é o número de subportadoras, T é a duração do símbolo e f_c é a freqüência da subportadora, então um símbolo OFDM iniciando em $t=t_s$ pode ser escrito como mostrado na equação (2.1) [1].

$$s(t) = \text{Re} \left\{ \sum_{i=-\frac{N_s}{2}}^{\frac{N_s}{2}-1} d_{i+N_s/2} \exp(j2\pi(f_c - \frac{i+0.5}{T})(t-t_s)) \right\}, t_s \leq t \leq t_s + T \quad (2.1)$$
$$s(t) = 0 \quad , t < t_s \quad \text{ou} \quad t > t_s + T$$

Na literatura, uma notação equivalente é utilizada, a qual é exibida na equação (2.2).

$$s(t) = \sum_{i=-\frac{N_s}{2}}^{\frac{N_s}{2}-1} d_{i+N_s/2} \exp(j2\pi \frac{i}{T}(t-t_s)) \quad , t_s \leq t \leq t_s + T \quad (2.2)$$

$$s(t) = 0 \quad , t < t_s \quad \text{ou} \quad t > t_s + T$$

Nessa representação, as partes real e imaginária correspondem às componentes em fase e quadratura do sinal OFDM que devem ser multiplicadas por um seno e um cosseno na frequência da portadora desejada para produzir o sinal OFDM final. A Figura 3 mostra a operação do modulador OFDM em um diagrama em bloco.

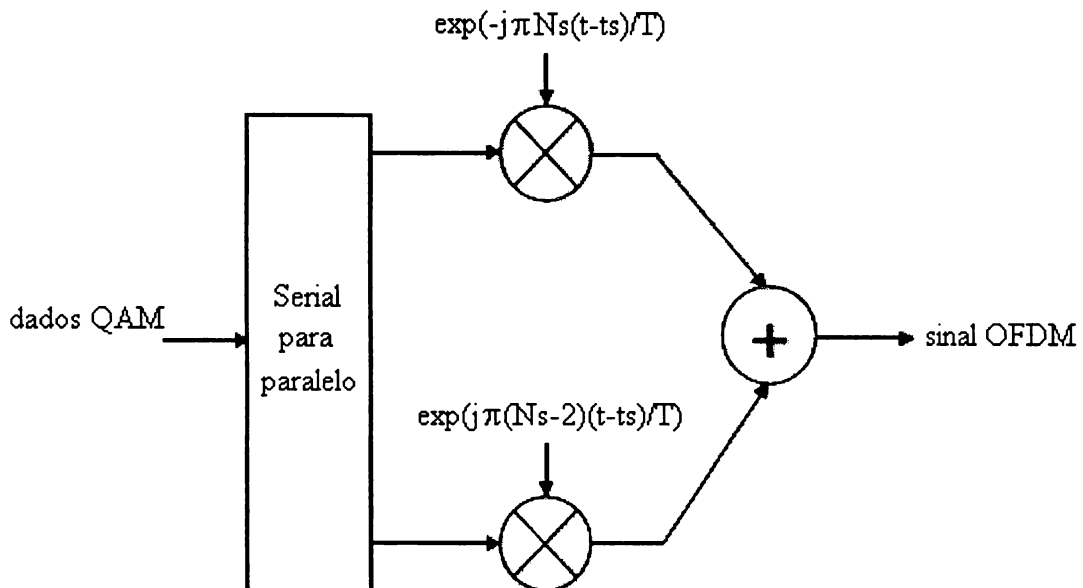


Figura 3: Modulador OFDM

O sinal OFDM definido pela equação (2.2) é a transformada inversa de Fourier de N_s símbolos de entrada QAM. O equivalente discreto no tempo é a transformada discreta inversa de Fourier (IDFT), a qual é dada pela equação (2.3),

onde o tempo t é substituído pelo número de amostra n . Na prática, essa transformada pode ser implementada de maneira muito eficiente através da Transformada Rápida Inversa de Fourier (IFFT).

$$s(n) = \sum_{i=0}^{N_i-1} d_i \exp(j2\pi \frac{in}{T}) \quad (2.3)$$

Em um processamento do sinal OFDM, dados binários podem ser utilizados como entrada para um código corretor de erro. Em seguida, esses dados codificados são mapeados em valores QAM e o sinal OFDM é formado utilizando a IFFT. Na recepção, o processo inverso é realizado. No caso da codificação com RS, as subportadoras são demoduladas pela FFT, os símbolos complexos QAM são transformados em símbolos que contêm bits e ocorre a decodificação. Já com os códigos convolucionais e turbo, como será mostrado nos próximos capítulos, o sinal é demodulado pela FFT e os símbolos são transformados em métricas que servirão de entrada para o decodificador.

3 CÓDIGOS CORRETORES DE ERRO

A utilização dos códigos controladores de erros provém da necessidade de armazenar ou transmitir grandes volumes de dados, muitos dos quais são sensíveis a erros. Os códigos controladores de erros são largamente utilizados em sistemas de comunicação via satélite, em redes locais de computadores, em discos a laser, em sistemas de tele-supervisão e controle e em automação bancária [20].

Dessa forma, em sistemas em que se deseja alta confiabilidade na transmissão ou armazenamento de dados, faz-se necessária a implementação com codificadores e decodificadores de códigos controladores de erro.

Nesse capítulo é mostrado um diagrama simplificado de um sistema de comunicação, bem como os blocos de codificação e decodificação de canal. Posteriormente, será realizada uma classificação dos códigos de canal bem como algumas das aplicações e descrições dos mesmos.

O diagrama básico de um sistema de comunicação digital é constituído por uma fonte de dados digitais, um sistema de codificação onde estão localizadas as funções de codificação de fonte e códigos de canal (códigos corretores de erro), de um modulador, de um canal de transmissão que pode ser a rede elétrica, um par de fios, um cabo de fibras ópticas, um enlace de microondas ou qualquer outro meio de comunicação. Na recepção, tem-se o decodificador de canal seguido pelo decodificador de fonte, pelo demodulador e, finalmente, pelo usuário do sistema. A Figura 4 mostra o esquema simplificado.

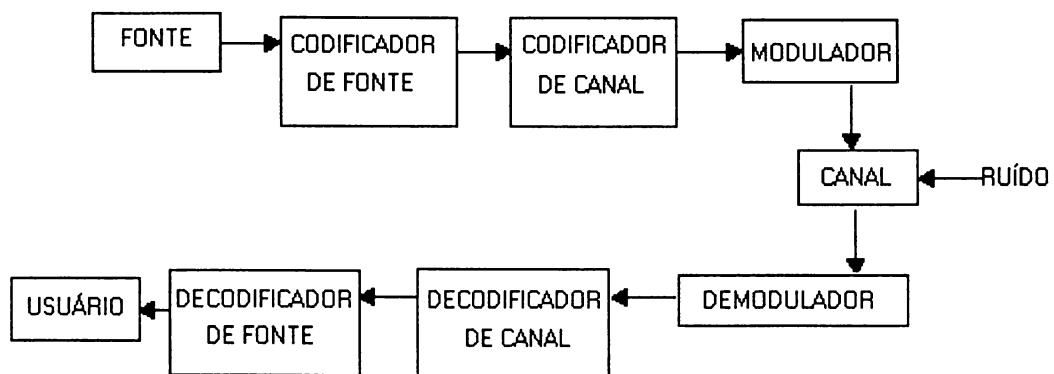


Figura 4: Modelo simplificado de um sistema de comunicação digital

Os códigos de fonte possuem a finalidade de codificar os dados gerados pela a fonte levando em consideração a estatística de acontecimento dos símbolos da fonte.

O codificador de canal está projetado para, através da inclusão de símbolos de redundância de forma inteligente, fazer com que a informação após trafegar pelo canal possa ser recuperada devido a eventuais erros de acordo com o critério de qualidade exigido.

O conjunto modulador-demodulador (*modem*) transforma o canal analógico em canal discreto para possibilitar o funcionamento do conjunto codificador-decodificador (*codec*).

3.1 TIPOS DE CÓDIGOS DE CANAL

Pode-se classificar os códigos controladores de erro [20-25] de duas maneiras: de acordo com a aplicação e com a estrutura.

De acordo com sua aplicação podem ser divididos em códigos detectores e corretores de erros.

Os sistemas que usam códigos para detecção de erro são mais simples, pois geralmente permitem uma interrupção no fluxo de dados e estão associados a protocolos do tipo ARQ (solicitação automática de retransmissão). A detecção de erros é largamente usada em redes locais de computadores [20].

A correção de erros é utilizada em sistemas de fluxo contínuo de dados, onde, por motivos vários, não é possível repetir a mensagem. A correção de erros é muito usada em sistemas de comunicação via satélite, onde, devido às limitações impostas pelo peso e pela potência de transmissão do satélite, a confiabilidade só pode ser aumentada com auxílio desses códigos.

Quanto à estrutura, pode-se dividir os códigos controladores de erro em dois grupos principais: códigos de bloco e códigos convolucionais.

Os códigos de bloco caracterizam-se pela mensagem ser subdividida em unidades de comprimento fixo. Já os códigos convolucionais são caracterizados por um fluxo contínuo de dados entrelaçando os símbolos de redundância com os de informação. Uma outra diferença que caracteriza os tipos de códigos é que os

códigos de bloco são códigos sem memória, ou seja, os símbolos de redundância dependem exclusivamente dos símbolos de informação da palavra a ser codificada. Já os códigos convolucionais são códigos com memória, cujos símbolos de redundância dependem também das últimas palavras codificadas.

A mais recente técnica de codificação é realizada através dos códigos turbo [6-9] que consistem de duas inovações importantes: codificação com concatenação em paralelo e decodificação iterativa. Os codificadores concatenados em paralelo são formados por dois ou mais codificadores elementares de bloco ou convolucionais.

Na sua forma mais simples, a concatenação em paralelo funciona da seguinte forma. Supondo que existem dois codificadores, um bloco de mensagem m é codificado utilizando o primeiro codificador, gerando a palavra-código (ou seqüência) c_1 . Em seguida, a mensagem original m passa através de um entrelaçador e o resultado é utilizado como a mensagem para o segundo codificador c_2 . Assim, m , c_1 e c_2 são multiplexados e transmitidos pelo canal até atingirem o receptor. Tipicamente, os codificadores elementares são codificadores convolucionais binários e o entrelaçador é um dispositivo que possui a característica de permutar suas coordenadas, embaralhando os bits que o atingem, ou seja, alterando as posições dos bits que chegam em sua entrada.

3.2 HISTÓRIA DOS CÓDIGOS CONTROLADORES DE ERRO

A história dos códigos controladores de erro teve início em 1948, com a publicação do artigo de Claude Shannon [26]. Em seu estudo, Shannon mostrou que existe um número C , chamado capacidade do canal e medido em bits por segundo, que está associado a cada canal. Sempre que a taxa de transmissão R em bits por segundo de um sistema de comunicação for menor que C , então é possível projetar para esse canal um sistema de comunicação com códigos controladores de erros, cuja probabilidade de erro na saída do sistema é tão pequena quanto se queira. Shannon não disse, porém como encontrar esses códigos.

Em 1950, foi concentrado muito esforço para encontrar uma regra de formação de classes de códigos que produzissem a prometida probabilidade de erro. A primeira tentativa veio com os códigos de bloco com uma forte estrutura algébrica.

O primeiro código de bloco foi introduzido por Hamming, em 1950, com a correção de apenas um erro.

Desde então, muitos tipos de códigos de blocos foram descobertos, sendo que os maiores progressos foram quando, em 1960, Bose e Ray-Chaudhuri e, em 1959, Hocquenghem encontraram uma classe de códigos corretores de múltiplos erros, os códigos BCH. Enquanto isso, também em 1960, Reed e Solomon encontraram a mesma classe de códigos para canais não-binários.

Uma segunda tentativa veio com os códigos convolucionais. Em 1967, foi desenvolvido o algoritmo de Viterbi, ótimo para a decodificação desses códigos. Esse algoritmo ganhou popularidade por sua baixa complexidade. Antes disso, os códigos convolucionais eram decodificados por algoritmos seqüenciais.

Em 1993, Berrou, Glavieux e Thitimajshima [7] introduziram uma nova técnica de codificação, os códigos turbo que produziam resultados muito eficientes e próximos ao limite de Shannon para canais AWGN.

3.3 APLICAÇÕES DOS CÓDIGOS CORRETORES DE ERRO

Os códigos controladores de erro são empregados em praticamente todos os sistemas de transmissão ou armazenamento de dados em que se deseja atingir uma determinada confiabilidade. Pode-se citar:

- Comunicação via satélite;
- Redes locais de computadores;
- Sistemas de tele-supervisão e telecontrole;
- Sistemas de automação bancária, ferroviária e industrial;
- Sistemas de gravação magnética e em discos ópticos;
- *Modems* de alta velocidade (modulação codificada);

3.4 CÓDIGOS DE BLOCO

Um código de bloco codifica um bloco de k símbolos de entrada em n símbolos codificados, onde n é maior do que k . Nesses códigos, cada bloco codificado de n elementos depende somente dos k bits de informação. Ao contrário dos códigos convolucionais, os códigos de bloco não possuem memória pelo fato da palavra-código não depender do histórico das informações. Dessa forma, cada palavra que sai do codificador, ou seja, cada palavra-código, depende unicamente do código utilizado e dos bits de informação na entrada do codificador.

Para k bits de informação e n bits de codificação, existem $(n-k)$ dígitos de redundância, os quais são chamados de dígitos de verificação de paridade.

3.4.1 Definições básicas

A taxa de codificação (R) de um código de bloco é definida como a relação entre o número de bits de informação (k) e o comprimento da palavra-código (n), como mostrada na equação (3.1).

$$R = \frac{k}{n} \quad (3.1)$$

Outras definições importantes são as de peso de Hamming e de distância de Hamming.

O peso de Hamming de uma palavra-código é o número de posições não-nulas dessa palavra. Como exemplo, pode-se obter o peso da palavra-código 1101000, a qual pertence ao código de Hamming de comprimento $n=7$ e $k=4$. Nota-se que o peso de Hamming para essa palavra é igual a 3.

A distância de Hamming entre duas palavras de mesmo número de componentes é igual ao número de posições em que as palavras diferem. Pode-se calcular a distância de Hamming entre duas palavras-código, como exemplo 1101000 e 0110100. Nesse caso, a distância é igual a 4 pelo fato das palavras diferirem na primeira, terceira, quarta e quinta posições.

3.4.2 Capacidade de detecção e correção de erro

Devido aos bits de redundância acrescentados de forma inteligente nas palavras do código, estas palavras podem diferir entre si de um número maior de bits em relação à situação sem redundância. Isto é equivalente a dizer que a distância de Hamming entre as palavras-código aumenta ao inserir-se bits de redundância. Como as palavras, após a codificação tornam-se mais distanciadas entre si, fica evidente que, mesmo com algumas “distorções”, elas podem ser mais facilmente reconhecidas em relação à situação sem codificação.

Chama-se detecção de erro, o evento de reconhecimento de que houve um erro, mas que nem sempre pode-se localizá-lo. Já a correção de erro é o evento de detecção e localização do erro.

Outro conceito importante é o de padrão de erro. Chama-se padrão de erro, uma palavra recebida com erros em determinadas posições. Como exemplo, seja o código binário $C(3,1)$ onde $n=3$ e $k=1$ e que possui as palavras-código: 000, 111. Para esse código, a palavra 100 representa um padrão de um erro, pois supondo-se que foi transmitida a palavra-código 000, a palavra recebida possui um erro na primeira posição.

A capacidade de detecção t_d de um código é dada pela equação (3.2), onde d_{min} é a distância mínima de Hamming, a qual é o número mínimo de símbolos diferentes entre quaisquer duas palavras-código [20].

$$t_d = d_{min} - 1 \quad (3.2)$$

A capacidade de correção t_c é dada pela equação (3.3) [20].

$$t_c \leq \left\lfloor \frac{d_{min} - 1}{2} \right\rfloor \quad (3.3)$$

A distância mínima de Hamming d_{min} de qualquer código linear satisfaz a inequação (3.4), conhecida como Limitante de Singleton [20].

$$d_{\min} \leq n - k + 1 \quad (3.4)$$

É interessante observar que a equação (3.3) garante a correção de um número de mínimo de padrões de erro, porém nada diz a respeito do número total de padrões de erro corrigidos. Dessa forma, um código com distância mínima de Hamming igual a três, de acordo com a equação (3.3), corrige com certeza todos os padrões de um erro, porém pode corrigir alguns padrões de dois erros ou mais erros e jamais todos os padrões de dois ou mais erros.

3.5 CÓDIGOS DE REED-SOLOMON

Foi visto que um código de bloco codifica um bloco de k símbolos de entrada em n símbolos codificados, onde n é maior do que k . O propósito da adição de $n-k$ símbolos de redundância é aumentar a distância mínima de Hamming, a qual é o número mínimo de símbolos diferentes entre quaisquer duas palavras-código. Para uma distância mínima de Hamming d_{\min} , o código pode corrigir t_c erros, como mostrado na equação (3.3).

Uma importante classe dos códigos BCH não-binários que obedecem à equação (3.4) são os códigos de Reed-Solomon [1]. Nesse caso, o Limitante de Singleton reduz-se à equação (3.5) [20].

$$d_{\min} = n - k + 1 \quad (3.5)$$

Devido às suas ótimas propriedades de distância e à disponibilidade de algoritmos de decodificação eficientes, como o de Berlekamp e Massey [27,28], os códigos de Reed-Solomon tornam-se os códigos de bloco mais utilizados.

Os códigos de Reed-Solomon são definidos por blocos de símbolos com m bits por símbolo, onde o comprimento da palavra-código n está relacionado com m através da equação (3.6).

$$n = 2^m - 1 \quad (3.6)$$

O número de símbolos de entrada k relaciona-se com m e com a distância mínima de Hamming através da equação (3.7).

$$k = 2^m - d_{\min} \quad (3.7)$$

Observando a equação (3.7), parece haver pouca flexibilidade na disponibilidade de comprimentos de códigos. Todavia, um código de Reed-Solomon pode ser facilmente encurtado para qualquer comprimento zerando-se um certo número de bits de entrada e retirando a mesma quantidade de bits de saída.

De acordo com as equações (3.3) e (3.4), um código de Reed-Solomon pode corrigir até $\lfloor (n-k)/2 \rfloor$ símbolos errados. Cada símbolo contém m bits, portanto uma quantidade máxima de $m \cdot \lfloor (n-k)/2 \rfloor$ bits errados podem ser corrigidos. Se um código de Reed-Solomon é designado para corrigir até dois símbolos errados contendo 8 bits por símbolo, ele não poderá corrigir qualquer combinação de três erros de bits, pois eles poderão ocorrer em três símbolos diferentes. Essa característica faz dos códigos de Reed-Solomon úteis para a correção em canais com ruído em rajada. Um exemplo é um sistema OFDM na presença de desvanecimento em multipercurso, o qual causa erros concentrados em poucas subportadoras.

O código de Reed-Solomon adotado nas simulações desse trabalho é o código C(255,216) em $GF(2^8)$, o qual é recomendado pela norma do ADSL [5] e significa que cada símbolo do código corresponde a 8 bits.

3.6 CÓDIGOS CONVOLUCIONAIS

Os códigos convolucionais diferem dos códigos de bloco porque o codificador contém memória e suas n saídas em uma certa unidade de tempo não dependem somente das k entradas naquele tempo, mas também dos m blocos de entrada anteriores [1].

Um codificador convolucional mapeia k bits de um fluxo contínuo de entrada em n bits de saída, onde o mapeamento é realizado através da convolução dos bits de entrada com uma resposta ao impulso binária. O codificador convolucional pode

ser implementado através de simples registradores de deslocamento e de somadores módulo 2. Como exemplo, a Figura 5 mostra um codificador com taxa $\frac{1}{2}$, o qual atualmente é um dos mais utilizados e é adotado nas simulações realizadas nesse trabalho. A taxa $\frac{1}{2}$ indica que para cada bit de entrada, há dois bits de saída. Esse codificador possui uma única entrada de dados e duas saídas A_i e B_i , as quais são alternadas para formar a seqüência codificada de saída $\{A_1B_1A_2B_2\dots\}$. Cada par de bits de saída $\{A_i, B_i\}$ depende de sete bits de entrada, ou seja, do bit de entrada atual mais os seis bits de entrada anteriores que ficaram armazenados no registrador de deslocamento de comprimento 6. Esse valor 7, ou de maneira geral o comprimento do registrador de deslocamento mais 1, é chamado de *constraint length*. As ligações dos registradores de deslocamento são especificadas pelos correspondentes polinômios geradores ou vetores geradores. Como exemplo, o vetor gerador do codificador mostrado na Figura 5 é $\{1011011, 1111001\}$ ou $\{133, 171\}$ na forma octal. Os uns no vetor gerador correspondem às ligações dos registradores de deslocamento.

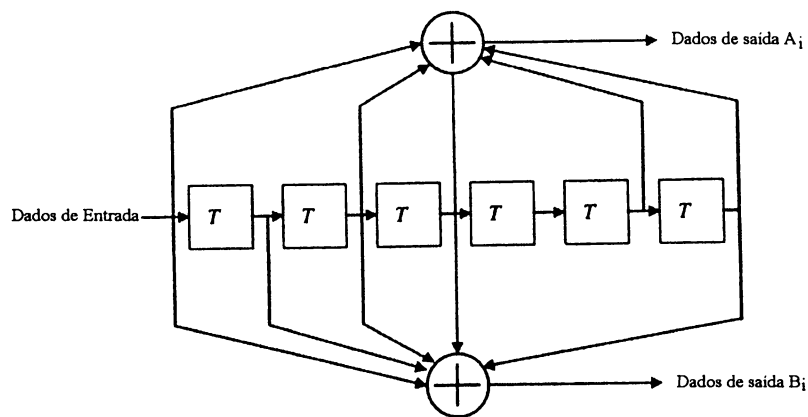


Figura 5: Diagrama em bloco de um codificador convolucional

A decodificação dos códigos convolucionais é freqüentemente realizada pelo decodificador de Viterbi com decisão suave, o qual é uma maneira eficiente de obter a estimativa ótima da seqüência codificada. Uma descrição dessa técnica de decodificação pode ser encontrada em [22]. A complexidade da decodificação de Viterbi cresce com o aumento de *constraint length*. Implementações práticas são realizadas até um *constraint length* de aproximadamente 10.

3.7 CÓDIGOS TURBO

Os códigos turbo, introduzidos por Berrou [7], são uma nova classe de códigos convolucionais cujo desempenho em termos da taxa de erro de bit (*BER*) está próxima ao limite de Shannon para canais AWGN. O codificador turbo é construído utilizando uma concatenação em paralelo de dois códigos convolucionais sistemáticos e recursivos (RSC). Para a decodificação é utilizada uma regra iterativa com realimentação e a implementação é feita com decodificadores elementares associados aos codificadores.

Esse item apresenta os esquemas de codificação e decodificação turbo baseados no artigo de Berrou [7].

Para um código convolucional binário com taxa $R=1/2$ com *constraint length* K e memória $M=K-1$, a entrada para o codificador no tempo k é um bit d_k e a correspondente palavra-código C_k é a dupla binária (X_k, Y_k) com

$$X_k = \sum_{i=0}^{K-1} g_{1i} d_{k-i} \quad g_{1i} = 0, 1 \quad (3.8a)$$

$$Y_k = \sum_{i=0}^{K-1} g_{2i} d_{k-i} \quad g_{2i} = 0, 1 \quad (3.8b)$$

Onde $G_1: \{g_{1i}\}$, $G_2: \{g_{2i}\}$ são as geradoras do codificador, geralmente escritas em forma octal e as operações realizadas são módulo 2.

Um código RSC binário com taxa $R=1/2$ é obtido a partir de um código não sistemático através de um *loop* de realimentação e fazendo com que uma das duas saídas X_k ou Y_k seja igual ao bit de entrada d_k . Para um código RSC, a entrada do registrador de deslocamento (memória) é uma nova variável binária a_k . Se $X_k = d_k$ (ou respectivamente $Y_k = d_k$), a saída Y_k (ou respectivamente X_k) é igual à equação (3.8b) (ou respectivamente 3.8a) apenas substituindo a_k em d_k e a variável a_k é recursivamente calculada como em (3.9).

$$a_k = d_k + \sum_{i=1}^{K-1} \gamma_i a_{k-i} \quad (3.9)$$

Onde γ_i é respectivamente igual a g_{1i} se $X_k = d_k$ ou igual a g_{2i} se $Y_k = d_k$. A equação (3.9) pode ser reescrita como em (3.10).

$$d_k = \sum_{i=0}^{K-1} \gamma_i a_{k-i} \quad (3.10)$$

Um exemplo de codificador RSC é mostrado na Figura 6.

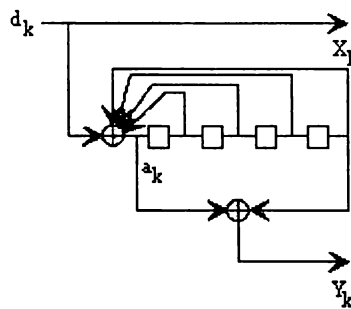


Figura 6: Código Sistemático Recursivo

A estrutura em treliça é idêntica para os códigos recursivos sistemáticos e não-recursivos sistemáticos. Contudo, as duas seqüências de saída $\{X_k\}$ e $\{Y_k\}$ não correspondem à mesma seqüência de entrada $\{d_k\}$ para códigos RSC e NSC. Essa é a principal diferença entre os dois códigos.

3.7.1 Codificador Turbo Padrão

A Figura 7 mostra um exemplo de dois codificadores com códigos RSC idênticos com concatenação paralela [7]. Ambas as entradas dos codificadores elementares (C_1 e C_2) utilizam o mesmo bit d_k , mas seguindo uma seqüência diferente devido à presença de um entrelaçador (I). Para uma seqüência de bits de entrada $\{d_k\}$, as saídas X_k e Y_k do codificador no tempo k são respectivamente iguais a d_k e à saída Y_{1k} do codificador C_1 ou à saída Y_{2k} do codificador C_2 .

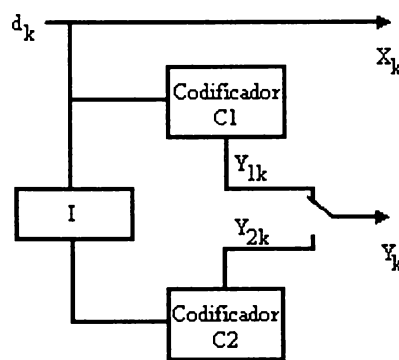


Figura 7: Codificador Turbo Padrão

A fim de se obter um maior entendimento do codificador turbo, pode-se dizer que o processo de codificação inicia-se como mostrado na Figura 8. A seqüência de informação (01101) atravessa o codificador convolucional (C_1), o qual a codifica em uma outra seqüência que possui os bits de paridade (10110). A seqüência de informação atravessa, ao mesmo tempo, um segundo codificador (C_2) idêntico ao primeiro. A entrada desse segundo codificador não é seqüência original dos bits de informação, mas uma nova seqüência com os bits de informação em outra ordem (10011), embaralhados por um dispositivo chamado entrelaçador. Esse codificador então lê os bits de informação embaralhados e obtém os bits de paridade (11100). Finalmente, o transmissor, através de uma função de *puncturing* (P), determina a taxa de codificação e transmite os dados codificados pelo canal. No caso da taxa igual a $\frac{1}{2}$, a função de *puncturing* fornece a seqüência de saída (0111110010) multiplexando os bits de informação com os bits de paridade ímpares gerados pelo codificador convolucional C_1 e com os bits de paridade par gerados por C_2 .

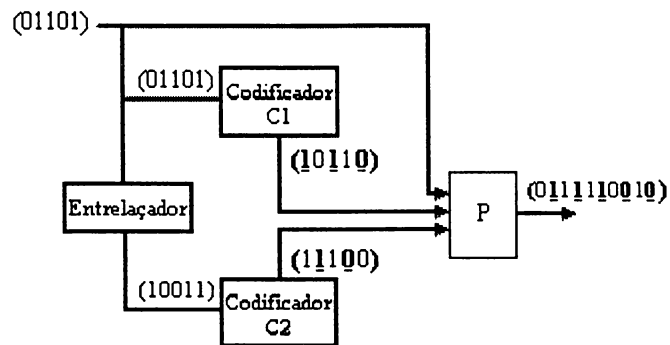


Figura 8: Codificação Turbo com função de *puncturing*

O embaralhamento dos bits no entrelaçador é um passo chave em todo o processo de codificação e decodificação e a função dessa permutação é introduzir um comportamento randômico ao código.

3.7.2 Princípio do Decodificador Turbo Padrão

O processo de decodificação utiliza dois decodificadores que trabalham em conjunto para realizar uma melhor estimativa dos bits decodificados. A função de cada decodificador é processar o dado que foi contaminado por ruído durante sua passagem pelo canal e decidir qual o valor transmitido (0 ou 1). Esse processo de estimativa é realizado para cada bit enviado.

Cada decodificador trabalha com informações de confiabilidade que ajudarão no processo de estimativa dos bits. Primeiramente, ele inspeciona o nível do sinal analógico dos bits recebido. Enquanto muitos esquemas transformam o sinal recebido em zeros e uns (decisão abrupta), o que desperdiça informações valiosas de confiabilidade do sinal, o decodificador turbo transforma o sinal recebido em números inteiros que fornecem a cada decodificador uma medida de confiabilidade sobre a estimativa de cada bit. Ainda, os decodificadores voltam-se aos seus bits de paridade, os quais verificam se os dados recebidos estão intactos ou contêm erros.

O resultado dessa análise é uma estimativa confiável para cada bit decodificado. Os códigos turbo trabalham com decisões de bits que levam em consideração valores de confiabilidade. Essas confiabilidades de bit são expressas por números denominados logaritmos da razão de verossimilhança (LLR) e podem variar, como exemplo, entre -7 e +7. Uma razão de +7 significa que o decodificador tem a absoluta certeza de que o bit transmitido é o bit 1. Já para um valor -5, o decodificador acredita que se trata do bit zero. Quando isso ocorre, diz-se que o decodificador realiza uma decisão suave dos dados transmitidos.

Embora o nível do sinal e os bits de verificação de paridade forneçam informações úteis para a decodificação, elas não são suficientes. Dessa forma, os dois decodificadores elementares devem trocar informações a respeito da confiabilidade do bit, pois os bits de paridade referem-se a mesma seqüência de informação transmitida, mas com os bits arranjados de forma diferente. Dessa forma, os decodificadores trocam informações de confiabilidade em um processo iterativo para melhorar a decodificação.

Após essa avaliação qualitativa do princípio do decodificador turbo, é necessário analisar o decodificador mostrado na Figura 9. Ele é constituído por dois decodificadores elementares (DEC_1 e DEC_2) em um esquema de concatenação em série. O primeiro decodificador elementar DEC_1 está associado ao codificador C_1 de taxa R_1 e realiza uma decisão suave.

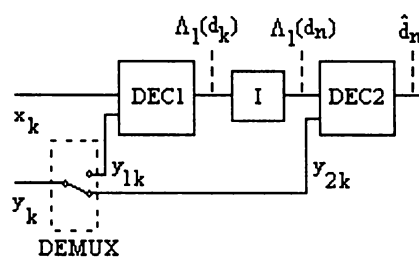


Figura 9: Princípio do decodificador Turbo Padrão

A entrada do decodificador é constituída por uma dupla R_k de variáveis randômicas x_k e y_k , no tempo k . A informação redundante y_k é demultiplexada e enviada ao decodificador DEC_1 quando $Y_k=Y_{1k}$ e ao decodificador DEC_2 quando $Y_k=Y_{2k}$. Quando a informação redundante de um certo codificador (C_1 ou C_2) não é

emitida, a correspondente entrada do codificador é igual a zero. Isso é realizado pelo bloco *Demux*.

Sabe-se que a decodificação suave é melhor que a abrupta, portanto o primeiro decodificador DEC_1 deve entregar ao segundo DEC_2 uma decisão suave. O logaritmo da razão de verossimilhança (LLR), $\Lambda_1(d_k)$ associado a cada bit decodificado d_k pelo primeiro decodificador DEC_1 é uma fração relevante de informação para o segundo decodificador DEC_2 e é calculado a partir da equação (3.11) [7].

$$\Lambda_1(d_k) = \text{Log} \frac{P_r\{d_k = 1 / \text{observação}\}}{P_r\{d_k = 0 / \text{observação}\}} \quad (3.11)$$

Em (3.11), $P_r\{d_k = i / \text{observação}\}$, $i = 0, 1$ é a probabilidade a posteriori (APP) do bit d_k .

Nota-se então que o primeiro decodificador fornece ao segundo uma informação de estimativa do bit que foi enviado no instante k . O entrelaçador (I) embaralha $\Lambda_1(d_k)$, produzindo $\Lambda_1(d_n)$. O segundo decodificador de posse dessa informação e de y_{2k} , produz uma estimativa do bit d_n .

3.7.3 Decodificação Ótima com Decisão Suave

O algoritmo de Viterbi é um método ótimo de decodificação que minimiza a probabilidade de erro da seqüência para os códigos convolucionais. Infelizmente, esse algoritmo não é capaz de produzir a APP para cada bit decodificado. Um algoritmo relevante para tanto foi proposto por Bahl, mostrado em Berrou [7]. Esse algoritmo minimiza a probabilidade de erro de bit na decodificação de códigos de blocos lineares e códigos convolucionais, produzindo a APP para cada bit decodificado. Para códigos RSC, o algoritmo de Bahl deve ser modificado para levar em consideração sua característica recursiva [7].

Considerando um código RSC com *constraint lenght* K , no tempo k , o estado do codificador S_k é representado por uma k -tupla

$$S_k = (a_k, a_{k-1}, \dots, a_{k-K+1}) \quad (3.12)$$

Supõe-se também que a seqüência dos bits de informação $\{d_k\}$ é formada por N bits d_k independentes que assumem os valores 0 e 1 com igual probabilidade e que o estado inicial do codificador S_0 e o estado final S_N são ambos iguais a zero, ou seja:

$$S_0 = S_N = (0, 0, \dots, 0) \quad (3.13)$$

A seqüência de saída do codificador (palavra-código), denotada por $C_1^N = \{C_1, \dots, C_k, \dots, C_N\}$ é a entrada para um canal de comunicação cuja saída será a seqüência $R_1^N = \{R_1, \dots, R_k, \dots, R_N\}$ onde $R_k = (x_k, y_k)$.

A APP de um bit d_k decodificado pode ser derivada a partir da probabilidade conjunta $\lambda_k^i(m)$ definida por

$$\lambda_k^i(m) = P_r \{d_k = i, S_k = m / R_1^N\} \quad (3.14)$$

A APP de um bit decodificado d_k é igual a

$$P_r \{d_k = i / R_1^N\} = \sum_m \lambda_k^i(m), \quad i = 0, 1. \quad (3.15)$$

Das relações (3.11) e (3.15), o LLR $\Lambda(d_k)$ associado ao bit decodificado d_k pode ser expresso por (3.16).

$$\Lambda(d_k) = \text{Log} \frac{\sum_m \lambda_k^1(m)}{\sum_m \lambda_k^0(m)} \quad (3.16)$$

Finalmente, o decodificador pode fazer uma decisão através da comparação do $\Lambda(d_k)$ a um limiar igual a zero

$$\begin{aligned}\hat{d}_k &= 1 \quad \text{se } \Lambda(d_k) > 0 \\ \hat{d}_k &= 0 \quad \text{se } \Lambda(d_k) < 0\end{aligned}\tag{3.17}$$

A fim de computar a probabilidade $\lambda_k^i(m)$, são introduzidas as funções probabilidades $\alpha_k^i(m)$, $\beta_k(m)$ e $\gamma_i(R_k, m', m)$, como mostrado em [7].

$$\alpha_k^i(m) = \frac{P_r\{d_k = i, S_k = m, R_1^k\}}{P_r\{R_1^k\}} P_r\{d_k = i, S_k = m / R_1^k\}\tag{3.18}$$

$$\beta_k(m) = \frac{P_r\{R_{k+1}^N / S_k = m\}}{P_r\{R_{k+1}^N / R_1^k\}}\tag{3.19}$$

$$\gamma_i(R_k, m', m) = P_r\{d_k = i, R_k, S_k = m / S_{k-1} = m'\}\tag{3.20}$$

A probabilidade conjunta $\lambda_k^i(m)$ pode ser reescrita utilizando a regra de Bayes:

$$\lambda_k^i(m) = \frac{P_r\{d_k = i, S_k = m, R_1^k, R_{k+1}^N\}}{P_r\{R_1^k, R_{k+1}^N\}}\tag{3.21}$$

Portanto, obtém-se

$$\lambda_k^i(m) = \frac{P_r\{d_k = i, S_k = m, R_1^k\}}{P_r\{R_1^k\}} \frac{P_r\{R_{k+1}^N / d_k = i, S_k = m, R_1^k\}}{P_r\{R_{k+1}^N / R_1^k\}}\tag{3.22}$$

Considerando que os eventos após um tempo k não são influenciados pela observação R_1^k e pelo bit d_k se o estado S_k é conhecido, a probabilidade $\lambda_k^i(m)$ é igual a

$$\lambda_k^i(m) = \alpha_k^i(m) \beta_k(m)\tag{3.23}$$

As probabilidades $\alpha_k^i(m)$ e $\beta_k(m)$ podem ser calculadas recursivamente a partir da probabilidade $\gamma_i(R_k, m', m)$ e obtém-se as equações (3.24) e (3.25) [7].

$$\alpha_k^i(m) = \frac{\sum_{m'} \sum_{j=0}^1 \gamma_i(R_k, m', m) \alpha_{k-1}^j(m')}{\sum_m \sum_{m'} \sum_{i=0}^1 \sum_{j=0}^1 \gamma_i(R_k, m', m) \alpha_{k-1}^j(m')} \quad (3.24)$$

$$\beta_k(m) = \frac{\sum_{m'} \sum_{i=0}^1 \gamma_i(R_{k+1}, m, m') \beta_{k+1}(m')}{\sum_m \sum_{m'} \sum_{i=0}^1 \sum_{j=0}^1 \gamma_i(R_{k+1}, m', m) \alpha_k^j(m')} \quad (3.25)$$

A partir da relação (3.20), $\gamma_i(R_k, m', m)$ é dada por

$$\begin{aligned} \gamma_i(R_k, m', m) &= p(R_k / d_k = i, S_k = m, S_{k-1} = m') \\ q(d_k = i / S_k = m, S_{k-1} = m') &= \pi(S_k = m, S_{k-1} = m') \end{aligned} \quad (3.26)$$

Onde $p(\dots)$ é a probabilidade de transição do canal discreto sem memória. Condicionados a $(d_k = i, S_k = m, S_{k-1} = m')$, x_k e y_k são duas variáveis não-correlacionadas e portanto, obtém-se

$$\begin{aligned} p(R_k / d_k = i, S_k = m, S_{k-1} = m') &= p(x_k / d_k = i, S_k = m, S_{k-1} = m') \\ p(y_k / d_k = i, S_k = m, S_{k-1} = m') & \end{aligned} \quad (3.27)$$

Como o codificador convolucional é uma máquina determinística, $q(d_k = i / S_k = m, S_{k-1} = m')$ é igual a 0 ou 1. As probabilidades de transição de estado $\pi(S_k = m / S_{k-1} = m')$ da treliça são definidas pelas estatísticas de entrada do codificador. Geralmente, $P_r\{d_k = 1\} = P_r\{d_k = 0\} = 1/2$ e como existem duas transições possíveis para cada estado, $\pi(S_k = m / S_{k-1} = m') = 1/2$ para cada uma dessas transições.

Os passos diferentes do algoritmo de Bahl estão mostrados abaixo.

Passo 0: As probabilidades $\alpha_0^i(m)$ e $\beta_N(m)$ são inicializadas de acordo com a relação

$$\begin{aligned}\alpha_0^i(0) &= 1 & \alpha_0^i(m) &= 0 & \forall m \neq 0, i = 0, 1 \\ \beta_N(0) &= 1 & \beta_N(m) &= 0 & \forall m \neq 0\end{aligned}\quad (3.28)$$

Passo 1: Para cada observação R_k , as probabilidades $\alpha_k^i(m)$ e $\gamma_i(R_k, m', m)$ são calculadas utilizando as relações (3.24) e (3.26) respectivamente.

Passo 2: Quando a seqüência R_1^N for completamente recebida, as probabilidades $\beta_k(m)$ são computadas utilizando a relação (3.25) e as probabilidades $\alpha_k^i(m)$ e $\beta_k(m)$ são multiplicadas para a obtenção de $\lambda_k^i(m)$. Finalmente, a LLR associada com cada bit decodificado d_k é computado a partir da relação (3.20).

3.7.4 A Informação Extrínseca do Decodificador RSC

A LLR $\Lambda(d_k)$ associada a cada bit decodificado d_k é a soma da LLR de d_k na entrada do decodificador e de uma outra informação chamada informação extrínseca, gerada pelo decodificador.

Utilizando a equação (3.16) de definição de LLR e as relações (3.24) e (3.25), obtém-se a equação (3.29) abaixo.

$$\Lambda(d_k) = \text{Log} \frac{\sum_m \sum_{m'} \sum_{j=0}^1 \gamma_1(R_k, m', m) \alpha_{k-1}^j(m') \beta_k(m)}{\sum_m \sum_{m'} \sum_{j=0}^1 \gamma_0(R_k, m', m) \alpha_{k-1}^j(m') \beta_k(m)} \quad (3.29)$$

Uma vez que o codificador é sistemático ($X_k = d_k$), a probabilidade de transição $p(x_k / d_k = i, S_k = m, S_{k-1} = m')$ na expressão $\gamma_i(R_k, m', m)$ é independente dos valores de estado S_k e S_{k-1} . Assim, pode-se fatorar essa probabilidade no numerador e denominador.

$$\Lambda(d_k) = \text{Log} \frac{p(x_k / d_k = 1)}{p(x_k / d_k = 0)} + \text{Log} \frac{\sum_m \sum_{m'} \sum_{j=0}^1 \gamma_1(y_k, m', m) \alpha_{k-1}^j(m') \beta_k(m)}{\sum_m \sum_{m'} \sum_{j=0}^1 \gamma_0(y_k, m', m) \alpha_{k-1}^j(m') \beta_k(m)} \quad (3.30)$$

$$W_k = \Lambda(d_k) \Big|_{x_k=0} = \text{Log} \frac{\sum_m \sum_{m'} \sum_{j=0}^1 \gamma_1(y_k, m', m) \alpha_{k-1}^j(m') \beta_k(m)}{\sum_m \sum_{m'} \sum_{j=0}^1 \gamma_0(y_k, m', m) \alpha_{k-1}^j(m') \beta_k(m)} \quad (3.31)$$

Nessa equação, W_k é uma função redundante da informação introduzida pelo codificador. Essa quantidade representa a informação extrínseca fornecida pelo decodificador e não depende da entrada x_k do decodificador. Essa propriedade é usada para a decodificação de dois codificadores concatenados em paralelo.

3.7.5 Decodificador Turbo Padrão com Realimentação

Como exposto anteriormente, a informação de confiabilidade gerada por um decodificador é útil para o outro decodificador produzir uma melhor estimativa do bit enviado e vice-versa, devido ao fato das seqüências de bits de paridade referirem-se aos mesmos bits de informação, mas codificados em outra ordem.

Os dois decodificadores devem trocar informações de maneira iterativa para melhorar seu desempenho. O coração da decodificação turbo é esse processo iterativo, em que cada decodificador elementar utiliza uma "informação" gerada pelo outro decodificador no passo anterior de decodificação (DEC_2 utiliza $\Lambda_1(d_n)$ e DEC_1 , a informação extrínseca como realimentação). Após um certo número de iterações, os decodificadores elementares entram em concordância com as estimativas de todos os bits.

Pode-se agora mostrar como esse processo iterativo funciona. Considera-se que ambos os decodificadores DEC_1 e DEC_2 utilizam o algoritmo de Bahl modificado. Foi visto em seções anteriores que a LLR na saída do decodificador pode ser expressa por uma soma de dois termos se as entradas do decodificador forem independentes. Portanto, se as entradas $\Lambda_1(d_k)$ e y_{2k} são independentes, a LLR

$\Lambda_2(d_k)$ na saída do decodificador DEC_2 pode ser escrita como mostrada abaixo, onde $\Lambda_1(d_k)$ é dada por (3.30).

$$\Lambda_2(d_k) = f(\Lambda_1(d_k)) + W_{2k} \quad (3.32)$$

A informação extrínseca W_{2k} do decodificador DEC_2 é função da seqüência $\{\Lambda_1(d_k)\}_{n \neq k}$. Devido à presença do entrelaçador entre os decodificadores DEC_1 e DEC_2 , a informação extrínseca W_{2k} e as observações x_k, y_{1k} estão fracamente correlacionadas. Dessa forma, a informação extrínseca W_{2k} e as observações x_k e y_{1k} podem ser utilizadas em conjunto para produzir uma nova decodificação do bit d_k e a informação extrínseca $z_k = W_{2k}$ atua com grande efeito em um processo iterativo.

A Figura 10 mostra esse novo esquema de decodificação utilizando a informação extrínseca W_{2k} gerada pelo decodificador DEC_2 em um *loop* com realimentação.

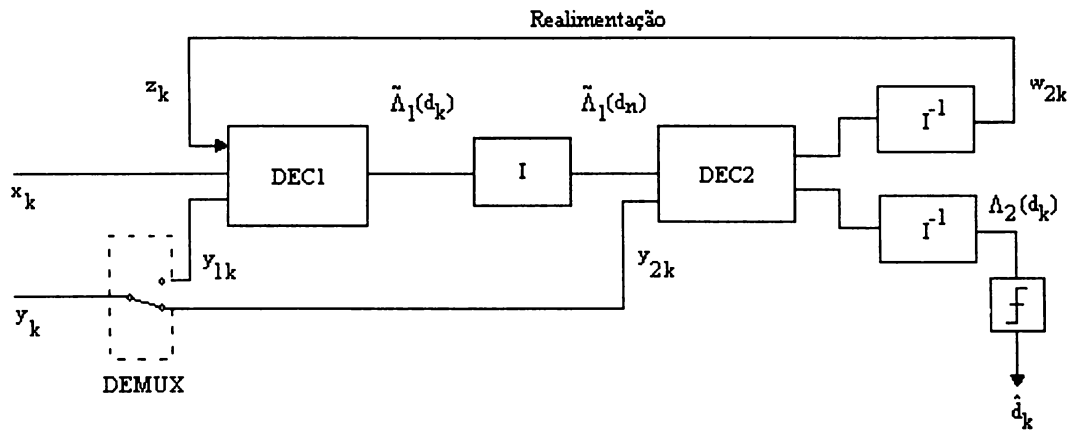


Figura 10: Decodificador Turbo padrão

O primeiro decodificador DEC_1 possui agora três entradas, (x_k, y_{1k}, z_k) e as probabilidades $\alpha'_{1k}(m)$ e $\beta_{1k}(m)$ são computadas substituindo $R_k = (x_k, y_{1k})$ por $R_k = (x_k, y_{1k}, z_k)$ nas relações (3.24) e (3.25).

O primeiro decodificador obtém uma informação de redundância com z_k , a qual melhora significativamente o seu desempenho. O termo código turbo é dado em relação a esse esquema iterativo do decodificador com referência ao princípio de máquina turbo. Quanto maior o número de iterações, melhor a estimativa dos bits.

Com o primeiro decodificador de realimentação, o LLR $\Lambda_1(d_k)$ gerado pelo decodificador DEC_1 é agora, como em (3.30), função de x_k , z_k e W_{1k} .

$$\Lambda_1(d_k) = \text{Log} \frac{p(x_k / d_k = 1)}{p(x_k / d_k = 0)} + \text{Log} \frac{p(z_k / d_k = 1)}{p(z_k / d_k = 0)} + W_{1k} \quad (3.33)$$

Na equação (3.33), W_{1k} depende da seqüência $\{z_n\}_{n \neq k}$. A informação z_k foi construída pelo decodificador DEC_2 no passo de decodificação anterior. Portanto, a seqüência de entrada do decodificador DEC_2 no passo p ($p \geq 2$) serão as seqüências $\{\tilde{\Lambda}_1(d_n)\}$ e $\{y_{2k}\}$ com

$$\tilde{\Lambda}_1(d_n) = \Lambda_1(d_n)_{z_n=0} \quad (3.34)$$

Finalmente, a partir da relação (3.32), a informação extrínseca $z_k = W_{2k}$, após passar pelo de-entrelaçador, pode ser escrita como em (3.35).

$$z_k = W_{2k} = \Lambda_2(d_k) \Big|_{\tilde{\Lambda}_1(d_k)=0} \quad (3.35)$$

A decisão na saída do decodificador turbo será, portanto:

$$\hat{d}_k = \text{sign}[\Lambda_2(d_k)] \quad (3.36)$$

3.8 SIMULAÇÃO DE SISTEMAS COM CONTROLE DE ERRO

Uma maneira eficiente de se medir o desempenho de um sistema codificado é através do ganho de codificação. Ele é definido como o ganho na relação sinal-ruído por bit E_b/N_0 relativo a um sistema sem codificação para atingir uma determinada taxa de erro de bit ou taxa de erro de palavra. O ganho E_b/N_0 é equivalente ao ganho na relação sinal-ruído (SNR) menos a taxa de perda em dB devido aos bits de redundância.

Nas curvas que serão apresentadas no próximo capítulo, é utilizada a relação E_b/N_0 , a qual é equivalente à relação de potência do sinal e de potência do ruído na largura de banda igual à taxa de bit N_0/T_b , onde T_b é o tempo de bit. Outras definições importantes são a de SNR de entrada e a de densidade espectral de energia de símbolo por densidade espectral de ruído E_s/N_0 , a qual é equivalente à relação da potência do sinal e da potência do ruído na largura de banda igual à taxa de símbolo N_0/T_s , onde T_s é a duração do símbolo. E_s/N_0 é relacionada com E_b/N_0 da seguinte maneira [1]:

$$\frac{E_s}{N_0} = \frac{E_b T_s}{N_0 T_b} \quad (3.37)$$

A relação sinal-ruído de entrada SNR_i é relacionada com E_b/N_0 de acordo a equação (3.38) [1].

$$SNR_i = \frac{E_b}{N_0} \frac{1}{BT_b} = \frac{E_b}{N_0} \frac{bN_s R}{BT_s} \quad (3.38)$$

Nessa equação, B é a largura de banda do ruído de entrada, b é o número de bits codificados por subportadora, N_s é o número de subportadoras, R é a taxa de codificação. Basicamente, o SNR_i é igual a E_b/N_0 multiplicado pela razão da taxa de bit e largura de banda. Essa última equação é equivalente à eficiência espectral em bps/Hz. A eficiência espectral depende do número de bits por subportadora b , o qual

é determinado pelo tamanho da constelação, a taxa de codificação R e a duração de símbolo T_s .

A vantagem da utilização de códigos corretores de erro em sistemas de comunicação fica visível ao realizar simulações (via programa de computador) do enlace completo do sistema. Em muitos casos, a validação e análise de algoritmos de decodificação dependem de resultados de simulações.

Nesse item, será mostrado o método de simulação de Monte Carlo para sistemas de comunicação [20]. A Figura 4 já mostrada anteriormente ilustra todas as etapas do processo. A fonte gera a mensagem binária a ser transmitida. O codificador de canal codifica a mensagem na palavra-código e a entrega ao modulador, que simplesmente mapeia a palavra-código obtida no vetor a ser transmitido pelo canal.

Após a passagem pelo canal, a palavra-código está adicionada de ruído e o receptor pode realizar dois tipos de decisão: abrupta ou suave.

O sistema é dito com decisão abrupta quando o demodulador passa ao decodificador somente a informação do sinal do símbolo. O sistema utiliza decisão suave quando o demodulador, além do sinal, passa ao decodificador a informação de confiabilidade do símbolo recebido. Dessa forma, na decisão abrupta, a palavra-código é re-mapeada em 0 ou 1 usando de forma contrária, a mesma regra de mapeamento. Em seguida, os bits atingem o decodificador com capacidade de correção de erro. Já na decisão suave, o demodulador realiza a decisão do vetor recebido utilizando níveis de quantização (oito em geral) e obtém as métricas que serão passadas ao decodificador para a estimativa dos bits enviados.

Para a simulação de um sistema codificado, os dados de entrada são E_b/N_0 (dB) ou SNR (dB) e o número de transmissões para a obtenção de uma boa amostragem estatística de erros. Como resultado da simulação, tem-se o número de bits errados em função de E_b/N_0 (dB) ou SNR (dB) e a curva de desempenho do sistema pode ser traçada.

É importante ressaltar que, para a obtenção de todas as curvas, o dado de entrada é a relação sinal-ruído (SNR) obtida pela razão de potência do sinal transmitido pelo canal (P_s) e do sinal de ruído (P_r), como mostrado na equação (3.39).

$$SNR|_{dB} = 10 \log \left(\frac{P_s}{\sigma^2 P_r} \right) \quad (3.39)$$

É necessário, portanto, obter a variância do ruído (σ^2) a partir do SNR conhecido, como em (3.40) e (3.41). O desvio padrão (σ) é fator multiplicativo do sinal de ruído e é extremamente importante na obtenção das curvas de desempenho.

$$10^{\frac{SNR|_{dB}}{10}} = \frac{P_s}{\sigma^2 P_r} \quad (3.40)$$

$$\sigma = \sqrt{\frac{\frac{P_s}{P_r}}{10^{\frac{SNR|_{dB}}{10}}}} \quad (3.41)$$

Para o levantamento da curva de taxa de erro de bit (BER) por relação sinal-ruído por bit, basta encontrar E_b/N_0 a partir da equação (3.38).

O próximo capítulo descreverá separadamente a metodologia para a obtenção dos resultados de cada sistema sem e com codificação.

4 METODOLOGIA

O OFDM é um sistema com múltiplas portadoras em que uma parte delas pode ser afetada por ruído. Os códigos corretores de erro podem ser aplicados a esse sistema a fim de melhorar seu desempenho.

Nesse capítulo, os sistemas de transmissão/recepção utilizando OFDM sem codificação e com codificação de Reed-Solomon, convolucional e turbo são apresentados. A metodologia e os resultados obtidos são mostrados para o canal com AWGN e para o canal com ruído medido da rede elétrica.

4.1 SISTEMA DE TRANSMISSÃO OFDM SEM CODIFICAÇÃO

Em um sistema de transmissão OFDM, como mostrado na Figura 11, os bits a serem transmitidos são alocados a N subcanais e são mapeados em símbolos complexos QAM. Esses valores complexos junto ao seu conjugado simétrico são então enviados para a operação de IFFT, formando $2N$ valores. O vetor resultante é convertido em um sinal contínuo no tempo através de um conversor digital para analógico (D/A) para ser transmitido pelo canal. Durante a transmissão, esse sinal estará sujeito a ruído, o qual causará erros no sistema.

Nas simulações, 256 subcanais são considerados e 4 bits de entrada (b_i , onde i indica o número do subcanal) são alocados a cada subcanal. Um mapeador 16-QAM e um operador IFFT de comprimento 512 são adotados. Os conversores D/A e A/D são utilizados em situações práticas e como simulações por computador são realizadas nesse trabalho, não são empregados nesse estudo.

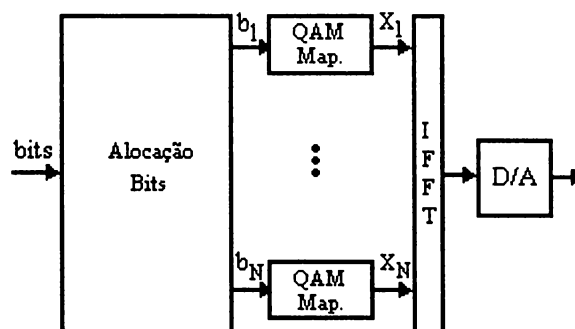


Figura 11: Sistema de transmissão OFDM sem codificação

A Figura 12 mostra o diagrama em blocos da recepção. Após a conversão analógico-digital (A/D), o sinal recebido atinge o operador FFT, fornecendo Y_i valores complexos. Em seguida, o sinal resultante é convertido em bits pelos re-mapeadores QAM que são as entradas para um conversor paralelo-serial.

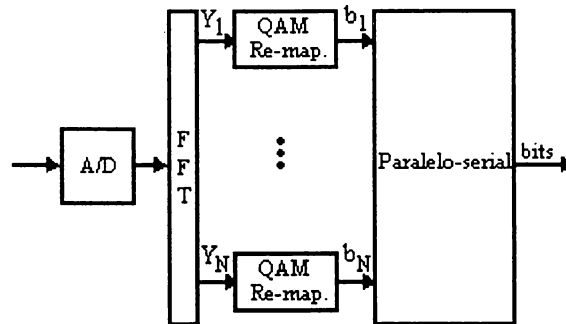


Figura 12: Sistema de recepção OFDM sem codificação

4.2 CÓDIGOS DE REED-SOLOMON EM SISTEMAS OFDM

Os códigos de Reed-Solomon, como descritos no capítulo 3, são códigos BCH não-binários utilizados em diversas aplicações. No padrão ADSL [5], os códigos RS em $GF(256)$ são recomendados, ou seja, cada símbolo do código é um byte. Um código RS de comprimento de bloco n e de k símbolos de informação é denotado por $RS(n,k)$. Ele possui redundância $r = n - k$ e tem capacidade de correção de até t símbolos errados, onde $r = 2t$.

Nas simulações apresentadas nesse artigo, é adotado o código $RS(255,216)$ em $GF(256)$ como em [29] e é realizada uma decodificação em decisão abrupta.

O sistema de transmissão está ilustrado na Figura 13. Os bits de informação atravessam o codificador RS e são alocados em 256 subcanais (N). No caso da modulação 16-QAM, 4 bits ($b_i, i=1...N$) são inseridos em cada subcanal. Em seguida, um mapeamento natural QAM é aplicado ao longo desses subcanais, formando X_i símbolos complexos. O resultado é a entrada para um operador IFFT com 512 pontos.

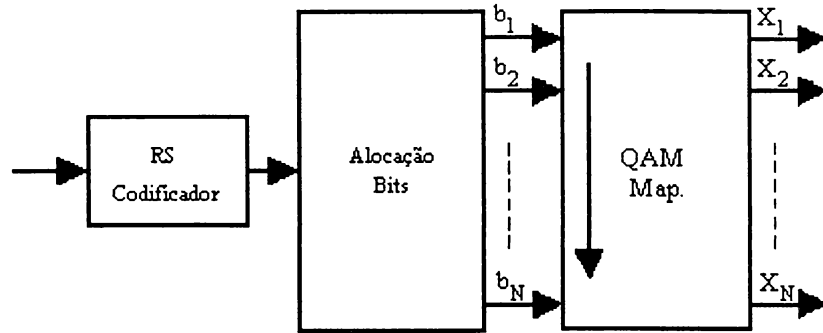


Figura 13: Codificador RS no sistema OFDM

A Figura 14 mostra o esquema para a decodificação de Reed-Solomon. Na recepção, a operação de FFT é realizada e os Y_i símbolos QAM são re-mapeados. Os bits em cada subcanal (b_i) passam por um conversor paralelo-serial e atingem o decodificador RS.

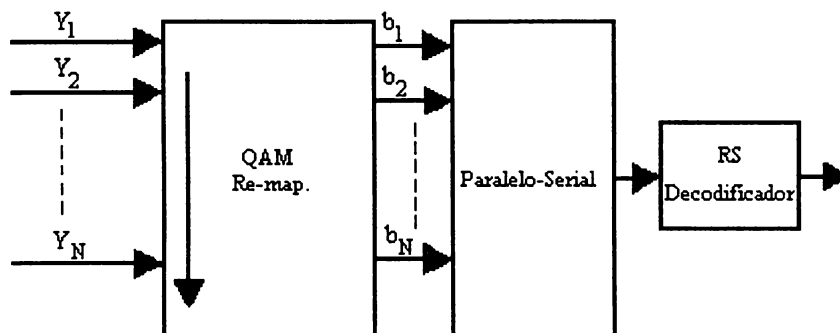


Figura 14: Decodificador RS no sistema OFDM

4.3 CÓDIGOS CONVOLUCIONAIS EM SISTEMAS OFDM

Uma alternativa para a codificação em treliça é a utilização de um código convolucional binário junto a um esquema de modulação não-binária, como o QAM [1,30]. Os dados binários de entrada são convertidos em símbolos QAM de acordo com um mapeamento Gray. Para o caso de 16-QAM, as componentes em fase e quadratura são tratadas separadamente como sinais PAM (*Phase Amplitude Modulation*) de 4 níveis, determinadas por dois bits b_0 e b_1 , como mostrado na Figura 15. Como exemplo, se os bits 1011 são alocados em um determinado

subcanal, o mapeador Gray deverá traduzir os dois primeiros bits (10) para +3 e os dois últimos (11) para +1, formando o símbolo complexo $3+j$.

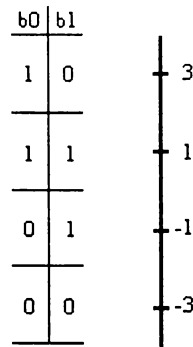


Figura 15: Mapeamento Gray de dois bits em PAM de 4 níveis

Na recepção, os símbolos QAM devem ser transformados em valores unidimensionais, os quais caracterizam as métricas que serão quantizadas em 8 níveis para o decodificador de Viterbi. No caso do 16-QAM, os valores em fase e quadratura são tratados independentemente como sinais PAM de 4 níveis, que são re-mapeados em duas métricas, como mostrado na Figura 16. Supondo que o símbolo complexo $3+j$ é recebido. A parte real deverá produzir duas métricas de acordo com as retas da Figura 16. Nesse caso, ela é traduzida em dois valores: +3 e -1. A componente imaginária também fornecerá dois valores: +1 e +1. São essas métricas (+3 -1 +1 +1) que, após quantizadas, são enviadas para a decodificação.

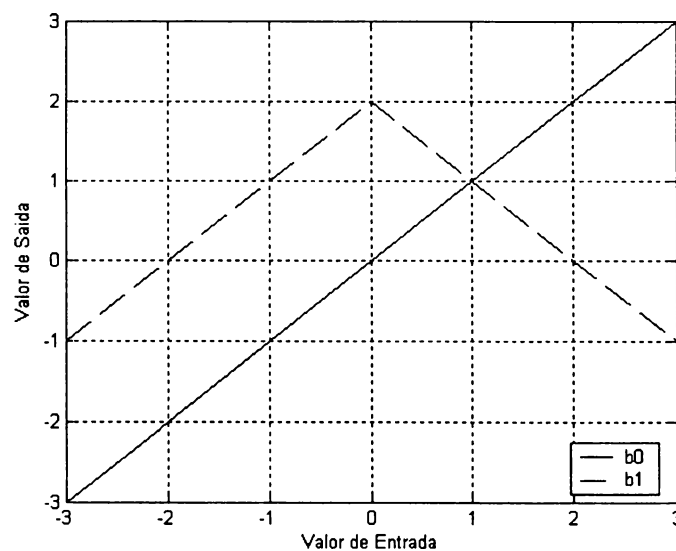


Figura 16: Re-mapeamento de 4 níveis PAM em 2 métricas

Nessa análise, foi empregado um código convolucional de taxa $\frac{1}{2}$, *constraint length* 7 e vetor gerador $g=[133,171]$. Esse código é um dos mais utilizados, como citado em [1].

A Figura 17 mostra o esquema para transmissão. Os bits de informação atravessam o codificador convolucional e são alocados em 256 subcanais, formando 4 bits (b_i) em cada subcanal. Esses bits servem de entrada para um mapeador Gray, resultando em símbolos complexos (X_i) que são enviados a um bloco de IFFT com 512 pontos.

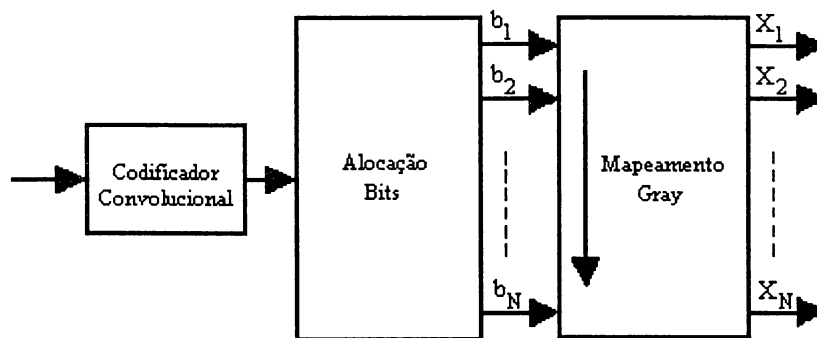


Figura 17: Codificador convolucional no sistema OFDM

Na recepção, como mostrado na Figura 18, o sinal recebido é a entrada para o operador de FFT. Em seguida, os símbolos complexos (Y_i) são separados em componentes em fase e quadratura que são tratadas independentemente para fornecer as métricas binárias para o decodificador de Viterbi. No caso da modulação 16-QAM, quatro métricas são obtidas. Essas métricas são então quantizadas em níveis de 3 bits e são as entradas para o decodificador binário de Viterbi, que realizará uma decisão suave estimando os bits enviados.

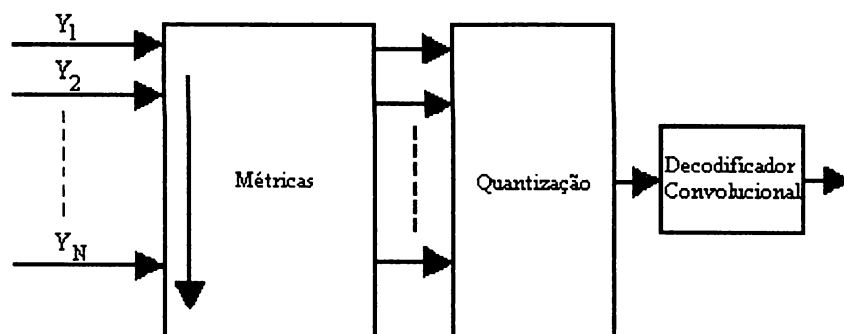


Figura 18: Decodificador convolucional no sistema OFDM

4.4 CÓDIGOS TURBO EM SISTEMAS OFDM

O desempenho dos códigos turbo [7] tem se mostrado próximo ao limite de Shannon para canais AWGN. Um codificador binário padrão é formado por dois códigos convolucionais sistemáticos recursivos separados por um entrelaçador. Um algoritmo de decodificação MAP (*maximum a posteriori*) é utilizado em um processo iterativo para fornecer o melhor desempenho do sistema. Nesse trabalho, um código turbo binário de taxa $\frac{1}{2}$ e vetor gerador $g=[11111; 10001]$ é utilizado para codificar os bits de informação. Foram realizadas 3 iterações durante a decodificação, o que representa um compromisso entre qualidade (menos erros) e velocidade.

Vários esquemas de modulação codificada em treliça (TCM) para os códigos turbo, os quais envolvem a associação de codificação convolucional com um esquema de modulação M-ária, foram investigados [8], [9]. Nessas referências, os códigos turbo são associados à modulação QAM e a sua performance também se aproxima do limite de Shannon. Os códigos turbo podem ser aplicados ao sistema DMT [31], [32]. A Figura 19 mostra o esquema para a codificação. As saídas do codificador turbo são demultiplexadas e separadas em componentes sistemáticas e de paridade. Uma função de *puncturing* (P) é inserida para possibilitar a obtenção de uma família maior de códigos, com diferentes taxas R. Nas simulações, o *puncturing* foi utilizado a fim de se obter uma taxa de codificação igual a $\frac{1}{2}$, transmitindo-se a informação sistemática e o bit ímpar de paridade do RSC1 e então a próxima informação sistemática e o bit par de paridade do RSC2. A fim de obter símbolos afetados por ruídos não-correlacionados na entrada do decodificador turbo, os bits passam por um entrelaçador (I). Nesse trabalho, um entrelaçador pseudo-randômico de tamanho 512 é adotado para embaralhar os bits codificados. Esses bits são então alocados em 256 subcanais, cada um contendo 4 bits e são as entradas para os mapeadores QAM Gray. As saídas dos mapeadores são convertidas em sinal de transmissão através da operação de IFFT com 512 pontos.

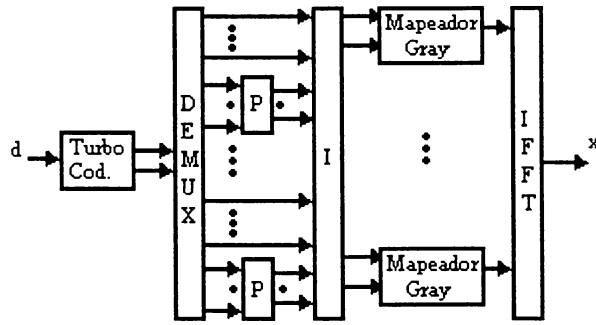


Figura 19: Codificador turbo no sistema OFDM

No receptor, como na Figura 20, obtém-se a FFT do sinal e o bloco receptor obtém a *a posteriori log-likelihood do canal* (C-LLR) dos símbolos recebidos [8,31], de maneira análoga à obtenção das métricas usadas na decodificação convolucional. Em seguida, os C-LLRs (λ) atravessam um de-entrelaçador (I^{-1}), são demultiplexados e usados como entrada para o decodificador turbo padrão.

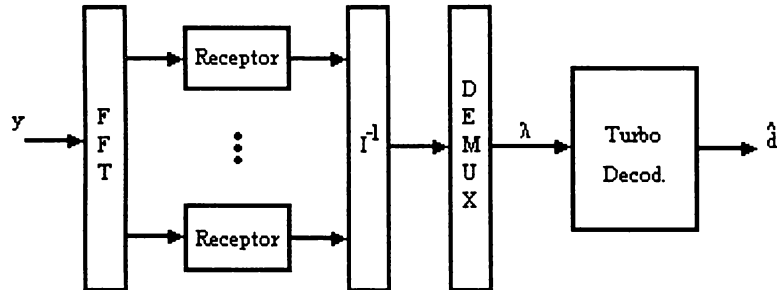


Figura 20: Decodificador turbo no sistema OFDM

4.5 SIMULAÇÕES

A simulação do sistema de comunicação foi realizada através do programa Matlab. Três canais de comunicação foram considerados nas simulações realizadas nesse trabalho: um canal com ruído gaussiano branco aditivo (AWGN), um canal com ruído colorido e um canal com ruído medido da própria rede elétrica do laboratório.

O canal AWGN é bem conhecido e pode ser facilmente implementado com a função *randn()* do Matlab que gera um ruído pseudo-aleatório branco com distribuição gaussiana, como mostrado no anexo I. A fim de verificar a influência do ruído colorido, dois tipos diferentes de ruído foram considerados. O primeiro foi obtido filtrando-se o ruído AWGN através de um filtro passa-faixa digital *butterworth* de ordem 5 com frequências de corte em 12kHz e 16kHz. O segundo também foi obtido através da filtragem do ruído AWGN utilizando-se o mesmo filtro com largura de banda menor (15kHz a 16kHz).

Um modelamento do canal da rede elétrica não é bem estabelecido, embora existam alguns trabalhos a respeito do tema [33, 34]. Há vários fatores que contribuem para essa dificuldade de modelagem: variações em frequência e atenuações variantes no tempo que ocorrem no meio, dependência do modelo do canal com a localização, topologia de rede e cargas conectadas, alta interferência devido a cargas ruidosas, elevado ruído colorido e ruído impulsivo. Nesse trabalho, somente o ruído de interferência foi considerado e a função de transferência do sistema foi suposta como um passa-tudo em série com um passa-altas para reproduzir o acoplamento do transceptor com a rede. Dessa forma, embora se tenha um cenário menos realista, o efeito do ruído em diferentes códigos corretores de erro pode ser mais bem isolado, permitindo uma comparação controlada. Optou-se, também, por realizar medições para a obtenção do ruído da rede elétrica para se avaliar como um caso específico com ruído real se afasta de uma modelagem genérica com ruído sintético branco gaussiano. Foi utilizado um conversor analógico-digital de 12 bits com frequência de amostragem em 40kHz para obter 60.000 amostras de ruído da rede elétrica. Um filtro passa-altas com frequência de corte em 200 Hz e um filtro de *anti-aliasing* (passa-baixas) em 20 kHz foram utilizados. Além

da medição direta da rede elétrica, foram feitas mais duas medições adicionando como interferência um sinal senoidal na frequência de 19kHz, com amplitudes arbitrárias de 1V e 6V injetados por um gerador de sinais acoplado através de um transformador, com o intuito de simular a interferência de uma onda contínua em uma frequência fixa.

Os sistemas sem codificação e com diferentes tipos de codificação (convolucional, Reed-Solomon e turbo) foram implementados e os desempenhos foram analisados para o caso de um canal com AWGN para a modulação OFDM com 16-QAM. A fim de verificar a influência do ruído colorido, o qual também está presente na rede elétrica, simularam-se os mesmos sistemas codificados para esse tipo de canal. Em uma segunda etapa, esses mesmos sistemas foram submetidos ao ruído típico medido da rede elétrica com e sem a interferência de 19kHz. A largura de banda considerada em todos os casos foi de 20kHz e a transmissão ocorreu a partir de 10kHz. Foi adotado um filtro digital *butterworth* passa-altas de ordem 5 e com frequência de corte em 6kHz na entrada do receptor a fim de eliminar resquícios da frequência de 60 Hz e suas harmônicas. Os gráficos da taxa de erro de bit (BER) em função da relação sinal-ruído (SNR) foram obtidos utilizando-se o método de Monte Carlo através da simulação do enlace completo dos sistemas. Para tanto, o dado de entrada é a relação sinal-ruído obtida pela razão de potências do sinal de transmissão e do sinal do ruído. Como resultado, tem-se a BER para transmissões até a ocorrência de 500 bits errados.

A Figura 21 mostra a densidade espectral de potência para uma mesma relação sinal-ruído para o canal AWGN e para os ruídos coloridos, um filtrado por um passa-faixa em 12kHz e 16kHz e outro por um passa-faixa em 15kHz e 16kHz. A Figura 22 exhibe a densidade espectral de potência do ruído medido da rede elétrica e do ruído AWGN para uma mesma potência e para ambos filtrados em 6kHz. Enquanto o ruído branco exhibe amplitudes praticamente constantes ao longo das frequências, o ruído medido da rede elétrica contém interferências, ruído colorido e ruído impulsivo.

O item 4.6 descreve o fluxograma da metodologia para a simulação.

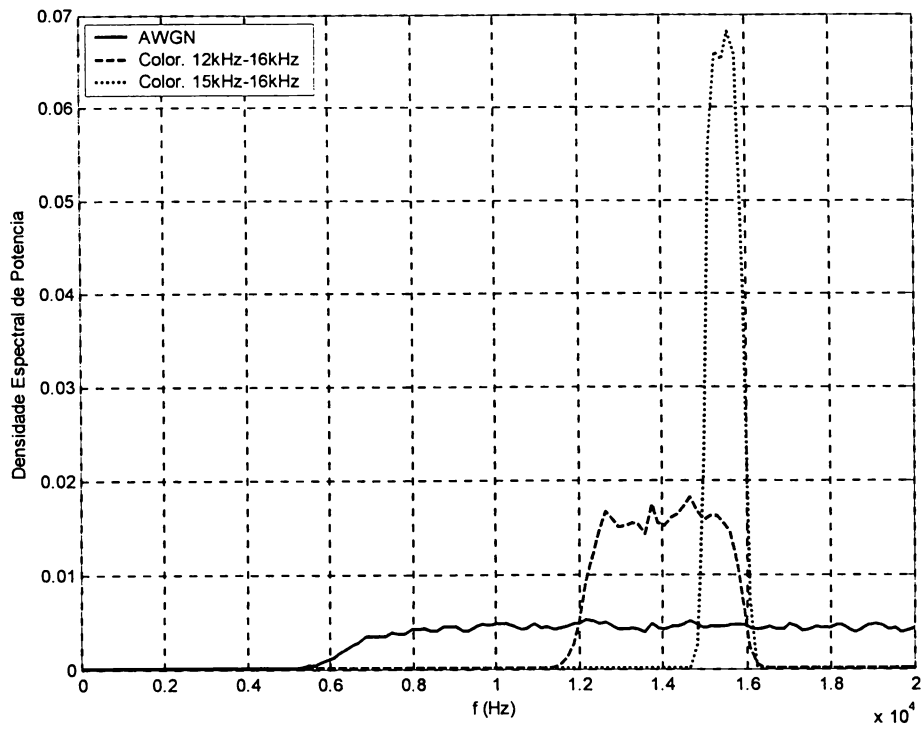


Figura 21: Densidade Espectral de Potência dos ruídos coloridos e AWGN

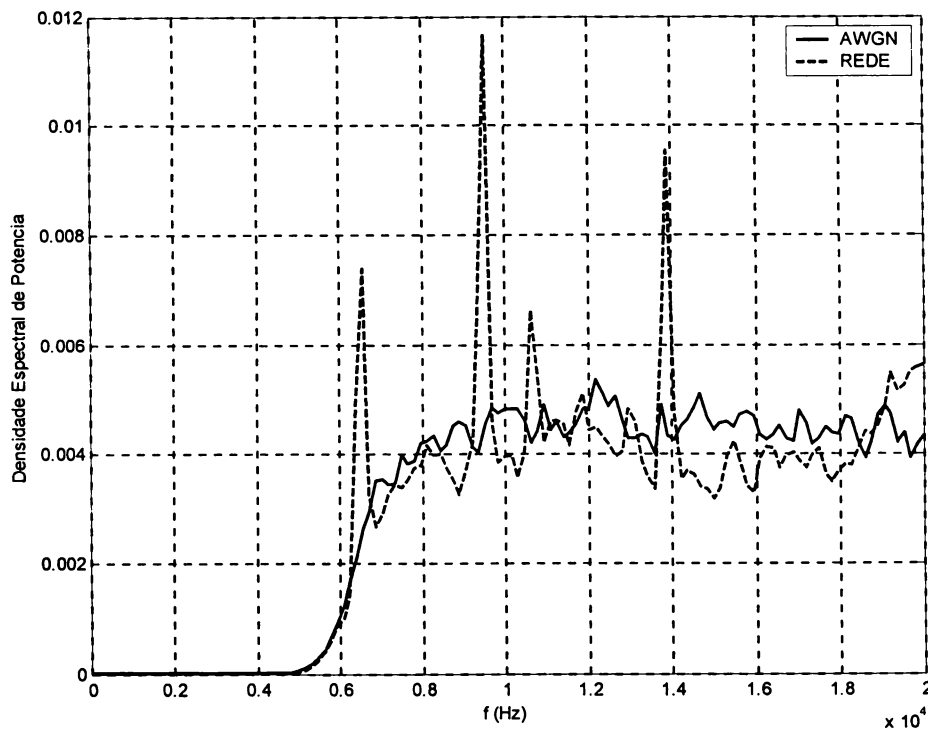


Figura 22: Densidade Espectral de Potência do ruído medido PLC e do AWGN

4.6 FLUXOGRAMA DA METODOLOGIA

A simulação dos sistemas com controle de erro foi realizada através do método de Monte Carlo, como já descrito no item 3.8. A Figura 23 mostra o fluxograma da metodologia para a obtenção das curvas de desempenho da BER em função do SNR.

Para cada valor de SNR, como no anexo I, o transmissor OFDM gera os bits de informação, codifica-os usando um dos três codificadores adotados (RS, convolucional ou turbo), distribui os dados nos subcanais, realiza a modulação, no caso do código RS, ou mapeamento Gray para os códigos turbo e convolucional e em seguida a operação de IFFT. Tem-se então o sinal a ser transmitido pelo canal. Como na entrada da recepção há a necessidade de um filtro passa-altas com frequência de corte em 6kHz para eliminar resquícios da frequência de 60 Hz e suas harmônicas, optou-se por filtrar o sinal transmitido e o sinal de ruído separadamente a fim de se obter a potência do sinal de transmissão (P_{sf}) e de ruído filtrados (P_{rf}). Conhecendo-se o SNR e as potências dos sinais, calcula-se o valor de σ através da equação (3.41) que produzirá a relação sinal-ruído desejada. Em seguida, soma-se o sinal filtrado com o produto de σ e do ruído filtrado.

No receptor OFDM, o processo inverso é realizado. A operação de FFT é aplicada e o sinal é demodulado no caso da codificação de Reed-Solomon. Para os codificadores convolucional e turbo, é eliminada a presença do demodulador* para que haja a decodificação suave. Em seguida, contam-se os erros de bits ocorridos comparando-se a seqüência gerada pela fonte binária e a seqüência decodificada. O sistema verifica então se o número de erros é menor que 500 bits errados. Em caso afirmativo, as iterações continuam para um mesmo valor de SNR. Caso contrário, o *loop* é interrompido, o valor de SNR é incrementado de 1 unidade e uma nova iteração é iniciada até o contador atingir 500 erros. Esse processo repete-se até que o valor de SNR seja igual ao do SNR final.

O resultado da simulação são valores da taxa de erro de bit para cada valor da relação sinal-ruído (SNR). Para a obtenção da curva BER X E_b/N_0 , basta converter o valor de SNR utilizando a equação (3.38).

$$SNR_i = \frac{E_b}{N_0} \frac{1}{BT_b} = \frac{E_b}{N_0} \frac{bN_s R}{BT_s} \quad (3.38)$$

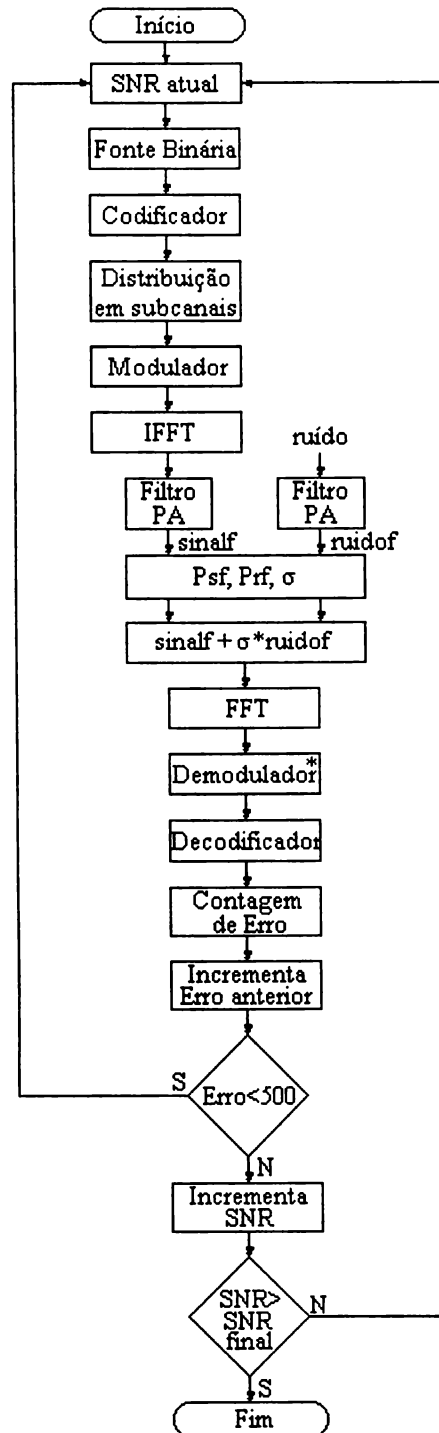
Para uma frequência de amostragem (f_s) de 40 kHz, a largura de banda (B) adotada é igual a 20 kHz, ou seja, metade do valor da frequência de amostragem. Devido à inserção do filtro passa-altas em 6 kHz, essa largura de banda reduz-se a uma fração da largura anterior, passando a 14 kHz. Como é realizada uma transmissão a partir de 10 kHz, o número de portadoras utilizadas reduz-se pela metade. A duração de um símbolo (T_s) é igual à quantidade de símbolos enviados a cada transmissão na saída do operador de IFFT, que é igual ao dobro do número de portadoras pela frequência de amostragem. A equação (4.1) relaciona o SNR com E_b/N_0 para esse caso.

$$SNR = \frac{E_b}{N_0} \frac{b \frac{N_s}{2} R}{14 \cdot \left(2 \cdot N_s \cdot \frac{1}{f_s} \right)} \quad (4.1)$$

Substituindo a frequência de amostragem f_s por 40 kHz e realizando as simplificações necessárias na equação (4.1), obtém-se (4.2).

$$SNR = \frac{E_b}{N_0} \left(b \cdot \frac{1}{2} \cdot R \cdot \frac{20}{14} \right) \quad (4.2)$$

Dessa forma, conhecendo-se a quantidade de bits por subportadora e a taxa do código, pode-se calcular o valor de E_b/N_0 a partir do SNR conhecido e então obter a curva da taxa de erro de bit pela relação sinal-ruído por bit.



* Demodulador presente somente na codificação de Reed-Solomon

Figura 23: Fluxograma da metodologia

5 RESULTADOS

A Figura 24 mostra as curvas teórica e simulada da taxa de erro de bit (BER) em função da relação sinal ruído E_b/N_0 para o caso do sistema sem codificação com modulação OFDM e 16-QAM no canal AWGN. Essas curvas são muito importantes para validar a metodologia apresentada. Deve-se observar que as curvas teórica [1, 23], definida pela equação (5.1), e simulada mostram-se muito próximas.

$$P_b = \frac{1}{k} \left\{ 2 \left(1 - \sqrt{\frac{1}{M}} \right) \operatorname{erfc} \left(\sqrt{\frac{3}{2(M-1)}} k \frac{E_b}{N_0} \right) \left[1 - \frac{1}{2} \left(1 - \frac{1}{\sqrt{M}} \right) \operatorname{erfc} \left(\sqrt{\frac{3}{2(M-1)}} k \frac{E_b}{N_0} \right) \right] \right\} \quad (5.1)$$

Na equação (5.1), P_b é a probabilidade de erro de bit, M é a quantidade de pontos na constelação QAM, k é a quantidade de bits por símbolo e E_b/N_0 é a relação sinal-ruído por bit.

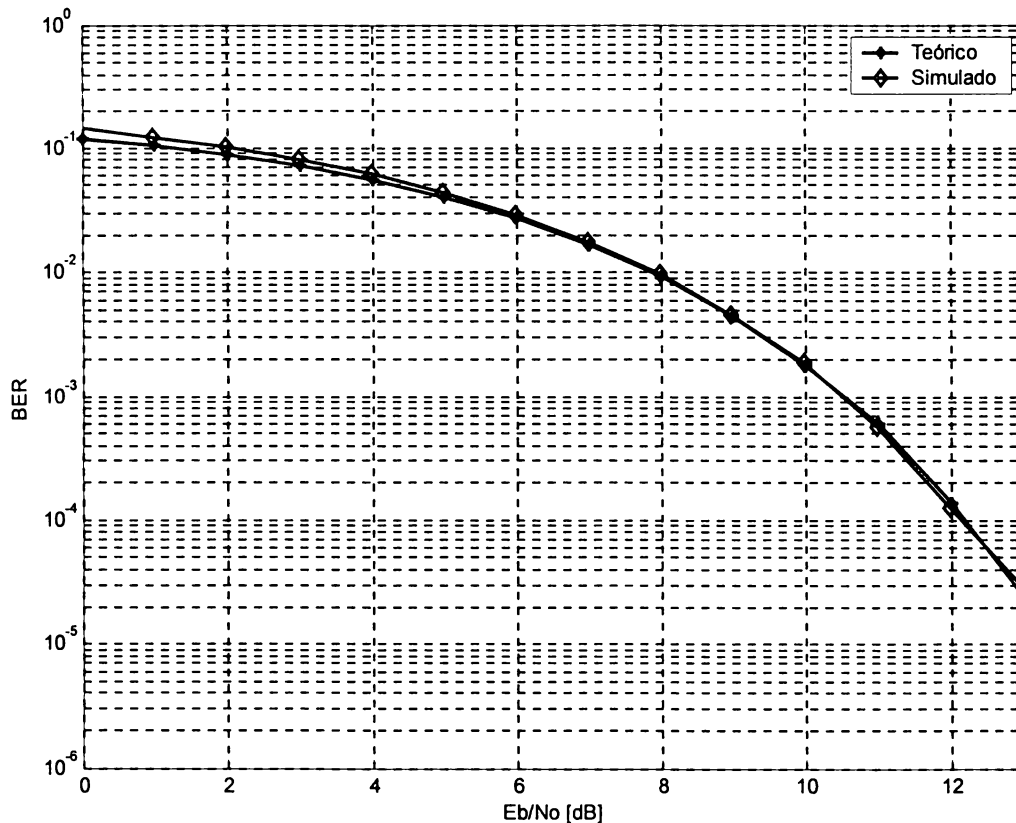


Figura 24: Desempenhos teórico e simulado no canal AWGN

A Figura 25 mostra as curvas da taxa de erro de bit em função da relação sinal-ruído E_b/N_0 dos sistemas OFDM com modulação 16-QAM em um canal AWGN. Essas curvas obtidas estão de acordo com as curvas apresentadas na literatura [1]. Observa-se a superioridade dos códigos turbo em canais com ruído AWGN. Para 16-QAM, o sistema com codificação turbo apresenta-se superior quando comparado com os demais. Em uma taxa de erro de bit (BER) de 10^{-6} , o ganho de codificação turbo em relação ao produzido pelo código de Reed-Solomon é de 7 dB. Quando comparado com o código convolucional, o ganho é de 2 dB.

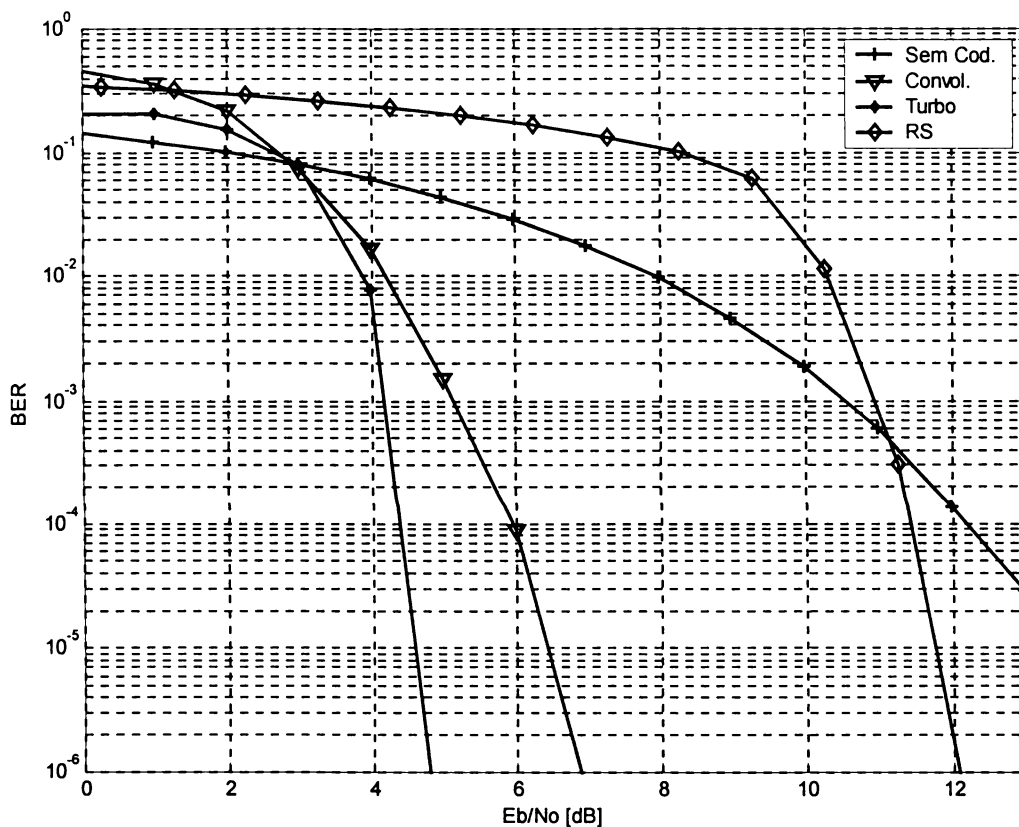


Figura 25: Desempenho dos sistemas para o canal com ruído AWGN

A fim de verificar o efeito do ruído colorido, os mesmos sistemas foram simulados em canais AWGN filtrados. Faz-se necessário definir o parâmetro J_0 que representa a densidade espectral de potência média de ruído equivalente para que seja possível apresentar as curvas BER versus E_b/J_0 . A relação E_b/J_0 pode ser obtida da mesma forma que E_b/N_0 . Os desempenhos dos sistemas codificados para

o canal com ruído colorido, filtrado com passa-faixa em 12 kHz a 16 kHz, estão mostrados na Figura 26. Como uma quantidade menor de subportadoras é afetada pelo ruído colorido, nota-se melhora de desempenho para todos os códigos. Percebe-se ainda que em uma BER de 10^{-6} , o código convolucional apresenta um ganho menor que 0,5 dB quando comparado com o ganho de codificação do sistema convolucional para o canal AWGN (Figura 25). No caso do código turbo, esse ganho é igual a 3 dB e para o código de Reed-Solomon atinge 5 dB. O código turbo continua apresentando-se superior aos demais.

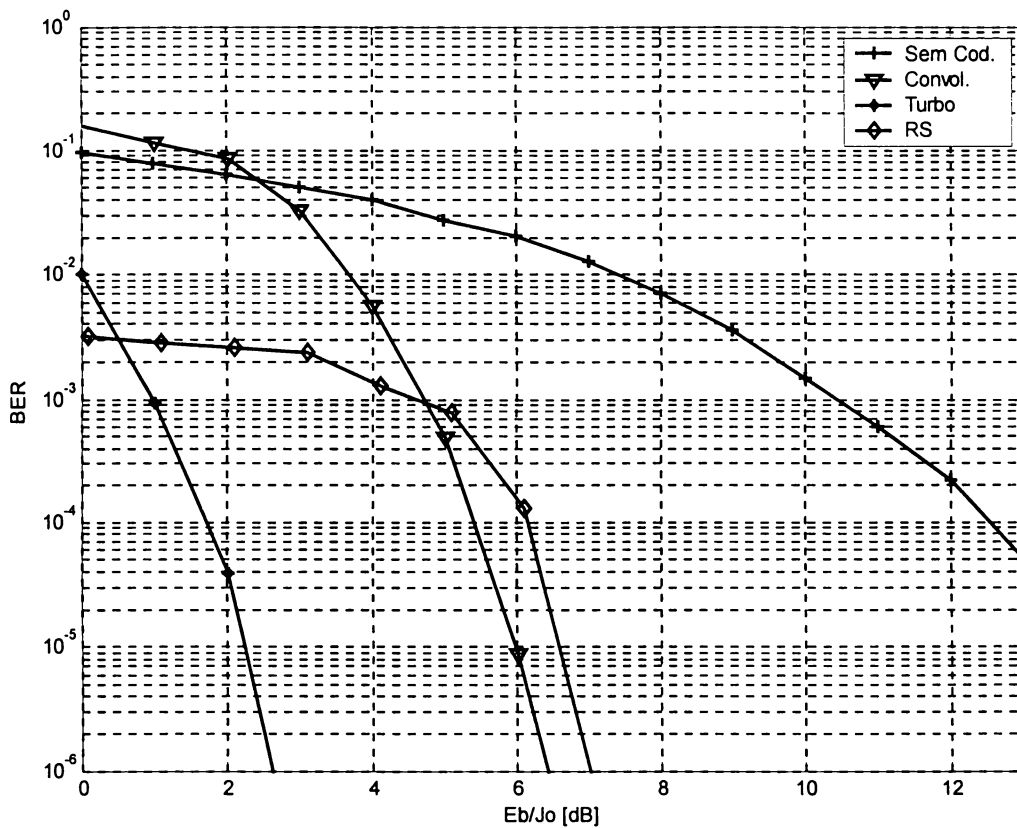


Figura 26: Desempenho dos sistemas com ruído colorido (12kHz a 16kHz)

Essa diferença em termos de ganho ocorre devido às distintas propriedades dos códigos corretores de erro. Como o código de Reed-Solomon é um código não-binário (cada símbolo corresponde a 8 bits nesse estudo), quando o ruído atinge uma quantidade menor de subportadoras, apenas alguns símbolos serão afetados, o

que aumenta o desempenho do sistema. A Figura 27 mostra os desempenhos para um ruído colorido de menor largura de banda, filtrado com passa-faixas em 15 kHz a 16 kHz, que atinge uma faixa muito pequena de subportadoras. Nesse caso, o código de Reed-Solomon mostra-se superior aos demais, pois apenas poucos símbolos são afetados pelo ruído.

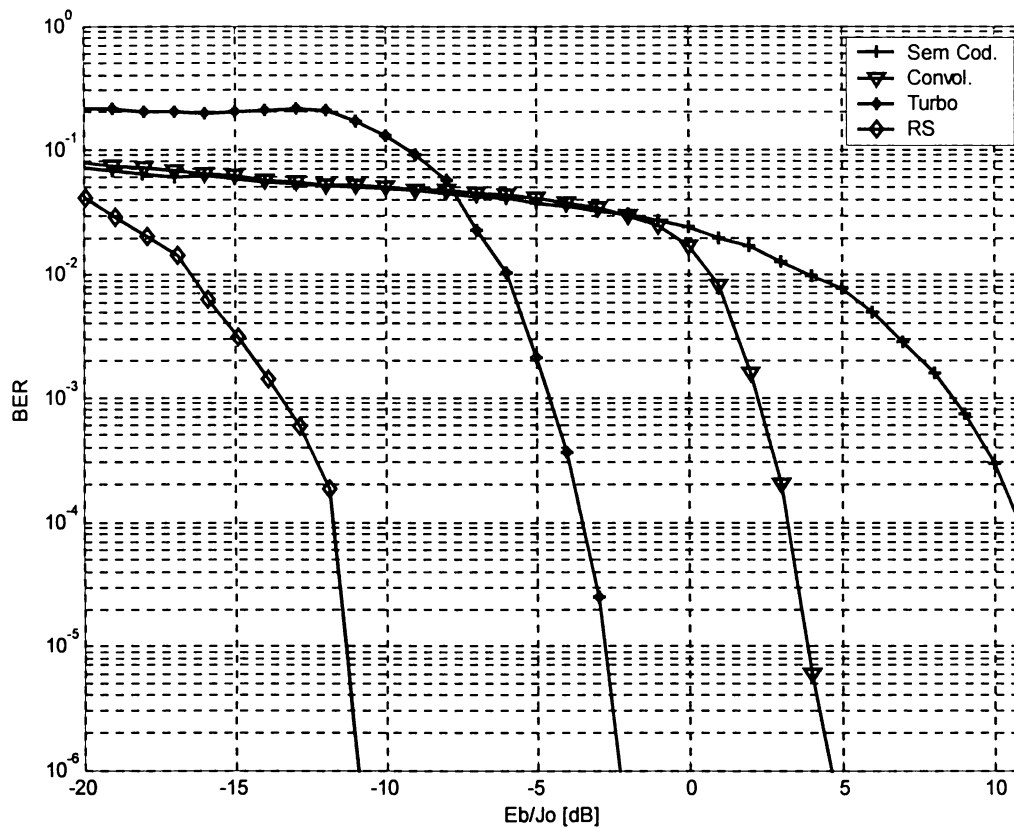


Figura 27: Desempenho dos sistemas com ruído colorido (15kHz a 16kHz)

Em uma segunda etapa, os mesmos sistemas sem codificação e com codificação foram submetidos ao ruído medido da rede elétrica da cidade de Curitiba. A Figura 28 exibe o desempenho desses sistemas. Verifica-se que o código turbo mantém um desempenho superior quando comparado com o sistema sem codificação e apresenta-se melhor que o código convolucional e de Reed-Solomon. Nesse caso, em uma BER de 10^{-6} , o ganho de codificação turbo é de 6 dB em relação ao código de Reed-Solomon e de 2 dB em relação ao código

convolucional. Nota-se que o ganho de codificação para o canal da rede elétrica é inferior ao ganho quando o sistema é submetido ao canal AWGN.

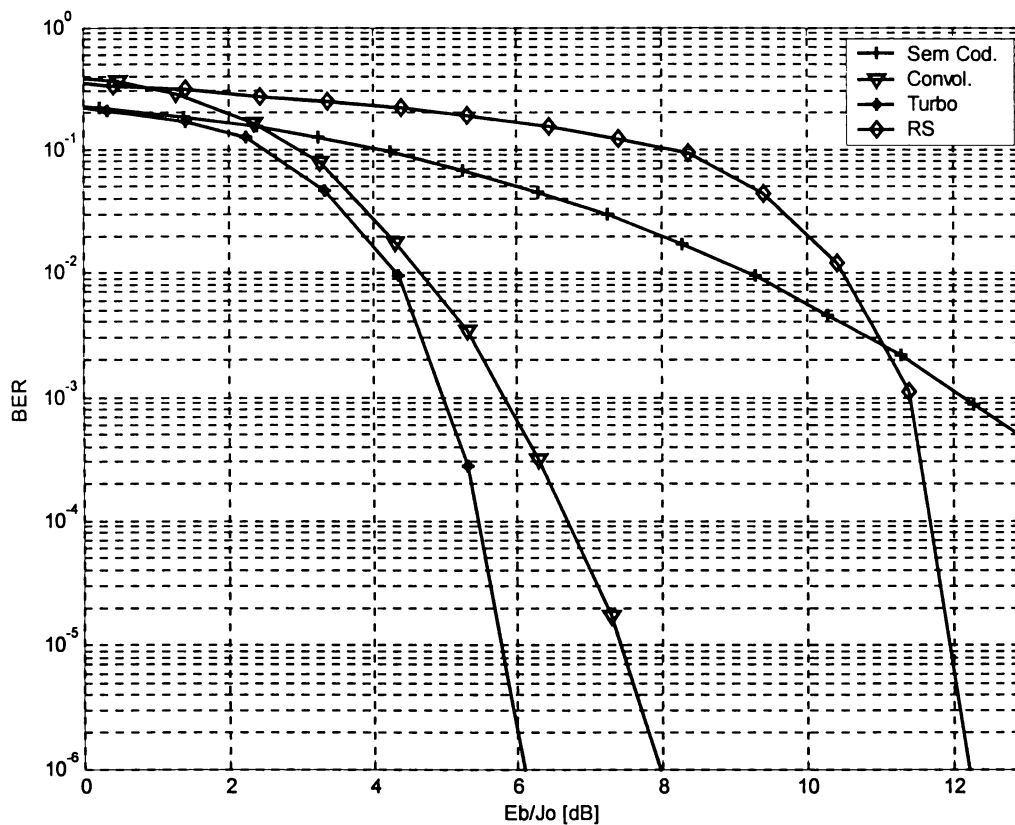


Figura 28: Desempenho dos sistemas com ruído medido da rede elétrica

A Figura 29 mostra uma comparação entre o desempenho do código convolucional para o canal AWGN e para a rede elétrica sem sinal de interferência aplicado e com sinal de amplitudes arbitrárias em 1V e 6V. À medida que o sinal de tensão aumenta, o desempenho do sistema com ruído da rede elétrica torna-se inferior. Nota-se que a diferença de relação sinal-ruído por bit para o canal AWGN em 10^{-5} é de 1 dB em relação ao canal da rede elétrica, o que afasta o AWGN de um bom modelo para este caso.

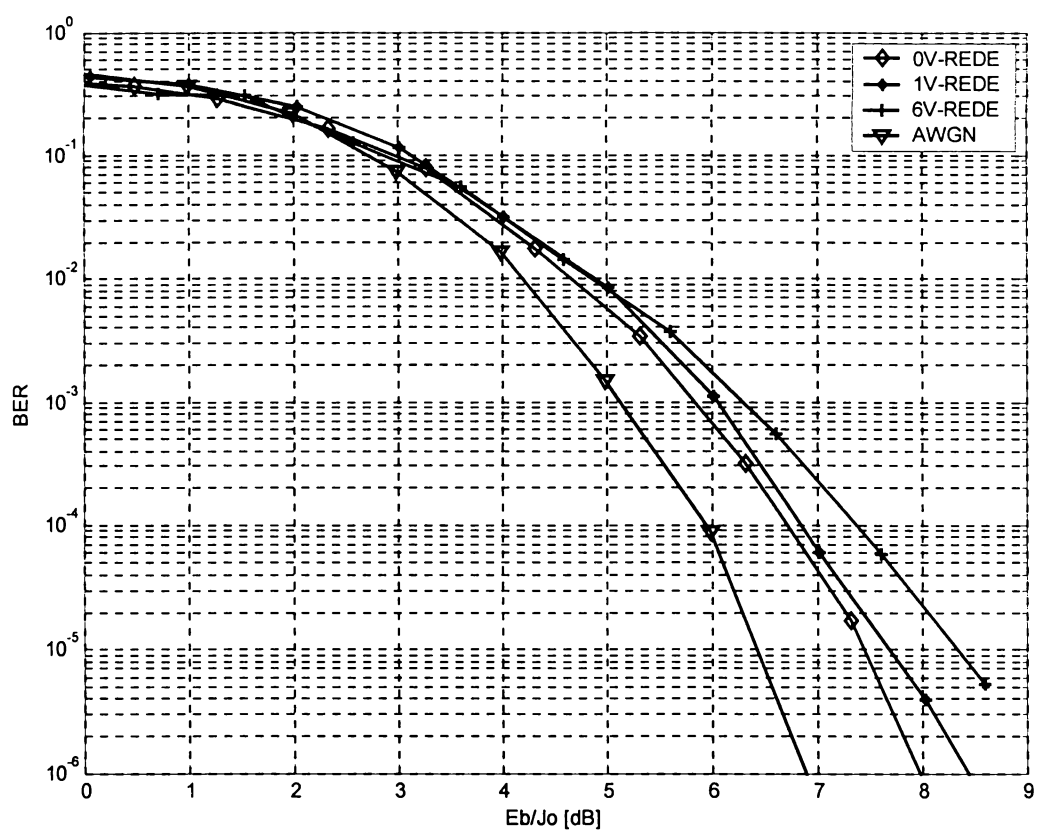


Figura 29: Comparação do desempenho dos sistemas com convolucional

A Figura 30 mostra um comparativo entre o canal AWGN e a rede elétrica para um sistema codificado com código turbo. Os desempenhos dos sistemas são praticamente os mesmos acima para uma relação sinal-ruído E_b/J_0 igual a 3 dB, todavia para maiores valores de E_b/J_0 , o canal AWGN apresenta menos erros.

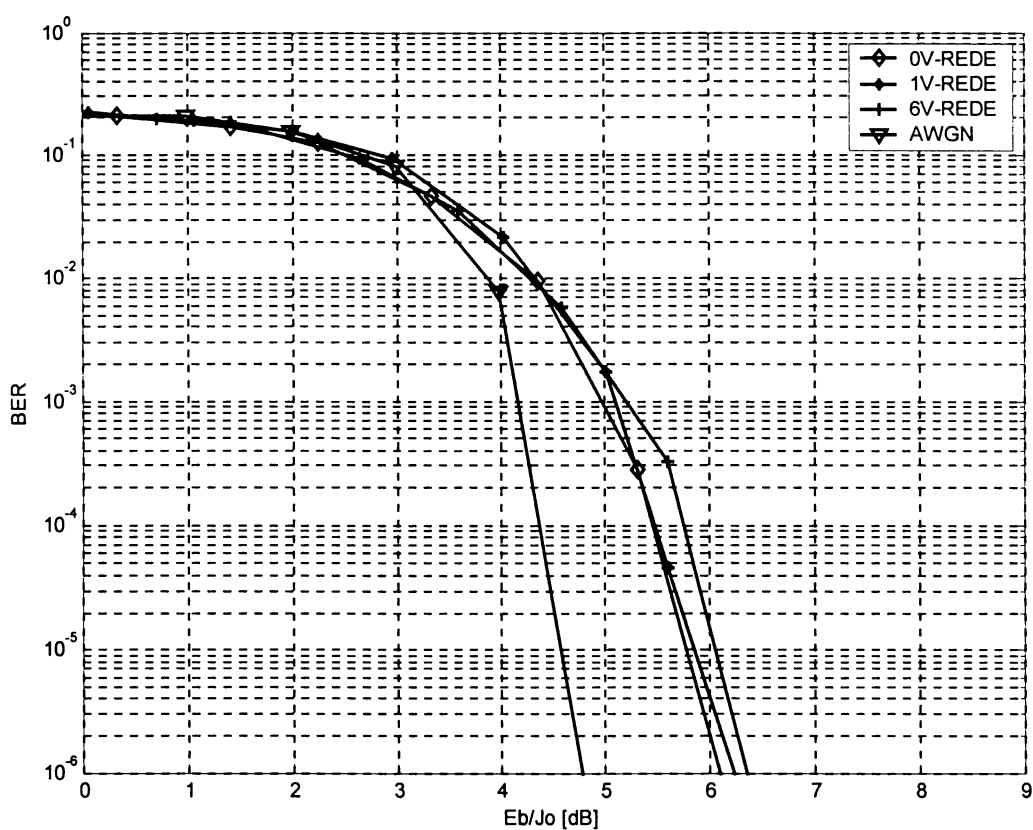


Figura 30: Comparação do desempenho dos sistemas com código turbo

A Figura 31 mostra a comparação do sistema com AWGN e com ruído da rede elétrica para o código de Reed-Solomon. Nesse caso, as curvas de erro dos sistemas resultaram muito próximas. O desempenho no canal AWGN afasta-se do desempenho obtido quando o sistema é submetido ao canal com ruído medido. O aumento do sinal de tensão aplicada na rede elétrica causa uma pequena variação na curva de desempenho.

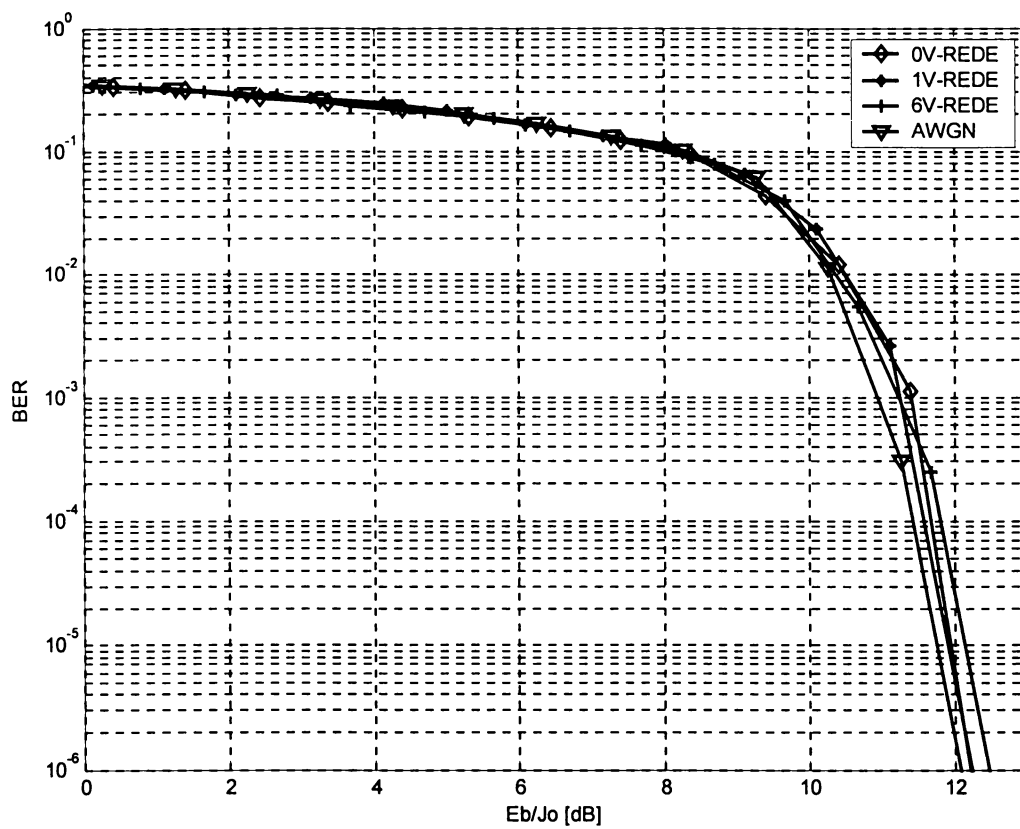


Figura 31: Comparação do desempenho dos sistemas com código RS

6 CONSIDERAÇÕES FINAIS E CONCLUSÕES

6.1 CONSIDERAÇÕES FINAIS

O problema central nas comunicações de dados há um certo tempo era uma transmissão eficiente por canais afetados somente por ruído gaussiano branco aditivo (AWGN). O aumento da demanda de serviços de dados em alta velocidade para as comunicações digitais tem incentivado o uso de meios de comunicação que não foram a princípio projetados para esse propósito e, portanto, apresentam desafios formidáveis aos projetistas de sistemas.

No passado, a rede de telefonia pública comutada migrou de um canal puro de voz para um meio capaz de fornecer vários serviços digitais. Uma transformação semelhante ocorre com a rede de energia elétrica, a qual era utilizada originalmente para controle e medição e atualmente é considerada para serviços de dados de alta velocidade.

A rede elétrica é um meio conveniente e barato para a comunicação de dados, uma vez que todas as construções, como casas ou prédios industriais, já estão interligadas por essas redes de energia. Todavia, ela apresenta grandes desafios, pois a comunicação de dados é realizada por um meio designado para o fornecimento de energia elétrica e não para a transmissão de dados.

O mérito desse trabalho é avaliar, para uma transmissão de dados pela rede elétrica, os principais códigos corretores de erro em sistemas OFDM, entre eles o código turbo no intuito de verificar qual deles mais bem se desempenha em um ambiente simplificado que utilizou como interferência aditiva amostras de ruído medido da rede elétrica. Os capítulos 2 e 3 apresentaram o sistema OFDM e os códigos corretores de erro adotados. O capítulo 4 associou os códigos convolucional, de Reed-Solomon e turbo ao sistema OFDM para uma comunicação de dados pela rede elétrica. O capítulo 5 apresentou os resultados obtidos.

6.2 CONCLUSÕES

Um cenário para simulação dos sistemas de transmissão em OFDM em canais ruidosos foi implementado e testado. Os resultados obtidos através da simulação para o canal AWGN estão de acordo com as formulações teóricas. Os desempenhos dos códigos de Reed-Solomon, convolucional e turbo aplicados a um sistema OFDM foram verificados para a transmissão de dados em um canal AWGN, em um canal com ruído colorido e em um canal com ruído medido da rede elétrica.

Os resultados mostraram que os códigos produziram ganhos de codificação quando submetidos ao ruído real da rede elétrica que não foram iguais aos ganhos produzidos no canal AWGN. Tal diferença pode ser explicada pelas interferências presentes no canal da rede elétrica. Isso confirma que uma escolha cuidadosa do código é muito importante para melhores desempenhos e que a análise em cenários mais reais é necessária.

Para os códigos convolucional e turbo, os desempenhos para o canal da rede elétrica considerado mostrou uma diferença de 1 a 1,5 dB para uma BER de 10^{-6} quando comparado ao canal AWGN. Para o código de Reed-Solomon, tal diferença mostrou-se menor e inferior a 0,2 dB. Constatou-se também a superioridade dos códigos turbo em relação aos demais sistemas com e sem codificação, tanto para o canal com AWGN como para o canal da rede elétrica em questão, apresentando um ganho de codificação de cerca de 2 dB em relação ao código convolucional de mesma taxa de codificação. A exceção seria um ruído colorido com largura de banda muito estreita, como é o caso do ruído filtrado por um passa-faixa em 15kHz e 16kHz, o qual afeta poucas portadoras. Nesse caso, o código de Reed-Solomon comportou-se melhor devido a suas propriedades, pois é um código não-binário que trabalha com símbolos (conjuntos de bits), assim quando o ruído atinge poucas subportadoras, apenas alguns símbolos são afetados, o que aumenta o desempenho do sistema.

É interessante observar que os desempenhos dos códigos turbo, convolucional e de Reed-Solomon foram sempre superiores para o canal AWGN, afastando-se do desempenho para o caso do ruído medido da rede elétrica em até 2 dB.

Essa metodologia pode ser facilmente estendida a maiores ordens de frequências (MHz), permitindo a avaliação dos mesmos esquemas de correção de erro onde o ruído pode possuir características diferentes.

6.3 TRABALHOS FUTUROS

Após a avaliação do desempenho dos códigos corretores de erro em sistemas OFDM com modulação 16-QAM, podem-se citar alguns trabalhos futuros na área de comunicação de dados pela rede elétrica:

- incorporação de mais medidas do ruído da rede elétrica em diversos locais e horários diferentes a fim de realizar um estudo estatístico ampliado sobre o desempenho dos códigos;
- avaliação de sistemas com maiores constelações, como 64-QAM e 128-QAM;
- avaliação dos códigos corretores de erro em esquemas adaptativos, uma vez que as características da rede elétrica variam com o tempo devido à remoção ou inserção de cargas elétricas.

Esses esquemas adaptativos podem ser obtidos modificando-se os parâmetros de transmissão:

- nível de potência adaptativa: através do controle da potência, os parâmetros de transmissão são inalterados com as variações do canal, *todavia o nível de transmissão varia com as flutuações do canal;*
- tamanho de constelação adaptativa: modulação adaptativa aumenta a eficiência da transmissão de dados. Monitorando o estado do canal, o tamanho da constelação pode ser variada para adaptar o esquema da modulação ao SNR do canal;
- taxa de código adaptativa: o esquema de codificação pode responder às variações de canal alterando-se a taxa de código.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] R. van Nee e R. Prasad, *OFDM for Wireless Multimedia Communications*, Artech House, 2000.
- [2] J. Abad, et al., Extending the Power Line Lan up to the Neighborhood Transformer, *IEEE Communication Magazine*. p. 64-70, Abr. 2003.
- [3] P. S. Chow, J. M. Cioffi and J. A. C. Bingham, DMT-Based ADSL: Concept, Architecture and Performance, *IEE Colloquium on High speed Access Technology and Services*, 1994.
- [4] I. Kallet, The Multitone Channel, *IEEE Transactions on Communications*, 1989.
- [5] ANSI T1.413: Network and Customer Installation Interfaces - Asymmetric Digital Subscriber Line (ADSL) Metallic Interface.
- [6] C. Heegard, S. B. Wicker, *Turbo Coding*, Kluwer Academic Publishers, 1999.
- [7] C. Berrou, A. Glavieux and P. Thitimajshima, Near Shannon Limit Error-correcting Coding and Decoding: Turbo Codes, *IEEE International Conference on Communications*, vol. 2, p.1064-1070, Maio 1993.
- [8] S. Le Goff, A. Glavieux and C. Berrou, Turbo Codes and High Spectral Efficiency Modulation, *IEEE Transactions on Communications*, vol. 2, p. 645 – 649, Maio 1994.
- [9] P. Robertson and T. Wörz, A Novel Bandwidth Efficient Coding Scheme Employing Turbo Codes, *IEEE Transactions on Communications*, vol. 2, p. 962-967, Jun. 1996.
- [10] R. W. Chang, Synthesis of Band Limited Orthogonal Signals for Multichannel Data Transmission, *Bell Syst. Tech. J.*, vol. 45, p. 1775-1796, Dez. 1996.
- [11] Salzberg, B. R., Performance of an efficient parallel data transmission system, *IEEE Transactions Communications*, vol. 15, p. 805-813, Dez. 1967.
- [12] S. B. Weinstein, e P. M. Ebert, Data Transmission by Frequency Division Multiplexing Using the Discrete Fourier Transform, *IEEE Transactions on Communications*, vol. 19, p. 628-634, Out. 1971.
- [13] W. Y. Zou e Y. Wu, COFDM: an overview, *IEEE Transactions on Broadcasting*, vol.41, p. 1-8, Mar. 1995.

- [14] R. R. Mosier e R. G. Clabaugh, Kineplex, a Bandwidth Efficient Binary Transmission System, *AIEE Transactions*, vol. 76, p. 723-728, Jan. 1958.
- [15] G. C. Porter, Error Distribution and Diversity Performance of a Frequency Differential PSK HF modem", *IEEE Transactions on Communications*, vol. 16, p. 567-575, Ago.1968.
- [16] M. S. Zimmerman e A. L. Kirsh, "The AN/GSC-10 (KATHRYN) variable rate data modem for HF radio", *IEEE Transactions on Communications*, vol. 15, p.197-205, Abr. 1967.
- [17] W. E. Keasler e D. L. Bitzer, High Speed modem suitable for operating with a switched network, *U.S. Patent N. 4.206.320*, Jun. 1980.
- [18] P. S. Chow, J. C. Tu, e J. M. Cioffi, Performance Evaluation of a Multichannel Transceiver System for ADSL and VHSL services, *IEEE J. Selected Area*, vol. 9, p. 909-919, Ago.1991.
- [19] P. S. Chow, J. C. Tu e J. M. Cioffi, A Discrete Multitone Transceiver System for HDSL Applications, *IEEE J. Selected Area in Communications*, vol. 9, p. 909-919, Ago.1991.
- [20] W. Godoy Jr., Esquemas de Modulação Codificada com Códigos de Bloco BCM, Editora do CEFET PR, 1991.
- [21] R.E Blahut, *Theory and Practice of Error Control Codes*, Addison Wesley Pub. Co., Canadá, 1984.
- [22] S. Lin e D. J. Costello Jr., *Error Control Coding: Fundamentals and Applications*, Prentice Hall, 1983.
- [23] J. G. Proakis, *Digital Communication*, McGraw Hill, 1983.
- [24] H. Taub, e D. L. Schilling, *Principles of Communication Systems*, McGraw Hill Kogakusha, Ltda 1971.
- [25] G. C. Clark e J. B. Cain, *Error Correction Coding for Digital Communications*, Plenum Press, Nova Iorque, 1981.
- [26]C. E. Shannon, A mathematical Theory of Communication, *Bell System Technical Journal*, Out. 1948.
- [27]J. L. Massey, Shift-Register Synthesis and BCH Decoding, *IEEE Transactions on Information Theory*, IT-15, p. 122-127, Jan. 1979.

- [28] E.R Berlekamp. The Technology of Error Correcting Codes, *Proceedings of the IEEE*, Vol.68, p.564-593, Maio 1980.
- [29] L. Zhang and A. Yongacoglu, Bandwidth Efficient RS Coding in Asymmetric Digital Subscriber Lines, *International Conference on Communication Technology Proceedings*, 2000.
- [30] Q. Wang and L. Y. Onetera, Coded QAM Using a Binary Convolutional Code, *IEEE Transactions on Communications*, 1995.
- [31] J. P. Lauer and J. M. Cioffi, Turbo Coding for Discrete Multitone Transmission Systems, *IEEE Transactions on Communications*, 1998.
- [32] L. Zhang and A. Yongacoglu, Turbo Coding in ADSL DMT Systems, *IEEE Transactions on Communications*, 2001.
- [33] G. Marubayashi and S. Tachikawa, Spread Spectrum Transmission on Residential Power Line, *IEEE 4th International Symposium on Spread Spectrum Techniques and Applications Proceedings*, 1996.
- [34] E. Biglieri, Coding and Modulation for a Horrible Channel, *IEEE Communication Magazine*, 2003.
- [35] Turbo encoder/decoder–Matlab, Yufei Wu,
<http://www.ee.vt.edu/~yufei/turbo.html>, acessado em set/2002
- [36] Página do fabricante DS2, <http://www.ds2.es>, acessado em set/2002
- [37] M. F. Miranda, E. P. Ribeiro e W. Godoy Jr., Simulations of Error Correction Codes for Data Communication over Power Lines, Aprovado para apresentação no *lasted International Conference on Applied Simulation and Modelling, Grécia 2004*.

ANEXO I

Sistema com código convolucional no canal AWGN

```
clear all;
N=256; %Número de subcanais do Sistema OFDM
R=1/2; %Código Convolucional com taxa 1/2
trellis=poly2trellis(7,[133 171]);
tblen=34;
k1=log2(trellis.numInputSymbols); %Número de entradas do codificador
erro=input('Entre com o número de erros para a iteração (erro=500):');
snr_dB=input('Entre com a relação sinal-ruído - SNR máxima:');
M=16; %Modulação 16-QAM
K=log2(M);
snr=0; %Inicializa SNR

%Inicia loop para obtenção da BER
for i=1:snr_dB
    sr=10^(snr/10); %Transforma para SNR linear
    mat_snr(i)=snr;
    pacotes=0;
    nbiterro=0;
    snr=snr+1;

    while nbiterro<=erro
        pacotes=pacotes+1;
        bit_channel(1:N-1)=K; %Alocação de dados em cada subcanal
        Ltotal=sum(bit_channel);
        u1=round(rand(1,floor(Ltotal/4))); %Fonte binária
        u=convenc(u1,trellis); %Codificador convolucional
        u_aux=zeros(1,Ltotal);
```

```

u_aux((2*ceil(Ltotal/4))+1:Ltotal)=u;
%Distribuição de dados em cada subcanal
a=1;b=0;% Inicializa variáveis de auxílio
for j=1:N-1
    data_channel=[];
    m=bit_channel(j);
    b=b+m;
    val=u_aux(a:b);
    data_channel=[data_channel val];
    a=a+m;
data{j}=data_channel; % Data contém todos os valores de data_channel
end
clear j p;
t=qamgray(N,K,bit_channel,data); %Mapeador Gray
t(1:(N/2)-1)=zeros(1,(N/2)-1); %Transmissão a partir de 10kHz
t_ifft=ifft([1 t 1 fliplr(conj(t))],2*N); %IFFT do símbolo complexo mapeado
%Canal AWGN
ruído=randn(1,length(t_ifft));
[b,a]=butter(5,6000/20000,'high'); %PA em 6kHz na entrada do receptor
t_ifftf=filter(b,a,t_ifft); %Filtra sinais
ruídof=filter(b,a,ruído);
Psf=abs(sum(t_ifftf.^2)/length(t_ifftf));%Potência do sinal filtrado
Prf=abs(sum(ruídof.^2)/length(ruídof));%Potência do ruído filtrado
sigma=sqrt((Psf/Prf)/sr);
rxf=t_ifftf+sigma*ruídof;
%Receptor
rx_fft=fft(rxf); %FFT do sinal recebido
rx_fft2=rx_fft(2:N); %Remove partes conjugadas (espelhos) da FFT
ab=metric(N,K,Ltotal,rx_fft2); %Obtém métricas para Viterbi
ab_q=quantiz(ab);ab_q=ab_q'; %Quantização

```

```

ab_q2=ab_q((2*ceil(Ltotal/4))+1:Ltotal);
ur1=vitdec(ab_q2,trellis,tblen,'cont','soft',3); %Decodificador convolucional
ur2=ur1(tblen*k1+1:end); % Arruma vetores para contagem de erro
u2=u1(1:end-tblen*k1);
nbiterro=nbiterro+biterr(ur2,u2); %Contador de erro
end
mat_BER(i)=nbiterro/(length(u2)*pacotes);%Calcula quantidade de erro
end
% Plota gráfico BER x SNR
semilogy(mat_snr,mat_BER); axis([0 12 10^-6 10^0]);
xlabel('SNR');ylabel('log(BER)');grid on;legend('Convol.');
```

ANEXO II
Sistema com código RS no canal AWGN

```
clear all;
N=256; %Número de subcanais do Sistema OFDM
k1=216; %Código RS (255,216)
n1=255;
erro=input('Entre com o número de erros para a iteração (erro=500):');
snr_dB=input('Entre com a relação sinal-ruído-SNR máxima:');
M=16; %Modulação 16-QAM
K=log2(M);
snr=0; %Inicializa SNR

%Inicia loop para obtenção da BER
for i=1:snr_dB
    sr=10^(snr/10); %Transforma para SNR linear
    mat_snr(i)=snr;
    pacotes=0;
    nbiterro=0;
    snr=snr+1;

    while nbiterro<=erro
        pacotes=pacotes+1;
        bit_channel(1:N-1)=K; %Alocação de dados em cada subcanal
        Ltotal=sum(bit_channel);
        u1=round(rand(1,(Ltotal/2)-8*(n1-k1))); %Fonte binária
        u=encode(u1,n1,k1,'rs');u=u'; %Codificador RS
        u_aux=zeros(1,Ltotal);
        u_aux(Ltotal/2+1:Ltotal)=u(1:Ltotal/2);
        %Distribuição de dados em cada subcanal
```

```

a=1;b=0;% Inicializa variáveis de auxílio
for j=1:N-1
    data_channel=[];
    m=bit_channel(j);
    b=b+m;
    val=u_aux(a:b);
    data_channel=[data_channel val];
    a=a+m;
data{j}=data_channel; % Data contém todos os valores de data_channel
end
clear j p;
t=modqam(N,bit_channel,data); %Modulador Natural QAM
t(1:(N/2)-1)=zeros(1,(N/2)-1);
t_ifft=ifft([1 t 1 fliplr(conj(t))],2*N); %IFFT do símbolo complexo modulado
%Canal AWGN
ruído=randn(1,length(t_ifft));
[b,a]=butter(5,6000/20000,'high'); %PA em 6kHz na entrada do receptor
t_ifftf=filter(b,a,t_ifft); %Filtra sinais
ruídof=filter(b,a,ruído);
Psf=abs(sum(t_ifftf.^2)/length(t_ifftf));%Potência do sinal filtrado
Prf=abs(sum(ruídof.^2)/length(ruídof));%Potência do ruído filtrado
sigma=sqrt((Psf/Prf)/sr);
rxf=t_ifftf+sigma*ruídof;
%Receptor
rx_fft=fft(rxf); %FFT do sinal recebido
rx_fft2=rx_fft(2:N);%Remove partes conjugadas (espelhos) da FFT
r=demodqam(N,bit_channel,rx_fft2); %Demodulador Natural QAM
r1=[];
for j=1:N-1
    RX=r(j);

```

```

    RX=deal(RX{:});
    r1=[r1 RX];
end
clear j;
ur1=floor((r1+1)/2);
ur11=ur1((Ltotal/2+1):Ltotal);
ur=decode(ur11,n1,k1,'rs');ur=ur'; %Decodificador RS
ur2=ur(1:(Ltotal/2)-8*(n1-k1));
nbiterro=nbiterro+biterr(ur2,u1); %Contador de erro
end
    mat_BER(i)=nbiterro/(length(u1)*pacotes);
end
% Plota gráfico BER x SNR
semilogy(mat_snr,mat_BER); axis([0 12 10^-6 10^0]);
xlabel('SNR');ylabel('log(BER)');grid on;legend('RS');

```

ANEXO III

Sistema com código turbo no canal com ruído medido da rede elétrica

```
clear all;
N=256; %Número de subcanais do Sistema OFDM
R=1/2; %Código turbo
dec=0;%dec=0 para MAP
niter=3;%Número de iterações para decodificador turbo
g=[1 1 1 1 1; 1 0 0 0 1]; %Matriz geradora dos códigos convolucionais RSC
[n,K1]=size(g);
m=K1-1; %Memória dos codificadores RSC
erro=input('Entre com o numero de erros para a iteração (erro=500):');
snr_dB=input('Entre com a relação sinal-ruído - SNR máxima:');
M=16; %Modulação 16-QAM
K=log2(M);
load sinal_rede;%Sinal medido da rede elétrica
index=floor(length(sinal_rede)/(2*N));
index2=0;
ruído_rede=sinal_rede(1:index*2*N);
snr=0; %Inicializa SNR

%Inicia loop para obtenção da BER
for i=1:snr_dB
    sr=10^(snr/10); %Transforma para SNR linear
    mat_snr(i)=snr;
    pacotes=0;
    nbiterro=0;
    snr=snr+1;

    while nbiterro<=erro
```

```

pacotes=pacotes+1;
bit_channel(1:N-1)=K; %Alocação de dados em cada subcanal
Ltotal=sum(bit_channel);
u1=round(rand(1,(floor(Ltotal/4)-m))); %Fonte binária
[u,alpha]=turbo_encoder(u1,g); %Codificador turbo binário
[temp1, beta]=sort(rand(1,length(u))); %Entrada para entrelaçador
for j=1:(length(u))
    ui(1,j)=u(1,beta(j));
end
u_aux=zeros(1,Ltotal);
u_aux(2*ceil(Ltotal/4)+1:Ltotal)=ui;
% Distribuição de dados em cada subcanal
a=1;b=0;% Inicializa variáveis de auxílio
for j=1:N-1
    data_channel=[];
    m1=bit_channel(j);
    b=b+m1;
    val=u_aux(a:b);
    data_channel=[data_channel val];
    a=a+m1;
data{j}=data_channel;% Data contém todos os valores de data_channel
end
clear j p;
t=qamgray(N,K,bit_channel,data); %Mapeador Gray
t(1:(N/2)-1)=zeros(1,(N/2)-1);
t_ifft=ifft([1 t 1 fliplr(conj(t))],2*N);%IFFT do símbolo complexo mapeado
%Canal com ruído medido da rede elétrica
ruído=ruído_rede((1+index2):(2*N+index2));ruído=ruído';
if index2<(index*2*N-(2*N)) % Calcula contador que define ruído
    index2=index2+(2*N);

```

```

else
    index2=0;
end
[b,a]=butter(5,6000/20000,'high'); %Filtro PA em 6kHz
t_ifftf=filtfilt(b,a,t_ifft); %Filtra sinais
ruidof=filtfilt(b,a,ruido);
Psf=abs(sum(t_ifftf.^2)/length(t_ifftf));%Potência do sinal filtrado
Prf=abs(sum(ruidof.^2)/length(ruidof));%Potência do ruído filtrado
sigma=sqrt((Psf/Prf)/sr);
rxf=t_ifftf+sigma*ruidof;
%Receptor
rx_fft=fft(rxf); %FFT do sinal recebido
rx_fft2=rx_fft(2:N);%Remove partes conjugadas (espelhos) da FFT
ab=metric(N,K,Ltotal,rx_fft2); %Obtém métricas (LLR) para decodificador
ab_aux=ab(2*ceil(Ltotal/4)+1:Ltotal);
for j=1:(length(u))
    ab2(1,beta(j))=ab_aux(j); %Entrada para De-entrelaçador
end
ur1=turbo_decoder(ab2,g,alpha,dec,niter); %Decodificador turbo
nbiterro=nbiterro+biterr(ur1,u1); %Contador de erro
end
mat_BER(i)=nbiterro/(length(u1)*pacotes);
end
% Plota gráfico BER x SNR
semilogy(mat_snr,mat_BER); axis([0 12 10^-6 10^0]);
xlabel('SNR');ylabel('log(TEB)');grid on;legend('Turbo');

```