

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
ESCOLA DE ENGENHARIA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

**Jonatan Dellagostin**

## **PROJETO DE DIPLOMAÇÃO**

**Algoritmos para estimação do canal de rádio móvel em sistemas  
de comunicação digital baseados em OFDM (Orthogonal  
Frequency-Division Multiplexing)**

Porto Alegre

2009

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
ESCOLA DE ENGENHARIA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

**Algoritmos para estimação do canal de rádio móvel em sistemas  
de comunicação digital baseados em OFDM (Orthogonal  
Frequency-Division Multiplexing)**

Projeto de Diplomação apresentado ao  
Departamento de Engenharia Elétrica da Universidade  
Federal do Rio Grande do Sul, como parte dos  
requisitos para Graduação em Engenharia Elétrica.

ORIENTADOR: Prof. Dr. Roger Pierre Fabris Hoefel

Porto Alegre

2009

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
ESCOLA DE ENGENHARIA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

Jonatan Dellagostin

**Algoritmos para estimação do canal de rádio móvel em sistemas  
de comunicação digital baseados em OFDM (Orthogonal  
Frequency-Division Multiplexing)**

Este projeto foi julgado adequado para fazer jus aos créditos da Disciplina de “Projeto de Diplomação”, do Departamento de Engenharia Elétrica e aprovado em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: \_\_\_\_\_

Prof. Dr. Roger Pierre Fabris Hoefel

Doutor pela UNICAMP, CAMPINAS, BRASIL.

Banca Examinadora:

Prof. Dr. Roger Pierre Fabris Hoefel, UFRGS

Doutor pela UNICAMP, CAMPINAS, BRASIL.

Eng. Julio Klemm, UFRGS

Engenheiro Eletricista pela UFRGS, PORTO ALEGRE, BRASIL.

Prof. Dr. Alvaro Augusto Almeida de Salles, UFRGS

Doutor pela UNIVERSITY OF LONDON – LONDRES, INGLATERRA.

Porto Alegre, dezembro de 2009.

## **DEDICATÓRIA**

Dedico este trabalho à minha mãe, cujo apoio e amor incondicionais foram imprescindíveis, especialmente no período de graduação, que culminou com este trabalho de conclusão.

## **AGRADECIMENTOS**

A todos aqueles que pagam seus impostos e viabilizam o ensino de qualidade e gratuito que é proporcionado a milhares de estudantes em todo o nosso país.

Aos professores que possuem comprometimento sério e dedicação à pesquisa científica, especialmente ao professor Roger Pierre Fabris Hoefel, cuja orientação, incentivo e conhecimentos repassados, tanto durante o período em que fui bolsista de iniciação científica quanto durante o desenvolvimento deste trabalho de conclusão, foram inestimáveis e essenciais para a consecução dos objetivos propostos para este projeto.

Aos amigos e familiares que ajudaram, incentivaram e apoiaram nos muitos momentos difíceis desde o começo do curso de Engenharia Elétrica até a finalização deste trabalho de conclusão.

## RESUMO

Pode-se dividir este projeto em quatro partes principais. Em primeiro lugar, é feito um estudo descritivo da norma 802.11a do IEEE (Institute of Electrical and Electronics Engineers), que se refere à transmissão e recepção de pacotes em um sistema de comunicação digital sem fio. Em segundo lugar, é analisado o funcionamento de um simulador 802.11a implementado no software Matlab e se documentarão todas as funções que o compõem. Em terceiro lugar, são realizadas modificações no código do simulador, de maneira que este possa extrair e transmitir o conteúdo binário de uma imagem, bem como possa gerar gráficos que mostrem as probabilidades de erro de bit e de pacote em função da razão sinal/ruído. Por final, são analisadas na literatura alternativas de algoritmos de estimação de canal, de maneira que possam ser implementados no simulador e comparados entre si, relativamente à performance de erros de bit e de pacote.

**Palavras-chaves: Engenharia Elétrica. Telecomunicações. Comunicação Digital. Sistemas sem Fio, Redes Locais sem Fio, 802.11, 802.11a.**

## **ABSTRACT**

This project can be divided in four main parts. In first place, it is carried out a descriptive study of IEEE (Institute of Electrical and Electronics Engineers) 802.11a standard, which refers to the transmission and reception of packets in a wireless digital communication system. In second place, it is analyzed the functioning of an IEEE 802.11a simulator implemented in the software Matlab and its component functions will be documented. In third place, it is done modifications in the simulator's code, in a way that it could generate graphics showing the bit and packet error probabilities as a function of signal to noise ratio. Finally, it is analyzed in the literature alternatives of channel estimation algorithms, such as they can be implemented in the simulator and compared among them, relatively to the bit and packet error performance.

**Keywords: Electrical Engineering. Telecommunication. Digital communication.  
Wireless systems, Wireless Local Area Networks, 802.11, 802.11a, Channel  
Estimation.**

## SUMÁRIO

1 INTRODUÇÃO.....	13
1.1 IEEE 802.11.....	13
1.2 OFDM.....	13
1.3 Simulador 802.11a.....	14
2 TRANSMISSOR.....	16
2.1 Formato do quadro da camada física segundo a norma 802.11a (IEEE).....	16
2.2 Processo de encapsulamento do quadro 802.11a no bloco do Transmissor do simulador.....	20
3 CANAL.....	39
3.1 Descrição do canal de rádio-móvel.....	39
3.2 Descrição da implementação do canal de rádio móvel no simulador 802.11a.....	41
4 RECEPTOR.....	47
4.1 Descrição da implementação do receptor no simulador 802.11a.....	47
5 SIMULAÇÕES: RESULTADOS E ANÁLISES.....	73
6 CONCLUSÃO.....	87
7 REFERÊNCIAS BIBLIOGRÁFICAS.....	89
APÊNDICE.....	91
Transmissor.....	91
Canal.....	100
Receptor.....	102



# LISTA DE ILUSTRAÇÕES

## LISTA DE ILUSTRAÇÕES

Figura 1.1. Diagrama de blocos do simulador 802.11a.....	15
Figura 2.1. Formato do quadro PLCP segundo a norma 802.11a.....	16
Figura 2.2. Diagrama de blocos do transmissor do simulador 802.11a.....	20
Figura 2.3. Codificador convolucional utilizado no IEEE 802.11a. ....	23
Figura 2.4. Ganho de código na BER para um codificador convolucional de 64 estados, com taxa de código 1/2, em um canal AWGN. ....	24
Figura 2.5. Dois padrões diferentes de apagamento utilizados pela IEEE 802.11a.....	26
Figura 2.6. Mapeamento de constelações do código de Gray para a modulação BPSK.....	29
Figura 2.7. Mapeamento de constelações do código de Gray para a modulação QPSK.....	29
Figura 2.8. Mapeamento de constelações do código de Gray para a modulação 16-QAM.....	30
Figura 2.9. Mapeamento de constelações do código de Gray para a modulação 64-QAM.....	30
Figura 2.10. Embaralhador de dados.....	35
Figura 2.11. Alocação de frequência para as subportadoras.....	36
Figura 2.12. Estrutura de treinamento OFDM.....	37
Figura 3.1. Diagrama de blocos do canal de rádio-freqüência.....	42
Figura 4.1. Diagrama de blocos do receptor do simulador.....	47
Figura 4.2. Estrutura de fluxo do sinal e do algoritmo de atraso e correlação.....	49
Figura 4.3. Localização das subportadoras piloto no tempo e na freqüência.....	55
Figura 4.4. Rotação de frequência para um offset de 3kHz durante 10 símbolos.....	56
Figura 4.5. Performance de BER para seleção de diversidade em um canal com desvanecimento Rayleigh.....	60
Figura 4.6. Performance de BER para MRC em um canal com desvanecimento Rayleigh.....	62
Figura 4.7. Limites de decisão Hard para a constelação QPSK.....	64
Figura 4.8. Performance de BER no IEEE 802.11a no modo 12 Mbits/s em um canal AWGN utilizando-se decisões de decodificação soft e hard.....	67
Figura 4.9. Performance de PER no IEEE 802.11a no modo 12 Mbits/s em um canal AWGN utilizando-se decisões de decodificação soft e hard.....	68
Figura 4.10. Efeito sobre a PER com ponderamento de métricas na decodificação de Viterbi, para a IEEE 802.11a no modo 12 Mbits/s em um canal de desvanecimento Rayleigh com atraso de espalhamento de 75ns rms.....	69
Figura 5.1. Interface gráfica do simulador 802.11a. ....	71
Figura 5.2. Performances de BER teórica e simulada para fins de validação do simulador.....	73
Figura 5.3. Performance de BER e PER, utilizando algoritmo de estimação de canal baseado nas subportadoras piloto. ....	74
Figura 5.4. Performance de BER e PER, utilizando algoritmo de estimação de canal baseado nos símbolos de treinamento. ....	74
Figura 5.5. Performance de BER e PER sem algoritmos de sincronização.....	75
Figura 5.6. Performance de BER e PER, com algoritmo de estimação de canal.....	76
Figura 5.7 Performance de BER e PER, utilizando 5 algoritmos de melhoria de performance.....	77
Figura 5.7. Performance de BER e PER, utilizando todos os algoritmos citados na figura 5.6, com adição do interleaver.....	77
Figura 5.8. Performance de BER e PER, utilizando 6 algoritmos de melhoria de performance.....	78

Figura 5.9. Performance de BER e PER, utilizando 7 algoritmos de melhoria de performance..... 79

## LISTA DE TABELAS

Tabela 2.1. Parâmetros de modulação dependentes da taxa de transmissão de dados.....	19
Tabela 2.2. Parâmetros relacionados ao domínio do tempo. ....	20
Tabela 2.3. Distâncias livres dos códigos convolucionais de 64 estados utilizados no IEEE 802.11a.....	26
Tabela 2.4. Fator de normalização $K_{MOD}$ para as modulações BPSK, QPSK, 16-QAM e 64-QAM. ....	29
Tabela 2.5. Tabela de codificação de Gray para BPSK.....	31
Tabela 2.6. Tabela de codificação de Gray para QPSK. ....	31
Tabela 2.7. Tabela de codificação de Gray para 16-QAM.....	31
Tabela 2.8. Tabela de codificação de Gray para 64-QAM.....	32

## **LISTA DE ABREVIATURAS**

A/D: Analog to Digital

AGC: Automatic Gain Control

AWGN: Additive White Gaussian Noise

BER: Bit Error Rate

BPSK: Binary Phase Shift Keying

C-MPDU: Coded MPDU

D/A: Digital to Analog

DC: Direct Current

DELET: Departamento de Engenharia Elétrica

DFT: Discrete Fourier Transform

FEC: Forward Error Correction

FFT: Fast Fourier Transform

FIR: Finite Impulse Response

GI: Guard Interval

ICI: Inter-Carrier Interference

IEEE: Institute of Electrical and Electronics Engineers

IFFT: Inverse Fast Fourier Transform

ISI: Inter-Symbol Interference

LOS: Line Of Sight

LSB: Least Significant Bits

MBPS: Mega Bits por Segundo

MLME: MAC Sublayer Management Entity

OFDM: Orthogonal Frequency Division Multiplexing

OSI: Open Systems Interconnection

PA: Ponto de Acesso

PDU: Protocol Data Unit

PER: Packet Error Rate

PHY: Physical Layer

PLCP: Physical Layer Convergence Procedure

PLME: PHY Management Entity

PMD: Physical Medium Dependent

PPDU: PLCP Protocol Data Unit

PSDU: PHY Sublayer service Data Units

QAM: Quadrature Amplitude Modulation

QPSK: Quadrature Phase Shift Keying

RF: Radio Frequency

RSR: Razão Sinal para Ruído

SNR: Signal to Noise Ratio

UFRGS: Universidade Federal do Rio Grande do Sul

WI-FI: Wireless Fidelity

WLAN: Wireless Local Area Network

## **1. INTRODUÇÃO**

### **1.1 IEEE 802.11**

A norma 802.11 do IEEE (Institute of Electrical and Electronics Engineers), define o padrão para redes sem fio, que também podem ser denominadas Wi-Fi. Este padrão é largamente utilizado no mundo inteiro nas aplicações de telecomunicações que envolvam conectividade sem fio, fato comprovado pela grande quantidade de PAs (pontos de acesso), localizados em inúmeros locais nos centros urbanos (parques, restaurantes, escolas, aeroportos, universidades, hotéis, residências, etc.), e de dispositivos de rede fabricados que são compatíveis com a norma. Também é notável que a maioria dos computadores portáteis são fabricados com interfaces de rede já adaptadas para o padrão 802.11.

O padrão divide-se em várias partes, das quais a de maior interesse neste trabalho é a norma IEEE 802.11a, sobre a qual versará o presente trabalho.

#### **1.1.1 IEEE 802.11a**

A norma determina oito possíveis velocidades de transmissão em Mbps (6, 9, 12, 18, 24, 36, 48 e 54), embora alguns fabricantes também implementem velocidades não padronizados de 72 a 108 Mbps. A frequência de operação da portadora em RF (rádio-freqüência) está na faixa de 5 GHz. O grande trunfo do padrão 802.11a, que o torna especialmente eficaz e confiável frente adversidades severas no canal de transmissão (desvanecimento) é a utilização de uma técnica sofisticada de modulação, chamada de OFDM (Orthogonal Frequency-Division Multiplexing), a qual será resumidamente explicada no item subsequente.

### **1.2 OFDM**

O princípio básico do OFDM é a divisão de um fluxo inicial de dados, em alta taxa de transmissão, em vários fluxos, onde cada um deles possui uma taxa de transmissão mais baixa que a

do fluxo inicial. Estes fluxos são transmitidos paralelamente e simultaneamente sobre certo número de subportadoras. Em virtude do aumento no período de símbolo, devido à taxa de transmissão mais baixa para as subportadoras em paralelo, a dispersão total no tempo causada pelo atraso de múltiplos percursos é diminuída. A interferência intersimbólica é quase completamente eliminada pela introdução de um intervalo de guarda (GI, do inglês Guard Interval) em cada símbolo OFDM. Para tanto, estende-se o sinal, repetindo-se parte do símbolo OFDM, de forma a torná-lo cíclico, com o objetivo de evitar a interferência entre as portadoras.

No projeto de um sistema OFDM, há um certo número de parâmetros a se considerar, tais quais o número de subportadoras, GI, período de símbolo, espaçamento entre as subportadoras, e o tipo de codificação para correção de erros (FEC, do inglês Forward Error Correction). A escolha destes parâmetros é influenciada pelos requerimentos do sistema, como banda disponível, taxa de transmissão de bits, atraso tolerável, e valores relativos ao efeito Doppler. Esta escolha não é simples, visto que alguns requerimentos são conflitantes entre si. Como exemplo: para conseguir uma boa tolerância ao atraso de propagação, um grande número de subportadoras com pequeno espaçamento entre as subportadoras adjacentes é desejável, mas a situação contrária também é favorável para alcançar-se boa tolerância contra efeito Doppler e ruído de fase.

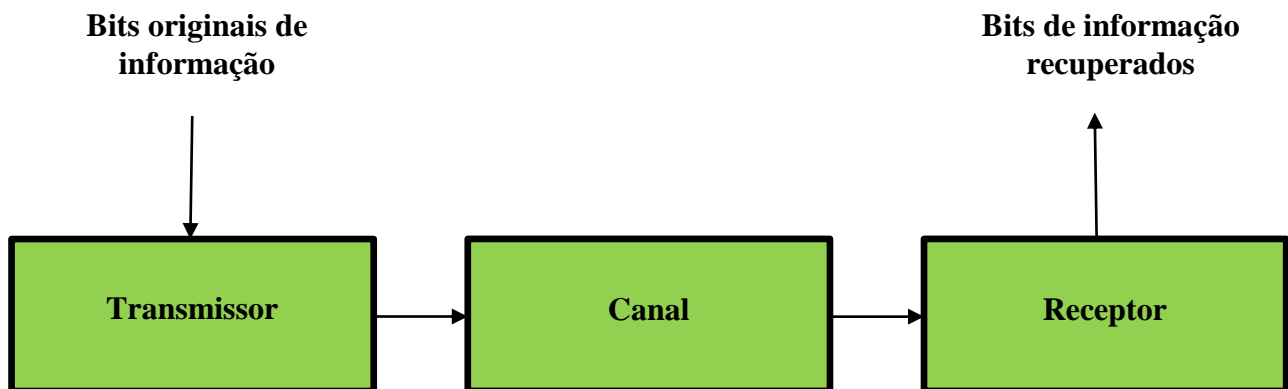
### **1.3 Simulador IEEE 802.11a**

Para os objetivos que serão explicitados a seguir, utilizar-se-á um simulador, construído por meio do software Matlab, que simula a transmissão de pacotes 802.11a. Este simulador pode ser resumidamente descrito por meio do diagrama de blocos da figura 1.1, que demonstra o fluxo dos quadros desde sua transmissão até a recepção. Os blocos serão detalhados nos capítulos subsequentes, incluindo a referência às funções correspondentes em Matlab que os compõem.

Os objetivos do presente trabalho são:

- Estudo e análise da literatura, no que se refere à norma 802.11a e sistemas OFDM;

- Modificações no código do simulador 802.11a, de modo que este possa extrair e transmitir o conteúdo binário de uma imagem, bem como possa gerar gráficos mostrando as probabilidades de erro de bit e de pacote em função da razão sinal para ruído;
- Implementação de algoritmos que realizem a estimação do canal de rádio móvel em sistemas de comunicação digital baseados em OFDM. A estimação adequada da resposta impulsiva do canal é essencial para a correta recuperação do sinal em sistemas que utilizam detecção coerente, devido à degradação sofrida em virtude da presença de ruído, de múltiplos percursos e de desvanecimento da potência do sinal, ao longo do percurso entre a antena transmissora e a antena receptora. De posse da resposta impulsiva do canal, podemos recuperar o sinal transmitido, a partir do sinal recebido, tornando a demodulação e a decodificação mais confiáveis.



**Figura 1.1. Diagrama de blocos do simulador 802.11a.**

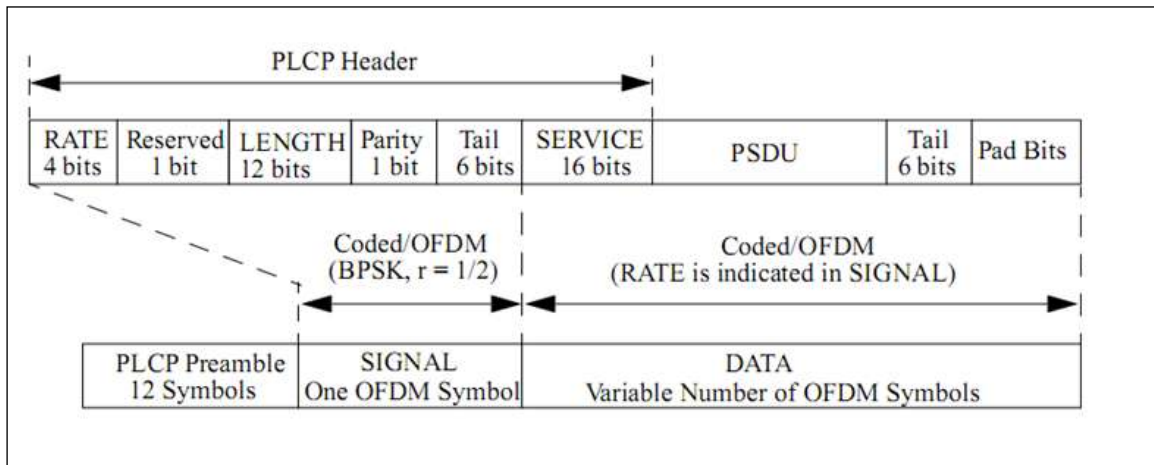
Ao final da simulação, portanto, têm-se resultados relativos à probabilidade de erro de bit e de pacote em função da razão sinal para ruído aplicada, os quais podem ser comparados com os valores obtidos teoricamente por meio das expressões analíticas correspondentes. Esta comparação permite verificar a consistência dos resultados do simulador.



## 2. TRANSMISSOR

### 2.1 Formato do quadro da camada física segundo a norma 802.11a (IEEE)

O formato do quadro PLCP (Physical Layer Convergence Procedure), ou PPDU (PLCP Protocol Data Unit), a ser transmitido pela camada física, é mostrado na figura 2.1.



**Figura 2.1. Formato do quadro PLCP.**

Em termos de modulação, os campos LENGTH (número de bytes do campo DATA), RATE (taxa de transmissão do campo DATA), reserved bit (bit reservado), e parity bit (bit de paridade), com 6 bits “zero” inseridos no final, constituem um único símbolo OFDM, chamado de SIGNAL, o qual é transmitido com uma combinação robusta de modulação BPSK (Binary Phase-Shift Keying) e taxa de codificação  $R=1/2$ .

O campo SERVICE (serviço) do cabeçalho PLCP e o PSDU (PHY sublayer service data unit), com 6 bits “zero” e bits de preenchimento inseridos no final, são chamados de DATA (dados) e transmitidos com a taxa de dados descrita no campo RATE, e podem constituir múltiplos símbolos OFDM. Os bits de preenchimento no símbolo SIGNAL permitem decodificação dos campos RATE e LENGTH imediatamente após a recepção daqueles. Estes dois campos são necessários para decodificar o campo DATA do pacote.

#### 2.1.1 O processo de encapsulamento do PPDU

O processo de encapsulamento do PPDU consiste resumidamente nos seguintes itens:

a) Produção do campo do preâmbulo PLCP, composto por 10 repetições de uma seqüência de treinamento curto (do inglês Short Training Sequence). Esta seqüência é utilizada para controle automático de ganho (AGC, do inglês Automatic Gain Control), diversidade de antenas, sincronização, e sintonia grossa de freqüência no receptor. Em seguida, há duas repetições de uma seqüência de treinamento longo (do inglês Long Training Sequence). Esta seqüência é utilizada para estimação do canal e sintonia fina de freqüência no receptor. O preâmbulo é precedido por um intervalo de guarda GI.

b) Produção do campo header (cabeçalho), composto pelos campos RATE, LENGTH, e SERVICE, cujos bits são preenchidos de acordo com a figura 2.1. Os campos RATE e LENGTH são codificados por um código convolucional a uma taxa de  $R = \frac{1}{2}$ , e são subsequentemente mapeados para um único símbolo OFDM, modulado em BPSK e a uma taxa de transmissão de dados a 6 Mbits/s. Este símbolo é chamado SIGNAL. De modo a facilitar uma detecção confiável e sincronizada dos campos RATE e LENGTH, 6 bits “zero” de preenchimento são inseridos no cabeçalho PLCP. A codificação do campo SIGNAL para um símbolo OFDM segue a ordem descrita abaixo:

- 1) Codificação convolucional;
- 2) Intercalação de bits (no tempo e na freqüência);
- 3) Modulação BPSK;
- 4) Inserção de subportadoras piloto;
- 5) Transformada de Fourier;
- 6) Anteposição de um GI.

c) Calcula-se o número de bits de dados por símbolo OFDM ( $N_{DBPS}$ ), a taxa de codificação ( $R$ ), o número de bits em cada subportadora OFDM ( $N_{BPS}$ ), e o número de bits codificados por símbolo OFDM ( $N_{CBPS}$ ), conforme mostra a Tabela 2.1.

- d) Concatenação do PSDU (PHY sublayer service data unit) logo após o campo SERVICE. Complementa-se com bits “zero” (em um mínimo de 6) de modo que o tamanho resultante seja um múltiplo de  $N_{DBPS}$ . A seqüência de bits resultante constitui o campo DATA do pacote.
- e) Inicialização do embaralhador (do inglês scrambler) com uma semente pseudo-randômica não-nula (do inglês pseudo-random non-zero seed), geração de uma seqüência de embaralhamento, e operação XOR com a seqüência de bits de dados resultante acima. A seqüência de inicialização é inserida no campo SERVICE.
- f) Substituição dos seis bits “zero” que seguem os bits de dados com seis bits “zero” não embaralhados.
- g) Codificação da seqüência de bits de dados embaralhada por meio de um codificador convolucional ( $R=1/2$ ).
- h) Apagamento de bits da seqüência de dados para alcançar a taxa de codificação desejada.
- i) Divisão da seqüência de bits codificados em grupos de tamanho  $N_{CBPS}$ . Em cada grupo, reordenam-se os bits (interleaving) de acordo com a regra correspondente à taxa (RATE) desejada.
- j) Mapeamento de cada grupo de  $N_{CBPS}$  bits para um número complexo de acordo com o tipo de modulação, conforme explicitado mais adiante no item 2.2.6.
- k) Divisão da seqüência de números complexos em 48 grupos. Cada um desses grupos será associado a um símbolo OFDM. Em cada grupo, os números complexos serão numerados de 0 a 47 e mapeados em subportadoras OFDM numerados de -26 a -22, -20 a -8, -6 a -1, 1 a 6, 8 a 20, e 22 a 26. As subportadoras -21, -7, 7, e 21 são evitadas, e, subseqüentemente, usadas para a inserção de subportadoras piloto. A subportadora “0”, associada à freqüência central, é omitida e preenchida com valor zero.

- l) Quatro subportadoras são inseridas como pilotos nas posições -21, -7, 7 e 21. O número total de subportadoras é de 52 (48 +4).
- m) Para cada grupo de subportadoras de -26 a 26, convertem-se as mesmas para o domínio tempo através da transformada inversa de Fourier. Adiciona-se ao início da forma de onda uma extensão circular de si mesma de modo a formar o GI, e trunca-se a forma de onda periódica resultante para o tamanho de um único símbolo OFDM, por meio da aplicação de uma janela no domínio do tempo.
- n) Concatena-se os símbolos OFDM um após o outro, começando depois do símbolo SIGNAL.
- o) Converte-se a forma de onda resultante na banda básica para uma frequência RF (Radio Frequency) de acordo com a frequência central do canal desejado e inicia-se a transmissão.

**Tabela 2.1. Parâmetros de modulação dependentes da taxa de transmissão de dados.**

Taxa de dados (Mbits/s)	Modulação	Taxa de codificação	Bits codificados por subportadora ( $N_{BPSC}$ )	Bits codificados por símbolo OFDM ( $N_{CBPS}$ )	Bits de dados por símbolo OFDM ( $N_{DBPS}$ )
6	BPSK (Binary Phase-Shift Keying)	1/2	1	48	24
9	BPSK	3/4	1	48	36
12	QPSK (Quadrature Phase Shift Keying)	1/2	2	96	48
18	QPSK	3/4	2	96	72
24	16-QAM (16-Quadrature Amplitude Modulation)	1/2	4	192	96
36	16-QAM	3/4	4	192	144
48	64-QAM (64-Quadrature Amplitude Modulation)	2/3	6	288	192
54	64-QAM	3/4	6	288	216

Tabela 2.2. Parâmetros relacionados ao domínio do tempo.

Parâmetro	Valor
$N_{SD}$ : Número de subportadoras de dados	48
$N_{SP}$ : Número de subportadoras piloto	4
$N_{ST}$ : Número total de subportadoras	52 ( $N_{SD} + N_{SP}$ )
$\Delta_F$ : Espaçamento entre as frequências das subportadoras	0.3125 MHz (=20 MHz/64)
$T_{FFT}$ : período da IFFT/FFT	3.2 $\mu$ s ( $1 / \Delta_F$ )
$T_{PREÂMBULO}$ : do preâmbulo PLCP	16 $\mu$ s ( $T_{SHORT} + T_{LONG}$ )
$T_{SINAL}$ : Duração do SINAL BPSK- símbolo OFDM	4.0 $\mu$ s ( $T_{GI} + T_{FFT}$ )
$T_{GI}$ : Duração do GI	0.8 $\mu$ s ( $T_{FFT} / 4$ )
$T_{GI2}$ : Duração do GI do símbolo de treinamento	1.6 $\mu$ s ( $T_{FFT} / 2$ )
$T_{SYM}$ : Intervalo de símbolo	4 $\mu$ s ( $T_{GI} + T_{FFT}$ )
$T_{SHORT}$ : Duração da seqüência de treinamento curto	8 $\mu$ s ( $10 \times T_{FFT} / 4$ )
$T_{LONG}$ : Duração da seqüência de treinamento longo	8 $\mu$ s ( $T_{GI2} + 2 \times T_{FFT}$ )

## 2.2 Processo de encapsulamento do quadro 802.11a no bloco do Transmissor do simulador

Nesta seção, o encapsulamento do quadro IEEE 802.11a, que foi resumidamente descrito no item 2.1.1, será detalhado de acordo com a perspectiva da implementação no Transmissor do simulador. A figura 2.2 apresenta um diagrama de blocos contendo os passos necessários para este encapsulamento.

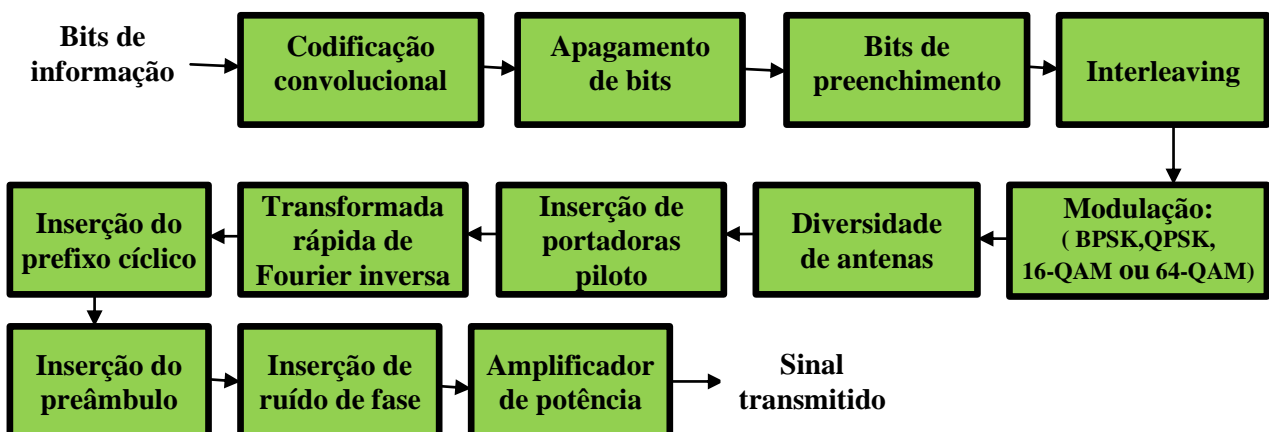


Figura 2.2. Diagrama de blocos do transmissor do simulador.

### **2.2.1 Bits de informação:**

Inicialmente, no simulador desenvolvido em [TER 2001], os bits de dados das camadas superiores do modelo OSI (Open Systems Interconnection), que constituem o campo DATA do quadro PPDU, eram gerados por meio de números aleatórios, não constituindo propriamente informação útil que possa ser decodificada.

Com as modificações realizadas, entretanto, tem-se agora a possibilidade de comparar arquivos de imagem, que contém efetivamente bits de informação. No início da simulação, qualquer imagem no formato jpeg, que esteja contida na mesma pasta do simulador, e cujo nome seja *imagein.jpg*, terá seu conteúdo binário extraído. Este conteúdo será organizado em uma seqüência de bits, que será tratada pelo simulador da mesma forma que inicialmente tratava os bits aleatórios de informação.

### **Arquivos correspondentes no Matlab:**

O código que gera bits aleatórios encontra-se no arquivo *runsim.m*, enquanto que o código que extrai o conteúdo binário de uma imagem encontra-se no arquivo *imgtobin.m*.

### **2.2.2 Codificação convolucional:**

A codificação convolucional é um tipo de codificação de canal que permite, considerando um dado valor de taxa de erro de bit requerido, reduzir a (Razão Sinal para Ruído, do inglês SNR - Signal to Noise Ratio) necessária, aumentando, portanto, a eficiência na utilização de potência pelo transmissor.

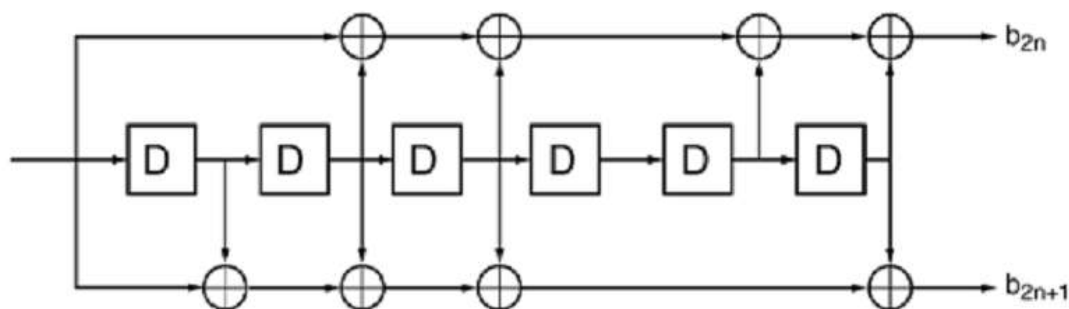
A técnica consiste, basicamente, na introdução de bits de redundância (que não carregam informação) no pacote a ser transmitido. Os bits de redundância permitem, de um modo geral, a detecção e a correção de erros nos bits recebidos. Para tanto se utiliza um polinômio gerador, que

pode ser representado por uma máquina de estados com memória, para o codificador, e, para decodificação, um decodificador de Viterbi.

Códigos convolucionais são largamente utilizados como códigos de canal nos sistemas atuais. Os padrões WLAN (Wireless Local Area Network) IEEE 802.11a e HiperLAN/2 utilizam códigos convolucionais para correção de erros, e o padrão IEEE 802.11b inclui um modo opcional que possibilita sua utilização. Os códigos convolucionais devem sua popularidade à sua boa performance e flexibilidade para trabalhar com diferentes taxas de código.

Um código convolucional pode ser definido por um conjunto de conexões entre estágios de um ou mais registradores de deslocamento (do inglês shift registers) e os bits de saída do codificador. A taxa de um código convolucional é geralmente expressa como  $k/n$ . O  $k$  é igual ao número de bits de informação e  $n$  é igual ao número de bits de saída.

A figura 2.3 mostra o codificador convolucional utilizado no IEEE 802.11a. Ele apresenta uma taxa de código de  $\frac{1}{2}$  com conexões  $(133)_8$  e  $(171)_8$ . Estas conexões são definidas como números em base octal, cujas representações em base binária são 001 011 011 e 001 111 001. A notação octal é utilizada para encurtar as expressões, quando conexões para diferentes códigos convolucionais são definidas. A partir da notação binária, a estrutura do codificador é facilmente construída. As conexões são alinhadas pelo final de cada registrador de deslocamento, e um valor 1 indica que o estágio de saída do registrador de deslocamento é conectado a um dos bits de saída do codificador por meio de uma operação XOR binária. Na figura 2.3 a conexão  $(133)_8$  define os valores dos bits com índice par  $b_{2n}$  e a conexão  $(171)_8$  define os valores dos bits com índice ímpar  $b_{2n+1}$ .



**Figura 2.3. Codificador convolucional utilizado no IEEE 802.11a.**

O número de registradores de deslocamento determina o ganho de código que pode ser alcançado pelo código convolucional. Quanto mais registradores, maior será o ganho de código; entretanto, a complexidade do decodificador de máxima verossimilhança de Viterbi cresce exponencialmente com o número de registradores. O limite de complexidade atual limita os códigos convolucionais a oito registradores de deslocamento. No caso do IEEE 802.11a, utiliza-se apenas seis, devido às altas taxas de transmissão requeridas.

A performance de um código convolucional é determinada pela distância livre mínima do código. A distância livre é definida por meio da distância de Hamming, que é igual ao número de posições nas quais duas palavras código do código convolucional diferem. A distância livre de um código convolucional é então a mínima distância de Hamming entre duas palavras código diferentes. Um ganho de código assintótico com alta RSR para um código convolucional pode ser calculado a partir da distância livre e da taxa do código, de acordo com a equação (2.1):

$$\text{ganho de código} = 10 \log_{10}(\text{taxa} \cdot \text{distância livre}). \quad (2.1)$$

O código utilizado no IEEE 802.11a possui uma distância livre de 10 e uma taxa de código de  $\frac{1}{2}$ , o que fornece um ganho de código assintótico de  $10 \log_{10}\left(\frac{1}{2} \cdot 10\right) = 7.0$  dB. Contudo, este é um resultado assintótico; a figura 2.4 mostra que para uma BER (Bit Error Rate) de  $10^{-5}$ , o ganho de código é de 5.5 dB, e diminui ainda mais para baixas RSRs de bit ( $\frac{E_b}{N_0}$ , onde  $E_b$  é a energia de bit e  $N_0$  é a energia do ruído). O resultado assintótico é obtido somente em uma região de RSRs muito altas, que são impraticáveis para sistemas reais.



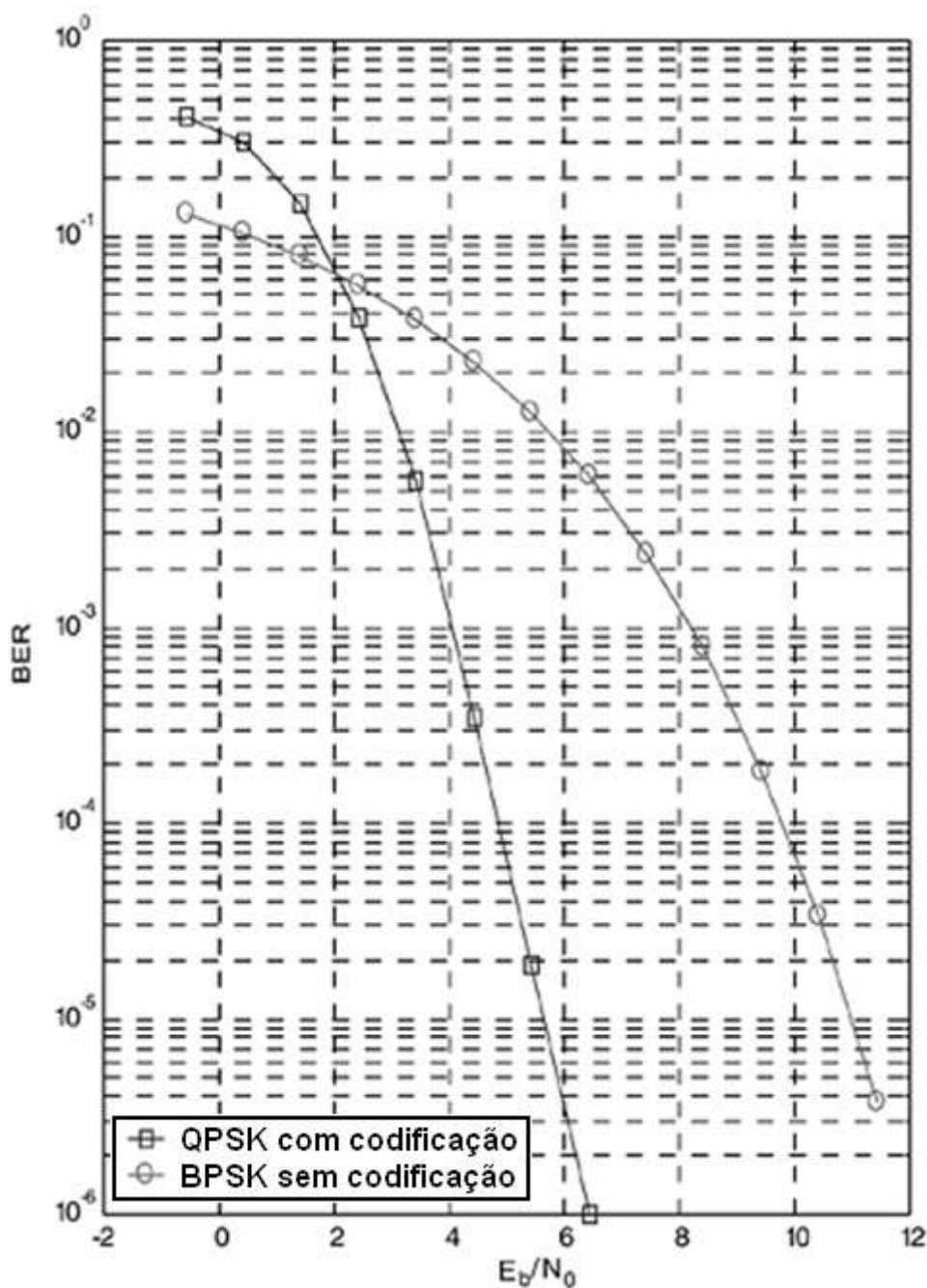


Figura 2.4. Ganho de código na BER para um codificador convolucional de 64 estados, com taxa de código  $\frac{1}{2}$ , em um canal AWGN. [TER 2001]

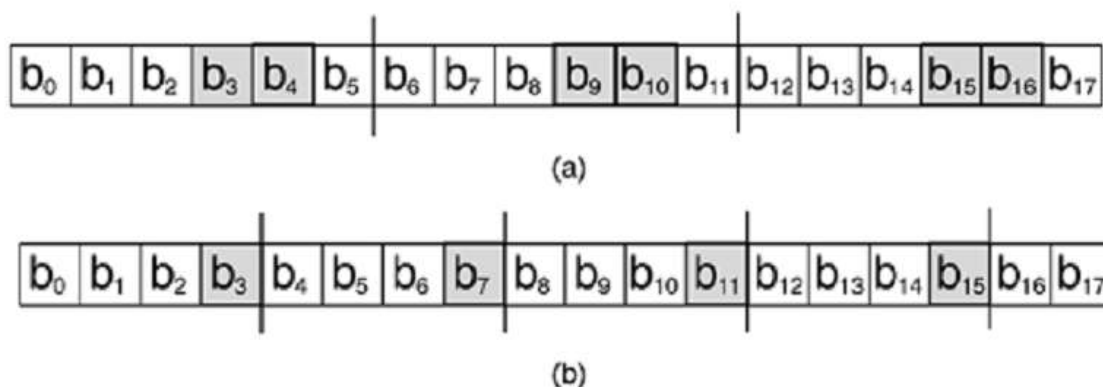
Arquivos correspondentes no Matlab:

O código que realiza a codificação convolucional de uma seqüência de bits encontra-se no arquivo *tx\_conv\_encoder.m*.

### 2.2.3 Apagamento de bits:

Sistemas de comunicação geralmente oferecem um conjunto de taxas de dados possíveis para a transmissão. No caso do IEEE 802.11a, tem-se oito taxas diferentes, que são 6, 9, 12, 18, 24, 36, 48 e 54 Mbits/s, conforme mostra a tabela 2.1. Uma solução prática para a implementação destas diferentes taxas é o apagamento de bits, que permite gerar diferentes taxas sem a necessidade de implementar diferentes codificadores no transmissor e decodificadores convolucionais no receptor. A idéia básica desta técnica é deixar de transmitir determinados bits da saída do codificador, aumentando desta forma a taxa original do código. Este aumento na taxa diminui a distância livre, mas geralmente a distância livre resultante é muito próxima da distância ótima que seria obtida utilizando-se um codificador convolucional específico para cada taxa de dados de transmissão. O receptor insere bits “dummy” para substituir os bits apagados no transmissor, de maneira que é necessário apenas um par codificador/decodificador para gerar as diferentes taxas de código requeridas.

Os bits que não são transmitidos (apagados) são definidos por um *padrão de apagamento*, que é simplesmente um mapeamento de quais bits serão apagados dentro de um certo período de bits. A figura 2.4 mostra dois padrões diferentes de apagamento utilizados pela IEEE 802.11a. O padrão (a) é utilizado para gerar um taxa de código de  $\frac{3}{4}$  a partir do código convolucional original de  $\frac{1}{2}$ . Este padrão de apagamento tem um período de 6 bits, e os bits 3 e 4 são apagados (não transmitidos) em cada período de bits. A taxa de apagamento é igual a  $\frac{4}{6} = \frac{2}{3}$  e a taxa global de código é igual a  $\frac{1}{\left(2 \cdot \left(\frac{2}{3}\right)\right)} = \frac{3}{4}$ , visto que  $\frac{2}{3}$  dos bits codificados originais são apagados. O outro padrão de apagamento possui uma taxa de código de  $\frac{2}{3}$ . O padrão mostrado na figura 2.5 tem um período de 4 bits, sendo que o quarto bit é apagado. A taxa de apagamento é, portanto, de  $\frac{3}{4}$ , levando a uma taxa global de código de  $\frac{1}{\left(2 \cdot \left(\frac{3}{4}\right)\right)} = \frac{2}{3}$ .



**Figura 2.5. Dois padrões diferentes de apagamento utilizados pela IEEE 802.11a.**

A tabela 2.3 mostra as distâncias livres e os ganhos de código assintóticos para três taxas de código utilizadas no IEEE 802.11a. A tabela mostra a taxa ótima para os códigos  $\frac{3}{4}$  e  $\frac{2}{3}$ . Pode-se aferir que a perda de performance, devido à utilização de códigos com apagamento ao invés dos códigos ótimos, é muito pequena. Esta tabela não mostra a distância livre e o ganho de código assintótico com apagamento para a taxa de  $\frac{1}{2}$ , pois esta é a taxa original obtida por codificação convolucional sem apagamentos.

**Tabela 2.3. Distâncias livres dos códigos convolucionais de 64 estados utilizados no IEEE 802.11a.**

Taxas de Código	Distância Livre com Apagamentos	Ganho de Código com Apagamentos	Distância livre Ótima	Ganho de Código Ótimo
$1/2$	-	-	10	7.0 dB
$2/3$	6	6.0 dB	7	6.7 dB
$3/4$	5	5.7 dB	6	6.5 dB

#### **Arquivos correspondentes no Matlab:**

O código que realiza o apagamento de uma seqüência de bits encontra-se no arquivo *tx\_puncture.m*.

#### **2.2.4 Bits de preenchimento:**

Inserir-se o número de bits necessários para que o total de bits do pacote a ser transmitido forme um múltiplo inteiro do número de bits de um símbolo OFDM.

### Arquivos correspondentes no Matlab:

O código que realiza a inserção de bits de preenchimento encontra-se no arquivo *tx\_make\_int\_num\_ofdm\_syms.m*.

### 2.2.5 Interleaving:

A transposição tem como objetivo redistribuir os bits a serem transmitidos, a fim de que os erros também sejam distribuídos. Desta forma, evita-se que uma seqüência grande de dados seja corrompida de forma a tornar-se irrecuperável.

Todos os bits de dados devem ser transpostos por um bloco de transposição com um tamanho correspondente ao número de bits por símbolo OFDM,  $N_{CBPS}$ , conforme a tabela 2.1. O interleaver, ou transpositor, é definido por uma permutação em dois passos. A primeira permutação garante que dois bits codificados adjacentes são mapeados para subportadoras não adjacentes. A segunda garante que bits codificados adjacentes são mapeados alternadamente para os bits menos e mais significativos da constelação e, desta forma, longas seqüências de bits de baixa confiabilidade, menos significativos (do inglês LSB- Least Significant Bits), são evitados.

Denotar-se-á  $k$  como o índice dos bits codificados antes da primeira permutação;  $i$  o índice depois da primeira e antes da segunda permutação, e  $j$  o índice depois da segunda permutação, imediatamente antes de ser mapeado para a modulação.

A primeira permutação é definida pela seguinte regra:

$$i = \left( \frac{N_{CBPS}}{16} \right) \cdot (k \bmod 16) + \text{floor} \left( \frac{k}{16} \right) \quad k = 0, 1, \dots, N_{CBPS} - 1.$$

(2.2) A função  $\text{floor}(\cdot)$  retorna o maior inteiro que não exceda o parâmetro dentro dos parênteses.

A segunda permutação é definida por:

$$j = s \cdot \text{floor}\left(\frac{i}{s}\right) + \left(i + N_{CBPS} - \text{floor}\left(16 \cdot \frac{i}{N_{CBPS}}\right)\right) \bmod s \quad i = 0, 1, \dots, N_{CBPS} - 1. \quad (2.3)$$

O valor de  $s$  é determinado pelo número de bits codificados por subportadora,  $N_{BPSC}$ , de acordo com:

$$s = \max\left(\frac{N_{BPSC}}{2}, 1\right). \quad (2.4)$$

### Arquivos correspondentes no Matlab:

O código que executa o interleaving encontra-se nos arquivos *tx\_interleaver.m* e *tx\_gen\_intlvr\_patt.m*.

### 2.2.6 Modulação:

As subportadoras OFDM podem ser moduladas utilizando as modulações BPSK, QPSK, 16-QAM, ou 64 QAM, dependendo da taxa RATE requisitada (de acordo com a tabela 2.1). Os bits de dados de entrada, codificados pelo codificador convolucional e intercalados pelo interleaver, são então divididos em grupos de  $N_{BPSC}$  (1, 2, 4 ou 6, conforme a tabela 2.1) bits e convertidos para números complexos representando os pontos das constelações das modulações BPSK, QPSK, 16-QAM, ou 64-QAM. Esta conversão deve ser executada de acordo com o mapeamento de constelações do código de Gray, ilustrado nas figuras 2.6 a 2.9, onde o bit de entrada,  $b_0$ , é o mais recente na seqüência de bits de entrada. Os valores de saída,  $d$ , são gerados multiplicando o valor  $(I + j \cdot Q)$  por um fator de normalização  $K_{MOD}$ , como descrito na equação (2.6):

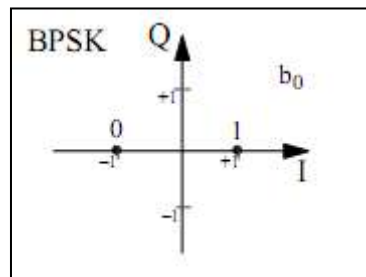
$$d = (I + j \cdot Q) \cdot K_{MOD}. \quad (2.6)$$

O fator de normalização,  $K_{MOD}$ , depende basicamente da modulação utilizada, tal como descrito na tabela 2.4. Note-se que o tipo de modulação utilizada pode ser diferente entre o início e o final da transmissão, pois o sinal modifica-se ao passar do campo SIGNAL para o campo DATA,

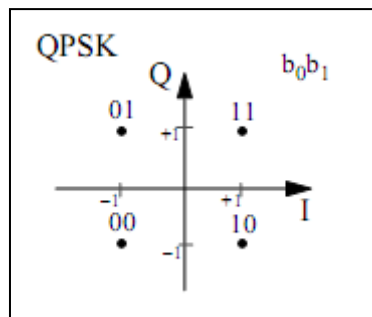
como mostrado na figura 2.1. O propósito de se utilizar um fator de normalização é o de alcançar a mesma potência média para todos os mapeamentos.

**Tabela 2.4. Fator de normalização  $K_{MOD}$  para as modulações BPSK, QPSK, 16-QAM e 64-QAM.**

Modulação	$K_{MOD}$
BPSK	1
QPSK	$\frac{1}{\sqrt{2}}$
16-QAM	$\frac{1}{\sqrt{10}}$
64-QAM	$\frac{1}{\sqrt{42}}$



**Figura 2.6. Mapeamento de constelações do código de Gray para a modulação BPSK. [IEEE 1999]**



**Figura 2.7. Mapeamento de constelações do código de Gray para a modulação QPSK. [IEEE 1999]**

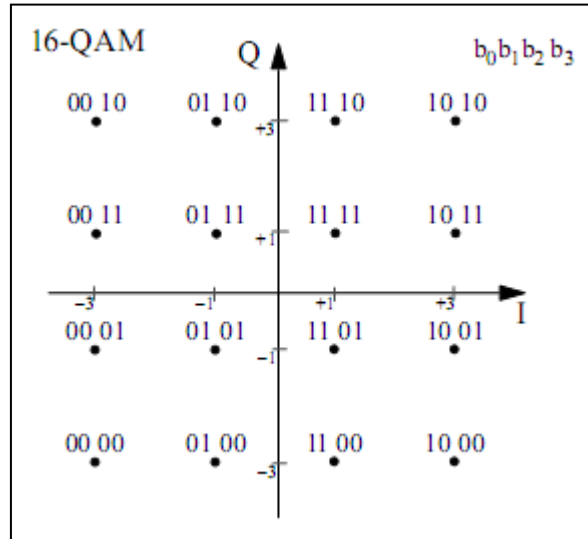


Figura 2.8. Mapeamento de constelações do código de Gray para a modulação 16-QAM. [IEEE 1999]

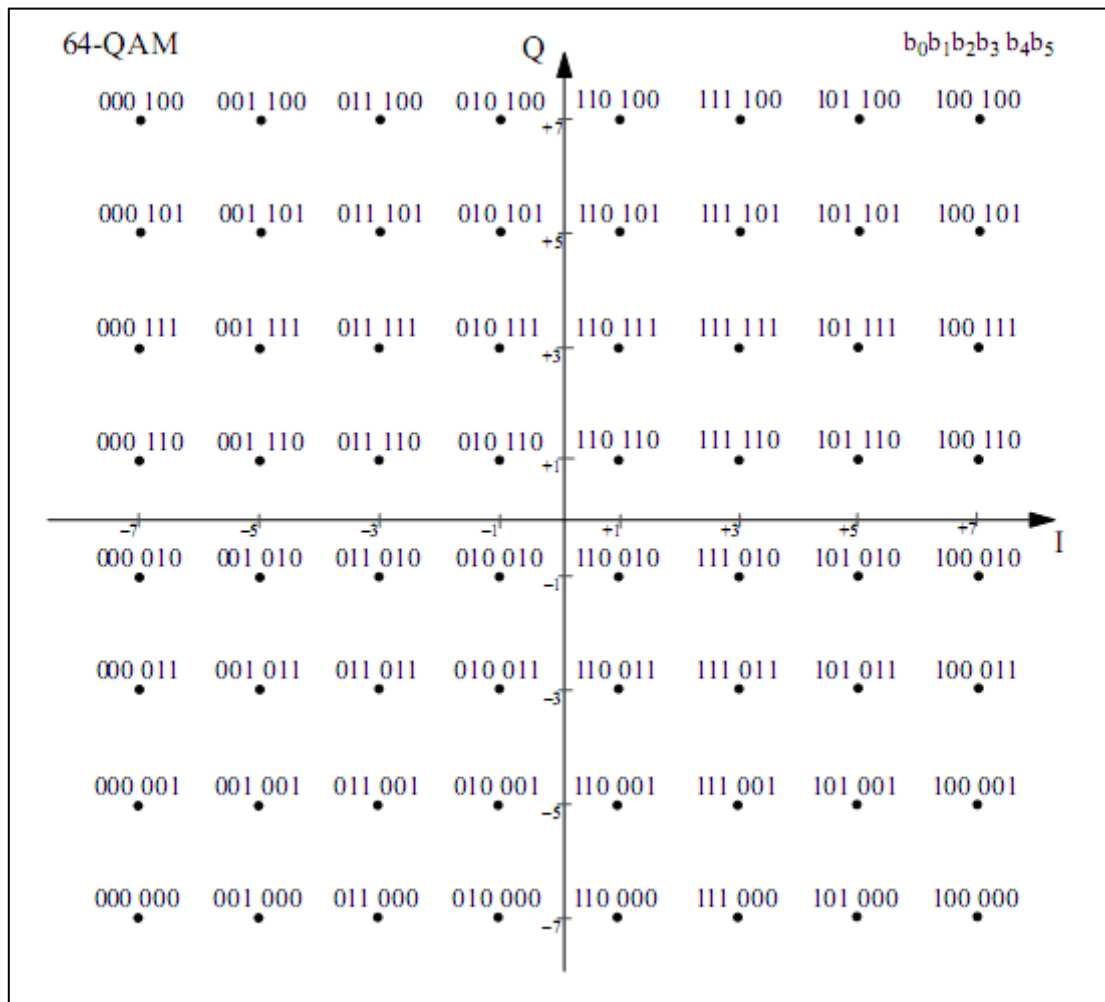


Figura 2.9. Mapeamento de constelações do código de Gray para a modulação 64-QAM. [IEEE 1999]

Para BPSK,  $b_0$  determina o valor de I, como ilustrado na tabela 2.5. Para QPSK,  $b_0$  determina o valor de I e  $b_1$  o valor de Q, como ilustrado na tabela 2.6. Para 16-QAM,  $b_0b_1$  determinam o valor de I e  $b_2b_3$  o valor de Q, como se pode ver na tabela 2.7. Para 64 QAM,  $b_0b_1b_2$  determinam o valor de I, enquanto que  $b_3b_4b_5$  determinam o valor de Q, conforme a tabela 2.8.

**Tabela 2.5. Tabela de codificação de Gray para BPSK.**

Input bit ( $b_0$ )	I-out	Q-out
0	-1	0
1	1	0

**Tabela 2.6. Tabela de codificação de Gray para QPSK.**

Input bit ( $b_0$ )	I-out	Input bit ( $b_1$ )	Q-out
0	-1	0	-1
1	1	1	1

**Tabela 2.7. Tabela de codificação de Gray para 16-QAM.**

Input bit ( $b_0b_1$ )	I-out	Input bit ( $b_2b_3$ )	Q-out
00	-3	00	-3
01	-1	01	-1
11	1	11	1
10	3	10	3



Tabela 2.8. Tabela de codificação de Gray para 64-QAM.

Input bit ( $b_0b_1b_2$ )	I-out	Input bit ( $b_3b_4b_5$ )	Q-out
000	-7	000	-7
001	-5	001	-5
011	-3	011	-3
010	-1	010	-1
110	1	110	1
111	3	111	3
101	5	101	5
100	7	100	7

### Arquivos correspondentes no Matlab:

O simulador suporta 4 tipos de modulação, são elas: BPSK, QPSK, 16-QAM e 64-QAM.

O código que realiza a modulação dos bits, de acordo com o código de Gray, encontra-se no arquivo *tx\_modulate.m*.

### 2.2.7 Diversidade de antenas:

Utilização de mais de uma antena no emissor/receptor. Geralmente, utilizam-se duas antenas: uma principal e uma auxiliar, de modo a aumentar o desempenho do sistema (taxa de erros e/ou tráfego). Esta técnica geralmente monitora o sinal de cada antena e muda automaticamente para aquela com melhor sinal. A técnica utilizada é a transformação de Radon-Hurwitz ou código de Alamouti. Este código foi projetado para transmissão com duas antenas e é definido pela seguinte matriz de codificação:

$$C = \begin{bmatrix} s_1 & s_2 \\ -s_2^* & s_1^* \end{bmatrix}$$

onde \* denota conjugado complexo.  $s_1$  é o primeiro símbolo OFDM e  $s_2$  é o segundo, de uma seqüência de símbolos a serem transmitidos. Os elementos de cada linha da matriz  $C$  representam os símbolos OFDM transmitidos em cada intervalo de tempo. Os elementos de cada coluna representam os símbolos enviados por cada antena de transmissão. Ou seja, no primeiro intervalo de tempo  $T_1$  a antena  $n_1$  transmite  $s_1$ , enquanto que a segunda antena  $n_2$  transmite  $s_2$ . No segundo intervalo de tempo  $T_2$ ,  $n_1$  transmite  $-s_2^*$  e  $n_2$  transmite  $s_1^*$ .

Este arranjo resulta em ortogonalidade entre os símbolos transmitidos, o que possibilita alcançar altas performances de erro de bit e de pacotes na recepção, utilizando algoritmos apropriados de processamento de diversidade de antenas, conforme será visto com maiores detalhes no item 4.1.7.

#### **Arquivos correspondentes no Matlab:**

O código que distribui o sinal OFDM nas antenas de transmissão encontra-se no arquivo *tx\_diversity.m*, enquanto que o código que realiza a transformação de Radon-Hurwitz encontra-se no arquivo *tx\_radon\_hurwitz*.

#### **2.2.8 Inserção de portadoras piloto:**

Em cada símbolo OFDM, quatro das subportadoras são dedicadas para a transmissão de sinais piloto, cujo objetivo é tornar a detecção coerente robusta contra offsets de frequência e ruído de fase. Estes sinais piloto devem ser alocados nas subportadoras de índices -21, -7, 7 e 21 e modulados com BPSK por uma seqüência pseudo-binária, de modo a prevenir a geração de linhas espectrais (polarização).

A seqüência de números complexos resultante da modulação explicitada em 2.2.6 é dividida em grupos de  $N_{SD} = 48$  números complexos:

$$\begin{aligned} d_{k,n} &\equiv d_k + N_{SD} \cdot n & k = 0, \dots, N_{SD} - 1, \\ n &= 0, \dots, N_{SYM} - 1. \end{aligned} \quad (2.7)$$

Onde  $N_{SYM}$  é o número de símbolos OFDM, e  $d_{k,n}$  é o número complexo correspondente à subportadora  $k$  do  $n$ ésimo símbolo OFDM.

Um símbolo OFDM,  $r_{DATA,n}(t)$ , é definido como:

$$\begin{aligned} r_{DATA,n}(t) = \\ w_{T_{SYM}}(T) \left( \sum_{k=0}^{N_{SD}-1} d_{k,n} \cdot e^{j \cdot 2 \cdot \pi \cdot M(k) \cdot \Delta_F \cdot (t - T_{GI})} + p_{n+1} \cdot \sum_{k=-\frac{N_{ST}}{2}}^{\frac{N_{ST}}{2}} P_k \cdot e^{j \cdot 2 \cdot \pi \cdot M(k) \cdot \Delta_F \cdot (t - T_{GI})} \right). \end{aligned} \quad (2.8)$$

onde  $T_{SYM}, T_{GI}, N_{ST}, N_{SD}$  e  $\Delta_F$  estão definidos na tabela 2.2.

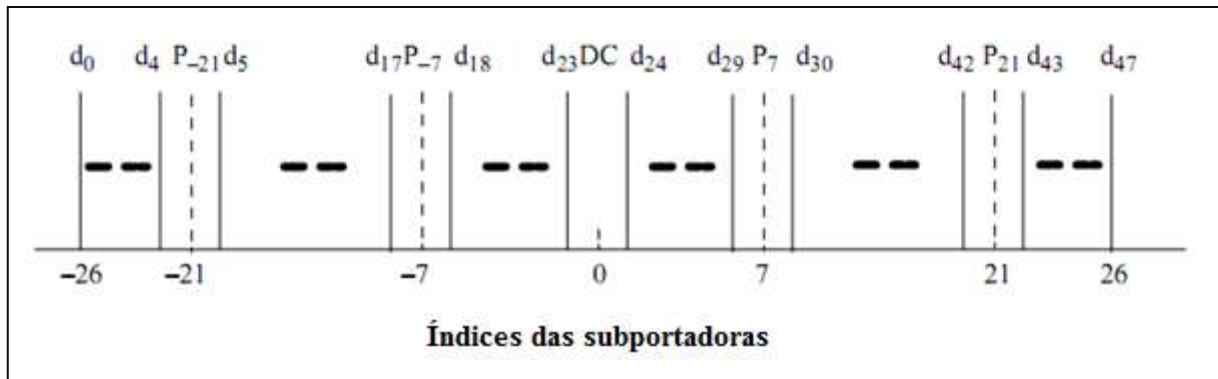
$w_{T_{SYM}}(T)$  é uma função que define uma janela no tempo, a qual limita o símbolo  $r_{DATA,n}(t)$  a um período de símbolo igual a  $T_{SYM}$ .

A função  $M(k)$  define um mapeamento dos índices lógicos das subportadoras de 0 a 47 para os índices de frequência de -26 a 26, evitando os locais das subportadoras piloto e da subportadora 0 (nível DC (Direct Current)).

$$\begin{aligned} M(k) = \left\{ \begin{array}{ll} k - 26 & 0 \leq k \leq 4 \\ k - 25 & 5 \leq k \leq 17 \\ k - 24 & 18 \leq k \leq 23 \\ k - 23 & 24 \leq k \leq 29 \\ k - 22 & 30 \leq k \leq 42 \\ k - 21 & 43 \leq k \leq 47 \end{array} \right. \end{aligned} \quad (2.9)$$



A alocação de frequência para as subportadoras é mostrada na figura 2.11. A fim de evitar dificuldades na conversão digital/analógica (D/A – Digital to Analog) e analógica digital (A/D – Analog to Digital) e no bloco de conversão para RF, a subportadora em DC (índice zero) não é utilizada.



**Figura 2.11. Alocação de frequência para as subportadoras.**

#### **Arquivos correspondentes no Matlab:**

O código que insere os sinais piloto que devem ser alocados nas subportadoras de índices - 21, -7, 7 e 21, encontra-se no arquivo *tx\_add\_pilot\_syms.m*.

#### **2.2.9 Transformada rápida de Fourier inversa:**

Para que se possa transmitir a informação, ela deve estar representada no domínio do tempo. Nesta etapa, passam-se os símbolos do domínio da frequência para o domínio do tempo, através da transformada rápida de Fourier inversa.

#### **Arquivos correspondentes no Matlab:**

O código que executa a transformada rápida de Fourier inversa encontra-se no arquivo *tx\_freqd\_to\_timed.m*.

### 2.2.10 Inserção do prefixo cíclico:

Adição do prefixo cíclico, que é a repetição da última parte do símbolo OFDM. A utilização deste prefixo cíclico como intervalo de guarda entre os símbolos OFDM auxilia na sincronização de pacotes, já que evita a interferência entre símbolos.

#### Arquivos correspondentes no Matlab:

O código que executa a adição do prefixo cíclico encontra-se no arquivo *tx\_add\_cyclic\_prefix.m*.

### 2.2.11 Preâmbulo: PLCP preamble field

Criação do preâmbulo, necessário para fins de detecção de pacote, controle automático de ganho, sincronização de frequência e estimação de offset de frequência no receptor. Este preâmbulo é composto de 12 símbolos: 10 repetições de uma seqüência de “short training” (utilizada para AGC (controle automático de ganho), diversidade de antenas, sincronização, e sintonia grossa de frequência no receptor), 2 repetições de uma seqüência de “long training” (utilizada para estimação de canal e sintonia de frequência no receptor) e um apêndice de GI, necessário para evitar ISI (Inter-Symbol Interference)

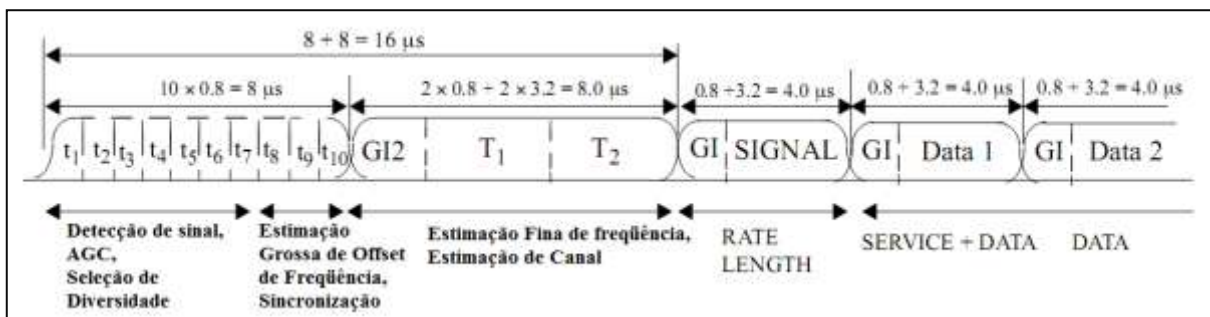


Figura 2.12. Estrutura de treinamento IEEE 802.11a

#### Arquivos correspondentes no Matlab:

O código que cria e insere o preâmbulo no pacote OFDM encontra-se no arquivo *tx\_gen\_preamble.m*.

### **2.2.12 Inserção de ruído de fase:**

Geração de ruído de fase, que surge ao transladar-se o sinal da banda base para a banda passante. Esta translação é realizada por um oscilador, que modula o sinal original por meio de uma portadora em RF.

#### **Arquivos correspondentes no Matlab:**

O código que insere ruído de fase no sinal a ser transmitido encontra-se no arquivo *phase\_noise\_model.m*.

### **2.2.13 Amplificador de potência:**

Modelagem do amplificador de potência do transmissor, o qual introduz não linearidades no sinal transmitido, como distorção Amplitude/Amplitude e Amplitude/Fase.

#### **Arquivos correspondentes no Matlab:**

O código que insere não linearidades, correspondentes a um Amplificador de Potência, no sinal a ser transmitido, encontra-se no arquivo *tx\_power\_amplifier.m*.

### 3. CANAL

#### 3.1 Descrição do canal de rádio-móvel

O sinal OFDM transmitido na banda básica pode ser modelado como um processo complexo passa-baixas relacionado com o envelope complexo  $x(t)$  do sinal na banda passante pela seguinte expressão:

$$s(t) = x(t) \cdot e^{-j \cdot 2\pi \cdot f_c \cdot t}.$$

(3.1)

onde  $f_c$  é a frequência da portadora.

Assumindo espalhamentos Gaussianos e múltiplos caminhos de propagação até o receptor, o canal pode ser caracterizado, segundo [PRO 2001], por atrasos de propagação variáveis no tempo, fatores de atenuação, e deslocamentos Doppler. A resposta impulsiva do canal variável no tempo pode ser dada por:

$$c(\tau_n, t) = \sum_n \alpha_n(\tau_n(t)) \cdot e^{-j \cdot 2\pi \cdot f_{D_n} \cdot \tau_n(t)} \cdot \delta[t - \tau_n(t)]. \quad (3.2)$$

onde:

$c(\tau_n, t)$  é a resposta do canal no tempo  $t$  devido a um impulso aplicado no tempo  $t - \tau_n(t)$ ;

$\alpha_n(t)$  é o fator de atenuação para o sinal recebido no  $n$ ésimo caminho;

$\tau_n(t)$  é o atraso de propagação para o  $n$ ésimo caminho;

$f_{D_n}$  é o deslocamento Doppler para o sinal recebido no  $n$ ésimo caminho.

O efeito Doppler devido ao movimento relativo entre o emissor e o receptor pode ser expresso da seguinte forma:

$$f_{D_n} = \frac{v \cdot \cos(\phi_n)}{\lambda}. \quad (3.3)$$

Onde :

$v$  é a velocidade relativa entre transmissor e receptor em m/s;



$\lambda$  é o comprimento de onda da portadora em metros;

$\phi_n$  é um ângulo de fase aleatório uniformemente distribuído entre 0 e  $2\pi$ .

Desta forma, a saída do canal é simplesmente a convolução entre as equações 3.1 e 3.2, ou seja:

$$z(t) = c(\tau_n, t) * s(t) , \quad (3.4)$$

$$z(t) = \sum_n \alpha_n(\tau_n(t)) \cdot e^{-j \cdot 2\pi \cdot (f_{Dn} + f_c) \cdot \tau_n(t)} \cdot x[t - \tau_n(t)] \cdot e^{-j \cdot 2\pi \cdot f_c \cdot t} , \quad (3.5)$$

$$z(t) = \sum_n \beta_n \cdot x[t - \tau_n(t)] \cdot e^{-j \cdot 2\pi \cdot f_c \cdot t} . \quad (3.6)$$

onde:

$$x[t - \tau_n(t)] = \delta[t - \tau_n(t)] * x(t) . \quad (3.7)$$

$\beta_n$  é um processo aleatório Gaussiano. O envelope complexo de  $c(\tau_n, t)$ , em qualquer instante  $t$ , apresenta uma distribuição Rayleigh, visto que é constituído pela soma de vários processos aleatórios Gaussianos. A função densidade de probabilidades para um canal de desvanecimento Rayleigh é dada por:

$$f_Z(Z) = \frac{Z}{\sigma^2} e^{-\frac{Z^2}{2\sigma^2}} . \quad (3.9)$$

Um canal com este tipo de distribuição é geralmente chamado de canal com desvanecimento Rayleigh. No caso de haver espalhamentos fixos e uma linha de visada (LOS- line of sight) no caminho até o receptor, o envelope de  $c(\tau_n, t)$  possui uma distribuição Rice, cuja função densidade de probabilidades é dada por :

$$f_Z(Z) = \frac{Z}{\sigma^2} \cdot I_0\left(\frac{Z\eta}{\sigma^2}\right) \cdot e^{-\frac{Z^2 + \eta^2}{2\sigma^2}} . \quad (3.10)$$

Onde  $I_0$  é a função de Bessel modificada de tipo zero e  $\eta$  é a média devida aos espalhamentos fixos ou caminhos LOS. Neste caso, o canal é chamado de canal de desvanecimento Rician.

Pode-se demonstrar que a função de autocorrelação para  $c(\tau, t)$  é dada por

$$\Lambda_c(\tau, \Delta t) = E\{c(\tau, t) \cdot c^*(\tau, t + \Delta t)\}. \quad (3.11)$$

Esta função de auto-correlação pode ser medida empiricamente transmitindo pulsos muito estreitos e fazendo a correlação cruzada do sinal recebido com uma versão conjugada e atrasada dele mesmo. Além disto, a potência média de saída do canal é encontrada fazendo  $\Delta t = 0$ , ou seja,  $\Lambda_c(\tau, 0) \equiv \Lambda_c(\tau)$ .

Esta quantidade é chamada de *perfil de intensidade de multipercurso* ou *espectro de potência atrasado* do canal. A faixa de valores de  $\tau$  nos quais  $\Lambda_c(\tau)$  é não-nula é chamada de *espalhamento multipercurso do canal*, simbolizado por  $T_m$ . O recíproco do espalhamento multipercurso é uma medida da largura de banda coerente do canal, dada por:

$$B_m \approx \frac{1}{T_m}. \quad (3.12)$$

Sinais de informação cuja largura de banda é pequena em comparação com a largura de banda coerente do canal sofrem desvanecimento não seletivo em frequência, ou desvanecimento plano (flat fading). Entretanto, se o sinal de informação tiver largura de banda maior que a largura de banda coerente do canal, o canal é dito seletivo em frequência.

Canais cujas medidas estatísticas permanecem aproximadamente constantes ao longo de vários intervalos de símbolo são considerados canais de desvanecimento lento, ao contrário de canais cujas medidas estatísticas variem rapidamente durante um intervalo de símbolo. Canais assim são chamados de canais de desvanecimento rápido. Em geral, canais de wireless em ambientes fechados são bem-caracterizados por meio de modelos de desvanecimento lento.

### 3.2 Descrição da implementação do canal de rádio móvel no simulador 802.11a

A figura 3.1 mostra o diagrama de blocos do canal de rádio móvel, do ponto de vista da implementação no simulador 802.11a.

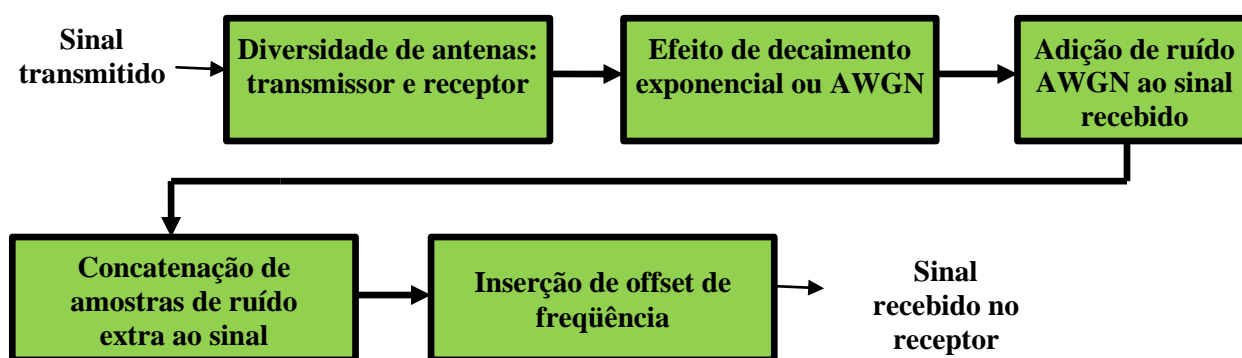


Figura 3.1. Diagrama de blocos do canal de rádio-frequência.

Descreve-se nas próximas seções o funcionamento de cada um dos blocos citados na figura 3.1.

#### 3.2.1 Diversidade de antenas

A utilização de mais de uma antena auxilia na compensação dos efeitos de múltiplos caminhos no sinal transmitido. Podem ser utilizadas uma ou duas antenas tanto no transmissor quanto no receptor. O efeito do canal será aplicado ao sinal emitido por cada uma das antenas.

#### Arquivos correspondentes no Matlab:

O código que busca o número de antenas utilizados na diversidade de antenas encontra-se no arquivo *get\_n\_antennas.m*.

### 3.2.2 Decaimento exponencial ou AWGN

O canal pode apresentar decaimento exponencial ou canal AWGN (Additive White Gaussian Noise). Neste último, o canal pode ser considerado como apenas um ganho unitário sem outros efeitos, visto que a adição de ruído AWGN será realizada nos blocos posteriores. No caso discreto, modela-se o sinal recebido como uma matriz *rx\_signal*; cada linha nesta matriz representa as amostras do sinal recebido por cada antena do receptor. No caso de existirem mais de uma antena transmissora e/ou mais de uma antena receptora, cada linha (amostras do sinal recebido) da matriz *rx\_signal* é o resultado do somatório dos vários caminhos (sinal resultante do efeito do canal sobre o sinal emitido por cada uma das antenas transmissoras) que chegam a cada antena receptora.

O canal de desvanecimento exponencial possui uma resposta impulsiva formada por amostras complexas, com magnitude possuindo comportamento Rayleigh. Cada uma destas amostras possui um valor médio diferente, fazendo com que o envelope da resposta impulsiva deste canal apresente um formato de decaimento exponencial.

Temos então, para canal caracterizado por decaimento exponencial:

Resposta impulsiva do canal:

$$cir(k) = stdDevReOrIm(k) \cdot ([randn] + j \cdot [randn]) \quad (3.13)$$

Desvio padrão:

$$stdDevReOrIm(k) = \sqrt{\frac{env(k)}{2}} \quad (3.14)$$

Envelope do decaimento exponencial:

$$env(k) = var0 \cdot e^{\left( -\frac{k}{SampFreq \cdot (ExpDecayTrms \cdot (10^{-9}))} \right)}$$

(3.15)

$$var0 = \frac{\left(1 - e^{-\frac{1}{SampFreq \cdot (ExpDecayTrms \cdot (10^{-9}))}}\right)}{\left(1 - e^{-1 \cdot \left(\frac{SampFreq}{ExpDecayTrms \cdot (10^{-9})}\right)}\right)}; \quad (3.16)$$

para  $0 \leq k \leq Kmax$  ;

$$Kmax = \lceil (10 \cdot (ExpDecayTrms \cdot (10^{-9})) \cdot SampFreq) \rceil; \quad (3.17)$$

Variância:

onde *randn* é uma matriz de (*n\_channels*) linhas e (*Kmax*) colunas de variáveis aleatórias de distribuição normal, com desvio padrão zero e variância iguala a 1; *ExpDecayTrms* é o período de decaimento exponencial em nanossegundos; e *k* é uma base de tempo discreto.

#### Arquivos correspondentes no Matlab:

O código que modela o canal de rádio-móvel conforme a escolha do usuário (AWGN ou decaimento exponencial) encontra-se no arquivo *get\_channel\_ir.m*.

### 3.2.3 Adição de Ruído AWGN ao Sinal Recebido

São inseridas amostras de ruído gaussiano branco ao sinal recebido. A variância é definida por meio de um vetor de razão sinal/ruído SNR. Este valor é multiplicado por 64/52 que é um fator de escala para normalizar o ruído, por um fator que relaciona o número de sub-portadoras efetivamente utilizadas (52) com o número total de subportadoras (64). Portanto, este fator é  $52/64=0.812$ .

Temos então, para canal AWGN:

Variância do ruído:

$$noise = \sqrt[2]{(noise_{var})} \cdot ([randn(n_{rx_{antennas}}, len)] + j \cdot [randn(n_{rx_{antennas}}, len)]); \quad (3.18)$$

$$noise_{var} = \left(\frac{64}{52}\right) \cdot \frac{SNR}{2 \cdot (10)^{\frac{10}{10}}}$$

(3.19)

onde  $SNR$  é a razão sinal/ruído;  $n_{rx\_antennas}$  é o número de antenas do receptor,  $noise$  é o ruído gaussiano branco,  $len$  é número de elementos do vetor de sinal recebido  $rx\_signal$  e  $randn$  é uma matriz de ( $n_{rx\_antennas}$ ) linhas e ( $len$ ) colunas de variáveis aleatórias de distribuição normal, com desvio padrão zero e variância igual a 1.

### Arquivos correspondentes no Matlab:

O código que simula a transmissão de um sinal OFDM por um canal de rádio-móvel encontra-se no arquivo *channel.m*, enquanto que o código que contém parâmetros necessários para esta simulação encontra-se no arquivo *set\_sim\_consts.m*.

### 3.2.4 Concatenação de amostras de ruído extra ao sinal

São inseridas amostras adicionais de ruído AWGN, conforme número definido na estrutura *sim\_consts*, concatenadas no início e no final do vetor do sinal recebido. O ruído ao final é utilizado para simular os efeitos da incorreta sincronização no receptor, enquanto que o inicial tem a função de testar o algoritmo de detecção de pacotes (em ambos os casos, este efeito representa a situação crítica em que o receptor recebe apenas amostras de ruído, sem presença de amostras do sinal transmitido).

Concatenação das amostras de ruído extra e do ruído final:

$$rx\_signal = [extra\_noise \ rx\_signal \ end\_noise]; \quad (3.20)$$

onde *ExtraNoiseSamples* é o número de elementos de ruído desejados.

Amostras extra de ruído gaussiano branco:

$$extra\_noise = \sqrt[2]{(noise\_var)} \cdot ([randn(n_{rx\_antennas}, ExtraNoiseSamples)] + j \cdot [randn(n_{rx\_antennas}, ExtraNoiseSamples)]); \quad (3.21)$$

Adição de ruído final:

$$end\_noise = \sqrt[2]{(noise\_var)} \cdot ([randn(n_{rx\_antennas}, 170)] + j \cdot [randn(n_{rx\_antennas}, 170)]); \quad (3.22)$$

### Arquivos correspondentes no Matlab:

O código que simula a transmissão de um sinal OFDM por um canal de rádio-móvel encontra-se no arquivo *channel.m*, enquanto que o código que contém parâmetros necessários para esta simulação encontra-se no arquivo *set\_sim\_consts.m*.

### 3.2.5 Inserção de offset de frequência

Durante a propagação do sinal ao longo do canal, bem como no oscilador do receptor, tem-se uma alteração na frequência das subportadoras, devido a efeitos não-lineares, tais como o efeito Doppler.

Modelagem matemática:

$$offset_{sign}(time\_base) = input\_signal(time\_base).e^{\frac{j.2\pi.freq\_offset.time\_base}{SampFreq}}. \quad (3.23)$$

onde  $offset_{sign}$  é o sinal recebido corrompido pelo offset de frequência,  $freq\_offset$  é o offset de frequência,  $input\_signal$  é o sinal transmitido,  $SampFreq$  é a frequência de amostragem e  $time\_base$  é a variável tempo.

### Arquivos correspondentes no Matlab:

O código que insere um offset de frequência no sinal OFDM transmitido encontra-se no arquivo *create\_freq\_offset.m*.

## 4. RECEPTOR

A figura 4.1 mostra o diagrama de blocos do receptor, do ponto de vista da implementação no simulador 802.11a.

### 4.1 Descrição da implementação do receptor no simulador 802.11a

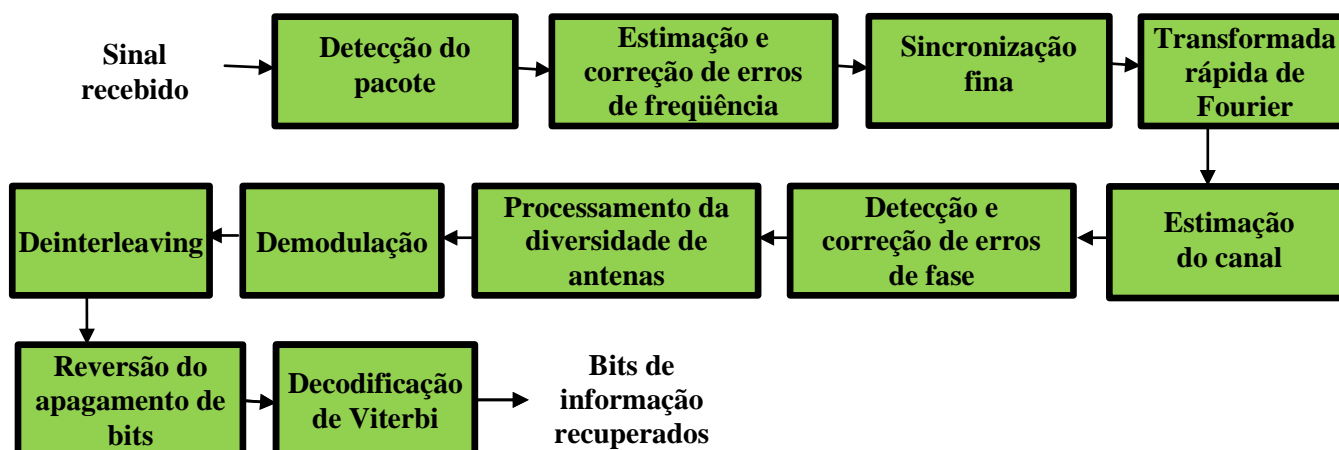


Figura 4.1. Diagrama de blocos do receptor do simulador.

Descreve-se nas próximas seções o funcionamento de cada um dos blocos citados na figura 4.1.

#### 4.1.1 Detecção do pacote

A detecção de pacote é a tarefa de encontrar uma estimativa da chegada de um pacote de dados de entrada. Este é o primeiro algoritmo de sincronização que é efetuado. Logo, o restante do processo de sincronização depende de uma boa performance de detecção de pacote. A detecção de pacotes pode ser genericamente descrita como um teste de hipóteses binário. Este teste consiste em duas afirmativas complementares acerca de um parâmetro de interesse. Estas afirmativas são chamadas de hipótese nula,  $H_0$ , e de hipótese alternativa,  $H_1$ . No teste de detecção de pacotes, as hipóteses indicam se um pacote está presente ou não, da forma explicitada abaixo:

$H_0$ : O pacote não está presente

$H_1$ : O pacote está presente



Na prática, este teste é realizado por meio de comparação de uma variável de decisão  $m_n$  com um limiar pré-definido  $T_h$ . A detecção de pacote segue então o esquema abaixo:

$H_0: m_n < T_h \rightarrow$  O pacote não está presente

$H_1: m_n > T_h \rightarrow$  O pacote está presente

A performance do algoritmo de detecção de pacotes pode ser resumido em duas probabilidades: probabilidade de detecção  $P_d$  e probabilidade de alarme falso  $P_{fa}$ .  $P_d$  é a probabilidade de detectar um pacote quando este realmente está presente, o que faz com que uma alta  $P_d$  seja desejável.  $P_{fa}$  é a probabilidade de que o algoritmo decida incorretamente que um pacote está presente, enquanto na verdade não há nenhum pacote. Desta forma, deseja-se tornar  $P_{fa}$  o menor possível. Entretanto, um aumento em  $P_d$  implica um aumento em  $P_{fa}$ , o que leva a concluir que o desenvolvedor do algoritmo deve alcançar um ponto de equilíbrio entre estes pontos conflitantes. Geralmente, pode-se dizer que um alarme falso é menos severo que não detectar um pacote, visto que no segundo caso sempre há perda de dados. No caso de alarme falso, o que pode ocorrer é uma provável perda de dados, se enquanto o algoritmo estiver ocupado tentando sincronizar este falso pacote um pacote de dados for recebido.

A estrutura do preâmbulo WLAN possibilita ao receptor utilizar um algoritmo simples e eficiente para detectar pacotes. Este método aproveita a periodicidade dos símbolos de treinamento curto localizados no início do preâmbulo, e é chamado de algoritmo “delay and correlate” (atrasa e correlaciona). A figura 4.2 mostra duas janelas deslizantes,  $C$  e  $P$ . A janela  $C$  é a correlação cruzada entre o sinal recebido e uma versão deslocada do sinal recebido. O deslocamento  $z^{-D}$  é igual ao período do início do preâmbulo; por exemplo, na norma IEEE 802.11a,  $D$  é igual a 16, o período dos símbolos de treinamento curto. A janela  $P$  calcula a energia do sinal recebido durante a janela de correlação cruzada. O valor de  $P$  é utilizado para normalizar a decisão estatística, de modo que a torne independente do valor absoluto da energia recebida.

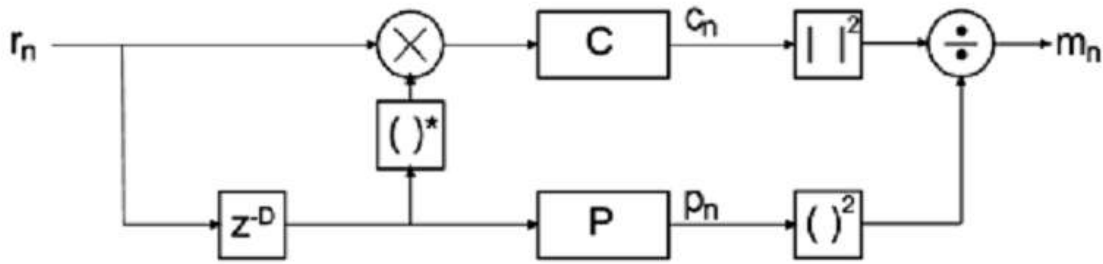


Figura 4.2. Estrutura de fluxo do sinal e do algoritmo de atraso e correlação.

$$c_n = \sum_{k=0}^{L-1} r_{n+k} \cdot r_{n+k+D}^* \quad (4.1)$$

$$p_n = \sum_{k=0}^{L-1} r_{n+k+D} \cdot r_{n+k+D}^* = \sum_{k=0}^{L-1} |r_{n+k+D}|^2 \quad (4.2)$$

$$m_n = \frac{|c_n|^2}{(p_n)^2} \quad (4.3)$$

#### Arquivos correspondentes no Matlab:

O código que busca o momento de chegada de um pacote 802.11a encontra-se no arquivo *rx\_find\_packet\_edge.m*.

#### 4.1.2 Estimação e correção de erros de frequência

Um dos maiores problemas dos sistemas OFDM é a sensibilidade para desvios de frequência da portadora. Observa-se uma degradação na relação sinal para ruído, causada basicamente por dois fenômenos: redução da amplitude da subportadora desejada e ICI (Inter-Carrier Interference) causada por portadoras vizinhas.

A perda de amplitude ocorre porque a subportadora desejada já não é mais amostrada no pico da função sinc  $\left(\frac{\text{sen}(x)}{x}\right)$  da DFT (Discrete Fourier Transform). Portadoras adjacentes causarão interferência, porque estas não são mais amostradas nos pontos de cruzamento com zero de suas respectivas funções sinc.

Para o padrão IEEE 802.11a, a solução mais importante são os algoritmos baseados na informação de treinamento contida na estrutura do sinal transmitido. O preâmbulo permite ao

receptor utilizar algoritmos de máxima verossimilhança para estimar e corrigir offsets de frequência, antes que a recepção do restante do pacote se inicie.

No simulador, trabalha-se com um destes estimadores de máxima verossimilhança que operam no sinal recebido no domínio do tempo. A informação de treinamento necessária é de pelo menos dois símbolos consecutivos repetidos. O preâmbulo dos padrões WLAN satisfaz esse requerimento tanto para os símbolos de treinamento curtos quanto para os símbolos de treinamento longo.

Consideremos o sinal transmitido  $s_n$ , logo o modelo complexo da banda básica do sinal transmitido é dado por

$$y_n = s_n \cdot e^{j \cdot 2\pi \cdot f_{tx} \cdot n \cdot T_s} . \quad (4.4)$$

Onde  $f_{tx}$  é a frequência da portadora do transmissor. Depois de o receptor converter para a banda básica o sinal com uma frequência da portadora  $f_{rx}$ , o sinal complexo recebido na banda básica  $r_n$ , desconsiderando o ruído, é

$$\begin{aligned} r_n &= s_n \cdot e^{j \cdot 2\pi \cdot f_{tx} \cdot n \cdot T_s} \cdot e^{-j \cdot 2\pi \cdot f_{rx} \cdot n \cdot T_s} , \\ &= s_n \cdot e^{j \cdot 2\pi \cdot (f_{tx} - f_{rx}) \cdot n \cdot T_s} . \end{aligned} \quad (4.5)$$

O estimador de offset de frequência é desenvolvido como segue, iniciando com a variável intermediária  $z$ :

$$z = \sum_{n=0}^{L-1} r_n \cdot r_{n+D}^* , \quad (4.6)$$

$$= s_n \cdot e^{j \cdot 2\pi \cdot f_{tx} \cdot n \cdot T_s} \cdot e^{-j \cdot 2\pi \cdot f_{rx} \cdot n \cdot T_s} , \quad (4.7)$$

$$= \sum_{n=0}^{L-1} s_n \cdot e^{j \cdot 2\pi \cdot f_{\Delta} \cdot n \cdot T_s} \cdot (s_{n+D} \cdot e^{j \cdot 2\pi \cdot f_{\Delta} \cdot (n+D) \cdot T_s})^* , \quad (4.8)$$

$$= \sum_{n=0}^{L-1} s_n \cdot s_{n+D}^* \cdot e^{j \cdot 2\pi \cdot f_{\Delta} \cdot n \cdot T_s} \cdot e^{-j \cdot 2\pi \cdot f_{\Delta} \cdot (n+D) \cdot T_s} ,$$

(4.9)

$$= e^{-j \cdot 2\pi \cdot f_{\Delta} \cdot D \cdot T_s} \cdot \sum_{n=0}^{L-1} |s_n|^2 .$$

A equação acima é uma soma de variáveis complexas com um ângulo proporcional ao offset de frequência. Finalmente, a estimativa do erro de frequência é dada por

$$\hat{f}_{\Delta} = -\frac{1}{2\pi \cdot D \cdot T_s} \angle z .$$

(4.10)

onde o operador  $\angle z$  corresponde ao ângulo do argumento  $z$ .

### Arquivos correspondentes no Matlab:

O código que estima e corrige offsets de frequência encontra-se no arquivo *rx\_frequency\_sync.m*.

### 4.1.3 Sincronização fina - Sincronização (timing) de símbolo

Sincronização de símbolo é a tarefa de buscar o exato momento em que um símbolo OFDM individual começa e onde termina. O resultado desta busca define a janela da DFT, isto é, o conjunto de amostras usadas para calcular a DFT para cada símbolo OFDM recebido. O resultado da DFT é utilizado para a demodulação das subportadoras contidas no símbolo OFDM.

Receptores IEEE 802.11a possuem conhecimento prévio da estrutura do preâmbulo dos pacotes a serem recebidos, o que possibilita ao receptor utilizar um algoritmo de sincronização baseado em uma simples correlação cruzada. Após o detector de pacotes ter fornecido uma estimativa do início do pacote, o algoritmo de sincronização refina esta estimativa para uma precisão ao nível de amostra. Este refinamento é desempenhado através do cálculo da correlação cruzada do sinal recebido  $r_n$  e uma referência conhecida  $t_k$ ; por exemplo, o final dos símbolos de treinamento curto ou o início dos símbolos de treinamento longo, para descobrir a estimativa de sincronização de símbolo. A equação (4.13) mostra como calcular a correlação cruzada. O valor de

n que corresponde ao máximo valor absoluto da correlação cruzada é a estimativa de sincronização de símbolo.

$$\hat{t}_s = \sum_{k=0}^{L-1} \arg \max_n |r_{n+k} \cdot t_k^*|^2. \quad (4.11)$$

Na equação (4.13), o comprimento  $L$  da correlação cruzada determina a performance do algoritmo. Valores maiores melhoram a performance, mas também aumentam a quantidade de cálculos computacionais requeridos.

#### **Arquivos correspondentes no Matlab:**

O código que busca o exato momento em que um símbolo OFDM individual começa e termina encontra-se no arquivo *rx\_fine\_time\_sync.m*.

#### **4.1.4 Transformada rápida de Fourier**

Necessita-se transformar o sinal recebido do domínio do tempo para o domínio da frequência, para que se possa adequadamente analisar este sinal separado em cada uma de suas subportadoras. Esta passagem é realizada pela Transformada Rápida de Fourier.

A Transformada rápida de Fourier (em inglês fast Fourier transform, ou FFT) é um algoritmo eficiente para se calcular a Transformada discreta de Fourier (DFT) e a sua inversa.

#### **Arquivos correspondentes no Matlab:**

O código que transforma o sinal recebido do domínio do tempo para o domínio da frequência encontra-se no arquivo *rx\_timed\_to\_freqd.m*.

#### **4.1.5 Estimação do canal**

A estimação de canal é a tarefa de estimar a resposta em frequência do canal de rádio móvel percorrido pelo sinal transmitido antes de alcançar a antena do receptor. A resposta impulsiva de

um canal de rádio móvel pode ser representada por um filtro FIR (Finite Impulse Response) discreto:

$$h(\tau; t) = \sum_n \alpha_n(t) \cdot e^{-j \cdot 2 \cdot \pi \cdot f_c \cdot \tau_n(t)} \cdot \delta(\tau - \tau_n(t)). \quad (4.12)$$

Em aplicações de WLAN, geralmente assume-se que o canal é quase estacionário, ou seja, que ele não muda suas características durante o intervalo de um pacote. Com esta suposição, a dependência com relação ao tempo desaparece e a equação (4.14) torna-se:

$$h(\tau) = \sum_n \alpha_n \cdot e^{-j \cdot 2 \cdot \pi \cdot f_c \cdot \tau_n} \cdot \delta(\tau - \tau_n). \quad (4.13)$$

A resposta em frequência discreta do canal é obtida através da transformada de Fourier de sua resposta impulsiva:

$$H_k = DFT\{h_n\}. \quad (4.14)$$

O processo de estimação produz a saída  $\hat{H}_k$ , que é uma estimativa da resposta em frequência do canal para cada subportadora.

A estimação de canal é obrigatória para sistemas OFDM que empregam esquemas de modulação coerente. Caso contrário, a correta modulação não seria possível. O conhecimento do canal também pode melhorar a performance de sistemas OFDM com esquemas de modulação não-coerente, embora  $\hat{H}_k$  não seja realmente necessário neste caso. Este aprimoramento pode ser alcançado quando for utilizado um código de controle de erros no sistema; neste caso, conhecimento acerca do canal pode ajudar o decodificador deste código a realizar sua função com uma melhor performance.

#### 4.1.5.1 Estimação de canal no domínio da frequência

Segundo [TSE 2005], a estimação de canal no domínio da frequência pode ser efetuada por dois métodos principais. O primeiro utiliza os dados de treinamento transmitidos em cada uma das subportadoras, sendo aplicável para sistemas IEEE 802.11a. O segundo método utiliza as propriedades de correlação da resposta impulsiva de um canal de múltiplos percursos, e também informação de treinamento que é transmitida em um subconjunto do total de subportadoras, sendo também aplicado em sistemas IEEE 802.11a.

#### 4.1.5.1.1 Estimação de canal utilizando dados de treinamento

Os símbolos de treinamento longo no preâmbulo 802.11a permitem uma estimativa simples e eficaz da resposta em frequência do canal para todas as subportadoras. O conteúdo dos dois símbolos de treinamento longo é idêntico, logo, tomar-se a média de ambos pode melhorar a qualidade da estimativa do canal. A DFT é uma operação linear, o que permite tomar-se a média antes de calculá-la. Desta forma, apenas uma operação de DFT é necessária para calcular a estimativa do canal. Após o processamento da DFT, os símbolos de treinamento recebidos  $R_{1,k}$  e  $R_{2,k}$  são o produto dos símbolos de treinamento  $X_k$  e do canal  $H_k$  somado ao ruído gaussiano aditivo  $W_{l,k}$

$$R_{l,k} = H_k \cdot X_k + W_{l,k}. \quad (4.15)$$

Logo, a estimativa do canal pode ser calculada como:

$$\hat{H}_k = \frac{1}{2} \cdot (R_{1,k} + R_{2,k}) \cdot X_k^*, \quad (4.16)$$

$$= \frac{1}{2} (H_k \cdot X_k + W_{1,k} + H_k \cdot X_k + W_{2,k}) \cdot X_k^*, \quad (4.17)$$

$$= H_k \cdot |X_k|^2 + \frac{1}{2} (W_{1,k} + W_{2,k}) \cdot X_k^*, \quad (4.18)$$

$$= H_k + \frac{1}{2}(W_{1,k} + W_{2,k}) \cdot X_k^* .$$

(4.19)

onde as amplitudes dos dados de treinamento foram selecionadas para serem iguais à unidade. As amostras de ruído  $W_{1,k}$  e  $W_{2,k}$  são estatisticamente independentes, portanto a variância de sua soma dividida por dois é igual à metade da variância das amostras individuais de ruído.

#### 4.1.5.1.2 Estimação de canal utilizando as subportadoras piloto

As subportadoras piloto contêm informação conhecida no receptor, portanto podem ser utilizadas para a estimação da resposta em frequência do canal. De acordo com [NAN 2009] e [ATH 2003], após a execução da Transformada de Fourier no sinal recebido e da separação das componentes em frequência em componentes de dados e componentes piloto, temos que  $L_{p,n}$ , o símbolo piloto recebido correspondente ao  $n$ -ésimo símbolo OFDM, é o produto do símbolo piloto original  $P_{p,n}$  e da amostra da resposta em frequência do canal  $H_{p,n}$ , somado ao ruído gaussiano aditivo  $W_{p,n}$ :

$$L_{p,n} = H_{p,n} \cdot P_{p,n} + W_{p,n} .$$

(4.20)

Logo, a estimativa do canal pode ser calculada, para cada símbolo piloto, como:

$$\hat{H}_{p,s} = \frac{L_{p,n}}{P_{p,n}} .$$

(4.21)

Como se tem  $n$  símbolos OFDM em cada pacote, para cada uma das quatro subportadoras piloto  $p$ , estima-se a resposta em frequência do canal por meio da média simples:

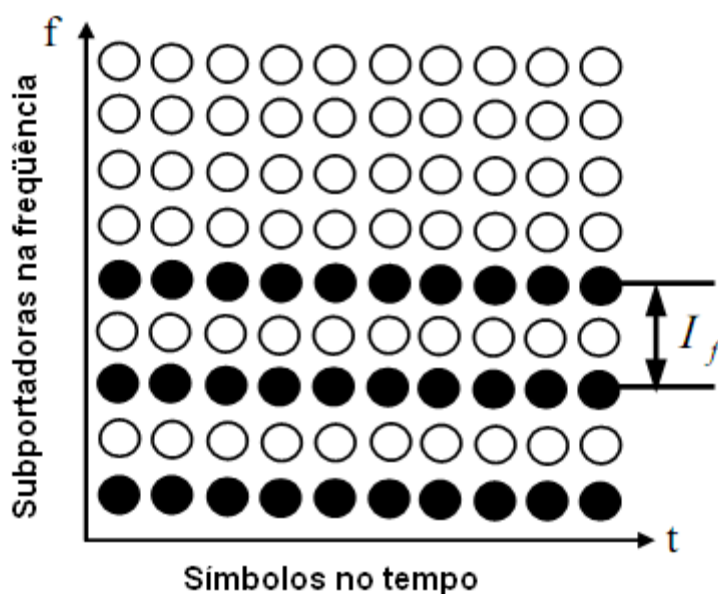
$$\hat{H}_{p,k} = \frac{1}{n} \cdot \sum_{i=1}^n \frac{L_{p,i}}{P_{p,i}} .$$

(4.22)

Em seguida, estima-se a resposta em frequência do pacote  $k$  para as demais 48 subportadoras (de dados) por meio de interpolação utilizando-se splines cúbicas. Esta interpolação



é tal que, para um dado conjunto  $X$  contendo  $n+1$  amostras, constroem-se  $n$  polinômios cúbicos (de grau 3), os quais gerarão as amostras intermediárias que serão interpoladas no conjunto original  $X$ .



**Figura 4.3. Localização das subportadoras piloto no tempo e na frequência.**

A figura 4.3 mostra como as subportadoras piloto (círculos negros) estão presentes em todos os símbolos no tempo, porém são transmitidas apenas em algumas frequências. É por este motivo que se utiliza a interpolação, para estimar a resposta em frequência do canal para as demais frequências, correspondentes às subportadoras de dados (círculos brancos).

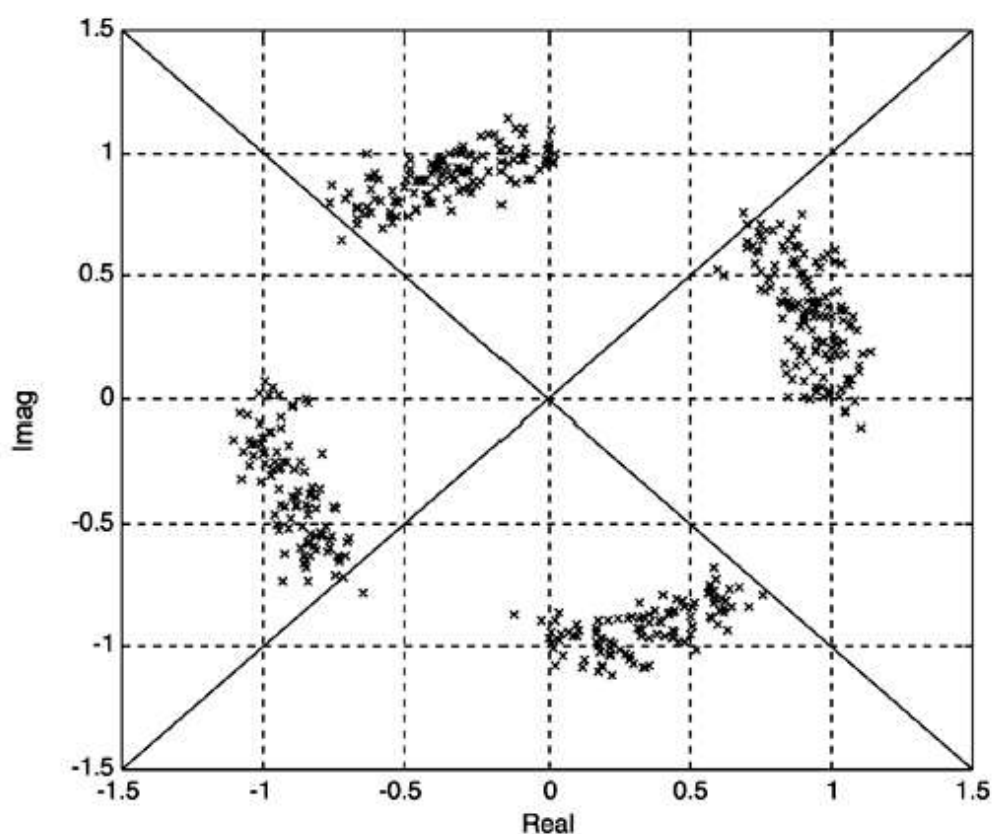
#### **Arquivos correspondentes no Matlab:**

O código que realiza a estimativa do canal por meio dos símbolos de treinamento longo do preâmbulo do pacote 802.11a encontra-se no arquivo *rx\_estimate\_channel.m*.

#### **4.1.6 Detecção e correção de erros de fase - Rastreamento da fase da portadora**

A estimação de frequência não é um processo perfeito, logo há sempre algum erro de frequência residual. A perda de RSR devido ao ICI gerado não será um problema se o estimador for projetado para reduzir o erro de frequência para um valor abaixo do limite requerido para uma perda de desempenho desprezível, na modulação utilizada. O principal problema do erro de

frequência residual é a rotação causada na constelação. A rotação na constelação é a mesma para todas as subportadoras. A figura 4.4 mostra o quanto a constelação rotaciona para modulação QPSK, em um sistema IEEE 802.11a, durante 10 símbolos OFDM, para um erro de frequência de 3 kHz. A princípio, este erro corresponde a apenas 1% do espaçamento entre subportadoras, logo, o efeito na perda de SNR é desprezível. Entretanto, como podemos verificar na figura 4.4, os pontos da constelação sofreram uma rotação tal que ultrapassaram os limites de decisão (linhas sólidas), tornando a demodulação correta não mais possível.



**Figura 4.4. Rotação de frequência para um offset de 3kHz durante 10 símbolos. [TSE 2005]**

Este efeito força o receptor a rastrear a fase da portadora durante a recepção dos símbolos de dados.

O método mais simples de corrigir este problema é o rastreamento da fase da portadora baseado nos dados recebidos. As normas IEEE 802.11a e HiperLAN/2 incluem quatro subportadoras predefinidas entre os dados transmitidos. Essas subportadoras especiais são conhecidas como subportadoras piloto. O principal propósito dessas portadoras piloto é auxiliar o

receptor a rastrear a fase da portadora. Depois de a DFT processar o  $n$ -ésimo símbolo recebido, desconsiderando a influência do ruído AWGN, as subportadoras piloto  $R_{n,k}$  são iguais ao produto da resposta impulsiva do canal  $H_k$  e do símbolo piloto conhecido  $P_{n,k}$ , rotacionado devido ao erro de frequência residual.

$$R_{n,k} = H_k \cdot P_{n,k} \cdot e^{j \cdot 2\pi \cdot f \Delta} \quad (4.23)$$

Assumindo que uma estimativa da resposta impulsiva do canal  $\hat{H}_k$  está disponível, a estimativa de fase é

$$\hat{\Phi}_n = \angle \left[ \sum_{k=1}^{N_p} R_{n,k} \cdot (\hat{H}_k \cdot P_{n,k})^* \right], \quad (4.24)$$

$$= \angle \left[ \sum_{k=1}^{N_p} H_k \cdot P_{n,k} \cdot e^{j \cdot 2\pi \cdot f \Delta} \cdot (\hat{H}_k \cdot P_{n,k})^* \right]. \quad (4.25)$$

Se for assumido que a estimativa do canal é perfeitamente acurada, tem-se o estimador

$$\hat{\Phi}_n = \angle \left[ \sum_{k=1}^{N_p} |H_k|^2 \cdot |P_{n,k}|^2 \cdot e^{j \cdot 2\pi \cdot f \Delta} \right], \quad (4.26)$$

$$= \angle \left[ e^{j \cdot 2\pi \cdot f \Delta} \cdot \sum_{k=1}^{N_p} |H_k|^2 \right]. \quad (4.27)$$

Na prática, entretanto, a estimativa do canal não é perfeitamente acurada, portanto esta também contribui com ruído na estimação da fase da portadora.

#### **Arquivos correspondentes no Matlab:**

O código que rastreia a fase da portadora durante a recepção dos símbolos de dados encontra-se no arquivo *rx\_phase\_tracker.m*.

### **4.1.7 Processamento da diversidade de antenas**

O uso de múltiplas antenas no receptor, que é referido como diversidade no receptor, ainda é pouco explorado no padrão IEEE 802.11a. Em resumo, múltiplas cópias do sinal transmitido são recebidas, as quais podem ser eficientemente combinadas usando técnicas apropriadas de processamento de sinais. Na medida em que o número de antenas aumenta, a probabilidade de interrupção no sinal tende a zero, e o canal efetivamente se aproxima de um canal AWGN. As duas técnicas mais populares de diversidade no receptor são seleção (do inglês Selection) e máxima taxa de combinação (do inglês MRC - Maximal Ratio Combining).

#### 4.1.7.1 Seleção de diversidade

O modo mais simples de diversidade no receptor é a seleção de diversidade. Dado  $M$  antenas receptoras, a seleção de diversidade escolhe a antena com a maior RSR em cada intervalo de símbolo. Outra característica atrativa da seleção de diversidade é que ela não requer uma cadeia adicional de receptores RF. Em outras palavras, todas as antenas de recepção compartilham o mesmo sistema de recepção em RF, o que é um fator essencial para manter um baixo custo no terminal móvel (MT – Mobile Terminal).

A performance da seleção de diversidade pode ser facilmente computada assumindo que cada antena é sujeita a ruído gaussiano I.I.D. (Independent and Identically Distributed). Logo, selecionar a antena com maior RSR é equivalente a selecionar a antena receptora com maior potência instantânea. Além disso, assume-se que a confiabilidade média de cada um dos caminhos é igual. Portanto, a SNR média  $\bar{\gamma}$  entre as antenas receptoras é dada por:

$$\bar{\gamma} = \frac{\bar{\varepsilon}}{N_0} = E(h_l^2) \cdot \frac{\varepsilon}{N_0}, \quad l = 1, \dots, M. \quad (4.28)$$

Conseqüentemente, a SNR instantânea  $\gamma$  é dada por:

$$\gamma = h_l^2 \cdot \frac{\varepsilon}{c}, \quad l = 1, \dots, M.$$

(4.29)

A probabilidade de que a SNR para a  $l$ -ésima antena receptora seja menor que um limiar  $Z$  é dada por:

$$P_r(\gamma_l \leq z) = \int_0^z f_{\gamma_l} d\gamma_l .$$

(4.30)

Onde  $f_{\gamma_l}$  representa a função densidade de probabilidades (pdf) de  $\gamma_l$ , a qual se assume ser a mesma para todas as antenas. Com M antenas receptoras independentes, a probabilidade de que todas elas tenham uma SNR abaixo do limiar Z é dada por:

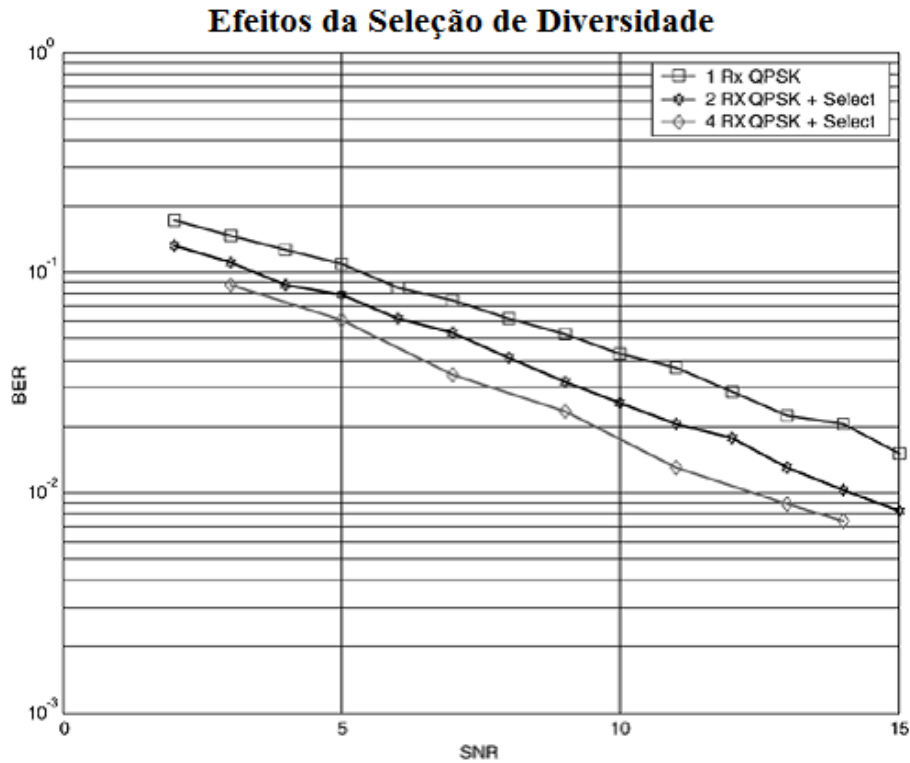
$$P_r(\gamma_l \leq z, \dots, \gamma_M \leq z) = [P_r(\gamma_l \leq Z)]^M .$$

(4.31)

$$\bar{\gamma} = \max\{\gamma_l, \dots, \gamma_M\} .$$

(4.32)

Portanto,  $\bar{\gamma}$  será menor que z se somente se  $\gamma_l, \dots, \gamma_M$  forem todos menores que z. A pdf é então obtida diretamente da derivada da CDF (Cumulative Distribution Function) em relação à z. Este resultado pode ser utilizado para obter a probabilidade de erro de um esquema de modulação digital na presença de seleção de diversidade, integrando a probabilidade de erro condicional com relação a  $\bar{\gamma}$ . Para verificar os benefícios da seleção de diversidade, considere-se um sinal modulado em QPSK em um ambiente de desvanecimento Rayleigh com 1, 2 e 4 antenas receptoras. A probabilidade de erro de bit em função da RSR é dada na figura 4.4.



**Figura 4.5. Performance de BER para seleção de diversidade em um canal com desvanecimento Rayleigh. [TSE 2005]**

Da figura 4.5, é evidente que a performance de taxa de erros melhora com o aumento do número de antenas receptoras. Pode-se observar que para uma BER de 1%, a melhoria na performance da taxa de erros é de 2 a 3 dB para duas antenas receptoras e 3 a 4 dB para quatro antenas. Ressalta-se que esta máxima melhoria na performance somente é obtida se cada caminho até as antenas receptoras for independente dos outros caminhos, conforme a equação (4.31).

#### 4.1.7.2 Combinação de Máxima razão

Na MRC, os sinais na saída das M antenas receptoras são combinados linearmente de forma a maximizar a SNR instantânea. Os coeficientes que proporcionam a máxima SNR são descobertos pela aplicação da teoria da otimização. Seja o sinal recebido na l-ésima antena dado por:

$$r_l = h_l \cdot s_l + v_l ,$$

$$(4.33)$$

com amostras de ruído independentes  $v_l$  e com a mesma densidade espectral de potência  $2.N_0$ . Assuma-se também perfeito conhecimento acerca do canal em cada antena e que o sinal transmitido  $r_l$  foi normalizado de tal forma que a energia média do sinal é igual a  $2.\varepsilon$ .

O MRC consiste em utilizar a combinação linear

$$y = \sum_{l=1}^M w_l^* \cdot r_l = \sum_{l=1}^M w_l^* \cdot h_l \cdot s_l + \sum_{l=1}^M w_l^* \cdot v_l \quad (4.34)$$

antes da detecção. A densidade espectral de potência  $S_v$  do ruído depois do MRC é dada por:

$$S_v = 2.N_0 \cdot \sum_{l=1}^M |w_l|^2 . \quad (4.35)$$

Enquanto que a energia instantânea do sinal é:

$$2.\varepsilon \cdot \left| \sum_{l=1}^M |w_l^* \cdot h_l|^2 \right| . \quad (4.36)$$

A razão dessas duas quantidades é a RSR na entrada do demodulador, dada por:

$$\gamma = \frac{\varepsilon \cdot \left| \sum_{l=1}^M |w_l^* \cdot h_l|^2 \right|}{N_0 \cdot \sum_{l=1}^M |w_l|^2} . \quad (4.37)$$

A expressão acima pode ser maximizada como segue. Lembrando a inequação de Cauchy-Schwartz:

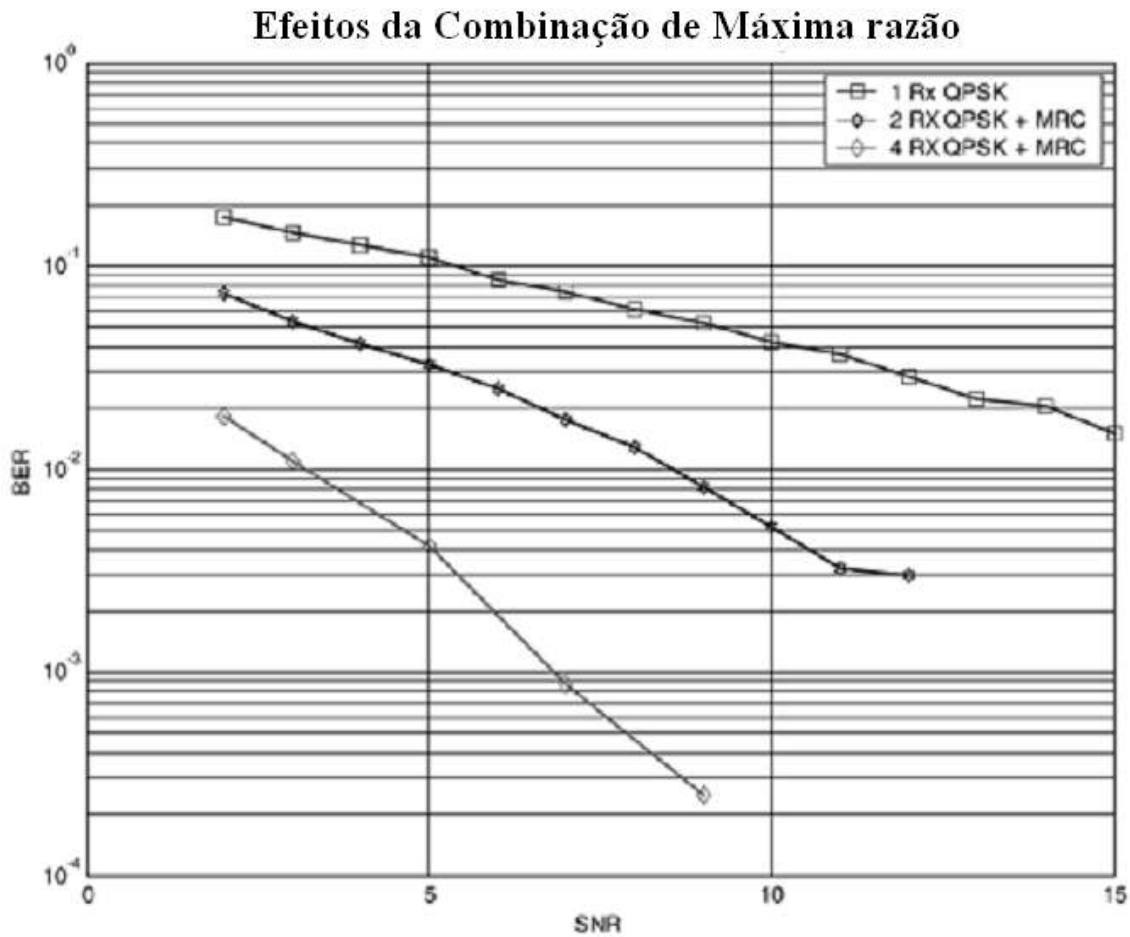
$$\left| \sum_{l=1}^M |a_l^* \cdot b_l|^2 \right| = \sum_{l=1}^M |a_l|^2 \cdot \sum_{l=1}^M |b_l|^2 , \quad (4.38)$$

onde a igualdade na equação (4.38) é obtida para  $w_l = h_l$  para todo  $l$ , o que proporciona os coeficientes ou pesos para a MRC. Ou seja, cada antena é ponderada proporcionalmente por sua atenuação por desvanecimento correspondente. As antenas mais afetadas pelo desvanecimento, as quais são menos confiáveis, possuem pesos menores que aquelas antenas pouco afetadas. A RSR alcançada pela MRC é dada por:

$$\gamma_{MRC} = \frac{\varepsilon}{N_0} \cdot \sum_{l=1}^M |w_l|^2.$$

(4.39)

Verificando que  $\frac{\varepsilon}{N_0} \cdot |w_l|^2$  é a RSR por antena, então a equação (4.39) é apenas a soma das RSRs para cada antena, o que significa que  $\gamma_{MRC}$  pode ser grande mesmo quando as RSRs individuais forem pequenas. Como um exemplo dos resultados da melhoria de performance com a MRC, considere-se o mesmo cenário anterior com a seleção de diversidade. Da figura 4.6, podemos aferir que, para uma BER de 1%, a MRC com duas antenas receptoras proporciona aproximadamente 10 dB de aumento, enquanto que para seleção de diversidade, tínhamos 2 a 3 dB. Para quatro antenas receptoras, o aumento chega a ser próximo a 15 dB.



**Figura 4.6. Performance de BER para MRC em um canal com desvanecimento Rayleigh. [TSE 2005]**



Em suma, o potencial de ganho utilizando MRC pode ser enorme, o que não é realmente uma surpresa, visto que é a solução ótima para este caso. Deve-se lembrar, porém, que se assumiu estimação perfeita do canal, o que não é verdade, na maioria das vezes. A acurácia da estimação de canal depende de certa forma na RSR de operação.

#### **Arquivos correspondentes no Matlab:**

O código que executa o algoritmo de MRC no sinal recebido encontra-se no arquivo *rx\_diversity\_proc.m*, enquanto que o código que reverte o processo de Radon-Hurwitz executado no transmissor encontra-se no arquivo *rx\_radon\_hurwitz.m*.

### **4.1.8 Demodulação**

A demodulação trata de decidir para cada símbolo recebido quais os bits mais prováveis que foram transmitidos. As decisões são divididas em decisões hard e soft, dependendo de quanta informação acerca de cada bit transmitido é produzida.

#### **4.1.8.1 Detecção com decisão Hard**

Um demodulador de decisão hard determina se um bit zero ou bit um foi transmitido, logo a saída deste demodulador é composta de zeros ou uns. Um demodulador de decisão hard é também definido com base no número de possíveis entradas no modulador do transmissor. Se este número é igual ao número de possíveis saídas do demodulador no receptor, o demodulador utiliza decisão hard.

Se o demodulador pode gerar mais informação do que as entradas do modulador, o sistema utiliza decisão soft, que será discutida mais à frente. Temos na figura 4.7 os limites de decisão para a constelação da modulação QPSK. Os limites de decisão determinam a maneira pela qual os símbolos recebidos são mapeados para bits. Essencialmente, a decisão de máxima verossimilhança é tomada como o ponto da constelação que está mais próximo do símbolo recebido, e as decisões hard tomam os bits correspondentes àquele ponto da constelação como sendo os bits transmitidos. Como exemplo, na figura 4.7, o ponto x corresponde ao símbolo recebido; ele está mais próximo do

ponto da constelação no eixo imaginário positivo, o que faz com que os bits por decisão hard sejam 00.

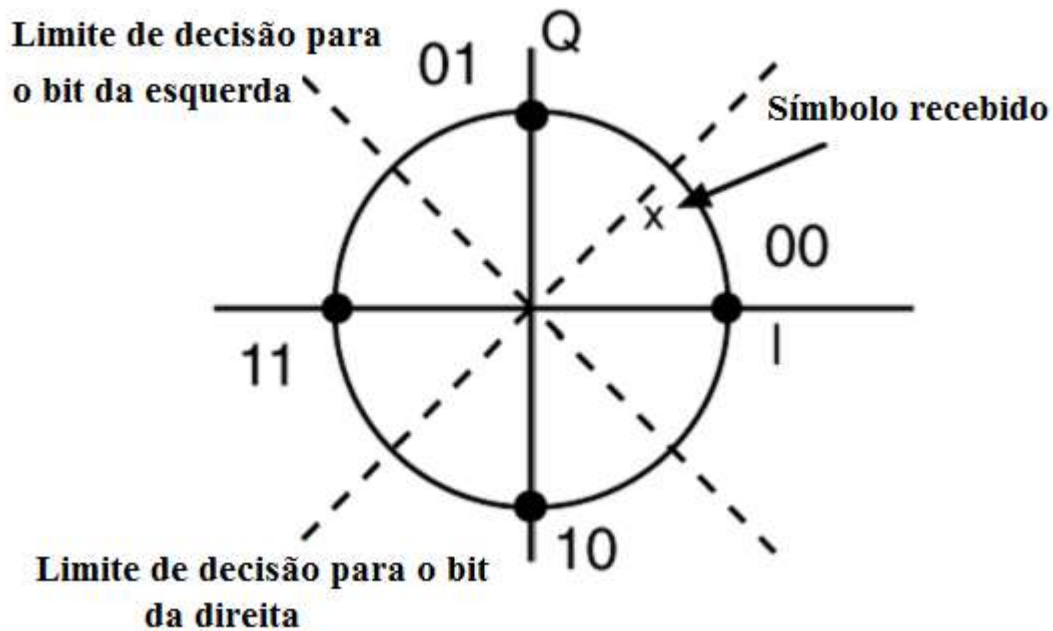


Figura 4.7. Limites de decisão Hard para a constelação QPSK.

### Detecção com decisão soft

Um demodulador com decisão soft gera bits “soft” de saída, que além de indicar se o bit é um zero ou um “um”, contém informação acerca da confiabilidade da decisão. Esta informação adicional pode aumentar consideravelmente a performance da codificação de canal. Para executar as decisões soft, o demodulador necessita considerar os bits recebidos individualmente. Vejamos novamente o símbolo recebido na figura 4.6. Ele está localizado muito perto do limite de decisão entre os símbolos correspondentes aos bits 00 e 01. Para o bit da esquerda mudar seu valor para 1, o símbolo teria que se mover para além do limite de decisão para este bit. O símbolo recebido está longe deste limite, logo o bit da esquerda é relativamente confiável. Por outro lado, o bit da direita pode mudar de 0 para 1, bastando para isso que o símbolo recebido estivesse acima do limite de decisão para o bit da direita, ao invés de estar logo abaixo deste limite. Para esta mudança, bastaria apenas um pequeno aumento no ruído, o que torna este bit pouco confiável. De maneira a refletir esta diferença de confiabilidade, os bits soft possuem diferentes valores: um grande valor absoluto

para o primeiro bit e um pequeno valor para o segundo bit. O sinal da decisão soft indica um bit 0 ou 1. O valor absoluto de cada decisão soft é a distância até o limite de decisão.

O simulador 802.11a analisado implementa apenas a detecção soft.

#### Arquivos correspondentes no Matlab:

O código que busca o tipo de modulação utilizada no transmissor encontra-se no arquivo `rx_demodulate.m`, enquanto que os códigos que realizam a demodulação para as modulações BPSK, QPSK, 16-QAM e 64-QAM, encontram-se, respectivamente, nos arquivos `rx_bpsk_demod.m`, `rx_qpsk_demod.m`, `rx_qam16_demod.m` e `rx_qam64_demod.m`.

#### 4.1.8.2 Deinterleaving

O deinterleaver desempenha a função inversa do interleaver, e, como este último, é também definido por duas permutações.

Denotar-se-á  $j$  como o índice do bit original recebido, antes da primeira permutação;  $i$  denota o índice depois da primeira e antes da segunda permutação, e  $k$  denota o índice depois da segunda permutação, imediatamente antes dos bits codificados serem enviados para o decodificador convolucional de Viterbi.

A primeira permutação é definida pela seguinte regra:

$$i = s \cdot \text{floor} \left( \frac{j}{s} \right) + \left( j + \text{floor} \left( 16 \cdot \frac{j}{N_{CBPS}} \right) \right) \text{mod } s \quad j = 0, 1, \dots, N_{CBPS} - 1 .$$

(4.40)

O valor de  $s$  é determinado pelo número de bits codificados por subportadora,  $N_{BPSC}$ , de acordo com:

$$s = \max \left( \frac{N_{BPSC}}{2}, 1 \right) .$$

(4.41)

A função `floor(.)` retorna o maior inteiro que não exceda o parâmetro dentro dos parênteses.

Esta permutação é o processo inverso da permutação descrita na equação 2.3.

A segunda permutação é definida por:

$$k = 16 \cdot i - (N_{CBPS} - 1) \cdot \text{floor} \left( 16 \cdot \frac{i}{N_{CBPS}} \right) \quad i = 0, 1, \dots, N_{CBPS} - 1 . \quad (4.42)$$

Esta permutação é o processo inverso da permutação descrita na equação 2.2.

#### **Arquivos correspondentes no Matlab:**

O código que desempenha a função inversa do interleaver encontra-se no arquivo *rx\_deinterleave.m*.

#### **4.1.10 Reversão do apagamento de bits**

Para que os bits codificados e processados pelo bloco de apagamento no transmissor possam ser decodificados adequadamente, os bits que foram removidos devem ser inseridos de volta na seqüência de bits, pois o decodificador do código original de taxa  $\frac{1}{2}$  é utilizado na decodificação. A reversão do apagamento é realizada simplesmente inserindo bits “dummy” nos locais em que os bits foram apagados no transmissor. Os valores dos bits “dummy” dependem se o sistema utiliza decisões hard ou soft. Para o caso de decisão hard, será inserido um bit aleatório (zero ou um) nos locais de apagamento. Para o caso de decisão soft, o receptor insere uma decisão soft de valor zero. Para o caso de decodificação utilizando o algoritmo de Viterbi, o bit “dummy” com valor zero não afeta a saída do decodificador (não altera a métrica do caminho).

#### **Arquivos correspondentes no Matlab:**

O código que realiza a reversão do apagamento de bits executado no transmissor encontra-se no arquivo *rx\_depuncture.m*.

#### **4.1.11 Decodificação de Viterbi**

Existem diversos algoritmos que podem ser utilizados para decodificar códigos convolucionais. Estes algoritmos variam em complexidade e performance, sendo que a regra é que maior complexidade implica em melhor performance. Nos últimos anos, o algoritmo de Viterbi alcançou uma posição dominante como o método de decodificação de códigos convolucionais, especialmente em aplicações sem fio; os outros métodos praticamente não são mais utilizados. A razão para tal é que o algoritmo de Viterbi é um estimador de palavras-código de máxima verossimilhança, fornecendo a melhor estimativa possível para a palavra código transmitida.

O algoritmo de Viterbi pode ser facilmente implementado utilizando tanto demodulação hard quanto soft. As figuras 4.8 e 4.9 apresentam o aumento na performance na BER e na PER ganho ao utilizar decisões soft. O ganho é de aproximadamente 2 dB para BER de  $10^{-3}$  e PER de  $10^{-1}$ . O ganho aumenta ligeiramente para SNRs maiores. Portanto, decodificação com complexidade soft é o método recomendado para se utilizar com a decodificação de Viterbi, visto que o aumento na performance alcançado não requer recursos de comunicação adicionais .

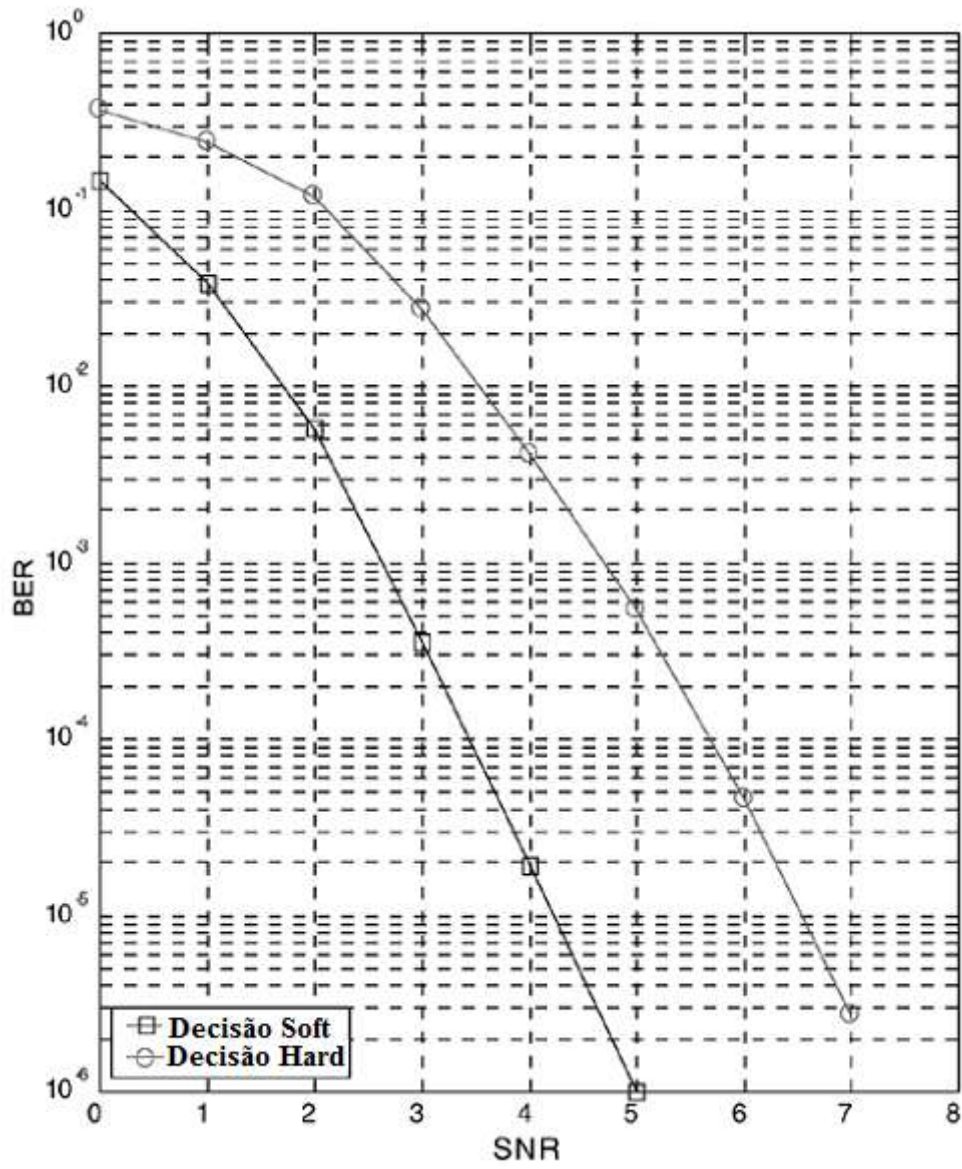
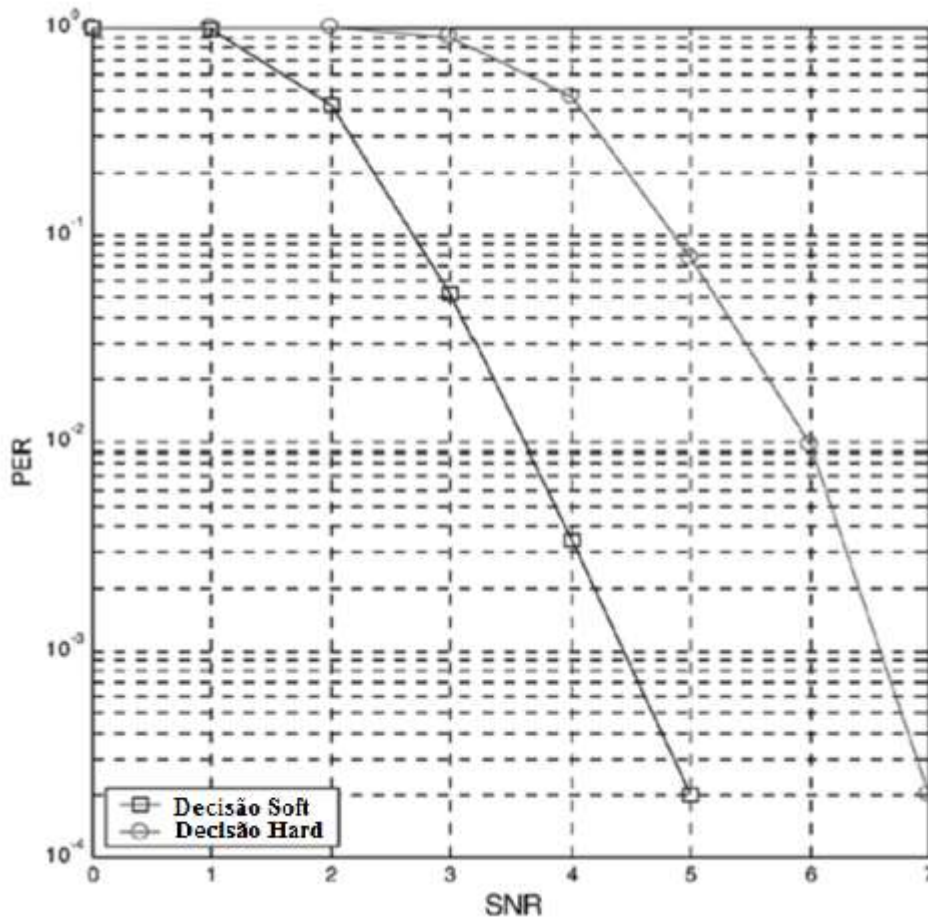


Figura 4.8. Performance de BER no IEEE 802.11a no modo 12 Mbits/s em um canal AWGN utilizando-se decisões de decodificação soft e hard. [TSE 2005]



**Figura 4.9. Performance de PER no IEEE 802.11a no modo 12 Mbits/s em um canal AWGN utilizando-se decisões de decodificação soft e hard. [TSE 2005]**

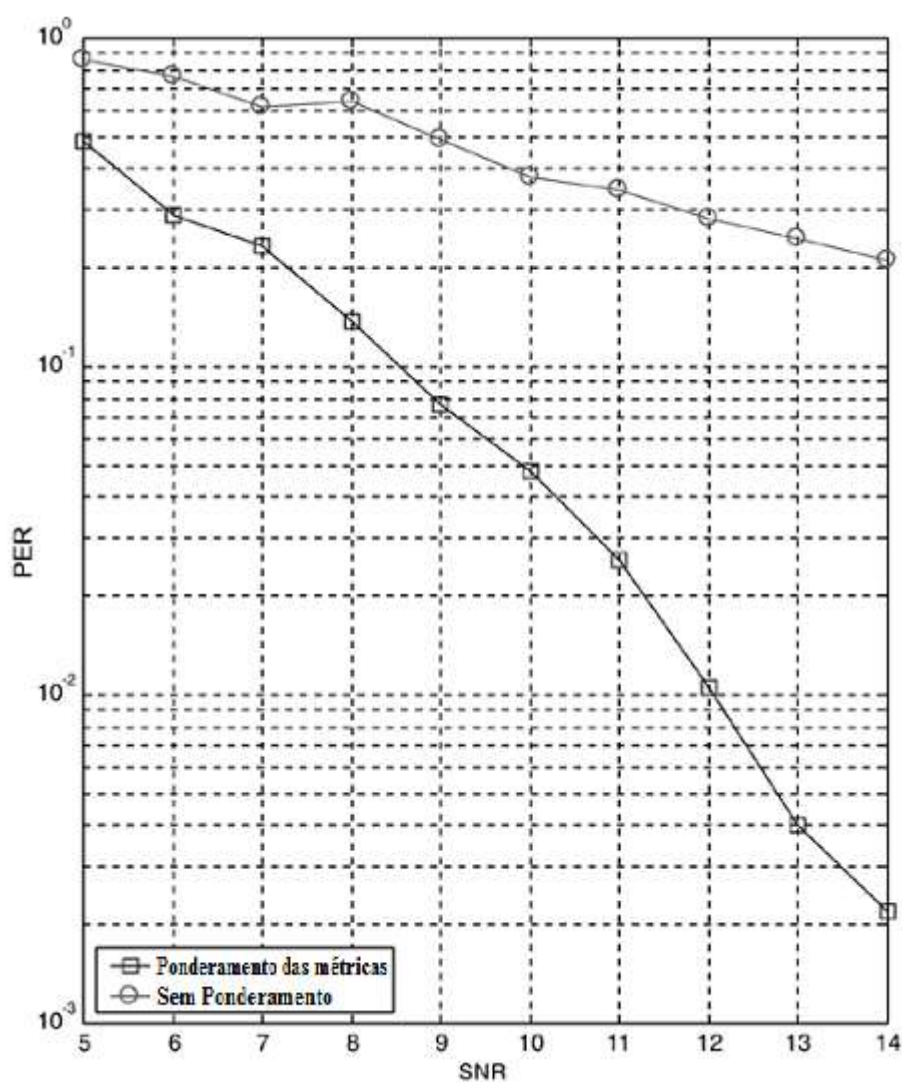
Um sistema OFDM coerente precisa estimar a resposta em frequência do canal. Informação relativa às amplitudes das diversas subportadoras pode ser incorporada ao algoritmo de Viterbi para fornecer aumento na performance. A equação 4.43 mostra como as métricas  $p_n$  de cada caminho no algoritmo de Viterbi podem ser calculadas ponderando a distância Euclidiana quadrática entre a decisão soft  $\hat{b}_n$  e o valor de referência  $b_n$  pela amplitude quadrática da subportadora  $k$  na qual o bit foi transmitido.

$$p_n = |H_k|^2 \cdot |\hat{b}_n - b_n|^2.$$

(4.43)

O efeito na performance com esta ponderação é significativo em canais com desvanecimento. A figura 4.10 mostra a PER para o IEEE 802.11a com 12 Mbits/s quando a

decodificação do código convolucional é realizada utilizando ponderação no decodificador de Viterbi ou pesos iguais para todas as métricas dos caminhos. O impacto da ponderação na performance é muito grande. A razão é a diversidade; quando a subportadora sofre o efeito do desvanecimento, sua amplitude é pequena e a métrica do caminho é ponderada para quase zero. Isto significa que os bits que foram transmitidos em subportadoras comprometidas pelo desvanecimento tem pouco impacto nas decisões feitas pelo decodificador, o que implica em melhoria na performance.



**Figura 4.10. Efeito sobre a PER com ponderamento de métricas na decodificação de Viterbi, para a IEEE 802.11a no modo 12 Mbits/s em um canal de desvanecimento Rayleigh com atraso de espalhamento de 75ns rms. [TSE 2005]**



**Arquivos correspondentes no Matlab:**

O código que busca a amplitude das subportadoras, de modo a obter os pesos requeridos para as decisões soft feitas antes da decodificação de Viterbi, encontra-se no arquivo *rx\_gen\_chan\_amps.m*. O código que realiza as decisões soft tomadas utilizando-se os pesos obtidos da amplitude das subportadoras encontra-se no arquivo *channel\_amps.m*. O código que realiza a decodificação através do algoritmo de Viterbi encontra-se no arquivo *rx\_viterbi\_decode.m*.

## 5 SIMULAÇÕES: RESULTADOS E ANÁLISES

A partir das modificações realizadas no código do simulador IEEE 802.11a, foi possível realizar uma série de simulações, alterando determinados parâmetros de modo a testar e comparar a efetividade dos algoritmos implementados nos diversos blocos componentes do simulador. Podem-se dividir estas simulações em duas partes:

1. Análise das performances de erro de bit e de pacote em função da RSR de bit  $\frac{E_b}{N_o}$ ;
2. Análise visual do efeito do canal de rádio-móvel e das não linearidades introduzidas pelo transmissor e receptor, sobre o conteúdo binário de uma imagem.

A figura 5.1 mostra a interface gráfica do simulador, através da qual podem ser modificados os parâmetros desejados. Todas as simulações foram realizadas com pacotes de tamanho igual a 100 bytes. Este tamanho foi escolhido de maneira a manter o tempo de simulação dentro de limites práticos aceitáveis.

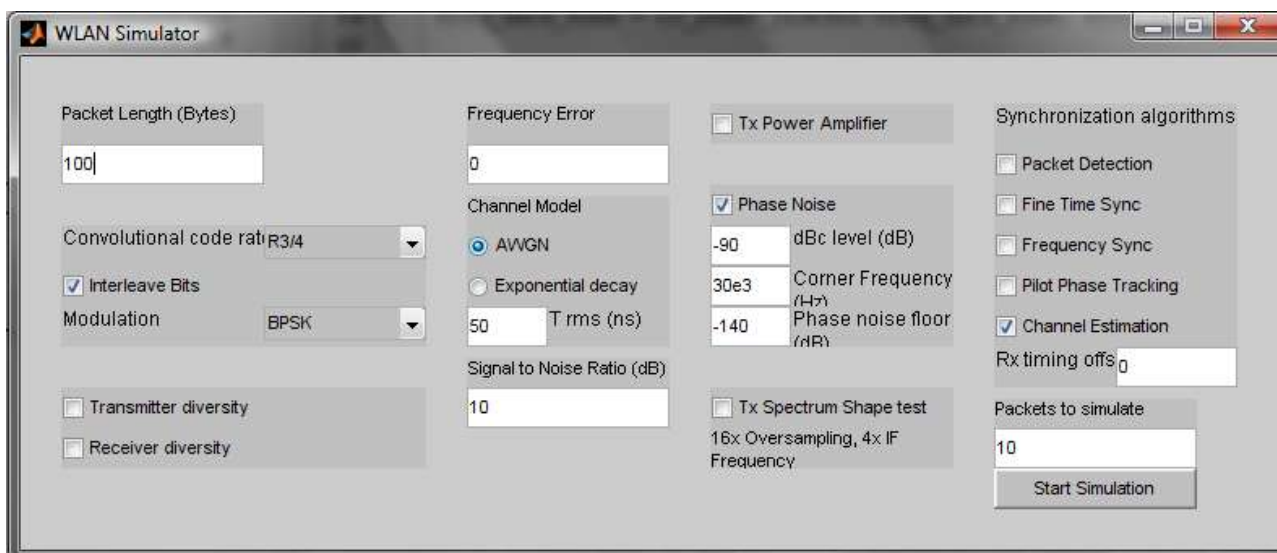


Figura 5.1. Interface gráfica do simulador 802.11a.

Opções de interação com o usuário por meio da interface gráfica do simulador:

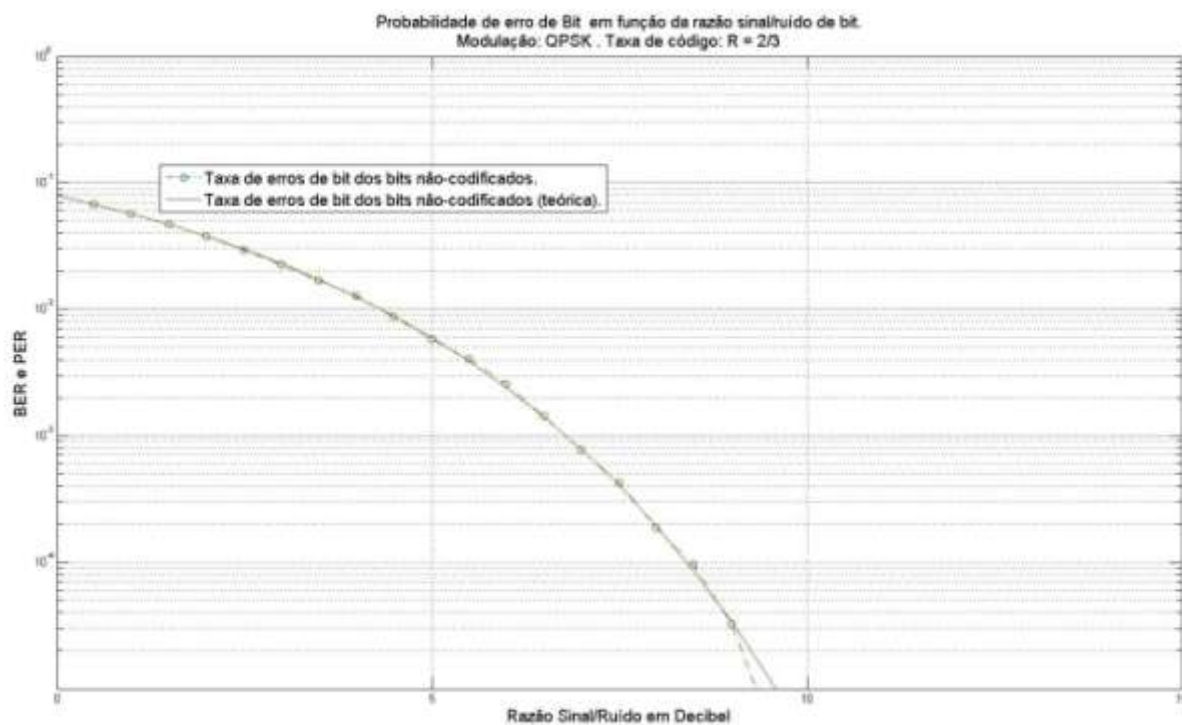
- Tamanho do pacote;
- Taxa de código convolucional (1/2, 2/3 ou 3/4);

- Modulação (BPSK, QPSK, 16-QAM ou 64-QAM);
- Diversidade de antenas (receptor e transmissor);
- Tipo de canal (AWGN ou desvanecimento exponencial);
- Erro de frequência introduzido pelo canal;
- Razão sinal para ruído;
- Período do decaimento exponencial;
- Modelagem de amplificador de potência;
- Modelagem de ruído de fase;
- Teste do espectro do sinal transmitido;
- Número de pacotes transmitidos;
- Interleaving;
- Algoritmos de sincronização no receptor (estimação de canal baseado nos símbolos de treinamento, detecção de pacotes, sincronizações de frequência e de símbolo e rastreamento da fase da portadora.);
- Offset do algoritmo de detecção de pacotes.

## **5.1 Análise das performances de erro de bit e de pacote**

### **5.1.1 Validação do simulador 802.11a**

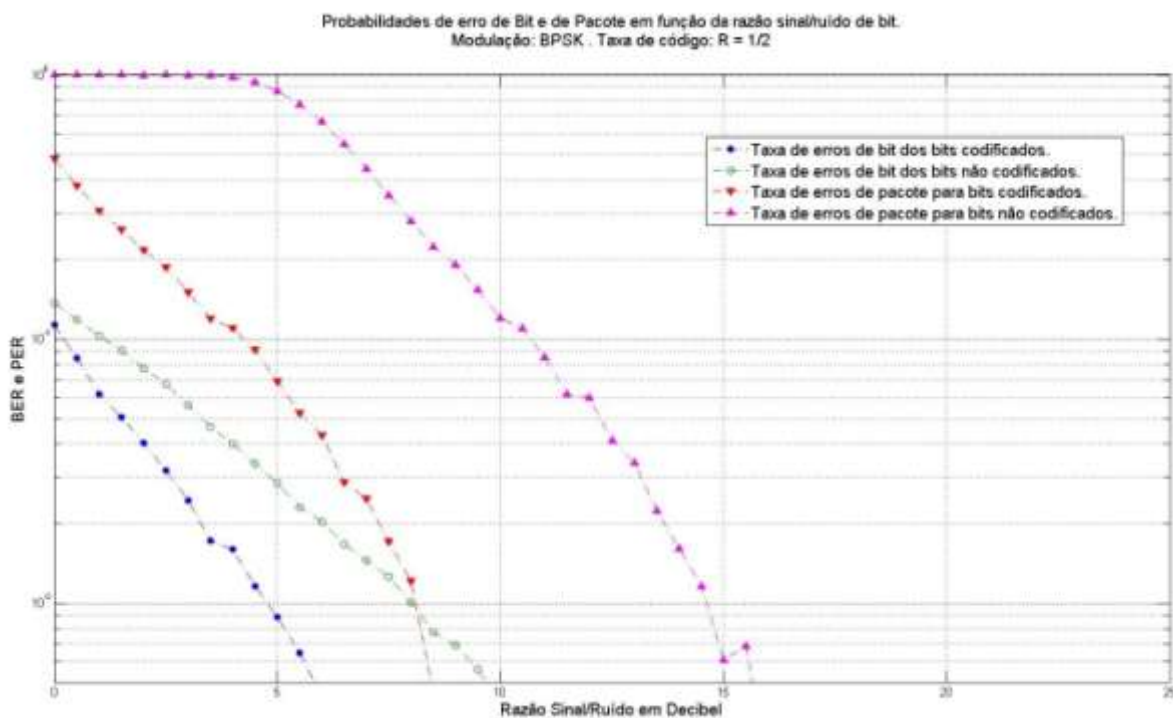
A figura 5.2 mostra a validação do simulador, na medida em que os valores de taxa de erros de bit obtidos por meio da simulação de Monte Carlo (linha com círculos) correspondem adequadamente aos valores previstos teoricamente (linha cheia). Utilizou-se modulação QPSK e taxa de código 2/3.



**Figura 5.2.** Performances de BER teórica e simulada para fins de validação do simulador.

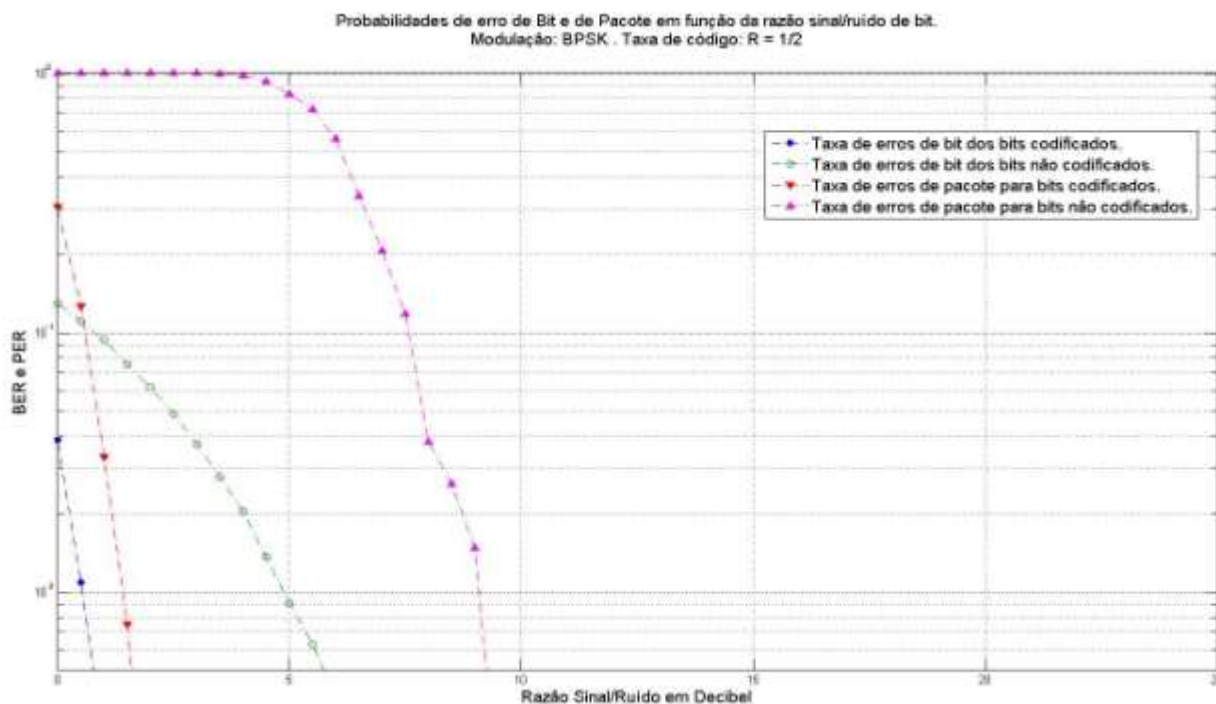
### 5.1.1 Canal AWGN

A figura 5.3 mostra a performance de BER e PER, utilizando algoritmo de estimação de canal baseado nas subportadoras piloto, com modulação 16-QAM e taxa de código  $\frac{3}{4}$ .



**Figura 5.3. Performance de BER e PER, utilizando algoritmo de estimação de canal baseado nas subportadoras piloto.**

A figura 5.2 mostra a performance de BER e PER, utilizando algoritmo de estimação de canal baseado nos símbolos de treinamento, com modulação 16-QAM e taxa de código  $\frac{3}{4}$ .



**Figura 5.4. Performance de BER e PER, utilizando algoritmo de estimação de canal baseado nos símbolos de treinamento.**

Pode-se inferir, da inspeção nas figuras 5.3 e 5.4, que a performance de erros do algoritmo de estimação de canal baseado nos símbolos de treinamento supera grandemente o algoritmo de estimação de canal baseado nas subportadoras piloto, chegando a um ganho de 4 dB com relação a este último, para taxa de erro de bits de  $10^{-2}$  e codificação convolucional. Por esta razão, este algoritmo foi escolhido para permanecer como o responsável pela estimação de canal do simulador.

### 5.1.2 Canal com decaimento exponencial e inserção de ruído de fase

A figura 5.5 mostra a performance de erro de bits e de pacotes em função da RSR de bit, em um canal do tipo decaimento exponencial na presença de ruído de fase, com modulação 16-QAM, taxa de código  $\frac{3}{4}$ , sem algoritmos de melhoria de performance. Verifica-se que a inserção de ruído de fase, somada ao canal do tipo decaimento exponencial, impossibilita a detecção correta de pacotes, corrompendo totalmente o sinal transmitido.

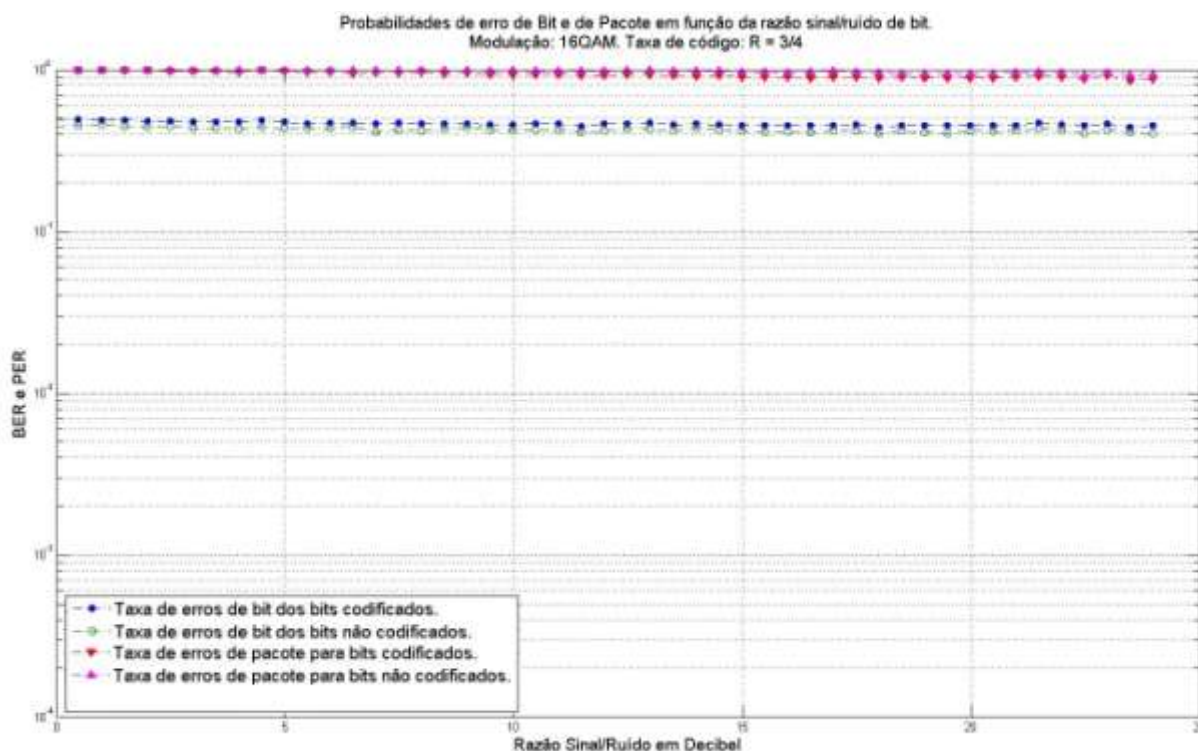
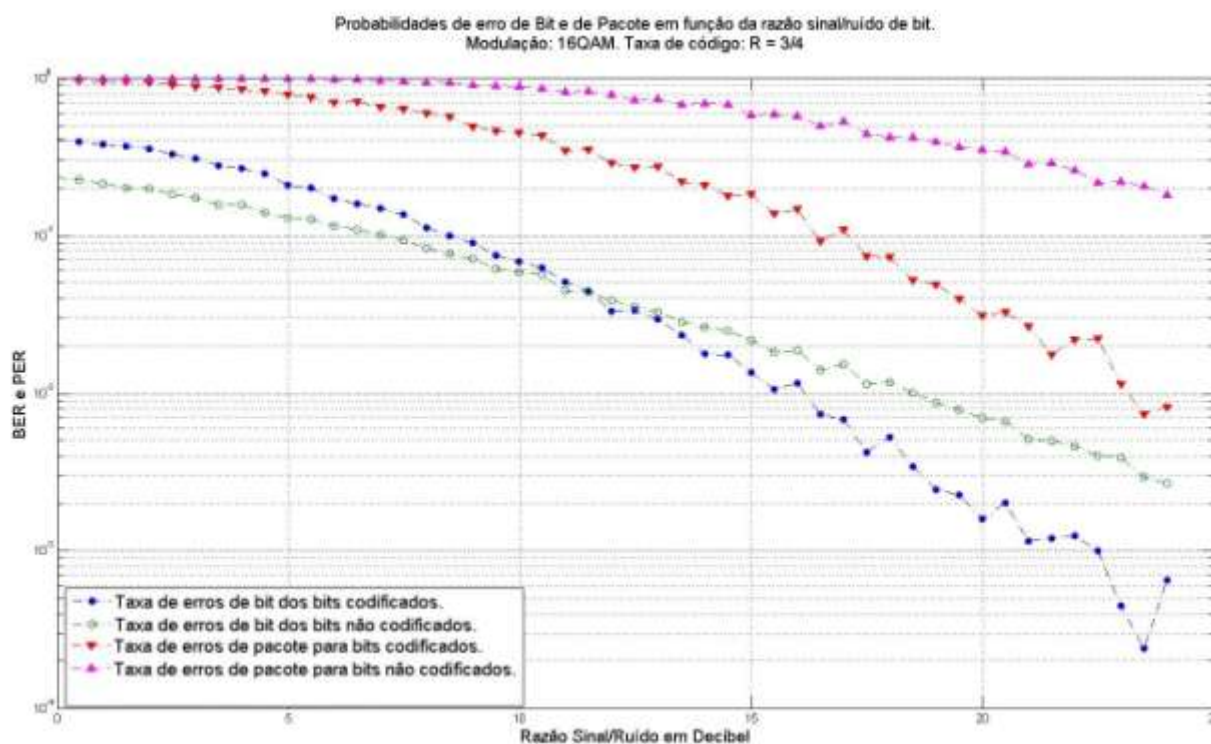


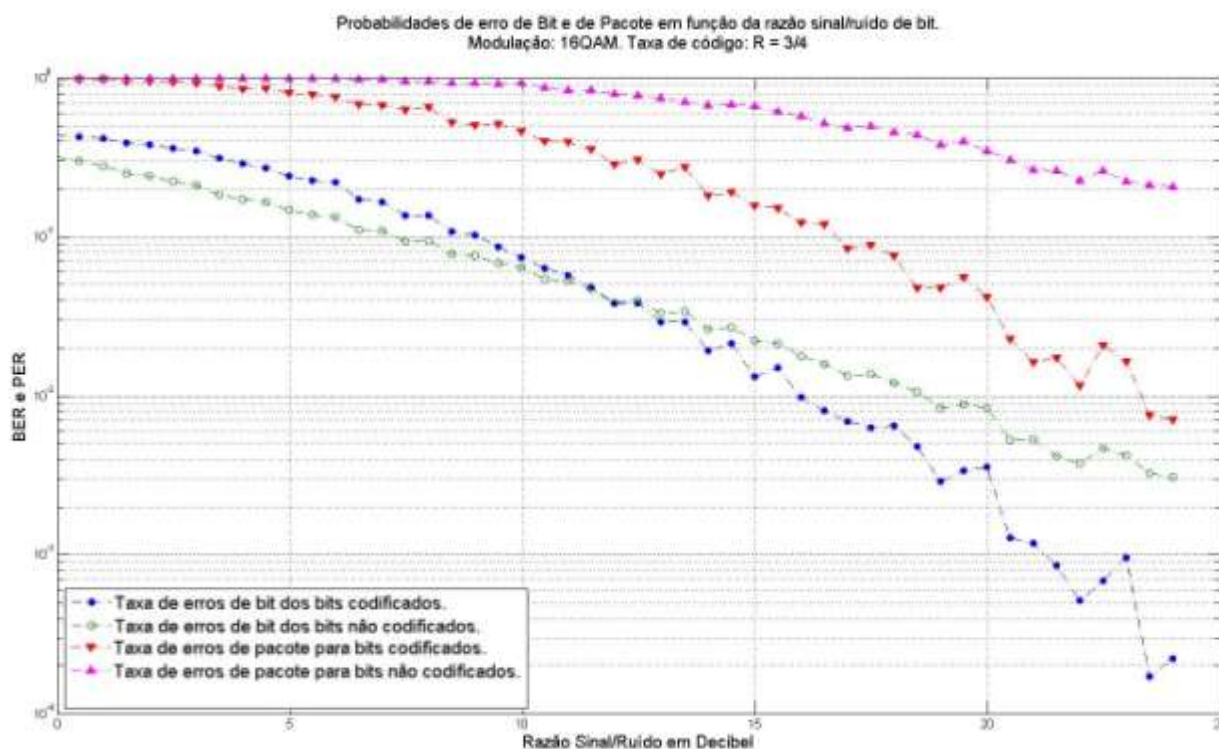
Figura 5.5. Performance de BER e PER sem algoritmos de sincronização.

A figura 5.6 mostra a performance de erro de bits e de pacotes em função da RSR de bit, em um canal do tipo decaimento exponencial na presença de ruído de fase, com modulação 16-QAM, taxa de código  $\frac{3}{4}$ , com algoritmo de estimação de canal baseado nos símbolos de treinamento. Pode-se concluir que apenas a utilização do algoritmo de estimação de canal é capaz de recuperar a degradação ocasionada na performance, tornando novamente possível a demodulação e detecção de pacotes.



**Figura 5.6. Performance de BER e PER, com algoritmo de estimação de canal.**

A figura 5.7 mostra a performance de erro de bits e de pacotes em função da RSR de bit, em um canal do tipo decaimento exponencial na presença de ruído de fase, com modulação 16-QAM, taxa de código  $\frac{3}{4}$ , utilizando algoritmos de estimação de canal baseado nos símbolos de treinamento, algoritmos de detecção de pacotes, algoritmos de sincronizações de frequência e de símbolo e rastreamento da fase da portadora.

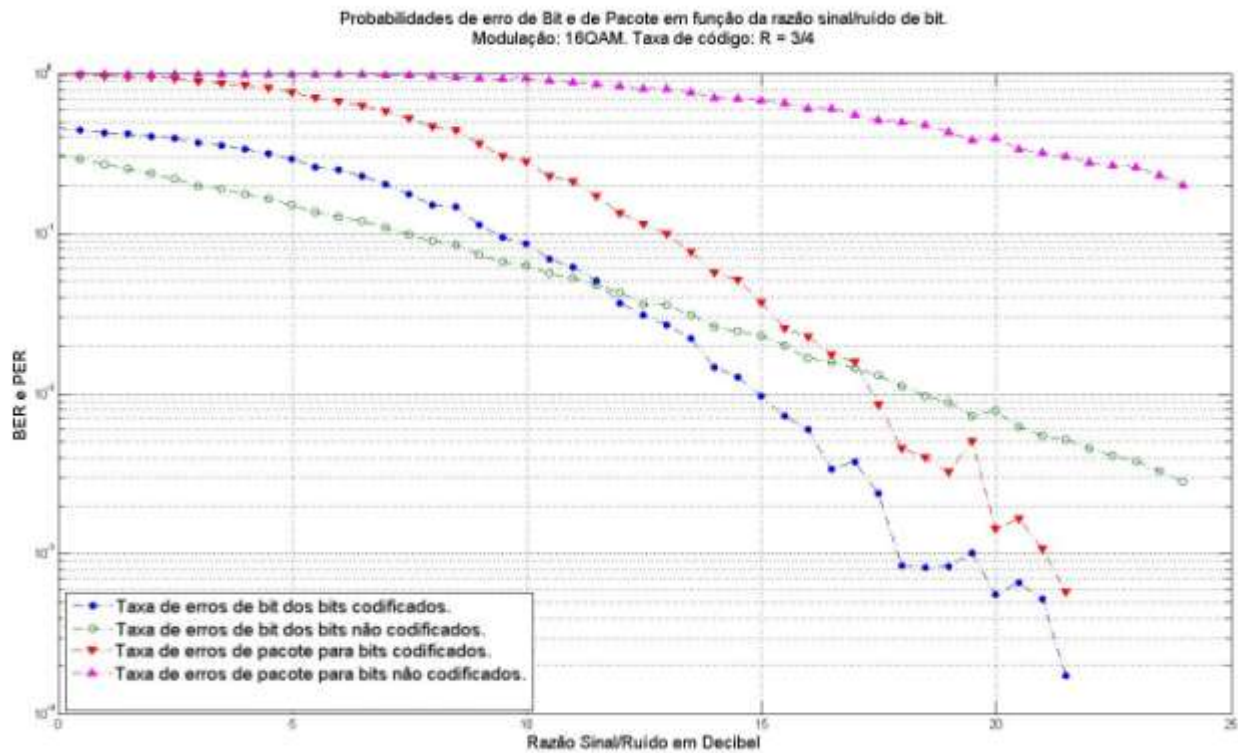


**Figura 5.7 Performance de BER e PER, utilizando 5 algoritmos de melhoria de performance.**

Pode-se inferir, da inspeção nas figuras 5.6 e 5.7, que a utilização dos algoritmos citados acima, combinados (figura 5.7), representa um ganho pouco significativo, quando comparados com a utilização apenas do algoritmo de estimação de canal (figura 5.6), utilizado para a decodificação *soft de Viterbi*, em um canal de decaimento exponencial.

A figura 5.8 mostra a performance de erro de bits e de pacotes em função da RSR de bit, em um canal do tipo decaimento exponencial na presença de ruído de fase, com modulação 16-QAM, taxa de código  $\frac{3}{4}$ , utilizando algoritmos de estimação de canal baseado nos símbolos de treinamento, algoritmos de detecção de pacotes, algoritmos de sincronizações de frequência e de símbolo, rastreamento da fase da portadora e interleaving.

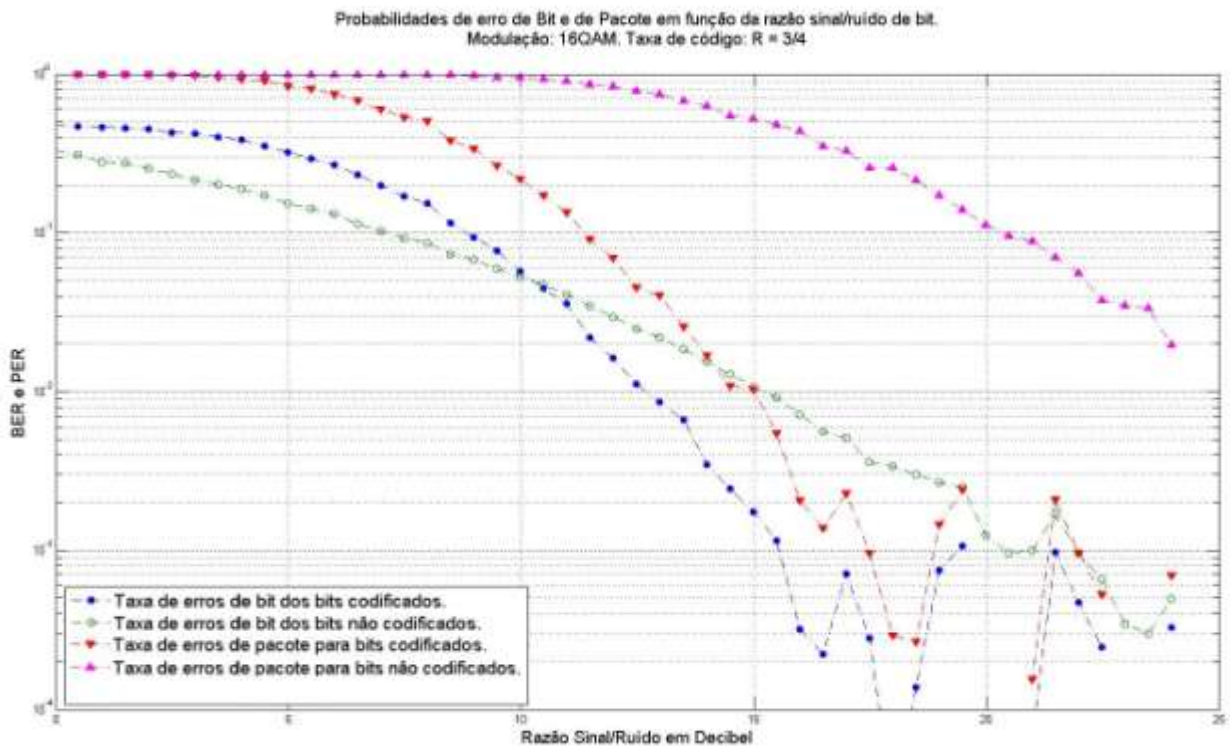




**Figura 5.8. Performance de BER e PER, utilizando 6 algoritmos de melhoria de performance.**

Percebe-se um ganho de código, para uma probabilidade erro de bit de  $10^{-2}$ , de 1.5 dB com o acréscimo do algoritmo de interleaving com relação à estimação de canal (figura 5.6), para codificação convolucional.

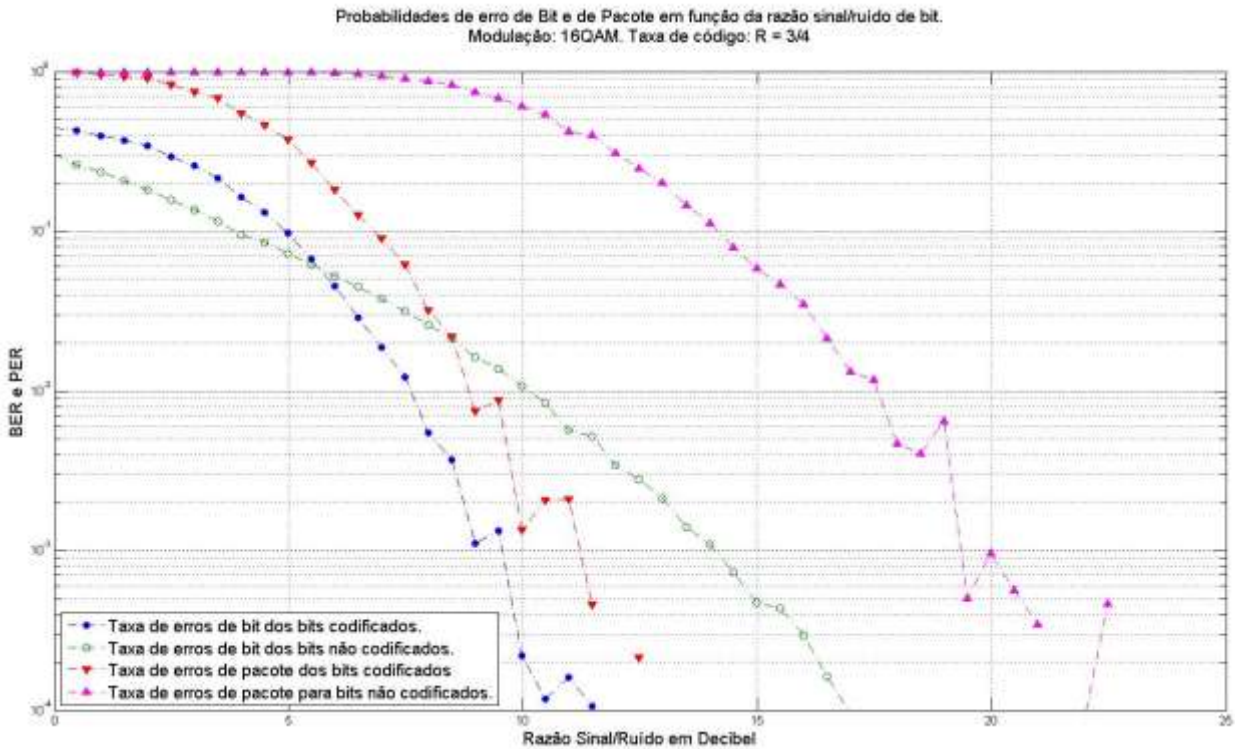
A figura 5.9 mostra a performance de erro de bits e de pacotes em função da RSR de bit, em um canal do tipo decaimento exponencial na presença de ruído de fase, com modulação 16-QAM, taxa de código  $\frac{3}{4}$ , utilizando algoritmos de estimação de canal baseado nos símbolos de treinamento, algoritmos de detecção de pacotes, algoritmos de sincronizações de frequência e de símbolo, rastreamento da fase da portadora, interleaving e diversidade de antenas no transmissor.



**Figura 5.9. Performance de BER e PER, utilizando 7 algoritmos de melhoria de performance.**

Percebe-se um ganho de código, para uma probabilidade erro de bit de  $10^{-2}$ , de 2.5 dB com o acréscimo do algoritmo de diversidade de antenas no transmissor (figura 5.10), com relação à figura 5.9, para codificação convolucional.

A figura 5.10 mostra a performance de erro de bits e de pacotes em função da RSR de bit, em um canal do tipo decaimento exponencial na presença de ruído de fase, com modulação 16-QAM, taxa de código  $\frac{3}{4}$ , utilizando algoritmos de estimação de canal baseado nos símbolos de treinamento, algoritmos de detecção de pacotes, algoritmos de sincronizações de frequência e de símbolo, rastreamento da fase da portadora, interleaving e diversidade de antenas no transmissor e no receptor.



**Figura 5.10. Performance de BER e PER, utilizando 8 algoritmos de melhoria de performance.**

Percebe-se um ganho de código, para uma probabilidade erro de bit de  $10^{-2}$ , de 5 dB, com o acréscimo do algoritmo de diversidade de antenas no receptor (figura 5.11), com relação à figura 5.10, para codificação convolucional.

## 5.2 Análise visual do efeito do canal de rádio-móvel AWGN e das não linearidades introduzidas pelo transmissor e receptor, sobre o conteúdo binário de uma imagem

A figura 5.11 escolhida para esta simulação foi uma fotografia de Lena, que é tradicionalmente escolhida para comparar algoritmos de processamento de imagens. Todas as imagens que seguem foram moduladas com 16-QAM e taxa de código 2/3.



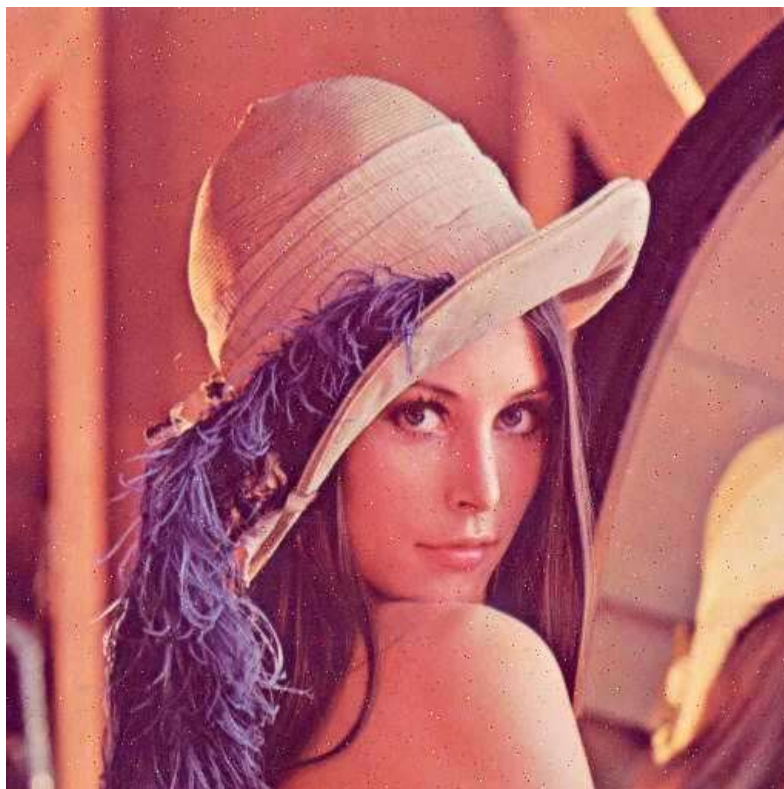
**Figura 5.11. Imagem original que será transmitida.**

A figura 5.12 mostra o efeito do ruído sobre uma imagem transmitida em um canal AWGN, com RSR de 5 dB.



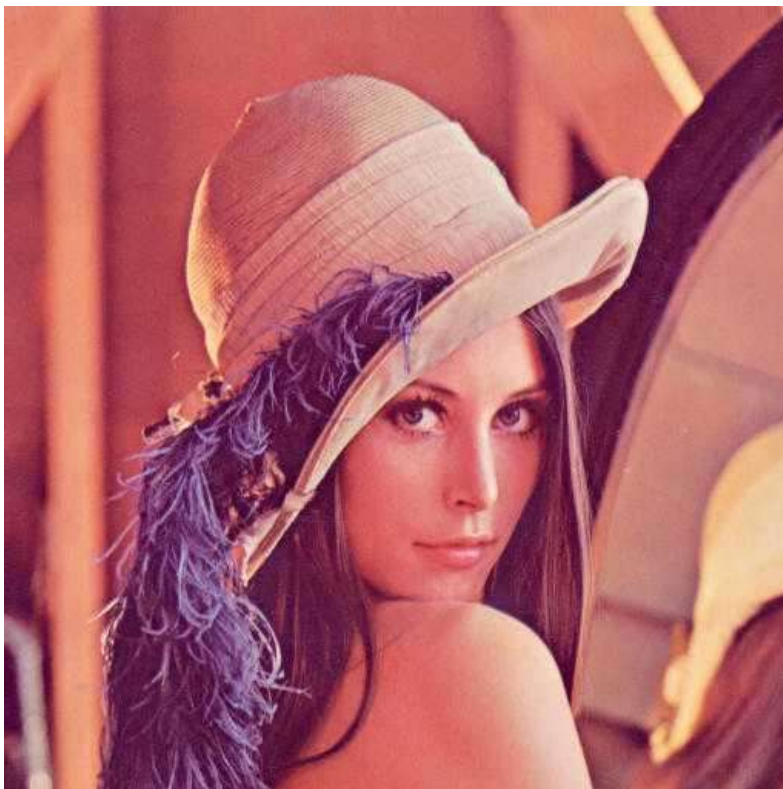
**Figura 5.12. Imagem recebida com RSR de 5dB.**

A figura 5.13 mostra o efeito do ruído sobre uma imagem transmitida em um canal AWGN, com RSR de 10 dB.



**Figura 5.13. Imagem recebida com RSR de 10 dB.**

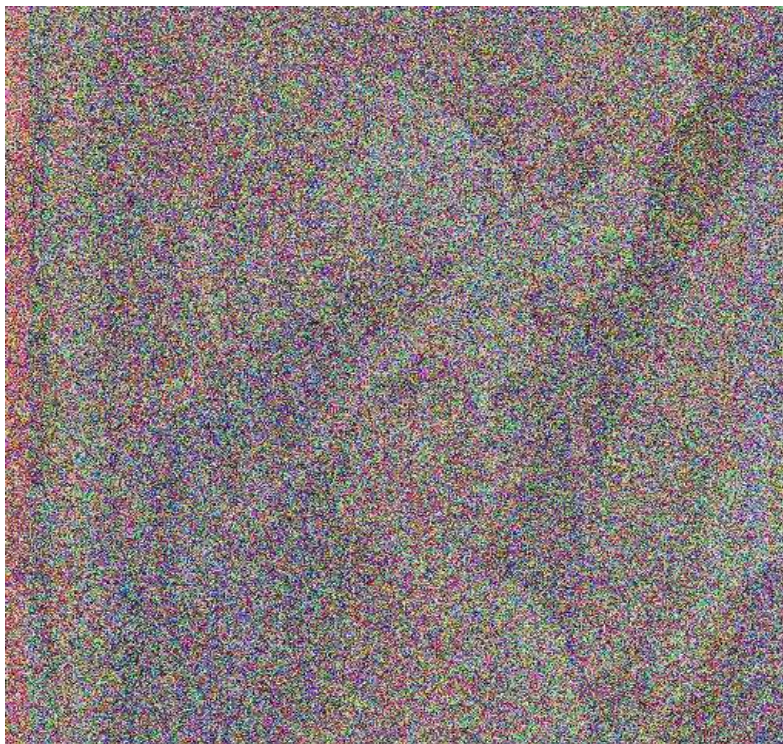
A figura 5.14 mostra o efeito do ruído sobre uma imagem transmitida em um canal AWGN, com RSR de 15 dB.



**Figura 5.14. Imagem recebida com RSR de 15 dB.**

Pode-se inferir, da inspeção nas figuras 5.12 a 5.14, que é necessário uma RSR de pelo menos 15 dB para recuperar-se adequadamente a imagem transmitida sem deformações perceptíveis, sem a utilização de algoritmos adicionais.

A figura 5.15 mostra o efeito do ruído de fase sobre uma imagem transmitida em um canal AWGN, com RSR de 15 dB.



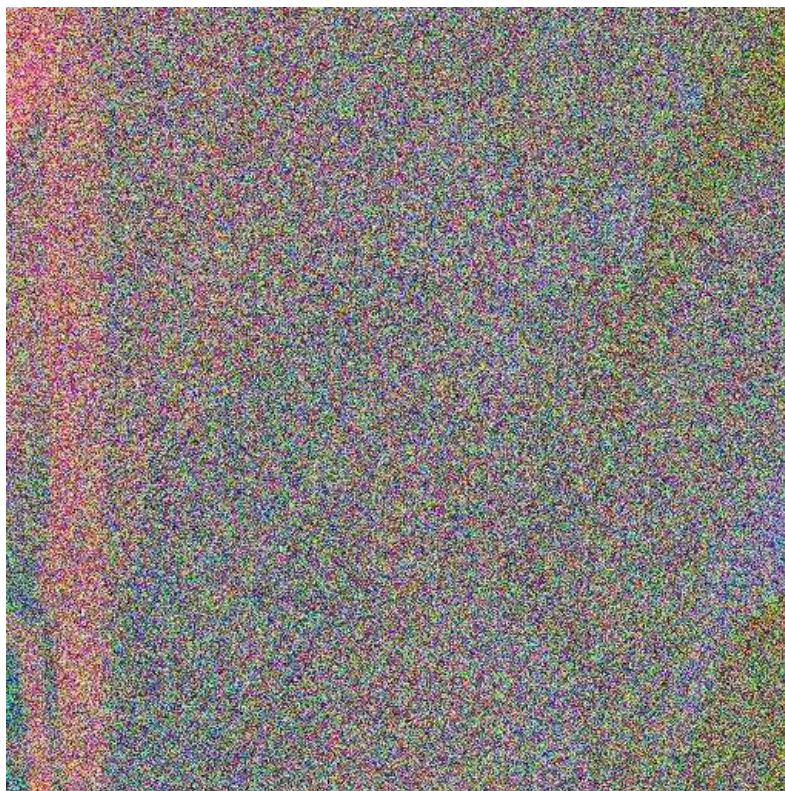
**Figura 5.15. Imagem recebida com RSR de 15 dB e com inserção de ruído de fase.**

A figura 5.16 mostra o efeito do ruído de fase sobre uma imagem transmitida em um canal AWGN, com RSR de 30 dB.



**Figura 5.16. Imagem recebida com RSR de 30 dB e com inserção de ruído de fase.**

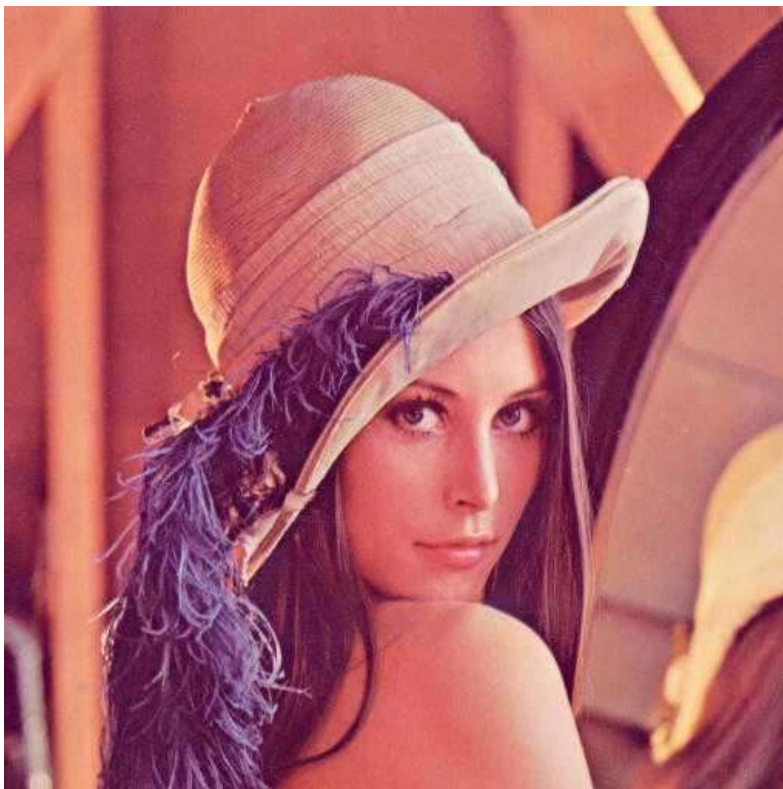
A figura 5.17 mostra o efeito do ruído de fase sobre uma imagem transmitida em um canal AWGN, com RSR de 60 dB.



**Figura 5.17. Imagem recebida com RSR de 60 dB e com inserção de ruído de fase.**

A figura 5.18 mostra o efeito do ruído de fase sobre uma imagem transmitida em um canal AWGN, com RSR de 15 dB e utilização de algoritmo de detecção de canal baseado nos símbolos de treinamento.





**Figura 5.18. Imagem recebida com RSR de 15 dB, com inserção de ruído de fase e utilização de algoritmo de estimação de canal.**

Verifica-se, nas figuras 5.16 e 5.17, que o ruído de fase altera de tal forma a imagem transmitida, rotacionando as subportadoras, que não é mais possível fazer uma demodulação e detecção adequadas, mesmo aumentando a RSR para valores elevados como 60 dB. No entanto, verifica-se, na figura 5.18, que se utilizando um algoritmo de estimação de canal, esta rotação de fase é mitigada de tal forma que é possível uma detecção e demodulação adequadas para uma RSR de 15 dB.

## 6 CONCLUSÃO

O período despendido para a realização do presente trabalho mostrou-se muito proveitoso, principalmente no que concerne ao contato e aprendizado de tecnologias até então desconhecidas ou de pouco conhecimento de minha parte. O estudo do simulador da camada física de um sistema de comunicação sem fio baseado na norma IEEE 802.11a requereu uma revisão e um aprofundamento nos conceitos aprendidos até então, principalmente nas disciplinas de Princípios de Comunicação, Telefonia-A, Tópicos Especiais em Telecomunicações e Sistemas e Sinais.

Muitos conceitos são fundamentais para o entendimento do funcionamento do sistema OFDM, no qual é baseada a norma 802.11a, como teoria da probabilidade, processamento digital de sinais, Transformada de Fourier e processos estocásticos. Estes conhecimentos necessitaram ser revisados para o entendimento do funcionamento do simulador.

Pode-se dividir o tempo despendido neste trabalho em quatro etapas:

1. Estudo teórico da norma 802.11a e das referências bibliográficas básicas;
2. Processo de análise e documentação do funcionamento do simulador implementado por meio do Software Matlab;
3. Modificação de funções deste simulador, bem como criação de novas funções, necessárias para que fossem gerados os resultados desejados, conforme explicitado no capítulo 5;
4. Implementação de um algoritmo de estimação de canal, baseado na estrutura das subportadoras piloto.

A realização plena das duas primeiras etapas foi fundamental para o sucesso na terceira e quarta etapas, onde foram explicitados os resultados palpáveis. A criação das funções que permitem a extração do conteúdo binário de uma imagem, para logo em seguida ser codificado, modulado, transmitido e recebido, demodulado e decodificado, possibilitam ao usuário do simulador verificar visualmente o efeito do ruído AWGN e do desvanecimento sobre o sinal transmitido.

Foram feitas modificações que permitem gerar gráficos que mostram visualmente as probabilidades de erro de bit e de pacote em função da razão sinal para ruído. Estes gráficos são essenciais para a comparação de performance entre os diferentes algoritmos utilizados neste simulador, bem como para validar a confiabilidade dos resultados obtidos por meio da simulação de Monte Carlo, ao equiparar na mesma janela gráfica estes resultados com as curvas teóricas previstas.

Finalmente, foi criado, por final, após revisão da literatura, um algoritmo de estimação de canal que utiliza as subportadoras piloto para estimar a resposta em frequência do canal de rádio-móvel para todas as outras subportadoras.

Salienta-se que o aprendizado do funcionamento do simulador e da norma 802.11a, bem como o estudo da estimação de canal, em variados livros de referência e artigos científicos do IEEE, permitem que se possa seguir, na seqüência deste trabalho de conclusão, com a implementação efetiva de novos algoritmos, não só de estimação de canal, mas também relativos aos demais blocos do simulador, como rastreamento da fase da portadora, estimação da chegada de pacotes, etc..

É importante também ressaltar que o estudo da norma 802.11a, bem como de algoritmos de estimação de canal, em variados livros de referência e artigos científicos do IEEE, juntamente com o processo de entendimento, documentação e modificação do simulador 802.11a realizados neste trabalho de conclusão, permitirão a subsequente reutilização deste simulador, com o intuito de modificar e implementar novos algoritmos em quaisquer dos blocos componentes. Poder-se-á então comparar o desempenho destes novos algoritmos.

Uma proposta interessante seria a de tomar-se como base este simulador que implementa o padrão 802.11a para construir outro simulador, baseado no novo padrão 802.11n, que será largamente utilizado em um futuro próximo, devido às altas taxas de transferência de dados oferecidas.

## 7 REFERÊNCIAS BIBLIOGRÁFICAS

[ATH 2003] Athaudage, C. R. N. ; Jayalath, D (2003). Low-complexity channel estimation for wireless OFDM systems. In: 14th IEEEConference on Personal, Indoor and Mobile Radio Communications. Beijing, [China], 2003, p.521-525.

[IEEE 1999] IEEE Computer Society LAN/MAN Standards Committee, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. High-speed Physical Layer in the 5GHz Band. In: IEEE Std 802.11a-1999(R2003) (Supplement to IEEE Std 802.11-1999), The Institute of Electrical and Electronics Engineers. New York, 1999.

[NAN 2009] Nanping, L; Yuan, Y; Kewen, X; Zhiwei, Z.. Study on Channel Estimation Technology in OFDM System. In: 2009 International Joint Conference on Artificial Intelligence. Hai Kou [China],2009, p.773-776.

[PRO 2001] Proakis, J. G., **Digital Communications**, 4th edition. New York: McGraw-Hill International Editions, 2001. 1024 p. ISBN 0071181830, 2001.

[TER 2001] Terry, J.; Heiskala, J.. **OFDM Wireless LANs: A Theoretical and Practical Guide Book**. Indianapolis: Sams, 2001. 315 p. ISBN: 0521845270.

[TSE 2005] Tse, D.; Viswanath, P.. **Fundamentals of Wireless Communication**. New York: Cambridge University Press, 2005. 586 p. ISBN: 0521845270.

## APÊNDICE:

Protótipos das funções em Matlab correspondentes ao simulador IEEE 802.11a



















































