

Universidade Federal do ABC
Graduação em Engenharia de Informação

Bruno Crepaldi Ambrózio

ALOCAÇÃO DE RECURSOS EM SISTEMAS OFDM

Santo André – SP
2016

Bruno Crepaldi Ambrózio

ALOCAÇÃO DE RECURSOS EM SISTEMAS OFDM

Monografia

Monografia apresentada ao curso de graduação da Universidade Federal do ABC,
como requisito parcial para obtenção do grau de Engenheiro de Informação

Orientador: Prof. Dr. Ivan Roberto Santana Casella

Santo André – SP

2016

Bruno Crepaldi Ambrózio

ALOCAÇÃO DE RECURSOS PARA SISTEMAS OFDM / Bruno Crepaldi Ambrózio – Santo André, SP: 2016.

Orientador: Prof. Dr. Ivan Roberto Santana Casella.
152p.

Trabalho de Graduação – Universidade Federal do ABC – UFABC,
2016.

1. OFDM. 2. Alocação. 3. *Water-filling*. I. Ivan Roberto Santana Casella. II. Universidade Federal do ABC. III. Centro de Engenharia, Modelagem e Ciências Sociais Aplicadas. IV. Alocação de Recursos para Sistemas OFDM.



Universidade Federal do ABC

ATA DE DEFESA DE TRABALHO DE GRADUAÇÃO EM ENGENHARIA DE INFORMAÇÃO

Ata de Defesa do Trabalho de Graduação em Engenharia de Informação da Universidade Federal do ABC

No dia 30 de Novembro de 2016 reuniu-se a banca examinadora do trabalho apresentado como Trabalho de Graduação em Engenharia de Informação de Bruno Crepaldi Ambrózio, intitulado: "Alocação de Recursos em Sistemas OFDM". Após a exposição oral, o aluno foi arguido pelos componentes da banca que se reuniram reservadamente, e decidiram atribuir o conceito final A.

Orientador
Prof. Dr. Ivan Roberto Santana Casella

Avaliador 1
Prof. Dr. João Henrique Kleinschmidt

Avaliador 2
Prof. Dr. Murilo Bellezoni Lolola

DEDICATÓRIA

À minha família, pelo apoio em minha escolha.

A todos meus professores de graduação da

Universidade Federal do ABC.

Aos meus colegas de curso e estudo.

Ao meu orientador Prof. Dr. Ivan Roberto Santana Casella.

AGRADECIMENTOS

Agradeço, primeiramente, à Universidade Federal do ABC por estes cinco anos de intenso aprendizado e crescimento pessoal. Às amigadas que fiz no curso de Engenharia de Informação e que me auxiliaram direta ou indiretamente para conclusão desta monografia.

Ao meu orientador Prof. Dr. Ivan Roberto Santana Casella pela sua imensa vontade de me auxiliar nos momentos de dúvidas e, principalmente, pela paciência e compartilhamento de informações que foram de primordial importância para realização deste trabalho. Também agradeço pelos conselhos pessoais, que me ajudaram a tomar decisões importantes em minha graduação.

Agradeço aos professores do curso de Engenharia de Informação da Universidade Federal do ABC, que sempre foram muito solícitos e também fazem parte de meu amadurecimento por estes cinco anos.

Por fim, agradeço aos meus pais e minha família, que sempre me apoiaram em minha decisão de seguir o curso de Engenharia de Informação e estiveram presentes nesta nova etapa da minha vida.

EPÍGRAFE

*“Julgue seu sucesso pelas coisas que
você teve que renunciar para conseguir.”*

(Dalai Lama)

RESUMO

Com a evolução de comunicação por redes sem fio, a busca por transmissão de dados em altas taxas se faz cada vez mais necessária frente à comunicação de voz pelas redes celulares. Neste âmbito, a transmissão por canais de radiopropagação considera aspectos de larga e pequena escalas para modelar o comportamento do sinal no caminho entre transmissor e receptor. A técnica de OFDM (*Orthogonal Frequency Division Multiplexing*) é amplamente utilizada para transmissão de sinais em canais com desvanecimento seletivo em frequência e permite a eliminação de interferência intersimbólica (ISI) entre as subportadoras, se um intervalo de guarda é corretamente utilizado.

Neste trabalho, a técnica de OFDM foi estudada, com uma breve modelagem matemática de sistemas que a utilizam, de tal forma a simular um sistema OFDM em MATLAB. Os resultados mostraram que, em comparação com canais AWGN (*Additive White Gaussian Noise*), canais de multipercurso sob as mesmas condições de simulação possuem pior desempenho com relação à probabilidade de erro de bit, ou seja, existem mais erros de bit na transmissão pelo canal de radiopropagação considerando a mesma SNR (*Signal-to-Noise Ratio*). Entretanto, existem diversas técnicas que possuem como objetivo melhorar este desempenho do sistema, por exemplo, codificação, *interleaving* e alocação de recursos.

Sendo assim, três tipos de técnicas de alocação de recursos em sistemas OFDM foram estudadas (*water-filling* e algoritmos de Chow e Levin-Campello), com o desenvolvimento de algoritmos em MATLAB. Com os resultados obtidos, foi possível constatar que a utilização destas técnicas tem por objetivo alocar potência e taxa dependendo das SNRs de cada subportadora. Desta maneira, subportadoras que apresentam baixas SNRs proporcionadas pelo canal de radiopropagação transmitem em taxas menores, em contraste com as que possuem melhores SNRs.

Também foi possível observar por meio das simulações que a solução *water-filling* é ótima. Entretanto, as soluções de Chow e Levin-Campello, apesar de sub-ótimas, consideram granularidade finita do sistema OFDM, o que ocorre na prática em sistemas reais.

ABSTRACT

With the evolution of communication over wireless networks, the search for data transmission at high data rates is increasingly needed compared to voice communication over cellular networks. In this context, the transmission by radio propagation channels considers large and small scale aspects in order to model the signal's behavior between the transmitter and the receiver. The OFDM (Orthogonal Frequency Division Multiplexing) technique is widely used for signal transmission in channels with selective fading and allows the removal of intersymbol interference (ISI) between subcarriers, if a guard interval is correctly used.

In this text, the OFDM technique was studied, with a brief mathematical modeling of systems that use it, in order to simulate an OFDM system with MATLAB. The results showed that, compared to AWGN (Additive White Gaussian Noise) channels, multipath channels under the same simulation conditions have worse performance in relation to the probability of bit error, that is, there are more bit errors in the transmission over the radio propagation channel considering the same SNR (Signal-to-Noise Ratio). However, there are several techniques that aim to improve this system performance, such as coding, interleaving and resource allocation.

Therefore, three types of resource allocation techniques in OFDM systems were studied (water-filling, Chow's and Levin-Campello's algorithms), also with the development of MATLAB algorithms. With the obtained results, it was possible to verify that the use of these techniques aims to allocate power and rate depending on each subcarrier SNR. In this way, subcarriers that have low SNR provided by the radio propagation channel transmit at lower rates, in contrast to those with better SNR.

It was also possible to observe, through MATLAB simulations, that the water-filling solution is optimal. However, Chow and Levin-Campello solutions, although sub-optimal, consider finite granularity of the OFDM system, which occurs in practice in real systems.

LISTA DE FIGURAS

Figura 1.1: Elementos básicos de um sistema de comunicação (HAYKIN e MOHER, 2008).	1
Figura 1.2: Regeneração de um pulso digital (PROAKIS e SALEHI, 2002).	2
Figura 1.3: Modelo de um sistema de comunicação digital (PROAKIS e SALEHI, 2002).	3
Figura 1.4: Subportadoras OFDM sobrepostas no domínio da frequência (BAHAL e SALTZBERG, 1999).	5
Figura 2.1: Mecanismos básicos de propagação em multipercursos.	8
Figura 2.2: Espectro Doppler (LINNARTZ).	11
Figura 2.3: Desvanecimento plano em frequência (RAPPAPORT, 2008).	12
Figura 2.4: Desvanecimento seletivo em frequência (RAPPAPORT, 2008).	13
Figura 2.5: Tipos de desvanecimento em canais de propagação (RAPPAPORT, 2008).	14
Figura 2.6: Comportamento da distribuição de Rayleigh no domínio temporal (RAPPAPORT, 2008).	15
Figura 2.7: Conceito de transmissão de dados em paralelo (TAROKH, 2009).	18
Figura 2.8: Modulação por múltiplas portadoras (RETHNAKARAN e DAWID, 2003).	19
Figura 2.9: Modulação OFDM para um bloco de dados específico x_n (RETHNAKARAN e DAWID, 2003).	20
Figura 2.10: Prefixo cíclico na técnica de OFDM.	25
Figura 2.11: Diagrama de blocos geral de um sistema OFDM (PRASAD, 2004).	25
Figura 2.12: Modelo banda-base de um sistema OFDM (PRASAD, 2004).	27

Figura 2.13: Constelações <i>M</i> -PSK representadas na base ortonormal (PROAKIS e SALEHI, 2002).	30
Figura 2.14: Constelações <i>M</i> -QAM representadas na base ortonormal (PROAKIS e SALEHI, 2002).	32
Figura 2.15: Exemplo de codificador convolucional com $N_{Conv} = 3$ e $l = 2$ (LATHI e DING, 2009).	39
Figura 2.16: Representação do codificador da Figura 2.15 em árvore (LATHI e DING, 2009).	40
Figura 2.17: Representação do codificador da Figura 2.15 em máquina de estados (LATHI e DING, 2009).	41
Figura 2.18: Representação do codificador da Figura 2.15 em diagrama de treliças (LATHI e DING, 2009).	42
Figura 2.19: Análise dos dois primeiros estágios do algoritmo de Viterbi para exemplo da Figura 2.18 (LATHI e DING, 2009).	43
Figura 2.20: Análise do terceiro estágio do algoritmo de Viterbi para exemplo da Figura 2.18 (LATHI e DING, 2009).	44
Figura 2.21: Análise do último estágio do algoritmo de Viterbi para exemplo da Figura 2.18 (LATHI e DING, 2009).	44
Figura 2.22: Probabilidades de erro para <i>M</i> -QAM, em função da SNR.	48
Figura 2.23: Taxa normalizada em função da SNR para diferentes valores de <i>gap</i>	50
Figura 2.24: Algoritmos de <i>Water-Filling</i> , soluções RA e MA (LI e STÜBER, 2006).	59
Figura 2.25: Resumo das soluções RA e MA para o método de Levin-Campello (LI e STÜBER, 2006).	70
Figura 3.1: Entrada de parâmetros OFDM para um exemplo de simulação.	73

Figura 3.2: Módulo (acima) e fase (abaixo) da resposta em frequência para o canal simulado neste exemplo.....	78
Figura 3.3: Constelações 16-QAM recebidas antes da equalização ZF, SNR = 20dB.	79
Figura 3.4: Constelações 16-QAM recebidas após equalização ZF, SNR = 20dB...81	
Figura 3.5: Desempenho de um sistema OFDM apenas considerando ruído AWGN.	82
Figura 3.6: Desempenho de um sistema OFDM considerando multipercurso.	83
Figura 3.7: Desempenho de um sistema OFDM considerando diferentes tamanhos de CP.	84
Figura 3.8: Desempenho de um sistema OFDM considerando multipercurso e modulação 8-PSK.	85
Figura 3.9: Constelações 16-QAM equalizadas com ZF, SNR = 10dB.	86
Figura 3.10: Constelações 16-QAM equalizadas com MMSE, SNR = 10dB.....	87
Figura 3.11: Desempenho de um sistema OFDM considerando equalização ZF e MMSE (16-QAM).....	88
Figura 3.12: Desempenho de um sistema OFDM com codificação de repetição.	89
Figura 3.13: Desempenho de um sistema OFDM com codificação de Hamming.	90
Figura 3.14: Códigos convolucionais utilizados na simulação (WADE, 1994).....	91
Figura 3.15: Desempenho de um sistema OFDM com codificação convolucional. ...	91
Figura 3.16: Desempenho de um sistema OFDM com código de repetição e <i>interleaving</i>	93
Figura 3.17: Desempenho de um sistema OFDM com código de Hamming e <i>interleaving</i>	94

Figura 3.18: Desempenho de um sistema OFDM com código convolucional e <i>interleaving</i> .	95
Figura 3.19: Módulo da resposta em frequência do canal simulado, para 8 subportadoras.	96
Figura 3.20: Solução <i>water-filling</i> RA em gráfico de barras.	101
Figura 3.21: Solução <i>water-filling</i> , comparação entre a resposta do canal e a alocação de potência e taxa normalizada.	102
Figura 3.22: Solução <i>water-filling</i> RA, saída no console do MATLAB.	103
Figura 3.23: Solução Chow RA, saída no console do MATLAB.	109
Figura 3.24: Solução Chow RA, comparação da taxa com o caso ótimo, antes do arredondamento.	110
Figura 3.25: Tabela de potências incrementais em função da taxa normalizada para os oito subcanais simulados neste exemplo.	110
Figura 3.26: Solução Levin-Campello RA, saída no console do MATLAB.	113
Figura 3.27: Solução <i>water-filling</i> RA para 64 subportadoras.	114
Figura 3.28: Solução <i>water-filling</i> MA em gráfico de barras.	119
Figura 3.29: Solução <i>water-filling</i> MA, saída no console do MATLAB.	120
Figura 3.30: Solução Chow MA, saída no console do MATLAB.	123
Figura 3.31: Solução Levin-Campello MA, saída no console do MATLAB.	125
Figura 3.32: Solução <i>water-filling</i> MA para 64 subportadoras.	125
Figura 3.33: Alocação de potência para os algoritmos de alocação de recursos analisados.	128
Figura 3.34: Comparação entre taxas normalizadas por subportadora em função da SNR para os algoritmos analisados.	129

Figura 3.35: Comparação entre taxas normalizadas por subportadora em função da SNR para os algoritmos analisados, com modificação do *gap* de SNR. 130

LISTA DE TABELAS

Tabela 2.1: Expoentes de perda de percurso para diferentes ambientes (RAPPAPORT, 2008).....	16
Tabela 2.2: Eficiências de banda para diversos tipos de modulação.	47
Tabela 2.3: Análise comparativa de <i>gap</i> de SNR para modulações <i>M</i> -QAM.....	49
Tabela 3.1: Parâmetros utilizados para simulação de um sistema OFDM com canal fixo.....	77
Tabela 3.2: Parâmetros utilizados para simulação de um sistema OFDM com diversas iterações de canais.	82
Tabela 3.3: Alocação de potência para as soluções RA.....	115
Tabela 3.4: Alocação de taxa normalizada para as soluções RA.	116
Tabela 3.5: Alocação de potência para as soluções MA.	126
Tabela 3.6: Alocação de taxa normalizada para as soluções RA.	127

LISTA DE SIGLAS

ADSL	<i>Asymmetric Digital Subscriber Line</i>
ASK	<i>Amplitude-Shift Keying</i>
AWGN	<i>Additive White Gaussian Noise</i>
BER	<i>Bit Error Rate</i>
BPSK	<i>Binary Phase-Shift Keying</i>
CP	<i>Cyclic Prefix</i>
DAB	<i>Digital Audio Broadcasting</i>
DC	<i>Direct Current</i>
DFT	<i>Discrete Fourier Transform</i>
DMT	<i>Discrete Multitone Modulation</i>
DVB	<i>Digital Video Broadcasting</i>
ERB	<i>Estação Rádio Base</i>
FDM	<i>Frequency Division Multiplexing</i>
FFT	<i>Fast Fourier Transform</i>
FSK	<i>Frequency-Shift Keying</i>
ICI	<i>Intercarrier Interference</i>
IDFT	<i>Inverse Discrete Fourier Transform</i>
IDTFT	<i>Inverse Discrete Time Fourier Transform</i>
IFFT	<i>Inverse Fast Fourier Transform</i>
ISI	<i>Intersymbol Interference</i>
MA	<i>Margin Adaptive</i>
MCM	<i>Multicarrier Modulation</i>
MLD	<i>Maximum Likelihood Decision</i>
MIMO	<i>Multiple-Input and Multiple-Output</i>
MMSE	<i>Minimum Mean-Square Error</i>
NRZ	<i>Non-return-to-zero</i>
OFDM	<i>Orthogonal Frequency Division Multiplexing</i>
OFDMA	<i>Orthogonal Frequency Division Multiple Access</i>
PAPR	<i>Peak-to-Average Power Ratio</i>
PLC	<i>Power Line Communication</i>
PSK	<i>Phase-Shift Keying</i>

QAM	<i>Quadrature Amplitude Modulation</i>
RA	<i>Rate Adaptive</i>
RDSI	<i>Rede Integrada de Serviços Digitais</i>
RMS	<i>Root Mean Square</i>
SCS	<i>Single Carrier Systems</i>
SER	<i>Symbol Error Rate</i>
SNR	<i>Signal-to-Noise Ratio</i>
WLAN	<i>Wireless Local Area Networks</i>
ZF	<i>Zero-Forcing</i>

SUMÁRIO

1. INTRODUÇÃO.....	1
1.1. Justificativa	5
1.2. Objetivos.....	6
2. REVISÃO DE LITERATURA.....	7
2.1. Canais de radiopropagação.....	7
2.2. Técnica de OFDM.....	17
2.2.1 Modelagem matemática	17
2.2.2 Deficiências de sistemas OFDM.....	27
2.2.3 Modulações utilizadas com OFDM	29
2.2.3.1 Modulação M -ária por deslocamento de fase (M -PSK).....	29
2.2.3.2 Modulação M -ária de amplitude em quadratura (M -QAM)	31
2.3. Códigos corretores de erro	33
2.3.1 Códigos de bloco lineares	34
2.3.2 Códigos convolucionais	38
2.3.3 <i>Interleaving</i>	45
2.4. Técnicas de alocação de recursos em sistemas sem fio OFDM.....	46
2.4.1 O conceito de <i>gap</i> de SNR	46
2.4.2 Classificação dos algoritmos de alocação de recursos	52
2.4.3 <i>Water-filling</i>	53
2.4.4 Algoritmos de Chow.....	60
2.4.5 Algoritmos de Levin-Campello	64
3. ANÁLISE DE RESULTADOS	71
3.1. Sistema OFDM	71
3.2. Simulação de algoritmos de alocação de recursos	95
3.2.1 Soluções RA.....	99

3.2.2	Soluções MA.....	116
3.2.3	Análise comparativa	127
4.	CONSIDERAÇÕES FINAIS.....	131
	REFERÊNCIAS BIBLIOGRÁFICAS	132
	ANEXO A: Distribuição de Rayleigh.....	135

1. INTRODUÇÃO

A busca por informação caminha em conjunto com a evolução do ser humano. No passado, mensagens eram transportadas por corredores, pombos-correio, batidas de tambor e tochas. Estes esquemas eram adequados para as distâncias e “taxas de dados” da época. Na maior parte do mundo, estes modos de comunicação foram suplantados por sistemas de comunicação elétricos, que podem transmitir sinais por distâncias muito maiores (até mesmo para planetas distantes e galáxias) e na velocidade da luz (LATHI, 1998).

Neste âmbito, um sistema de comunicação envolve a transmissão da informação de um ponto a outro por uma sucessão de processos. Independente do processo de comunicação em análise, existem três elementos básicos que delineiam o funcionamento de um sistema deste tipo, a saber: transmissor, canal e receptor. Estes três elementos são visualizados na Figura 1.1, exposta abaixo.

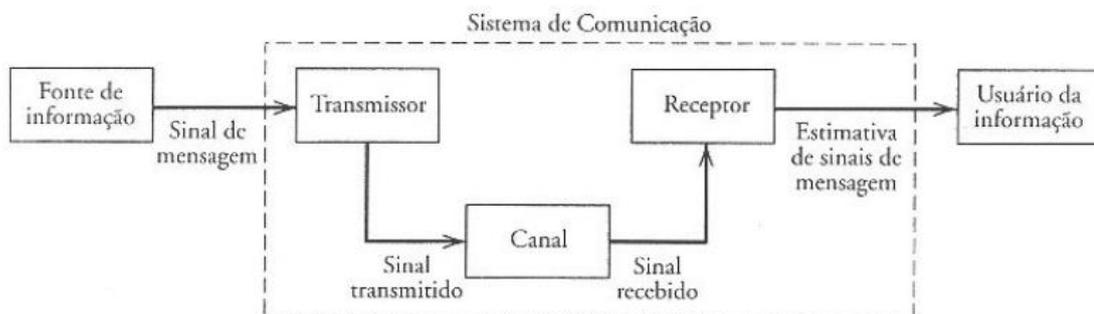


Figura 1.1: Elementos básicos de um sistema de comunicação (HAYKIN e MOHER, 2008).

O propósito do transmissor é o de converter o sinal de mensagem produzido pela fonte de informação em uma forma adequada à transmissão por um canal. Entretanto, à medida que o sinal se propaga, este é distorcido devido às imperfeições do próprio canal. Além do mais, ruídos e sinais interferentes (que se originam de outras fontes) são acrescentados à saída do canal, resultando no sinal recebido. O receptor tem a tarefa de operar sobre o sinal recebido a fim de reconstruir uma forma reconhecível do sinal original para o usuário (HAYKIN, 2007). Por canal de comunicação, pode-se entender como todo o meio físico que existe entre e ao redor do conjunto transmissor/receptor.

Em sistemas de comunicação, dois recursos são de principal importância: potência transmitida e largura de banda do canal. O objetivo de projeto de um sistema de comunicação é utilizar esses dois recursos da maneira mais eficiente possível (HAYKIN, 2007). Por este motivo, deseja-se transmitir a informação de maneira eficiente pelo canal de comunicação, ocupando a menor banda de frequência possível, bem como reduzindo os efeitos de interferência e ruído simultaneamente.

Considerando sistemas de comunicação elétricos, os sinais de mensagem podem assumir duas naturezas distintas: analógica e digital. A primeira destas envolve variação contínua da amplitude (tensão) deste sinal em função do tempo, de tal forma que existe um número infinito de níveis que o sinal pode assumir. Por outro lado, a natureza digital é caracterizada pela variação discreta da amplitude em função do tempo, de tal forma que existe um número finito de níveis de tensão disponíveis.

Analisando as características expostas no parágrafo anterior, é possível observar que sistemas que operam com sinais de natureza digital possuem maior imunidade ao ruído e distorções do canal, visto que o receptor precisa distinguir entre um número finito de símbolos, apenas. Ademais, um sinal digital pode ser regenerado completamente, ao passo que, em sistemas analógicos, o ruído seria amplificado em conjunto, dificultando uma transmissão por longas distâncias. A Figura 1.2 mostra um processo simples de regeneração de um pulso digital.

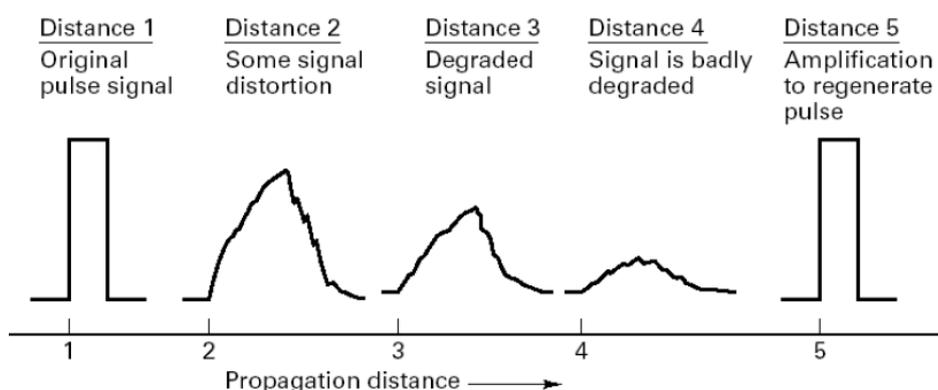


Figura 1.2: Regeneração de um pulso digital (PROAKIS e SALEHI, 2002).

Além do benefício exposto na figura anterior, sinais analógicos podem ser redundantes, característica que pode ser amenizada ou até removida em sistemas

de comunicações digitais que utilizam codificação antes do processo de transmissão.

Do ponto de vista da transmissão eficiente do sinal de mensagem (informação), modificações neste sinal podem ser realizadas com o intuito de representá-lo em uma forma de onda mais adequada, proporcionando melhorias no sistema. Este processo é chamado de modulação e pode ser feito tanto em frequências próximas da componente DC (*Direct Current*), o que se chama de modulação passa-baixa, banda base ou codificação de linha, quanto em frequências mais altas (modulação passa-faixa ou banda passante). Sendo assim, características como amplitude, frequência e fase de uma ou mais portadoras são modificadas de acordo com o sinal original. Exemplos de modulações passa-baixa são as codificações unipolar e polar *Non-Return-to-Zero* (polar NRZ), enquanto que para modulações passa-faixa tem-se como exemplos a modulação ASK (*Amplitude-Shift Keying*) e FSK (*Frequency-Shift Keying*). Neste âmbito, a Figura 1.3 ilustra um modelo mais detalhado para um sistema básico de comunicação digital.

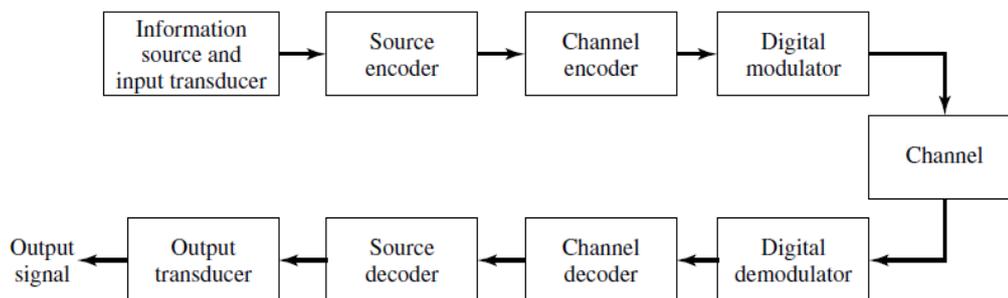


Figura 1.3: Modelo de um sistema de comunicação digital (PROAKIS e SALEHI, 2002).

Na figura anterior, é importante ressaltar que a etapa de codificação da fonte (*source encoder*) consiste em codificar a mensagem emitida pela fonte de informação, de maneira a possuir, ao final deste processo, uma sequência de dígitos binários (0s e 1s) com pouca ou nenhuma redundância. Esta nova sequência passa pela etapa de codificação de canal, na qual são introduzidas, de maneira controlada, redundâncias, com o intuito de identificar e corrigir os erros provenientes do ruído e interferências que inevitavelmente afetarão o sinal em seu caminho pelo canal de comunicação. Por fim, tem-se a etapa de modulação, cujo objetivo principal já foi discutido anteriormente.

No receptor digital, o processo inverso é realizado, no sentido de recuperar a informação inicialmente transmitida. O sinal de saída, por muitas vezes, possui erros relacionados ao processo de transmissão e difere do sinal de mensagem. Um fator importante a ser considerado é a interferência intersimbólica (em inglês, *intersymbol interference* ou ISI) que pode afetar gravemente o desempenho de um sistema de comunicação.

A ISI ocorre quando um símbolo (combinações de bits) transmitido interfere em símbolos subsequentes, de tal forma que, no momento de decisão do receptor digital, pode ocorrer uma escolha errada, prejudicando a recuperação da mensagem original. Este problema ocorre de maneira evidenciada em sistemas de transmissão banda base e em comunicação sem fio, devido à limitação de frequência do canal de comunicação e à propagação por multipercursos, respectivamente.

Até então, o esquema geral de um sistema de comunicação digital foi exposto e algumas observações foram realizadas. Conforme citado anteriormente, a modulação pode ocorrer com uma única portadora ou com múltiplas portadoras. Quando mais de uma portadora é utilizada (em inglês, *Multicarrier Modulation* ou MCM), um esquema interessante adotado é o de transmitir paralelamente a informação em subcanais de frequência, caracterizando uma multiplexação por divisão de frequência (em inglês, *Frequency Division Multiplexing* ou FDM).

Neste âmbito, a multiplexação ortogonal por divisão de frequência (em inglês, *Orthogonal Frequency Division Multiplexing* ou OFDM) é um esquema de MCM onde as subportadoras são ortogonais no domínio da frequência. Isto faz com que o sistema economize banda no espectro de frequências utilizado, já que não há interferência mútua entre os subcanais, conforme ilustra a Figura 1.4. Além disto, cada subportadora é modulada por algum tipo de técnica de modulação digital.

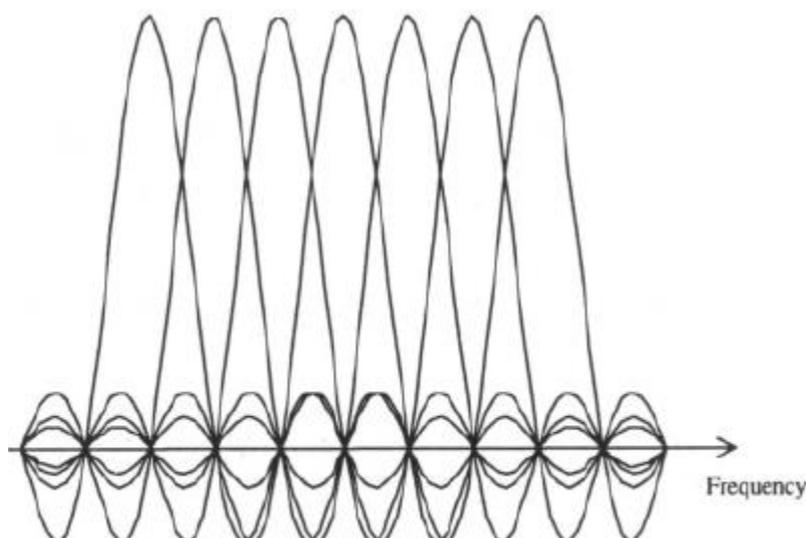


Figura 1.4: Subportadoras OFDM sobrepostas no domínio da frequência (BAHAI e SALTZBERG, 1999).

Assim como citado anteriormente, é interessante alocar de maneira eficiente recursos para cada uma das subportadoras, já que estas são afetadas de diferentes maneiras pelo canal de transmissão, o que será analisado com maiores detalhes posteriormente. Sendo assim, a potência pode ser aumentada ou reduzida em cada subportadora, bem como a taxa de transmissão pode ser ajustada pela mudança do tipo de modulação adotado.

1.1. Justificativa

A concepção da técnica de OFDM, proposta por (CHANG, 1966), possibilitou um novo pensamento para transmissão de dados em altas taxas. Entretanto, na época, a construção do *hardware* necessário para implementá-la não era viável por apresentar alto custo computacional.

Por este motivo, a maturidade de sistemas OFDM para aplicações comerciais só foi alcançada por volta da época de 1990 (HANZO e KELLER, 2006). Com o rápido crescimento da Internet e das redes de acesso, a taxa da rede digital integrada de serviços (RDSI) já não era mais suficiente, o que fez com que novas tecnologias como o ADSL (do inglês, *Assymmetric Digital Subscriber Line*) surgissem e alavancassem um novo mercado. Aliado ao forte crescimento da rede móvel de

telefonia celular, novas tecnologias tiveram de ser desenvolvidas para atender à demanda dos usuários (ENGELS, 2002).

Exemplos de aplicações que utilizam a técnica de OFDM são: *Digital Audio Broadcasting* (DAB), *Digital Video Broadcasting* (DVB) e *Wireless Local Area Networks* (WLAN) (HANZO e KELLER, 2006). Além disto, atualmente, existem diversos estudos que introduzem novos meios de aplicação em diversas áreas, como, por exemplo, a aplicação em transmissões com fibras ópticas (MA, YANG, *et al.*, 2009), (SHIEH, BAO e TANG, 2008), tecnologia de múltiplos usuários (em inglês, *Orthogonal Frequency Division Multiple Access* ou OFDMA) (YAGHOUBI, 2004), (SEONG, MOHSENI e CIOFFI, 2006) e sistemas com múltiplas entradas e múltiplas saídas (em inglês, *Multiple-Input and Multiple-Output* ou MIMO) (CHOO, KIM, *et al.*, 2010).

Por este motivo, nota-se que o estudo de funcionamento da técnica de OFDM é muito interessante tanto do ponto de vista tecnológico quanto do ponto de vista de aperfeiçoamento teórico, completando as informações vistas ao longo do curso de graduação em Engenharia de Informação. Além de estudar como a técnica funciona, também é importante entender como alocar os recursos disponíveis de maneira otimizada, buscando aproveitá-los da melhor maneira possível.

1.2. Objetivos

Este trabalho de graduação tem por objetivo principal o estudo da técnica de OFDM, com ênfase em alocação de recursos, visando à melhoria de desempenho do sistema. Além disto, também é objetivo deste trabalho a simulação de sistemas que utilizam OFDM utilizando o *software* MATLAB, para verificar as principais peculiaridades de algoritmos para alocação de recursos.

De maneira mais estrita, será analisado como alguns parâmetros de cada subportadora, como a potência e o tipo de modulação, podem ser modificados dependendo da resposta em frequência de cada subcanal. Para isto, uma análise matemática a respeito das características de canais de radiopropagação e da técnica de OFDM é feita inicialmente, com o intuito de servir como base teórica para aplicação dos conceitos de alocação de recursos.

2. REVISÃO DE LITERATURA

Nesta seção, é apresentada uma revisão dos conceitos básicos utilizados neste trabalho, alicerçada em diferentes fontes técnico-científicas como, por exemplo, livros, artigos e outras monografias. Inicialmente, uma visão geral sobre desvanecimento em canais de radiopropagação é apresentada, seguida de uma análise da técnica de OFDM e, finalmente, uma análise relacionada à alocação de recursos em sistemas que utilizam OFDM.

2.1. Canais de radiopropagação

Com o avanço dos sistemas de comunicação sem fio, cada vez mais a transmissão de dados em altas taxas se faz necessária frente à comunicação de voz pelas redes celulares. Entretanto, em um modelo sem fio, deve-se considerar que o sinal de informação propaga por meio de diversos percursos. Este comportamento de multipercursos ocorre, majoritariamente, devido a três mecanismos básicos de propagação, são eles: reflexão, dispersão e difração. Na Figura 2.1, são apresentadas a componente de propagação direta e algumas componentes de multipercurso, sujeitas aos diferentes mecanismos de propagação.

A reflexão ocorre quando uma onda eletromagnética em propagação colide com um objeto que possui dimensões muito grandes em comparação com o comprimento de onda. A dispersão, por sua vez, ocorre quando o meio pelo qual a onda trafega consiste de objetos com dimensões que são pequenas em comparação com o comprimento de onda, de tal forma que a onda é “espalhada” por estes obstáculos. Por outro lado, a difração ocorre quando o caminho de rádio entre o transmissor e o receptor é obstruído por uma superfície que possui irregularidades afiadas (arestas). Sendo assim, as ondas secundárias resultantes da superfície de obstrução estão presentes pelo espaço e até mesmo por trás do obstáculo, fazendo surgir uma curvatura de ondas ao seu redor, até mesmo quando não existe um caminho de linha de visão entre transmissor e receptor (RAPPAPORT, 2008).

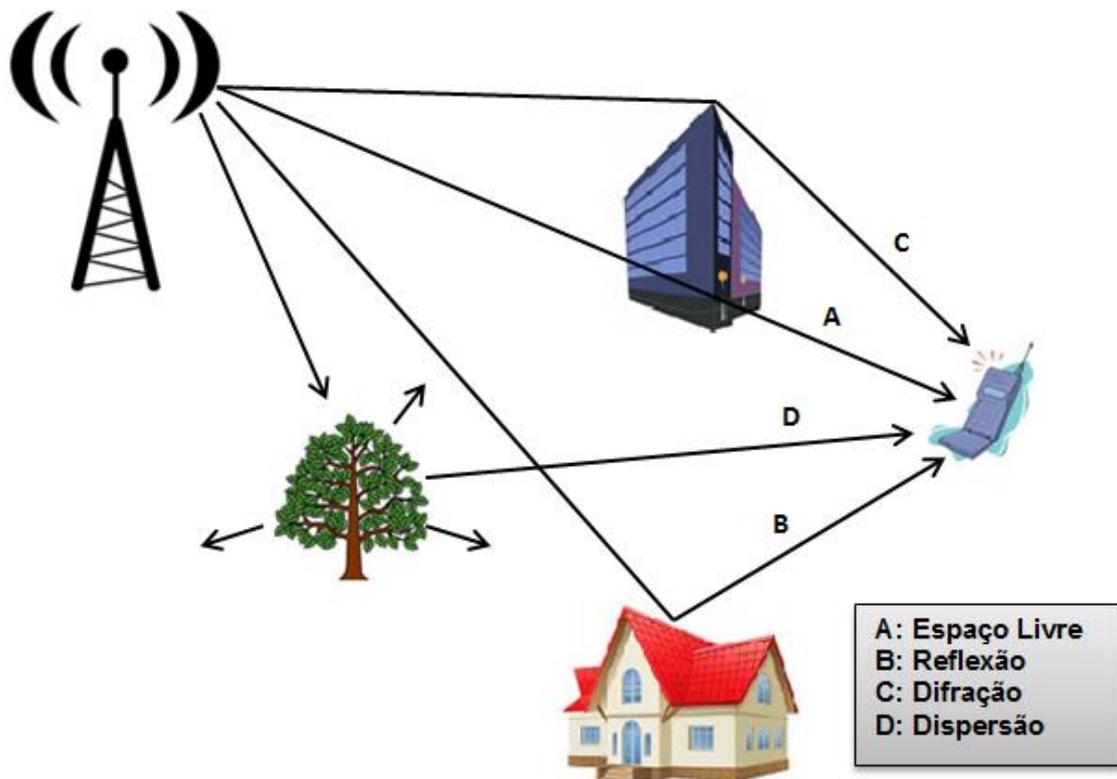


Figura 2.1: Mecanismos básicos de propagação em multipercursos.

Os mecanismos ilustrados na figura anterior possibilitam a análise do canal de propagação de maneira mais detalhada, ou seja, definindo parâmetros com o intuito de caracterizá-lo. Neste âmbito, parâmetros temporais são aqueles que podem ser medidos a partir de um perfil de atraso de potência de um canal. Este tipo de perfil representa a potência recebida em função do atraso temporal, para canais de multipercurso.

Com relação aos parâmetros de dispersão temporal, podem ser determinados o atraso médio em excesso, o valor RMS do espalhamento de atraso (do inglês, *Root Mean Square*) e o máximo atraso em excesso (XdB), dado um perfil de atraso de potência. As definições destes parâmetros foram retiradas de (RAPPAPORT, 2008) e são expostas a seguir.

O atraso médio em excesso é o primeiro momento do perfil de atraso de potência e é definido por:

$$\bar{\tau} = \frac{\sum_k P(\tau_k)\tau_k}{\sum_k P(\tau_k)} \quad (1)$$

Onde τ_k representa o atraso de caminho múltiplo do canal de multipercurso para o k -ésimo sinal recebido (para um valor temporal fixo) e $P(\tau_k)$ a potência recebida normalizada em relação ao máximo valor, também para o k -ésimo sinal recebido.

O valor RMS do espalhamento de atraso é obtido pela raiz quadrada do segundo momento central do perfil de atraso de potência, sendo definido como:

$$\sigma_\tau = \sqrt{\overline{\tau^2} - (\bar{\tau})^2} \quad (2)$$

Onde:

$$\overline{\tau^2} = \frac{\sum_k P(\tau_k)\tau_k^2}{\sum_k P(\tau_k)} \quad (3)$$

Vale ressaltar que estes atrasos são medidos com respeito ao primeiro sinal detectável que chega ao receptor ($\tau_0 = 0$).

O máximo atraso em excesso (XdB) é definido como o atraso de tempo durante o qual a energia do caminho múltiplo cai para X dB abaixo do máximo.

Além destes parâmetros temporais, é interessante notar que se pode obter uma descrição equivalente no domínio da frequência. Neste âmbito, a largura de banda de coerência (B_C) é utilizada para caracterizar o sinal no domínio da frequência e é derivada do valor RMS do espalhamento de atraso. O conceito de largura de banda de coerência está relacionado com uma medida estatística da faixa de frequências sobre as quais o canal pode ser considerado “uniforme” (ou seja, todas as componentes espectrais sofrem o mesmo ganho e a alteração de fase é linear). Existem duas maneiras de definir este parâmetro, dependendo do nível de correlação entre as frequências que abrangem a banda total.

Considerando correlação de 90%, tem-se:

$$B_c \cong \frac{1}{50\sigma_\tau} \quad (4)$$

Por outro lado, para correlação de 50%, tem-se:

$$B_c \cong \frac{1}{5\sigma_\tau} \quad (5)$$

Até então, os parâmetros descritos estão relacionados com a dispersão no tempo do canal de propagação em um local. Entretanto, sabe-se que, no domínio da frequência, também existe dispersão espectral no canal. Por este motivo, dois parâmetros adicionais caracterizam esta dispersão, a saber: espalhamento Doppler e tempo de coerência.

Antes de caracterizar o espalhamento Doppler, é interessante caracterizar o chamado efeito Doppler. Este efeito ocorre devido à mudança aparente de frequência de uma onda causada pelo movimento relativo entre a fonte que emite e o observador considerado. No contexto de sistemas de comunicação sem fio, a frequência do sinal transmitido pode ser alterada (diminuída ou aumentada) dependendo do movimento entre um usuário e sua estação rádio base (ERB).

Sendo assim, o espalhamento Doppler (B_D) mede o alargamento espectral causado pela taxa de tempo da mudança do canal e é definido como a faixa de frequências sobre a qual o espectro Doppler recebido é essencialmente diferente de zero. A Figura 2.2 exibe o espectro Doppler, onde f_m é o máximo desvio Doppler e pode ser calculado com o auxílio de (6).

$$f_M = \frac{v}{\lambda} \cos(\theta) \quad (6)$$

Onde v representa a velocidade da estação móvel (considerada constante), λ é o comprimento de onda do sinal considerado e θ é o ângulo espacial entre a direção do movimento da estação móvel e a direção da chegada da onda. Para o máximo desvio Doppler, pode-se considerar $\theta = 0^\circ$, de tal forma que:

$$f_m = \frac{v}{\lambda} \cos(0^\circ) = \frac{v}{\lambda}$$

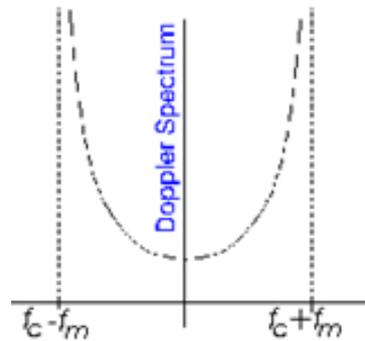


Figura 2.2: Espectro Doppler (LINNARTZ).

Outro parâmetro interessante é o tempo de coerência (T_c), que está relacionado, basicamente, com o intervalo de tempo em que a resposta ao impulso do canal de propagação permanece invariante. Em outras palavras, o tempo de coerência é a máxima separação de tempo entre dois sinais sendo que estes apresentam forte correlação com respeito aos efeitos de desvanecimento. Na prática, define-se este parâmetro como (RAPPAPORT, 2008):

$$T_c = \sqrt{\frac{9}{16\pi f_m^2}} \cong \frac{0,423}{f_m} \quad (7)$$

O interesse em determinar todos os parâmetros citados é o de classificar o tipo de desvanecimento que o sinal transmitido sofre ao passar pelo canal de transmissão.

Iniciando esta discussão, os canais de propagação podem ser classificados por apresentarem desvanecimento plano em frequência ou desvanecimento seletivo em frequência. Na primeira destas duas condições, conforme ilustrado na Figura 2.3, o canal possui ganho constante e resposta de fase linear em uma largura de banda maior do que a do sinal transmitido, de tal maneira que as características espectrais deste sinal são preservadas no receptor. Todavia, a intensidade do sinal recebido varia com o tempo, devido a flutuações no ganho do canal causadas pelos

multipercursos. Em termos dos parâmetros de dispersão analisados até então, tem-se que, no caso de desvanecimento plano em frequência:

$$\begin{aligned} B_S &\ll B_C \\ T_S &\gg \sigma_\tau \end{aligned} \quad (8)$$

Onde B_S e T_S representam a largura de banda do sinal e o período de símbolo correspondente. Para aplicações práticas, considera-se que um valor de T_S dez vezes maior do que σ_τ é apropriado para a análise de (8).

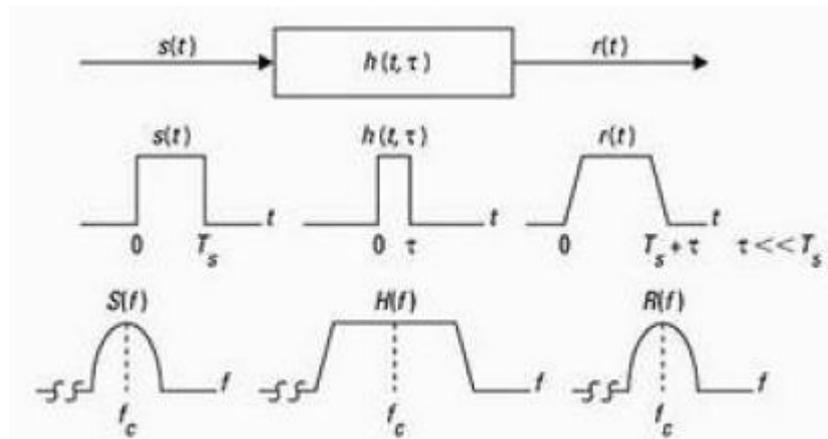


Figura 2.3: Desvanecimento plano em frequência (RAPPAPORT, 2008).

O desvanecimento seletivo em frequência ocorre quando o canal possui uma resposta de fase de ganho constante e linear em uma largura de banda menor do que a do sinal transmitido. Nesta condição, a resposta ao impulso do canal possui σ_τ maior do que o período de símbolo do sinal. Sendo assim, o sinal recebido é composto de várias versões atrasadas e atenuadas do sinal original. Além disto, pode ocorrer dispersão temporal dos símbolos transmitidos, de tal forma a introduzir ISI, conforme ilustra a Figura 2.4. Em suma, tem-se:

$$\begin{cases} B_S > B_C \\ T_S < \sigma_\tau \end{cases} \quad (9)$$

Onde, novamente, utiliza-se uma aproximação prática de T_S dez vezes menor do que σ_τ em (9).

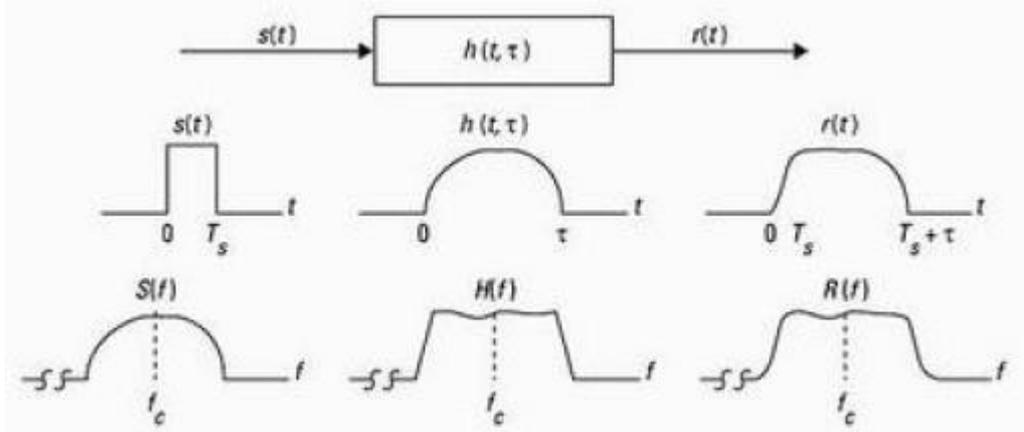


Figura 2.4: Desvanecimento seletivo em frequência (RAPPAPORT, 2008).

Para finalizar esta subseção, ainda é possível diferenciar o desvanecimento como rápido ou lento. Em um canal de desvanecimento rápido, a resposta ao impulso do canal muda dentro da duração de símbolo. Desta maneira, o desvanecimento é rápido se:

$$\begin{cases} B_S < B_D \\ T_S > T_C \end{cases} \quad (10)$$

Na prática, o desvanecimento rápido ocorre normalmente para taxas de transmissão baixas, dadas as características citadas.

Por fim, um desvanecimento é dito lento se a resposta ao impulso do canal muda muito mais lentamente que a duração do período de símbolo. Neste caso, pode-se considerar que o canal é estático por um ou mais períodos de símbolo. Neste caso:

$$\begin{cases} B_S \gg B_D \\ T_S \ll T_C \end{cases} \quad (11)$$

Para resumir esta discussão final, nota-se que, é possível combinar os tipos de desvanecimentos expostos nesta seção de maneira a identificar o comportamento do canal de propagação. Exemplificando, um canal com

desvanecimento seletivo em frequência e rápido implica que as fases, amplitudes e atrasos de tempo de qualquer um dos componentes de multipercurso variam mais rápido que a mudança do sinal transmitido.

Sendo assim, podem existir quatro tipos de combinações possíveis para o desvanecimento em um canal de propagação, conforme exposto na Figura 2.5, a seguir.

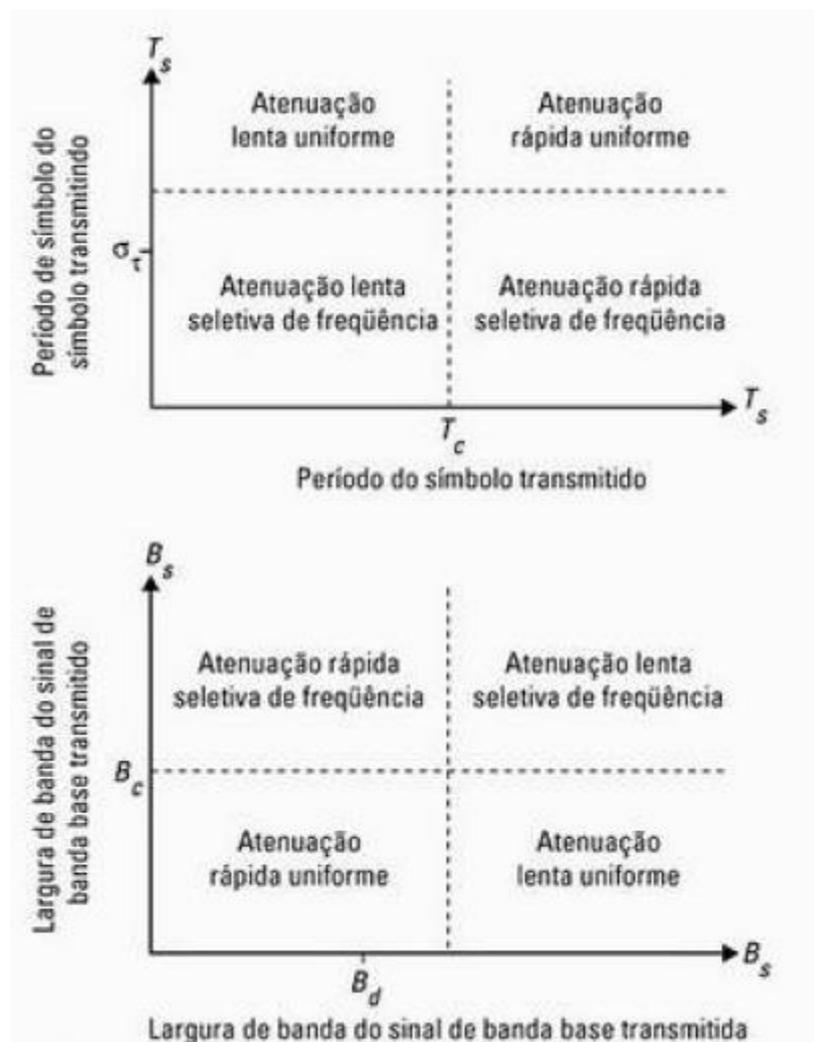


Figura 2.5: Tipos de desvanecimento em canais de propagação (RAPPAPORT, 2008).

Tendo discutido os tipos de desvanecimentos que afetam os canais de radiopropagação, é interessante notar que estes tipos de efeitos buscam estudar a atenuação em pequena escala do canal, isto é, rápidas flutuações de amplitude, fases e atrasos de multipercurso em um período de tempo pequeno ou por pequenas distâncias.

Um tipo comum de distribuição utilizada para descrever este tipo de atenuação em pequena escala é a distribuição de Rayleigh, cujo comportamento geral é exposto na Figura 2.6, a seguir. Nota-se que existem diversos momentos com grande atenuação do sinal (maior que 30dB). Nesta figura, assume-se que a frequência da portadora é de 900MHz.

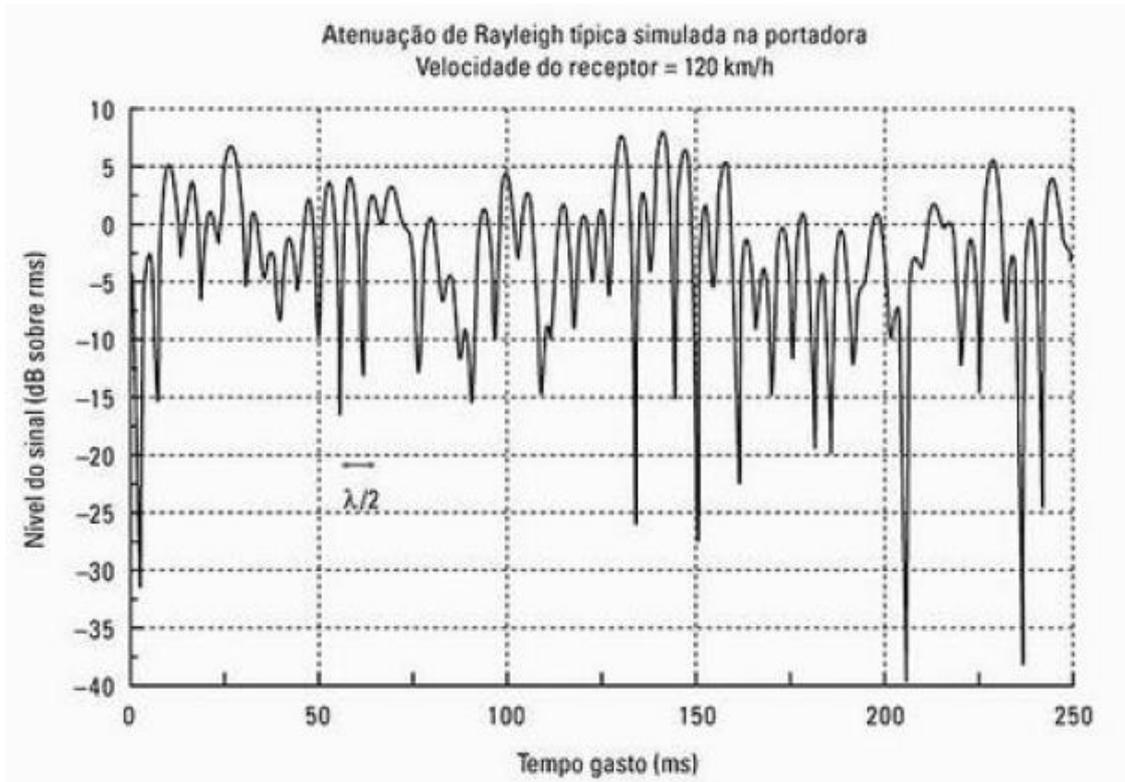


Figura 2.6: Comportamento da distribuição de Rayleigh no domínio temporal (RAPPAPORT, 2008).

Sabe-se que a distribuição de Rayleigh é obtida da soma do envelope de duas variáveis aleatórias (VAs) gaussianas em quadratura (vide Anexo A). Desta maneira, a função de densidade de probabilidade é dada por:

$$f_R(r) = \begin{cases} \frac{r}{\sigma^2} \exp\left(-\frac{r^2}{2\sigma^2}\right), & 0 \leq r \leq \infty \\ 0, & r < 0 \end{cases} \quad (12)$$

Onde r representa a amplitude da envoltória do sinal recebido e σ^2 é a potência média do sinal recebido. A probabilidade de que o envelope do sinal

recebido não ultrapasse um valor R é dada pela função de distribuição acumulada correspondente (RAPPAPORT, 2008):

$$P(R) = \Pr(r \leq R) = \int_0^R f_R(r) dr = 1 - \exp\left(-\frac{R^2}{2\sigma^2}\right) \quad (13)$$

Nas simulações de canais, geralmente quando se considera que não há visada direta entre o transmissor e o receptor, uma distribuição de Rayleigh é utilizada para simular o comportamento em pequena escala do canal.

Entretanto, é de se esperar que o sinal seja atenuado para grandes distâncias entre o transmissor e o receptor. Este comportamento é estudado pelos modelos de larga escala. Neste âmbito, existem diversos modelos teóricos e empíricos para estudo da intensidade média do sinal e que podem considerar, ou não os três mecanismos básicos de propagação estudados inicialmente.

Dois exemplos clássicos de modelos de larga escala são: Modelo de propagação no espaço livre e modelo de dois raios (que considera a reflexão no solo). No primeiro destes, obtém-se que a intensidade do sinal é proporcional ao inverso do quadrado da distância. Por outro lado, no modelo de dois raios, a intensidade do sinal é proporcional ao inverso da distância à quarta potência.

De maneira geral, pode-se considerar que a perda de percurso de larga escala é proporcional a um expoente de perda (η) que pode assumir diferentes valores, conforme ilustra a Tabela 2.1.

Tabela 2.1: Expoentes de perda de percurso para diferentes ambientes (RAPPAPORT, 2008).

Ambiente	Expoente de perda de percurso, η
Espaço Livre	2
Rádio-celular em área urbana	2,7 a 3,5
Rádio-celular urbano sombreado	3 a 5
Na linha de visão do prédio	1,6 a 1,8
Obstruído no prédio	4 a 6
Obstruído em fábricas	2 a 3

2.2. Técnica de OFDM

2.2.1 Modelagem matemática

Conforme citado anteriormente, a evolução dos sistemas de comunicação vem acompanhada pelo desejo de transmissão de dados em altas taxas. Considerando sistemas convencionais que utilizam apenas uma portadora (em inglês, *Single Carrier Systems* ou SCS), a taxa de símbolo (R_S) é inversamente proporcional a T_S .

Por este motivo, caso a taxa de transmissão seja muito elevada, T_S será muito pequeno. De acordo com a Seção 2.1, em canais de propagação com efeitos de multipercurso, a condição de ISI afetaria gravemente o sinal de mensagem, já que o período de símbolo seria pequeno quando comparado a σ_τ . Para solucionar problemas deste tipo, os sistemas deveriam utilizar equalizadores complexos em cada um dos receptores, comprometendo seu desempenho e elevando o custo total.

Sendo assim, técnicas como OFDM buscam transmitir, paralelamente, os dados desejados, de tal maneira a aumentar T_S e reduzir os potenciais problemas relacionados ao desvanecimento seletivo em frequência.

A ideia principal de transmissão utilizando OFDM é dividir a banda total (B) em subcanais de frequência menores e igualmente espaçados. Cada um destes subcanais é modulado de forma individual e transmitido simultaneamente e paralelamente. Desta maneira, a alta taxa de transmissão por portadora única é dividida em vários sinais de taxas menores, cuja transmissão não será (ou será muito menos) prejudicada pelo desvanecimento seletivo em frequência.

Um sinal OFDM consiste de N subportadoras adjacentes e ortogonais, com espaçamento Δf . Todos os sinais destas subportadoras são mutualmente ortogonais dentro do período de símbolo OFDM T , se ele for escolhido como $T = 1/\Delta f$. Sendo assim, este intervalo de tempo de símbolo é muito maior do que σ_τ , o que faz com que os efeitos da ISI sejam drasticamente reduzidos (ROHLING, 2011). Por outro lado, N não pode ser escolhido arbitrariamente, visto que um período de símbolo muito longo faz com que a transmissão se torne muito sensível às variações temporais do canal, que estão relacionadas ao efeito Doppler (discutido brevemente na Seção 2.1).

Supondo um bloco de símbolos sendo transmitidos serialmente, ao realizar a conversão para dados paralelos, tem-se que o período de símbolo OFDM resultante é $T = NT_S$. A Figura 2.7 mostra que, com esta condição, a taxa de transmissão de cada subcanal diminui consideravelmente dependendo do número de portadoras utilizadas, ou seja, vale $1/(NT_S)$.

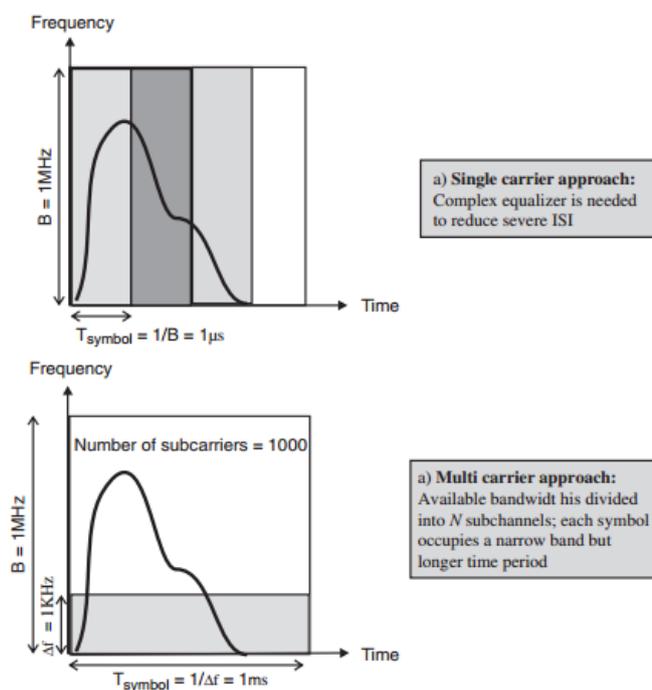


Figura 2.7: Conceito de transmissão de dados em paralelo (TAROKH, 2009).

Levando em consideração a discussão realizada nos parágrafos anteriores, nota-se que sistemas que utilizam múltiplas portadoras podem ser esquematizados de acordo com a Figura 2.8, exposta a seguir.

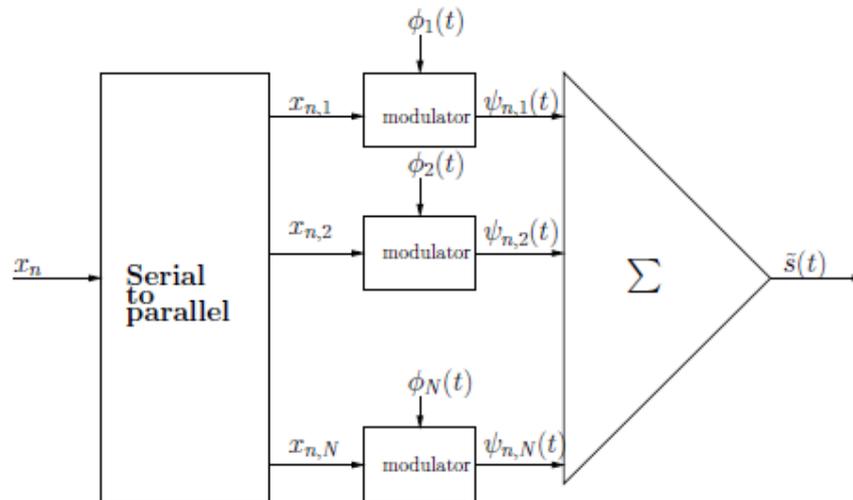


Figura 2.8: Modulação por múltiplas portadoras (RETHNAKARAN e DAWID, 2003).

Analisando a figura anterior, é possível constatar que o sinal de informação é convertido de serial para paralelo e modulado pelas N subportadoras correspondentes, denotadas por $\phi_1(t), \phi_2(t), \dots, \phi_N(t)$. No caso da OFDM, tem-se que as portadoras são ortogonais entre si no domínio da frequência, ou seja (CASELLA, 2015):

$$\int_{-\infty}^{\infty} \phi_i(f) \phi_k^*(f) df = 0, \quad \text{se } j \neq k, \quad (14)$$

$$\int_{-\infty}^{\infty} |\phi_i(f)|^2 df = k_j, \quad \text{se } j = k, \quad (15)$$

Desta maneira, mesmo que o espectro de frequências possua sobreposição, se o controle de frequência for realizado adequadamente, as subportadoras serão ortogonais entre si. Além disto, é possível que as subportadoras sejam ortogonais no domínio do tempo, o que não será abordado com detalhes nesta seção. Na prática, o conjunto de funções $\phi_k(t) = \{e^{j2\pi kt/T}\}$, onde $0 \leq k \leq N$, também pode ser utilizado para manter a ortogonalidade temporal das subportadoras (RETHNAKARAN e DAWID, 2003).

Observando a Figura 2.8 e considerando como funções base o conjunto $\phi_k(t)$ citado no parágrafo anterior, sabe-se que o envelope complexo de um sinal OFDM genérico é dado por:

$$\tilde{s}(t) = \sum_n b(t - nT, x_n) \quad (16)$$

Onde:

$$b(t, x_n) = \sum_{k=0}^{N-1} x_{n,k} e^{\frac{j2\pi(k-\frac{N-1}{2})t}{T}} \quad (17)$$

Nesta notação, \mathbf{x}_n representa um bloco de entrada de símbolos M-ários de índice n e pode ser representado como o conjunto de coeficientes $x_{n,k}$ conforme ilustrado na Figura 2.9, ou seja:

$$\mathbf{x}_n = \{x_{n,0}, x_{n,1}, \dots, x_{n,N-1}\} \quad (18)$$

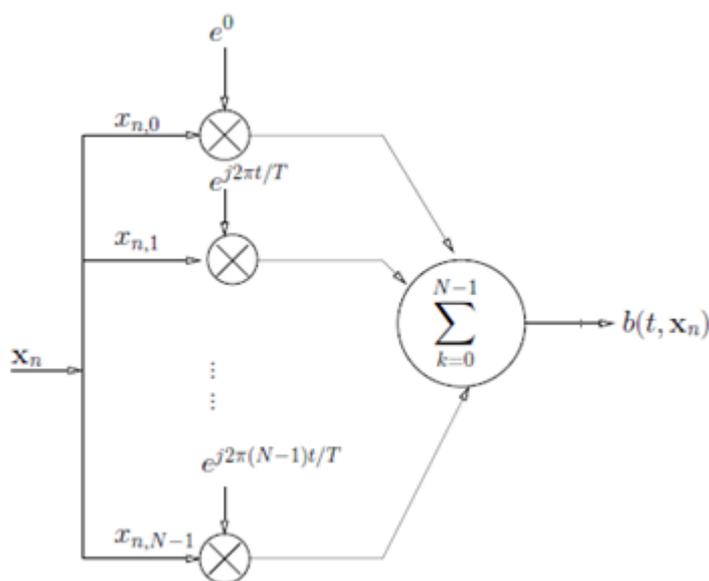


Figura 2.9: Modulação OFDM para um bloco de dados específico \mathbf{x}_n (RETHNAKARAN e DAWID, 2003).

Vale ressaltar que os símbolos $x_{n,k}$ são complexos e geralmente representam constelações relacionadas à modulação de amplitude em quadratura (em inglês, *quadrature amplitude modulation* ou QAM) ou à modulação por deslocamento de fase (em inglês, *phase-shift keying* ou PSK).

Expandindo a Equação (16) utilizando os conceitos expostos em (17), é possível notar que o envelope complexo de um sinal OFDM é dado por (RETHNAKARAN e DAWID, 2003):

$$\tilde{s}(t) = \sum_{n=-\infty}^{\infty} \underbrace{\sum_{k=0}^{N-1} x_{n,k} e^{\frac{j2\pi(k-\frac{N-1}{2})t}{NT_s}}}_{\text{Equivalente passa-baixa } s_b(t)} \quad (19)$$

Onde $T = NT_s$.

Além disto, o sinal passa-faixa pode ser escrito como:

$$s(t) = \text{Re}[\tilde{s}(t)e^{j2\pi f_c t}] \quad (20)$$

Observando apenas a parte relacionada ao equivalente passa-baixa em (19), nota-se que, removendo o termo em comum $(N-1)/2$ de atraso fixo e observando apenas um dos blocos OFDM transmitidos, sem perda de generalidade, tem-se que:

$$s_b(t) = \sum_{k=0}^{N-1} x_k e^{\frac{j2\pi kt}{NT_s}} \quad (21)$$

Na prática, a técnica de OFDM é implementada através da IDFT (em inglês, *Inverse Discrete Fourier Transform* ou IDFT), cuja aplicação é bastante otimizada considerando os avanços nos algoritmos de FFT (em inglês, *Fast Fourier Transform*).

De acordo com (RETHNAKARAN e DAWID, 2003), amostrando (21) em intervalos regulares de $1/T_s$, ou seja, respeitando-se a taxa de Nyquist para posterior reconstrução do sinal, tem-se:

$$X_m = \sum_{k=0}^{N-1} x_k e^{\frac{j2\pi km}{N}} \quad (22)$$

Onde $m = 0, 1, 2, \dots, N-1$ e (22) representa a IDFT de x_k .

No início desta subseção, foi exposto que uma grande vantagem da técnica de OFDM é a de eliminar a ISI que ocorre em sistemas com apenas uma portadora, transmitindo em altas taxas. Isto é feito com a inserção de um intervalo de guarda entre os blocos de dados, de tal forma a garantir que o efeito de multipercurso não influencie no sinal modulado.

A seguir, uma análise matemática breve é realizada a respeito do intervalo de guarda. A ideia geral foi retirada de (RETHNAKARAN e DAWID, 2003) e uma análise mais detalhada pode ser encontrada nesta dissertação.

Considerando um canal de transmissão discreto e linear, a relação entre os sinais transmitido e recebido é dada pela convolução do sinal transmitido com a resposta do canal, ou seja:

$$Y_n = \sum_{m=0}^{L-1} h_m X_{n-m} = h_n * X_n \quad (23)$$

Onde Y_n representa o sinal recebido, h_m a resposta ao impulso do canal e X_n o sinal OFDM dado por (22). Além disto, L representa o tamanho da janela de convolução, onde a resposta do canal é não nula (portanto, assume-se que L é maior do que o *delay spread* do canal de transmissão).

Expandindo (23) e utilizando a notação Y_n^k , que denota o n -ésimo símbolo do k -ésimo bloco, tem-se:

$$Y_0^k = \sum_{m=0}^{L-1} h_m X_{-m}^k = h_0 X_0^k + \underbrace{\sum_{m=1}^{L-1} h_m X_{L-m}^{k+1}}_{\text{ISI para o primeiro símbolo}} \quad (24)$$

$$Y_1^k = \sum_{m=0}^{L-1} h_m X_{1-m}^k = \sum_{m=0}^1 h_m X_{1-m}^k + \underbrace{\sum_{m=2}^{L-1} h_m X_{L+1-m}^{k+1}}_{\text{ISI para o segundo símbolo}} \quad (25)$$

Prosseguindo com a análise, é possível constatar que:

$$Y_{L-1}^k = \sum_{m=0}^{L-1} h_m X_{L-1-m}^k \quad (26)$$

Ou seja, o símbolo $L-1$ não possui ISI, o que é desejado. Por outro lado, todos os outros símbolos deste bloco sofrem com este fenômeno. Isto sugere que, para eliminar este problema para os primeiros $L-2$ símbolos de cada bloco, $L-1$ símbolos de guarda (zeros) devem ser inseridos (basta observar que, se L zeros são inseridos, a ISI é completamente eliminada em (24) e (25), assim como para os outros símbolos subsequentes).

No entanto, outro problema que existe em sistemas OFDM é a interferência entre portadoras (do inglês, *intercarrier interference*, ou ICI). Neste âmbito, utilizar zeros no intervalo de guarda elimina a ISI, mas não a ICI, visto que esta última ocorre devido à perda de ortogonalidade entre as subportadoras.

Por este motivo, se o intervalo de guarda é substituído por uma cópia dos dados do final de cada bloco, de tal forma que estes sejam repetidos inicialmente, tem-se o que se chama de prefixo cíclico (em inglês, *cyclic prefix* ou CP). É muito importante ressaltar que, ao utilizar um intervalo de guarda qualquer (zeros ou CP, por exemplo), a taxa de transmissão total do sistema é reduzida, em detrimento das vantagens alcançadas.

Com a inserção de uma cópia dos dados do final do bloco no início deste, a Equação (26) é substituída por uma convolução circular, ou seja, sendo X_n^g o sinal a ser transmitido com CP, tem-se:

$$Y_n^g = \sum_{m=0}^{L-1} h_m X_{n-m}^g \quad (27)$$

No receptor, o CP é removido seguindo uma regra simples e dada por (RETHNAKARAN e DAWID, 2003):

$$Y_n = Y_{N_g + \text{mod}(n - N_g, N)}^g \quad (28)$$

Onde *mod* indica a operação de módulo (resto da divisão) e N_g representa o número de amostras equivalentes ao CP, sendo escolhido tal que $N_g \geq L - 1$ (para permitir que toda a janela se encaixe no intervalo considerado). Sendo assim, quando esta relação é respeitada, tem-se:

$$Y_n = \sum_{m=0}^{L-1} h_m X_{\text{mod}(n-m, N)} = h_n \circledast X_n \quad (29)$$

Que é a convolução circular entre o sinal OFDM e a resposta ao impulso do canal de transmissão. Uma propriedade muito interessante da convolução circular, combinada com a operação de DFT, é que:

$$DFT(a \circledast b) = DFT(a) \times DFT(b) \quad (30)$$

Utilizando (29), uma análise matemática simplificada pode ser realizada com respeito à inserção do CP. Através de uma regra simples, este é aplicado ao sinal transmitido, ou seja:

$$X_n^g = X_{\text{mod}(n+N-N_g, N)}, \quad n = 0, 1, \dots, N + N_g - 1 \quad (31)$$

Ao passar pelo canal de transmissão, tem-se a relação já exposta em (28). Por fim, estes são removidos de tal forma que:

$$Y_n = Y_{N_g+n}^g, \quad n = 0, 1, \dots, N - 1 \quad (32)$$

Em suma, a aplicação do CP como intervalo de guarda permite, além de eliminar a ISI, também acabar com o problema da ICI, citado anteriormente. A Figura 2.10 mostra uma ilustração do funcionamento do prefixo cíclico, em comparação com um intervalo de guarda composto apenas por zeros.

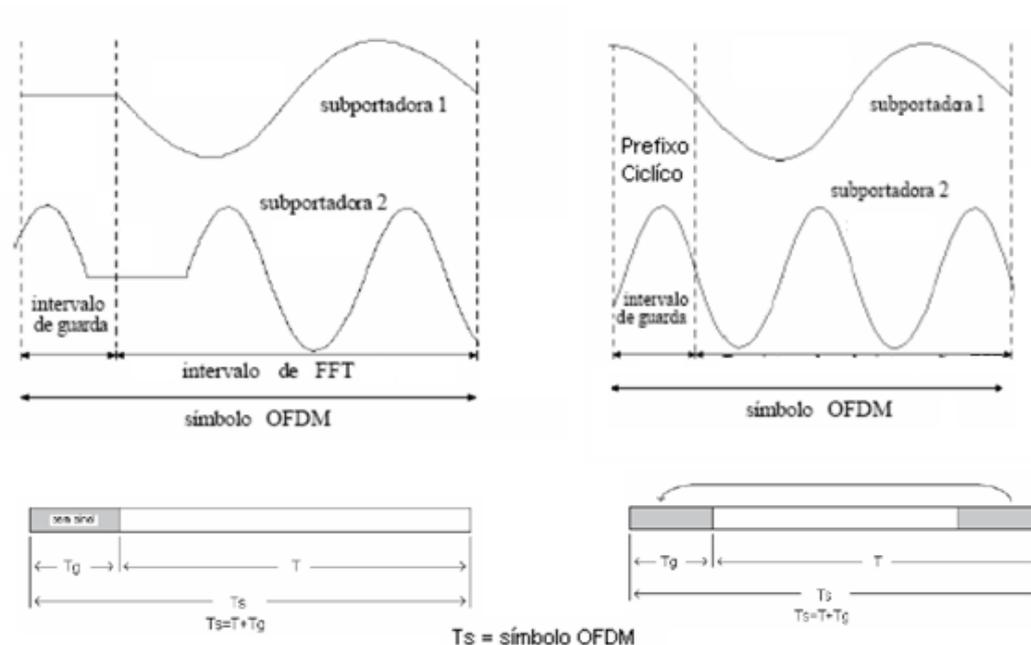


Figura 2.10: Prefixo cíclico na técnica de OFDM.

Resumindo a análise matemática exposta neste capítulo, a Figura 2.11 expõe o diagrama de blocos geral da técnica de OFDM em uma transmissão de dados.

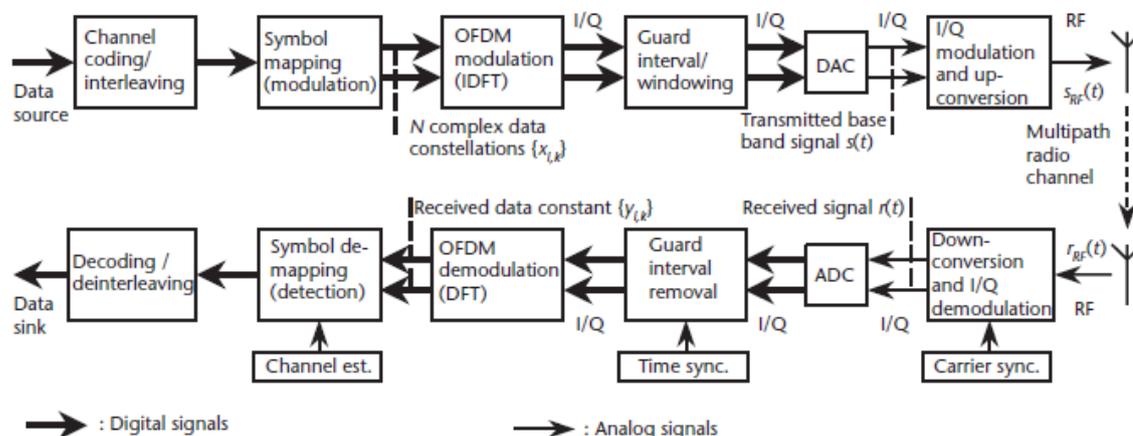


Figura 2.11: Diagrama de blocos geral de um sistema OFDM (PRASAD, 2004).

A observação do diagrama anterior permite constatar que uma etapa de codificação de canal pode existir, seguida do mapeamento de símbolos e da técnica de OFDM, no transmissor. Em seguida, podem ser aplicados o intervalo de guarda e o janelamento (para possibilitar ortogonalidade temporal). Por fim, após passar por um conversor digital/análogo, o sinal é modulado e transmitido pelo canal de propagação (sinal banda-passante).

No receptor, o processo inverso é realizado. Sendo assim, é realizada a operação de DFT no demodulador OFDM, cujo cálculo é dado por:

$$\hat{x}_k = \sum_{m=0}^{N-1} Y_k e^{\frac{-j2\pi km}{N}}, \text{ com } m = 0, 1, \dots, N - 1 \quad (33)$$

Vale notar que (22) e (33) podem ser normalizadas para sua aplicação prática, ou seja, basta multiplicar estas equações pelo termo $1/\sqrt{N}$.

Para finalizar esta discussão, vale notar que, quando a técnica de OFDM é utilizada para aplicações em banda base, o vetor de sinais aplicado à IFFT deve ser simétrico hermetiano, de tal forma que, após o cálculo da IFFT, a saída resultante seja composta de números reais, o que possibilita a transmissão pelo canal passa-baixa.

Mesmo com a eliminação da ISI e da ICI, utilizando CP, outra característica importante que pode ser incorporada ao projeto é a transmissão de subportadoras piloto, com o intuito de estimar a resposta em frequência do canal. Isto é feito pois, em cada subcanal, ainda há desvanecimento plano em frequência, o que faz com que ocorram mudanças na magnitude e fase do sinal transmitido.

Por fim, quando se considera apenas um modelo banda-base do sistema OFDM, conforme ilustra a Figura 2.12, uma breve análise mostra que o i -ésimo símbolo OFDM recebido, sem a presença de ISI/ICI (com o correto dimensionamento do prefixo cíclico) é dado por:

$$\mathbf{Y}_i = DFT_N\{IDFT_N(\mathbf{X}_i) \odot \mathbf{h}_i + \mathbf{w}_i\} \quad (34)$$

Onde \mathbf{Y}_i , \mathbf{X}_i , \mathbf{h}_i e \mathbf{w}_i representam o símbolo OFDM recebido, o símbolo OFDM transmitido, a resposta do canal e a contribuição do ruído AWGN (do inglês *Additive White Gaussian Noise*). Vale ressaltar que todas as representações são vetores ($N \times 1$).

Desta maneira, denotando \odot como multiplicação de matrizes ponto a ponto, o símbolo OFDM recebido é dado por:

$$\mathbf{Y}_i = \mathbf{X}_i \odot DFT_N\{\mathbf{h}_i\} + DFT_N\{\mathbf{w}_i\} = \mathbf{X}_i \odot \mathbf{H}_i + \mathbf{W}_i \quad (35)$$

Na expressão anterior, observa-se que o símbolo OFDM é dado pela multiplicação ponto a ponto do sinal transmitido com a resposta do canal, com a contribuição do ruído aditivo.

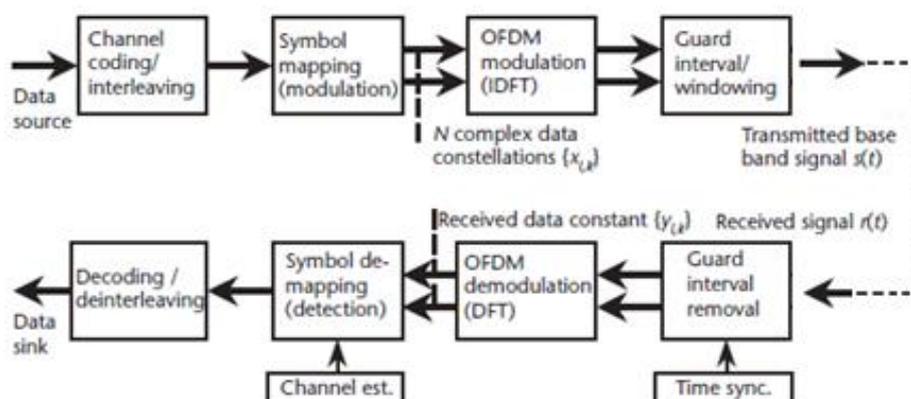


Figura 2.12: Modelo banda-base de um sistema OFDM (PRASAD, 2004).

As principais etapas de um sistema OFDM serão analisadas posteriormente. Contudo, é importante ressaltar que as etapas de codificação de canal e embaralhamento (*interleaving*) buscam melhorar o desempenho do sistema frente a erros de bit.

2.2.2 Deficiências de sistemas OFDM

Até então, foram exaltados os pontos fortes de sistemas OFDM com relação ao desvanecimento seletivo em frequência. Entretanto, vale ressaltar que a técnica de OFDM é baseada na ortogonalidade entre as subportadoras, o que permite transmissão de dados em paralelo sem que ocorra ISI. Se a ortogonalidade é perdida, entretanto, este efeito começa a ocorrer em conjunto com a ICI, ocasionando perda de desempenho, o que torna o sincronismo essencial em sistemas OFDM.

Apesar de não apresentar a modelagem matemática, as deficiências de sistemas OFDM serão brevemente comentadas nesta subseção. Uma análise mais detalhada pode ser encontrada em (ZIVKOVIC, 2013).

Três efeitos que afetam a ortogonalidade entre as subportadoras OFDM e que podem ser citados são (ZIVKOVIC, 2013):

- *Symbol Time Offset*: No receptor, para que a DFT seja realizada, as mesmas amostras submetidas à IDFT no transmissor precisam estar alinhadas corretamente na janela. Se isto não ocorrer, a operação de DFT poderá considerar partes de outros símbolos, o que acarreta em ISI. Por este motivo, os erros de sincronismos temporais relativos a este problema devem ser pequenos quando comparados aos intervalos de guarda.
- *Carrier Frequency Offset*: Este problema pode ser ocasionado simplesmente por erros de frequência entre os osciladores do transmissor e do receptor ou pelo efeito Doppler (movimento entre transmissor e receptor).
- *Sampling Frequency Offset*: Ocorre quando a frequência do relógio do conversor digital-analógico do transmissor e o relógio do conversor analógico-digital do receptor são diferentes. Com isto, a amostragem dos sinais ocorre em tempos diferentes, o que pode até levar a perda de informação.

Conforme citado, a análise matemática dos efeitos citados não faz parte do escopo deste trabalho, mas é válido ressaltar que a técnica de OFDM possui pontos que devem ser levados em consideração no momento do projeto do sistema, de tal forma que o desempenho geral não seja prejudicado demasiadamente.

Outro problema encontrado em sistemas OFDM é o alto valor da razão entre a potência de pico e a potência média do sinal (*Peak-to-Average Power Ratio* ou PAPR). Isto ocorre pois o sinal OFDM no domínio do tempo é a soma de N subportadoras com diferentes frequências, o que acarreta em momentos onde há alta contribuição desta soma para o valor de pico do sinal em relação à média deste. É possível mostrar que a PAPR depende do tipo de modulação utilizado, bem como do número de subportadoras OFDM (LI e STÜBER, 2006). Deste modo, o intervalo para o qual um amplificador de potência é linear pode não ser respeitado, causando distorções no sinal.

Sendo assim, soluções propostas para PAPR em OFDM podem ser impostas tanto no sistema quanto na procura de técnicas que aumentem o intervalo de linearidade para amplificadores de potência comerciais (LI e STÜBER, 2006).

2.2.3 Modulações utilizadas com OFDM

Na seção anterior, foi visto que uma das etapas de um sistema OFDM é a de modular o sinal de mensagem. Em âmbito geral, um sistema de comunicação passa-faixa pode ser representado pelo componente relacionado à portadora e o equivalente passa-baixa, conforme comentado na seção anterior para a análise do sistema OFDM. De modo geral, tem-se o comportamento da expressão (20).

Também foi exposto, anteriormente, que duas constelações comumente utilizadas com OFDM são M -PSK e M -QAM. Por este motivo, nesta subseção, estes dois tipos de modulações serão revisados. Vale ressaltar que M representa o número de símbolos possíveis (ou ordem da modulação), sendo que o número de bits/símbolo é dado por $\log_2 M$.

2.2.3.1 Modulação M -ária por deslocamento de fase (M -PSK)

Este tipo de modulação possui como característica principal a transmissão de informação por variações de fase da portadora (de maneira coerente ou não-coerente). A representação para o diagrama de constelação pode ser obtida pelo método de Gram-Schmidt. Apenas uma análise breve será realizada a seguir e uma análise detalhada pode ser obtida em (LATHI e DING, 2009).

De maneira geral, o sinal M -PSK pode ser representado por:

$$s_i(t) = A \cos(\omega_0 t - \theta_i), \quad 0 \leq t \leq T_s \quad (36)$$

Onde $\theta_i = 2\pi(i - 1)/M$, $i = 1, 2, \dots, M$.

Para sinais senoidais, tem-se que a amplitude é relacionada à potência da seguinte maneira:

$$P = \frac{A^2}{2} \rightarrow E_s = PT_s = \frac{A^2 T_s}{2} \rightarrow A = \sqrt{\frac{2E_s}{T_s}} \quad (37)$$

Onde E_s representa a energia de símbolo. Desta maneira, utilizando as informações expostas anteriormente, tem-se:

$$s_i(t) = \sqrt{\frac{2E_s}{T_s}} \cos\left(\omega_0 t - \frac{2\pi(i-1)}{M}\right) \quad (38)$$

Utilizando o método de Gram-Schmidt, uma base ortonormal para representação destes sinais é dada por:

$$\begin{cases} \phi_1(t) = \sqrt{\frac{2}{T_s}} \cos(\omega_0 t) \\ \phi_2(t) = \sqrt{\frac{2}{T_s}} \sin(\omega_0 t) \end{cases}, 0 \leq t \leq T_s \quad (39)$$

Ou seja, o sinal $s(t)$ pode ser escrito da seguinte forma:

$$s(t) = z_1(t)\sqrt{E_s} \phi_1(t) + z_2(t)\sqrt{E_s} \phi_2(t) \quad (40)$$

Onde $\{z_1(t), z_2(t)\} \in \left\{ \cos\left(\frac{2\pi(i-1)}{M}\right), \sin\left(\frac{2\pi(i-1)}{M}\right) \right\}, i = 1, \dots, M$. Desta forma, diagramas de constelação típicos para este tipo de modulação são expostos na Figura 2.13, a seguir.

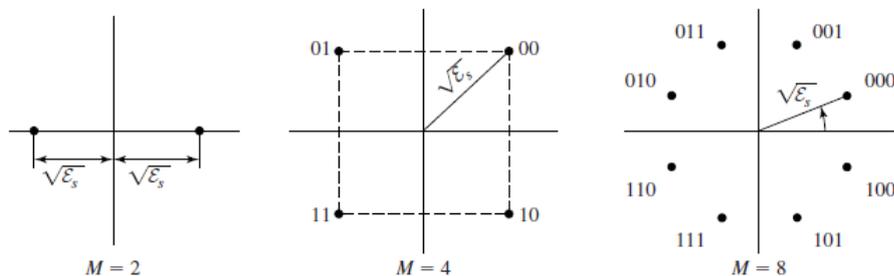


Figura 2.13: Constelações M -PSK representadas na base ortonormal (PROAKIS e SALEHI, 2002).

A análise de desempenho frente ao ruído para a constelação M -PSK resulta na seguinte expressão de probabilidade de erro de símbolo (P_e), cuja demonstração detalhada pode ser encontrada em (LATHI e DING, 2009):

$$P_e \approx 2Q \left(\sqrt{\frac{2E_s}{N_0}} \operatorname{sen} \left(\frac{\pi}{M} \right) \right) \quad (41)$$

Onde N_0 representa a densidade espectral de potência do ruído. Se o sistema utiliza codificação Gray para arranjar os M símbolos possíveis, sabe-se que a probabilidade de erro de bit (P_b) é dada por (LATHI e DING, 2009):

$$P_b \approx \frac{P_e}{\log_2 M} \quad (42)$$

2.2.3.2 Modulação M -ária de amplitude em quadratura (M -QAM)

Este tipo de modulação possui como característica principal a transmissão de informação por variações tanto na fase quanto na amplitude da portadora. De maneira geral, o sinal M -QAM pode ser representado por:

$$s_i(t) = A_i \cos(\omega_0 t - \theta_i), \quad 0 \leq t \leq T_s \quad (43)$$

Desta forma, tem-se:

$$s_i(t) = \sqrt{\frac{2E_i}{T_s}} \cos(\omega_0 t - \theta_i) \quad (44)$$

Utilizando a mesma base ortonormal apresentada para o caso M -PSK, o sinal $s(t)$ pode ser representado como:

$$s(t) = z_1(t) \phi_1(t) + z_2(t) \phi_2(t) \quad (45)$$

Onde $\{z_1(t), z_2(t)\} \in \{\sqrt{E_i} \cos(\theta_i), \sqrt{E_i} \sin(\theta_i)\}$. Considerando o caso de constelações M -QAM quadradas e sendo E_{min} a mínima energia de todos os símbolos possíveis, tem-se:

$$s_i(t) = \sqrt{\frac{2E_{min}}{T_s}} a_i \cos(\omega_0 t) + \sqrt{\frac{2E_{min}}{T_s}} b_i \sin(\omega_0 t) \quad (46)$$

Com a_i e b_i sendo números inteiros escolhidos de acordo com a posição na constelação, ou seja, $\{a_i, b_i\} \in \{-\sqrt{M} + 1, -\sqrt{M} + 3, \dots, -\sqrt{M} - 1\}$, $i = 1, 2, \dots, \sqrt{M}$. Alguns diagramas de constelação típicos para M -QAM são expostos na Figura 2.14.

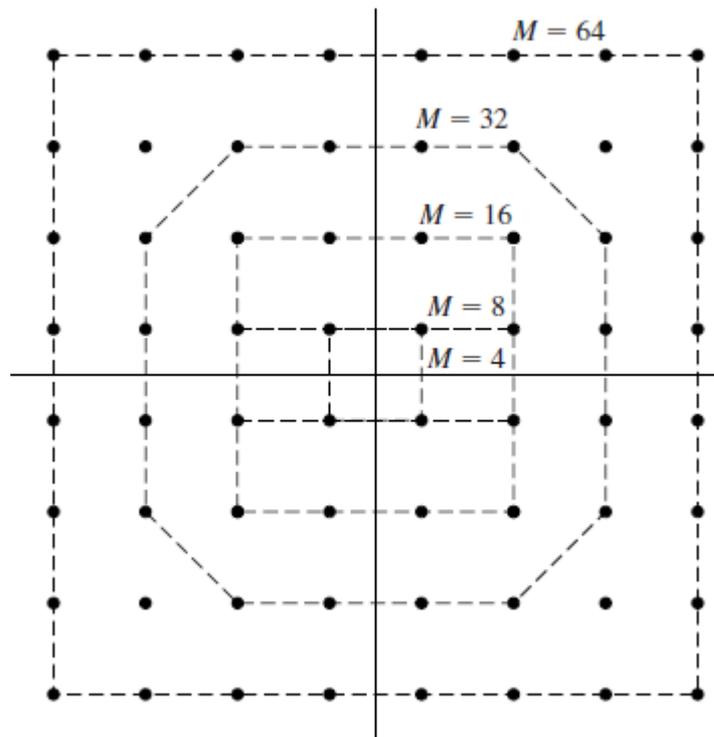


Figura 2.14: Constelações M -QAM representadas na base ortonormal (PROAKIS e SALEHI, 2002).

A expressão de probabilidade de erro de símbolo associada a este tipo de modulação é dada por (LATHI e DING, 2009):

$$P_e \approx \frac{4(\sqrt{M} - 1)}{\sqrt{M}} Q \left(\sqrt{\frac{3}{(M-1) N_0} \frac{E_s}{}} \right) \quad (47)$$

Assim como para o caso M -PSK, se codificação Gray é utilizada, a probabilidade de erro de bit é aproximadamente igual à expressão anterior dividida pelo número de bits/símbolo da modulação M -QAM correspondente.

2.3. Códigos corretores de erro

Tendo em vista a aplicação de OFDM em sistemas reais, onde a BER desejada é pequena, o desempenho pode ser melhorado com códigos corretores de erro.

Esta seção expõe, brevemente, os conceitos básicos relacionados a códigos corretores de erro, que desempenham um papel fundamental em sistemas OFDM. O foco principal é em códigos de repetição, codificação de Hamming e código convolucional. Geralmente, existem duas classes de códigos corretores de erro, a saber, códigos de bloco e códigos convolucionais.

Em códigos de bloco, um bloco de entrada de k dígitos passa por um codificador e é transformado em uma sequência maior de tamanho n . Desta maneira, cada sequência de k dígitos determina unicamente uma palavra de código de tamanho n . Por outro lado, em códigos convolucionais, a sequência codificada de tamanho n depende não apenas dos k dígitos do bloco de entrada atual, mas também de dígitos relacionados a blocos anteriores (ou seja, o codificador possui memória).

Considerando a notação do parágrafo anterior, a taxa de código é dada por $\rho = k/n$. Além disto, códigos de bloco são comumente denotados por (n, k) , enquanto códigos convolucionais são representados por seus polinômios geradores, na base octal.

De acordo com a teoria de Shannon de capacidade do canal, conforme dito anteriormente, se o sistema transmite a uma taxa menor do que a capacidade do canal, existem códigos que possibilitam BER arbitrariamente pequena. Entretanto, na prática, este resultado se traduz em um tamanho n grande para BER muito pequena e decodificadores com muita memória, conseqüentemente. Por este

motivo, a chave para o problema reside na tarefa de encontrar bons códigos corretores de erro com grandes valores de n , porém decodificadores simples de serem implementados. Toda a análise feita nas subseções posteriores foi retirada de (LATHI e DING, 2009).

2.3.1 Códigos de bloco lineares

Considerando uma sequência de bits (palavra) de entrada (\mathbf{d}) e sua respectiva palavra codificada (\mathbf{q}), estas podem ser representadas por vetores k e n -dimensionais, ou seja:

$$\begin{cases} \mathbf{d} = (d_1, d_2, \dots, d_k) \\ \mathbf{q} = (q_1, q_2, \dots, q_n) \end{cases} \quad (48)$$

Em um tipo específico de família de códigos denominada de códigos sistemáticos, pode-se ter $q_1 = d_1, q_2 = d_2, \dots, q_k = d_k$ e os m bits restantes da palavra codificada (q_{k+1} a q_n), denominados de bits de paridade, são formados por combinações lineares dos dígitos da palavra de entrada, ou seja:

$$\left\{ \begin{array}{l} q_1 = d_1 \\ q_2 = d_2 \\ \dots \\ q_k = d_k \\ q_{k+1} = r_{11}d_1 \oplus r_{12}d_2 \oplus \dots \oplus r_{1k}d_k \\ q_{k+2} = r_{21}d_1 \oplus r_{22}d_2 \oplus \dots \oplus r_{2k}d_k \\ \dots \\ q_n = r_{m1}d_1 \oplus r_{m2}d_2 \oplus \dots \oplus r_{mk}d_k \end{array} \right. \quad (49)$$

Onde $m = n-k$ e \oplus representa uma operação ou exclusivo (XOR). A representação matricial é dada por:

$$\mathbf{q} = \mathbf{dG} \quad (50)$$

Onde \mathbf{G} é chamada de matriz geradora do código e pode ser representada por:

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & r_{11} & r_{21} & \dots & r_{m1} \\ 0 & 1 & 0 & \dots & 0 & r_{12} & r_{22} & \dots & r_{m2} \\ & & & \ddots & & & & & \\ 0 & 0 & 0 & \dots & 1 & r_{1k} & r_{2k} & \dots & r_{mk} \end{bmatrix} \quad (51)$$

$\underbrace{\hspace{10em}}_{I_k(k \times k)} \quad \underbrace{\hspace{10em}}_{P(k \times m)}$

Nota-se que a matriz geradora pode ser particionada em uma matriz identidade ($k \times k$) e uma matriz \mathbf{P} ($k \times m$), cujos coeficientes valem 0 ou 1. Com esta notação, o vetor que identifica as palavras de código pode ser representado por:

$$\mathbf{q} = \mathbf{dG} = \mathbf{d}[I_k \mathbf{P}] = [\mathbf{d} \mathbf{dP}] = [\mathbf{d} \mathbf{q}_p] \quad (52)$$

Onde \mathbf{q}_p representa os bits de paridade. Desta maneira, sabendo a palavra de entrada, é possível de calcular os bits de paridade facilmente, se as relações representadas pela matriz \mathbf{P} são conhecidas.

Em um código de bloco linear, a operação XOR de cada par de palavras codificadas \mathbf{q}_a e \mathbf{q}_b do código também resulta em outra palavra deste código. É por este motivo que a palavra de código que possui todos os bits zeros é incluída neste tipo de código. Definindo o peso de um vetor como o número de 1s que este possui e a distância entre duas palavras de código como:

$$d(\mathbf{q}_a, \mathbf{q}_b) = \text{peso}(\mathbf{q}_a \oplus \mathbf{q}_b) \quad (53)$$

Tem-se que, em códigos de bloco lineares, a mínima distância entre duas palavras de código é igual ao mínimo peso entre as palavras de código.

Considerando a decodificação do código e tomando (52) como referência, tem-se que:

$$\mathbf{dP} \oplus \mathbf{q}_p = [\mathbf{d} \mathbf{q}_p] \begin{bmatrix} \mathbf{P} \\ I_m \end{bmatrix} = 0 \quad (54)$$

Onde na expressão anterior foi utilizado o fato de que a operação XOR entre dois valores iguais equivale a zero e I_m é uma matriz identidade $m \times m$. Então:

$$\mathbf{qH}^T = 0 \quad (55)$$

Já que $[d \ q_p] = q$ e $H^T = \begin{bmatrix} P \\ I_m \end{bmatrix}$. Mais especificamente, a matriz H é chamada de matriz de verificação de paridade. Como toda palavra de código deve satisfazer (55), se uma palavra recebida r_i é corrompida por erros de canal, tem-se:

$$r_i = q_i \oplus e_i \quad (56)$$

Onde e_i é uma palavra de erro e cada bit representa se há erro ou não no respectivo bit da palavra recebida. Se a palavra de erro fosse nula, então $r_i = q_i$ e seria possível escrever $r_i H^T = 0$. Em geral, o valor da expressão anterior não é nulo (pois existem erros associados à transmissão por um canal ruidoso). Sendo assim, define-se s como a síndrome de uma palavra recebida, ou seja:

$$s = r_i H^T = (q_i \oplus e_i) H^T = q_i H^T + e_i H^T = e_i H^T \quad (57)$$

Desta maneira, se a palavra r_i é recebida, a síndrome s é calculada para encontrar a palavra de erro e_i correspondente. Um problema é que o conhecimento da síndrome não define unicamente a palavra de erro. Isto ocorre pois r_i pode ser escrito em função de outras palavras de código q_j , com $j \neq i$. De fato, existem 2^k palavras de código possíveis para a mesma síndrome e isto é um problema no momento de decodificar a palavra recebida.

Para resolver o problema exposto no parágrafo anterior, o que se faz na prática é montar uma tabela de síndromes e associá-las aos seus respectivos erros por (57). Como o tamanho da palavra de síndrome é m , existem 2^m síndromes possíveis (e palavras de erro associadas) e, se a palavra de código é muito grande, a tabela começa a adquirir proporções de armazenamento equivalentemente grandes, o que pode dificultar o processo de decodificação no receptor do sistema.

Considerando um código corretor de erro simples, intuitivamente pode-se pensar que repetir o bit transmitido n vezes irá refletir em uma melhoria na recepção, se existem bits suficientes para que a escolha seja feita pela maioria. Para este tipo de código a matriz geradora G é dada por:

$$\mathbf{G} = \left[\underbrace{1 \quad 1 \quad \dots \quad 1}_{1 \times n} \right] \quad (58)$$

Na divisão realizada anteriormente para a matriz geradora, pode-se interpretar o primeiro termo da matriz como \mathbf{I}_1 e os demais termos como a matriz \mathbf{P} . Com isto, a matriz de verificação de paridade é dada por:

$$\mathbf{H}^T = \begin{bmatrix} 1 & 1 & \dots & \dots & 1 \\ 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ & & & \ddots & \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix} \quad (59)$$

Onde $m = n - 1$.

Por outro lado, supondo um código de bloco que consiga corrigir t bits errados, sabe-se que a relação conhecida como condição de Hamming (em inglês, *Hamming Bound*) é dada por:

$$2^m \geq \sum_{j=0}^t \binom{n}{j} \quad (60)$$

Esta condição é necessária (mas não suficiente) para que um código corrija t erros, no caso geral. Entretanto, para códigos que corrigem um erro ($t = 1$), sabe-se que esta condição é necessária e suficiente. Além disto, se a inequação anterior atinge a igualdade, diz-se que o código é perfeito.

O código de Hamming é um exemplo de código binário, perfeito e com $t = 1$. Assumindo que a distância mínima (d_{min}) entre as palavras de código seja igual a 3, tem-se:

$$2^m = \sum_{j=0}^1 \binom{n}{j} = 1 + n \rightarrow n = 2^m - 1 \quad (61)$$

Generalizando, códigos de Hamming possuem $n = 2^m - 1$ e $k = 2^m - m - 1$, onde $d_{min} = m$. Um exemplo clássico de código de Hamming possui $n = 7$, $k = 4$ e $d_{min} = 3$.

Neste caso, o tamanho da síndrome é de três bits, ou seja, existem oito casos possíveis (com um caso trivial onde não há erro). Desta maneira, existem sete padrões de erros que podem ser corrigidos e, contanto que todas as síndromes sejam diferentes, existe decodificação única entre as palavras de código e as palavras de mensagem.

Com as informações citadas, nota-se que a única restrição para a matriz de paridade de bits é a de que esta possua linhas que distintas e não nulas. Uma forma possível para o código clássico citado anteriormente é:

$$\mathbf{H}^T = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{P} \\ \mathbf{I}_3 \end{bmatrix} \quad (62)$$

A matriz geradora é dada por:

$$\mathbf{G} = [\mathbf{I}_4 \ \mathbf{P}] = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \quad (63)$$

Como citado inicialmente, nesta seção apenas os aspectos básicos de códigos corretores de erro lineares foram vistos, com base nos códigos de repetição e Hamming e considerando a notação matricial utilizada em (LATHI e DING, 2009). Na seção de resultados e discussões, o desempenho destes dois tipos de codificação será observado em canais com multipercurso.

2.3.2 Códigos convolucionais

Em códigos convolucionais, uma palavra de código de tamanho n gerada pelo codificador em um determinado instante de tempo depende não apenas da palavra de mensagem de tamanho k no mesmo instante, mas também dos bits em intervalos de tempo anteriores. Para distinguir a notação utilizada na seção anterior, a saída para a palavra codificada é denotada por \mathbf{v} .

Neste âmbito, um codificador convolucional com restrição de comprimento N_{Conv} consiste de um registrador de deslocamento de N_{Conv} estágios e ℓ operações XOR. Um comutador possibilita a obtenção das respostas das operações XOR em sequência, durante cada intervalo de bit. A Figura 2.15 mostra um exemplo onde $N_{Conv} = 3$ e $\ell = 2$.

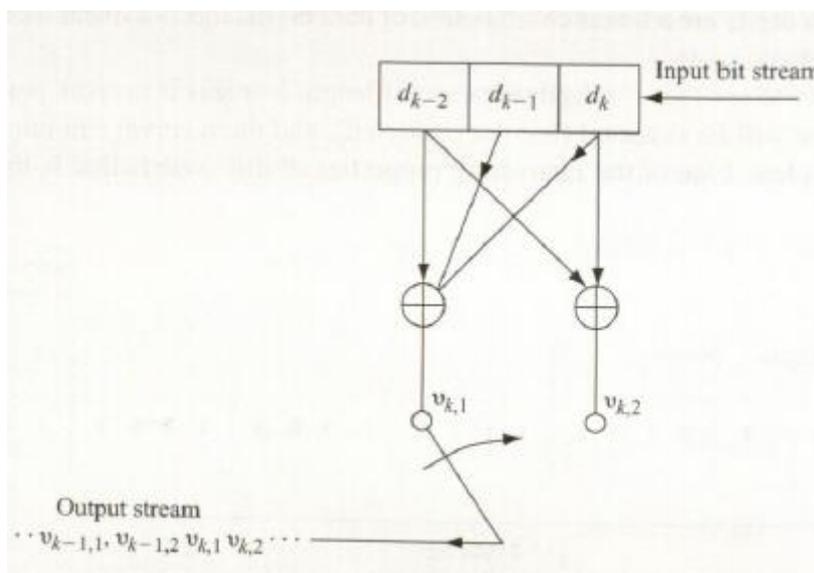


Figura 2.15: Exemplo de codificador convolucional com $N_{Conv} = 3$ e $\ell = 2$ (LATHI e DING, 2009).

Um ponto importante a ser ressaltado é que $N_{Conv} - 1$ zeros podem ser adicionados ao final da palavra de mensagem, de tal forma que o último bit alcance a última posição do registrador de deslocamento. De modo geral, tem-se $n = (N_{Conv} + k - 1) \ell$ bits na palavra de código. Também vale ressaltar que a estrutura da figura anterior não representa um caso geral e diversas combinações de codificadores convolucionais podem ser concebidas, variando-se os valores de N_{Conv} e ℓ . Além disto, a estrutura pode apresentar realimentação, gerando códigos convolucionais recursivos.

Uma das maneiras de representação de um codificador convolucional é por uma árvore de codificação, conforme ilustra a Figura 2.16 para o caso do codificador da Figura 2.15.

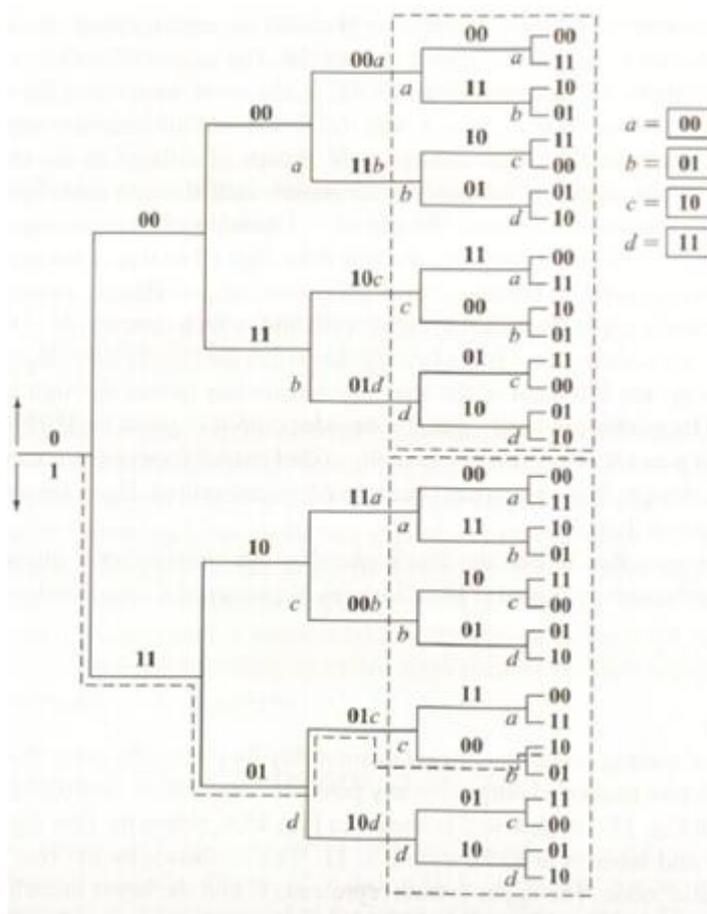


Figura 2.16: Representação do codificador da Figura 2.15 em árvore (LATHI e DING, 2009).

Observando a figura anterior, nota-se que esta representação mostra, passo a passo, a saída codificada dependendo do bit de entrada do codificador. Ao observar o primeiro passo, à esquerda, a escolha é realizada para cima ou para baixo se o primeiro bit a entrar no codificador é 0 ou 1, respectivamente. Nos próximos passos, adota-se a mesma notação. O exemplo mostrado remete à mensagem 11010 e é possível notar que a saída codificada é obtida simplesmente percorrendo os ramos da árvore para cima caso o próximo bit seja 0 ou para baixo caso o próximo bit seja 1.

Um fato importante é que a árvore começa a ficar repetitiva após certo momento. Para este caso, isto ocorre após o terceiro passo (“ramo”). Observando a estrutura da Figura 2.15, nota-se que quando o quarto bit entra no codificador, este para de influenciar os próximos bits de saída do comutador. De maneira geral, isto ocorre para a partir do bit $N_{Conv} + 1$. É por este motivo que os ramos são denominados *a*, *b*, *c* e *d* na Figura 2.16, de forma a reforçar esta repetição.

Outra maneira de representar um codificador convolucional é por um diagrama de máquina de estados. Considerando um bit de entrada, os $N_{Conv} - 1$ bits restantes do registrador de deslocamento possuem $2^{N_{Conv}} - 1$ possíveis estados associados. Por exemplo, para o caso do codificador tratado até o momento, os dois bits restantes poderiam ser: 00, 01, 10 ou 11.

Cada bit d_k entrando em um determinado instante de tempo no codificador ocasiona uma saída que depende do estado dos $N_{Conv} - 1$ bits restantes. Desta maneira, denotando cada transição do diagrama por $\{d_k\}/\{v_{k,1} \dots v_{k,N_{Conv}-1}\}$, pode-se representar a saída momentânea do codificador, dado o estado atual. Após esta saída, o novo estado é obtido por meio de uma transição no diagrama. A Figura 2.17 mostra o codificador da Figura 2.15 representado por máquina de estados.

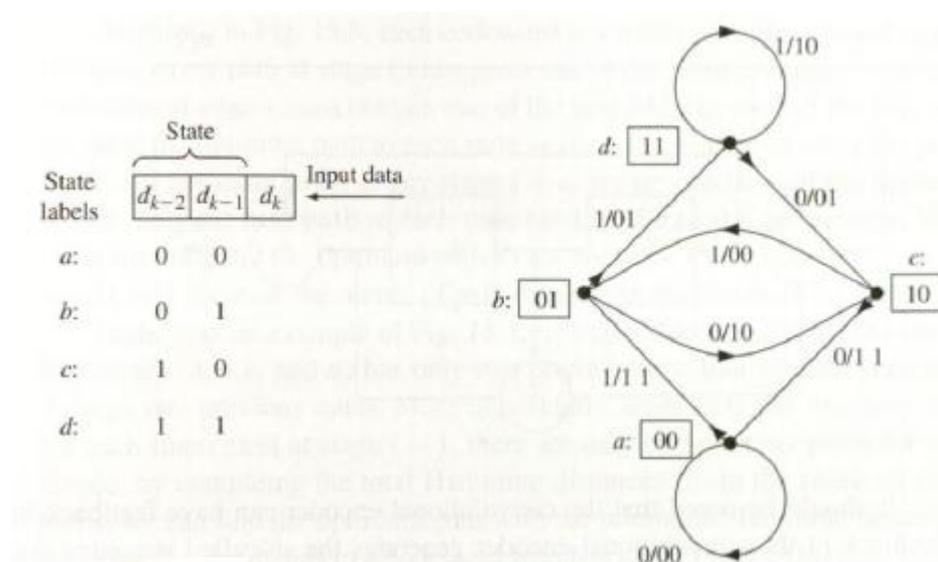


Figura 2.17: Representação do codificador da Figura 2.15 em máquina de estados (LATHI e DING, 2009).

Antes de prosseguir para os aspectos de decodificação de um código convolucional, outro tipo de representação conhecido como diagrama de treliças deve ser exposto. Este diagrama começa com o registrador de deslocamento preenchido com 0s e mostra as transições passo a passo para cada dígito de entrada. A notação é a mesma utilizada para a representação de máquina de estados e o codificador da Figura 2.15 é representado na Figura 2.18.

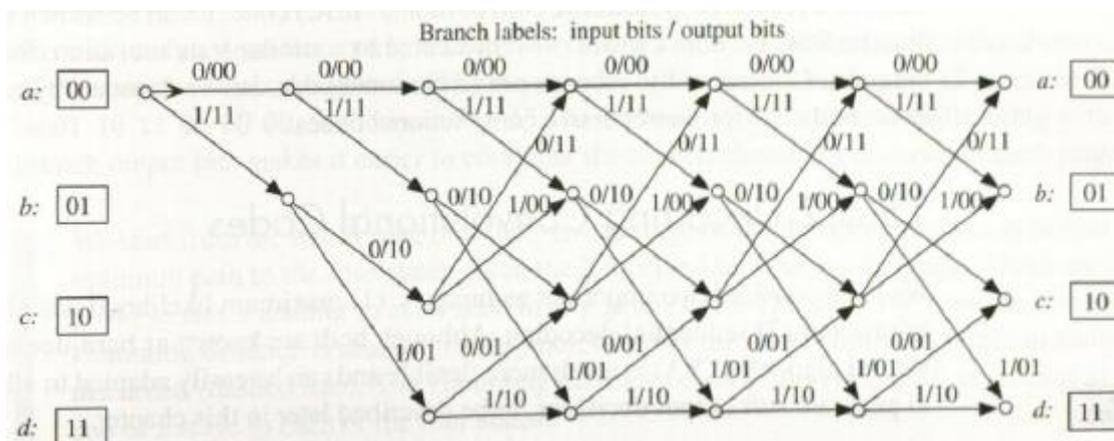


Figura 2.18: Representação do codificador da Figura 2.15 em diagrama de treliças (LATHI e DING, 2009).

Observando a figura anterior, nota-se que as transições podem ser visualizadas normalmente no diagrama de máquina de estados. Entretanto, o diagrama de treliças é útil para entendimento do processo de decodificação de códigos convolucionais, conforme será visto. Também é possível constatar que o diagrama evolui passo a passo da esquerda para a direita à medida que novos bits entram no registrador de deslocamento.

Para decodificar um código deste tipo, existem diversos métodos. Nesta seção, apenas o algoritmo de Viterbi (em inglês, *Viterbi's Maximum Likelihood Algorithm*) será analisado. Em canais AWGN, o critério de decisão MLD (*Maximum Likelihood Decision*) consiste em selecionar a palavra de código mais próxima da palavra recebida. Com isto, é necessário o armazenamento de 2^k palavras do dicionário de código, o que pode se tornar impraticável.

Observando a estrutura da Figura 2.18, observa-se que qualquer caminho do diagrama de treliças representa uma palavra de código válida. Sendo assim, a solução MLD de Viterbi consiste em encontrar o melhor caminho pela estrutura de treliças tal que a palavra encontrada seja a mais próxima possível da palavra recebida. A proximidade é estabelecida pela distância de Hamming, já definida na seção anterior.

Neste âmbito, em cada estágio ou passo do algoritmo na estrutura de treliças, a mínima distância de Hamming representa a sequência mais provável até o momento. Como cada caminho no i -ésimo estágio é originado dos caminhos no estágio anterior ($i - 1$), tem-se que o caminho ótimo é encontrado com base nos

caminhos predecessores, em um processo gradual. Desta maneira, o algoritmo apenas armazena o melhor caminho em cada estágio, que é chamado de caminho sobrevivente.

Para entender o algoritmo, um exemplo é exposto, novamente considerando o diagrama da Figura 2.18. Supondo uma sequência de bits 011011000000 recebida, a Figura 2.19 mostra os primeiros dois estágios do diagrama de treliças, com os cálculos de distâncias de Hamming associadas a cada caminho.

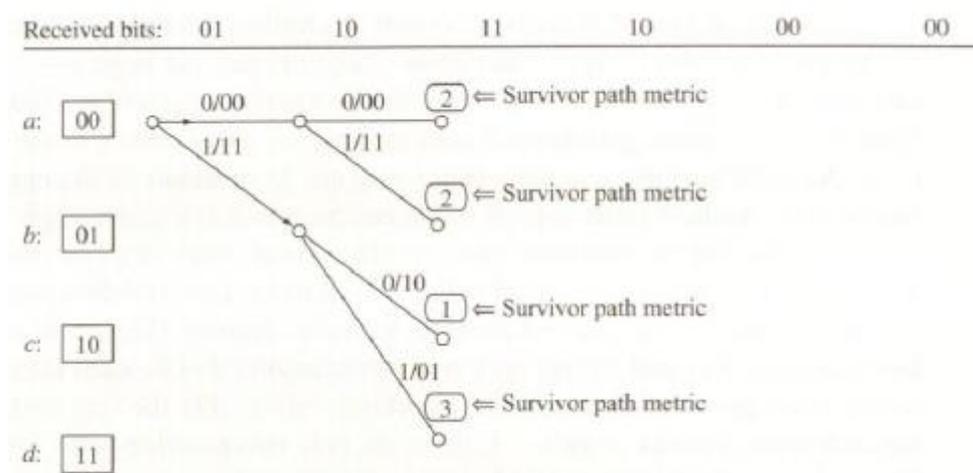


Figura 2.19: Análise dos dois primeiros estágios do algoritmo de Viterbi para exemplo da Figura 2.18 (LATHI e DING, 2009).

A figura anterior mostra que, inicialmente, até que os quatro estados possíveis sejam alcançados, a distância de Hamming é calculada para os possíveis caminhos do diagrama de treliças. Exemplificando para o caminho superior: Se um bit 0 é recebido no primeiro estágio, a saída é 00, que possui uma distância unitária com relação aos dois primeiros bits da palavra recebida. Se, novamente, um bit 0 for recebido, a saída será 00 e a distância com relação ao terceiro e quarto bits da palavra recebida é unitária. Sendo assim, a distância para o caminho no terceiro passo é a soma das distâncias anteriores e vale 2. O mesmo procedimento é aplicado para os outros caminhos expostos.

A Figura 2.20 mostra o próximo passo do algoritmo.

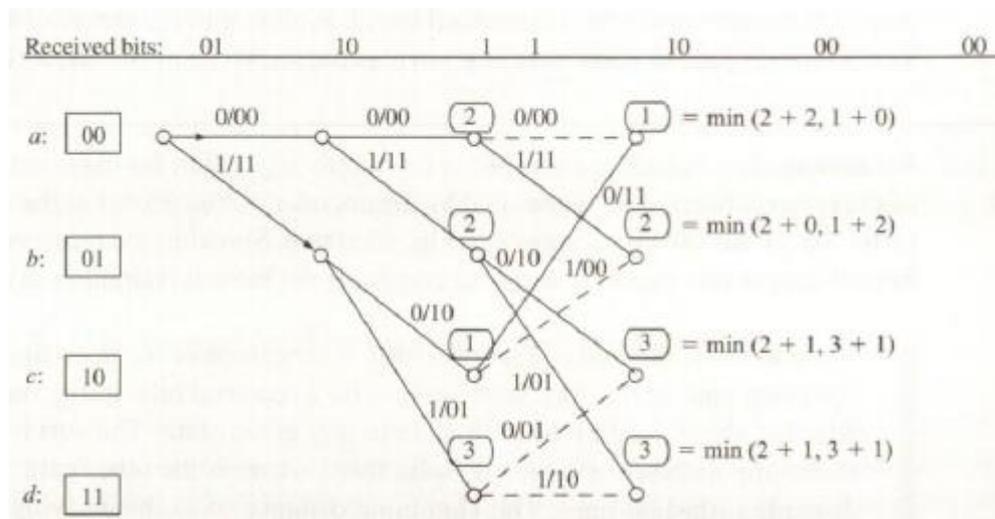


Figura 2.20: Análise do terceiro estágio do algoritmo de Viterbi para exemplo da Figura 2.18 (LATHI e DING, 2009).

No próximo passo, têm-se duas possibilidades de caminho para cada um dos quatro estados possíveis. Desta maneira, apenas o caminho com menor distância é escolhido (linha sólida), em detrimento do que possui maior distância (linha tracejada). Sendo assim, cada estado possui um caminho sobrevivente que será analisado nos passos posteriores.

Avançando a análise para o último passo, a Figura 2.21 mostra o resultado obtido para este exemplo.

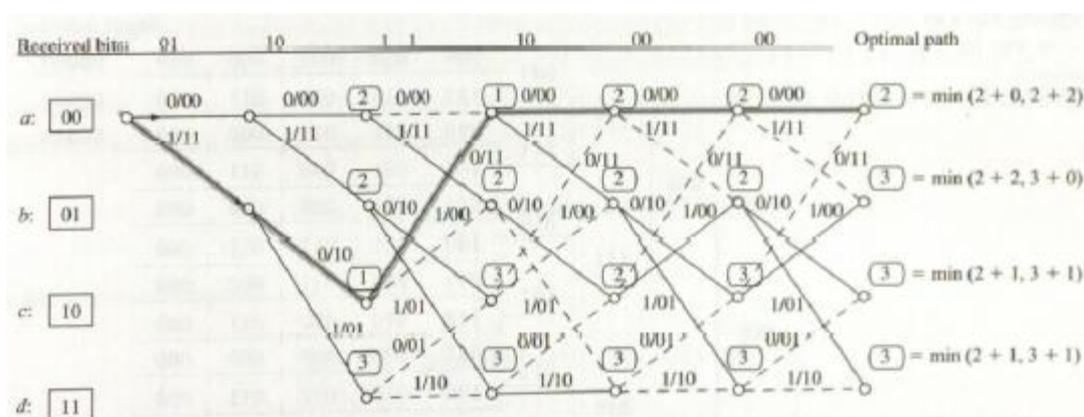


Figura 2.21: Análise do último estágio do algoritmo de Viterbi para exemplo da Figura 2.18 (LATHI e DING, 2009).

Observando o diagrama da figura anterior, nota-se que o caminho destacado é o selecionado, de tal forma que a palavra recebida mais próxima seria

111011000000, cuja palavra de mensagem correspondente é 100000. Caso houvesse uma situação de empate entre os caminhos, um processo de truncamento deveria ser realizado.

É válido ressaltar que o custo computacional deste algoritmo é proporcional a $2^{N_{Conv}-1}$ e esta é uma boa alternativa se $N_{Conv} < 10$. Para valores maiores, o algoritmo de decodificação sequencial pode ser uma boa escolha (LATHI e DING, 2009).

Por fim, a solução exposta para o algoritmo de Viterbi considera uma aplicação do tipo *hard-decision*, ou seja, o decodificador convolucional fornece apenas a palavra de código mais provável de ter sido transmitida. Entretanto, não se sabe o quanto o decodificador está confiante da resposta encontrada. Para aplicações em que esta informação é importante, existe uma modificação para o algoritmo de Viterbi que realiza *soft-decision*, podendo fornecer mais de uma palavra de código, cada qual com uma medida de confiabilidade correspondente.

2.3.3 Interleaving

Em aplicações reais de sistemas de comunicação digital, surtos de erros podem ocorrer, o que afeta severamente uma ou muitas palavras de código, de tal maneira que a decodificação se torna totalmente errônea. Uma técnica simples e bastante empregada é chamada de intercalação, ou em inglês, *interleaving*. Esta técnica consiste simplesmente em reordenar ou embaralhar múltiplas palavras de código geradas pelo codificador.

Ao utilizar esta técnica, os surtos de erros são espalhados por diversas palavras de código, de tal forma que cada uma destas possa corrigir apenas uma fração dos erros de bit. No decodificador, as palavras de código são reordenadas para que esta ação seja possível.

Existem diversas maneiras de realizar *interleaving*, dentre as quais podem ser destacadas a intercalação por bloco e pseudoaleatória. Em sistemas OFDM, esta é uma etapa importante e que ajuda na melhoria de desempenho geral (LATHI e DING, 2009).

2.4. Técnicas de alocação de recursos em sistemas sem fio OFDM

Conforme citado na Seção 2.2, a técnica de OFDM busca transmitir os dados paralelamente em cada subcanal, buscando eliminar os efeitos de desvanecimento seletivo em frequência e possibilitar a transmissão em altas taxas. Entretanto, para cada subportadora, existe um comportamento diferente da resposta em frequência do canal, o que implica em diferentes relações sinal-ruído (em inglês, *Signal-to-Noise Ratio* ou SNR) em cada subcanal OFDM.

Uma forma ligeiramente diferente da técnica de OFDM é chamada de modulação discreta multitonal (em inglês, *Discrete Multitone Modulation*, ou DMT), onde as operações básicas de processamento de sinal são idênticas à OFDM. A única diferença entre DMT e OFDM é que o transmissor da primeira técnica possui informações sobre cada subcanal correspondente, fornecidas pelo receptor (LATHI e DING, 2009). Sendo assim, sinais com diferentes constelações são transmitidos por cada subportadora, o que permite um controle da taxa e da potência, com o intuito de melhorar o desempenho do sistema para uma dada taxa de erro desejada.

Nesta seção, serão expostas diversas técnicas de alocação de recursos em sistemas OFDM. Antes de iniciar a análise destas técnicas, alguns conceitos básicos devem ser introduzidos, conforme exposto em (CIOFFI, 2008) e também (LI e STÜBER, 2006).

2.4.1 O conceito de *gap* de SNR

De acordo com a técnica de OFDM, é possível considerar que cada um dos subcanais sofre com ruído AWGN. Além disto, cada canal possui um ganho específico, conforme citado na seção anterior. Desta maneira, a SNR em cada um destes subcanais é dada por:

$$SNR = \frac{P_n}{\sigma^2} |H_n|^2 \quad (64)$$

Em (64), P_n representa a potência do sinal, σ^2 a potência de ruído por dimensão e H_n o ganho específico do n -ésimo subcanal. De acordo com a teoria de

Shannon, é possível mostrar que a capacidade de um canal AWGN (em bits/s) é dada por (LATHI e DING, 2009):

$$C = W \log_2(1 + SNR) \quad (65)$$

Outro modo de representar a capacidade de um canal é dividir o valor em bits/s pela banda do sistema, ou seja:

$$c = \frac{C}{W} = \log_2(1 + SNR) \quad (66)$$

Onde a unidade em questão é de bits/s/Hz. Muitas vezes a capacidade dada em (66) é chamada de eficiência de banda de um sistema, ou seja, quantos bits/s podem ser enviados por cada Hz de banda. Vale ressaltar que a capacidade do canal identifica que existe uma eficiência de banda $b \leq c$ tal que é possível transmitir informação pelo canal com probabilidade de erro arbitrariamente pequena. Caso $b > c$, não se pode garantir que a informação será recuperada no receptor. Desta maneira, todos os sistemas reais devem transmitir a uma eficiência b menor do que a eficiência máxima de capacidade do canal se desejam uma comunicação confiável entre o transmissor e o receptor. Além disto, quanto mais próximo da capacidade do canal, maior a taxa de transmissão do sistema.

A Tabela 2.2 mostra alguns exemplos para as eficiências de banda de diversos tipos de modulação, em bits/s/Hz.

Tabela 2.2: Eficiências de banda para diversos tipos de modulação.

Modulação	b (bits/s/Hz)
BPSK	1
QPSK	2
8-PSK	3
16-QAM	4
32-QAM	5
64-QAM	6

O conceito de *gap* de SNR está diretamente relacionado à capacidade do canal e reflete a diferença entre a taxa real transmitida e a taxa máxima possível

para transmissão confiável de dados, dada uma taxa de erro de bit (em inglês, *Bit Error Rate* ou BER). Sua definição é apresentada a seguir:

$$\Gamma = \frac{2^c - 1}{2^b - 1} = \frac{SNR}{2^b - 1} \quad (67)$$

Um ponto importante a ser observado é que, com o uso de codificação no sistema, pode-se diminuir o *gap* e chegar mais perto da capacidade de Shannon, o que é desejado na prática. Para muitas aplicações, considera-se que o valor do *gap* é constante para quase todos os valores de b . Para analisar mais detalhadamente esta hipótese, é apresentada, a seguir, uma análise de comparação para diferentes modulações M -árias QAM. A Figura 2.22 mostra as curvas teóricas de probabilidade de erro para um sistema que utiliza este tipo de modulação, considerando o canal AWGN, em função da SNR.

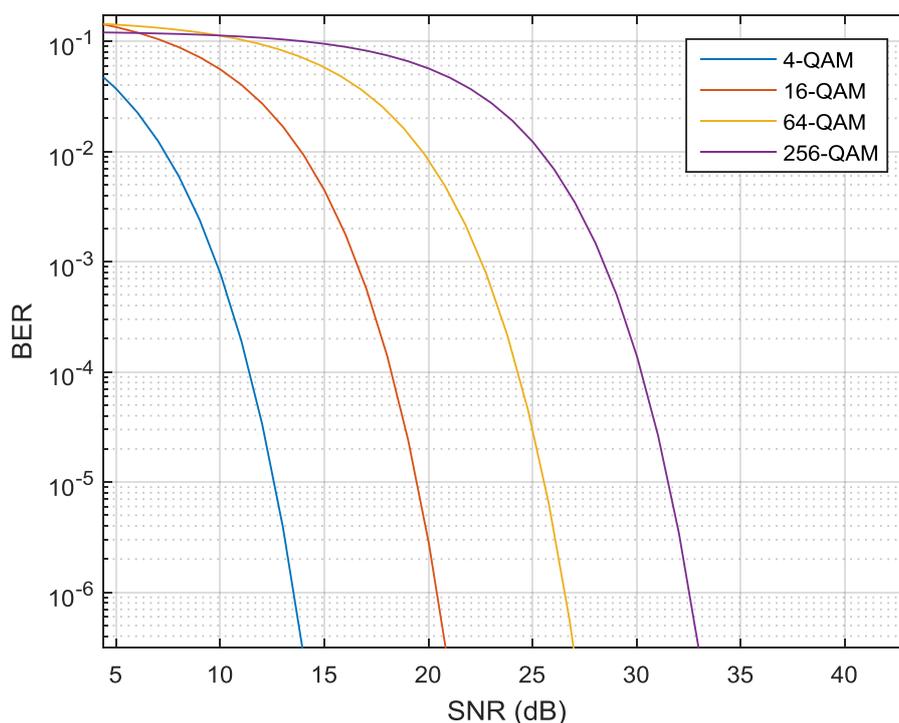


Figura 2.22: Probabilidades de erro para M -QAM, em função da SNR.

Considerando uma probabilidade de erro de 10^{-6} , as SNRs necessárias para atingir esta taxa de erro de bit são expostas na Tabela 2.3, a seguir. Além disto, esta tabela mostra o cálculo do *gap* para cada um dos casos em análise.

Tabela 2.3: Análise comparativa de *gap* de SNR para modulações *M*-QAM.

Modulação	SNR (dB) para BER = 10 ⁻⁶	2 ^{<i>b</i>} -1 (dB)	Γ (dB)
4-QAM	13,5	4,7	8,8
16-QAM	20,5	11,7	8,8
64-QAM	26,8	18,0	8,8
256-QAM	32,9	24,1	8,8

Como é possível observar, para este tipo de modulação, o *gap* é constante e vale 8,8 dB. Se o mesmo cálculo fosse feito para uma probabilidade de erro de 10⁻⁷, o valor obtido seria de aproximadamente 9,5 dB.

Conforme citado anteriormente, com o uso de técnicas de codificação, o valor do *gap* diminui. De acordo com o teorema de Shannon, o limite máximo para o *gap*, considerando transmissão confiável, é de 0 dB (ou seja, o sistema transmite no limite da capacidade do canal). Com a introdução deste conceito, a taxa normalizada¹ de transmissão de um sistema pode ser representada como:

$$b = \log_2 \left(1 + \frac{SNR}{\Gamma} \right) \quad (68)$$

A Figura 2.23 mostra algumas curvas de taxa em função da SNR, para diferentes valores de *gap*. Esta curva é baseada na equação anterior. Vale ressaltar que esta equação assume que a taxa normalizada *b* pode assumir qualquer valor. Em sistemas práticos, sabe-se que alguns métodos de modulação são escolhidos e a taxa normalizada pode assumir alguns valores discretos. Caso o sistema empregasse as constelações da Tabela 2.3, os valores de *b* poderiam ser 2, 4, 6 ou 8 bits/s/Hz, apenas. Neste caso, as curvas de taxa normalizada se tornam funções degrau, onde a taxa normalizada permanece constante por um intervalo de SNRs específico. Este comportamento também é exposto na Figura 2.23.

¹ A partir deste momento, a eficiência de banda será substituída por taxa normalizada, com unidades de bits/s/Hz.

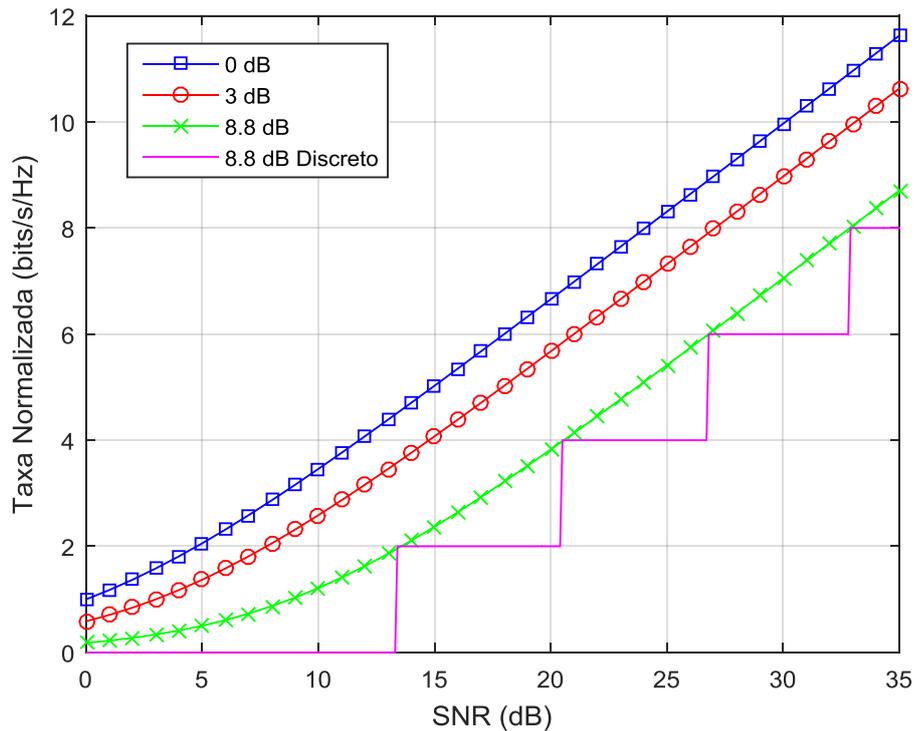


Figura 2.23: Taxa normalizada em função da SNR para diferentes valores de *gap*.

Nota-se que a curva para 0 dB representa a capacidade do canal e que, conforme o valor do *gap* aumenta, as curvas se distanciam do limite máximo. Além disto, o caso discreto mostra que a taxa normalizada se mantém constante para um intervalo de SNRs, o que é o caso de sistemas práticos.

Outro fato importante é que, por diversas vezes, se deseja uma taxa normalizada fixa para um sistema com múltiplas portadoras. Por exemplo, o sistema possui um requisito mínimo de taxa de transmissão (em bits/s), o que também impõe um requisito na taxa normalizada (em bits/s/Hz). Neste caso, pode-se definir a margem do sistema γ_m como a quantidade de SNR que pode ser reduzida mantendo-se a mesma probabilidade de erro. Sendo assim, uma margem negativa significa que a SNR precisa ser aumentada para que o sistema alcance a probabilidade de erro desejada.

Para determinar a margem, a seguinte expressão pode ser utilizada, tendo em vista o *gap* de SNR já definido anteriormente:

$$\gamma_m = \frac{2^{b_{desejada}} - 1}{2^b - 1} = \frac{SNR}{\Gamma} \quad (69)$$

Incluindo a margem na expressão da taxa normalizada, tem-se:

$$b = \log_2 \left(1 + \frac{SNR}{\Gamma \gamma_m} \right) \quad (70)$$

Até então, a análise das taxas normalizadas está relacionada com a SNR do sistema. No caso de um sistema com múltiplas portadoras e margem unitária, cada um dos N subcanais sofre com ruído AWGN e possui uma SNR específica. Neste caso, a taxa normalizada média do sistema é dada por:

$$\begin{aligned} \bar{b} &= \frac{1}{N} \sum_{n=1}^N b_n = \frac{1}{N} \sum_{n=1}^N \log_2 \left(1 + \frac{SNR_n}{\Gamma} \right) = \frac{1}{N} \log_2 \left(\prod_{n=1}^N \left(1 + \frac{SNR_n}{\Gamma} \right) \right) \\ &= \log_2 \left(1 + \frac{SNR_{mu}}{\Gamma} \right) \end{aligned} \quad (71)$$

Onde SNR_{mu} é chamada de SNR de multicanal e pode ser expressa por:

$$SNR_{mu} = \left[\left(\prod_{n=1}^N \left(1 + \frac{SNR_n}{\Gamma} \right) \right)^{\frac{1}{N}} - 1 \right] \Gamma \quad (72)$$

Observando as expressões anteriores, quando os termos +1 puderem ser ignorados (relação SNR_n/Γ alta), então a SNR multicanal é aproximadamente igual à uma média geométrica das SNRs de cada subcanal, ou seja:

$$SNR_{mu} \approx SNR_{geo} = \prod_{n=1}^N \left(\frac{SNR_n}{\Gamma} \right)^{\frac{1}{N}} \quad (73)$$

No caso desta aproximação, todos os subcanais são analisados em conjunto como um sistema com apenas uma portadora, sem ISI e SNR equivalente à SNR_{mu} .

Em todas as análises anteriores, o *gap* é considerado constante para todos os subcanais e constelações, como proposto em (LI e STÜBER, 2006) e (ARMADA, 2006). Na prática, esta é uma aproximação bastante utilizada para algoritmos de

alocação de recursos e será considerada durante todo o desenvolvimento das técnicas das seções posteriores. O erro associado a esta aproximação pode ser encontrado em (ZIVKOVIC, 2013).

2.4.2 Classificação dos algoritmos de alocação de recursos

De modo geral, os algoritmos de alocação de recursos são classificados em dois tipos. Existem os algoritmos que maximizam a taxa normalizada, mantendo-se uma restrição de potência, cujo critério de taxa adaptativa (em inglês, *Rate Adaptive* ou RA) é dado por:

$$\begin{aligned} \max_{P_n} b &= \sum_{n=1}^N \log_2 \left(1 + \frac{SNR_n}{\Gamma} \right), \text{ tal que:} \\ \sum_{n=1}^N P_n &= P_{total}, n = 1, \dots, N \end{aligned} \quad (74)$$

Onde P_n e P_{total} são positivas e representam a potência do n -ésimo subcanal e a potência total disponível no transmissor, respectivamente. Nota-se que as potências também podem ser consideradas no receptor, de tal forma que uma compensação para perda de percurso seja calculada. Desta maneira, os valores de SNR são calculados na entrada do receptor. Na Seção 3.2, isto será mostrado com detalhes nos exemplos apresentados.

Por outro lado, existem algoritmos que minimizam a potência utilizada para uma taxa normalizada fixa. Desta maneira, a otimização é feita na margem do sistema e o critério de margem adaptativa (em inglês, *Margin Adaptive* ou MA) é dado por:

$$\begin{aligned} \min_{P_n} P_{utilizada} &= \sum_{n=1}^N P_n, \text{ tal que:} \\ \sum_{n=1}^N \log_2 \left(1 + \frac{SNR_n}{\Gamma} \right) &= b_{desejada}, n = 1, \dots, N \end{aligned} \quad (75)$$

Um sistema que satisfaz o critério anterior possui uma margem máxima dada por:

$$\gamma_{max} = \frac{P_{total}}{P_{utilizada}} \quad (76)$$

2.4.3 Water-filling

Conforme citado na seção anterior, a abordagem RA consiste em maximizar a taxa normalizada, mantendo-se uma restrição de potência. Utilizando os conceitos anteriores, a taxa normalizada do n -ésimo canal pode ser escrita como:

$$b_n = \log_2 \left(1 + \frac{SNR_n}{\Gamma} \right) \quad (77)$$

Expandindo a SNR do subcanal, tem-se:

$$b_n = \log_2 \left(1 + \frac{P_n |H_n|^2}{\Gamma \sigma_n^2} \right) = \log_2 \left(1 + \frac{P_n g_n}{\Gamma} \right) \quad (78)$$

Onde o termo g_n representa a relação sinal-ruído do canal para potência do sinal unitária (é um valor fixo para cada um dos subcanais). Deseja-se otimizar a potência em cada subcanal de tal forma que a taxa total normalizada (dada pela somatória das taxas dos N subcanais) transmitida pelo sistema multiportadora seja máxima.

A solução deste problema é dada pelo conceito de multiplicadores de Lagrange, que são utilizados quando é necessário otimizar alguma variável com alguma restrição (BOYD e VANDENBERGHE, 2004). Aplicando-se o conceito para o problema em questão, tem-se a seguinte função custo:

$$L_{RA}(P_n, g_n, \lambda) = \sum_{n=1}^N \log_2 \left(1 + \frac{P_n g_n}{\Gamma} \right) + \lambda \left(\sum_{n=1}^N P_n - P \right) \quad (79)$$

Onde $P = P_{Total}$. Utilizando a seguinte propriedade:

$$\log_a b = \frac{\ln(b)}{\ln(a)} \quad (80)$$

A função custo pode ser reescrita como:

$$L_{RA}(P_n, g_n, \lambda) = \frac{1}{\ln(2)} \sum_{n=1}^N \ln\left(1 + \frac{P_n g_n}{\Gamma}\right) + \lambda \left(\sum_{n=1}^N P_n - P\right) \quad (81)$$

Onde λ é o multiplicador de Lagrange (constante). Para encontrar a solução ótima, deve-se derivar a função de custo com relação à potência do respectivo subcanal e igualar o resultado à zero, ou seja:

$$\frac{\partial L_{RA}(P_n, g_n, \lambda)}{\partial P_n} = \frac{1}{\ln(2)} \frac{\partial}{\partial P_n} \sum_{n=1}^N \ln\left(1 + \frac{P_n g_n}{\Gamma}\right) + \lambda \frac{\partial}{\partial P_n} \left(\sum_{n=1}^N P_n - P\right) = 0 \quad (82)$$

Derivando a expressão e particularizando para um subcanal específico, tem-se:

$$\frac{1}{\ln(2)} \frac{1}{\left(1 + \frac{P_n g_n}{\Gamma}\right)} \frac{g_n}{\Gamma} + \lambda = 0 \quad (83)$$

Simplificando a expressão:

$$\frac{1}{\ln(2)} \frac{1}{\left(\frac{\Gamma}{g_n} + P_n\right)} = -\lambda$$

$$\frac{\Gamma}{g_n} + P_n = \frac{-1}{\lambda \ln(2)} \quad (84)$$

Ou seja, como λ é constante, tem-se:

$$P_n + \frac{\Gamma}{g_n} = \text{constante} = K \quad (85)$$

Tendo em vista que esta solução é válida para um subcanal, tem-se um sistema com $N+1$ equações, incluindo-se a restrição de potência:

$$\left\{ \begin{array}{l} P_1 + \frac{\Gamma}{g_1} = K \\ P_2 + \frac{\Gamma}{g_2} = K \\ \vdots \\ P_n + \frac{\Gamma}{g_n} = K \\ P_1 + P_2 + \dots + P_n = P_{total} \end{array} \right.$$

Se as N primeiras equações são somadas e a última equação é utilizada:

$$K = \frac{1}{N} \left(\sum_{n=1}^N P_n + \sum_{n=1}^N \frac{\Gamma}{g_n} \right) = \frac{1}{N} \left(P_{total} + \Gamma \sum_{n=1}^N \frac{1}{g_n} \right) \quad (86)$$

Por fim, a potência de cada subcanal é dada por:

$$P_n = K - \frac{\Gamma}{g_n} \quad (87)$$

Esta solução é conhecida como *water-filling*, pois o termo Γ/g_n é utilizado com analogia a uma vasilha onde água (potência) é despejada, até que o nível máximo seja atingido. Este nível máximo é determinado justamente pela constante K , que é calculada tendo em vista o comportamento de cada um dos subcanais.

É importante ressaltar que o problema anterior foi resolvido tendo em vista uma solução RA, ou seja, maximizando a taxa normalizada total de transmissão. Sendo assim, a seguir, será discutida a solução MA, que maximiza a margem do

sistema multiportadora. Neste caso, é desejado minimizar a potência mantendo-se uma taxa normalizada constante, ou seja, a função custo de Lagrange pode ser escrita como:

$$L_{MA}(P_n, g_n, \lambda) = \sum_{n=1}^N P_n + \lambda \left(\sum_{n=1}^N \log_2 \left(1 + \frac{P_n g_n}{\Gamma} \right) - b_{desejada} \right) \quad (88)$$

Reescrevendo a função custo em termos do logaritmo natural, tem-se:

$$L_{MA}(P_n, g_n, \lambda) = \sum_{n=1}^N P_n + \lambda \left(\frac{1}{\ln(2)} \sum_{n=1}^N \ln \left(1 + \frac{P_n g_n}{\Gamma} \right) - b_{desejada} \right) \quad (89)$$

Assim como no caso RA, derivando-se a função e custo com relação à P_n e igualando a zero, particularizando para um subcanal, tem-se:

$$\frac{\partial L_{MA}(P_n, g_n, \lambda)}{\partial P_n} = \frac{\partial}{\partial P_n} \sum_{n=1}^N P_n + \frac{\lambda}{\ln(2)} \frac{\partial}{\partial P_n} \left(\sum_{n=1}^N \ln \left(1 + \frac{P_n g_n}{\Gamma} \right) - b_{desejada} \right) = 0 \quad (90)$$

Resolvendo as derivadas e particularizando para o caso de um subcanal:

$$1 + \frac{\lambda}{\ln(2)} \frac{1}{\left(1 + \frac{P_n g_n}{\Gamma} \right)} \frac{g_n}{\Gamma} = 0 \quad (91)$$

Simplificando a expressão:

$$\frac{\lambda}{\ln(2)} \frac{1}{\left(\frac{\Gamma}{g_n} + P_n \right)} = -1$$

$$P_n + \frac{\Gamma}{g_n} = -\frac{\lambda}{\ln(2)} = \text{constante} = K_{MA} \quad (92)$$

A expressão anterior é muito semelhante à encontrada para a solução RA, exceto pela diferença entre as constantes. Por outro lado, a restrição no caso MA é imposta na taxa normalizada desejada, ou seja:

$$b = \sum_{n=1}^N \log_2 \left(1 + \frac{P_n g_n}{\Gamma} \right) = b_{desejada} \quad (93)$$

Observando a expressão que define a constante K_{MA} , nota-se que, multiplicando ambos os lados por g_n/Γ tem-se:

$$\frac{g_n}{\Gamma} \left(P_n + \frac{\Gamma}{g_n} \right) = \frac{g_n}{\Gamma} K_{MA}$$

$$\frac{P_n g_n}{\Gamma} = \frac{g_n}{\Gamma} K_{MA} - 1 \quad (94)$$

Substituindo este resultado na restrição de taxa normalizada:

$$b = \sum_{n=1}^N \log_2 \left(1 + \frac{g_n}{\Gamma} K_{MA} - 1 \right) = \sum_{n=1}^N \log_2 \left(\frac{g_n}{\Gamma} K_{MA} \right) \quad (95)$$

A soma de logaritmos exposta na expressão anterior pode ser substituída pelo produtório dentro do logaritmo, como se segue:

$$b = \sum_{n=1}^N \log_2 \left(\frac{g_n}{\Gamma} K_{MA} \right) = \log_2 \left(\prod_{n=1}^N \frac{g_n}{\Gamma} K_{MA} \right) \quad (96)$$

Para encontrar o valor da constante K_{MA} , basta isolá-la na expressão anterior:

$$b = \log_2 \left(\prod_{n=1}^N \frac{g_n}{\Gamma} K_{MA} \right)$$

$$2^b = \prod_{n=1}^N \frac{g_n}{\Gamma} K_{MA} = \left(\frac{K_{MA}}{\Gamma} \right)^N \prod_{n=1}^N g_n$$

$$\left(\frac{K_{MA}}{\Gamma} \right)^N = \frac{2^b}{\prod_{n=1}^N g_n}$$

$$K_{MA} = \Gamma \left(\frac{2^b}{\prod_{n=1}^N g_n} \right)^{1/N} \quad (97)$$

Por fim, a solução para o *water-filling* MA é dada por:

$$P_n = K_{MA} - \frac{\Gamma}{g_n} \quad (98)$$

Na prática, algoritmos computacionais são realizados para simular o *water-filling*. A Figura 2.24 mostra os passos necessários para calcular as soluções RA e MA. Observando os algoritmos, nota-se que a solução é iterativa e os canais com baixos valores de g_n são eliminados gradualmente, de acordo com a constante K correspondente. Vale ressaltar que N^* corresponde ao número de canais utilizados efetivamente, visto que alguns podem ser eliminados devido ao baixo desempenho nas condições de operação do sistema.

RA Water-filling Algorithm	MA Water-filling Algorithm
1. Sort g_n in a descending order such that $g_1 = \max_n g_n$ and $g_N = \min_n g_n$	
2. Set the number of used subchannels $N^* = N$	
3. Initialize	3. Initialize
$\mathcal{G} = \sum_{n=1}^N \frac{1}{g_n}$ $K = \frac{1}{N}(\mathcal{E}_x + \Gamma \mathcal{G})$	$\mathcal{G} = \sum_{n=1}^N \log_2 g_n$ $K = \Gamma 2^{(b-\mathcal{G})/N}$
4. While $(\mathcal{E}_{N^*} = K - \frac{\Gamma}{g_{N^*}}) < 0$	4. While $(\mathcal{E}_{N^*} = K - \frac{\Gamma}{g_{N^*}}) < 0$
$\mathcal{G} \leftarrow \mathcal{G} - \frac{1}{g_{N^*}}$ $N^* \leftarrow N^* - 1$ $K = \frac{1}{N^*}(\mathcal{E}_x + \Gamma \mathcal{G})$	$\mathcal{G} \leftarrow \mathcal{G} - \frac{1}{g_{N^*}}$ $N^* \leftarrow N^* - 1$ $K = \Gamma 2^{(b-\mathcal{G})/N^*}$
5. Compute	5. Compute
$\mathcal{E}_n = \begin{cases} K - \frac{\Gamma}{g_n} & n=1, \dots, N^* \\ 0 & n=N^*, \dots, N \end{cases}$ $b_n = \log_2\left(1 + \frac{\mathcal{E}_n g_n}{\Gamma}\right)$ $= \log_2\left(1 + \frac{K g_n}{\Gamma}\right)$	$\mathcal{E}_n = \begin{cases} K - \frac{\Gamma}{g_n} & n=1, \dots, N^* \\ 0 & n=N^*, \dots, N \end{cases}$ $b_n = \log_2\left(1 + \frac{\mathcal{E}_n g_n}{\Gamma}\right)$ $= \log_2\left(1 + \frac{K g_n}{\Gamma}\right)$

Figura 2.24: Algoritmos de *Water-Filling*, soluções RA e MA (LI e STÜBER, 2006).

As soluções de *water-filling* são ótimas e representam a melhor alocação de recursos possível. Por outro lado, na prática, existem outros algoritmos sub-ótimos que tentam aproximar a solução *water-filling* para taxas normalizadas discretas e disponíveis no sistema. Nas seções posteriores, dois destes algoritmos serão estudados.

Entretanto, antes de prosseguir para a análise dos algoritmos, o conceito de granularidade deve ser exposto. Quando são utilizadas taxas normalizadas discretas, apenas alguns valores são possíveis para implementação prática. Sendo assim, a granularidade (β) é o menor incremento de taxa normalizada possível para o subcanal em questão. Portanto, a taxa normalizada de cada subcanal (b_n) pode ser representada como:

$$b_n = \beta B_n \quad (99)$$

Onde B_n é um número inteiro não negativo. Como exemplo, pode-se considerar um sistema prático que possui como modulações disponíveis BPSK, QPSK, 8-PSK, 16-QAM, 32-QAM e 64-QAM. Neste caso, as taxas normalizadas possíveis variam de 1 a 6 bits/s/Hz, com passos (granularidade) de 1 bit/s/Hz. O caso da Figura 2.23 mostrou um sistema com granularidade finita, ou seja, a taxa normalizada em função da SNR pode ser representada por funções do tipo degrau.

2.4.4 Algoritmos de Chow

Os algoritmos propostos por Peter Chow (CHOW, CIOFFI e BINGHAM, 2002) possuem resultados próximos ao do *water-filling* (CIOFFI, 2008) e buscam considerar a granularidade finita de um sistema prático. Estes algoritmos alocam potência igualmente para todos os subcanais utilizados e, apesar de resultarem em uma boa solução, são sub-ótimos e podem ser computacionalmente custosos.

Assim como exposto na seção anterior, os algoritmos de Chow podem ser separados nas soluções RA e MA. Contudo, ambas soluções partem de um algoritmo inicial de Chow, que nada mais é do que um ajuste das potências de cada subcanal e da taxa normalizada total do sistema. O algoritmo inicial de Chow para alocação de recursos é exposto a seguir:

1. Ordenar g_n de cada subcanal em ordem decrescente.
2. Seja i o número de subcanais utilizados e $b_{temp}(i)$ a taxa total do sistema com os i canais. Inicialmente, $i = N$ e $b_{temp}(N+1) = 0$.
3. Calcule $P_n = P_{total}/i$ (divisão igual de potência) para os canais utilizados. Para os demais canais, $P_n = 0$. Sendo assim, $SNR_n = g_n P_n$.
4. Calcule a taxa total do sistema $b_{temp}(i) = \sum b_n$.
5. Verifique se $b_{temp}(i) < b_{temp}(i+1)$. Se este for o caso, parar o algoritmo com a solução $b_{temp}(i+1)$. Caso contrário, remover o subcanal com menor g_n e retornar ao terceiro passo.

É interessante notar que o algoritmo anterior termina no momento em que a taxa normalizada total do sistema diminui com a remoção de subcanais. Além disto, ao final destas etapas, as taxas normalizadas dos subcanais não serão discretas. Outro ponto importante é que, ao implementar este algoritmo no MATLAB, o primeiro passo não é obrigatório, tendo em vista a utilização da função `find()`.

Conforme citado anteriormente, a solução RA é limitada pela potência total disponível, alocando taxa para os subcanais utilizados. Para encontrar esta solução partindo do algoritmo inicial de Chow, três passos adicionais são aplicados para os subcanais utilizados:

6. Se $b_n/\beta - \lfloor b_n/\beta \rfloor < 0.5\beta$, arredonde b_n para baixo, ou seja, $B_n = \lfloor b_n/\beta \rfloor$. Com esta alteração de taxa, a potência precisa ser ajustada por um fator de $2^{\beta B_n} - 1 / 2^{b_{n,antiga}} - 1$. Como o valor de taxa normalizada arredondado é menor do que a taxa normalizada assumindo granularidade infinita, a potência é reduzida no subcanal em questão.
7. Se $b_n/\beta - \lfloor b_n/\beta \rfloor > 0.5\beta$, arredonde b_n para cima, ou seja, $B_n = \lceil b_n/\beta \rceil$. Com esta alteração de taxa, a potência precisa ser ajustada por um fator de $2^{\beta B_n} - 1 / 2^{b_{n,antiga}} - 1$. Como o valor de taxa normalizada arredondada é maior do que a taxa normalizada assumindo granularidade infinita, a potência aumenta no subcanal em questão.
8. Calcule a potência total utilizada e ajuste a potência em todos os subcanais pelo fator $P_{total}/P_{utilizada}$, de tal forma que a potência total do algoritmo seja igual à disponível pelo sistema (solução RA).

Observando os passos 6 e 7, nota-se que um ajuste de potência é feito para os subcanais que têm sua taxa normalizada arredondada. Este ajuste está de acordo com a equação de taxa normalizada exposta inicialmente. Neste âmbito, tem-se:

$$b_n = \log_2 \left(1 + \frac{SNR_n}{\Gamma} \right) = \log_2 \left(1 + \frac{\frac{P_n}{\sigma^2} |H_n|^2}{\Gamma} \right) = \log_2 \left(1 + \frac{P_n g_n}{\Gamma} \right)$$

Ou seja:

$$P_n = \frac{\Gamma}{g_n} (2^{b_n} - 1) \quad (100)$$

Para granularidade infinita (ou seja, $b_n = b_{n,antiga}$). Supondo uma taxa nova arredondada $b_n = \beta B_n$, a expressão anterior seria semelhante, ou seja:

$$P_{n,nova} = \frac{\Gamma}{g_n} (2^{\beta B_n} - 1) \quad (101)$$

Para encontrar o fator de ajuste de potência utilizado por Chow, basta relacionar as duas equações anteriores da seguinte maneira:

$$\frac{P_{n,nova}}{P_n} = \frac{\frac{\Gamma}{g_n} (2^{\beta B_n} - 1)}{\frac{\Gamma}{g_n} (2^{b_n} - 1)} = \frac{(2^{\beta B_n} - 1)}{(2^{b_{n,antiga}} - 1)} \quad (102)$$

Que é, de fato, o fator utilizado. Vale ressaltar que este fator pode ser maior ou menor do que um, dependendo se o arredondamento é feito para cima ou para baixo, respectivamente.

A solução MA de Chow (taxa normalizada fixa desejada) também parte do algoritmo mostrado nos cinco passos anteriores. A partir de então, alguns passos adicionais são necessários:

6. Seja i o número de subcanais utilizados. Calcule a SNR multicanal, dada por:

$$SNR_{mu} = \left[\left(\prod_{n=1}^i \left(1 + \frac{SNR_n}{\Gamma} \right) \right)^{\frac{1}{i}} - 1 \right] \Gamma \quad (103)$$

7. Calcule a margem temporária do sistema γ_{temp} :

$$\gamma_{temp} = \frac{\frac{SNR_{mu}}{\Gamma}}{2^{\bar{b}} - 1} \quad (104)$$

8. Calcule a distribuição de bits temporária do sistema, dada por:

$$B_{temp,n} = \frac{1}{\beta} \log_2 \left(1 + \frac{SNR_n}{\Gamma \gamma_{temp}} \right), n = 1, \dots, i \quad (105)$$

9. Arredonde $B_{temp,n}$ para os inteiros mais próximos (considerando a granularidade finita) e recalcule as potências associadas utilizando:

$$P_{n,nova,temp} = \frac{\Gamma}{g_n} (2^{\beta B_{temp,n}} - 1) \quad (106)$$

10. Calcule a taxa normalizada total temporária do sistema, dada por:

$$B_{temp} = \sum_{n=1}^i B_{temp,n} \rightarrow b_{temp} = B_{temp} \beta \quad (107)$$

Após calcular a taxa normalizada, observe se $b_{temp} = b_{desejada}$. Caso a taxa normalizada encontrada não seja igual à desejada, um algoritmo iterativo de correção deve existir para corrigir esta discrepância. Um fato interessante é que, se houver potência suficiente no sistema e a taxa normalizada temporária for diferente da desejada, a diferença entre estas será muito pequena (1 a 3 bits/s/Hz normalmente). Sendo assim, um algoritmo simples é o de retornar ao nono passo e arredondar as taxas normalizadas mais próximas de $\beta/2$ para a outra direção (por exemplo, se uma taxa de 0,51 fora arredondada para 1, esta seria arredondada para 0 na correção).

11. Possuindo uma taxa normalizada total do sistema que é igual à desejada, basta ajustar a potência dos subcanais utilizados pelo fator $P_{total}/P_{utilizada}$.

Como é possível notar das soluções de Chow, os algoritmos possuem etapas bem definidas e fornecem uma solução para a alocação de recursos RA e MA. Vale ressaltar que o arredondamento é realizado tendo em vista a granularidade finita do sistema e, conforme citado anteriormente (e será visto na seção de resultados), a solução encontrada é sub-ótima, apesar de próxima do *water-filling*.

2.4.5 Algoritmos de Levin-Campello

Uma solução ótima para algoritmos de alocação discretos (considerando granularidade finita do sistema) parte do princípio de um algoritmo ganancioso (em inglês, *greedy algorithm*) (BEDNORZ, 2008). Este tipo de algoritmo analisa iteração por iteração, encontrando a melhor solução local momentânea. Espera-se que, ao escolher soluções locais ótimas, a solução final também seja ótima.

O método proposto por Jorge Campello de Souza (CAMPELLO, 1999) e Howard Levin (LEVIN, 2001) parte do princípio de que cada incremento de informação adicional fornecido ao sistema é proporcionado ao subcanal que necessita da menor alocação de potência incremental correspondente. Este tipo de algoritmo ganancioso encontra uma solução ótima quando a granularidade é igual para todos os subcanais do sistema (CIOFFI, 2008), o que de fato é bastante comum em aplicações práticas.

Antes de iniciar a análise das etapas de alocação, mais conceitos devem ser introduzidos. Inicialmente, foi dito que a alocação é realizada aos canais com menor potência incremental necessária. O conceito de potência incremental parte do princípio de que a potência em cada subcanal é uma função da taxa normalizada alocada para este. Explicitando esta relação, tem-se:

$$P_n \rightarrow P_n(b_n) \quad (108)$$

Então, a potência incremental p_n necessária para transmitir b_n unidades de informação em um subcanal é a quantidade de potência adicional necessária para

transmitir uma unidade a mais de taxa normalizada, considerando a granularidade do sistema:

$$p_n(b_n) = P_n(b_n) - P_n(b_n - \beta) \quad (109)$$

Conforme visto anteriormente, agora explicitando a relação da potência com relação à taxa normalizada, tem-se:

$$P_n(b_n) = \frac{\Gamma}{g_n} (2^{b_n} - 1) \quad (110)$$

Então a potência incremental é dada por:

$$\begin{aligned} p_n(b_n) &= P_n(b_n) - P_n(b_n - \beta) = \frac{\Gamma}{g_n} (2^{b_n} - 1) - \frac{\Gamma}{g_n} (2^{b_n - \beta} - 1) \\ &= \frac{\Gamma}{g_n} (2^{b_n} - 1 - 2^{b_n - \beta} + 1) = \frac{\Gamma}{g_n} (2^{b_n} - 2^{b_n - \beta}) \end{aligned}$$

Logo:

$$p_n(b_n) = \frac{\Gamma}{g_n} 2^{b_n} (1 - 2^{-\beta}) \quad (111)$$

Se $\beta = 1$, é possível notar que:

$$p_n(b_n) = \frac{\Gamma}{g_n} 2^{b_n} (1 - 2^{-1}) = \frac{\Gamma}{g_n} \frac{2^{b_n}}{2} = \frac{\Gamma}{g_n} 2^{b_n - 1} = 2p_n(b_n - 1) \quad (112)$$

A expressão anterior mostra que, quando a granularidade é unitária, a potência incremental para uma constelação com b_n bits/s/Hz é o dobro da potência incremental para a constelação anterior do sistema, ou seja, com $b_n - 1$ bits/s/Hz.

Após selecionar um sistema com valor de β específico, uma tabela com as potências incrementais em função da taxa normalizada pode ser construída. Esta tabela é essencial para o correto funcionamento do algoritmo de Levin-Campello,

cujas etapas serão discutidas posteriormente. Tendo definido o valor de β e as potências incrementais para as taxas normalizadas do sistema, o objetivo é encontrar uma distribuição destas taxas para os subcanais utilizados que seja ótima.

Para a solução MA, a solução ótima é a distribuição que requer menor quantidade de potência para a taxa normalizada desejada. Por outro lado, para a solução RA, podem existir diversas distribuições que alcançam a taxa normalizada máxima. Sendo assim, a escolha ótima é a que necessita de menor potência total (LI e STÜBER, 2006), ou seja, a mais eficiente. Uma distribuição é dita eficiente se:

$$\max_n p_n(b_n) \leq \min_m p_m(b_m + \beta) \quad (113)$$

A definição anterior mostra que se uma distribuição é eficiente, então não existe movimento possível de bits de informação de um subcanal para outro que diminua ainda mais a potência total necessária para o sistema, ou seja, a distribuição eficiente já utiliza a menor potência possível. O início do método de Levin-Campello é baseado em uma etapa que torna a distribuição de taxas normalizadas eficiente. O algoritmo relacionado é exposto a seguir:

1. $m \leftarrow \operatorname{argmin}_{1 \leq i \leq N} p_i(b_i + \beta)$
2. $n \leftarrow \operatorname{argmax}_{1 \leq j \leq N} p_j(b_j)$
3. Enquanto $p_m(b_m + \beta) < p_n(b_n)$ faça
 - a. $b_m \leftarrow b_m + \beta$
 - b. $b_n \leftarrow b_n - \beta$
 - c. $m \leftarrow \operatorname{argmin}_{1 \leq i \leq N} p_i(b_i + \beta)$
 - d. $n \leftarrow \operatorname{argmax}_{1 \leq j \leq N} p_j(b_j)$

O algoritmo anterior troca unidades de informação entre os subcanais até que a condição de distribuição eficiente seja alcançada. Além disto, em sistemas práticos, como sempre é necessária maior potência para que haja aumento de taxa, a distribuição gerada é sempre eficiente (LI e STÜBER, 2006).

Basicamente, o algoritmo apresentado seleciona, em cada passo, o subcanal (m -ésimo) que possui a menor potência incremental, considerando um nível de granularidade superior. Além disto, o algoritmo também seleciona o subcanal (n -

ésimo) que possui a maior potência incremental, considerando a taxa normalizada atual.

Se a potência incremental do m -ésimo subcanal é menor do que a potência incremental do n -ésimo subcanal, então é possível retirar um bit de informação do n -ésimo subcanal e adicioná-lo ao m -ésimo (há potência suficiente para esta operação). Caso contrário, a distribuição já é eficiente e não existe movimento de unidades de informação de um subcanal para o outro sem que a potência necessária para tal operação diminua a potência total do sistema.

As soluções de Levin-Campello possuem como ponto inicial uma distribuição de taxas normalizadas eficiente. Após obter uma distribuição deste tipo, é possível alcançar as soluções RA e MA.

Para obter a solução RA, é necessário que a distribuição eficiente não viole a restrição de potência total imposta pelo sistema prático. Sendo assim, o conceito de *P-Tightness* (ajustar a potência do sistema) deve ser respeitado. Uma distribuição é dita justa em potência (em inglês, *P-Tight*) se:

$$0 \leq P_{total} - \sum_{n=1}^N p_n(b_n) \leq \min_{1 \leq i \leq N} p_i(b_i + \beta) \quad (114)$$

Ou seja, a solução RA, além de não violar a potência total imposta pelo sistema, garante que não é possível movimentar nenhum bit entre quaisquer subcanais sem que o sistema utilize mais potência do que já possui. O algoritmo para tornar uma distribuição justa em potência é exposto a seguir:

1. Calcule $S = \sum_{n=1}^N P_n(b_n)$
2. Enquanto $P_{total} - S \geq \min_{1 \leq i \leq N} p_i(b_i + \beta)$ ou $P_{total} - S < 0$ faça:
 - a. Se $P_{total} - S < 0$ então:
 - i. $n \leftarrow \operatorname{argmax}_{1 \leq i \leq N} p_i(b_i + \beta)$
 - ii. $S \leftarrow S - p_n(b_n)$
 - iii. $b_n \leftarrow b_n - \beta$

b. Caso contrário:

- i. $m \leftarrow \operatorname{argmin}_{1 \leq i \leq N} p_i(b_i + \beta)$
- ii. $S \leftarrow S + p_m(b_m + \beta)$
- iii. $b_m \leftarrow b_m + \beta$

Analisando o algoritmo anterior, nota-se que este é simples. Primeiramente, a soma das potências incrementais da distribuição eficiente é realizada e armazenada na variável S .

A partir de então, dentro do laço de execução, observa-se primeiramente se S é maior do que a potência total disponível pelo sistema. Se isto for verdade, então é necessário diminuir a potência total da distribuição eficiente. Para isto, o subcanal que apresenta a maior potência incremental para um nível acima de granularidade é selecionado e um bit de informação é retirado deste. Com isto, a potência incremental em excesso desta operação é removida do sistema. Vale notar que este processo é repetido até que a potência total da distribuição seja menor do que a potência total disponível pelo sistema.

Por outro lado, se é possível adicionar mais bits de informação ao sistema, isto é feito ao subcanal que possui menor requisição de potência incremental para esta operação. Este processo também é repetido até que o critério de potência total RA seja obedecido.

Sendo assim, ao utilizar o algoritmo de tornar uma distribuição eficiente em conjunto com o algoritmo de *P-tight*, tem-se a solução RA para o método de Levin-Campello. A solução MA também necessita de uma distribuição eficiente, porém, esta distribuição precisa ser justa em taxa normalizada (em inglês, *B-Tight*). Uma distribuição é dita justa em taxa se a taxa desejada é a mesma obtida pelo algoritmo, ou seja:

$$b_{desejada} = \sum_{n=1}^N b_n \quad (115)$$

No caso da solução MA, além de alcançar uma taxa normalizada igual à desejada, há a garantia de uma distribuição eficiente de subcanais. O algoritmo que torna uma distribuição justa em taxa é exposto a seguir:

1. Calcule $b' = \sum_{n=1}^N b_n$
2. Enquanto ($b' \neq b_{desejada}$) faça
 - a. Se $b' > b_{desejada}$ então
 - i. $n \leftarrow \operatorname{argmax}_{1 \leq i \leq N} p_i(b_i)$
 - ii. $b' \leftarrow b' - \beta$
 - iii. $b_n \leftarrow b_n - \beta$
 - b. Caso contrário
 - i. $m \leftarrow \operatorname{argmin}_{1 \leq i \leq N} p_i(b_i + \beta)$
 - ii. $b' \leftarrow b' + \beta$
 - iii. $b_m \leftarrow b_m + \beta$

O algoritmo anterior é muito semelhante ao de ajuste de potência, utilizado na solução RA. A diferença é que o ajuste ocorre devido à restrição de taxa normalizada desejada para o sistema. Sendo assim, inicialmente, a variável b' guarda o valor da taxa normalizada total do sistema dada pela distribuição eficiente. Enquanto o valor de taxa normalizada atual não for igual à taxa normalizada total desejada, os bits são ajustados nos subcanais específicos.

Se a taxa normalizada atual é maior do que a desejada, o subcanal com maior potência incremental considerando-se sua taxa normalizada atual é selecionado e um bit de informação (considerando a granularidade do sistema) é removido deste subcanal (o ajuste também é feito para a variável b' , atualizando-a). Por outro lado, se a taxa normalizada atual é menor do que a desejada, ao subcanal com menor potência incremental considerando-se um nível de granularidade superior é adicionado um bit de informação (considerando a granularidade do sistema) e a variável b' é atualizada.

Ao final, a combinação de uma distribuição eficiente com o algoritmo de *B-Tight* produz a solução MA para o método de Levin-Campello. A Figura 2.25 mostra um resumo das soluções RA e MA de Levin-Campello.

Levin-Campello General RA Algorithm	Levin-Campello General MA Algorithm
<ol style="list-style-type: none">1. Choose any b2. Make b efficient with the EF algorithm.3. E-tighten the resultant b with the ET algorithm.	<ol style="list-style-type: none">1. Choose any b2. Make b efficient with the EF algorithm.3. B-tighten the resultant b with the BT algorithm.

Figura 2.25: Resumo das soluções RA e MA para o método de Levin-Campello (LI e STÜBER, 2006).

3. ANÁLISE DE RESULTADOS

Esta seção apresenta os resultados obtidos neste trabalho. A Seção 3.1 mostra uma análise de desempenho de sistemas OFDM, tanto para um sistema considerando diversos canais a cada iteração como para um sistema com canal fixo. Posteriormente, a Seção 3.2 mostra o desempenho dos algoritmos de alocação citados anteriormente considerando canais fixos.

3.1. Sistema OFDM

Tendo em vista a simulação de um sistema OFDM, um modelo foi construído fazendo-se uso do *software* MATLAB (MATHWORKS). Para isto, uma versão banda base de um sistema OFDM foi implementada, conforme exposto na Seção 2.2.1, Figura 2.12.

A escolha por um modelo deste tipo se deu pela possibilidade de estudar as etapas principais da técnica de OFDM e pela maior facilidade de implementação. Sendo assim, nota-se que também é possível estudar as técnicas de alocação de recursos em sistemas OFDM banda base. Conforme citado, a parte de resultados deste trabalho possui duas etapas principais, onde a primeira consiste da simulação de um sistema OFDM (obtenção de curvas de BER $\times E_b/N_0$, onde E_b é a energia de bit) em MATLAB.

Neste âmbito, as análises de sistemas de comunicação digital geralmente consideram o desempenho medido em função da razão de E_b/N_0 , o que possibilita uma comparação mais justa entre diferentes esquemas de modulação, independente da banda do sistema. Desta maneira, definindo a SNR como:

$$SNR = \frac{P_{\text{sinat}}}{P_{\text{ruído}}} = \frac{E_s/T_s}{N_0 W} = \frac{E_s R_s}{N_0 W} \quad (116)$$

Onde R_s representa a taxa de símbolo do sistema, sabe-se que a taxa de bit $R_b = R_s \log_2 M$ e também $E_b = E_s / \log_2 M$, de tal forma que:

$$SNR = \frac{E_s R_s}{N_0 W} = \frac{E_b R_b}{N_0 W} \quad (117)$$

Considerando pulsos de Nyquist ideais, com fator de *roll-off* α nulo (LATHI e DING, 2009), sabe-se que:

$$SNR = \frac{E_s R_s}{N_o W} = \frac{E_s R_s}{N_o (1 + \alpha) R_s} = \frac{E_s}{N_o} \quad (118)$$

De tal forma que:

$$SNR = \log_2 M \frac{E_b}{N_o} \quad (119)$$

É comum, em algoritmos de alocação de recursos, encontrar curvas de BER em função da SNR, visto que esta varia para cada subportadora. Entretanto, para análises de sistemas que empregam diferentes tipos de modulações e codificação, a análise em função de E_b/N_o pode se tornar mais adequada. É por este motivo que (119) é considerada como relação importante a partir de então e estabelece uma relação tanto do ponto de vista da SNR como de E_b/N_o .

Sendo assim, considerando o modelo da Figura 2.12, foi possível construir um código em MATLAB para simular o comportamento de um sistema OFDM banda base considerando diferentes situações para canais de multipercurso, bem como desempenho em AWGN. Este algoritmo possui como entrada diversos parâmetros, fornecidos pelo próprio usuário que deseja simular o sistema.

Existem duas versões do algoritmo que simula o sistema OFDM, são elas: análise das curvas de desempenho considerando iterações para diferentes canais simulados, e análise da curva de desempenho para um canal fixo. A primeira destas permite que uma visão geral do desempenho do sistema seja obtida dependendo dos parâmetros de simulação, enquanto a segunda permite a observação mais detalhada do comportamento do sistema para um canal fixo.

Outro ponto importante e considerado nas simulações é que o canal não varia para cada *frame* (conjunto de símbolos OFDM) transmitido em uma dada iteração. Desta maneira, a análise é feita em desvanecimento seletivo em frequência sem que o efeito Doppler seja considerado.

A Figura 3.1 expõe como é realizada a entrada dos parâmetros do algoritmo via console do MATLAB.

Entrada dos Parâmetros OFDM

Entre com o número de subportadoras (Default: $N_c = 64$):
 Entre com o número de símbolos OFDM que serão enviados (Default: $n_{\text{simbs}} = 10$):
 Entre com o número de iterações de canais para cada SNR (Default: $n_{\text{iter}} = 400$):

Entrada dos Parâmetros de Modulação

Entre com o tipo de modulação (qam ou psk) (Default: qam):
 Entre com a ordem da modulação (M) (Default: 16):

Entrada dos Parâmetros de Canal

Entre com o tipo de canal (0 para apenas AWGN, 1 para AWGN + Multipercurso) (Default: 1):
 Entre com o número de taps do canal (Default: 5):
 Entre com o fator de atenuação exponencial do canal (Default: 2):
 Entre com o tipo de equalização de canal (1 - ZF (Default), 2 - MMSE):

Entrada dos Parâmetros de Codificação

Entre com o tipo de codificação desejada:
 n para Nenhuma (Default)
 r para Repetição
 h para Hamming
 c para Convolutacional:
 Deseja simular Interleaving? (1 para sim, 0 para não - default):

Entrada do Parâmetro de Eb/No

Entre com o intervalo de simulação de EbNo (em dB) (Default : 0:2:30):

Figura 3.1: Entrada de parâmetros OFDM para um exemplo de simulação.

Primeiramente, o número de subportadoras é escolhido, bem como o número de símbolos OFDM a serem enviados e quantos canais serão simulados para os diferentes valores de SNR propostos. Também é possível escolher o tipo de modulação utilizado, bem como sua ordem.

Com relação ao canal, pode-se optar por analisar o desempenho da técnica de OFDM em canais AWGN ou de multipercurso. Especificamente para canais com multipercurso, o número de *taps* (equivalente ao número de percursos possíveis) é especificado, bem como o fator de atenuação exponencial (relacionado a efeitos de larga escala em canais de radiopropagação). Os coeficientes de cada *tap* são gerados de acordo com uma distribuição de Rayleigh. Também é possível equalizar o sinal utilizando duas técnicas diferentes, que serão discutidas posteriormente nesta subseção.

Por fim, pode-se simular o sistema sem código corretor de erro ou empregando códigos de repetição, Hamming ou convolutacional, com ou sem

interleaving. É possível notar que a análise é feita para um intervalo de E_b/N_o especificado pelo usuário do algoritmo.

A interface para o algoritmo de canal fixo é semelhante, com exceção de não existir a entrada para quantidade de simulações de canais por SNR.

Uma análise resumida de como o código está estruturado é realizada a partir de então. Após a entrada dos parâmetros já discutidos, um cálculo simples da SNR é realizado. Existem duas situações a serem consideradas:

- Se a análise é realizada considerando apenas o efeito do AWGN, a SNR pode ser expressa de acordo com (119). Em dB, tem-se a seguinte relação:

$$SNR(dB) = 10 \log_{10}(\log_2 M) + 10 \log_{10} \frac{E_b}{N_o} \quad (120)$$

- Ao considerar os efeitos de multipercurso e conseqüentemente a adição de prefixo cíclico (não necessário no caso AWGN), pode-se adicionar um fator à relação anterior, de tal forma que:

$$SNR(dB) = 10 \log_{10}(\log_2 M) + 10 \log_{10} \frac{E_b}{N_o} + 10 \log_{10} \left(\frac{N}{N + N_g} \right) \quad (121)$$

Além disto, se algum tipo de código corretor de erro é utilizado, a SNR também deve ser ajustada de maneira a considerar o fator de codificação. Por exemplo, se um código de repetição com $\rho = 1/3$ é considerado (ou seja, cada bit é repetido três vezes), este valor é compensado na SNR, de tal forma que:

$$SNR(dB) = 10 \log_{10}(\log_2 M) + 10 \log_{10} \frac{E_b}{N_o} + 10 \log_{10} \left(\frac{N}{N + N_g} \right) + 10 \log_{10} \rho \quad (122)$$

A partir de então, a simulação para o transmissor OFDM, canal e receptor OFDM é iniciada. Existe um *loop* principal que controla as iterações para os diversos valores de SNR calculados. Vale ressaltar que, no algoritmo que possui simulações para diversos canais diferentes, outro *loop* secundário controla o número de canais diferentes, considerando cada SNR do intervalo desejado.

O primeiro passo do algoritmo é gerar um sinal de mensagem aleatório, constituído de bits 0s e 1s. A partir deste momento, este sinal é codificado e o *interleaving* é aplicado, caso estas opções sejam seleccionadas no momento da simulação (a codificação e o *interleaving* são implementados com o auxílio de funções já disponíveis no MATLAB). A próxima etapa consiste em modular o sinal, utilizando constelações do tipo QAM ou PSK e normalizando-as, de tal maneira que a energia de símbolo seja unitária.

Possuindo o sinal de mensagem modulado, uma transformação de serial para paralelo é realizada, tendo em vista a aplicação da operação de IDFT (implementada pela IFFT no MATLAB). É interessante ressaltar que o resultado da IDFT é multiplicado por \sqrt{N} no transmissor e dividido por $1/\sqrt{N}$ no receptor. Isto é feito tendo em vista a normalização deste cálculo, respeitando o teorema de Parseval. Desta maneira, tem-se uma matriz onde cada coluna representa um símbolo OFDM diferente.

A próxima etapa no transmissor OFDM é a de adicionar o CP, caso o canal seja de multipercurso. Para isto, as $L-1$ amostras finais de cada símbolo OFDM são adicionadas ao seu começo, para evitar ISI/ICI. Após isto, basta retornar o sinal para a forma serial e realizar sua transmissão pelo canal.

Conforme citado anteriormente, existem duas opções para simulação de canal, a saber, AWGN ou de multipercurso. Na primeira destas, o sinal transmitido apenas é corrompido por ruído, enquanto na segunda, o canal de multipercurso é considerado.

Com relação ao canal de multipercurso, este é simulado pela soma de duas VAs gaussianas em quadratura, de acordo com a demonstração do Anexo A. Além disto, existe a contribuição do decaimento exponencial, que também ocorre quando um sinal é transmitido por canais de radiopropagação.

Ao passar pelo canal, o sinal chega ao receptor OFDM. As ações realizadas neste momento são opostas às realizadas no transmissor. Desta forma, o sinal é transformado para paralelo novamente, o CP é removido e descartado (para canais com multipercurso) e a DFT (via FFT) é realizada.

Antes de prosseguir para a demodulação, tem-se a equalização no próprio domínio da frequência, caso um canal de multipercurso seja simulado. Esta equalização simples é uma das grandes vantagens da técnica de OFDM, de maneira

que, na ausência de ISI/ICI, uma solução possível é conhecida como *Zero-Forcing* (ZF).

A solução ZF pode ser escrita como:

$$Y_{eq}[k] = H[k]^{-1}Y[k] \quad (123)$$

Onde Y_{eq} representa o sinal recebido equalizado, Y é o sinal OFDM recebido e H a resposta em frequência do canal, considerando o k -ésimo símbolo OFDM. Desta maneira, nota-se que, de acordo com (35), o efeito do AWGN não é considerado na solução ZF. A equalização ZF resulta em:

$$Y_{eq}[k] = H[k]^{-1}Y[k] + H[k]^{-1}W[k] \quad (124)$$

Onde W representa a DFT da contribuição do ruído AWGN, como definido em (35). Isto faz com que, em situações onde a resposta do canal possui valores muito pequenos, o segundo termo amplifique a contribuição do AWGN. Outro método de equalização é baseado na minimização do erro quadrático médio (MMSE). A solução para este método é dada por (VILAR, 2014):

$$Y_{eq}[k] = \frac{H_k^*}{(|H_k|^2 + SNR)} Y[k] \quad (125)$$

Desta maneira, a solução MMSE também leva em consideração a SNR, considerando o ruído AWGN na análise e minimizando o problema encontrado na solução ZF.

Em sequência, a demodulação QAM/PSK é realizada, com posterior remoção do *interleaving* e da codificação, se estes forem utilizados. Por fim, a contagem de erros para representação da curva de BER $\times E_b/N_0$ é realizada e os valores são armazenados para observação gráfica dos resultados obtidos.

Tendo exposto o funcionamento básico do código, os resultados obtidos das simulações são apresentados a partir de então. Primeiramente, uma análise acerca do algoritmo de sistema de OFDM com canal fixo será realizada. Vale ressaltar que a análise deste código tem como principais objetivos a observação do

comportamento do sinal em etapas específicas do algoritmo, sendo que a análise da curva de desempenho ($\text{BER} \times E_b/N_0$) para o sistema é mais precisa com o algoritmo que considera diversas iterações para canais diferentes. Outro ponto importante a ser observado é que a relação entre E_b/N_0 considerada é simples e dada por (122), em âmbito geral.

Sendo assim, um exemplo considerando multipercurso será comentado, de maneira a explicitar as etapas de interesse do código. Para este exemplo, os parâmetros utilizados são expostos na Tabela 3.1, a seguir.

Tabela 3.1: Parâmetros utilizados para simulação de um sistema OFDM com canal fixo.

Nome do Parâmetro	Valor do Parâmetro
<i>Número de subportadoras</i>	64
<i>Número de símbolos OFDM</i>	100000
<i>Modulação e ordem</i>	16-QAM
<i>Número de taps do canal</i>	5
<i>Fator de atenuação exponencial</i>	2
<i>Equalização</i>	ZF
<i>Codificação</i>	Nenhuma

O canal simulado com cinco componentes de multipercurso (*taps*) pode ser representado vetorialmente por:

$$\mathbf{h} = [-0.2167 + 0.2059i, -0.6404 + 0.1817i, -0.2450 - 0.6005i, 0.0865 + 0.1558i, 0.1117 + 0.0517i] \quad (126)$$

Com estes parâmetros, as etapas do algoritmo citadas no início desta seção são seguidas. Nesta seção de resultados, uma análise gráfica breve será realizada para o caso de canal fixo e desempenho do sistema OFDM. A resposta em frequência do canal simulado neste exemplo é dada pela DFT das amostras considerando uma janela igual ao número de subportadoras (no caso, 64). A Figura 3.2 mostra a resposta em frequência deste canal, com módulo e fase.

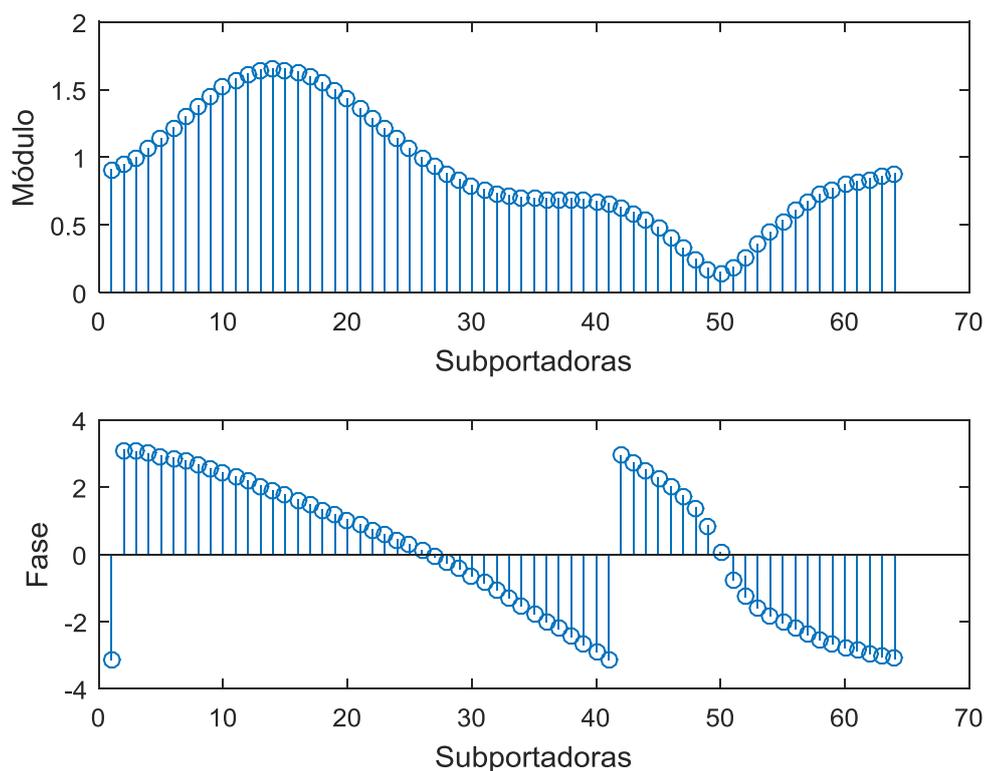


Figura 3.2: Módulo (acima) e fase (abaixo) da resposta em frequência para o canal simulado neste exemplo.

Ao observar o módulo da resposta em frequência, nota-se que as primeiras subportadoras apresentam ganho, enquanto que, próximo da subportadora de índice 50, existe uma grande atenuação do sinal imposta pelo canal. Ao passar por este canal de transmissão, a saída é dada pela convolução entre o sinal modulado e a resposta ao impulso do canal, conforme exposto anteriormente. Se existe o uso de CP, a convolução se torna circular.

O algoritmo em questão identifica as melhores e piores subportadoras, baseando-se na taxa de erro de símbolo (em inglês, *Symbol Error Rate* ou SER). Sendo assim, é possível comparar, no receptor, os sinais recebidos antes e depois da equalização no domínio da frequência, considerando a melhor e a pior subportadoras, uma subportadora com desempenho médio e uma composição de todas. A Figura 3.3 mostra as constelações 16-QAM recebidas antes da equalização ZF, considerando uma parte dos símbolos QAM e SNR de 20dB.

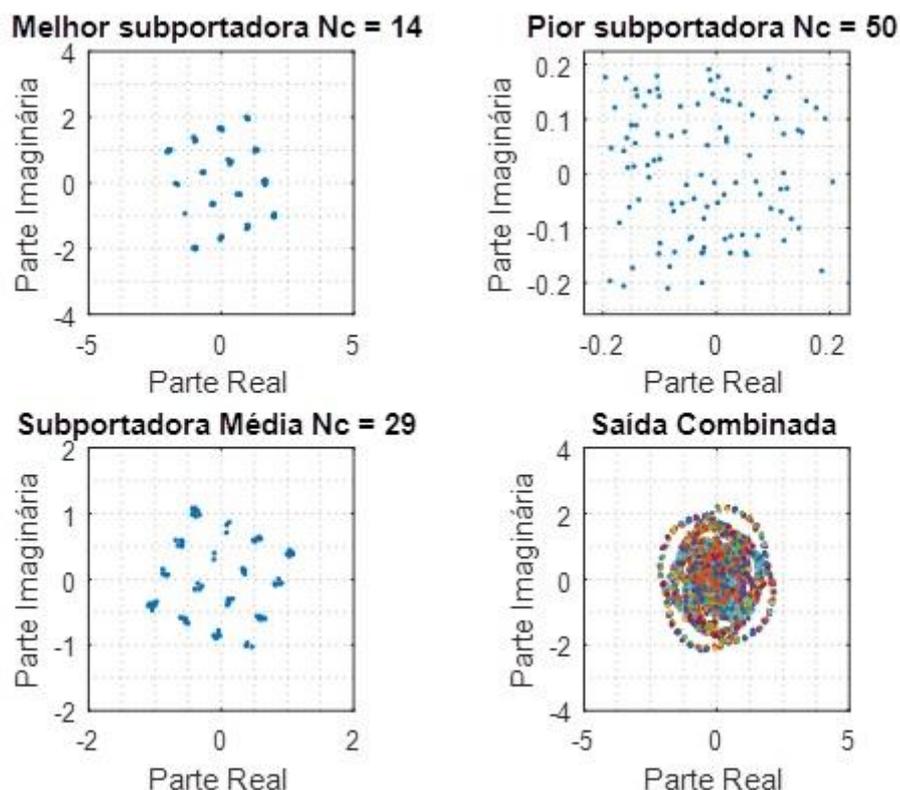


Figura 3.3: Constelações 16-QAM recebidas antes da equalização ZF, SNR = 20dB.

Ao analisar a figura anterior, nota-se que as subportadoras selecionadas estão de acordo com o comportamento observado no gráfico superior da Figura 3.2. Observando as constelações 16-QAM e comparando-as com a da Figura 2.14, observa-se que existem desvios de fase provocados pelo canal de multipercurso, além de atenuação.

Para a 14ª subportadora, tem-se a menor SER do sistema. Neste âmbito, nota-se que, os intervalos de decisão encontram-se bem definidos e basta que uma compensação de fase correta seja imposta pela equalização para que os símbolos sejam demodulados corretamente. Também se observa um ganho imposto pelo canal.

Para o caso da pior subportadora, de índice 50, existe grande atenuação por parte do canal, o que é possível de ser observado pelos eixos x e y do gráfico correspondente. Além disto, há desvio de fase, visto que o comportamento é muito diferente de uma constelação 16-QAM. Desta forma, os intervalos de decisão não estão bem definidos e a equalização desta subportadora se torna difícil nas condições deste exemplo.

A subportadora com SER média (índice 29) apresenta um bom desempenho neste exemplo, de tal maneira que uma compensação de fase também possibilita uma boa detecção. Vale ressaltar, entretanto, que embora seja possível observar os intervalos de decisão corretamente, os símbolos se encontram um pouco mais espalhados do que no caso da melhor subportadora.

Por fim, o gráfico que apresenta a superposição das respostas das 64 subportadoras mostra que o comportamento, na média, é muito diferente de uma constelação 16-QAM. Também é possível observar neste gráfico a atenuação de algumas subportadoras e os desvios de fase acarretados pelo canal de multipercurso.

Após equalização ZF, a Figura 3.4 mostra as mesmas constelações no receptor. Nota-se a correção de fase imposta em todos os casos, de tal maneira a rotacionar a constelação para sua forma original. Para os casos da melhor subportadora e da subportadora com desempenho médio, têm-se bons resultados, que possibilitam correta decisão dos símbolos 16-QAM para este valor de SNR.

Por outro lado, mesmo com a equalização ZF, a pior subportadora ainda apresenta um desempenho pobre e interfere negativamente no desempenho geral do sistema OFDM. Os intervalos de decisão não se encontram bem definidos, de tal maneira que a decisão acarreta em erros de símbolos.

Por fim, a saída combinada passa a exibir um comportamento mais semelhante ao de uma constelação 16-QAM, o que indica uma grande melhoria do sistema considerando a aplicação de equalização no domínio da frequência e demonstra a primordial importância deste processo em um sistema OFDM.

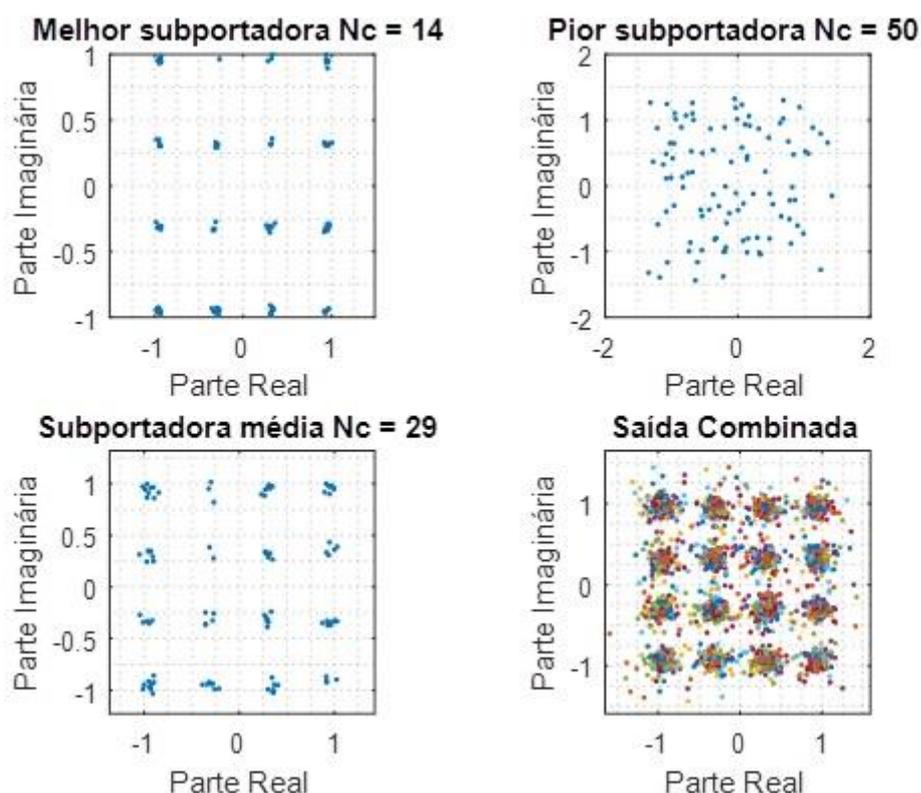


Figura 3.4: Constelações 16-QAM recebidas após equalização ZF, SNR = 20dB.

A análise com canal fixo permite observar que algumas subportadoras podem comprometer muito o desempenho de um sistema OFDM. É por este motivo que algoritmos de alocação de recursos são desejados, na prática, já que cada subportadora apresenta diferentes atenuações impostas pelo canal de multipercurso.

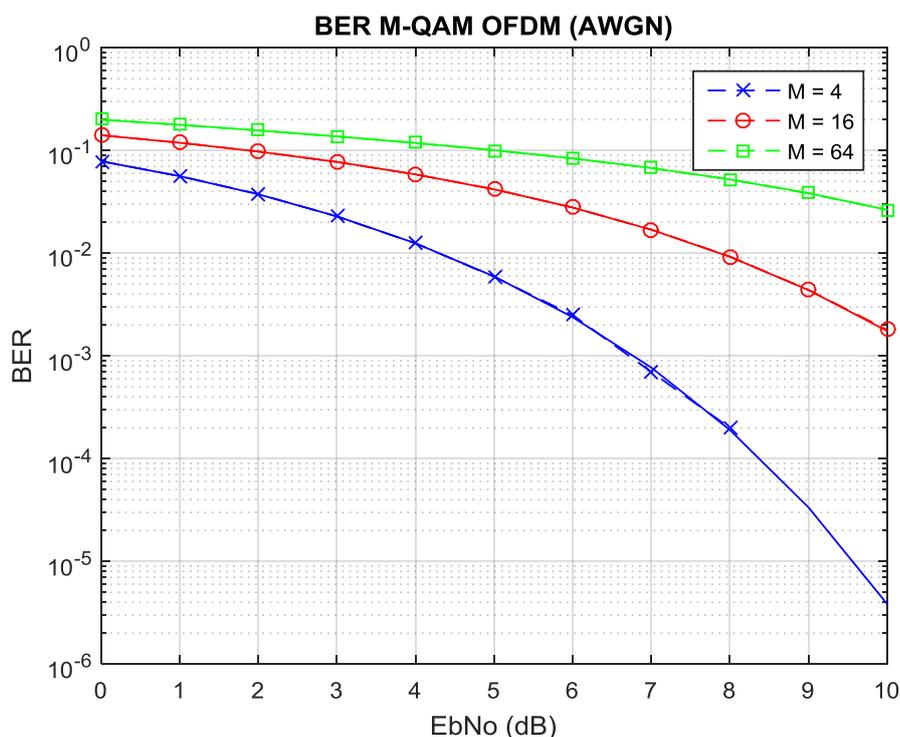
Entretanto, obter uma estimativa do desempenho de um sistema considerando apenas o caso de um canal não é interessante, visto que cada canal possui uma resposta em frequência diferente, o que leva a diferentes características no momento da recepção.

A partir deste momento, a análise do algoritmo com diversos canais será realizada. Neste sentido, as curvas de BER $\times E_b/N_0$ serão expostas, considerando as diversas situações possíveis de simulação. A Tabela 3.2 mostra os parâmetros básicos de simulação considerados.

Tabela 3.2: Parâmetros utilizados para simulação de um sistema OFDM com diversas iterações de canais.

Nome do Parâmetro	Valor do Parâmetro
Número de subportadoras	64
Número de símbolos OFDM por canal	10
Número de canais por iteração por SNR	1000 a 100000
Modulação e ordem	M-QAM e M-PSK
Tipos de canais	Apenas AWGN e Multipercurso
Número de taps do canal	5
Fator de atenuação exponencial	2
Equalização	ZF e MMSE
Codificação	Nenhuma, Repetição, Hamming e Convolutacional

A partir de então, diversas curvas de desempenho variando os parâmetros da tabela anterior serão apresentadas. Iniciando por canais AWGN apenas, o desempenho da técnica de OFDM foi analisado sem considerar o multipercurso, para comparação das curvas de BER. A Figura 3.5 mostra os resultados obtidos para M-QAM, com $M = 4, 16$ e 64 , equalização ZF, sem codificação e E_b/N_0 variando de 0 a 10dB (com passos de 1dB).

**Figura 3.5:** Desempenho de um sistema OFDM apenas considerando ruído AWGN.

Na figura anterior, as curvas com linha sólida representam os resultados teóricos, derivados de (47), enquanto as curvas com marcações e tracejadas representam os resultados simulados. Como é possível observar, a técnica de OFDM aplicada em AWGN apresenta o mesmo resultado das curvas teóricas.

Um ponto interessante é que, à medida que a ordem da modulação aumenta, ocorrem maiores erros de bit. Isto ocorre pois há uma troca de eficiência de potência por eficiência de banda. Desta maneira, transmitem-se mais bits/s/Hz de informação, porém o desempenho do sistema diminui para um mesmo valor de E_b/N_0 . Este comportamento é comum em sistemas de modulação digital.

Como o ponto principal desta seção está relacionado a canais de multipercurso, a Figura 3.6 mostra o desempenho do sistema considerando as mesmas modulações da figura anterior, porém em canais com multipercurso. Nestas simulações, são utilizadas equalização ZF e não há codificação. Além disto, o intervalo de E_b/N_0 é aumentado para 30dB (com passos de 3dB).

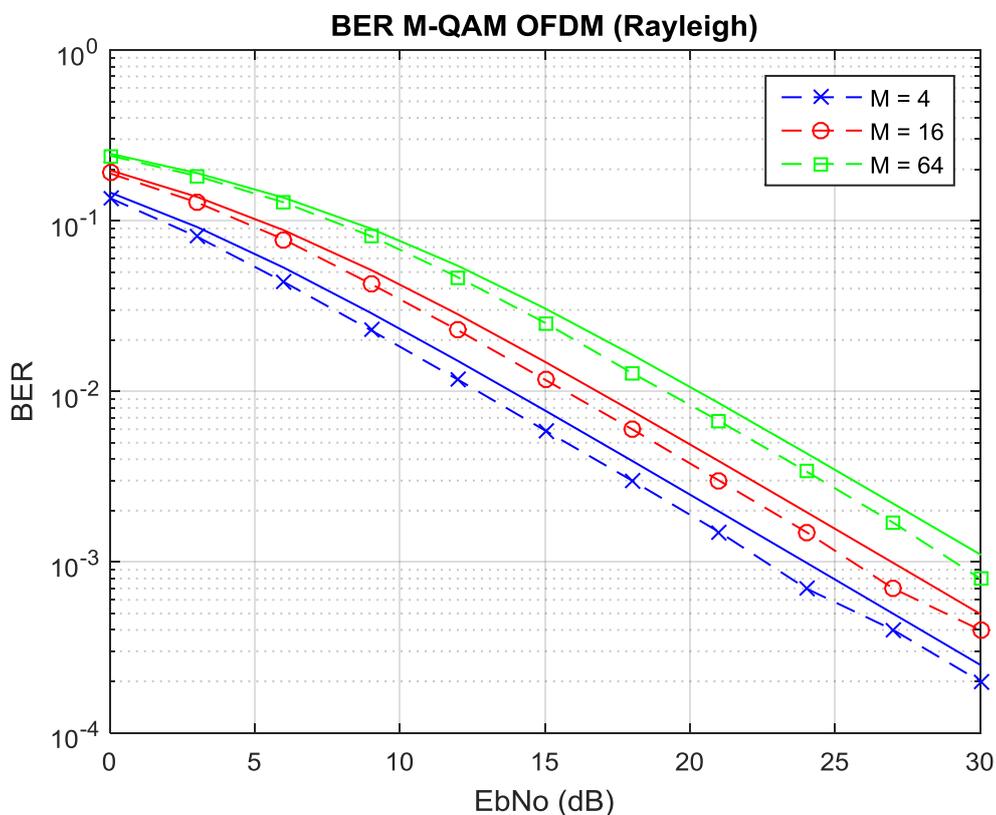


Figura 3.6: Desempenho de um sistema OFDM considerando multipercurso.

Ao observar a figura anterior, nota-se que o sistema OFDM em condições de multipercurso apresenta uma BER muito mais elevada do que no caso AWGN, para um mesmo valor de E_b/N_0 . Outro comportamento interessante é que, comparando as curvas teóricas (linhas sólidas) com as curvas simuladas (tracejadas com marcações), observa-se uma pequena diferença de desempenho, que se mantém para as diferentes ordens de modulação. Esta diferença está relacionada ao tamanho do CP e sua compensação na SNR, conforme exposto em (121). Sendo assim, ao aumentar o número de percursos do canal, o desempenho do sistema aproxima-se mais da curva teórica gerada pelo MATLAB.

Uma característica para o correto funcionamento da técnica de OFDM é o tamanho do CP. Conforme exposto na Seção 2.2.1, o mínimo tamanho do CP para prevenir ISI/ICI é $L - 1$. Considerando um canal com cinco percursos, isto significa que o limite mínimo é de quatro amostras para o CP. A Figura 3.7 mostra o que ocorre com a curva de desempenho do sistema quando $N_g = 4$ e $N_g = 3$. Neste exemplo, é considerada a modulação 16-QAM com equalização ZF e sem codificação do sistema.

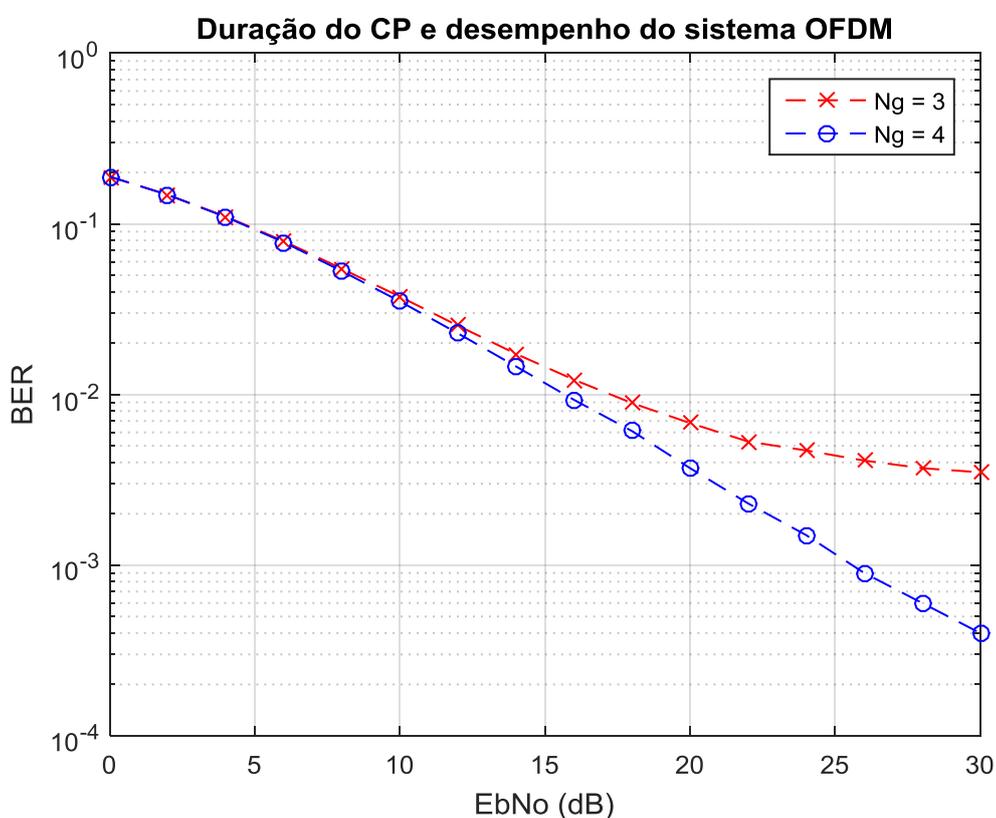


Figura 3.7: Desempenho de um sistema OFDM considerando diferentes tamanhos de CP.

A perda de desempenho do sistema com $N_g = 3$ fica clara neste exemplo (a ISI passa a dominar a queda do desempenho do sistema, ao invés do ruído). Com apenas mais uma amostra no CP, alcançando o limite mostrado na modelagem matemática, tem-se um comportamento semelhante ao da Figura 3.6, que está próximo da curva teórica.

Até então, curvas de desempenho baseadas no algoritmo de várias iterações de canais foram analisadas. Na verdade, a saída do algoritmo, cuja interface para as entradas foi exposta na Figura 3.1, representa o desempenho do sistema baseado em um tipo de ordem de modulação. A Figura 3.8 mostra um exemplo para 8-PSK, canal de multipercurso, equalização ZF e sem codificação, apenas para observação das curvas representadas no gráfico.

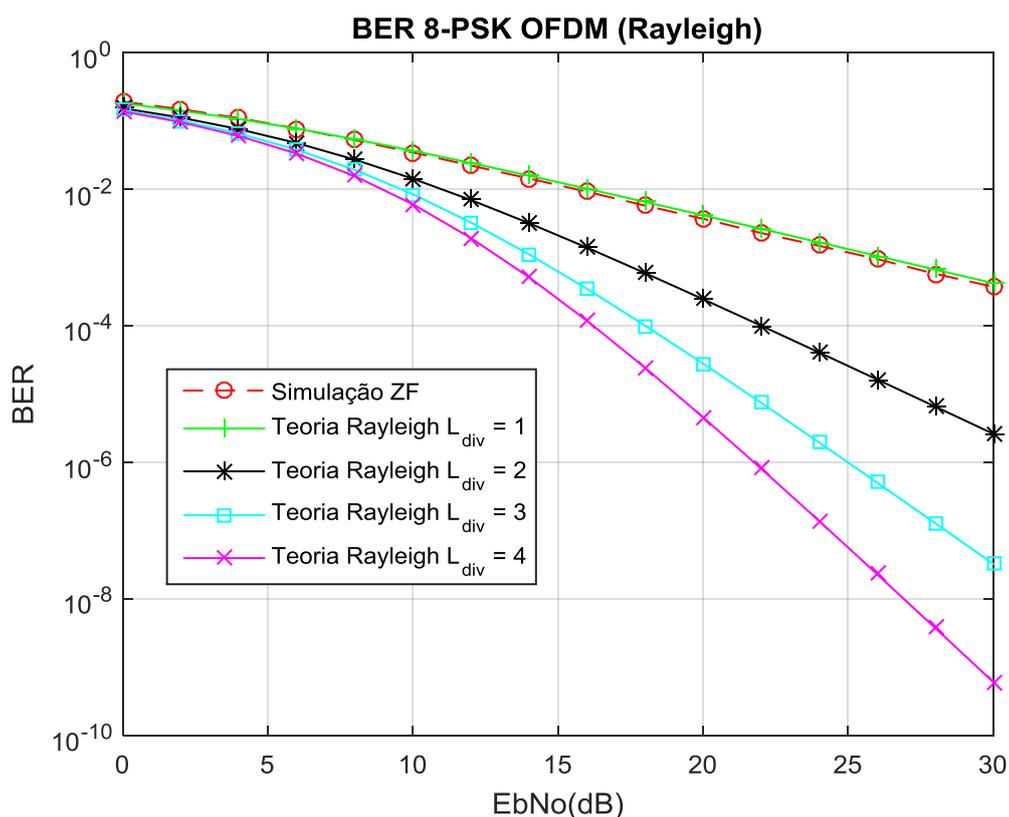


Figura 3.8: Desempenho de um sistema OFDM considerando multipercurso e modulação 8-PSK.

Nota-se que o resultado obtido é muito semelhante à curva teórica tracejada verde. Até então, foi considerada apenas uma antena na recepção do sinal e uma antena transmissora, o que caracteriza uma diversidade (L_{div}) unitária. Entretanto, é possível utilizar esquemas MIMO que aumentam a diversidade de maneira a

melhorar o desempenho do sistema (CHOO, KIM, *et al.*, 2010). Este comportamento é mostrado na figura anterior e, embora não seja o foco deste projeto, é interessante de ser constatado em análise gráfica com o auxílio do MATLAB.

Antes de analisar o sistema OFDM com codificação, uma breve comparação entre as equalizações ZF e MMSE em sistemas OFDM considerando multipercurso é exposta. Vale ressaltar que, nos exemplos realizados até então, apenas a equalização ZF foi utilizada. Para auxiliar na análise, o código com canal fixo, utilizando o mesmo canal da Figura 3.2 e 16-QAM (sem codificação) é utilizado. No entanto, considera-se uma SNR de 10dB nesta comparação.

A Figura 3.9 mostra as subportadoras equalizadas com ZF, enquanto a Figura 3.10 mostra as subportadoras equalizadas com MMSE.

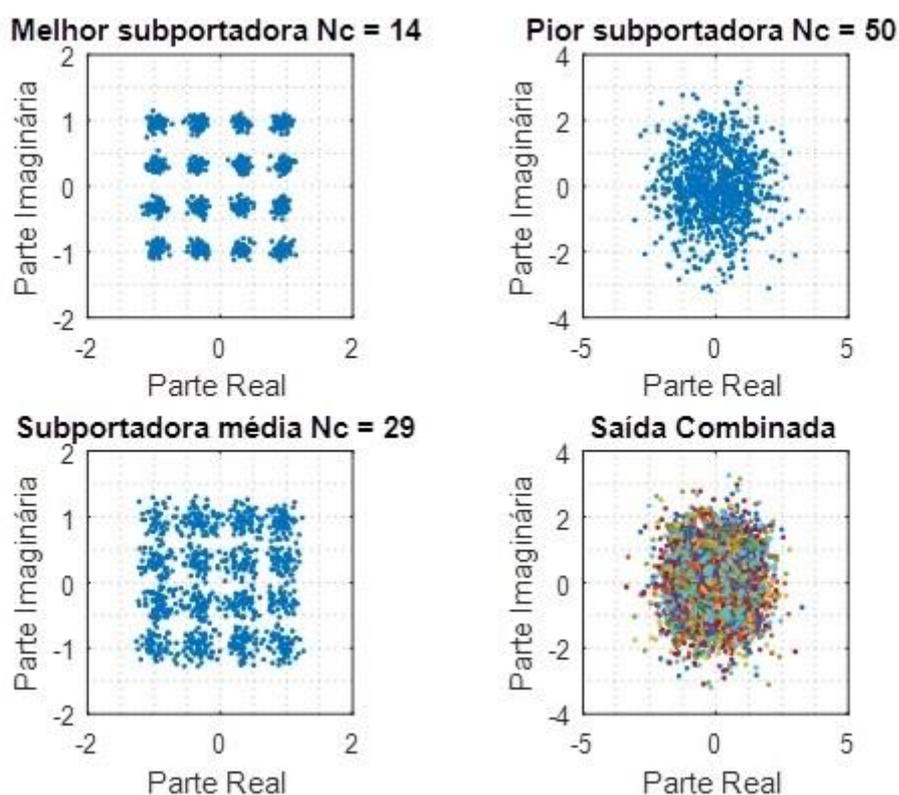


Figura 3.9: Constelações 16-QAM equalizadas com ZF, SNR = 10dB.

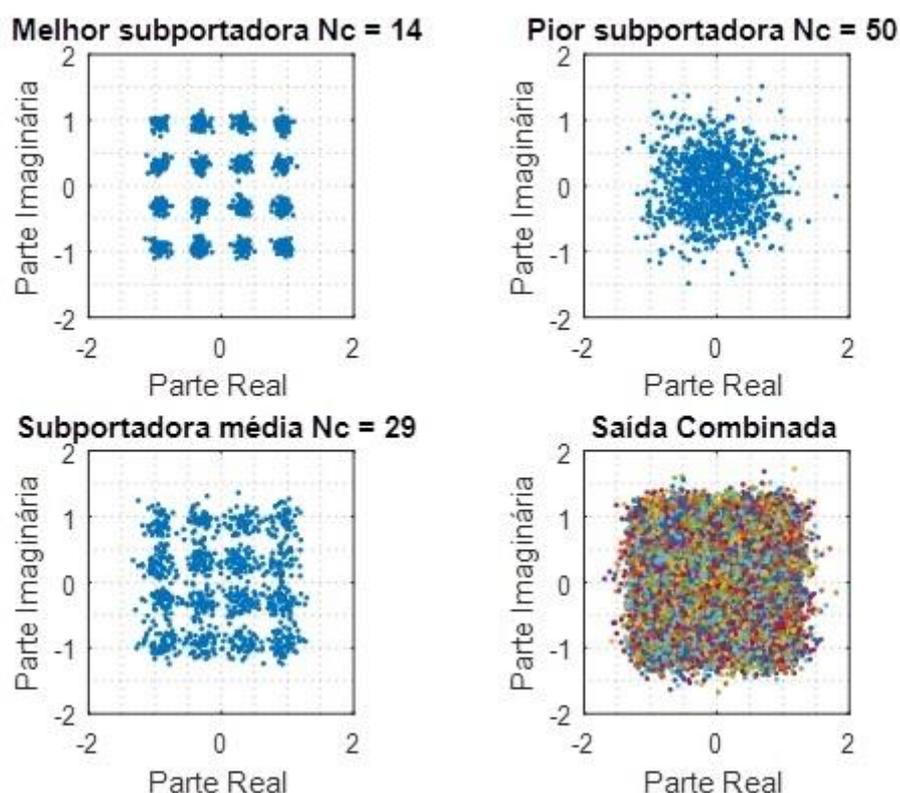


Figura 3.10: Constelações 16-QAM equalizadas com MMSE, SNR = 10dB.

O comportamento das duas figuras é semelhante, tendo em vista a melhor subportadora e a subportadora com desempenho médio. Entretanto, conforme citado anteriormente nesta seção, em situações de nulo ou ganho muito pequeno para a respectiva subportadora, a solução ZF tende a amplificar o ruído. Esta situação é percebida para a pior subportadora, observando as escalas do eixo x e y e comparando-as para as Figuras 3.9 e 3.10 (VILAR, 2014).

Entretanto, deve-se ressaltar que, no momento da decisão, ambas as soluções apresentam praticamente o mesmo desempenho, salvo exceções em que o ganho e a SNR interfiram demasiadamente na amplificação do ruído.

A Figura 3.11 mostra uma comparação para as curvas de desempenho considerando 16-QAM e as equalizações ZF e MMSE. Nota-se que o algoritmo que simula diversos canais é utilizado, tendo em vista a obtenção de uma resposta mais confiável.

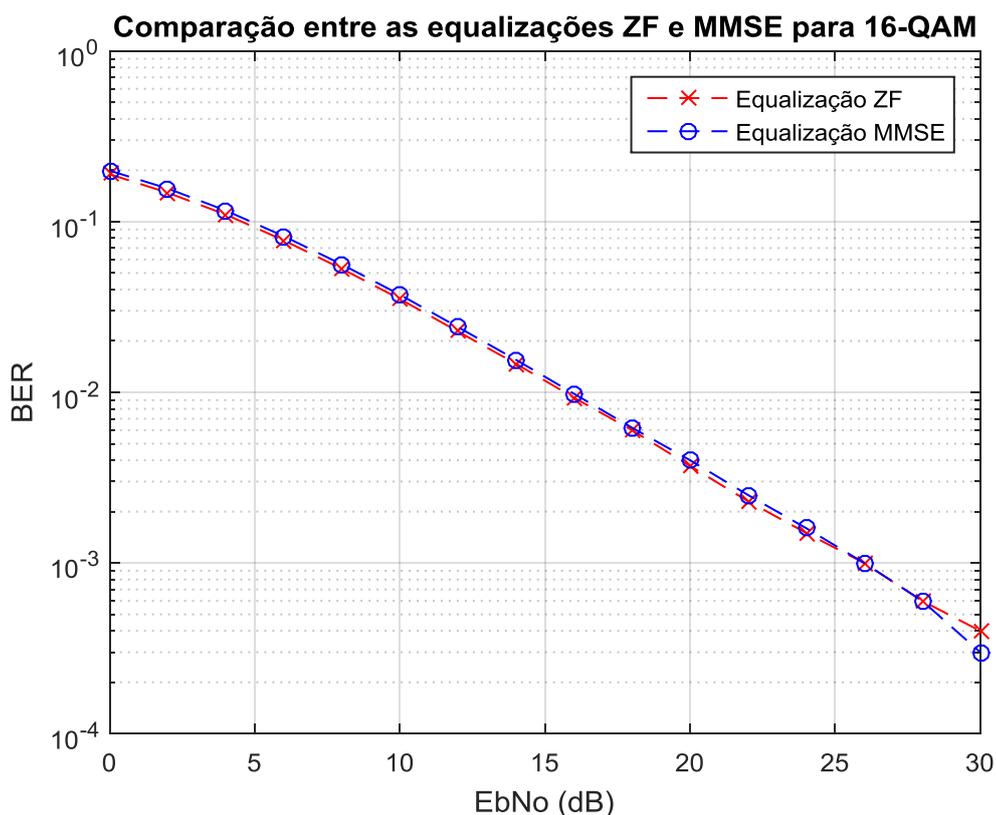


Figura 3.11: Desempenho de um sistema OFDM considerando equalização ZF e MMSE (16-QAM).

De fato, ambas as equalizações apresentam basicamente o mesmo resultado no desempenho do sistema (WANG e BLOSTEIN, 2003). Isto também é visualizado para outros tipos de constelações M -QAM e M -PSK simuladas. Este comportamento era esperado, dadas as comparações realizadas anteriormente.

Outra análise pertinente nesta seção é o uso de codificação em sistemas OFDM, de maneira a melhorar o desempenho geral e alcançar curvas de BER melhores do que as apresentadas até o momento. Conforme exposto na Tabela 3.2, a simulação contempla códigos de repetição, Hamming e convolucional, cujo funcionamento básico foi exposto na Seção 2.3. Estes tipos de codificações serão analisados a seguir, considerando 16-QAM e equalização ZF.

Iniciando pelo código de repetição, a Figura 3.12 mostra uma análise comparativa das curvas de desempenho do sistema OFDM considerando a repetição de 2 ($\rho = 1/2$), 3 ($\rho = 1/3$) e 5 bits ($\rho = 1/5$). Nesta figura, a curva de teoria apresenta o sistema sem nenhum tipo de codificação, considerando a modulação utilizada.

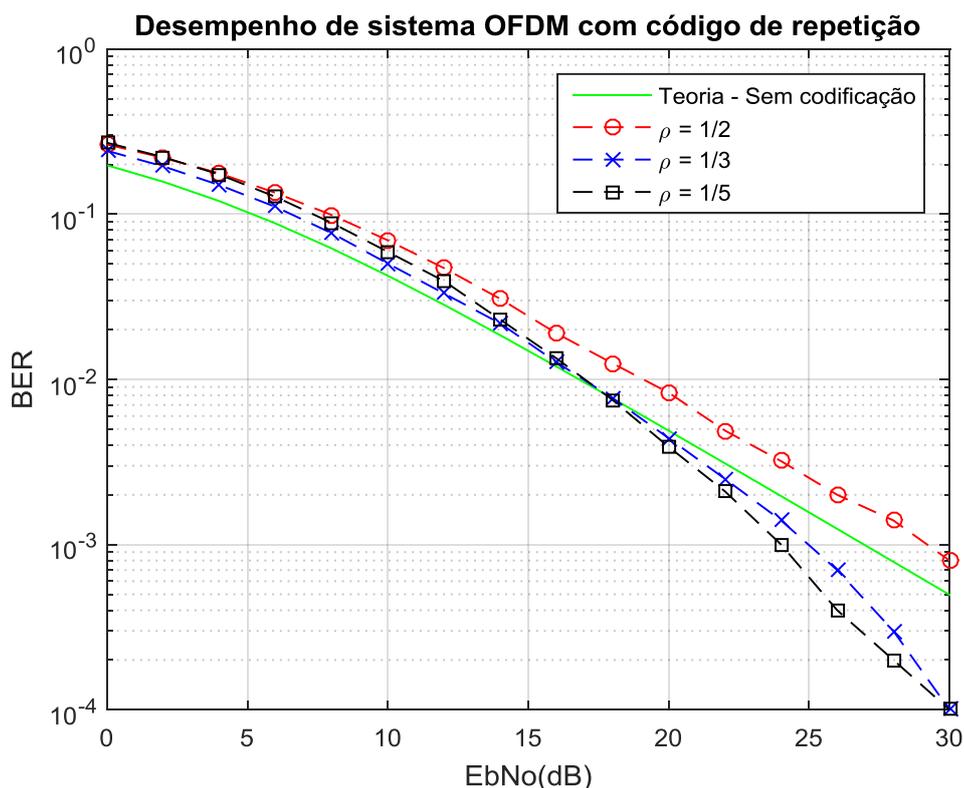


Figura 3.12: Desempenho de um sistema OFDM com codificação de repetição.

Ao observar a figura anterior, nota-se que, para $\rho = 1/2$, o desempenho do sistema é pior do que a curva original. Isto era esperado, já que utilizar apenas um bit de repetição acarreta em uma escolha aleatória se a palavra recebida é 10 ou 01, ou seja, é possível apenas a detecção de um bit de erro. Por outro lado, para $\rho = 1/3$ e $\rho = 1/5$, nota-se que o desempenho do sistema só melhora com relação à curva teórica para valores de E_b/N_0 elevados, o que mostra que o código de repetição é uma alternativa muito simples e que pode não ser atrativa do ponto de vista de melhoria do sistema. Mesmo assim, este tipo de código, por sua simplicidade, é interessante para observar que o desempenho do sistema pode ser melhorado de alguma maneira, se códigos corretores de erro mais sofisticados forem utilizados.

Um fato importante é que para $\rho = 1/3$ é possível corrigir erros em apenas um bit e detectar erros de até dois bits. Por outro lado, se $\rho = 1/5$, pode-se corrigir erros de até dois bits e detectar erros de até quatro bits.

A Figura 3.13 mostra a aplicação de códigos de Hamming com $\rho = 4/7$, $\rho = 11/15$ e $\rho = 31/26$. Novamente, a curva de teoria representa o sistema sem codificação.

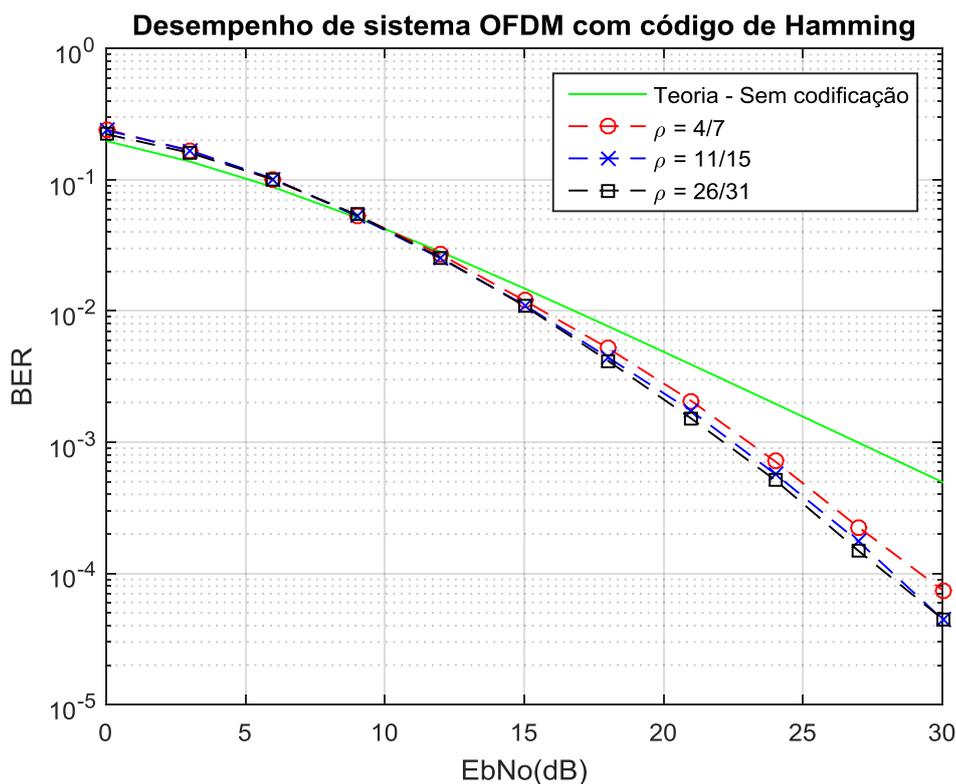


Figura 3.13: Desempenho de um sistema OFDM com codificação de Hamming.

Analisando esta figura e comparando para o caso de código de repetição, nota-se a melhoria alcançada. Enquanto as curvas para os códigos de repetição ultrapassam a curva sem codificação para aproximadamente 18dB, tem-se que as curvas apresentadas para codificação de Hamming ultrapassam em aproximadamente 9dB.

Além disto, é possível observar uma pequena melhoria quando existe o aumento de ρ , já que, quanto maior a taxa de código, maior a energia por bit útil da palavra de código. Também é possível notar uma leve melhoria na SNR do sistema com o aumento de ρ , cuja relação é exposta em (122).

O próximo caso de análise consiste na utilização de códigos convolucionais. Foram implementados dois códigos convolucionais, evidenciados na Figura 3.14. Observando o diagrama à esquerda, tem-se a representação $[1111001_2, 1011011_2]$, que em octal é representada por $[171_8, 133_8]$, com $N_{Conv} = 7$. Por outro lado, o diagrama à direita representa $[7_8, 5_8]$, com $N_{Conv} = 3$.

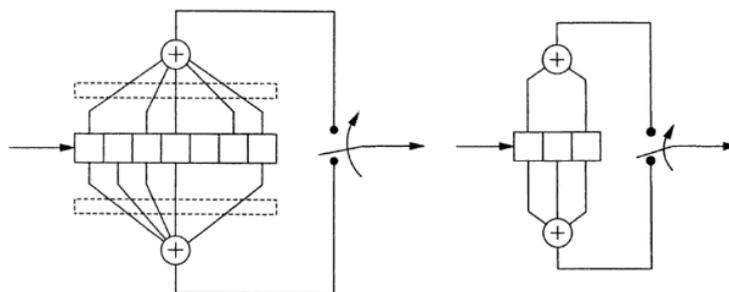


Figura 3.14: Códigos convolucionais utilizados na simulação (WADE, 1994).

No MATLAB, optou-se por utilizar a simulação da decodificação utilizando o algoritmo de Viterbi no modo *hard decision* e truncado, de tal maneira que o registrador de memória inicia zerado, porém não são adicionados $N_{Conv} - 1$ zeros ao final da palavra de mensagem de maneira a possibilitar a passagem do último bit ao final do registrador. A Figura 3.15 mostra a comparação de desempenho para os dois códigos da figura anterior. A curva de teoria expõe o sistema sem codificação.

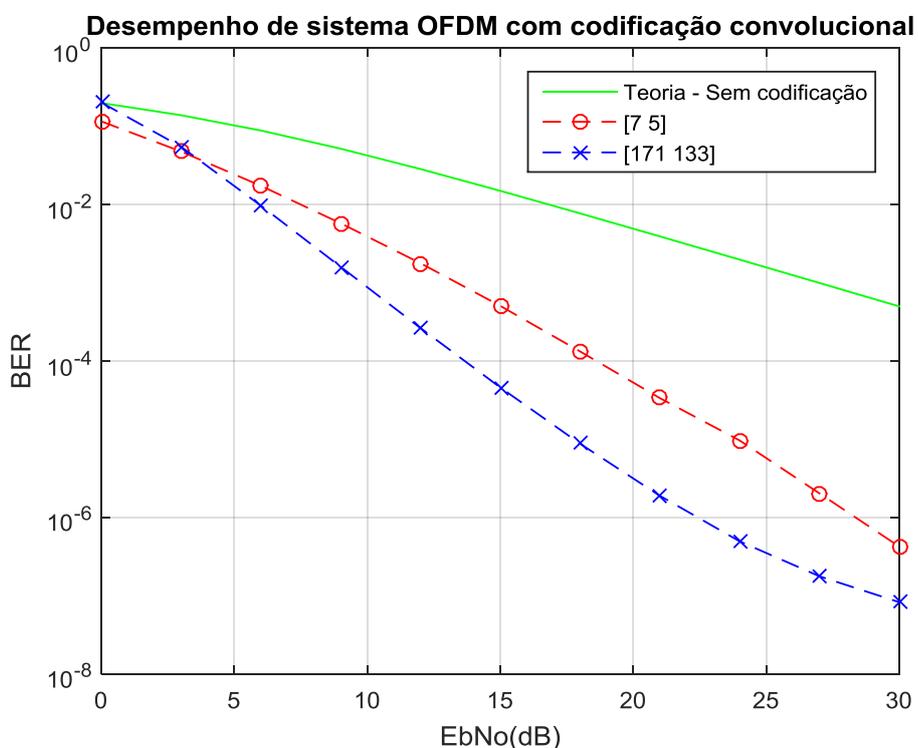


Figura 3.15: Desempenho de um sistema OFDM com codificação convolucionacional.

Da figura anterior, é possível concluir que a codificação convolucional apresenta resultados muito melhores do que os dois códigos de bloco propostos até então. Isto mostra que a concepção do código convolucional, de tal maneira que um bit pode afetar a saída de várias palavras de código, dependendo de N_{Conv} , permite que o sistema lide melhor com erros de bit.

Comparando os dois códigos simulados, observa-se também que o código $[171_8, 133_8]$ possui melhor desempenho frente ao código $[7_8, 5_8]$, já que o registrador do primeiro possui maior memória e conseqüentemente um bit influencia mais palavras de saída. Do ponto de vista de comparação com a curva teórica, ambos apresentam a melhor solução até o momento.

Em conjunto com a codificação, o uso de *interleaving* pode auxiliar a reduzir a probabilidade de erro de bit. Sendo assim, para cada um dos três tipos de códigos simulados, o uso de *interleaving* é proposto a seguir. Nos gráficos, a curva de teoria representa o sistema sem uso de codificação.

A Figura 3.16 mostra uma comparação considerando código de repetição, com $\rho = 1/3$. É possível constatar que a melhoria no desempenho do sistema é evidente, comparando-se o caso sem *interleaving* com o que utiliza esta técnica. Sendo assim, o fato de que surtos de ruído afetam o sistema se torna claro. Neste exemplo, o ganho com *interleaving* é de aproximadamente 10dB para $E_b/N_0 = 30$ dB.

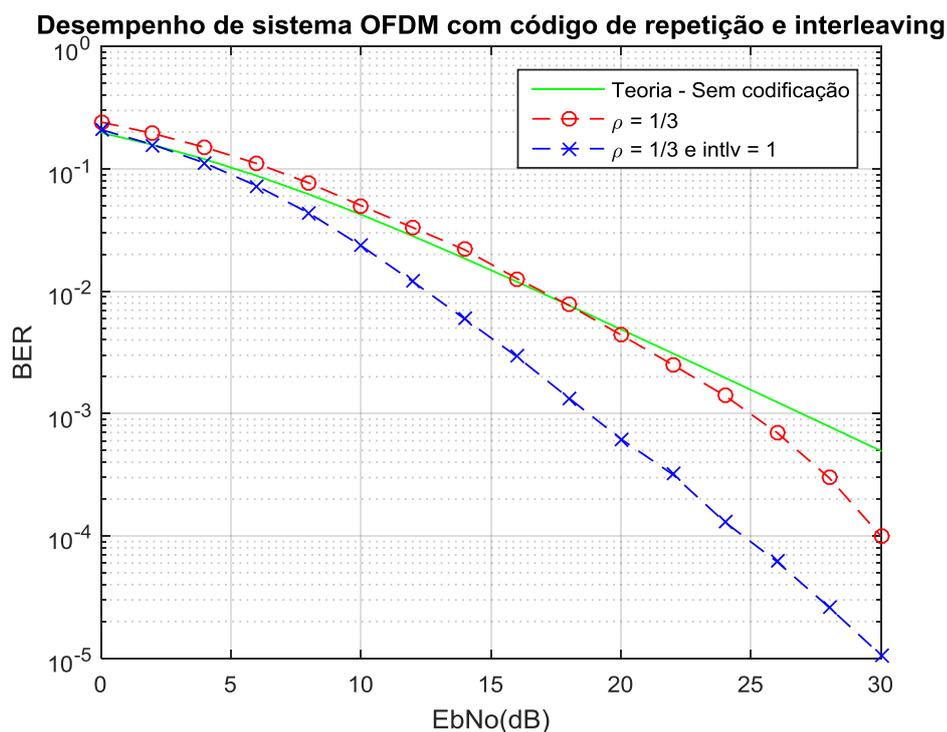


Figura 3.16: Desempenho de um sistema OFDM com código de repetição e *interleaving*.

A Figura 3.17 mostra a aplicação de *interleaving* em um código de Hamming com $\rho = 4/7$. Novamente, ocorre melhoria e o desempenho do sistema OFDM baseado na curva de BER é superior. Nota-se um ganho de aproximadamente 7 dB neste exemplo, também considerando $E_b/N_0 = 30\text{dB}$.

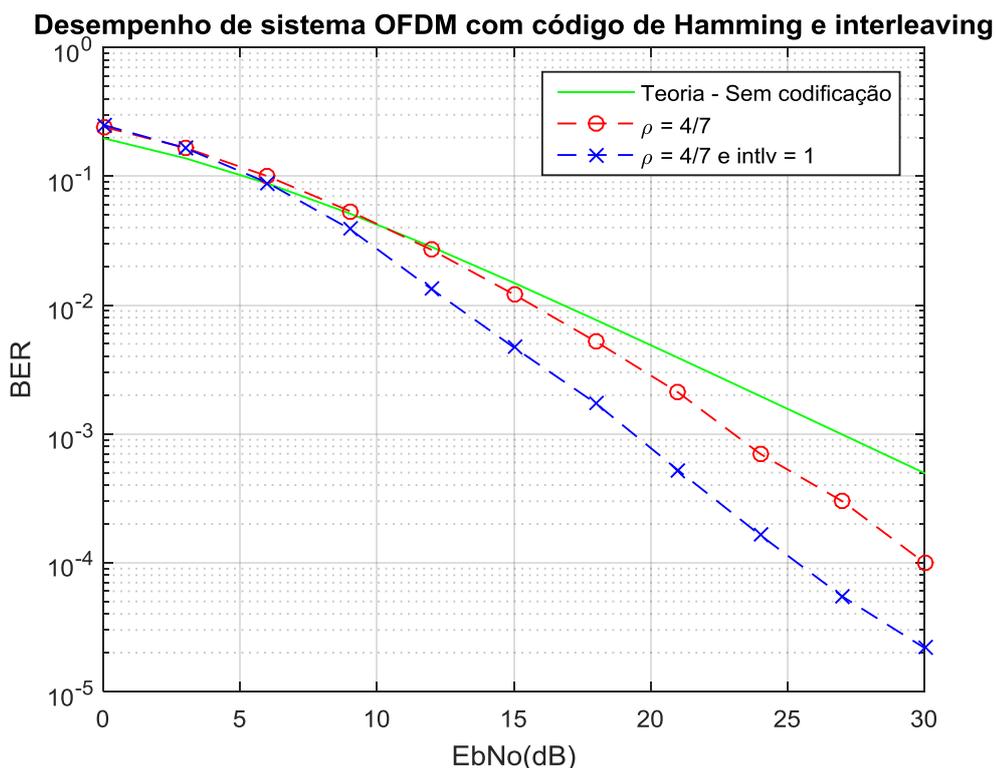


Figura 3.17: Desempenho de um sistema OFDM com código de Hamming e *interleaving*.

Por fim, a Figura 3.18 mostra o *interleaving* aplicado no código convolucional $[171_8, 133_8]$. Desta figura, é possível constatar que também existe melhoria no desempenho do sistema considerando *interleaving*. Por outro lado, nota-se que o ganho é muito menor do que nos casos de codificação por repetição e Hamming, o que leva à conclusão de que os mecanismos de codificação e decodificação convolucional já apresentam, por si só, melhorias com relação a surtos de erros de bit. Isto novamente vem da característica de que um bit de informação influencia N_{Conv} palavras de código geradas.

Desempenho de sistema OFDM com codificação convolucional e interleaving

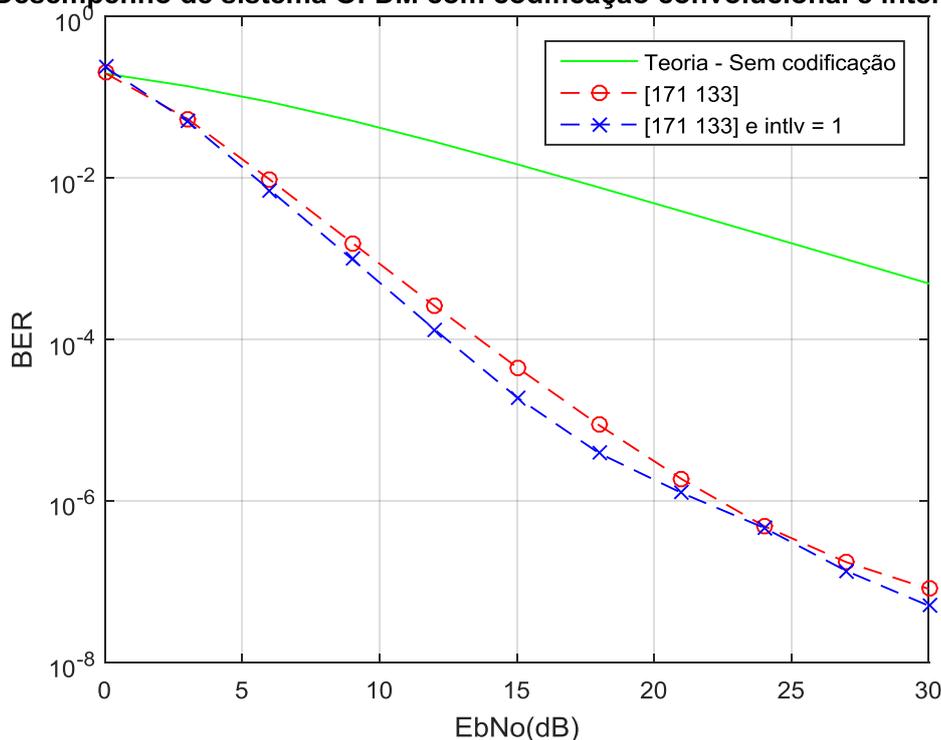


Figura 3.18: Desempenho de um sistema OFDM com código convolucional e *interleaving*.

3.2. Simulação de algoritmos de alocação de recursos

Os algoritmos de alocação de recursos também foram desenvolvidos e simulados individualmente em MATLAB. Sendo assim, todos são representados por funções que recebem parâmetros específicos, como banda do sistema, potência de transmissão, dentre outros. A análise detalhada dos algoritmos será realizada nesta subseção.

Desta maneira, tanto as soluções RA quanto as soluções MA retornam ao usuário as taxas alocadas para cada subcanal e as respectivas potências, dado o fornecimento dos dados necessários para cálculo das SNRs individuais de cada subcanal. Além disto, a solução RA necessita da potência total do sistema OFDM, enquanto a solução MA necessita da taxa normalizada total.

Com estes resultados, é possível construir gráficos de alocação de potência e taxa normalizada em função do índice do canal, comparando o comportamento obtido na alocação com as características do canal simulado. Este é um dos motivos pelo qual um algoritmo com canal fixo para o sistema OFDM é interessante, tendo em vista o entendimento das técnicas aplicadas.

A análise é iniciada pelos algoritmos com solução RA, realizando-se uma comparação entre estes. Entretanto, optou-se por diminuir o número de subportadoras utilizadas para oito, de tal forma a representar as etapas dos algoritmos e facilitar o entendimento. Além disto, os ganhos de cada subcanal considerados são obtidos a partir do módulo da resposta em frequência de (126), ou seja:

$$|\mathbf{H}| = [0.9039, 1.4509, 1.5937, 1.0630, 0.7103, 0.6536, 0.1739, 0.6681] \quad (127)$$

A Figura 3.19 representa os ganhos de cada subportadora considerados nesta subseção.

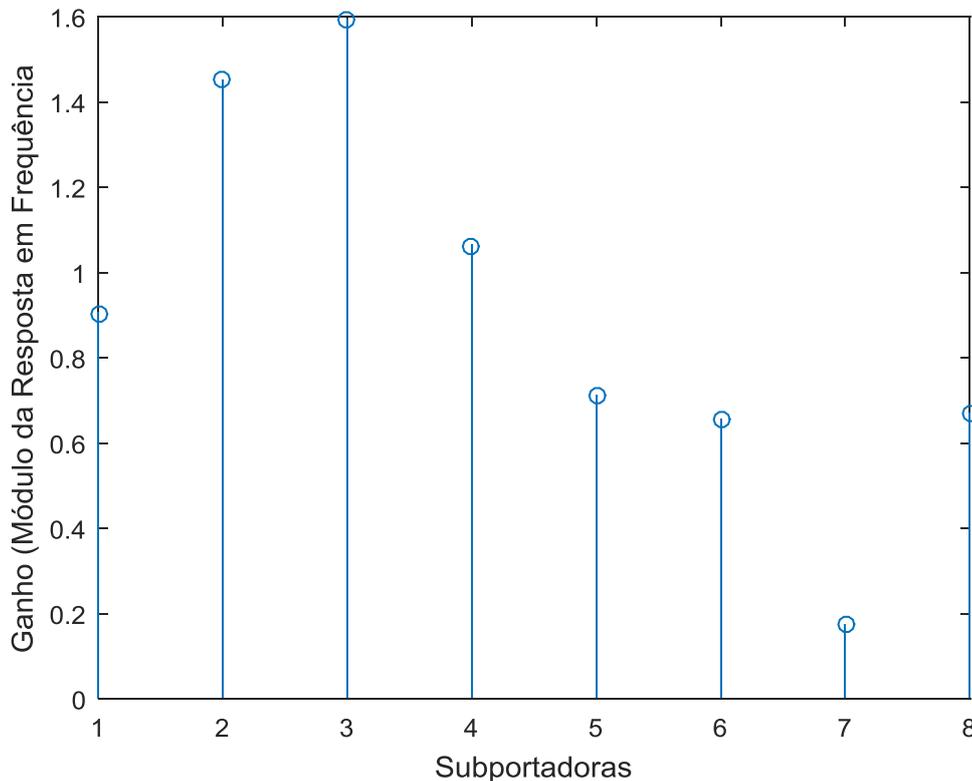


Figura 3.19: Módulo da resposta em frequência do canal simulado, para 8 subportadoras.

Conforme citado na Seção 2.4.2, pode-se utilizar na simulação a potência recebida, ao invés da potência no transmissor. Ao final, basta escalonar o resultado pela perda de percurso. Neste exemplo, optou-se por considerar um modelo log-distância, onde a perda de percurso é dada por (RAPPAPORT, 2008):

$$L_{Loss}(dB) = 20 \log_{10}(f_{MHz}) + 10\eta \log_{10}(d_{km}) - 27,558 + 10\eta \log_{10}(1000) \quad (128)$$

Onde $f_{MHz} = 2400$ (2,4GHz), $d_{km} = 0,1$ (100m) e $\eta = 3,5$ (escolhido com base na Tabela 2.1). Sendo assim:

$$\begin{aligned} L_{Loss}(dB) &= 20 \log_{10}(2400) + 35 \log_{10}(0,1) - 27,558 + 35 \log_{10}(1000) \\ &\cong 110dB \end{aligned} \quad (129)$$

Considerando uma potência total de transmissão de 100mW e ganhos unitários para as antenas transmissora e receptora, tem-se que, no receptor:

$$P_r = \frac{P_t}{L_{Loss}} = \frac{0,1}{10^{\frac{110}{10}}} = 1,0 \times 10^{-12}W \quad (130)$$

Além disto, no receptor, a densidade espectral de ruído é, com $T = 300K$:

$$N_0 = k_B T \cong 1,38 \times 10^{-23} \frac{W}{K.Hz} (273 + 27)K \cong 4,14 \times 10^{-21} \frac{W}{Hz} \quad (131)$$

De tal maneira que a potência de ruído é dada por:

$$P_{ruído} = N_0 W_{sis} = 4,14 \times 10^{-21} \frac{W}{Hz} 20 \times 10^6 Hz = 8,28 \times 10^{-14}W \quad (132)$$

Onde W_{sis} representa a banda do sistema, assumindo valor de 20MHz.

Portanto, a SNR média do sistema, sem considerar as componentes de multipercurso que influenciam cada uma das subportadoras, é dada por:

$$SNR = \frac{P_r}{P_{ruído}} = \frac{1,0 \times 10^{-12}W}{8,28 \times 10^{-14}W} = 12,1 \text{ ou } 10,8 \text{ dB} \quad (133)$$

Sendo assim, de acordo com a Seção 2.4.1, a SNR em cada subcanal é dada por (64):

$$SNR_n = \frac{P_{\text{signal},n}}{P_{\text{ruído},n}} = \frac{P_n}{\sigma^2} |H_n|^2 = \frac{P_n |H_n|^2}{(N_0 W/N)} \quad (134)$$

Onde $P_{\text{signal},n}$ é a potência do subcanal considerando o efeito de multipercurso, ou seja:

$$P_{\text{signal},n} = P_n |H_n|^2 \quad (135)$$

E P_n representa a potência média do subcanal em questão, que é ponderada pelo ganho da componente de multipercurso equivalente. Além disto, $P_{\text{ruído},n}$ é a potência de ruído em cada subportadora, calculada por:

$$P_{\text{ruído},n} = \sigma^2 = N_0 W/N = 8,28 \times 10^{-14} W/8 = 1,035 \times 10^{-14} W \quad (136)$$

Com os valores do módulo da resposta em frequência de cada subcanal e da potência de ruído por subcanal, pode-se calcular os respectivos valores de g_n , onde:

$$\begin{aligned} g_1 &= \frac{|H_1|^2}{\sigma^2} = \frac{(0,9039)^2}{1,035 \times 10^{-14}} \cong 7,9 \times 10^{13} W^{-1} \\ g_2 &= \frac{|H_2|^2}{\sigma^2} = \frac{(1,4509)^2}{1,035 \times 10^{-14}} \cong 2,0 \times 10^{14} W^{-1} \\ g_3 &= \frac{|H_3|^2}{\sigma^2} = \frac{(1,5937)^2}{1,035 \times 10^{-14}} \cong 2,5 \times 10^{14} W^{-1} \\ g_4 &= \frac{|H_4|^2}{\sigma^2} = \frac{(1,0630)^2}{1,035 \times 10^{-14}} \cong 1,1 \times 10^{14} W^{-1} \\ g_5 &= \frac{|H_5|^2}{\sigma^2} = \frac{(0,7103)^2}{1,035 \times 10^{-14}} \cong 4,9 \times 10^{13} W^{-1} \\ g_6 &= \frac{|H_6|^2}{\sigma^2} = \frac{(0,6536)^2}{1,035 \times 10^{-14}} \cong 4,1 \times 10^{13} W^{-1} \\ g_7 &= \frac{|H_7|^2}{\sigma^2} = \frac{(0,1739)^2}{1,035 \times 10^{-14}} \cong 3,0 \times 10^{12} W^{-1} \\ g_8 &= \frac{|H_8|^2}{\sigma^2} = \frac{(0,6681)^2}{1,035 \times 10^{-14}} \cong 4,3 \times 10^{13} W^{-1} \end{aligned}$$

Ao calculá-los, nota-se que o sétimo subcanal possui um valor que é duas ordens de grandeza menor do que o terceiro subcanal. Este comportamento reforça o fato de que subportadoras ruins podem prejudicar o sistema, dependendo da ordem da modulação utilizada e da potência fornecida, por exemplo. Por fim, o valor de *gap* de SNR utilizado nos exemplos será de 8,8dB, considerando uma BER de 10^{-6} para o sistema.

3.2.1 Soluções RA

Nesta seção, uma análise acerca das soluções RA será realizada, tendo em vista os algoritmos de *water-filling*, Chow e Levin-Campello e os parâmetros propostos anteriormente.

Iniciando pelo *water-filling*, que representa a alocação de recursos ótima, considerando granularidade infinita, tem-se $N^* = N = 8$, primeiramente. Sendo assim, as potências inicialmente alocadas para cada subcanal são calculadas, com base na constante K e g_n de cada um destes:

$$P_n = K - \frac{\Gamma}{g_n}$$

Com:

$$K = \frac{1}{N^*} \left(P_r + \Gamma \sum_{i=1}^{N^*} \frac{1}{g_n} \right) \cong 5,4 \times 10^{-13}$$

Logo, os valores iniciais de potência alocados são:

$$\begin{aligned} P_1 &\cong 4,5 \times 10^{-13} W \\ P_2 &\cong 5,1 \times 10^{-13} W \\ P_3 &\cong 5,2 \times 10^{-13} W \\ P_4 &\cong 4,8 \times 10^{-13} W \\ P_5 &\cong 3,9 \times 10^{-13} W \\ P_6 &\cong 3,6 \times 10^{-13} W \\ P_7 &\cong -0,21 \times 10^{-12} W \end{aligned}$$

$$P_8 \cong 3,7 \times 10^{-13}W$$

Observando a alocação inicial, nota-se que, para o valor de K definido, a potência do sétimo canal se torna negativa, fazendo com que a parte iterativa do algoritmo o elimine. Com esta eliminação, $N^* = N-1 = 7$, $P_7 = 0$ e o cálculo da constante K é atualizado:

$$K = \frac{1}{N^*} \left(P_r + \Gamma \sum_{i=1}^{N^*} \frac{1}{g_n} \right) \cong 2,5 \times 10^{-13}$$

Com isto, os valores de potência dos subcanais utilizados são calculados novamente:

$$P_1 \cong 1,5 \times 10^{-13}W$$

$$P_2 \cong 2,1 \times 10^{-13}W$$

$$P_3 \cong 2,2 \times 10^{-13}W$$

$$P_4 \cong 1,8 \times 10^{-13}W$$

$$P_5 \cong 9,4 \times 10^{-14}W$$

$$P_6 \cong 6,6 \times 10^{-14}W$$

$$P_7 \cong 0W$$

$$P_8 \cong 7,4 \times 10^{-14}W$$

Como não existe nenhum valor negativo de potência, a solução para o *water-filling* foi encontrada nestas condições de simulação. Com estes valores de potência, é possível calcular a taxa normalizada alocada em cada subcanal:

$$b_1 = \log_2 \left(1 + \frac{P_1 g_1}{\Gamma} \right) \cong 1,38 \frac{\text{bits}}{\text{s.Hz}}$$

$$b_2 = \log_2 \left(1 + \frac{P_2 g_2}{\Gamma} \right) \cong 2,75 \frac{\text{bits}}{\text{s.Hz}}$$

$$b_3 = \log_2 \left(1 + \frac{P_3 g_3}{\Gamma} \right) \cong 3,01 \frac{\text{bits}}{\text{s.Hz}}$$

$$b_4 = \log_2 \left(1 + \frac{P_4 g_4}{\Gamma} \right) \cong 1,85 \frac{\text{bits}}{\text{s.Hz}}$$

$$b_5 = \log_2 \left(1 + \frac{P_5 g_5}{\Gamma} \right) \cong 0,68 \frac{\text{bits}}{\text{s.Hz}}$$

$$b_6 = \log_2 \left(1 + \frac{P_6 g_6}{\Gamma} \right) \cong 0,44 \frac{\text{bits}}{\text{s.Hz}}$$

$$b_7 = \log_2 \left(1 + \frac{P_7 g_7}{\Gamma} \right) = 0$$

$$b_8 = \log_2 \left(1 + \frac{P_8 g_8}{\Gamma} \right) \cong 0,51 \frac{\text{bits}}{\text{s.Hz}}$$

A Figura 3.20 mostra o resultado encontrado em formato de gráfico de barras. Esta representação é amplamente utilizada em (CIOFFI, 2008) e (LI e STÜBER, 2006), tendo em vista que é possível observar como a potência é “despejada” em cada subcanal até atingir um nível, determinado pela constante K . Neste caso, observa-se que $K \cong 2,5 \times 10^{-13}$, que é o resultado encontrado no algoritmo exposto anteriormente.

Um ponto importante a ser notado é que o inverso de g_n de cada subcanal influencia na quantidade de potência alocada. Neste âmbito, a sétima subportadora é muito pior do que as demais e por isto é eliminada do sistema (está acima do nível definido). Por outro lado, ao segundo e terceiro subcanais mais potência é alocada (representada pelas barras de cor vermelha).

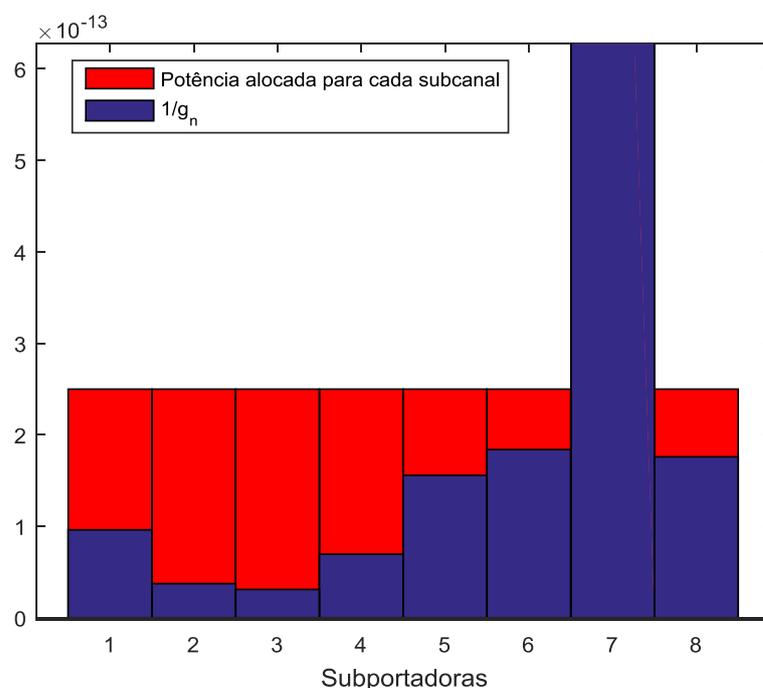


Figura 3.20: Solução *water-filling* RA em gráfico de barras.

Outra análise interessante é exposta na Figura 3.21, a seguir. Nesta figura, o módulo da resposta em frequência do canal é comparado com a solução encontrada.

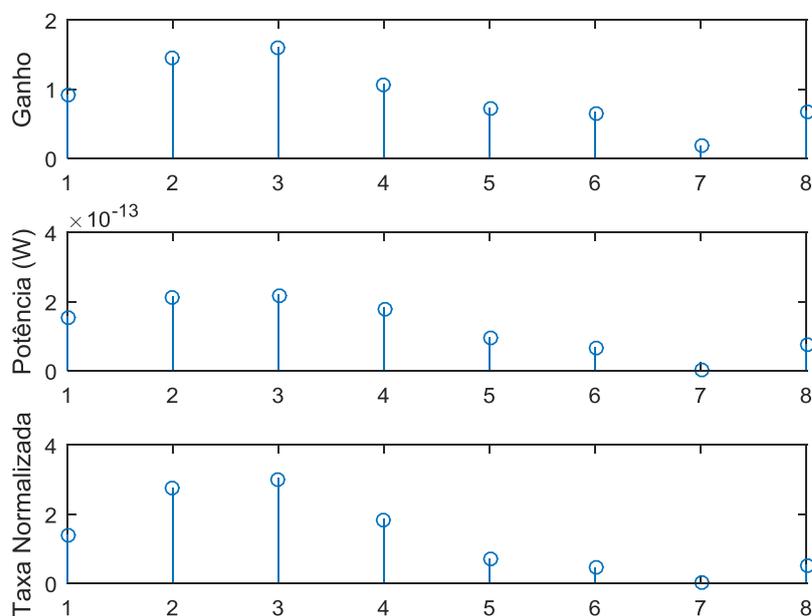


Figura 3.21: Solução *water-filling*, comparação entre a resposta do canal e a alocação de potência e taxa normalizada.

Como esperado, as alocações de taxa e potência seguem o comportamento dos ganhos de cada subcanal, no sentido de que quanto maior o ganho, mais potência é alocada, e vice-versa. Como a potência e a taxa normalizada são relacionadas, a taxa normalizada também segue este comportamento. A Figura 3.22 mostra a resposta no console do MATLAB, onde se observa a sintaxe utilizada para chamar a função adequadamente.

```

[bn Pn] = waterfilling_ra(Hn, W, No, P, N, gap)

bn =

    1.3786    2.7441    3.0149    1.8465    0.6832    0.4431         0    0.5064

Pn =

    1.0e-12 *
    0.1538    0.2126    0.2190    0.1804    0.0943    0.0661         0    0.0740

>> sum(Pn)

ans =

    1.0000e-12

>> sum(bn)

ans =

    10.6169

```

Figura 3.22: Solução *water-filling* RA, saída no console do MATLAB.

Para verificar se a solução encontrada se encaixa no critério RA, é necessário observar se a potência total alocada é igual à potência do sistema OFDM no momento da transmissão. De acordo com a figura anterior, a soma das potências alocadas é igual à considerada inicialmente na entrada do receptor do sistema, o que confere o caráter RA à solução encontrada, quando as perdas de percurso são consideradas. Além disto, observa-se que a soma das taxas normalizadas é igual à aproximadamente 10,62 bits/s/Hz, o que representa a solução ótima para alocação nas condições de simulação.

Vale ressaltar que as potências encontradas estão dimensionadas do ponto de vista do receptor. No transmissor, basta adicionar a perda de percurso para obter:

$$P_1 \cong 15mW$$

$$P_2 \cong 21mW$$

$$P_3 \cong 22mW$$

$$P_4 \cong 18mW$$

$$P_5 \cong 9,4mW$$

$$P_6 \cong 6,6mW$$

$$P_7 \cong 0W$$

$$P_8 \cong 7,4mW$$

O próximo algoritmo analisado é o Chow RA. A primeira etapa consiste no algoritmo inicial exposto na Seção 2.4.4, composto de cinco passos. Sendo assim, $i = N = 8$ e b_{temp} é um vetor com nove posições, com $b_{temp}(9) = 0$. A divisão de potência é realizada de maneira uniforme para todos os subcanais, ou seja:

$$P_n = \frac{P_{sinal}}{8} = \frac{1,0 \times 10^{-12}W}{8} \cong 1,25 \times 10^{-13}W$$

Com $n = 1, 2, \dots, 8$. As SNRs podem ser calculadas, já que os valores de g_n são conhecidos:

$$SNR_1 \cong 9,9dB$$

$$SNR_2 \cong 14,1dB$$

$$SNR_3 \cong 14,9dB$$

$$SNR_4 \cong 11,4dB$$

$$SNR_5 \cong 7,8dB$$

$$SNR_6 \cong 7,1dB$$

$$SNR_7 \cong -4,4dB$$

$$SNR_8 \cong 7,3dB$$

Considerando as SNRs até o momento, a taxa normalizada total do sistema é dada por:

$$b_{temp}(8) = \sum_{n=1}^N \log_2 \left(1 + \frac{P_n g_n}{\Gamma} \right) \cong 9,58 \frac{bits}{s.Hz}$$

Como $b_{temp}(8) > b_{temp}(9) = 0$, o canal com menor g_n (ou consequentemente menor SNR, já que a potência é dividida igualmente) é removido. Neste caso, tem-se a remoção do sétimo subcanal. Com esta remoção, $i = 7$ e:

$$P_n = \frac{P_{sinal}}{7} = \frac{1 \times 10^{-12}W}{7} \cong 1,43 \times 10^{-13}W$$

Com $n = 1, 2, \dots, 8$ e $n \neq 7$. As SNRs são recalculadas, considerando as novas potências:

$$SNR_1 \cong 10,5dB$$

$$SNR_2 \cong 14,6dB$$

$$SNR_3 \cong 15,4dB$$

$$SNR_4 \cong 11,9dB$$

$$SNR_5 \cong 8,4dB$$

$$SNR_6 \cong 7,7dB$$

$$SNR_7 \cong 0dB$$

$$SNR_8 \cong 7,9dB$$

Calculando a taxa normalizada total do sistema ao eliminar o sétimo subcanal, tem-se:

$$b_{temp}(7) = \sum_{\substack{n=1 \\ n \neq 7}}^N \log_2 \left(1 + \frac{P_n g_n}{\Gamma} \right) \cong 10,32 \frac{bits}{s.Hz}$$

Como $b_{temp}(7) > b_{temp}(8)$, novamente o canal com menor g_n é removido. Neste caso, tem-se a remoção do sexto subcanal. Com esta remoção, $i = 6$ e:

$$P_n = \frac{P_{sinal}}{6} = \frac{1 \times 10^{-12}W}{6} \cong 1,7 \times 10^{-13}W$$

Com $n = 1, 2, \dots, 8$ e $n \neq 6, 7$. Recalculando as SNRs:

$$SNR_1 \cong 11,2dB$$

$$SNR_2 \cong 15,3dB$$

$$SNR_3 \cong 16,1dB$$

$$SNR_4 \cong 12,6dB$$

$$SNR_5 \cong 9,1dB$$

$$SNR_6 \cong 0dB$$

$$SNR_7 \cong 0dB$$

$$SNR_8 \cong 8,6dB$$

Nesta situação, a taxa normalizada do sistema é:

$$b_{temp}(6) = \sum_{n=1}^6 \log_2 \left(1 + \frac{P_n g_n}{\Gamma} \right) \cong 10,36 \frac{bits}{s.Hz}$$

Ou seja, $b_{temp}(6) > b_{temp}(7)$, e o algoritmo prossegue removendo o oitavo subcanal, de tal forma que a potência nos restantes é dada por:

$$P_n = \frac{P_{sinal}}{5} = \frac{1 \times 10^{-12}W}{5} \cong 2,0 \times 10^{-13}W$$

Com $n = 1, 2, \dots, 5$. As SNRs neste caso são dadas por:

$$SNR_1 \cong 12,0dB$$

$$SNR_2 \cong 16,1dB$$

$$SNR_3 \cong 16,9dB$$

$$SNR_4 \cong 13,4dB$$

$$SNR_5 \cong 9,9dB$$

$$SNR_6 \cong 0dB$$

$$SNR_7 \cong 0dB$$

$$SNR_8 \cong 0dB$$

Por fim, a taxa normalizada total do sistema com cinco subcanais pode ser calculada:

$$b_{temp}(5) = \sum_{n=1}^5 \log_2 \left(1 + \frac{P_n g_n}{\Gamma} \right) \cong 10,34 \frac{bits}{s.Hz}$$

O cálculo anterior mostra que $b_{temp}(6) > b_{temp}(5)$, que é a condição de parada para a primeira etapa do algoritmo. Isto significa que a remoção do último

subcanal acarretou em diminuição da taxa normalizada total do sistema, o que não é desejado. Desta maneira, a solução consiste da taxa total armazenada em b_{temp} (6), que não utiliza o sexto e o sétimo subcanais.

Tendo encontrada a distribuição inicial de taxas (através do cálculo da taxa normalizada total), três passos adicionais buscam a solução RA. Calculando as taxas individuais de cada subcanal restante e assumindo granularidade unitária ($\beta = 1$) do sistema OFDM, o arredondamento é realizado.

$$b_1 = \log_2 \left(1 + \frac{P_1 g_1}{\Gamma} \right) \cong 1,45 \rightarrow 1 \frac{bit}{s.Hz}$$

$$b_2 = \log_2 \left(1 + \frac{P_2 g_2}{\Gamma} \right) \cong 2,45 \rightarrow 2 \frac{bits}{s.Hz}$$

$$b_3 = \log_2 \left(1 + \frac{P_3 g_3}{\Gamma} \right) \cong 2,67 \rightarrow 3 \frac{bits}{s.Hz}$$

$$b_4 = \log_2 \left(1 + \frac{P_4 g_4}{\Gamma} \right) \cong 1,77 \rightarrow 2 \frac{bits}{s.Hz}$$

$$b_5 = \log_2 \left(1 + \frac{P_5 g_5}{\Gamma} \right) \cong 1,05 \rightarrow 1 \frac{bit}{s.Hz}$$

$$b_6 = \log_2 \left(1 + \frac{P_6 g_6}{\Gamma} \right) = 0$$

$$b_7 = \log_2 \left(1 + \frac{P_7 g_7}{\Gamma} \right) = 0$$

$$b_8 = \log_2 \left(1 + \frac{P_8 g_8}{\Gamma} \right) \cong 0,96 \rightarrow 1 \frac{bit}{s.Hz}$$

Além de realizar o arredondamento, as potências de cada subcanal devem ser ajustadas pelo fator $2^{b_n} - 1 / 2^{b_{n,antiga}} - 1$. Isto faz com que a potência seja aumentada nos subcanais em que há aumento de taxa no arredondamento, enquanto a potência é diminuída nos subcanais em que há diminuição da taxa. Após este ajuste, obtém-se a seguinte distribuição de potências:

$$P_1 \cong 9,6 \times 10^{-14} W$$

$$P_2 \cong 1,1 \times 10^{-13} W$$

$$P_3 \cong 2,2 \times 10^{-13} W$$

$$P_4 \cong 2,1 \times 10^{-13} W$$

$$P_5 \cong 1,6 \times 10^{-13} W$$

$$P_6 \cong 0W$$

$$P_7 \cong 0W$$

$$P_8 \cong 1,8 \times 10^{-13}W$$

A soma das potências encontrada ainda não atinge a potência total. Sendo assim, o fator de escala $P_r/P_{utilizada} \cong 1,04$ é aplicado a cada subcanal, para alcançar a solução RA. Por fim, as potências totais encontradas são:

$$P_1 \cong 1,0 \times 10^{-13}W$$

$$P_2 \cong 1,2 \times 10^{-13}W$$

$$P_3 \cong 2,2 \times 10^{-13}W$$

$$P_4 \cong 2,2 \times 10^{-13}W$$

$$P_5 \cong 1,6 \times 10^{-13}W$$

$$P_6 \cong 0W$$

$$P_7 \cong 0W$$

$$P_8 \cong 1,8 \times 10^{-13}W$$

Ao calcular as taxas encontradas com cada uma destas potências, tem-se:

$$b_1 = \log_2 \left(1 + \frac{P_1 g_1}{\Gamma} \right) \cong 1,03 \frac{\text{bit}}{\text{s.Hz}}$$

$$b_2 = \log_2 \left(1 + \frac{P_2 g_2}{\Gamma} \right) \cong 2,04 \frac{\text{bits}}{\text{s.Hz}}$$

$$b_3 = \log_2 \left(1 + \frac{P_3 g_3}{\Gamma} \right) \cong 3,05 \frac{\text{bits}}{\text{s.Hz}}$$

$$b_4 = \log_2 \left(1 + \frac{P_4 g_4}{\Gamma} \right) \cong 2,04 \frac{\text{bits}}{\text{s.Hz}}$$

$$b_5 = \log_2 \left(1 + \frac{P_5 g_5}{\Gamma} \right) \cong 1,03 \frac{\text{bit}}{\text{s.Hz}}$$

$$b_6 = \log_2 \left(1 + \frac{P_6 g_6}{\Gamma} \right) = 0$$

$$b_7 = \log_2 \left(1 + \frac{P_7 g_7}{\Gamma} \right) = 0$$

$$b_8 = \log_2 \left(1 + \frac{P_8 g_8}{\Gamma} \right) \cong 1,03 \frac{\text{bit}}{\text{s.Hz}}$$

A Figura 3.23 mostra a função para o algoritmo de Chow no MATLAB, bem como o resultado obtido nesta seção. Comparando-se a resposta obtida com a solução *water-filling*, nota-se que um subcanal adicional é removido do sistema, justamente o que possuía o segundo pior desempenho no caso do *water-filling*. Com a remoção deste subcanal, a taxa total do sistema diminui, entretanto, a solução RA de Chow também alcança o limite de potência de transmissão imposto inicialmente de 100mW no transmissor (adicionando-se a perda de percurso).

```
[bn Pn] = chows_ra_cioffi(N, Hn, gap, W, No, P)

bn =

     1     2     3     2     1     0     0     1

Pn =

 1.0e-12 *
 0.0996  0.1160  0.2244  0.2162  0.1614      0      0  0.1824

>> sum(Pn)

ans =

 1.0000e-12

>> sum(bn)

ans =

 10
```

Figura 3.23: Solução Chow RA, saída no console do MATLAB.

Nota-se que este algoritmo se torna sub-ótimo por considerar taxas com granularidade finita do sistema, visto que a taxa normalizada total obtida é inferior à do caso ótimo. Além disto, após aplicação final do fator de escala, as taxas obtidas não são exatamente inteiras, mas bem próximas. Isto indica que o algoritmo de Chow também é sub-ótimo neste aspecto. Entretanto, mesmo com a soma destas taxas, tem-se uma solução pior do que o *water-filling*, conforme ilustra a Figura 3.24.

```

>> b_n_chow

b_n_chow =

    1.0264    2.0395    3.0460    2.0395    1.0264         0         0    1.0264

>> sum(b_n_chow)

ans =

    10.2042

```

Figura 3.24: Solução Chow RA, comparação da taxa com o caso ótimo, antes do arredondamento.

Outro ponto importante deste algoritmo reside na distribuição de potências obtida. Nota-se que no início a potência é dividida igualmente para todos os subcanais, diferente do caso ótimo. Entretanto, nos três passos finais para encontrar a solução RA, o fator de escala considera o arredondamento das taxas normalizadas como base para ajuste de potência.

A última solução RA apresentada é baseada no algoritmo de Levin-Campello. Considerando $\beta = 1$, a primeira etapa consiste em construir a tabela de potências incrementais, conforme ilustra a Figura 3.25 para este exemplo.

1	2	3	4	5	6	7	8
9.6095e-14	3.7296e-14	3.0912e-14	6.9482e-14	1.5562e-13	1.8379e-13	2.5962e-12	1.7590e-13
1.9219e-13	7.4593e-14	6.1824e-14	1.3896e-13	3.1123e-13	3.6758e-13	5.1924e-12	3.5179e-13
3.8438e-13	1.4919e-13	1.2365e-13	2.7793e-13	6.2247e-13	7.3515e-13	1.0385e-11	7.0359e-13
7.6876e-13	2.9837e-13	2.4730e-13	5.5586e-13	1.2449e-12	1.4703e-12	2.0770e-11	1.4072e-12
1.5375e-12	5.9674e-13	4.9459e-13	1.1117e-12	2.4899e-12	2.9406e-12	4.1539e-11	2.8143e-12
3.0750e-12	1.1935e-12	9.8918e-13	2.2234e-12	4.9797e-12	5.8812e-12	8.3079e-11	5.6287e-12
6.1501e-12	2.3870e-12	1.9784e-12	4.4469e-12	9.9595e-12	1.1762e-11	1.6616e-10	1.1257e-11
1.2300e-11	4.7739e-12	3.9567e-12	8.8937e-12	1.9919e-11	2.3525e-11	3.3232e-10	2.2515e-11
2.4600e-11	9.5478e-12	7.9135e-12	1.7787e-11	3.9838e-11	4.7050e-11	6.6463e-10	4.5030e-11
4.9201e-11	1.9096e-11	1.5827e-11	3.5575e-11	7.9676e-11	9.4099e-11	1.3293e-09	9.0059e-11

Figura 3.25: Tabela de potências incrementais em função da taxa normalizada para os oito subcanais simulados neste exemplo.

Cada coluna da tabela exposta na figura anterior representa um subcanal. Por outro lado, cada linha representa a potência incremental necessária para aumentar a taxa normalizada em uma unidade. Exemplificando, a primeira linha representa quantos Watts (no receptor) são necessários para aumentar a taxa normalizada de 0

para 1 bit/s/Hz, dependendo do subcanal. Então, no primeiro subcanal, são necessários $9,6 \times 10^{-14}$ W, enquanto que no sétimo subcanal (com a pior SNR), são necessários $2,5 \times 10^{-12}$ W, que é um valor muito maior (ou seja, é mais custoso aumentar a taxa em um subcanal pior).

Com a tabela de potências incrementais construída, a primeira etapa consiste em assumir uma distribuição de bits e utilizar o algoritmo que torna esta distribuição eficiente.

A distribuição pode ser gerada aleatoriamente, por exemplo. Considerando a distribuição inicial:

$$\mathbf{b} = [2, 4, 5, 3, 2, 1, 3, 2]$$

Os passos do algoritmo seriam:

- Passo 1: $m = 6$, $n = 7$. Logo a distribuição é ajustada para $\mathbf{b} = [2, 4, 5, 3, 2, 2, 2, 2]$.
- Passo 2: $m = 1$, $n = 7$. Sendo assim, $\mathbf{b} = [3, 4, 5, 3, 2, 2, 1, 2]$.
- Passo 3: $m = 4$, $n = 7$. Por fim, $\mathbf{b} = [3, 4, 5, 4, 2, 2, 0, 2]$.

A distribuição obtida é eficiente do ponto de vista de potência incremental. Tendo uma distribuição eficiente, é necessário aplicar o critério RA para obter uma solução que respeite a potência total disponível no transmissor. Alguns passos do algoritmo de *P-Tightness* para esta distribuição são expostos a seguir. A potência de cada subcanal obtida com a distribuição eficiente é dada por:

$$P_1 \cong 6,7 \times 10^{-13} W$$

$$P_2 \cong 5,6 \times 10^{-13} W$$

$$P_3 \cong 9,6 \times 10^{-13} W$$

$$P_4 \cong 1,0 \times 10^{-12} W$$

$$P_5 \cong 4,7 \times 10^{-13} W$$

$$P_6 \cong 5,5 \times 10^{-13} W$$

$$P_7 \cong 0 W$$

$$P_8 \cong 5,3 \times 10^{-13} W$$

De tal forma que:

$$S = \sum_{i=1}^N P_n(b_n) \cong 4,8 \times 10^{-12} W$$

Cujo valor é mais do que quatro vezes a potência recebida assumida inicialmente. Sendo assim, a próxima etapa consiste em ajustar a potência total do sistema (e conseqüentemente a taxa normalizada de cada subcanal) de maneira a alcançar uma solução RA eficiente.

Como a distribuição de bits está muito longe da resposta esperada (próxima da solução do *water-filling* e do Chow), muitos passos são realizados até que a solução final seja obtida. Apenas alguns passos são expostos para este exemplo:

$$\begin{aligned} \mathbf{b} &= [3, 4, 5, 4, 2, 2, 0, 2] \rightarrow [3, 4, 5, 3, 2, 2, 0, 2] \rightarrow [3, 4, 4, 3, 2, 2, 0, 2] \rightarrow \dots \\ &\rightarrow [1, 3, 3, 2, 1, 0, 0, 1] \rightarrow [1, 3, 3, 2, 1, 0, 0, 0] \end{aligned}$$

A solução anterior mostra que a distribuição final de Levin-Campello elimina, neste caso, os três últimos subcanais, alocando potência aos demais da seguinte maneira:

$$P_1 \cong 9,6 \times 10^{-14} W$$

$$P_2 \cong 2,6 \times 10^{-13} W$$

$$P_3 \cong 2,2 \times 10^{-13} W$$

$$P_4 \cong 2,1 \times 10^{-13} W$$

$$P_5 \cong 1,6 \times 10^{-13} W$$

$$P_6 \cong 0 W$$

$$P_7 \cong 0 W$$

$$P_8 \cong 0 W$$

Um ponto interessante e citado na Seção 2.4.5 é que a solução RA não necessariamente atinge a igualdade com relação à potência disponível no transmissor, para o caso de Levin-Campello. Desta maneira, o sistema define uma margem de funcionamento. Neste exemplo:

$$\gamma = \frac{P_r}{\sum_{i=1}^5 P_n} = \frac{1 \times 10^{-12} W}{9,4 \times 10^{-13} W} \cong 1,06 \text{ ou } 0,27 \text{ dB}$$

Ou seja, existe uma folga de potência, já que o sistema utiliza menos do que o disponível no transmissor. Isto ocorre pois, se um bit for adicionado de maneira eficiente em algum dos subcanais, o critério RA é violado.

Nota-se que uma escolha aleatória inicial da distribuição de bits antes de torná-la eficiente pode resultar em um número muito maior de passos até alcançar uma distribuição eficiente. É por este motivo que a própria solução do *water-filling* arredondada pode ser utilizada como ponto de partida e pode-se provar que esta é eficiente (LI e STÜBER, 2006).

Utilizando o fato exposto no parágrafo anterior e considerando $\mathbf{b}_{WF} = [1, 3, 3, 2, 1, 0, 0, 1]$ como a distribuição *water-filling* arredondada para este exemplo, realmente é possível constatar que o algoritmo que torna a distribuição eficiente não a modifica. Por fim, em apenas um passo, a mesma resposta é obtida, com mesma margem do sistema. A Figura 3.26 mostra o resultado obtido no console do MATLAB.

```
[bn Pn margin_dB] = LC_v4_RA(N, Hn, gap, W, No, P)

P_n_sub =

    1.0e-12 *

    0.0961    0.2611    0.2164    0.2084    0.1556         0         0         0

Elapsed time is 0.004190 seconds.

bn =

     1     3     3     2     1     0     0     0

Pn =

    9.3762e-13

margin_dB =

    0.2797

>> sum(bn)

ans =

    10
```

Figura 3.26: Solução Levin-Campello RA, saída no console do MATLAB.

Por fim, nota-se que esta solução é semelhante à obtida no *water-filling*, porém é novamente sub-ótima do ponto de maximização da taxa. Neste caso, a análise considera a granularidade finita em todo o algoritmo, de tal forma que a soma das taxas normalizadas é realizada com os valores inteiros. Além disto, o oitavo subcanal também é eliminado, o que não ocorreu nas soluções de Chow e *water-filling*. No entanto, o bit de informação perdido no último subcanal da solução de Chow é alocado para o segundo, que possui uma SNR maior.

Para finalizar esta subseção, outro exemplo é exposto a seguir, apenas para observação do *water-filling* com um maior número de subportadoras. A Figura 3.27 mostra a resposta obtida considerando-se os ganhos da Figura 3.2, SNR média de 20dB, 64 subportadoras e 800mW de potência no transmissor. Nota-se que a alocação de potência e taxa normalizada segue o padrão dos ganhos impostos pelo canal de multipercuro.

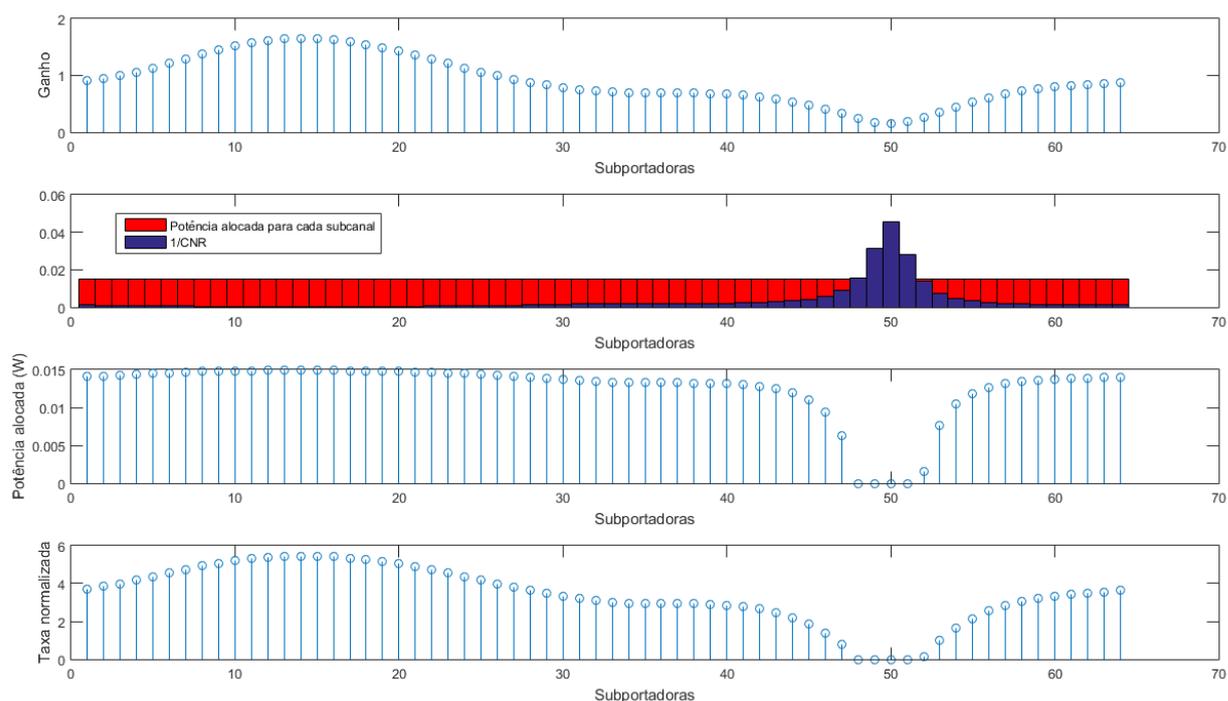


Figura 3.27: Solução *water-filling* RA para 64 subportadoras.

Um fato importante e que não foi citado até o momento é que as taxas normalizadas finais podem representar valores que não são disponíveis pelo sistema. Por exemplo, se um sistema possui apenas BPSK, QPSK, 16-QAM e 64-QAM como tipos possíveis de modulação, então os valores possíveis de taxa

normalizada (ou valores de taxas normalizadas reais) são 1 bit/s/Hz, 2 bits/s/Hz, 4 bits/s/Hz e 6 bits/s/Hz, respectivamente. Desta maneira as taxas obtidas em qualquer uma das soluções devem ser modificadas de tal forma que estas assumam valores possíveis. Sendo assim, o desempenho considerando taxas normalizadas reais pode ser inferior ao obtido inicialmente dos algoritmos de alocação de recursos.

Para finalizar esta subseção, um resumo das alocações de potência fornecidas pelas soluções RA nos exemplos citados é exposto na Tabela 3.3. Nota-se que a solução *water-filling* aloca mais potência para as subportadoras com maior SNR (vide Figura 3.19) e menos potência para as piores subportadoras do sistema. Um comportamento semelhante é encontrado para a solução de Levin-Campello.

Entretanto, para o caso do algoritmo de Chow, inicialmente a potência é igual para todas as subportadoras. Posteriormente, a alocação de potência faz com que estas dependam do arredondamento das taxas normalizadas para cada subportadora. Neste exemplo, a oitava subportadora possui maior potência do que a primeira, porém sua SNR é menor. Isto reforça o caráter sub-ótimo desta solução.

Outro ponto citado nesta subseção e importante é que a solução de Levin-Campello não utiliza toda a potência disponível, fornecendo uma margem de funcionamento para o sistema. Isto ocorre pois, se um bit de granularidade fosse inserido em alguma das subportadoras, a potência incremental necessária violaria a condição RA.

Tabela 3.3: Alocação de potência para as soluções RA.

Subportadora	Potência – <i>Water-filling</i> (W)	Potência – Chow (W)	Potência – Levin- Campello (W)
1	$1,5 \times 10^{-13}$	$1,0 \times 10^{-13}$	$9,6 \times 10^{-14}$
2	$2,1 \times 10^{-13}$	$1,2 \times 10^{-13}$	$2,6 \times 10^{-13}$
3	$2,2 \times 10^{-13}$	$2,2 \times 10^{-13}$	$2,2 \times 10^{-13}$
4	$1,8 \times 10^{-13}$	$2,2 \times 10^{-13}$	$2,1 \times 10^{-13}$
5	$9,4 \times 10^{-14}$	$1,6 \times 10^{-13}$	$1,6 \times 10^{-13}$
6	$6,6 \times 10^{-14}$	0	0
7	0	0	0
8	$7,4 \times 10^{-14}$	$1,8 \times 10^{-13}$	0
Total:	$1,0 \times 10^{-12}$	$1,0 \times 10^{-12}$	$9,4 \times 10^{-13}$

A Tabela 3.4 mostra um resumo para a alocação de taxas normalizadas em cada subportadora. Inicialmente, nota-se que a solução *water-filling* produz a maior taxa normalizada total para o sistema, em contraste com as soluções de Chow e Levin-Campello.

Com relação às taxas normalizadas para a solução de *water-filling*, nota-se que esta solução adota granularidade infinita do sistema, já que as taxas não são discretas. No algoritmo de Chow, as taxas encontradas são quase discretas, com um pequeno erro de aproximação. Mesmo assim, após arredondamento, a taxa normalizada total não ultrapassa a solução ótima. O algoritmo de Levin-Campello considera granularidade finita em toda sua concepção, mas também é sub-ótimo. Como na prática a granularidade é finita, tem-se que as soluções que a consideram são sempre inferiores ao *water-filling* (o que será discutido com maiores detalhes na seção 3.2.3).

Tabela 3.4: Alocação de taxa normalizada para as soluções RA.

Subportadora	Taxa Normalizada – <i>Water-filling</i> (bits/s/Hz)	Taxa Normalizada – Chow (bits/s/Hz)	Taxa Normalizada – Levin-Campello (bits/s/Hz)
1	1,38	1,03 → 1	1
2	2,75	2,04 → 2	3
3	3,01	3,05 → 3	3
4	1,85	2,04 → 2	2
5	0,68	1,03 → 1	1
6	0,44	0	0
7	0	0	0
8	0,51	1,03 → 1	0
Total:	10,62	10,20 → 10,00	10,00

3.2.2 Soluções MA

Assim como na seção anterior, as soluções MA *water-filling*, Chow e Levin-Campello serão analisadas. Como a solução MA necessita de uma taxa normalizada fixa e desejada, optou-se por escolher o valor 16 bits/s/Hz, de tal forma a representar uma taxa normalizada média de 2 bits/s/Hz por subportadora. As demais condições de simulação são mantidas para o caso de oito subcanais.

Iniciando pela solução ótima de *water-filling*, $N^* = N = 8$ e as potências iniciais são calculadas com base em:

$$P_n = K_{MA} - \frac{\Gamma}{g_n}$$

Os valores de g_n são iguais ao caso do *water-filling* RA, entretanto, a constante K_{MA} é dada por:

$$K_{MA} = \Gamma \left(\frac{2^b}{\prod_{n=1}^N g_n} \right)^{1/N} = 10^{\left(\frac{8,8}{10}\right)} \left(\frac{2^{16}}{\prod_{n=1}^N g_n} \right)^{1/8} \cong 5,3 \times 10^{-13}$$

De tal forma que:

$$P_1 \cong 4,4 \times 10^{-13} W$$

$$P_2 \cong 5,0 \times 10^{-13} W$$

$$P_3 \cong 5,0 \times 10^{-13} W$$

$$P_4 \cong 4,6 \times 10^{-13} W$$

$$P_5 \cong 3,8 \times 10^{-13} W$$

$$P_6 \cong 3,5 \times 10^{-13} W$$

$$P_7 \cong -0,21 \times 10^{-12} W$$

$$P_8 \cong 3,6 \times 10^{-13} W$$

Existe uma potência negativa para o sétimo canal, logo este é eliminado da análise ($N^* = 7$) e a constante K_{MA} é recalculada:

$$K_{MA} = \Gamma \left(\frac{2^b}{\prod_{\substack{n=1 \\ n \neq 7}}^N g_n} \right)^{\frac{1}{N^*}} = 10^{\left(\frac{8,8}{10}\right)} \left(\frac{2^{16}}{\prod_{\substack{n=1 \\ n \neq 7}}^N g_n} \right)^{1/7} \cong 4,3 \times 10^{-13}$$

Com este novo valor para a constante de *water-filling*, as potências são recalculadas:

$$P_1 \cong 3,3 \times 10^{-13} W$$

$$P_2 \cong 3,9 \times 10^{-13} W$$

$$P_3 \cong 4,0 \times 10^{-13} W$$

$$P_4 \cong 3,6 \times 10^{-13} W$$

$$P_5 \cong 2,7 \times 10^{-13} W$$

$$P_6 \cong 2,4 \times 10^{-13} W$$

$$P_7 \cong 0 W$$

$$P_8 \cong 2,5 \times 10^{-13} W$$

Como não existem potências negativas, as taxas podem ser calculadas para cada subcanal:

$$b_1 = \log_2 \left(1 + \frac{P_1 g_1}{\Gamma} \right) \cong 2,14 \frac{\text{bits}}{\text{s.Hz}}$$

$$b_2 = \log_2 \left(1 + \frac{P_2 g_2}{\Gamma} \right) \cong 3,51 \frac{\text{bits}}{\text{s.Hz}}$$

$$b_3 = \log_2 \left(1 + \frac{P_3 g_3}{\Gamma} \right) \cong 3,78 \frac{\text{bits}}{\text{s.Hz}}$$

$$b_4 = \log_2 \left(1 + \frac{P_4 g_4}{\Gamma} \right) \cong 2,62 \frac{\text{bits}}{\text{s.Hz}}$$

$$b_5 = \log_2 \left(1 + \frac{P_5 g_5}{\Gamma} \right) \cong 1,45 \frac{\text{bits}}{\text{s.Hz}}$$

$$b_6 = \log_2 \left(1 + \frac{P_6 g_6}{\Gamma} \right) \cong 1,21 \frac{\text{bits}}{\text{s.Hz}}$$

$$b_7 = \log_2 \left(1 + \frac{P_7 g_7}{\Gamma} \right) = 0$$

$$b_8 = \log_2 \left(1 + \frac{P_8 g_8}{\Gamma} \right) \cong 1,28 \frac{\text{bits}}{\text{s.Hz}}$$

Assim como no caso RA, a Figura 3.28 mostra o resultado em formato de gráfico de barras. Novamente, observa-se o desempenho ruim do sétimo subcanal, que é eliminado da alocação final. Além disto, as potências não ultrapassam o nível definido por K_{MA} .

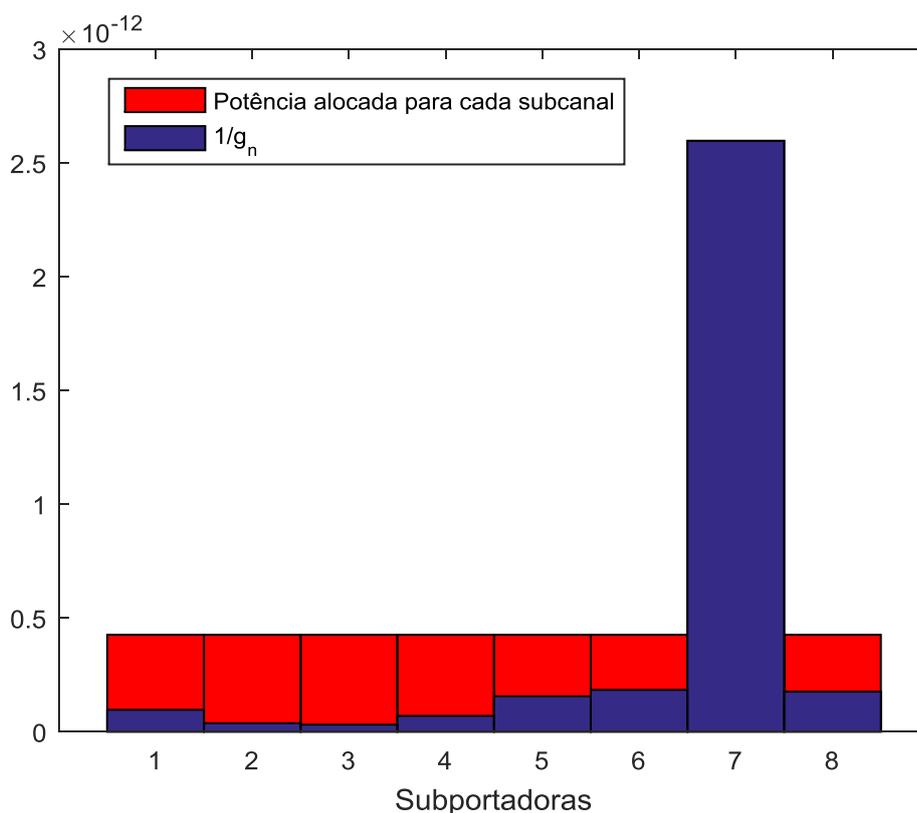


Figura 3.28: Solução *water-filling* MA em gráfico de barras.

Para observar se a solução se encontra em concordância com o critério MA, as taxas obtidas são somadas. A Figura 3.29 mostra os resultados no console do MATLAB. A soma das taxas resulta em 16 bits/s/Hz, indicando o correto funcionamento do algoritmo.

Com relação à potência utilizada pelas subportadoras para alcançar este valor fixo de taxa normalizada, nota-se que esta é maior do que a disponível no transmissor. Desta forma, a margem do sistema é dada por:

$$\gamma = \frac{P_r}{P_{utilizada}} = \frac{1,0 \times 10^{-12}}{2,232 \times 10^{-12}} \cong 0,45 \text{ ou } -3,4\text{dB}$$

Cujo resultado também é retornado da função do MATLAB. Isto significa que a potência disponível no transmissor precisa ser aumentada para que uma taxa total de 16 bits/s/Hz seja obtida do sistema, nas condições deste exemplo. Isto é coerente, já que, retornando ao caso RA, há a utilização de 100mW para taxa normalizada total de 10,62 bits/s/Hz, menor do que os 16 bits/s/Hz impostos.

```

[margin_dB bn Pn] = waterfilling_ma(Hn, W, No, b, P, N, gap)
Elapsed time is 0.176368 seconds.

margin_dB =

    -3.3367

bn =

    2.1477    3.5131    3.7840    2.6155    1.4522    1.2122         0    1.2755

Pn =

    1.0e-12 *
    0.3297    0.3885    0.3949    0.3563    0.2702    0.2420         0    0.2499

>> sum(bn)

ans =

    16.0000

>> sum(Pn)

ans =

    2.2315e-12

```

Figura 3.29: Solução *water-filling* MA, saída no console do MATLAB.

A próxima solução MA analisada vem do algoritmo de Chow. O algoritmo inicial é igual ao da solução RA. Continuando do final desta etapa, ou seja, obtendo a restrição MA, tem-se $i = 6$ e a seguinte distribuição de potências, divididas igualmente para os subcanais utilizados:

$$P_n = \frac{P_{\text{signal}}}{6} = \frac{1,0 \times 10^{-12} W}{6} \cong 1,7 \times 10^{-13} W$$

Com $n = 1, 2, \dots, 8$ e $n \neq 6, 7$. As SNRs são dadas por:

$$SNR_1 \cong 11,2dB$$

$$SNR_2 \cong 15,3dB$$

$$SNR_3 \cong 16,1dB$$

$$SNR_4 \cong 12,6dB$$

$$SNR_5 \cong 9,1dB$$

$$\begin{aligned} SNR_6 &\cong 0dB \\ SNR_7 &\cong 0dB \\ SNR_8 &\cong 8,6dB \end{aligned}$$

Nesta situação, a taxa normalizada do sistema é:

$$b_{temp}(6) = \sum_{\substack{n=1 \\ n \neq 6,7}}^N \log_2 \left(1 + \frac{P_n g_n}{\Gamma} \right) \cong 10,36 \frac{bits}{s.Hz}$$

Nota-se que esta taxa normalizada total é diferente da desejada, logo o algoritmo continua com o cálculo da SNR multicanal:

$$SNR_{mu} = \left[\left(\prod_{\substack{n=1 \\ n \neq 6,7}}^8 \left(1 + \frac{SNR_n}{\Gamma} \right) \right)^{\frac{1}{i}} - 1 \right] \Gamma \cong 11,02 \text{ ou } 10,4dB$$

A margem temporária correspondente à SNR multicanal é calculada a seguir:

$$\gamma_{temp} = \frac{SNR_{mu}}{\Gamma} = \frac{11,02}{2^b - 1} = \frac{10^{\left(\frac{8,8}{10}\right)}}{2^2 - 1} \cong 0,48 \text{ ou } -3,1dB$$

A distribuição de bits temporária é dada por ($\beta = 1$):

$$B_{temp,n} = \frac{1}{\beta} \log_2 \left(1 + \frac{SNR_n}{\Gamma \gamma_{temp}} \right) = \log_2 \left(1 + \frac{SNR_n}{\Gamma \gamma_{temp}} \right)$$

Representando $B_{temp,n}$ vetorialmente, tem-se:

$$\mathbf{B}_{temp} = [2.20, 3.36, 3.60, 2.57, 1.68, 0, 0, 1.56]$$

Cujo arredondamento para granularidade unitária é dado por:

$$\mathbf{B}_{temp} = [2, 3, 4, 3, 2, 0, 0, 2]$$

A soma das taxas normalizadas após arredondamento é 16 bits/s/Hz, ou seja, a solução MA é encontrada. Neste exemplo, não há ajuste das taxas, mas, se fosse o caso, este seria feito de acordo com o comportamento citado no passo 10 da Seção 2.4.4.

Por fim, tem-se:

$$b_1 = \beta B_{temp,1} \cong 2 \frac{\text{bits}}{\text{s.Hz}}$$

$$b_2 = \beta B_{temp,2} \cong 3 \frac{\text{bits}}{\text{s.Hz}}$$

$$b_3 = \beta B_{temp,3} \cong 4 \frac{\text{bits}}{\text{s.Hz}}$$

$$b_4 = \beta B_{temp,4} \cong 3 \frac{\text{bits}}{\text{s.Hz}}$$

$$b_5 = \beta B_{temp,5} \cong 2 \frac{\text{bits}}{\text{s.Hz}}$$

$$b_6 = \beta B_{temp,6} \cong 0$$

$$b_7 = \beta B_{temp,7} \cong 0$$

$$b_8 = \beta B_{temp,8} \cong 2 \frac{\text{bits}}{\text{s.Hz}}$$

De tal forma que:

$$P_1 \cong 2,9 \times 10^{-13} W$$

$$P_2 \cong 2,6 \times 10^{-13} W$$

$$P_3 \cong 4,6 \times 10^{-13} W$$

$$P_4 \cong 4,9 \times 10^{-13} W$$

$$P_5 \cong 4,7 \times 10^{-13} W$$

$$P_6 \cong 0 W$$

$$P_7 \cong 0 W$$

$$P_8 \cong 5,3 \times 10^{-13} W$$

A solução é exposta na Figura 3.30, no MATLAB. É interessante notar que esta solução usa mais potência do que a solução *water-filling* para alcançar 16 bits/s/Hz, o que resulta em uma margem pior do sistema e, conseqüentemente, uma resposta sub-ótima. Além disto, um ponto a ser observado é que o ajuste de potências realizado após o arredondamento em ambos os algoritmos de Chow considera um maior fator para taxas normalizadas que estão mais distantes do valor arredondado. Neste caso, a taxa normalizada do último subcanal, por ser arredondada de 1,56 bit/s/Hz para 2 bits/s/Hz apresenta uma potência de saída maior do que os outros subcanais.

```
[margin_dB bn Pn] = chows_ma_v2(N, Hn, gap, W, No, b, P)

margin_dB =

    -3.9689

bn =

     2     3     4     3     2     0     0     2

Pn =

    1.0e-12 *
    0.2883    0.2611    0.4637    0.4864    0.4669         0         0    0.5277

>> sum(Pn)

ans =

    2.4940e-12

>> sum(bn)

ans =

    16
```

Figura 3.30: Solução Chow MA, saída no console do MATLAB.

Por fim, a última solução apresentada é a de Levin-Campello. O início da solução é igual à apresentada para o caso RA, considerando a mesma tabela de potências incrementais da Figura 3.25. Nesta subseção, opta-se por utilizar a solução *water-filling* arredondada como distribuição de bits inicial, ou seja, $\mathbf{b}_{WF} = [1, 3, 3, 2, 1, 0, 0, 1]$.

A soma das taxas desta distribuição é 11 bits/s/Hz, de maneira que a solução MA deve acrescentar unidades de informação por meio do algoritmo de *b-tightness*. Neste âmbito, o progresso do algoritmo é dado por:

$$[1, 3, 3, 2, 1, 0, 0, 1] \rightarrow [1, 3, 3, 2, 1, 1, 0, 1] \rightarrow [2, 3, 3, 2, 1, 1, 0, 1] \rightarrow [2, 3, 4, 2, 1, 1, 0, 1] \\ \rightarrow [2, 3, 4, 3, 1, 1, 0, 1] \rightarrow [2, 4, 4, 3, 1, 1, 0, 1]$$

Ao adicionar cinco bits às subportadoras que necessitam de menor potência incremental a cada passo, a solução possui 16 bits/s/Hz para o sistema, conforme desejado. Calculando as potências associadas, tem-se:

$$P_1 \cong 2,9 \times 10^{-13} W$$

$$P_2 \cong 5,6 \times 10^{-13} W$$

$$P_3 \cong 4,6 \times 10^{-13} W$$

$$P_4 = 4,9 \times 10^{-13} W$$

$$P_5 \cong 1,6 \times 10^{-13} W$$

$$P_6 \cong 1,8 \times 10^{-13} W$$

$$P_7 \cong 0 W$$

$$P_8 \cong 1,8 \times 10^{-13} W$$

As respostas anteriores para as potências são mais coerentes com a solução *water-filling* do que as encontradas no algoritmo de Chow. Além disto, nota-se, na Figura 3.31, que a margem do sistema está mais próxima do *water-filling* do que a obtida no Chow. O cálculo da margem é dado por:

$$\gamma = \frac{P_r}{P_{utilizada}} = \frac{1,0 \times 10^{-12}}{2,31 \times 10^{-12}} \cong 0,43 \text{ ou } -3,6 dB$$

Por fim, apenas o sétimo subcanal é eliminado, assim como na solução ótima. Para este exemplo, a solução Levin-Campello se mostra melhor do que o algoritmo de Chow neste aspecto. Vale ressaltar, entretanto, que ambas são sub-ótimas se comparadas ao *water-filling*.

```

[margin_dB bn Pn] = LC_v4_MA(N, Hn, gap, W, No, b, P)

margin_dB =

    -3.6419

bn =

     2     4     4     3     1     1     0     1

Pn =

    2.3131e-12

>> sum(bn)

ans =

    16

```

Figura 3.31: Solução Levin-Campello MA, saída no console do MATLAB.

Assim como na seção anterior, o resultado *water-filling* MA é exposto para 64 subportadoras, SNR média de 20dB e taxa total do sistema de 256 bits/s/Hz (ou seja, 4 bits/s/Hz na média por subportadora). A Figura 3.32 mostra tal solução. Novamente é possível notar a distribuição de bits e potência em concordância com os ganhos de cada subportadora impostos pelo canal de multipercurso.

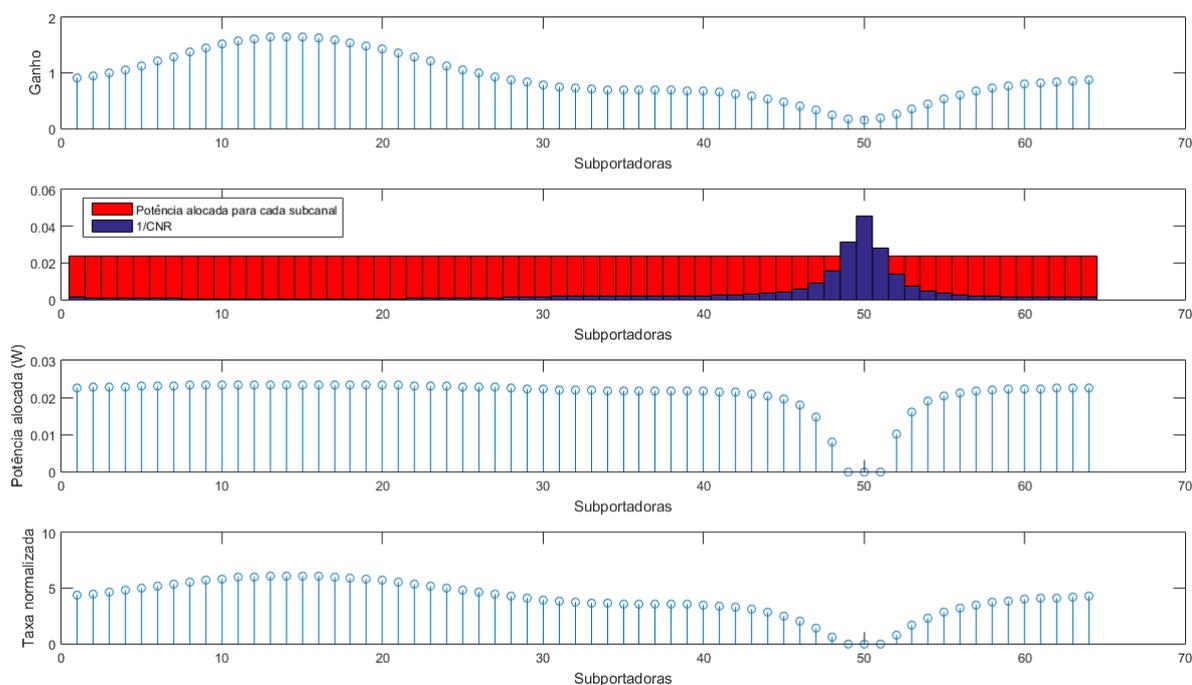


Figura 3.32: Solução *water-filling* MA para 64 subportadoras.

Assim como no caso RA, um resumo para os resultados obtidos nas soluções MA é exposto a partir de então. Considerando primeiramente a alocação de potência para cada subportadora nos exemplos vistos nesta subseção, a Tabela 3.5 mostra os resultados obtidos.

Primeiramente, observa-se que, para a taxa normalizada total desejada, a solução *water-filling* é a que aloca menor potência total para o sistema. Para este exemplo, a solução de Levin-Campello utiliza menos potência do que a solução de Chow. Novamente, a solução de Chow aloca uma grande quantidade de potência para a última subportadora, que possui pior SNR do que a primeira, por exemplo.

Tabela 3.5: Alocação de potência para as soluções MA.

Subportadora	Potência – <i>Water-filling</i> (W)	Potência – Chow (W)	Potência – Levin- Campello (W)
1	$3,3 \times 10^{-13}$	$2,9 \times 10^{-13}$	$2,9 \times 10^{-13}$
2	$3,9 \times 10^{-13}$	$2,6 \times 10^{-13}$	$5,6 \times 10^{-13}$
3	$4,0 \times 10^{-13}$	$4,6 \times 10^{-13}$	$4,6 \times 10^{-13}$
4	$3,6 \times 10^{-13}$	$4,9 \times 10^{-13}$	$4,9 \times 10^{-13}$
5	$2,7 \times 10^{-13}$	$4,7 \times 10^{-13}$	$1,6 \times 10^{-13}$
6	$2,4 \times 10^{-13}$	0	$1,8 \times 10^{-13}$
7	0	0	0
8	$2,5 \times 10^{-13}$	$5,3 \times 10^{-13}$	$1,8 \times 10^{-13}$
Total:	$2,2 \times 10^{-12}$	$2,5 \times 10^{-12}$	$2,3 \times 10^{-12}$

A Tabela 3.6 fornece uma análise das taxas normalizadas obtidas para cada subportadora e o total para o sistema, considerando os três exemplos MA. É possível notar que todas as soluções alcançam a taxa desejada, porém, novamente, a solução *water-filling* considera granularidade infinita do sistema. Por este motivo, a potência total alocada nesta solução é a menor dentre as três estudadas, o que ressalta o fato de que os algoritmos de Chow e Levin-Campello são sub-ótimos também na solução MA.

Tabela 3.6: Alocação de taxa normalizada para as soluções MA.

Subportadora	Taxa Normalizada – <i>Water-filling</i> (bits/s/Hz)	Taxa Normalizada – Chow (bits/s/Hz)	Taxa Normalizada – Levin-Campello (bits/s/Hz)
1	2,14	2	2
2	3,51	3	4
3	3,78	4	4
4	2,62	3	3
5	1,45	2	1
6	1,21	0	1
7	0	0	0
8	1,28	2	1
Total:	16,00	16,00	16,00

3.2.3 Análise comparativa

Nas subseções anteriores, foram expostos os funcionamentos de cada um dos algoritmos analisados, com exemplos de execução e apresentação dos resultados obtidos.

Nesta seção, uma breve análise comparativa entre os algoritmos é realizada, considerando a solução RA. Sabe-se que esta solução propõe a maximização da taxa normalizada, de acordo com uma restrição de potência. Desta maneira, foi visto que a solução *water-filling* e o algoritmo de Chow alcançam exatamente a restrição de potência imposta, enquanto a solução de Levin-Campello alcança um resultado com taxas normalizadas inteiras que não necessariamente utiliza toda a potência disponível no transmissor.

A Figura 3.33 mostra a potência total utilizada pelo sistema, em função da SNR, para os algoritmos de alocação estudados até então. Consideram-se os mesmos ganhos propostos em (127), com oito subportadoras, e a mesma perda de percurso, de tal maneira que os valores de potência representados estão dimensionados para o transmissor. Assume-se que a potência de ruído é mantida fixa (com o mesmo valor utilizado nas subseções anteriores) e a potência do transmissor é variada de tal forma a proporcionar uma SNR média de 0 a 30 dB, no receptor.

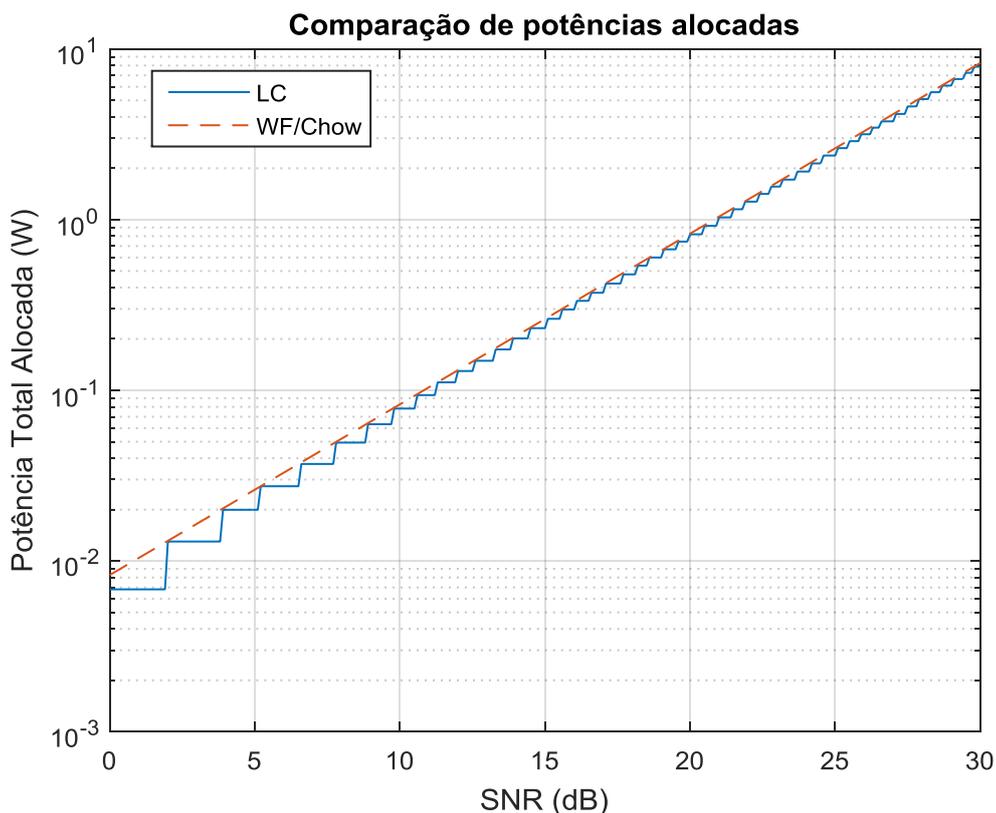


Figura 3.33: Alocação de potência para os algoritmos de alocação de recursos analisados.

Os gráficos anteriores mostram que a potência total alocada pelo algoritmo de Levin-Campello tende a alcançar a potência total do transmissor, para altos valores de SNR. Em baixos valores de SNR a margem obtida é maior, o que indica que os valores de potência incrementais são altos se comparados à potência de margem restante. Intuitivamente, pode-se pensar que em baixas SNRs a potência necessária para aumentar um bit de granularidade é maior e, se este for adicionado, o critério RA deixa de ser respeitado.

Também é possível analisar o desempenho dos algoritmos com relação à taxa normalizada. A Figura 3.34 mostra, para os ganhos de (127), a taxa normalizada média por subportadora em função da SNR para os algoritmos de *water-filling*, Levin-Campello e Chow.

Nota-se que o comportamento para o algoritmo de Levin-Campello é muito semelhante ao caso anterior de alocação de potência. Para baixos valores de SNR, a margem do sistema é maior e é necessário um intervalo de SNR maior para que a taxa normalizada por subportadora aumente em uma unidade de informação

(considerando a granularidade do sistema). Com o aumento da SNR, tem-se maior proximidade com o caso ótimo.

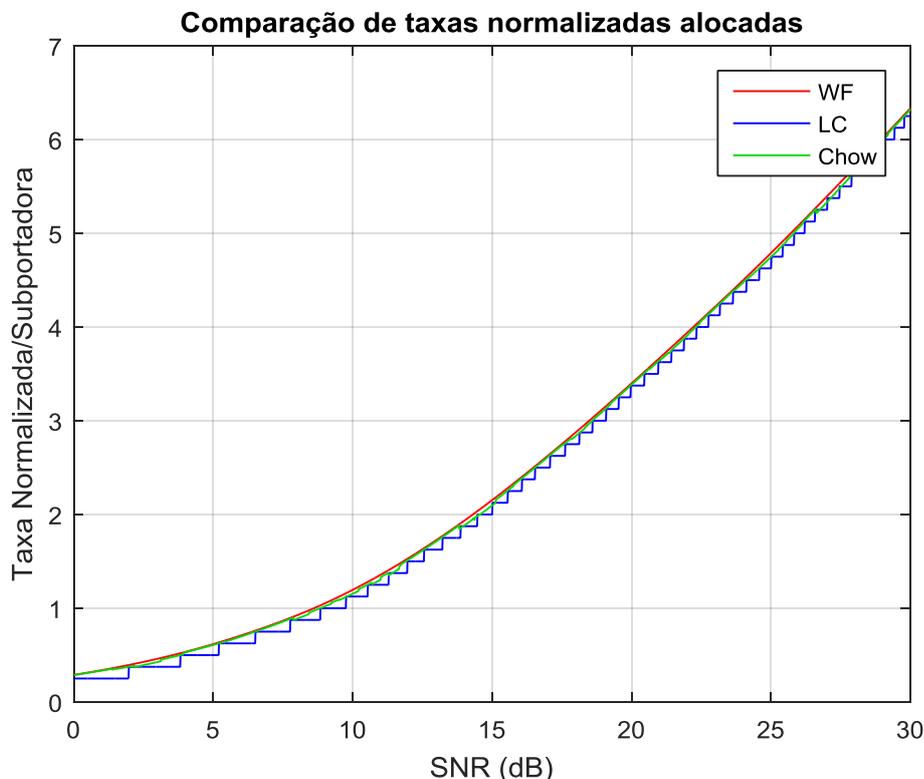


Figura 3.34: Comparação entre taxas normalizadas por subportadora em função da SNR para os algoritmos analisados.

Por outro lado, o algoritmo de Chow está muito próximo da solução *water-filling*, como é possível de observar, mas não a ultrapassa, já que é uma solução sub-ótima. Neste caso, por se considerar, no algoritmo inicial de Chow, uma otimização com taxa normalizada que assume granularidade infinita do sistema, a taxa média por subportadora é maior do que no algoritmo de Levin-Campello.

O algoritmo de Levin-Campello, por sua vez, é completamente discreto desde o início e fornece uma abordagem diferente, baseada na adição de unidades de informação tendo em vista as potências incrementais de cada subportadora.

No exemplo considerado, foi proposto o valor de *gap* de SNR como 8,8dB, proporcionando uma BER de aproximadamente 10^{-6} para o sistema. Um valor aproximado para BER de 10^{-8} é de 10,2dB (CIOFFI, 2008). A Figura 3.35 mostra o

que ocorre com a taxa normalizada por subportadora se este valor de *gap* é utilizado pelo mesmo sistema considerando (127), em comparação com a Figura 3.34.

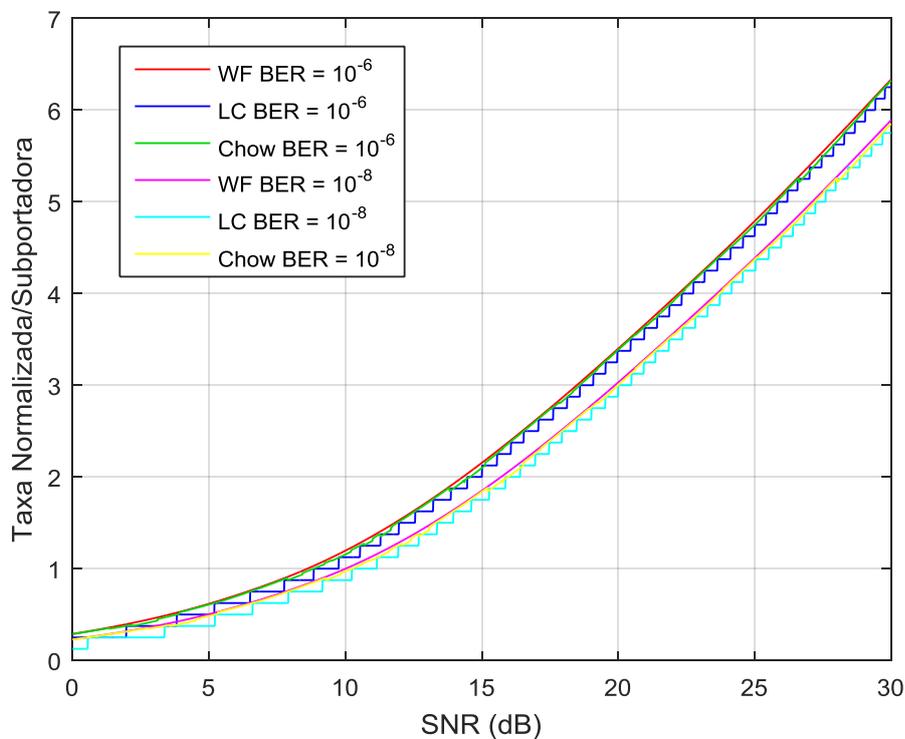


Figura 3.35: Comparação entre taxas normalizadas por subportadora em função da SNR para os algoritmos analisados, com modificação do *gap* de SNR.

Com a análise da figura anterior, é possível observar que, para obter uma BER menor, a taxa por subportadora é reduzida, mantendo-se a mesma SNR do sistema. Isto é coerente, já que, ao diminuir a taxa normalizada em cada subportadora (e conseqüentemente as ordens das modulações em cada uma das destas), a BER diminui, como considerado em (41) e (47). Entretanto, isto acarreta em perda na taxa normalizada total do sistema, como esperado.

4. CONSIDERAÇÕES FINAIS

Neste trabalho, os conceitos básicos da técnica de OFDM e alocação de recursos foram estudados. Inicialmente, um estudo teórico da técnica de OFDM e esquemas de modulação foi realizado, tendo em vista o breve desenvolvimento de um modelo matemático discreto para um sistema OFDM e sua posterior simulação com o MATLAB.

Na simulação, aspectos básicos de desempenho foram visualizados para modulações QAM e PSK com o uso de OFDM, incluindo também codificação. Sendo assim, uma noção geral de desempenho em canais de multipercurso foi obtida, em contraste com o modelo AWGN visto durante a graduação.

Também foram analisados algoritmos de alocação de recursos em sistemas multiportadora, que buscam o controle de taxa e potência no momento da transmissão com o intuito de melhorar o desempenho geral do sistema. Foi visto que existem dois tipos de abordagens, a saber, *Rate Adaptive* e *Margin Adaptive*, cada qual com suas restrições impostas no sistema.

A solução ótima de *water-filling* e os algoritmos de Chow e Levin-Campello foram simulados em ambas as abordagens, de tal forma que suas principais características foram observadas e uma comparação entre estas foi feita. Neste âmbito, o algoritmo de *water-filling* representa soluções que consideram granularidade infinita do sistema, enquanto as soluções de Chow e Levin-Campello buscam adaptar esta solução ótima para casos práticos, onde a granularidade é finita. Entretanto, sabe-se que as duas últimas soluções são sub-ótimas (LI e STÜBER, 2006) e este comportamento foi confirmado neste trabalho.

Por fim, considera-se que os conceitos estudados nesta monografia foram um ótimo complemento aos vistos na graduação de Engenharia de Informação. Como continuidade a este trabalho, seria interessante verificar com maiores detalhes o desempenho de sistemas utilizando os algoritmos de alocação de recursos estudados, incluindo também a utilização de múltiplas antenas transmissoras/receptoras (sistemas MIMO) e codificação.

REFERÊNCIAS BIBLIOGRÁFICAS

- ARMADA, A. A. SNR gap approximation for M-PSK-Based bit loading. **IEEE Transactions on Wireless Communications**, v. 5, n. 1, p. 57-60, Janeiro 2006.
- BAHAI, A. R. S.; SALTZBERG, B. R. **Multi-carrier Digital Communications: Theory and Applications of OFDM**. 2^a. ed. Kluwer Academic Publishers, 1999.
- BEDNORZ, W. **Advances in Greedy Algorithms**. 1. ed. In-Teh, 2008.
- BOYD, S.; VANDENBERGHE, L. **Convex Optimization**. 1. ed. Cambridge: Cambridge University Press, 2004.
- CAMPELLO, J. **Practical bit loading for DMT**. IEEE International Conference on Communications.1999. p. 801-805.
- CASELLA, I. R. S. **Introdução ao Canal de Radiopropagação**. Universidade Federal do ABC. Santo André, p. 240. 2015.
- CASELLA, I. R. S. **Sistemas Multiportadoras**. Universidade Federal do ABC. Santo André, p. 66. 2015.
- CHANG, R. W. Synthesis of band-limited orthogonal signals for multi-channel data transmission. **The Bell System Technical Journal**, v. 45, n. 10, p. 1775-1796, Dezembro 1966.
- CHOO, Y. S. et al. **MIMO-OFDM Wireless Communications with MATLAB**. 1^a. ed. John Wiley & Sons, Ltd., 2010.
- CHOW, P.; CIOFFI, J.; BINGHAM, J. A Practical Discrete Multitone Transceiver Loading Algorithm for Data Transmission over Spectrally Shaped Channels. **IEEE Transactions on Communications**, v. 43, n. 234, p. 773-775, Agosto 2002.
- CIOFFI, J. M. **Multi-channel Modulation**. Stanford University. p. 134. 2008.
- ENGELS, M. **Wireless OFDM Systems: How to Make them Work?** 1^a. ed. Springer, 2002.

HANZO, L.; KELLER, T. **OFDM and MC-CDMA: A Primer**. 1ª. ed. John Wiley & Sons, Ltd., 2006.

HAYES, M. H. **Statistical Digital Signal Processing and Modeling**. 1ª. ed. John Wiley & Sons, Inc., 1996.

HAYKIN, S. **Sistemas de Comunicação Analógicos e Digitais**. 4ª. ed. Bookman, 2007.

HAYKIN, S.; MOHER, M. **Introdução aos Sistemas de Comunicação**. 2ª. ed. Bookman, 2008.

HWANG, J. **Simplified Channel Estimation Techniques for OFDM Systems with Realistic Indoor Fading Channels**. Waterloo University. Ontario, p. 74. 2009.

LATHI, B. P. **Modern Digital and Analog Communication Systems**. 3ª. ed. Oxford University Press, Inc., 1998.

LATHI, B. P.; DING, Z. **Modern Digital and Analog Communication Systems**. 4ª. ed. Oxford University Press, 2009.

LEVIN, H. E. **A complete and optimal data allocation method for practical discrete multitone systems**. IEEE Global Telecommunications Conference. 2001. p. 369-374.

LI, Y. G.; STÜBER, G. L. **Orthogonal Frequency Division Multiplexing for Wireless Communications**. 1ª. ed. Springer, 2006.

LINNARTZ, J. M. G. Wireless Communication. Disponível em: <<http://www.wirelesscommunication.nl/reference/images/doppspec.gif>>. Acesso em: 23 Novembro 2015.

MA, Y. et al. 1-Tb/s single-channel coherent optical OFDM transmission over 600-km SSMF fiber with subwavelength bandwidth access. **Optics Express**, v. 17, n. 11, p. 9421-9427, 2009.

MATHWORKS. MATLAB - MathWorks. Disponível em: <<http://www.mathworks.com/products/matlab/>>. Acesso em: 20 Fevereiro 2016.

PRASAD, R. **OFDM for Wireless Communication Systems**. 1^a. ed. Artech House, Inc., 2004.

PROAKIS, J. G.; SALEHI, M. **Communication Systems Engineering**. 2^a. ed. Prentice Hall, 2002.

RAPPAPORT, T. S. **Comunicações sem fio: Princípios e práticas**. 2^a. ed. Prentice Hall, 2008.

RETHNAKARAN, P.; DAWID, H. **Orthogonal Frequency Division Multiplexing**. p. 27. 2003.

ROHLING, H. **OFDM: Concepts for Future Communication Systems**. 1^a. ed. Springer, 2011.

SEONG, K.; MOHSENI, M.; CIOFFI, J. M. Optimal Resource Allocation for OFDMA Downlink Systems. **Information Theory, 2006 IEEE International Symposium**, 2006. 1394-1398.

SHIEH, W.; BAO, H.; TANG, Y. Coherent optical OFDM: Theory and Design. **Optics Express**, v. 16, n. 2, p. 841-859, 2008.

TAROKH, V. **New Directions in Wireless Communications Research**. 1^a. ed. Springer, 2009.

VILAR, F. C. **Implementation of Zero Forcing and MMSE Equalization Techniques in OFDM**. Universidade de Fortaleza. Fortaleza, p. 52. 2014.

WADE, J. G. **Signal Coding and Processing**. 2^a. ed. Cambridge University Press, 1994.

WANG, N.; BLOSTEIN, S. D. **Power loading for CP-OFDM over frequency-selective fading channels**. IEEE Global Telecommunications Conference. 2003. p. 2305-2309.

YAGHOUBI, H. Scalable OFDMA Physical Layer in IEEE 802.16 Wireless MAN. **Intel Technology Journal**, v. 8, n. 3, p. 201-212, Agosto 2004.

ZIVKOVIC, M. **Design and Implementation of Adaptive OFDM in a Software Defined Radio Framework**. Aachen University. Aachen, p. 225. 2013.

ANEXO A: Distribuição de Rayleigh

Considerando um sinal aleatório dado pela soma de dois sinais de ruído de quadratura gaussiana. Sendo X e Y variáveis aleatórias gaussianas, este sinal pode ser escrito como:

$$y(t) = X\cos(\omega t) + Y\sin(\omega t) \quad (\text{A1})$$

Como X e Y são independentes, de média zero e variância σ^2 , a função de probabilidade conjunta destas variáveis aleatórias é dada por:

$$f_{XY}(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{1}{2}\frac{(x^2+y^2)}{\sigma^2}} \quad (\text{A2})$$

Para obter a função de densidade de probabilidade do sinal recebido, é mais simples realizar a transformação a seguir:

$$y(t) = R\cos(\omega t - \theta) \quad (\text{A3})$$

Onde:

$$R = \sqrt{X^2 + Y^2} \text{ e } \theta = \text{atan}\left(\frac{Y}{X}\right) \quad (\text{A4})$$

Seja $r = g(x, y)$ e $\theta = h(x, y)$ duas funções com valores regidos pelas variáveis aleatórias X e Y . Se $r \geq 0$ e $0 \leq \theta \leq 2\pi$, tem-se:

$$r = g(x, y) = \sqrt{x^2 + y^2} \text{ e } \theta = h(x, y) = \text{atan}\left(\frac{y}{x}\right) \quad (\text{A5})$$

A próxima etapa é encontrar as funções inversas $x = q(r, \theta)$ e $y = p(r, \theta)$. Sendo assim, tem-se:

$$r = \sqrt{x^2 + y^2} \rightarrow x = \sqrt{r^2 - y^2}$$

Mas $y = r \cdot \text{sen}(\theta)$, então:

$$x = q(r, \theta) = \sqrt{r^2 - y^2} = \sqrt{r^2 - r^2 \text{sen}^2(\theta)} = r \sqrt{1 - \text{sen}^2(\theta)} = r \cos(\theta) \quad (\text{A6})$$

Por outro lado:

$$\theta = \text{atan}\left(\frac{y}{x}\right) \rightarrow \tan(\theta) = \frac{y}{x} \rightarrow y = x \tan(\theta)$$

Como $x = r \cdot \cos(\theta)$, então:

$$y = p(r, \theta) = x \tan(\theta) = r \cos(\theta) \frac{\text{sen}(\theta)}{\cos(\theta)} = r \text{sen}(\theta) \quad (\text{A7})$$

Com as funções inversas, é possível calcular o Jacobiano em função das coordenadas polares, a saber:

$$J(r, \theta) = \begin{vmatrix} \frac{\partial x}{\partial r} & \frac{\partial x}{\partial \theta} \\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial \theta} \end{vmatrix} = \begin{vmatrix} \cos(\theta) & -r \text{sen}(\theta) \\ \text{sen}(\theta) & r \cos(\theta) \end{vmatrix} = r \cos^2(\theta) + r \text{sen}^2(\theta) = r \quad (\text{A8})$$

Desta maneira, a função de probabilidade conjunta é dada por:

$$f_{R\theta}(r, \theta) = f_{XY}(x, y) |J(r, \theta)| = \frac{r}{2\pi\sigma^2} e^{-\frac{1r^2}{2\sigma^2}} \quad (\text{A9})$$

Encontrando as funções de distribuição de probabilidade marginais, tem-se:

$$f_R(r) = \int_0^{2\pi} \frac{r}{2\pi\sigma^2} e^{-\frac{1r^2}{2\sigma^2}} d\theta = \frac{r}{\sigma^2} e^{-\frac{1r^2}{2\sigma^2}} \quad (\text{A10})$$

Que é a distribuição de Rayleigh, com $r \geq 0$. Com relação à fase, tem-se:

$$f_{\theta}(\theta) = \int_0^{\infty} \frac{r}{2\pi\sigma^2} e^{-\frac{1r^2}{2\sigma^2}} dr = \frac{1}{2\pi} \quad (\text{A11})$$

Que representa uma distribuição de fase uniforme, com $0 \leq \theta \leq 2\pi$.