

UNIVERSIDADE FEDERAL FLUMINENSE
ESCOLA DE ENGENHARIA
MESTRADO EM ENGENHARIA DE TELECOMUNICAÇÕES

ONÉSIMO FERREIRA

ANÁLISE DO DESEMPENHO DA TECNOLOGIA OFDM PARA OS SISTEMAS
MÓVEIS DE QUARTA GERAÇÃO

Influência da Densidade de Pilotos no Desempenho do Sistema OFDM

Vol. 1

NITERÓI
2010

ONÉSIMO FERREIRA

ANÁLISE DO DESEMPENHO DA TECNOLOGIA OFDM PARA OS SISTEMAS
MÓVEIS DE QUARTA GERAÇÃO

Influência da Densidade de Pilotos no Desempenho do Sistema OFDM

Dissertação apresentada ao Curso de Pós-Graduação em Engenharia de Telecomunicações da Universidade Federal Fluminense, como requisito parcial para obtenção do Grau de Mestre. Área de Concentração: Sistemas de Telecomunicações.

Orientador: Prof. Dr. EDUARDO RODRIGUES VALE

NITERÓI

2010

Ficha Catalográfica elaborada pela Biblioteca da Escola de Engenharia e Instituto de Computação da UFF

F383 Ferreira, Onésimo.

Análise do desempenho da tecnologia OFDM para os sistemas móveis de quarta geração: influência da densidade de pilotos no desempenho do sistema OFDM / Onésimo Ferreira. – Niterói, RJ: [s.n.], 2010.
310 f.

Dissertação (Mestrado em Engenharia de Telecomunicações) - Universidade Federal Fluminense, 2010.

Orientador: Eduardo Rodrigues Vale.

1. Sistema de comunicação móvel. 2. Multiplexação por divisão de frequência ortogonal. 3. Telefonia celular. 4. Engenharia de telecomunicação. I. Título.

CDD 621.38456

ONÉSIMO FERREIRA

ANÁLISE DO DESEMPENHO DA TECNOLOGIA OFDM PARA OS SISTEMAS
MÓVEIS DE QUARTA GERAÇÃO

Influência da Densidade de Pilotos no Desempenho do Sistema OFDM

Dissertação apresentada ao Curso de Pós-Graduação em Engenharia de Telecomunicações da Universidade Federal Fluminense, como requisito parcial para obtenção do Grau de Mestre. Área de Concentração: Sistemas de Telecomunicações.

Aprovada em Agosto de 2010.

BANCA EXAMINADORA

Prof. Dr. Eduardo Rodrigues Vale - Orientador
UFF

Prof^a. Dr^a. Leni Joaquim de Matos
UFF

Prof. Dr. Julio Cesar Rodrigues Dal Bello
UFF

Prof. Dr. José Santo Guiscafré Panaro
UFF

Prof. Dr. Eduardo Javier Arancibia Vasquez
CLARO

Prof. Dr. Marco Antonio Grivet Mattoso Maia
PUC - RJ

NITERÓI
2010

Aos meus pais.

AGRADECIMENTOS

Agradeço ao Prof. Dr. Eduardo Rodrigues Vale pela sua colaboração.

RESUMO

O objetivo desta dissertação é a análise do desempenho da tecnologia OFDM para os sistemas móveis de quarta geração. Composta de 7 capítulos, abrange todos os elementos necessários para a compreensão da técnica baseada em multiportadoras OFDM. O Capítulo 1 é introdutório, e apresenta uma breve visão geral das tecnologias de acesso celular à Internet, já existentes e emergentes. O Capítulo 2 mostra, sucintamente, as características dos canais com desvanecimento em ambientes com multipercursos de sinal. É o conhecimento básico para compreensão dos fenômenos que ocorrem nestes canais, e é essencial para a análise teórica e para a avaliação do desempenho, em canais de rádio, das técnicas baseadas em multiportadoras. O Capítulo 3 mostra a história e o princípio das técnicas baseadas em multiportadoras, incluindo a técnica OFDM. Inclui a história da origem até a forma atual. O estudo aborda o uso da transformada discreta de Fourier, a inserção de intervalo de guarda, a inserção de prefixo cíclico para a atual forma do OFDM e finaliza, mostrando os parâmetros do sistema OFDM em interface aérea LTE (*Long Term Evolution*). O Capítulo 4 discute as características dos sistemas OFDM. Apresenta um modelo do canal rádio, onde um sinal transmitido não é somente distorcido pelo desvanecimento multipercurso, mas também corrompido pelo ruído térmico. Mostra a relação entre o número de subportadoras, a seletividade em frequência e a seletividade no tempo. Mostra, também, a relação entre o comprimento do intervalo de guarda e a seletividade em frequência. Enfatiza a análise teórica e discute as vantagens e as desvantagens do sistema OFDM, incluindo a robustez contra o desvanecimento seletivo em frequência, a robustez contra os ruídos impulsivos, a sensibilidade ao deslocamento de frequência, a sensibilidade à amplificação não-linear e a sensibilidade para conversões A/D (analógico para digital) e D/A (digital para analógico). O Capítulo 5 apresenta um método de estimação de canal e as diferentes possibilidades para alocação de pilotos, tais como, alocação de piloto espaçado no tempo e considerando todas as frequências, alocação de piloto espaçado na frequência e considerando tempo integral e alocação de piloto espaçado no tempo e espaçado na frequência. Para finalizar é mostrado o sistema OFDM com símbolos pilotos utilizados para estimação de canal. O Capítulo 6 trata da parte de simulação da plataforma OFDM, explorando aspectos tais como alternativas para uma melhoria da relação sinal-ruído do sistema, e apresenta sugestões para trabalhos futuros. O Capítulo 7 apresenta as conclusões.

Palavras-chave: Multipercursos, Desvanecimento, OFDM, Estimação de Canal, Pilotos, Desempenho, BER, Vazão.

ABSTRACT

The objective of this dissertation is to analyze the performance of OFDM technology for the fourth generation mobile systems. Consisting of seven chapters, it covers all the elements necessary to understand the technique based on OFDM multicarrier. The chapter 1 is introductory, and it gives a brief overview of the technologies of mobile internet access. The chapter 2 briefly shows the characteristics of channels with fading in environments with ray multipath. It is the basic knowledge to understand the phenomena that occur in these channels, and is essential for the theoretical analysis and for the evaluation of the performance, in radio channels, of the techniques based on multicarrier. The chapter 3 shows the history and the principle of multicarrier-based techniques, including OFDM technique. Includes the history from the origin until the present shape. The study discusses the use of the discrete Fourier transform, the insertion of the guard interval, the insertion of the cyclic prefix for the current form of OFDM and ends showing the parameters of OFDM system in LTE (Long Term Evolution) air interface. The chapter 4 discusses the characteristics of OFDM systems. Presents a model of the radio channel, where a transmitted signal is not only distorted by multipath fading, but also corrupted by thermal noise. Shows the relationship between the number of subcarriers, the frequency selectivity and the selectivity in time. It also shows the relationship between the length of guard interval and the frequency selectivity. Emphasizes the theoretical analysis and discusses the advantages and the disadvantages of OFDM system, including the robustness against the frequency selective fading, the robustness against the impulsive noise, the sensitivity to frequency offset, the sensitivity to nonlinear amplifications and the sensitivity to conversions A / D (analog to digital) and D / A (digital to analog). The chapter 5 presents a method to estimate the channel and the different possibilities for allocating pilots, such as allocation of pilot spaced in time and considering all the frequencies, the allocation of pilot spaced in frequency and considering full time and the allocation of the pilot spaced in time and spaced in frequency. To finish is showed the OFDM system with pilot symbols used for channel estimation. The chapter 6 discusses the simulation of OFDM platform, exploring issues such as alternatives for improving the signal-noise ratio of the system, and presents suggestions for future works. The chapter 7 presents the conclusions.

Keywords: *Multipath, Fading, OFDM, Channel Estimation, Pilots, Performance, BER, Throughput.*

JUSTIFICATIVA E OBJETIVO

Alguns estudos têm sido feitos para analisar o desempenho de sistemas OFDM, por exemplo, Athanasios Doukas e Grigorios Kalivas (*Department of Electrical and Computer Engineering, Campus of Rion, Achaia, 26500, Greece, 2006*) analisaram a estimação do canal, em sistemas OFDM, utilizando interpolação linear. A estimação do canal e a análise do desempenho foram feitas através dos parâmetros MSE (*Mean Square Error*) e BER (*Bit Error Rate*) e caracterizadas pelo número e distância de pilotos inseridos e pelo espalhamento *Doppler*. Eles examinaram a BER e MSE, utilizando espalhamento *Doppler* de 0Hz, modulação BPSK e QPSK com 4, 5 e 6 pilotos e relação sinal-ruído (SNR) variando de 4 a 18 dB. A análise mostrou que MSE e BER são impactados pela seletividade em frequência dos canais. Mostrou, também, a importância da inserção de pilotos na eliminação de muitos erros na interpolação.

Dongxu Shen, Kai-Kit Wong, Zhengang Pan e Victor O. K. (*Department of Electrical & Electronic Engineering, the University of Hong Kong, Pokfulam Road, Hong Kong, China, 2003*) examinaram o *throughput* (vazão) em sistemas OFDM, com estimação de canais assistidos por pilotos. Especificamente, o foco da investigação era a influência da densidade de pilotos e do número de usuários no desempenho do sistema OFDM. Eles mostraram que o desempenho da BER melhora consistentemente com a inserção de mais pilotos, mas a vazão reduz devido à inserção de pilotos. O número ótimo de pilotos pode ser determinado pelo esquema de modulação, para maximizar a vazão efetiva, e em função da relação sinal-ruído (SNR). Resultados da simulação mostraram que o desempenho do sistema degrada com muitos usuários. As simulações feitas foram BER e vazão, em função das modulações QPSK, 16QAM, e 64-QAM, com espaçamento entre frequências $M_f = 4$ ou 8 e espaçamento no tempo $M_t = 3, 4$ ou 5, quantidade de usuários $U = 3$ e E_b/N_0 variando de 5 a 30 dB.

Sinem Coleri, Mustafa Ergen, Anuj Puri e Ahmad Bahai (*Department of Electrical Engineering and Computer Science at University of California, Berkeley, USA, 2002*) estudaram estimação de canal em sistemas OFDM com técnicas baseadas em pilotos (*Least Square - LS* e *Minimum Mean-Square-Error - MMSE*). Inicialmente, apresentaram um sistema OFDM em banda base com inserção de pilotos e inserção de intervalo de guarda. Abordaram inserção de pilotos no domínio do tempo e no domínio da frequência. As simulações feitas foram BER em função da relação sinal-ruído (SNR), que variava de 10 a 40 dB, e em função da modulação BPSK, QPSK, 16QAM e DPSK, com desvanecimento Rayleigh, e passando por um canal AWGN e com frequência *Doppler* de 70 Hz. Para a modulação 16QAM, foi feita a análise da BER em função da frequência *Doppler*, variando de 20 a 100Hz. Os resultados das simulações mostraram que o desempenho da interpolação com estimação de canal baseado em pilotos é muito melhor que a estimação baseada em algoritmos. As simulações também mostraram que a degradação do desempenho pode ser tolerada para altas taxas de dados com baixos valores de espalhamento *Doppler*.

Chi-Min Li, Hui-Che Chen, Po-Jen Wang, Jia-Chyi Wu e I-Tseng Tang (*Department of Electrical & Electronic Engineering, the University of Hong Kong, Pokfulam Road, Hong Kong, China, 2008*) também propuseram métodos baseados em pilotos para sincronização no tempo e estimação de canal em sistemas OFDM.

Yushi Shen e Ed Martinez (2006) apresentaram um modelo em banda base de um sistema OFDM, com divisão do espectro de frequências disponível em várias

subportadoras. O modelo consistia, basicamente, de modulação/demodulação, inserção/remoção de banda de guarda, uso de IDFT/DFT, inserção/remoção de prefixo cíclico, canal com desvanecimento por multipercursos e introdução de ruído gaussiano (AWGN). O estudo abordou estimação de canal por piloto no domínio da frequência (*Block-type*: LS, MMSE e MMSE modificado) e no domínio do tempo (*Comb-type*: LS, LI, SOI, LPI, SCI e TDI), mostrando o desempenho (SER vs. SNR) destes diversos métodos (SER - *Symbol Error Rate*).

Ali Ramadan, Tariq J. Khanzada e Abbas Omar (*Faculty of Electrical Engineering, University of Magdeburg, Germany, 2009*) apresentaram proposta para cancelamento de interferências entre subportadoras e estimação de canal utilizando interpolação de Newton e filtro de Wiener.

Todos os estudos citados acima focam no desempenho dos sistemas OFDM em função das modulações. Estudos que focam no desempenho de sistemas OFDM em função da variação da quantidade de pilotos inserida ainda são bastante vagos e superficiais. Por exemplo, nos estudos acima, apenas um deles varia a quantidade de pilotos utilizada e varia muito pouco (apenas 4, 5 e 6 pilotos), não mostrando, detalhadamente, o desempenho dos sistemas OFDM.

O objetivo deste trabalho é preencher esta lacuna, ou seja, mostrar o desempenho de sistemas OFDM em função da variação da quantidade de pilotos inserida; conforme dito no Resumo acima, a abordagem inicial será teórica, a fim de alcançar o pleno entendimento da tecnologia OFDM. Feito isso, será apresentada e explicada detalhadamente a plataforma OFDM feita no MATLAB (Apêndices A, B e C) e, em seguida, serão apresentados os resultados das diversas simulações realizadas nesta plataforma. Conforme será visto adiante, serão transmitidos quadros de 300 bits e a quantidade de pilotos inseridos variará consideravelmente (2, 3, 4, 5, 6, 7, 11, 13, 16, 21, 26, 31, 51, 61, 76, 101 e 151 pilotos), com o objetivo de retratar plenamente os efeitos da variação das quantidades de pilotos inseridos no desempenho dos sistemas OFDM.

Os resultados esperados são:

- BER vs. E_b/N_0 com a melhor curva de interpolação (variando, também, a quantidade de pilotos).
- Vazão vs. E_b/N_0 (variando, também, a quantidade de pilotos).
- BER e Vazão vs. E_b/N_0 (variando, também, a quantidade de pilotos).
- BER vs. Pilotos.
- Vazão vs. Pilotos.
- BER e Vazão vs. Pilotos.
- Vazão Efetiva vs. E_b/N_0 (variando, também, a quantidade de pilotos).
- BER vs. E_b/N_0 com 16 Pilotos e considerando aproveitamento da redundância dos bits de prefixo cíclico.

Para um aprofundamento das diversas simulações realizadas na plataforma OFDM, recomenda-se a leitura dos Apêndices, cujos objetivos são:

- Apêndice A: apresentar a plataforma OFDM desenvolvida no MATLAB (*Simulink*).
- Apêndice B: apresentar os *OFDM Transmitter* (Transmissores OFDM) e os *LMS Channel Estimator* (Estimadores de Canal LMS) das 17 Plataformas OFDM desenvolvidas, bem como o *Received Signal* (Sinal Recebido) e o *Spectrum Scope* (Escopo do Espectro), referentes a 13 pilotos. Trata-se de um aprofundamento do Apêndice A.
- Apêndice C: disponibilizar todos os programas feitos para as 17 plataformas OFDM.

SUMÁRIO

1. INTRODUÇÃO	19
1.1. Sistemas de Comunicações Móveis	19
1.2. Evolução para os Sistemas 4G	22
1.3. Técnicas de Multiportadoras para Sistemas 4G	24
1.4. 3GPP - Evolução a Longo Prazo (LTE - <i>Long Term Evolution</i>) e OFDM	25
1.4.1. <i>EUTRAN Interface de Rádio</i>	26
1.5. Conclusão	27
2. CARACTERÍSTICAS DOS CANAIS COM DESVANECIMENTO EM AMBIENTES COM MULTIPERCURSOS DE RAIOS	29
2.1. Introdução	29
2.2. Caracterização do Fenômeno de Multipercurso	29
2.3. Canais com Desvanecimento <i>Rayleigh e Rice</i>	31
2.4. Canais com Desvanecimento Seletivo em Frequência e Não Seletivo em Frequência	31
2.5. Canais com Desvanecimento Seletivo no Tempo e Não Seletivo no Tempo	31
2.6. Exemplos de Canais com Desvanecimento por Multipercursos	32
2.7. Conclusão	34
3. PRINCÍPIOS E HISTÓRIA DA MODULAÇÃO POR MULTIPORTADORAS (MCM - MULTICARRIER MODULATION) E OFDM (ORTHOGONAL FREQUENCY DIVISION MULTIPLEXING)	35
3.1. Introdução	35
3.2. Origem do OFDM	38
3.3. Uso da Transformada Discreta de Fourier	44
3.4. Inserção de prefixo cíclico para a atual forma de OFDM	48
3.5. Parâmetros do Sistema OFDM em Interface Aérea LTE (<i>Long Term Evolution</i>)	58
3.6. Conclusão	60
4. CARACTERÍSTICAS DO OFDM	62
4.1. Introdução	62
4.2. Modelo do Canal Rádio	64
4.3. Número Ótimo de Subportadoras e Comprimento Ótimo do Intervalo de Guarda	65
4.4. Robustez contra Desvanecimento Seletivo em Frequência	68
4.5. Robustez contra Ruídos Artificiais	69
4.6. Sensibilidade ao Deslocamento de Frequência	69
4.7. Sensibilidade a Amplificadores não-lineares	70
4.8. Sensibilidade para Conversões A/D e D/A	71
4.9 Conclusão	72
5. ESTIMAÇÃO DE CANAL E ALOCAÇÃO DE PILOTO	74
5.1. Introdução	74
5.2. Estimação de Canal	75
5.2.1. <i>Interpretação no Domínio do Tempo-Frequência e Alocação de Piloto</i>	78
5.3. OFDM com símbolos piloto para a estimação de canal	83
5.4. Conclusão	86
6. RESULTADOS DA SIMULAÇÃO DA PLATAFORMA OFDM NO MATLAB (SIMULINK)	87

6.1. Introdução	87
6.2. BER vs. Eb/No com a melhor curva de interpolação	87
6.3. Vazão vs. Eb/No	96
6.4. Taxa de Erro e Vazão vs. Eb/No.....	104
6.5. BER vs. Pilotos	107
6.6. Vazão vs. Pilotos.....	113
6.7. Taxa de Erro e Vazão vs. Pilotos	120
6.8. Vazão Efetiva vs. Eb/No	122
6.9. BER vs. Eb/No com 16 Pilotos e considerando aproveitamento da redundância dos bits de prefixo cíclico.....	128
6.10. Sugestões de simulações e testes futuros que podem ser realizados com a Plataforma OFDM	129
6.11. Conclusão	132
7. CONCLUSÃO FINAL.....	134
8. REFERÊNCIAS BIBLIOGRÁFICAS	141
A. APÊNDICE - PLATAFORMA OFDM NO MATLAB (<i>SIMULINK</i>)	161
B. APÊNDICE - TRANSMISSOR OFDM, ESTIMADOR DE CANAL LMS, SINAL RECEBIDO E ESCOPO DO ESPECTRO.....	236
C. APÊNDICE - PROGRAMAS FEITOS PARA A PLATAFORMA OFDM	254

LISTA DE ILUSTRAÇÕES

Figura 2.1 - Exemplo de uma transmissão com multipercursos de sinal.	30
Figura 2.2 - Exemplos de perfis de retardos de multipercursos: (a) Perfil de atraso de multipercurso com um decaimento exponencial, e (b) Perfil de atraso de multipercurso tipo i.i.d.	33
Figura 3.1 - Comparação entre SCM e MCM: (a) espectro de frequências dos sinais transmitidos; e (b) espectro de frequências dos sinais recebidos.	39
Figura 3.2 - Comparação das formas de onda transmitidas: (a) Sistema de transmissão de dados seriais de banda básica, e (b) o sistema OFDM.	43
Figura 3.3 - Comparação dos espectros de frequência: sinais ortogonais não sobrepostos de banda limitada (MCM), (b) sinais ortogonais sobrepostos limitados no tempo (MCM), (c) sinais ortogonais sobrepostos de banda limitada (MCM) e (d) um sinal limitado no tempo (SCM).	44
Figura 3.4 - Sistema OFDM: (a) transmissor e (b) receptor.	46
Figura 3.5 - FFT Básica, Transmissor e Receptor OFDM.	48
Figura 3.6 - Resposta ao impulso instantânea de um canal com desvanecimento seletivo de frequência.	49
Figura 3.7 - Sinais transmitidos: (a) não inserção de nenhum intervalo de guarda, (b) inserção de intervalo de guarda; e (c) inserção de intervalo de guarda com prefixo cíclico.	51
Figura 3.8 - Sinais Recebidos: (a) sem inserção de nenhum intervalo de guarda, (b) inserção de intervalo de guarda; e (c) inserção de intervalo de guarda com prefixo cíclico.	52
Figura 3.9 - Forma atual do OFDM: (a) e (b) técnica de extensão cíclica, (c) espectro de frequência do pulso de onda, e (d) espectro de frequência do sinal OFDM.	54
Figura 3.10 - Espectros de potência de sinais OFDM transmitidos: (a) 16 subportadoras, (b) 64 subportadoras e (c) 256 subportadoras.	56
Figura 3.11 - Estrutura do quadro genérico EUTRAN.	58
Figura 3.12 - Grade de Recursos <i>downlink</i> (estrutura do quadro genérico, prefixo cíclico normal).	59
Figura 4.1 - Modelo de Canal Rádio.	65
Figura 4.2 - Relação entre número de subportadoras, seletividade em frequência e seletividade no tempo.	67
Figura 4.3 - Relação entre o comprimento do intervalo de guarda e seletividade em frequência.	68
Figura 4.4 - Definição da resolução de conversores A/D e D/A.	71
Figura 5.1 - Modelo de um sistema OFDM baseado em piloto.	76
Figura 5.2 - Estimacão do canal no domínio da frequência.	77
Figura 5.3 - Transmissão OFDM no Domínio do Tempo-Frequência (estrutura genérica do quadro, prefixo cíclico normal).	78
Figura 5.4 - Diferentes possibilidades para alocação de pilotos. (a) Alocação de piloto espaçado no tempo e considerando todas as frequências. (b) Alocação de piloto espaçado na frequência e considerando tempo integral. (c) Alocação de piloto espaçado no tempo e espaçado na frequência. (d) Alocação de piloto espaçado no tempo e espaçado na frequência.	82
Figura 5.5 - Alocação de piloto em EUTRAN (estrutura de quadro genérica, prefixo cíclico normal).	83
Figura 5.6 - Exemplo de uma grade piloto retangular.	84
Figura 5.7 - Exemplo de uma grade piloto diagonal.	85

Figura 6.1 - BER vs. Eb/No com a melhor curva de interpolação (Pilotos 5, 3, 2, 4, 6, 7, 61, 101, 76, 51, 151, 16, 26, 11, 13, 31, 21). Ordem decrescente da taxa de erro. Escala logarítmica.	89
Figura 6.2 - BER vs. Eb/No com a melhor curva de interpolação (Pilotos 5, 3, 2, 4, 6, 7, 61, 101, 76, 51, 151, 16, 26, 11, 13, 31, 21). Ordem decrescente da taxa de erro. Escala linear.	89
Figura 6.3 - BER vs. Eb/No com a melhor curva de interpolação (Pilotos 5, 3, 2, 4, 6). Ordem decrescente da taxa de erro. Escala logarítmica.	92
Figura 6.4 - BER vs. Eb/No com a melhor curva de interpolação (Pilotos 5, 3, 2, 4, 6). Ordem decrescente da taxa de erro. Escala linear.	92
Figura 6.5 - BER vs. Eb/No com a melhor curva de interpolação (Piloto 7). Escala logarítmica.	93
Figura 6.6 - BER vs. Eb/No com a melhor curva de interpolação (Piloto 7). Escala linear.	93
Figura 6.7 - BER vs. Eb/No com a melhor curva de interpolação (Pilotos 16, 26, 11, 13, 31, 21). Ordem decrescente da taxa de erro. Escala logarítmica.	94
Figura 6.8 - BER vs. Eb/No com a melhor curva de interpolação (Pilotos 16, 26, 11, 13, 31, 21). Ordem decrescente da taxa de erro. Escala linear.	94
Figura 6.9 - BER vs. Eb/No com a melhor curva de interpolação (Pilotos 61, 101, 76, 51, 151). Ordem decrescente da taxa de erro. Escala logarítmica.	95
Figura 6.10 - BER vs. Eb/No com a melhor curva de interpolação (Pilotos 61, 101, 76, 51, 151). Ordem decrescente da taxa de erro. Escala linear.	95
Figura 6.11 - Vazão vs. Eb/No (Pilotos 11, 13, 16, 21, 26, 31, 51, 61, 76, 7, 101, 6, 4, 151, 2, 3, 5). Ordem Decrescente da Vazão. Escala logarítmica.	98
Figura 6.12 - Vazão vs. Eb/No (Pilotos 11, 13, 16, 21, 26, 31, 51, 61, 76, 7, 101, 6, 4, 151, 2, 3, 5). Ordem Decrescente da Vazão. Escala linear.	98
Figura 6.13 - Vazão vs. Eb/No (Pilotos 11, 13, 16, 21, 26, 31). Ordem Decrescente da Vazão. Escala logarítmica.	101
Figura 6.14 - Vazão vs. Eb/No (Pilotos 11, 13, 16, 21, 26, 31). Ordem Decrescente da Vazão. Escala linear.	101
Figura 6.15 - Vazão vs. Eb/No (Pilotos 51, 61, 76, 7, 101). Ordem Decrescente da Vazão. Escala logarítmica.	102
Figura 6.16 - Vazão vs. Eb/No (Pilotos 51, 61, 76, 7, 101). Ordem Decrescente da Vazão. Escala linear.	102
Figura 6.17 - Vazão vs. Eb/No (Pilotos 6, 4, 151, 2, 3, 5). Ordem Decrescente da Vazão. Escala logarítmica.	103
Figura 6.18 - Vazão vs. Eb/No (Pilotos 6, 4, 151, 2, 3, 5). Ordem Decrescente da Vazão. Escala linear.	103
Figura 6.19 - BER vs. Pilotos, com Eb/No igual a 13 dB (Pilotos 5, 2, 3, 4, 6, 7, 61, 51, 101, 76, 151, 16, 11, 26, 31, 21, 13). Ordem decrescente da taxa de erro. Escala logarítmica.	108
Figura 6.20 - BER vs. Pilotos, com Eb/No igual a 13 dB (Pilotos 5, 2, 3, 4, 6, 7, 61, 51, 101, 76, 151, 16, 11, 26, 31, 21, 13). Ordem decrescente da taxa de erro. Escala linear.	109
Figura 6.21 - BER vs. Pilotos, com Eb/No igual a 13 dB (Pilotos 5, 2, 3, 4, 6, 7). Ordem decrescente da taxa de erro. Escala logarítmica.	110
Figura 6.22 - BER vs. Pilotos, com Eb/No igual a 13 dB (Pilotos 5, 2, 3, 4, 6, 7). Ordem decrescente da taxa de erro. Escala linear.	111
Figura 6.23 - BER vs. Pilotos, com Eb/No igual a 13 dB (Pilotos 16, 11, 26, 31, 21, 13). Ordem decrescente da taxa de erro. Escala logarítmica.	111

Figura 6.24 - BER vs. Pilotos, com E_b/N_0 igual a 13 dB (Pilotos 16, 11, 26, 31, 21, 13). Ordem decrescente da taxa de erro. Escala linear.....	112
Figura 6.25 - BER vs. Pilotos, com E_b/N_0 igual a 13 dB (Pilotos 61, 51, 101, 76, 151). Ordem decrescente da taxa de erro. Escala logarítmica.....	112
Figura 6.26 - BER vs. Pilotos, com E_b/N_0 igual a 13 dB (Pilotos 61, 51, 101, 76, 151). Ordem decrescente da taxa de erro. Escala linear.....	113
Figura 6.27 - Vazão vs. Pilotos, com E_b/N_0 igual a 13 dB (Pilotos 11, 13, 16, 21, 26, 31, 51, 61, 76, 7, 101, 151, 6, 4, 2, 3, 5). Ordem decrescente de Vazão. Escala logarítmica.....	115
Figura 6.28 - Vazão vs. Pilotos, com E_b/N_0 igual a 13 dB (Pilotos 11, 13, 16, 21, 26, 31, 51, 61, 76, 7, 101, 151, 6, 4, 2, 3, 5). Ordem decrescente de Vazão. Escala linear.....	115
Figura 6.29 - Vazão vs. Pilotos, com E_b/N_0 igual a 13 dB (Pilotos 7, 6, 4, 2, 3, 5). Ordem decrescente de Vazão. Escala logarítmica.....	117
Figura 6.30 - Vazão vs. Pilotos, com E_b/N_0 igual a 13 dB (Pilotos 7, 6, 4, 2, 3, 5). Ordem decrescente de Vazão. Escala linear.....	117
Figura 6.31 - Vazão vs. Pilotos, com E_b/N_0 igual a 13 dB (Pilotos 11, 13, 16, 21, 26, 31). Ordem decrescente de Vazão. Escala logarítmica.....	118
Figura 6.32 - Vazão vs. Pilotos, com E_b/N_0 igual a 13 dB (Pilotos 11, 13, 16, 21, 26, 31). Ordem decrescente de Vazão. Escala linear.....	118
Figura 6.33 - Vazão vs. Pilotos, com E_b/N_0 igual a 13 dB (Pilotos 51, 61, 76, 101, 151). Ordem decrescente de Vazão. Escala logarítmica.....	119
Figura 6.34 - Vazão vs. Pilotos, com E_b/N_0 igual a 13 dB (Pilotos 51, 61, 76, 101, 151). Ordem decrescente de Vazão. Escala linear.....	119
Figura 6.35 - BER vs. E_b/N_0 considerando aproveitamento da redundância dos bits de prefixo cíclico (16 Pilotos).....	128

LISTA DE TABELAS

Tabela 1.1 - Técnicas de acesso à interface rádio.	21
Tabela 1.2 - Padrões e tecnologias celulares.	22
Tabela 1.3 - Evolução dos sistemas de comunicações móveis em 3GPP ao longo do tempo.	26
Tabela 1.4 - Especificações da Interface Rádio LTE.	27
Tabela 3.1. Parâmetros do Sistema OFDM em EUTRAN	60
Tabela 6.1 - BER vs. Eb/No com a melhor curva de interpolação (Pilotos 2, 3, 4, 5, 6, 7, 11, 13, 16, 21, 26, 31, 51, 61, 76, 101, 151).	90
Tabela 6.2 - BER vs. Eb/No com a melhor curva de interpolação (Pilotos 5, 3, 2, 4, 6, 7, 61, 101, 76, 51, 151, 16, 26, 11, 13, 31, 21). Ordem decrescente da taxa de erro.	91
Tabela 6.3 - Vazão vs. Eb/No (Pilotos 2, 3, 4, 5, 6, 7, 11, 13, 16, 21, 26, 31, 51, 61, 76, 101, 151).	99
Tabela 6.4 - Vazão vs. Eb/No (Pilotos 11, 13, 16, 21, 26, 31, 51, 61, 76, 7, 101, 6, 4, 151, 2, 3, 5). Ordem Decrescente da Vazão.	100
Tabela 6.5 - BER vs. Eb/No com a melhor curva de interpolação e Vazão vs. Eb/No (Pilotos 2, 3, 4, 5, 6, 7, 11, 13, 16, 21, 26, 31, 51, 61, 76, 101, 151).	106
Tabela 6.6 - BER e Vazão vs. Pilotos, com Eb/No igual a 13dB.	107
Tabela 6.7 - BER vs. Pilotos, com Eb/No igual a 13 dB (Pilotos 2, 3, 4, 5, 6, 7, 11, 13, 16, 21, 26, 31, 51, 61, 76, 101, 151).	109
Tabela 6.8 - BER vs. Pilotos, com Eb/No igual a 13 dB (Pilotos 5, 2, 3, 4, 6, 7, 61, 51, 101, 76, 151, 16, 11, 26, 31, 21, 13). Ordem decrescente da taxa de erro.	110
Tabela 6.9 - Vazão vs. Pilotos, com Eb/No igual a 13 dB (Pilotos 2, 3, 4, 5, 6, 7, 11, 13, 16, 21, 26, 31, 51, 61, 76, 101, 151).	116
Tabela 6.10 - Vazão vs. Pilotos, com Eb/No igual a 13 dB (Pilotos 11, 13, 16, 21, 26, 31, 51, 61, 76, 7, 101, 151, 6, 4, 2, 3, 5). Ordem decrescente de Vazão.	116
Tabela 6.11 - BER e Vazão vs Pilotos.	122
Tabela 6.12 - Vazão Efetiva vs. Eb/No (Pilotos 2, 3, 4, 5, 6, 7, 11, 13, 16, 21, 26, 31, 51, 61, 76, 101, 151).	125
Tabela 6.13 - Vazão Efetiva vs. Eb/No (Pilotos 11, 13, 16, 21, 26, 31, 51, 61, 76, 101, 151, 7, 6, 4, 2, 3, 5). Ordem Decrescente da Vazão.	126
Tabela 6.14 - BER vs. Eb/No com a melhor curva de interpolação e Vazão Efetiva vs. Eb/No (Pilotos 2, 3, 4, 5, 6, 7, 11, 13, 16, 21, 26, 31, 51, 61, 76, 101, 151).	127
Tabela 6.15 - BER vs. Eb/No considerando aproveitamento da redundância dos bits de prefixo cíclico (16 Pilotos).	129

LISTA DE SIGLAS

- ACI (*Adjacent Channel Interference* - Interferência entre Canais Adjacentes)
- ADSL (*Asymmetric Digital Subscriber Line* - Linha Digital Assimétrica de Assinante)
- AWGN (*Additive White Gaussian Noise* - Ruído Gaussiano Branco Aditivo)
- BDMA (*Band Division Multiple Access* - Acesso Múltiplo por Divisão de Banda)
- CDMA (*Code Division Multiple Access* - Acesso Múltiplo por Divisão de Código)
- COFDM (*Code Orthogonal Frequency Division Multiplexing* - Multiplexação Ortogonal Codificada por Divisão de Frequência)
- CRM (Canal Rádio Móvel)
- DAB (*Digital Audio Broadcasting* - Radiodifusão de Áudio Digital)
- DVB (*Digital Video Broadcasting* - Radiodifusão de Vídeo Digital)
- DFT (*Discrete Fourier Transform* - Transformada Discreta de Fourier)
- DS-WCDMA (*Direct Sequence-Wideband CDMA* - CDMA de Banda Larga de Sequência Direta)
- EDGE (*Enhanced Data Rates for Global Evolution* - Melhores Taxas de Dados para Evolução Global).
- EUTRAN (*Evolved UTRAN* - UTRAN Evoluída)
- FDM (*Frequency Division Multiplexing* - Multiplexação por Divisão de Frequência)
- FDMA (*Frequency Division Multiple Access* - Acesso Múltiplo por Divisão de Frequência)
- FDD (*Frequency Division Duplex* - Duplexação por Divisão de Frequência)
- FDP (*Frequency Domain Pilot* - Piloto no Domínio da Frequência)
- FEC (*Forward Error Correction* - Correção de Erro em Repasse)
- FFT (*Fast Fourier Transform* - Transformada Rápida de Fourier)
- GPRS (*General Packet Radio Service* - Serviço de Rádio de Pacote Geral)
- GSM (*Global System for Mobile Communications* - Sistema Global para Comunicações Móveis)
- HDR (*High Data Rate* - Alta Taxa de Dados)
- HSDPA (*High Speed Downlink Packet Access* - Acesso Pacote *Downlink* de Alta Velocidade)
- HSPA (*High Speed Packet Access* - Acesso Pacote de Alta Velocidade)
- HSUPA (*High Speed Uplink Packet Access* - Acesso Pacote *Uplink* de Alta Velocidade)
- IDFT (*Inverse Discrete Fourier Transform* - Transformada Discreta de Fourier Inversa)
- IFDMA (*Interleaved FDMA* - FDMA Intercalado)
- IMT (*International Mobile Telecommunications* - Telecomunicações Móveis Internacional)
- IP (*Internet Protocol* - Protocolo de Internet)
- ISI (*Interference Intersymbol* - Interferência Intersimbólica)
- ITU (*International Telecommunication Union* - União Internacional de Telecomunicações)
- LANs (*Local Area Networks* - Redes Locais)
- LFDMA (*Localized FDMA* - FDMA Localizado)
- LO (*Local Oscillator* - Oscilador Local)
- LTE (*Long Term Evolution* - Evolução à Longo Prazo)
- ME (*Mobile Equipment* - Equipamento Móvel)
- MSE (*Mean Square Error* - Erro Quadrado Médio)
- MCM (*Multicarrier Modulation* - Modulação de Multiportadoras)

- MIMO (*Multiple Input Multiple Output* - Múltiplas Entradas Múltiplas Saídas)
- MMAC (*Multimedia Mobile Access Communications* - Comunicação Multimídia de Acesso Móvel)
- MSC (*Mobile Switching Center* - Central de Comutação de Unidade Móvel)
- OFDM (*Orthogonal Frequency Division Multiplexing* - Multiplexação Ortogonal por Divisão de Frequência)
- PANs (*Personal Area Networks* - Redes Pessoais)
- PAR (*Peak to Average Ratio* - Relação Média de Pico)
- PAPR (*Peak-to-Average Power Ratio* - Relação Média de Pico de Potência)
- PBX (*Private Branch Exchange* - Troca de Ramais Privados)
- PN (*Pseudo Noise* - Pseudo Ruído)
- PSK (*Phase Shift Keying* - Modulação por Chaveamento de Frequência)
- PSTN (*Public Switched Telephone Network* - Rede Telefônica Pública Comutada)
- PTS (*Partial Transmit Sequences* - Sequência de Transmissão Parcial)
- QAM (*Quadrature Amplitude Modulation* - Modulação de Amplitude em Quadratura)
- SC-FDMA (*Single-Carrier Frequency Division Multiple Access* - Acesso Múltiplo por Divisão de Frequência com Portadora Única)
- SCM (*Singlecarrier Modulation* - Modulação com Portadora Única)
- SER (*Symbol Error Rate* - Taxa de Erro de Símbolos)
- SLM (*Selective Mapping* - Mapeamento Seletivo)
- SOFDMA (*Scalable Orthogonal Frequency Division Multiplexing* - Multiplexação Ortogonal Escalável por Divisão de Frequência)
- STDCC (*Swept Time-Delay Cross-Correlation* - Correlação Cruzada por Varredura de Tempo de Retardo)
- SIP (*Session Initiation Protocol* - Protocolo de Iniciação de Sessão)
- SU MIMO (*Single-User Multiple Input/Multiple Output* - Múltiplas Entradas Múltiplas Saídas com Usuário Único)
- TCP (*Transmission Control Protocol* - Protocolo de Controle de Transmissão)
- TDD (*Time Division Duplex* - Duplexação por Divisão no Tempo)
- TDM (*Time Division Multiplexing* - Multiplexação por Divisão no Tempo)
- TDMA (*Time Division Multiple Access* - Acesso Múltiplo por Divisão no Tempo)
- TDP (*Time Domain Pilot* - Piloto no Domínio do Tempo)
- TE (*Terminal Equipment* - Equipamento Terminal)
- TRs (*Technical Requirements* - Requisitos Técnicos)
- TSs (*Technical Specifications* - Especificações Técnicas)
- UE (*User Equipment* - Equipamento do Usuário)
- UMTS (*Universal Mobile Telecommunications Service* - Serviço Universal de Telecomunicação Móvel)
- USIM (*Universal Subscriber Identity Module* - Módulo de Identificação Universal do Assinante)
- UTRA (*Universal Terrestrial Radio Access* - Acesso Rádio Terrestre Universal)
- WCDMA (*Wideband Code Division Multiple Access* - Acesso Múltiplo por Divisão de Código de Banda Larga)
- WLANs (*Wireless Local Area Networks* - Redes Locais sem Fio)
- WSSUS (*Wide-Sense Stationary Uncorrelated Scatering* - Canal estacionário no sentido amplo no domínio do tempo e com espalhamento descorrelacionado no domínio de retardo)
- 3GPP (*3rd Generation Partnership Project*)

1. INTRODUÇÃO

1.1. Sistemas de Comunicações Móveis

Acesso celular à *internet*: Dado que a telefonia celular está presente em muitas áreas no mundo inteiro, uma estratégia natural foi estender as redes celulares de modo que suportassem não somente telefonia de voz, mas também acesso sem fio à *internet*. Idealmente, esse acesso à *Internet* teria uma velocidade alta e proveria mobilidade imperceptível e ininterrupta, permitindo que usuários mantivessem suas sessões TCP (*Transmission Control Protocol* - Protocolo de Controle de Transmissão) enquanto estivessem em trânsito, por exemplo, em um ônibus ou em um trem. Com taxas de bits suficientemente altas entre a fonte e o usuário e vice-versa, o usuário poderia até mesmo manter sessões de videoconferência enquanto estivesse em trânsito.

Uma visão geral da arquitetura celular: O termo celular refere-se ao fato de que uma área geográfica é dividida em várias áreas de cobertura, conhecidas como células. Cada célula contém uma estação-base que transmite sinais para, e recebe sinais de, estações móveis dentro de uma célula. A área de cobertura de uma célula depende de muitos fatores, incluindo potência de transmissão da estação-base, potência de transmissão da estação móvel, obstáculos na célula, como edifícios, e altura das antenas da estação-base. A estação-base não precisa estar posicionada no meio da célula. Hoje, muitos sistemas posicionam estações-base em pontos de interseção de três células, de modo que uma única estação-base com antenas direcionais possa atender a três células.

Quando um usuário (hospedeiro) móvel se desloca para fora da faixa de alcance de uma estação-base e entra na faixa de uma outra, ele muda seu ponto de

conexão com a rede maior (infra-estrutura de rede), isto é, muda a estação-base com a qual está associado, sendo o processo denominado transferência (*handoff*).

Arquitetura básica da rede: Cada estação-base está conectada a uma rede de longa distância, tal como uma Rede Telefônica Pública Comutada (PSTN - *Public Switched Telephone Network*), por meio de uma infra-estrutura cabeada. Especificamente, cada estação-base está conectada a uma Central de Comutação de Unidade Móvel (MSC - *Mobile Switching Center*), que gerencia o estabelecimento e o término de chamadas de e para usuários móveis. Um MSC contém muitas das funcionalidades encontradas em uma central de comutação telefônica normal, como um PBX (*Private Branch Exchange* - Troca de Ramais Privados) ou central telefônica, mas ampliada com a funcionalidade adicional requerida para gerenciar a mobilidade de seus usuários.

Características de enlaces e redes sem fio: Diferenças importantes ocorrem entre um enlace com fio e um enlace sem fio. São elas:

- Redução da intensidade do sinal: Ocorre desvanecimento da intensidade de um sinal à medida que ele se propaga pelo meio sem fio. Radiações eletromagnéticas são atenuadas quando atravessam algum tipo de matéria, por exemplo, um sinal de rádio ao atravessar uma parede. O sinal se dispersará mesmo ao ar livre, resultando na redução de sua intensidade, às vezes denominada atenuação de percurso, à medida que aumenta a distância entre emissor e receptor.
- Interferência de outras fontes: Várias fontes de rádio transmitindo na mesma banda de frequência sofrerão interferência umas das outras. Por exemplo, telefones sem fio em 2,4 GHz e LANs (*Local Area Networks* - Redes Locais) sem fio 802.11b transmitem na mesma banda de frequência. Assim, o usuário de uma LAN sem fio 802.11b que estiver se comunicando por um telefone sem fio em 2,4GHz pode esperar que nem a rede nem o telefone funcionem particularmente bem. Além da interferência de fontes transmissoras, o ruído eletromagnético presente no ambiente, por exemplo, um motor ou um equipamento de microondas próximo, pode resultar em interferência.
- Propagação multivias ou multicaminhos: A propagação multivias ocorre quando a energia da onda eletromagnética se reflete em objetos e no solo e toma caminhos de comprimentos diferentes entre um emissor e um receptor. Isso resulta na dispersão do sinal recebido no destinatário. Objetos que se

movimentam entre o emissor e o receptor podem fazer com que a propagação multivias mude ao longo do tempo.

Essas características sugerem que erros de bits serão mais comuns em enlaces sem fio do que em enlaces com fio.

Taxas de erros de bits mais altas e que variam ao longo do tempo não são as únicas diferenças entre um enlace com fio e um enlace sem fio. Deve-se lembrar que, no caso de enlaces *broadcast* cabeados, todos os nós recebem as transmissões de todos os outros nós. Já no caso de enlaces sem fio, a situação não é tão simples assim.

Supondo que a estação A esteja transmitindo para a estação B e que a estação C também esteja transmitindo para a estação B. O denominado problema do terminal oculto, obstruções físicas presentes no ambiente, por exemplo, uma montanha ou um edifício, pode impedir que A e C escutem as transmissões de um e de outro, mesmo que as transmissões de A e C estejam interferindo no destino B.

Técnicas de acesso à interface rádio: Normalmente ocorrem muitas chamadas simultâneas em uma dada célula. Essas chamadas precisam compartilhar a porção do espectro de rádio alocada à provedora de serviço celular. Hoje, a maioria dos sistemas celulares usa uma de duas abordagens gerais, conforme Tabela 1.1, para compartilhar o espectro de rádio.

Tabela 1.1 - Técnicas de acesso à interface rádio.

Técnicas de acesso à interface rádio
Uma combinação de FDM e TDM
CDMA

Padrões e tecnologias celulares: Quando alguém fala sobre a tecnologia celular, frequentemente classifica essa tecnologia como pertencente a uma de diversas gerações. As primeiras gerações foram projetadas, primordialmente, para tráfego de voz; os sistemas celulares mais recentes suportam acesso à internet, bem como voz. A Tabela 1.2 apresenta os padrões e as tecnologias celulares.

Tabela 1.2 - Padrões e tecnologias celulares.

Geração	Tecnologia
1G	FDMA
2G	TDMA IS-136 (FDM/TDM): evoluiu do FDMA
	GSM (FDM/TDM)
	CDMA IS-95
2,5G	GPRS: evoluiu do GSM
	EDGE: evoluiu do GPRS
	CDMA 2000 - <i>Phase 1</i> : evoluiu do CDMA IS-95
3G	UMTS (WCDMA): evoluiu do GSM
	CDMA 2000 - <i>Phase 2</i> : evoluiu do CDMA 2000 - <i>Phase 1</i>
3,5G	HSDPA e HSUPA: evoluiu do UMTS
4G	OFDMA e SC-FDMA
5G	MIMO-SCM

1.2. Evolução para os Sistemas 4G

Levando-se em conta as tecnologias de LAN sem fio e a gama completa das tecnologias celulares de acesso, é desejável, em matéria de acesso sem fio à internet, que:

- o acesso sem fio à *Internet* seja ubíquo, ou seja, onipresente. Seja em casa, no escritório, em um carro, em um café ou na praia, deseja-se poder acessar a *Internet*.
- o acesso à *Internet* ocorra na velocidade mais alta possível, como uma função da localização física do usuário e da velocidade de movimentação. Por exemplo, se o usuário estiver em uma esquina onde existem acessos disponíveis 802.11b de 11Mbps e 3G de 384kbps, é desejável que o sistema selecione automaticamente o sistema 802.11b, que oferece a taxa de bits mais alta naquele instante e naquele lugar.
- enquanto o usuário passeia por esse ambiente heterogêneo, seja comutado automática e transparentemente de uma tecnologia de acesso para outra (por exemplo, de 802.11 para 3G), dependendo da disponibilidade, sem nenhuma intervenção do usuário.

- manter as conexões TCP em curso, enquanto o usuário se desloca. Além do mais, é desejável que o sistema saiba onde o usuário está, de modo que novas chamadas continuem a alcançá-lo enquanto ele se desloca.
- o sistema suporte novas aplicações, produtos e serviços, tais como, voz e vídeo sobre IP (*Internet Protocol* - Protocolo de *Internet*) em tempo real.

O que realmente falta, agora, é integrar protocolos e tecnologias para transformar esta lista de desejos em realidade. Entre esses componentes estão tecnologias de acesso como 802.11 e 3G, bem como protocolos de gerenciamento de mobilidade, criptografia, protocolos de autenticação e protocolos de redes multimídia, tal como SIP (*Session Initiation Protocol* - Protocolo de Iniciação de Sessão) para voz sobre IP.

Para sistemas 4G haverá uma necessidade de uma nova tecnologia de acesso sem fio. Esta nova tecnologia de acesso sem fio complementarará o sistema 3.5G e os outros sistemas de rádio com os quais existe uma inter-relação. Prevê-se que esta nova interface de rádio irá suportar até cerca de 100Mbps para alta mobilidade e até cerca de 1Gbps para baixa mobilidade. (HARA; PRASAD, 2003, p. 3).

Neste contexto, mobilidade baixa abrange velocidade de pedestres ($\approx 3\text{km/h}$), mobilidade média corresponde à velocidade limitada para carros dentro das cidades ($\approx 50\text{-}60\text{km/h}$), mobilidade alta cobre alta velocidade como em estradas ou com trens rápidos ($\approx 60\text{km/h}$ a 250km/h , ou até mais). O grau de mobilidade é, basicamente, ligado ao tamanho das células em um sistema celular, assim como a capacidade do sistema. Em geral, o tamanho das células em um sistema celular tem que ser maior para um maior grau de mobilidade, a fim de limitar a carga de entrega na rede.

Em conjunto com o desenvolvimento do sistema 4G, haverá uma relação entre o aumento de redes de acesso rádio e sistemas de comunicação, tais como Redes Pessoais (PANs - *Personal Area Networks*), Redes Locais (LANs), *broadcast* digital, e acesso sem fio fixo.

Diferentes e complementares regimes de acesso serão parte dos futuros sistemas. Dispositivos terminais reconfiguráveis e infra-estrutura de rede serão uma parte essencial da futura arquitetura. Tal conceito de redes heterogêneas permite aos operadores de redes seguirem um caminho de migração e evolução das redes atuais para os futuros sistemas, reutilizando investimentos realizados. Novos componentes de acesso podem ser adicionados, onde e quando necessários, em

função das possibilidades econômicas. Isso garante a escalabilidade requerida do sistema.

Novos componentes possíveis de interface rádio fazem parte do conceito. Os diferentes sistemas de acesso usarão as bandas de frequências já atribuídas e identificadas e novas bandas de frequências para os novos elementos, portanto, nenhuma interferência direta entre as diferentes tecnologias é esperada. Todos os sistemas de acesso serão conectados a uma rede IP.

O futuro sistema será composto de tecnologias de acesso já disponíveis. Além disso, estão previstas novas tecnologias de acesso, com altas taxas de transmissão de dados, tanto para o caso com mobilidade reduzida como para o caso com alta mobilidade. A exigência de altas taxas de dados para o celular é um grande desafio do ponto de vista tecnológico e no que diz respeito à disponibilidade de espectro suficiente futuro.

1.3. Técnicas de Multiportadoras para Sistemas 4G

Técnicas de multiportadoras podem combater desvanecimento seletivo em frequência encontrado em comunicações móveis. A robustez contra desvanecimento seletivo em frequência é muito atraente, especialmente para altas velocidades de transmissão de dados.

A técnica OFDM (*Orthogonal Frequency Division Multiplexing* - Multiplexação por Divisão de Frequências Ortogonais) tem sido bem amadurecida, ao longo do seu desenvolvimento, para altas taxas de LANs sem fio e de radiodifusão terrestre de vídeo digital. Ao combinar OFDM com CDMA, pode-se ter efeitos sinérgicos, como o reforço da solidez ou eficiência contra desvanecimento seletivo em frequência, escalabilidade e aumento na taxa possível de transmissão de dados. (HARA; PRASAD, 2003, p. 8).

OFDM tem sido amplamente utilizado nos sistemas de transmissão. Como exemplo, tem-se redes locais sem fio (WLANs - *Wireless Local Area Networks*), como IEEE 802.11a, rádio de alto desempenho LAN tipo dois (HIPERLAN/2), e Comunicação Multimídia de Acesso Móvel (MMAC - *Multimedia Mobile Access Communications*), que proporcionam transmissão de dados até 54 Mbps em 5 GHz.

Elas são, principalmente, destinadas à comunicação entre computadores em um ambiente *indoor*, embora possam dar suporte à transmissão de áudio e vídeo em tempo real, e permitem aos usuários alguma mobilidade.

Alguns sistemas que utilizam OFDM:

- IEEE 802.11a&g *systems* (WLAN - Padrão *Wi-Fi*)
- IEEE 802.16a *systems* (WiMAX)
- ADSL *systems* (*Asymmetric Digital Subscriber Line* - DMT: *Discrete MultiTone*)
- DAB (*Digital Audio Broadcasting*)
- DVB-T (*Digital Video Broadcasting*)
- *Flash* - OFDM *cellular systems*
- *Power Line Communication* (PLC)
- *Optical fiber communications and Radio over Fiber systems* (RoF).

1.4. 3GPP - Evolução a Longo Prazo (LTE - *Long Term Evolution*) e OFDM

O 3GPP (*3rd Generation Partnership Project*) iniciou um projeto, em Dezembro de 2004, para a evolução a longo prazo da tecnologia celular (LTE - *Long Term Evolution*).

O sistema OFDM é um sistema baseado no 3GPP. O 3GPP é a organização responsável pela produção global de especificações técnicas aplicáveis (TSs - *Technical Specifications*) e requisitos técnicos (TRs - *Technical Requirements*) para sistemas de comunicações móveis, baseados em redes com núcleos GSM/UMTS e em tecnologias de acesso rádio que suportam (UTRA - *Universal Terrestrial Radio Access*). O 3GPP iniciou um estudo de viabilidade sobre UTRA e UTRAN - LTE. O objetivo do estudo é “o desenvolvimento de uma estrutura para a evolução da tecnologia de acesso rádio 3GPP para uma alta taxa de dados e baixa latência, e para a tecnologia de acesso rádio com pacotes otimizados”. O estudo assinala um passo na direção da UTRAN Evoluída (EUTRAN - *Evolved UTRAN*), como mostrado na Tabela 1.3.

Tabela 1.3 - Evolução dos sistemas de comunicações móveis em 3GPP ao longo do tempo.

Geração	2G	2.5G	3G	3.5G	LTE
Tecnologia	GSM	EDGE	UMTS	HSPA	EUTRAN

O projeto 3GPP - LTE resultará na liberação da norma 8, padrão UMTS. As modificações e ampliações propostas pelo projeto 3GPP - LTE são voltadas para a melhoria do sistema UMTS para OFDM, baseado no sistema internet de banda larga sem fio, com serviços de voz. O sistema EUTRAN - LTE é projetado para ser compatível e coexistir com as normas sem fio existentes como o GSM + EDGE/UMTS + HSPA.

1.4.1. EUTRAN Interface de Rádio

Em dezembro de 2005, as opções de interface rádio foram reduzidas para o uso do OFDM no *downlink* (transmissão de sinal no sentido da ERB para o sistema móvel) e *Single Carrier - Frequency Division Multiple Access* (SC-FDMA - Acesso Múltiplo por Divisão de Frequências - com uma Única Portadora) no *uplink* (transmissão de sinal no sentido do sistema móvel para a ERB).

O LTE oferece um quadro evolutivo para as redes 3G e 3.5G, e destina-se especificamente a realizar o seguinte (RUMNEY, 2008, p. 1):

- Aumento, no *uplink*, da taxa de pico de dados até 86,4 Mbps em uma banda de 20,0 MHz e com modulação 64QAM (Modulação de Amplitude em Quadratura).
- Aumento, no *downlink*, da taxa de pico de dados até 172,8 Mbps em uma banda de 20,0 MHz e com modulação 64QAM e SU-2x2 MIMO (*single-user multiple input/multiple output*).
- Máxima taxa de pico de dados, no *downlink*, de 326,4 Mbps, usando 4x4 SU-MIMO.
- *Uplink* escalonável com flexibilidade de espectro e, no *downlink*, canais com larguras de banda de 1,4 MHz a 20,0 MHz.
- Maior eficiência espectral, com uma melhoria de 2 a 4 vezes em relação ao lançamento do Release 6 - HSPA (*High Speed Packet Access* - Acesso de Pacotes com Altas Velocidades).

- Latência inferior a 5 ms para pequenos pacotes IP.
- Mobilidade otimizada para celulares de baixa velocidade de 0-15 km/h, móveis com velocidades mais altas, até 120 km/h, serão suportados com alto desempenho, e com sistema operacional para velocidades até 350 km/h.
- Coexistência com sistemas legados, enquanto a evolução ocorrer para toda a Rede IP.

Outras decisões da camada física incluem:

- QPSK, 16QAM e 64QAM como regimes de modulação de dados suportados no *uplink* e *downlink*.
- Uso do regime *Multiple Input Multiple Output* (MIMO - Múltiplas Entradas e Múltiplas Saídas), com a possibilidade de até quatro antenas na parte lateral móvel, e quatro antenas no local da célula.
- Reutilização de codificação de canal (códigos turbo) na especialização da UTRAN.
- Suporte para os regimes de acesso rádio FDD (*Frequency Division Duplex* - Duplexação por Divisão de Frequência) e TDD (*Time Division Duplex* - Duplexação por Divisão no Tempo).

As especificações da interface de rádio para a EUTRAN estão resumidas na Tabela 1.4.

Tabela 1.4 - Especificações da Interface Rádio LTE.

	<i>Downlink</i>	<i>Uplink</i>
Duplex	TDD, FDD	TDD, FDD
Multiplexação	OFDMA	SC-FDMA
Modulação	QPSK, 16QAM, 64QAM	QPSK, 16QAM, 64QAM
Taxa Máxima	100 Mbps	20 Mbps
Espectro	1.25 MHz, 2.5 MHz, 5 MHz, 10 MHz, 15 MHz e 20 MHz	
Mobilidade	Excelente desempenho para 0-15 km/h, desempenho moderado para 50-120 km/h e desempenho operacional para 120-350 km/h	
MIMO	<i>Up to 4x4</i>	

1.5. Conclusão

Neste capítulo foi apresentada uma breve visão geral das tecnologias de acesso celular à Internet, já existentes e emergentes. Foram abordados: uma visão

geral da arquitetura celular, as características de enlaces e redes sem fio, as técnicas de acesso à interface rádio, os padrões e tecnologias celulares. O capítulo finaliza com as técnicas multiportadoras para sistemas 4G, com as características do *uplink* LTE (SC-FDMA) e com as características do *downlink* LTE (OFDM).

2. CARACTERÍSTICAS DOS CANAIS COM DESVANECIMENTO EM AMBIENTES COM MULTIPERCURSOS DE RAIOS

2.1. Introdução

A caracterização da propagação rádio é um pré-requisito para a concepção de sistemas de comunicação sem fio.

Desvanecimento por multipercursos ocorre devido às reflexões em múltiplos percursos, de uma onda transmitida, por espalhadores locais, tais como casas, edifícios, teto de uma unidade móvel (veículo), ou objetos naturais, como florestas. Os sinais são recebidos dos diversos espalhadores, que contribuem com diferentes percursos, através da variável de retardo τ .

A função densidade de probabilidade (p.d.f) do sinal recebido segue, muitas vezes, uma distribuição de *Rayleigh* ou *Rice*.

A Seção 2.2 apresenta a caracterização do fenômeno de multipercurso.

A Seção 2.3 apresenta canais com desvanecimento *Rayleigh* e *Rice*; canais com desvanecimento seletivo em frequência e não seletivo em frequência são discutidos na Seção 2.4. Já a Seção 2.5 descreve sobre canais com desvanecimento seletivo no tempo e não seletivo no tempo.

A Seção 2.6 encerra, trazendo alguns exemplos de canais com desvanecimento por multipercursos.

2.2. Caracterização do Fenômeno de Multipercurso

Em áreas construídas, o maior problema é o espalhamento das ondas pelas superfícies dos edifícios e difração acima e no entorno dos mesmos, já que a antena

móvel se acha bem abaixo das construções vizinhas, não havendo linha de visada ao transmissor.

Devido aos diversos espalhadores encontrados no ambiente entre o móvel e a estação transmissora, réplicas atenuadas do sinal transmitido chegam ao móvel com diferentes retardos, causando a dispersão do sinal no tempo (esta dispersão provoca interferência entre símbolos). Esta é caracterizada por parâmetros como retardo médio, espalhamento de retardo e banda de coerência. Este fenômeno é chamado de desvanecimento seletivo em frequência.

Devido à mobilidade do meio, ou da estação receptora, associada à variação dos ângulos de chegada dos multipercursos, o efeito *Doppler* resulta como a dispersão do sinal na frequência (esta dispersão provoca interferência entre portadoras) , sendo caracterizado pelos parâmetros de deslocamento *Doppler*, espalhamento *Doppler* e tempo de coerência. Este fenômeno é chamado de desvanecimento seletivo no tempo.

Na Figura 2.1 está representado um típico canal com desvanecimento por multipercurso, frequentemente encontrado nas comunicações sem fio, onde há L caminhos.

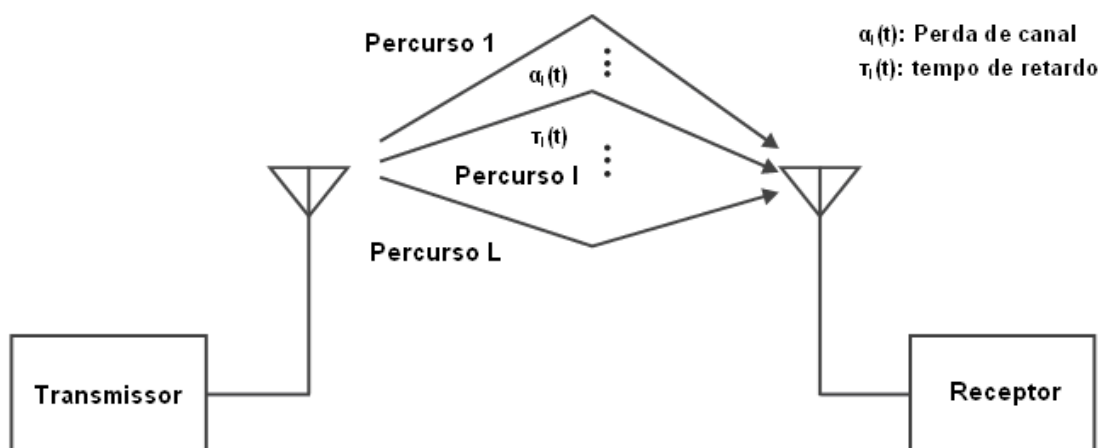


Figura 2.1 - Exemplo de uma transmissão com multipercursos de sinal.

onde:

$\alpha_i(t)$: perda (ou ganho) de canal com valor complexo para o i -ésimo percurso, perda de percurso e

$\tau_i(t)$: tempo de retardo com valor real para o i -ésimo percurso.

A flutuação do sinal recebido com a distância vai acarretar no desvanecimento do sinal.

2.3. Canais com Desvanecimento *Rayleigh* e *Rice*

A distribuição de *Rayleigh* é comumente usada para descrever a natureza estatística da variação no tempo da envoltória de um sinal com desvanecimento não seletivo na frequência (*flat*), ou a envoltória de um componente multipercurso individual. Neste caso, o canal é chamado de “um Canal com Desvanecimento *Rayleigh*”.

Na distribuição de *Rayleigh* são considerados os multipercursos.

Por outro lado, quando um percurso direto está disponível ou o canal tem fortes refletores de sinal, o sinal recebido $r(t)$ não pode ser modelado como um processo de média zero. Neste caso, a envoltória tem uma distribuição de *Rice*, e o canal é chamado de “Canal com Desvanecimento *Rice*”.

Na distribuição de *Rice* são considerados a visada (raio direto) e os multipercursos (raios espalhados).

2.4. Canais com Desvanecimento Seletivo em Frequência e Não Seletivo em Frequência

Quando um sinal é transmitido através de um canal, se a banda de coerência $(\Delta f)_c$ do canal é pequena, comparada com a largura de banda do sinal transmitido, o canal é chamado seletivo em frequência. Neste caso, o sinal é severamente distorcido pelo canal.

Por outro lado, se a banda de coerência $(\Delta f)_c$ é muito maior se comparada à largura de banda do sinal transmitido, o canal é chamado não seletivo em frequência ou *flat*.

Similar à banda de coerência, como medida de seletividade em frequência do canal, há dois parâmetros importantes, o retardo médio (*average excess delay*) e o espalhamento de retardo (*root mean square (RMS) delay spread*).

2.5. Canais com Desvanecimento Seletivo no Tempo e Não Seletivo no Tempo

Quando um sinal é transmitido através de um canal, se o tempo de coerência $(\Delta t)_c$ do canal é pequeno quando comparado com a duração do símbolo do sinal

transmitido, o canal é chamado seletivo no tempo ou rápido (*fast*). O termo rápido é devido à alta mobilidade do meio ou da estação receptora.

Por outro lado, se o tempo de coerência (Δt_c) é muito maior em comparação com a duração do símbolo do sinal transmitido, o canal é chamado não seletivo no tempo ou lento (*slow*). O termo lento é devido à baixa mobilidade do meio ou da estação receptora.

2.6. Exemplos de Canais com Desvanecimento por Multipercursos

Quando um sinal é transmitido através de um canal com desvanecimento por multipercursos, um modelo de canal é capaz de fornecer as características do sinal recebido.

A Figura 2.2 mostra exemplos de perfis de retardo de multipercursos para descrever a seletividade em frequência de um canal. Aqui, pode-se supor que a característica de desvanecimento de cada caminho é independente se o canal considerado for WSSUS (*Wide-Sense Stationary Uncorrelated Scatering* - canal estacionário no sentido amplo no domínio do tempo e com espalhamento descorrelacionado no domínio de retardo), e que muitas vezes é assumida uma distribuição de *Rayleigh* para cada envoltória e uma distribuição uniforme para cada fase. No caso real, essas simplificações equivalem a se tomar curtos intervalos de tempo ou pequenas distâncias, onde se pode considerar as estatísticas do canal praticamente estacionárias.

Considerando-se, inicialmente, a estacionaridade no sentido amplo (WSS), pode-se dizer que a média estatística não depende do instante de tempo ou da localização e que a função autocorrelação é invariável a uma translação de tempo ou de distância, portanto, um processo estocástico é dito estacionário no sentido amplo se sua média for constante e se sua autocorrelação é função, apenas, da diferença entre dois instantes.

Considerando, agora, que os espectros nos canais reais sejam descorrelacionados (US), os ecos associados aos retardos τ_1 e τ_2 , causados por diferentes percursos, apresentam independência na sua amplitude. Assim, assumindo que a perda e a mudança de fase do canal associado com o retardo de percurso τ_1 é descorrelacionado com a perda e a

mudança de fase do canal associado com o retardo de percurso τ_2 , tem-se espalhamento descorrelacionado.

Na Figura 2.2 (a), há um número fixo de percursos com atrasos equidistantes e as potências médias recebidas dos multipercursos são exponencialmente decrescentes. Geralmente esse tipo de perfil é encontrado em ambientes indoor e é chamado de "um perfil com decaimento exponencial". Por outro lado, na Figura 2.2 (b), há também um número fixo de percursos com retardos equidistantes, mas as potências médias recebidas dos multipercursos são todas iguais. Muitas vezes esse tipo de perfil é usado para teste de desempenho do sistema, e é chamado de "perfil independente e identicamente distribuído (i.i.d)".

A variação de um canal no tempo é determinada por uma série de fatores, tais como a altura da antena do emissor/receptor, a velocidade do emissor/ receptor em movimento, a forma da antena, a altura das estruturas circunvizinhas, e assim por diante.

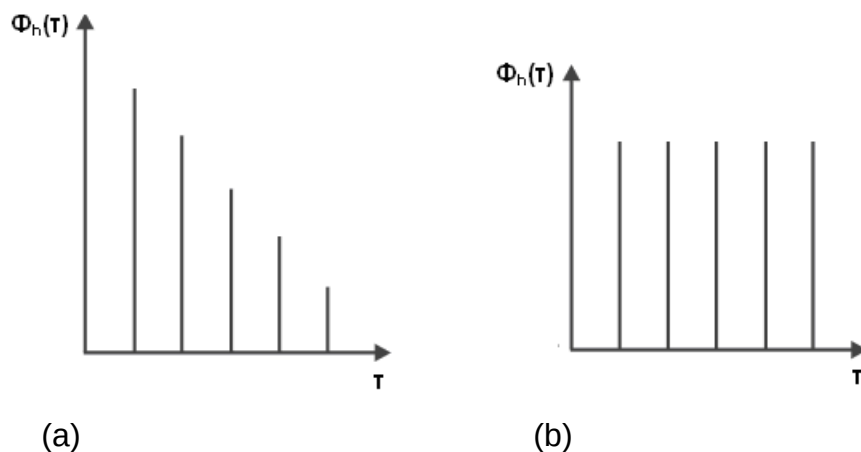


Figura 2.2 - Exemplos de perfis de retardos de multipercursos: (a) Perfil de atraso de multipercurso com um decaimento exponencial, e (b) Perfil de atraso de multipercurso tipo i.i.d.

onde:

τ : Tempo de retardo e

$\Phi_h(\tau)$: Potência relativa em dB.

Os três fatores para descrever as características do desvanecimento que um sinal transmitido experimenta em um canal, como a p.d.f da envoltória, a seletividade em frequência e a seletividade no tempo, são independentes, por isso há muitas combinações a considerar. Por exemplo, quando a componente linha de visada não

está disponível em um canal, a taxa de transmissão de dados é muito alta e o receptor está instalado em um veículo com alta velocidade, o canal será “um canal com desvanecimento *Rayleigh* rápido seletivo em frequência”, ao passo que, quando a componente linha de visada está disponível em um canal, a taxa de transmissão de dados é muito baixa e o receptor está instalado em um terminal fixo, o canal será “um canal com desvanecimento *Rice* lento não seletivo em frequência”.

2.7. Conclusão

O Capítulo 2 mostrou sucintamente as características dos canais de rádio.

Foi visto que a caracterização do canal de propagação rádio móvel permite que se obtenha seus principais parâmetros a partir do perfil de distribuição da potência recebida num extremo receptor.

Foi vista, também, a caracterização do fenômeno de multipercurso: A dispersão do sinal rádio móvel pode ocorrer tanto no tempo quanto na frequência e tem origem, principalmente, no fenômeno de multipercurso do sinal transmitido.

Foram descritos os canais com desvanecimento seletivo no tempo e não seletivo no tempo e canais com desvanecimento seletivo na frequência e não seletivo na frequência.

Todos os conceitos básicos apresentados são importantes para o pleno entendimento dos capítulos seguintes.

3. PRINCÍPIOS E HISTÓRIA DA MODULAÇÃO POR MULTIPORTADORAS (MCM - *MULTICARRIER MODULATION*) E OFDM (*ORTHOGONAL FREQUENCY DIVISION MULTIPLEXING*)

3.1. Introdução

OFDM é uma forma especial de Modulação por Multiportadoras (MCM - *Multicarrier Modulation*), onde um único fluxo de dados é transmitido através de um número de subportadoras, a uma taxa mais baixa. OFDM pode ser visto tanto como uma técnica de modulação ou como uma técnica de multiplexação.

OFDM (*Orthogonal Frequency Division Multiplexing*) é uma forma de modulação do sinal por diferentes subportadoras e OFDMA (*Orthogonal Frequency Division Multiple Access*) é uma forma de agrupar sinais de diferentes fontes, utilizando a tecnologia OFDM.

Uma das principais razões para se usar OFDM é o aumento da robustez (eficiência) contra o desvanecimento seletivo em frequência (tolerância à multipercursos), eficiência contra a interferência em banda estreita e a sua alta eficiência espectral.

O aumento da robustez contra o desvanecimento seletivo em frequência ocorre pois, no OFDM, as taxas de símbolos das subportadoras são baixas, com isso a largura de banda do sinal transmitido é pequena: próxima ou menor que a banda de coerência $(\Delta f)_c$ do canal. Como o período do símbolo é o inverso do espaçamento entre frequências e sendo este espaçamento entre frequências pequeno, o período do símbolo será longo, o que permitirá a inserção de intervalo de guarda, que, como será visto adiante, é eficiente para evitar a interferência entre símbolos.

A eficiência contra a interferência em banda estreita é devido à ortogonalidade entre as subportadoras.

A alta eficiência espectral ocorre já que as formas de onda dos espectros de frequência são amplamente difundidas e sobrepostas (graças à ortogonalidade) e por isso requerem menos largura de banda. Esta alta eficiência espectral é devido, também, à possibilidade das potências das portadoras e do regime de modulação serem controlados individualmente para cada portadora.

O sistema OFDM é muito eficaz em ambientes de multipercursos de raios. Com um único sistema transmissor (uma portadora), desvanecimento ou interferência podem causar a falha de todo o *link* (enlace), mas em um sistema com multiportadoras, apenas uma pequena percentagem de subportadoras será afetada. Um Código de Correção de Erro pode, então, ser usado para corrigir as poucas subportadoras erradas.

Os dois grandes problemas que qualquer canal sem fio sofre são a Interferência entre Portadoras e a Interferência entre Símbolos, descritas a seguir.

- Interferência entre Portadoras (ICI): O movimento do usuário em relação à estação base provoca, no domínio da frequência, um deslocamento *Doppler* e as subportadoras são, então, deslocadas em frequência deixando de ser ortogonais. Devido, no sistema, ser necessária a colocação exata da subportadora, se não houver uma compensação adequada a ser aplicada, este deslocamento *Doppler* causará problemas para o sistema OFDM, resultando em um aumento da taxa de erro de bit (BER).

Para resolver o problema ICI, tem-se diferentes técnicas, das quais uma dessas técnicas é usar um filtro de ajuste para manter a ortogonalidade entre as subportadoras, que foi perdida devido ao efeito *Doppler*.

Interferência entre Portadoras (ICI):

- Causa:
 - Desvanecimento Seletivo no Tempo.
 - Dispersão do Sinal na Frequência.
- Solução:
 - IDFT e DFT.
 - Prefixo Cíclico.
 - $\Delta t > T_c$ e $t_s < T_c$.

onde:

Δt : intervalo entre as retransmissões de mensagens,

T_c : Tempo de coerência do canal e

t_s : Duração do símbolo do sinal transmitido.

- Interferência entre Símbolos (ISI): ISI produz uma forma de distorção de um sinal que faz com que os símbolos transmitidos anteriormente possam ter um efeito sobre o símbolo atualmente recebido.

ISI é geralmente causada por ecos ou devido à resposta de frequência no canal não linear. Em outras palavras, ISI ocorre quando um sinal é transmitido através de múltiplos caminhos de propagação de diferentes comprimentos.

O sinal recebido é espalhado no tempo devido à variação do retardo do sinal nos múltiplos caminhos. No caso do espalhamento do retardo (σ_T) ser maior do que a duração do símbolo adjacente (t_s), símbolos são misturados e o receptor não é capaz de avaliar corretamente o símbolo, um intervalo de guarda é, então, utilizado para compensar a distorção do canal.

$\sigma_T > t_s \rightarrow$ símbolos são misturados e o receptor não é capaz de avaliar corretamente o símbolo.

Para que se obtenha um sinal sem distorções, é inserido um intervalo (banda) de guarda. Em razão da inserção deste intervalo, ocorre uma diminuição da eficiência espectral, já que nesta banda de guarda não há transmissão de símbolo útil, ou seja, não há transmissão de dados.

Para resolver o problema da ISI é preciso um equalizador de canal complexo no receptor. Uma solução mais simples é usar a modulação multiportadora.

A idéia principal é dividir a banda disponível do canal em um número de subcanais com larguras de banda iguais, onde a largura de banda de cada subcanal é suficientemente estreita para que as características de resposta da frequência dos subcanais sejam quase ideais.

O objetivo é ter $BW < B_c$, para que se tenha uma resposta plana ou em outras palavras $T_b > T_m$ (a fim de evitar a ISI).

$$BW < B_c \text{ ou } t_s > \sigma_T \rightarrow \text{evita a ISI.}$$

onde:

BW : largura de banda de cada subcanal (do sinal transmitido),

B_c ou $(\Delta f)_c$ ou BW_c : Banda de Coerência (do canal),

t_s ou T_b : duração do símbolo útil, onde a palavra útil significa que carrega os dados transmitidos e

T_m ou σ_T : espalhamento de retardo do canal multipercurso.

Interferência entre Símbolos:

- Causa:
 - Desvanecimento Seletivo em Frequência.
 - Dispersão do Sinal no Tempo.
- Solução:
 - Modulação Multiportadoras.
 - Intervalo de Guarda e Prefixo Cíclico.
 - $\Delta_G > T_{\max}$ e $\Delta_G < t_s$.
 - $BW < B_c$.
 - $t_s > \sigma_T$.
 - $\Delta f > BW_c$.

onde:

Δ_G : Intervalo de guarda e

$T_{\max}$: Máximo espalhamento de retardo.

Este capítulo aborda os princípios do OFDM e sua história desde a sua origem (Seção 3.2) até a sua forma atual. São apresentadas a utilização da Transformada Discreta de Fourier (DFT - *Discrete Fourier Transform*) (Seção 3.3), a inserção de intervalo de guarda e a inserção de prefixo cíclico, para a forma atual do OFDM (Seção 3.4). A Seção 3.5 finaliza, mostrando os parâmetros do sistema OFDM em Interface Aérea LTE (*Long Term Evolution*).

3.2. Origem do OFDM

MCM é o princípio de transmissão de dados através da divisão do fluxo de entrada em diversos fluxos de símbolos, cada qual com uma taxa de símbolos muito mais baixa, e usando esses subfluxos para modular várias subportadoras.

A Figura 3.1 compara uma Modulação com Portadora Única (SCM - *Singlecarrier Modulation*) e um MCM.

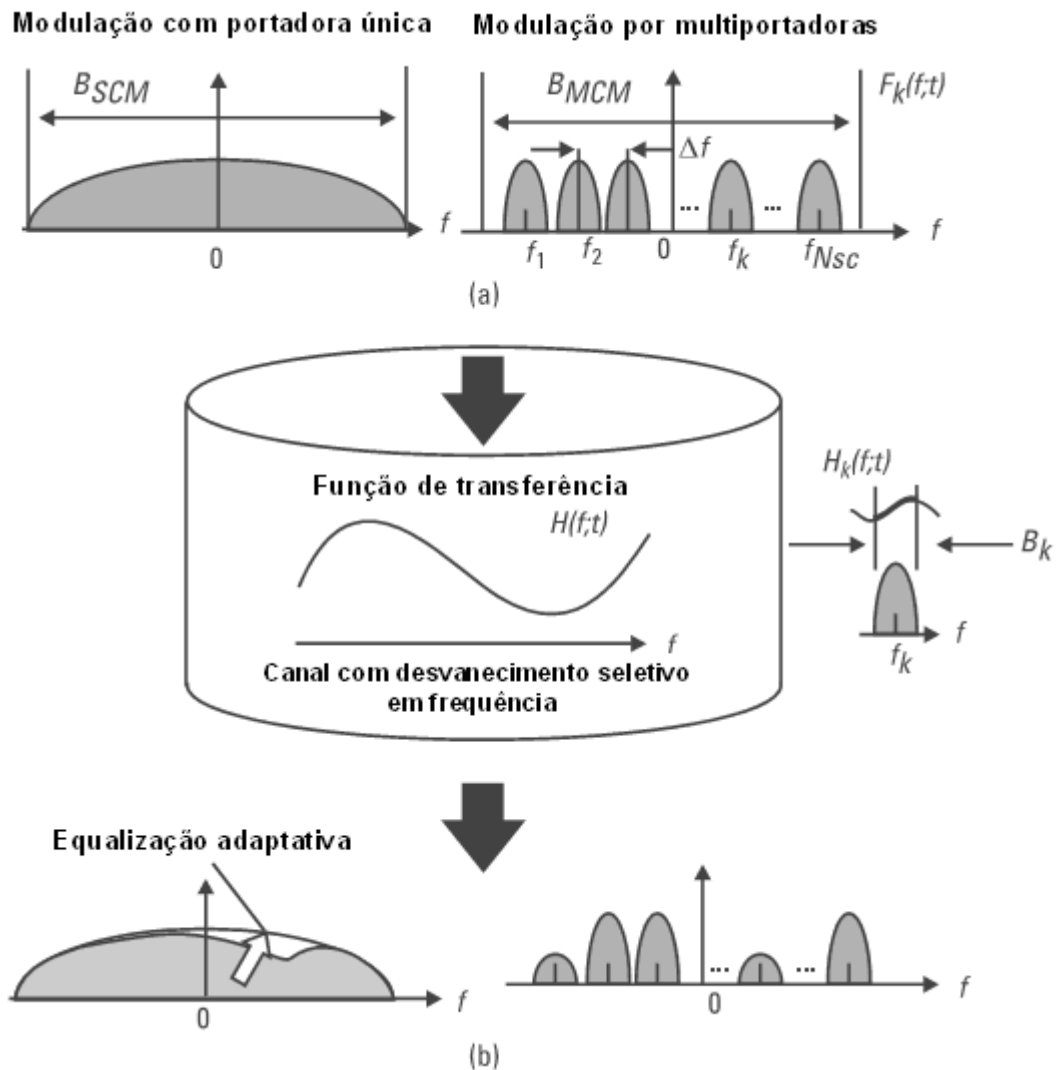


Figura 3.1 - Comparação entre SCM e MCM: (a) espectro de frequências dos sinais transmitidos; e (b) espectro de frequências dos sinais recebidos.

onde:

B_{SCM} : largura de banda de transmissão do sinal SCM,

B_{MCM} : largura de banda de transmissão do sinal MCM,

Δf : espaçamento entre as subportadoras,

f_k : frequência da k -ésima subportadora,

$H(f; t)$: função de transferência do canal,

$H_k(f; t)$: função de transferência do canal correspondente a B_k ,

B_k : banda de frequência ocupada pela k -ésima subportadora,

N_{SC} : número total de subportadoras e

$F_k(f; t)$: espectro de frequência do pulso de onda da k -ésima subportadora.

O espectro de frequência do sinal MCM ($S_{MCM}(f; t)$) é escrito como:

(3.1)

$$S_{MCM}(f; t) = \sum_{k=1}^{N_{SC}} F_k(f; t)$$

onde:

$S_{MCM}(f; t)$: espectro de frequência do sinal MCM.

Através de um canal com desvanecimento seletivo em frequência caracterizado pela função de transferência $H(f; t)$, os espectros de frequência recebidos dos sinais SCM e MCM são escritos respectivamente, como:

(3.2)

$$\begin{aligned} R_{SCM}(f; t) &= H(f; t) S_{SCM}(f; t) \\ R_{MCM}(f; t) &= H(f; t) S_{MCM}(f; t) \end{aligned}$$

onde:

$R_{SCM}(f; t)$: Espectro de frequência recebido do sinal SCM e

$R_{MCM}(f; t)$: Espectro de frequência recebido do sinal MCM.

(3.3)

$$R_{MCM}(f; t) = \sum_{k=1}^{N_{SC}} H_k(f; t) F_k(f; t)$$

onde $S_{SCM}(f; t)$ é o espectro de frequência do sinal SCM transmitido e $H_k(f; t)$ é a função de transferência do canal correspondente a B_k , que é a banda de frequência ocupada pela k -ésima subportadora.

Quando o número de subportadoras é grande, as respostas de amplitude e fase de $H_k(f; t)$ podem ser assumidas como constantes ao longo de B_k , assim $R_{MCM}(f, t)$ pode ser aproximado como:

(3.4)

$$R_{MCM}(f; t) \cong \sum_{k=1}^{N_{SC}} H_k(t) F_k(f; t)$$

onde:

$H_k(t)$: perda tolerável para B_k , com valor complexo.

A equação (3.4) demonstra que a MCM é eficaz e robusta em canais sem fio; ou seja, para combater o desvanecimento seletivo em frequência. MCM não exige nenhuma compensação ou, no máximo, um toque de equalização para cada subportadora; considerando SCM, ela requer uma complicada equalização adaptativa.

Os primeiros sistemas empregando MCM eram militares, ligações via rádio em HF no final dos anos 1950 e início dos anos 1960, como KINEPLEX e *Kathryn*, onde sinais ortogonais de banda limitada foram utilizados por causa da dificuldade no controle preciso da frequência de subportadora dos osciladores locais e da dificuldade na detecção de sinais de portadoras com filtros analógicos. Definindo a duração (ou período) de símbolo (de nível) da subportadora como T_s , a transmissão do sinal $s(t)$ é escrita como (HARA; PRASAD, 2003, p. 29):

(3.5)

$$s(t) = \sum_{i=-\infty}^{+\infty} \sum_{k=1}^{N_{SC}} c_{ki} e^{j2\pi f_k(t-iT_s)} f(t-iT_s)$$

onde:

$s(t)$: sinal transmitido,

T_s : duração de símbolo (de nível) da subportadora,

c_{ki} : informação do i -ésimo símbolo e da k -ésima subportadora e

$f(t)$: forma do pulso de onda do símbolo.

Quando o pulso de onda retangular é usado, $f(t)$ é dada por:

(3.6)

$$f(t) = \begin{cases} 1, & (0 < t \leq T_s) \\ 0, & (t \leq 0, t > T_s) \end{cases}$$

onde:

$f(t)$: forma do pulso de onda retangular (sinal limitado no tempo).

As variáveis f_k e Δf são, respectivamente, escritas como:

(3.7)

$$f_k = \frac{(k-1)}{T_s}, \Delta f = \frac{1}{T_s}$$

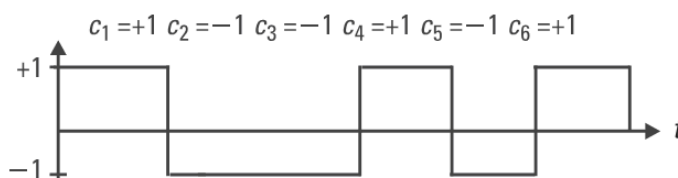
A Figura 3.2 compara um sistema de transmissão de banda base de dados seriais com um sistema OFDM. Já a Figura 3.3 compara os espectros de frequência. A MCM clássica, empregando sinais ortogonais de banda limitada, corresponde ao uso de filtros e osciladores de subportadoras analógicas, mas requer muito mais largura de banda [ver Figura 3.3 (a)]. Empregando o pulso de onda retangular para as subportadoras, as formas de onda dos espectros de frequência são amplamente difundidas e sobrepostas [ver Figura 3.3 (b)], mas requerem menos largura de banda.

A Idéia de utilizar o OFDM já foi abandonada, temporariamente, devido à dificuldade na recuperação da subportadora, sem interferência entre as subportadoras, por meio de filtros analógicos. Como resultado, uma série de estudos, na década de 1960, foi dedicada para MCM empregando sinais ortogonais sobrepostos de banda limitada [ver Figura 3.3 (c)]. Isso ocorre porque os filtros analógicos podem facilmente separar esses sinais. Observa-se que o nome OFDM apareceu nos EUA com a Patente N°. 3, emitida em 1970.

Uma das dificuldades era devido à enorme potência de processamento requerida para realizar as operações FFT necessárias. No entanto, o contínuo avanço da tecnologia de processamento de sinal fez com que esta não fosse mais uma razão para evitar o OFDM, e que agora constitui a base do *downlink* LTE.

A outra razão do OFDM ter sido, até então, evitado em sistemas móveis foi a alta taxa média de pico (PAR - *Peak to Average Ratio*) de sinais que ele cria, devido à transmissão paralela de muitas centenas de subportadoras estreitamente espaçadas. Para dispositivos móveis, esta alta PAR é problemática tanto para o projeto do amplificador de potência como para o consumo de bateria, e é essa preocupação que levou o 3GPP a desenvolver um novo esquema de transmissão para o uplink: o SC-FDMA. Ficando o esquema de transmissão no *downlink* baseado em OFDM e no *uplink* baseado no SC-FDMA.

A Figura 3.3 também contém o espectro de frequência do sinal limitado no tempo [ver Figura 3.3 (d)]. As Figuras 3.3 (b) e (d) correspondem às Figuras 3.2 (b) e (a), respectivamente.



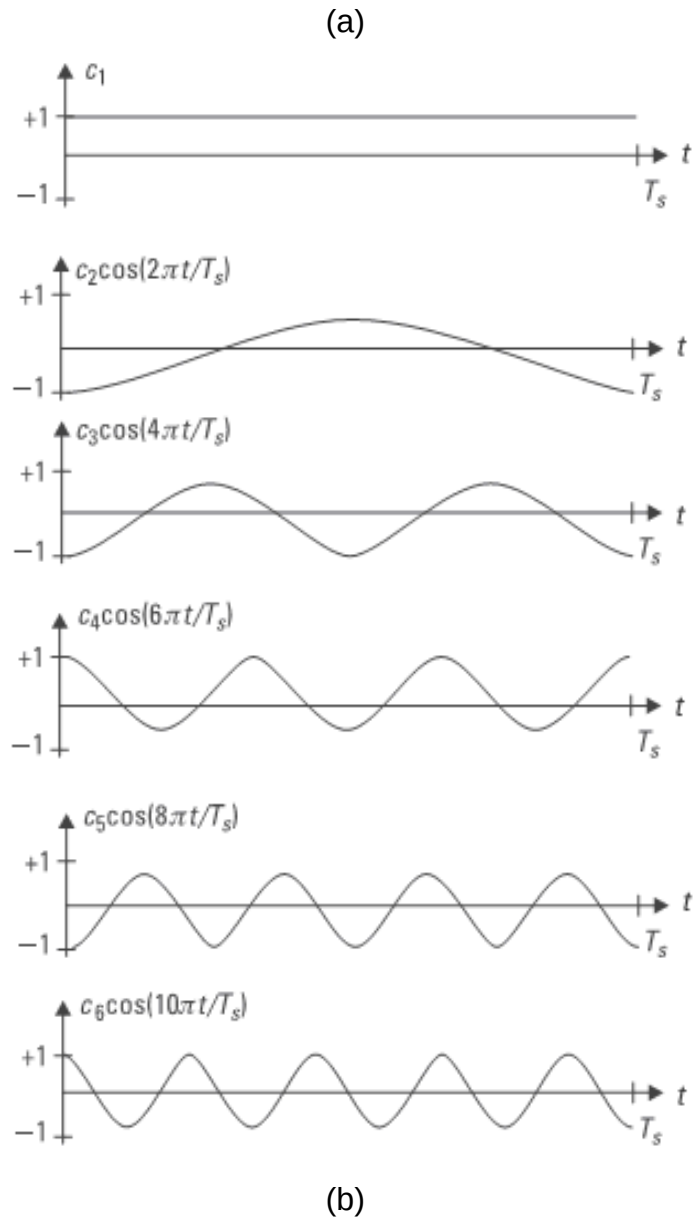
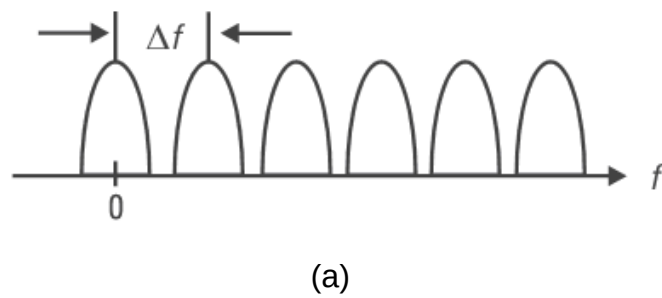


Figura 3.2 - Comparação das formas de onda transmitidas: (a) Sistema de transmissão de dados seriais de banda básica, e (b) o sistema OFDM.



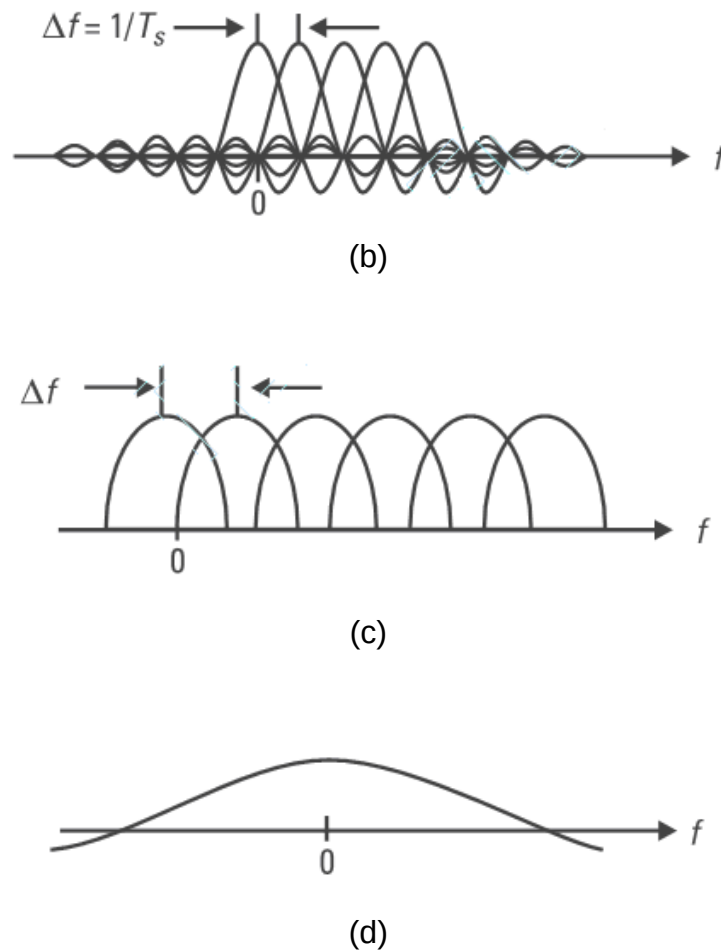


Figura 3.3 - Comparação dos espectros de frequência: sinais ortogonais não sobrepostos de banda limitada (MCM), (b) sinais ortogonais sobrepostos limitados no tempo (MCM), (c) sinais ortogonais sobrepostos de banda limitada (MCM) e (d) um sinal limitado no tempo (SCM).

3.3. Uso da Transformada Discreta de Fourier

A idéia de usar a transformada discreta de Fourier (DFT) retomou a MCM empregando sinais ortogonais limitados no tempo, ou seja, OFDM.

Seja a amostra do sinal transmitido $s(t)$, no intervalo $iT_s < t \leq (i + 1)T_s$ (onde T_s é a duração de símbolo da subportadora), com taxa de amostragem de $t_{spl} (=T_s / N_{sc})$, o sinal transmitido é escrito (na forma de um vetor coluna) como (HARA; PRASAD, 2003, p. 30):

(3.8)

$$\begin{aligned}\mathbf{S} &= [s(iT_s + t_{spl}), \dots, s(iT_s + qt_{spl}), \dots, s(iT_s + N_{SC} t_{spl})]^T \\ &= \mathbf{W}^{-1}(N_{SC}) \mathbf{C}_i\end{aligned}$$

onde T denota a transposta de $*$ (o sobrescrito T denota a matriz transposta) e $\mathbf{W}^{-1}(N_{SC})$ é dado por:

(3.9)

$$\begin{aligned}\mathbf{W}^{-1}(N_{SC}) &= \{w_{qk}^{-1}\} \\ w_{qk}^{-1} &= e^{j2\pi \frac{q(k-1)}{N_{SC}}}\end{aligned}$$

onde:

$\mathbf{W}^{-1}(N_{SC})$: N_{SC} - pontos da matriz inversa DFT (IDFT).

(3.10)

$$\mathbf{C}_i = [c_{1i}, c_{2i}, \dots, c_{N_{SC}i}]^T$$

onde:

C_i : i-ésimo símbolo do vetor coluna.

A equação (3.8) mostra, claramente, que o vetor de símbolos transmitidos é recuperado no receptor através da DFT:

(3.11)

$$\mathbf{C}_i = \mathbf{W}(N_{SC}) \mathbf{S}$$

onde:

$\mathbf{W}(N_{SC})$: N_{SC} Ponto da matriz DFT.

$\mathbf{W}(N_{SC})$ é dado por:

(3.12)

$$\begin{aligned}\mathbf{W}(N_{SC}) &= \{w_{qk}\} \\ w_{qk} &= e^{-j2\pi \frac{q(k-1)}{N_{SC}}}\end{aligned}$$

Matriz DFT:

A Figura 3.4 mostra o sistema OFDM empregando a IDFT (*Inverse Discrete Fourier Transform* - Transformada Discreta de Fourier Inversa) e DFT. Ao empregar DFT retangular na janela do receptor, a interferência entre subportadoras pode, perfeitamente, ser eliminada.

O uso de IDFT / DFT elimina, totalmente, a necessidade do banco de osciladores para as subportadoras no transmissor e no receptor e, além disso, se o número de subportadoras escolhido for potência de dois, pode-se substituir a DFT pela Transformada Rápida de Fourier (FFT - *Fast Fourier Transform*).

O Sistema OFDM transmite os dados como um conjunto de portadoras paralelas, com baixa largura de banda (100Hz - 50kHz). O espaçamento de frequência entre as portadoras é o inverso do período do símbolo útil. As portadoras resultantes são ortogonais. Elas são independentes entre si, mesmo que seus espectros se sobreponham.

Δf : espaçamento mínimo entre duas frequências a serem empregadas num sistema com diversidade em frequência.

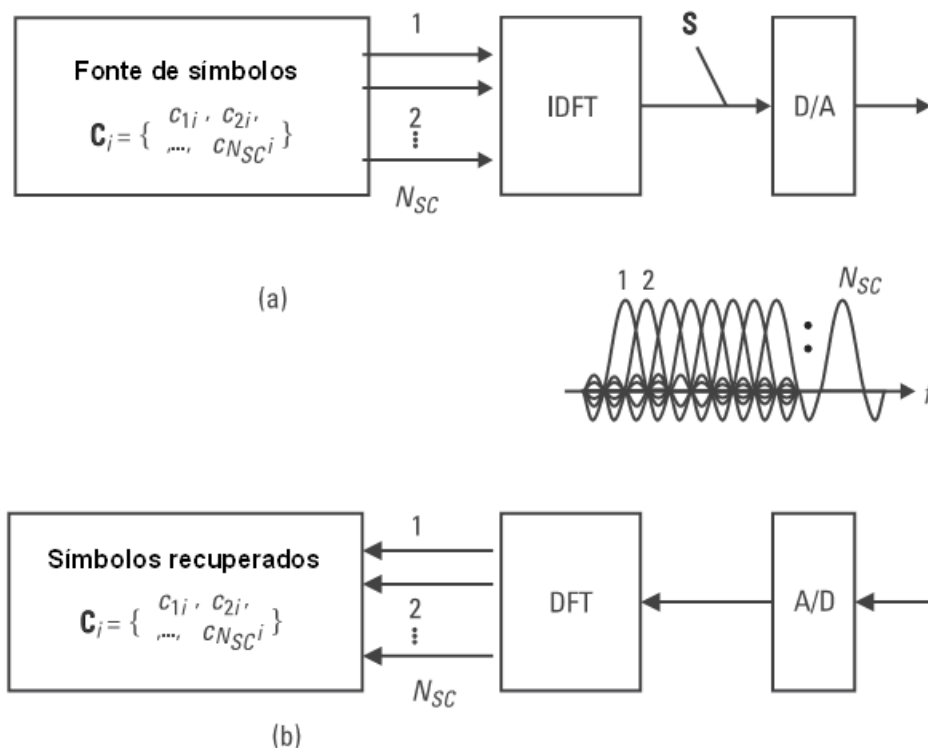


Figura 3.4 - Sistema OFDM: (a) transmissor e (b) receptor.

Geração do OFDM:

O Sistema OFDM pode ser facilmente gerado utilizando uma IFFT (*Inverse Fast Fourier Transform* - Transformada Inversa Rápida de Fourier) e recebido através de uma FFT (*Fast Fourier Transform* - Transformada Rápida de Fourier). Sistemas de alta taxa de dados são obtidos através de um grande número de portadoras, como exemplo, tem-se o DVB que trabalha entre 2000 e 8000 portadoras.

Como as potências das portadoras e o regime de modulação podem ser controlados individualmente, para cada portadora, o sistema OFDM permite uma alta eficiência espectral. No entanto, em sistemas *broadcast* eles são fixos.

Na maioria dos sistemas de comunicações, são necessárias comunicações de dois percursos e vários usuários são suportados. O sistema OFDM pode ser aplicado em um serviço com multiusuários produzindo flexibilidade e um sistema de comunicação eficiente. O projeto do sistema OFDM de um sistema multiusuário depende da aplicação pretendida e da complexidade do *hardware*.

Para gerar um sistema OFDM com sucesso, a relação entre todas as portadoras deve ser cuidadosamente controlada para manter a ortogonalidade entre elas. Por esta razão, o sistema OFDM é gerado escolhendo, primeiramente, o espectro necessário, com base nos dados de entrada e no regime de modulação utilizado. Em cada portadora a ser produzida, são atribuídos alguns dados para transmissão. A amplitude necessária e a fase da portadora são calculadas com base no regime de modulação (tipicamente BPSK, QPSK ou QAM). O espectro necessário é, então, convertido de volta para o seu sinal no domínio do tempo utilizando *Inverse Fourier Transform* (Transformada Inversa de Fourier). Na maioria das aplicações, uma IFFT é usada. A IFFT realiza a transformação de forma muito eficiente, e fornece uma maneira simples de assegurar que as portadoras dos sinais produzidos sejam ortogonais.

A FFT transforma um sinal cíclico no domínio do tempo em seu espectro de frequência equivalente. Isso é feito encontrando a forma de onda equivalente, gerada por uma soma de componentes senoidais ortogonais. A amplitude e a fase das componentes senoidais representam o espectro de frequência do sinal no domínio do tempo.

A IFFT executa o processo inverso, transformando um espectro (de amplitude e fase de cada componente) em um sinal no domínio do tempo. Uma IFFT converte um número de pontos complexos de dados, de comprimento que é uma potência de 2, para o sinal no domínio do tempo, com o mesmo número de pontos.

Ortogonalidade: as portadoras ortogonais necessárias para o sinal OFDM podem ser facilmente geradas pela configuração de amplitude e fase de cada frequência, em seguida, executa-se a IFFT. Uma vez que cada ponto de dados no espectro de frequência de uma IFFT corresponde à amplitude e fase de um conjunto de senóides ortogonais, o processo reverso garante que as portadoras geradas sejam ortogonais.

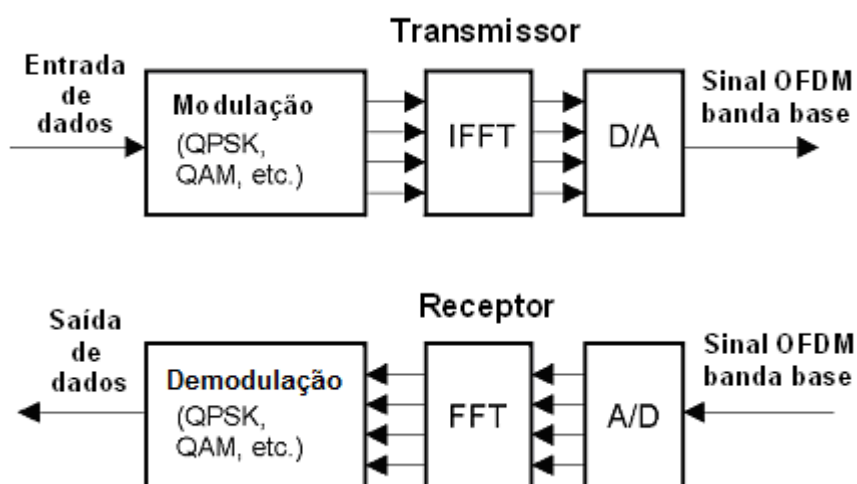


Figura 3.5 - FFT Básica, Transmissor e Receptor OFDM.

A Figura 3.5 mostra a configuração básica de um transmissor e um receptor OFDM. O sinal é gerado em banda base e, assim, para gerar um sinal de RF, o sinal deve ser filtrado e misturado com a frequência de transmissão desejada.

3.4. Inserção de prefixo cíclico para a atual forma de OFDM

Agora, será considerada uma distorção que um canal com desvanecimento seletivo em frequência provoca em um sinal OFDM (dispersão do sinal no tempo $\rightarrow$ ISI). Este canal pode ser caracterizado por uma resposta ao impulso com espalhamento de retardo no domínio do tempo, que não é insignificante quando comparado com um período de símbolo ($\sigma_T > t_S \rightarrow$ ISI).

A Figura 3.6 mostra uma resposta ao impulso instantânea de um canal com desvanecimento seletivo em frequência, onde se pode ver dois percursos e $T_{\max}$ denota o tempo de retardo entre o primeiro e o segundo percursos.

Através do canal, o primeiro percurso gera o sinal desejado e o segundo percurso, o sinal atrasado no receptor. A Figura 3.7 (a-c) mostra três sinais transmitidos e a Figura 3.8 (a-c) mostra os três sinais recebidos correspondentes.

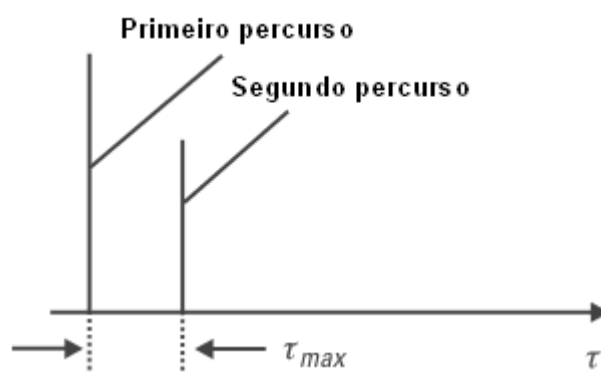


Figura 3.6 - Resposta ao impulso instantânea de um canal com desvanecimento seletivo de frequência.

onde:

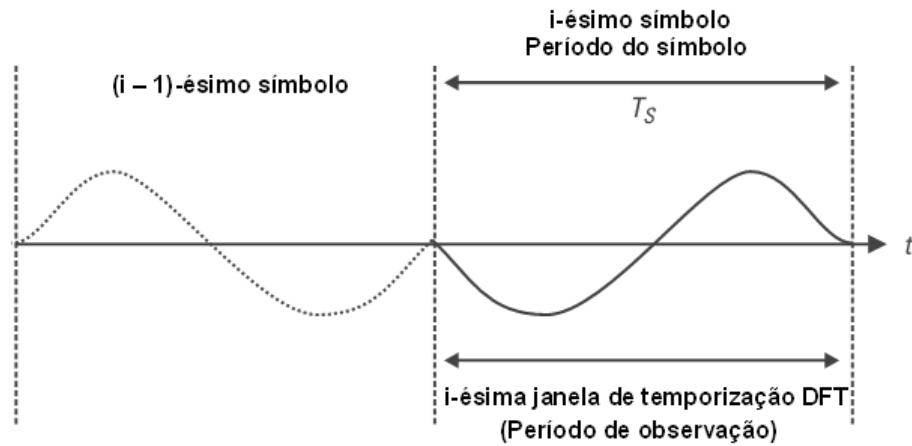
$T_{\max}$: tempo de retardo entre o primeiro e o segundo percursos (máximo espalhamento de retardo).

Considerando uma subportadora, verifica-se que, sem um intervalo de guarda (Δ_G) entre sucessivos símbolos OFDM, ocorrem interferências entre símbolos (ISI - *Interference Intersymbol*): distorção do $(i - 1)$ -ésimo símbolo no i -ésimo símbolo [comparar Figura 3.7 (a) com a Figura 3.8 (a) e ver a linha grossa na Figura 3.8 (a)].

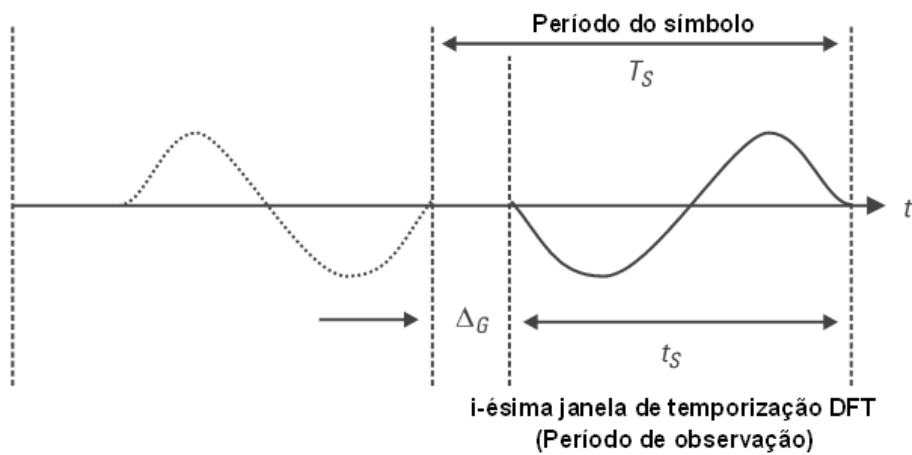
Se empregar um intervalo de guarda (sem transmissão de sinal) com comprimento de $\Delta_G > T_{\max}$, pode-se perfeitamente eliminar a ISI, mas uma mudança repentina na forma de onda, que contém mais componentes espectrais, resulta em interferência entre subportadoras (ICI) [comparar Figura 3.7 (b) com a Figura 3.8 (b) e ver a linha mais espessa na Figura 3.8 (b)].

A Figura 3.7 (c) mostra que a técnica de inserção de intervalos de guarda com prefixo cíclico pode perfeitamente eliminar a interferência entre subportadoras, onde o símbolo OFDM é ciclicamente estendido no período de guarda.

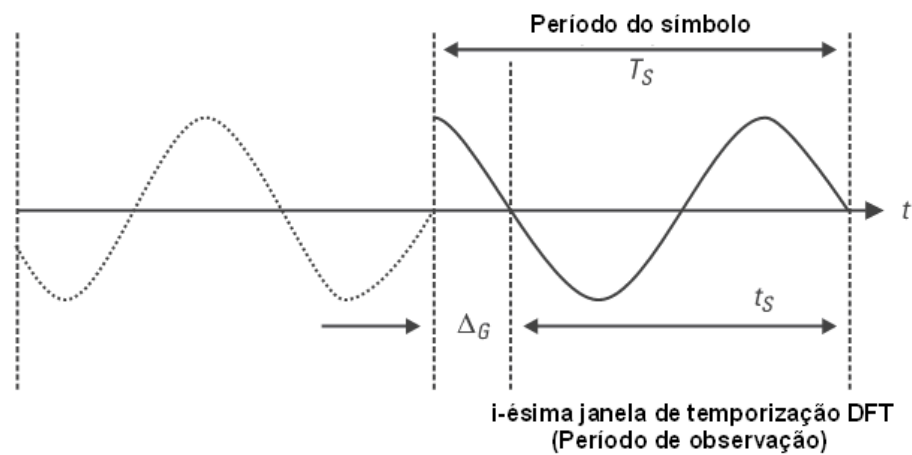
Na Figura 3.8 (c), no período de observação (i -ésima janela de tempo DFT) com largura t_s , pode-se ver dois sinais senoidais com largura total, sem resultar em nenhuma interferência entre subportadoras.



(a)

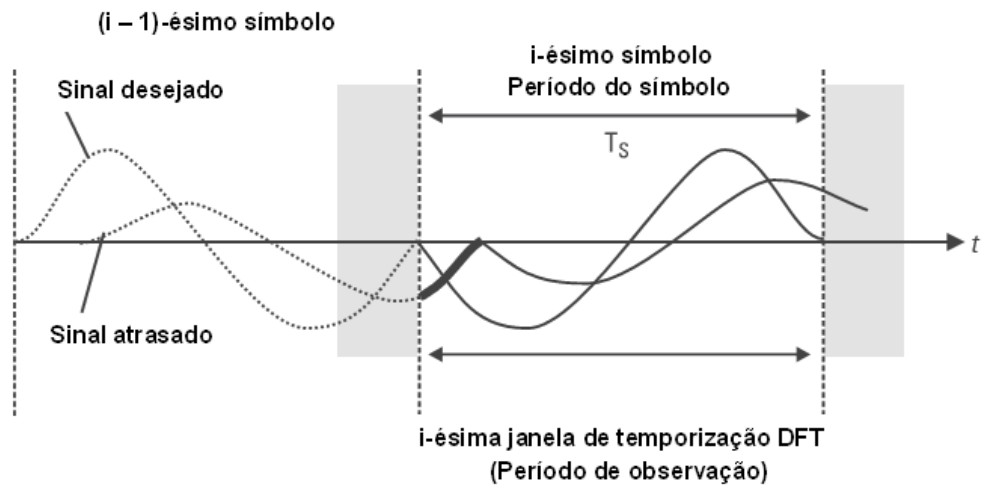


(b)

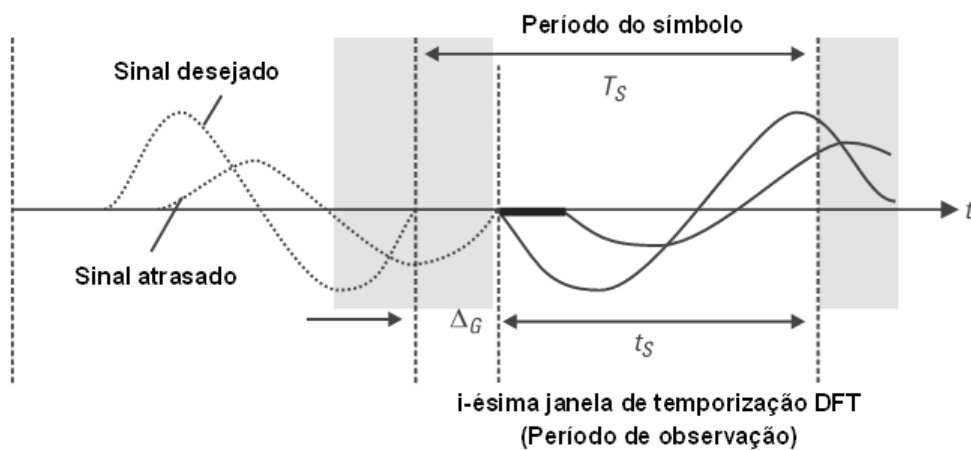


(c)

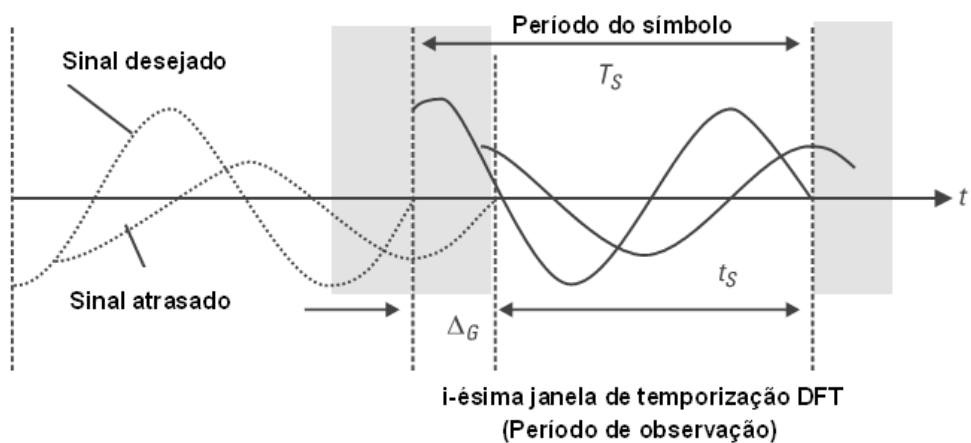
Figura 3.7 - Sinais transmitidos: (a) não inserção de nenhum intervalo de guarda, (b) inserção de intervalo de guarda; e (c) inserção de intervalo de guarda com prefixo cíclico.



(a)



(b)



(c)

Figura 3.8 - Sinais Recebidos: (a) sem inserção de nenhum intervalo de guarda, (b) inserção de intervalo de guarda; e (c) inserção de intervalo de guarda com prefixo cíclico.

Observa-se que o período do símbolo OFDM T_s é longo, mas a frequência da subportadora é um múltiplo inteiro de $1/t_s$. Isso implica que a separação entre subportadoras agora se torna um pouco maior, ou seja, $1/t_s$ ($\Delta f > BW_c$).

Modificando (3.5), (3.6) e (3.7), o sinal transmitido com a extensão cíclica é finalmente escrito como

$$s(t) = \sum_{i=-\infty}^{+\infty} \sum_{k=1}^{N_{SC}} c_{ki} e^{j2\pi f_k(t-iT_s)} f(t-iT_s) \quad (3.13)$$

onde: $s(t)$: sinal transmitido com a extensão cíclica

$$f(t) = \begin{cases} 1, & (-\Delta_G < t \leq t_s) \\ 0, & (t \leq -\Delta_G, t > t_s) \end{cases} \quad (3.14)$$

(3.15)

$$f_k = \frac{(k-1)}{t_s}, \Delta f = \frac{1}{t_s}$$

onde:

T_s : período do símbolo OFDM,

Δ_G ou T_g : comprimento do intervalo de guarda (prefixo cíclico do intervalo de guarda, que evita a interferência entre símbolos) e

t_s ou T_b : período de observação (comprimento útil do símbolo ou tempo útil de símbolo contendo a transmissão de dados).

onde T_s , Δ_G e t_s satisfazem a seguinte equação:

(3.16)

$$T_s = \Delta_G + t_s \quad \text{ou}$$

$$T_s = T_g + T_b$$

Existem várias aplicações do OFDM em sistemas de rádio comercial e em comunicações *broadcasting* (radiodifusão). A forma de onda OFDM de todos os sistemas é matematicamente expressa por (3.13) até (3.16) no sentido de que ela é transmitida e recebida com IDFT e DFT e que tem um intervalo de guarda. Esta é a forma atual do OFDM.

A Figura 3.9 mostra a técnica da extensão cíclica, o espectro de frequência do pulso de onda e o espectro de frequência do sinal transmitido sob a forma atual de OFDM.

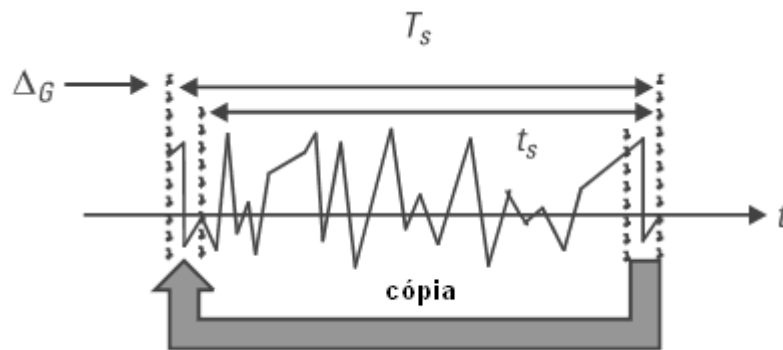
Para o sinal OFDM, a taxa de transmissão total de símbolo é dada por (HARA; PRASAD, 2003, p. 37):

(3.17)

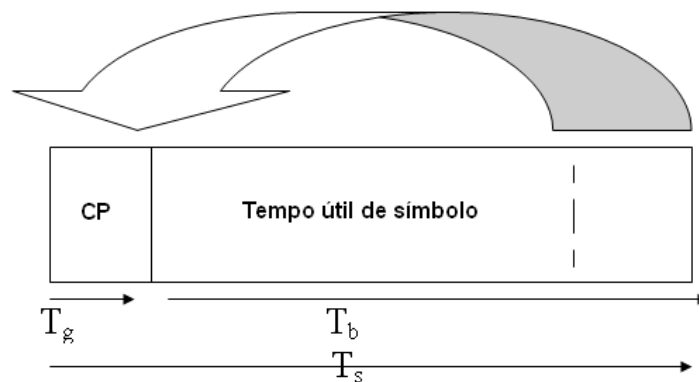
$$R = 1/T = N_{SC} / T_s$$

onde:

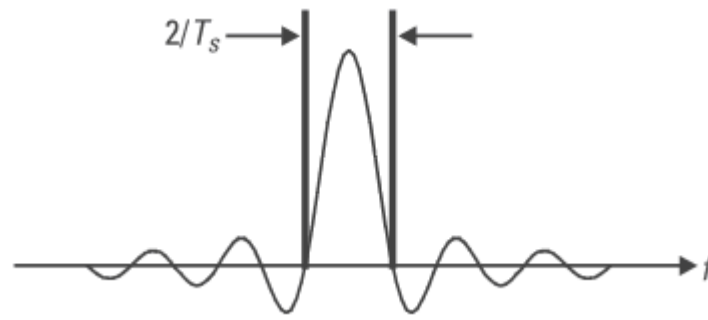
R: taxa de transmissão total de símbolo.



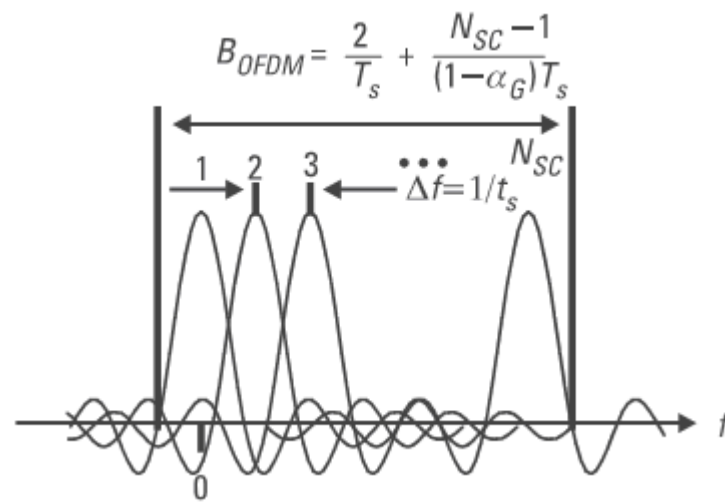
(a)



(b)



(c)



(d)

Figura 3.9 - Forma atual do OFDM: (a) e (b) técnica de extensão cíclica, (c) espectro de frequência do pulso de onda, e (d) espectro de frequência do sinal OFDM.

E a largura de banda, em termos do lóbulo principal, é escrita como (HARA; PRASAD, 2003, p. 38):

(3.18)

$$B_{OFDM} = \frac{2}{T_s} + \frac{N_{SC} - 1}{(1 - \alpha_G) T_s}$$

onde B_{OFDM} é a largura de banda em termos do lóbulo principal e α_G é definido como:

(3.19)

$$\alpha_G = \frac{\Delta_G}{T_s}$$

Quando o número de subportadoras é grande, a largura de banda do OFDM normalizada pelo sinal R é escrita como

(3.20)

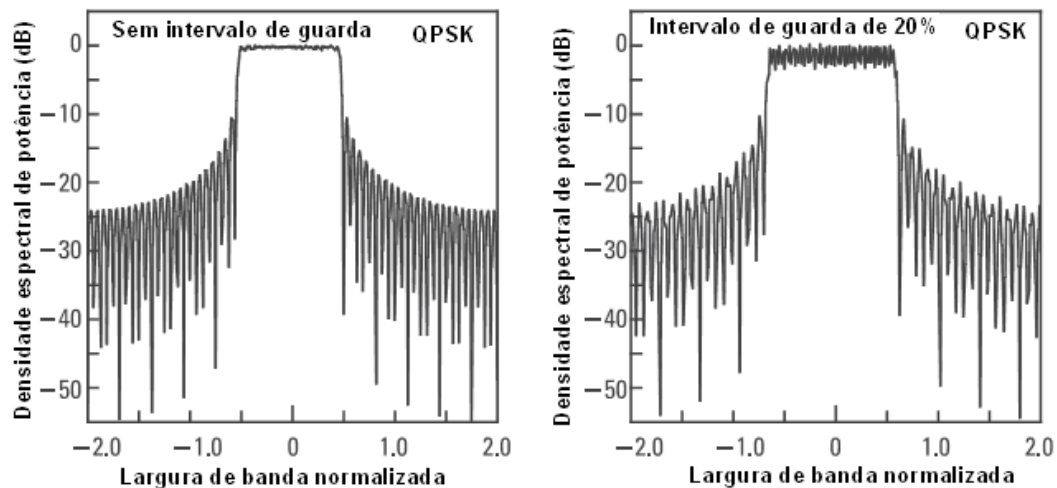
$$\frac{B_{OFDM}}{R} = \frac{1}{1 - \alpha_G}$$

onde:

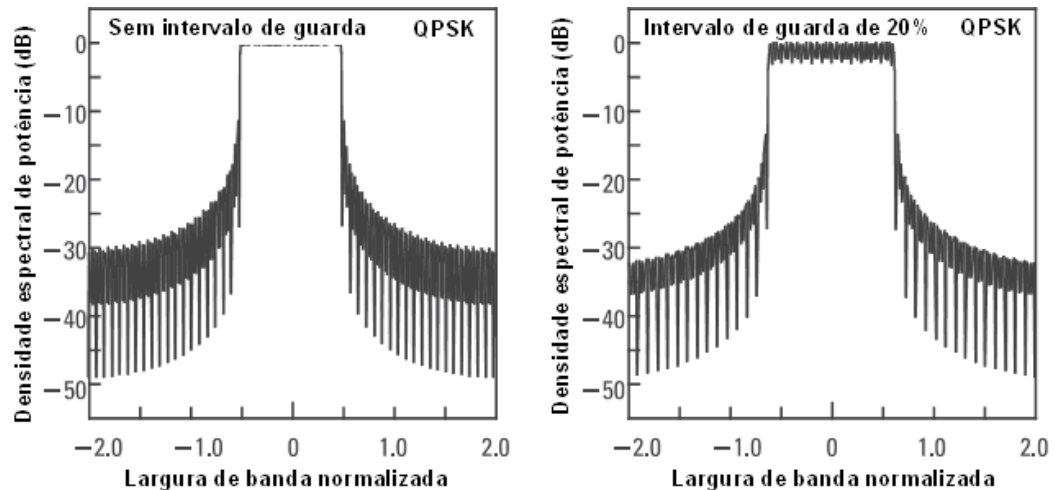
B_{OFDM}/R : Largura de banda do OFDM normalizada pelo sinal R.

A Figura 3.10 (HARA; PRASAD, 2003, p. 40) mostra os espectros de potência transmitida normalizados pela taxa de transmissão de símbolos. A coluna da esquerda mostra os espectros de potência para os sinais OFDM, sem intervalo de guarda, enquanto o lado direito da coluna mostra os espectros de potência para os sinais OFDM com intervalo de guarda 20% (ou seja, a atual forma de OFDM).

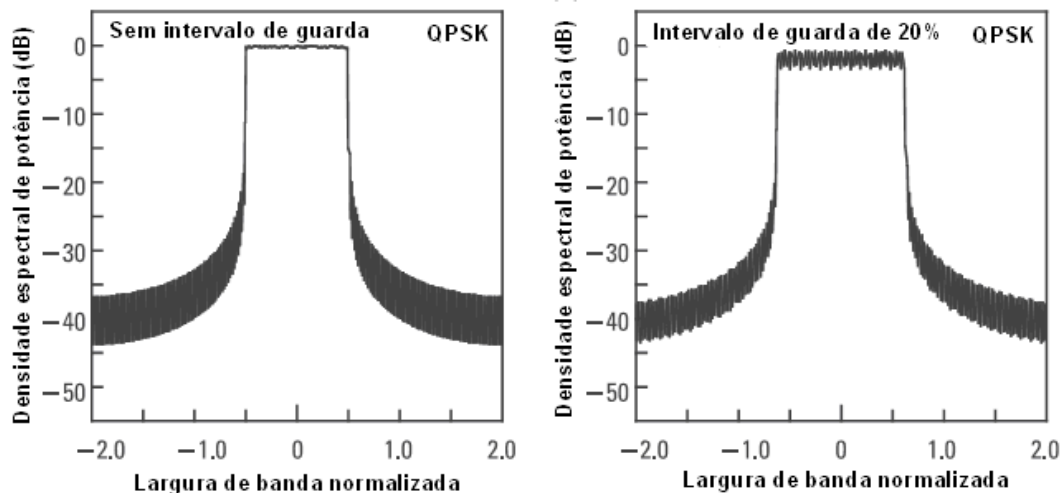
À proporção que aumenta o número de subportadoras, o nível do lóbulo lateral (a radiação fora da banda) pode ser reduzido. Além disso, a inserção do intervalo de guarda introduz uma expansão da largura de banda transmitida [ver Equação (3.20)].



(a)



(b)



(c)

Figura 3.10 - Espectros de potência de sinais OFDM transmitidos: (a) 16 subportadoras, (b) 64 subportadoras e (c) 256 subportadoras.

A grande vantagem do OFDM é que o tempo do símbolo é elevado devido à banda estreita das subportadoras. Isso oferece a possibilidade de introduzir uma sobrecarga no domínio do tempo para o benefício da robustez (eficiência) do sistema. Esta sobrecarga, conforme visto anteriormente, é chamada de intervalo de guarda ou prefixo cíclico. Devido ao espalhamento de retardo (múltiplos caminhos de propagação em ambientes NLOS - multipercursos) símbolos são estendidos no domínio do tempo. Ao tomar um retardo razoável, como o intervalo de guarda T_g , ninguém pode garantir que não haverá interferência entre símbolos no sistema. Todos os efeitos dos vários caminhos de propagação só afetam o intervalo de guarda e não afetam o tempo útil do símbolo T_b .

Multipercursos causam réplicas de retardos de tons que chegam ao receptor com algum espalhamento de retardo. Isso leva ao desalinhamento entre senóides, que precisam ser alinhadas para serem ortogonais. O espalhamento de retardo é um valor absoluto, ou seja, depende do ambiente de propagação e não pode ser alterado pelo homem. Nas áreas urbanas ocorrem muitas difrações e reflexões, gerando grande espalhamento de retardo.

Como visto anteriormente, o prefixo cíclico é uma cópia da última parcela dos dados do símbolo anexado à frente do símbolo, durante o intervalo de guarda.

Ao adicionar esse prefixo cíclico no canal, é garantida a ortogonalidade, impedindo essencialmente uma subportadora de interferir em outra e, assim, é resolvido o problema da ICI.

O prefixo cíclico deve ter o tamanho adequado para servir como um intervalo de guarda, a fim de eliminar a ISI, e que torna o espalhamento de retardo razoável com o intervalo de guarda ($\Delta_G > T_{\max}$). Assim, o intervalo de guarda deve ser maior que o máximo espalhamento de retardo do canal e pequeno em relação ao tempo útil do símbolo T_b , para minimizar a sobrecarga ($\Delta_G < t_s$).

O Padrão 802.16 oferece o seguinte conjunto de T_g / T_b : 1/4, 1/8, 1/16, 1/32. onde:

Δ_G ou T_g : comprimento do intervalo de guarda (prefixo cíclico do intervalo de guarda, que evita a interferência entre símbolos) e

t_s ou T_b : período de observação (comprimento útil do símbolo ou tempo útil de símbolo contendo a transmissão de dados).

Quanto maior o valor para o intervalo de guarda, maior será a proteção contra os efeitos dos multipercursos e menor a capacidade do sistema de rede.

Um intervalo de guarda não é possível em banda larga FDM porque o tempo do símbolo (largura da portadora) é curto. Ele seria apenas uma fração do intervalo de guarda necessário. Isso geraria uma sobrecarga extraordinária do intervalo de guarda.

3.5. Parâmetros do Sistema OFDM em Interface Aérea LTE (*Long Term Evolution*)

A estrutura do quadro do sistema LTE consiste de um quadro rádio de 10 ms de comprimento, contendo 20 compartimentos de 0,5 ms de duração como ilustrado na Figura 3.11 (AKRAM, 2007, p. 36). A estrutura do quadro é válida para transmissões de TDD e FDD.

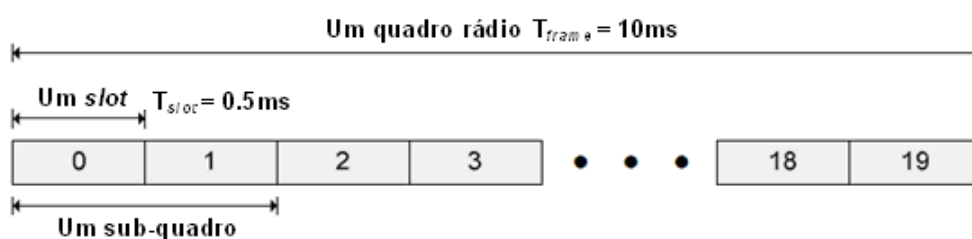


Figura 3.11 - Estrutura do quadro genérico EUTRAN.

O sinal transmitido (*downlink*) em cada compartimento (*slot*) de tempo é definido por uma grade de recursos de N_{BW}^{DL} subportadoras e N_{symbol}^{DL} símbolos OFDM como ilustrado na Figura 3.12. Cada elemento da grade de recursos é chamado elemento de recurso e é identificado unicamente pelo índice par (k, l) , onde k e l são os índices no domínio da frequência e tempo, respectivamente. O espaçamento (separação) de frequência entre as sub-portadoras está definida para 15 kHz.

O valor de subportadoras (N_{BW}^{DL}) depende da largura de banda de transmissão da célula e assume os valores no intervalo $72 \leq N_{BW}^{DL} \leq 2048$ (AKRAM, 2007, p. 37). O valor de símbolos (N_{symbol}^{DL}) está definido para 7 ou 6, dependendo do comprimento do prefixo cíclico usado no compartimento de tempo. O prefixo cíclico é mais comumente utilizado em canais com grande retardo, como terreno montanhoso. Um bloco de recursos é definido como N_{symbol}^{DL} símbolos OFDM consecutivos no domínio do tempo e $N_{BW}^{DL} = 12$ subportadoras consecutivas no domínio da frequência. Um bloco de recursos, portanto, consiste em $N_{BW}^{DL} \times N_{symbol}^{DL}$ elementos de recursos, e é o valor mínimo de elementos de recursos alocados para um usuário no *downlink*.

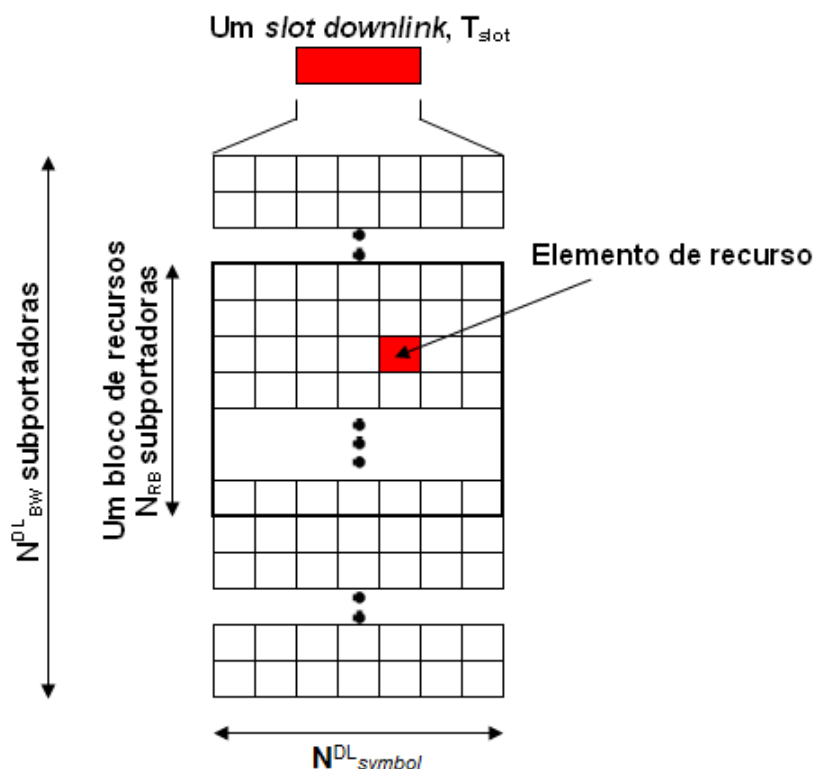


Figura 3.12 - Grade de Recursos *downlink* (estrutura do quadro genérico, prefixo cíclico normal).

Os parâmetros do OFDM (*downlink*) estão apresentados na Tabela 3.1. Para dar um exemplo, a largura de banda de uma célula de 5 MHz é dividida em 512 subportadoras, cada uma separada por 15 kHz. Das 512 subportadoras, apenas 300 subportadoras estão ocupadas com a transmissão de dados, enquanto que as subportadoras restantes servirão como proteção contra as faixas adjacentes. As subportadoras estão dispostas simetricamente em torno da frequência da portadora (0 Hz após conversão de descida (*down-conversion*)) ou DC, e não incluem o DC. Cada intervalo de tempo na célula é de 0,5 ms de comprimento, e contém 7 símbolos OFDM quando o prefixo cíclico normal é usado. Em tal configuração o prefixo cíclico é 4.69/36 ms para os 6 primeiros símbolos OFDM, e 5.21/40 ms para o símbolo OFDM passado. Quando o prefixo cíclico estendido é utilizado em uma célula, o compartimento de tempo de 0.5ms é composto por 6 símbolos OFDM e cada símbolo OFDM contém um prefixo cíclico de comprimento 16.67/128 ms.

Tabela 3.1. Parâmetros do Sistema OFDM em EUTRAN.

BW de Transmissão		1.25MHz	2.5MHz	5MHz	10MHz	15MHz	20MHz
Duração de um <i>slot</i>		0.5ms					
Espaçamento de Subportadora		15kHz					
Tamanho FFT		128	256	512	1024	1536	2048
Número de subportadoras ocupadas		72	150	300	600	900	1200
Número de símbolos OFDM por <i>slot</i> (normal/ CP estendido)		7/6					
Comprimento CP (μs)	normal	(4.69/9)x6, (5.21/10)x1	(4.69/18)x6, (5.21/20)x1	(4.69/36)x6, (5.21/40)x1	(4.69/72)x6, (5.21/80)x1	(4.69/108)x6, (5.21/120)x1	(4.69/144)x6, (5.21/160)x1
	estendido	16.67/32	16.67/64	16.67/128	16.67/256	16.67/384	16.67/512

3.6. Conclusão

O Capítulo 3 descreveu a história e o princípio das técnicas baseadas em multiportadoras, incluindo a técnica OFDM. Este tem sido estudado e implementado para combater as deficiências da transmissão no canal (desvanecimento seletivo em frequência, causado, principalmente, pelos raios secundários provenientes de reflexões).

A vantagem do OFDM, especialmente na propagação por multipercursos, com interferências, e em ambientes com desvanecimento, torna a tecnologia uma alternativa promissora em comunicações móveis.

A técnica de transmitir dados através da divisão da sequência de dados em sequências menores para modulação por subportadoras, já era conhecida há muito tempo. No entanto havia o problema de sincronizar bancos de osciladores. A técnica DFT (*Discrete Fourier Transform*) resolveu este problema.

OFDM é muito eficaz em ambientes de multipercursos de raios (*frequency selective fading*). Por esta razão, OFDM vem sendo utilizado em redes wireless. O 3GPP também decidiu adotar esta tecnologia.

O sinal OFDM no tempo é dado pela soma dos sinais das diferentes portadoras. Este sinal total $s(t)$ sofre grandes variações no tempo, o que caracteriza altos valores de PAPR (*Peek to Average Power Ratio*). Isso vem a limitar o uso do

OFDM nos enlaces *uplink*, pois o celular tem limitações para o emprego de potências mais elevadas, por exemplo, maior gasto de baterias. Então, o 3GPP estabeleceu que seria empregado o OFDM no *downlink* e o SC-FDMA no *uplink*. Devido ao fato do SC-FDMA apresentar portadora única, os seus níveis de PAPR são inferiores.

O PAPR (*Peak-to-Average Power Ratio*) para um sistema OFDM convencional é calculado através da seguinte forma:

$$\text{PAPR (dB)} = 10 \log_{10} (N)$$

onde N é o número de subportadoras.

Para N = 4, o PAPR de um sistema OFDM convencional é 6,02 dB.

Várias técnicas de codificação têm sido utilizadas para fornecer métodos de redução do PAPR em detrimento da largura de banda.

As três técnicas mais populares de codificação para atenuar a distorção não linear causada pelo HPA (*High Power Amplifier* - Amplificador de Alta Potência) são *Partial Transmit Sequences* (PTS - Sequências Parciais de Transmissão), *Selective Mapping* (SLM - Mapeamento Seletivo) e *Block Coding* (Codificação em Bloco).

A utilização destas técnicas está limitada a sistemas OFDM com um pequeno número de subportadoras e pequenos tamanhos de constelações, caso contrário, a busca exaustiva necessária para os blocos de códigos será intratável. Além disso, para conseguir a redução do PAPR quando há um maior número de subportadoras ou um grande tamanho de constelação, a taxa de código torna-se muito baixa. Assim, os benefícios reais da codificação para redução do PAPR para os sistemas OFDM de interesse prático são extremamente pequenos.

O estudo abordou o uso da Transformada Discreta de Fourier, a inserção de intervalo de guarda, a inserção de prefixo cíclico para a atual forma do OFDM e finalizou mostrando os parâmetros do sistema OFDM em Interface Aérea LTE (*Long Term Evolution*).

4. CARACTERÍSTICAS DO OFDM

4.1. Introdução

O princípio básico do OFDM é dividir uma alta taxa de fluxo de dados em uma série de fluxos de baixa velocidade, que são transmitidos, simultaneamente, sobre um número de subportadoras.

A ISI é eliminada quase totalmente pela introdução de um intervalo de guarda em cada símbolo OFDM, onde o sinal OFDM é ciclicamente estendido (prolongado) para evitar a interferência entre subportadoras. Tais frequências das subportadoras são escolhidas de modo que as subportadoras sejam ortogonais entre si, o que significa que conversas cruzadas (*cross-talk*) entre os subcanais são eliminadas. Isso simplifica muito o projeto do transmissor e do receptor e ao contrário do FDM convencional, não é necessário um filtro separado para cada subcanal.

Multiplexação é o método de partilha de largura de banda com outros canais de dados independentes. Em OFDM, o sinal é primeiro dividido em canais de dados independentes, modulados e, em seguida, novamente multiplexados para criar a portadora OFDM.

Duas outras definições de OFDM são:

- *Coded Orthogonal Frequency Division Multiplexing* (COFDM) é o mesmo que OFDM, exceto que a correção de erros em repasse (FEC - *Forward Error Correction*) é aplicada ao sinal, antes da transmissão. Trata-se de superar os erros na transmissão devido às portadoras perdidas no desvanecimento seletivo em frequência, devido ao ruído de canal e outros efeitos de propagação. Como o foco principal desta dissertação é o OFDM, para esta discussão os termos OFDM e COFDM são utilizados alternadamente, mas

presume-se que qualquer sistema prático irá usar a correção de erro, o que seria COFDM.

- *Scalable Orthogonal Frequency Division Multiplexing* (SOFDMA) é semelhante ao OFDMA, mas permite ajustar o número de subportadoras (tamanho de FFT) para a largura de banda do canal de transmissão. Também a vantagem do SOFDMA sobre OFDMA é que tem menor espaçamento de frequência (Δf). O canal com largura de banda (BW) de 3,5MHz para OFDMA possui 256 subportadoras enquanto para SOFDMA possui 512 subportadoras.

Em FDMA, cada usuário é tipicamente alocado a um único canal, que é usado para transmitir todas as informações do usuário. A largura de banda de cada canal é normalmente de 10 kHz a 30 kHz para comunicações de voz, no entanto, a largura de banda mínima exigida para a comunicação de voz é de apenas 3 kHz. A largura de banda alocada é maior do que a largura mínima necessária para evitar que haja interferência entre canais adjacentes. Essa largura de banda adicional é para permitir que os sinais dos canais vizinhos possam ser filtrados, e para permitir qualquer desvio na frequência central no transmissor ou no receptor. Em um sistema típico, até 50% do espectro total é desperdiçado devido ao espaçamento extra entre os canais. Este problema torna-se pior com largura de banda de canal mais estreita, e com os aumentos de banda de frequência.

A maioria dos sistemas de telefonia digital utiliza codificadores de voz para comprimir a voz digitalizada. Isso permite um aumento da capacidade do sistema devido a uma redução na largura de banda necessária para cada usuário. Codificadores de voz exigem uma taxa de dados entre 4-13 kbps, dependendo da qualidade do som e do tipo utilizado. Assim, cada usuário necessita de um mínimo de largura de banda entre 2-7 kHz, com modulação QPSK. No entanto, FDMA simples não controla tais bandas estreitas de forma muito eficiente.

OFDM supera a maioria dos problemas do FDMA e TDMA. OFDM divide a largura de banda disponível em vários canais de banda estreita, normalmente entre 100 e 8000 canais. As portadoras de cada canal são ortogonais entre si, permitindo-lhes serem espaçadas muito proximamente, sem custos adicionais, como no exemplo do FDMA. Devido a isso, não há grande necessidade para os usuários serem multiplexados no tempo como no TDMA, portanto, não há sobrecarga associada com alternância entre usuários.

Cada portadora de um sinal OFDM tem uma largura de banda muito estreita (1kHz), assim a taxa de símbolo resultante é baixa. Como o espalhamento de retardo deve ser muito grande para causar interferência intersimbólica significativa (por exemplo $> 100 \mu\text{seg}$), isso resulta em um sinal com uma alta tolerância ao espalhamento de retardo por multipercursos.

Este capítulo discute as características dos sistemas OFDM. Na Seção 4.2, é apresentado um modelo de canal rádio, onde um sinal transmitido não é somente distorcido pelo desvanecimento multipercurso, mas também, corrompido pelo ruído térmico.

Na Seção 4.3, mostra-se a relação entre o número de subportadoras, a seletividade em frequência e a seletividade no tempo. Mostra-se, também, a relação entre o comprimento do intervalo de guarda e a seletividade em frequência.

É enfatizada a análise teórica e discutidas as vantagens e as desvantagens do sistema OFDM, incluindo a robustez contra desvanecimento seletivo em frequência (Seção 4.4), a robustez contra ruídos impulsivos (Seção 4.5), a sensibilidade ao deslocamento de frequência (Seção 4.6), a sensibilidade à amplificação não-linear (Seção 4.7) e a sensibilidade para conversões A/D e D/A (Seção 4.8).

4.2. Modelo do Canal Rádio

Em um canal de rádio, em geral, um sinal transmitido não é somente distorcido pelo desvanecimento multipercurso, mas, também, corrompido pelo ruído térmico, como mostrado na Figura 4.1. O sinal recebido pelo canal é escrito como:

(4.1)

$$r(t) = \int_{-\infty}^{\infty} h(\tau; t) s(t - \tau) d\tau + n(t)$$

onde:

$r(t)$: sinal recebido pelo canal,

$h(\tau, t)$: resposta do canal ao impulso (função de transferência do canal ou função espalhamento de retardo de entrada) e

$n(t)$: ruído gaussiano AWGN (*Additive White Gaussian Noise*) com densidade espectral de potência igual a $N_0 / 2$.

A equação (4.1) mostra que, através de um canal de rádio, um sinal transmitido $s(t)$ é convoluído por $h(\tau, t)$ e, em seguida, é adicionado o ruído $n(t)$, portanto, pode-se considerar que, no domínio da frequência, o desvanecimento é um ruído multiplicativo.

Se a resposta do canal ao impulso é uma constante invariante no tempo, dada por (HARA; PRASAD, 2003, p. 45):

(4.2)

$$h(\tau; t) = h\delta(\tau)$$

onde h é o ganho do canal com valor complexo, então, pode-se ignorar o efeito do desvanecimento e há apenas um ruído gaussiano (AWGN) no canal. Esse canal é chamado de “canal AWGN”.

Por outro lado, se a resposta ao impulso do canal tem uma propriedade variante no tempo (espalhamento do retardo), então há um ruído gaussiano (AWGN) e desvanecimento no canal. Esse canal é chamado de “canal com desvanecimento”.

O desempenho da BER num esquema de modulação/demodulação depende amplamente do sinal recebido, ou seja, da taxa de energia do bit por densidade espectral de potência do ruído (E_b / N_0).

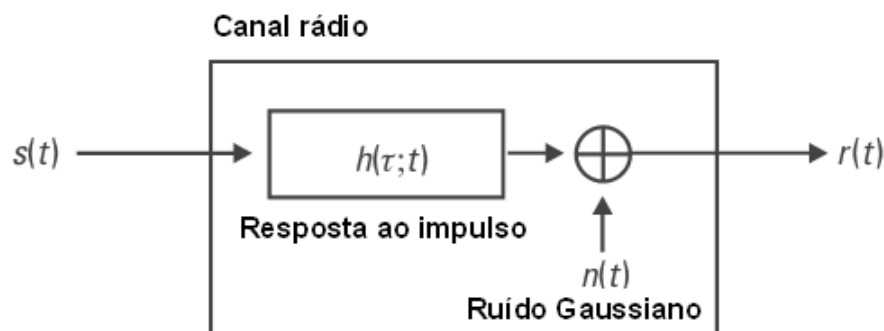


Figura 4.1 - Modelo de Canal Rádio

4.3. Número Ótimo de Subportadoras e Comprimento Ótimo do Intervalo de Guarda

Quando o número de subportadoras aumenta, ocorre uma maior distorção devido ao desvanecimento (rápido) seletivo no tempo (ruído FM aleatório), isso por causa da maior duração do símbolo ($\Delta t < T_c$ e $t_s > T_c$). Ocorre uma maior dispersão do sinal na frequência. O espaçamento entre as subportadoras diminui, ocasionando ICI.

Quando o número de subportadoras diminui, ocorre uma maior distorção devido ao desvanecimento seletivo em frequência, isso por causa do maior espectro de potência de cada subportadora ($\Delta f < BW_c$ e $BW > B_c$). Ocorre uma maior dispersão do sinal no tempo. A duração do símbolo diminui ($t_s < \sigma_T$), ocasionando ISI.

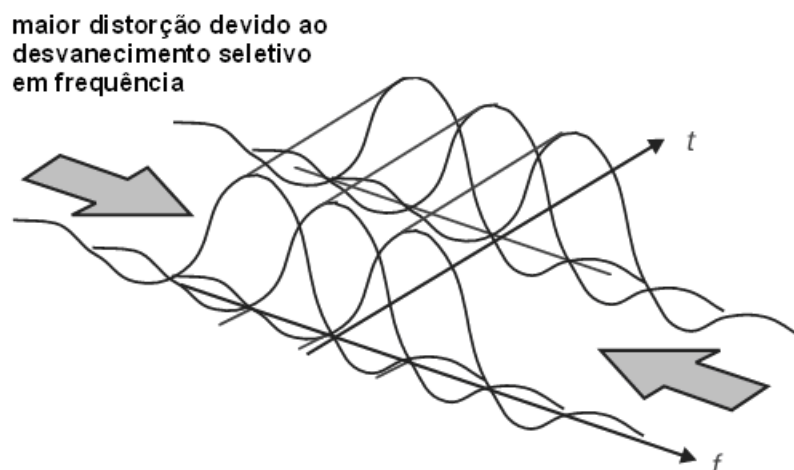
No OFDM, o desempenho da transmissão é mais sensível ao desvanecimento seletivo no tempo e à dispersão do sinal na frequência.

A Figura 4.2 mostra a relação entre o número de subportadoras, a seletividade em frequência, e a seletividade no tempo.

Quando o comprimento do intervalo de guarda aumenta, o desempenho da transmissão piora, por causa da maior perda de potência (menor eficiência espectral). O ideal é que $\Delta_G < t_s$.

Quando o comprimento do intervalo de guarda diminui, o desempenho da transmissão torna-se mais sensível ao desvanecimento seletivo em frequência (dispersão do sinal no tempo), por causa do espalhamento do retardo. O ideal é que $t_s > \sigma_T$, $\Delta_G < t_s$ e $\Delta_G > T_{\max}$ a fim de minimizar a ISI.

A Figura 4.3 mostra a relação entre o comprimento do intervalo de guarda e a seletividade em frequência.



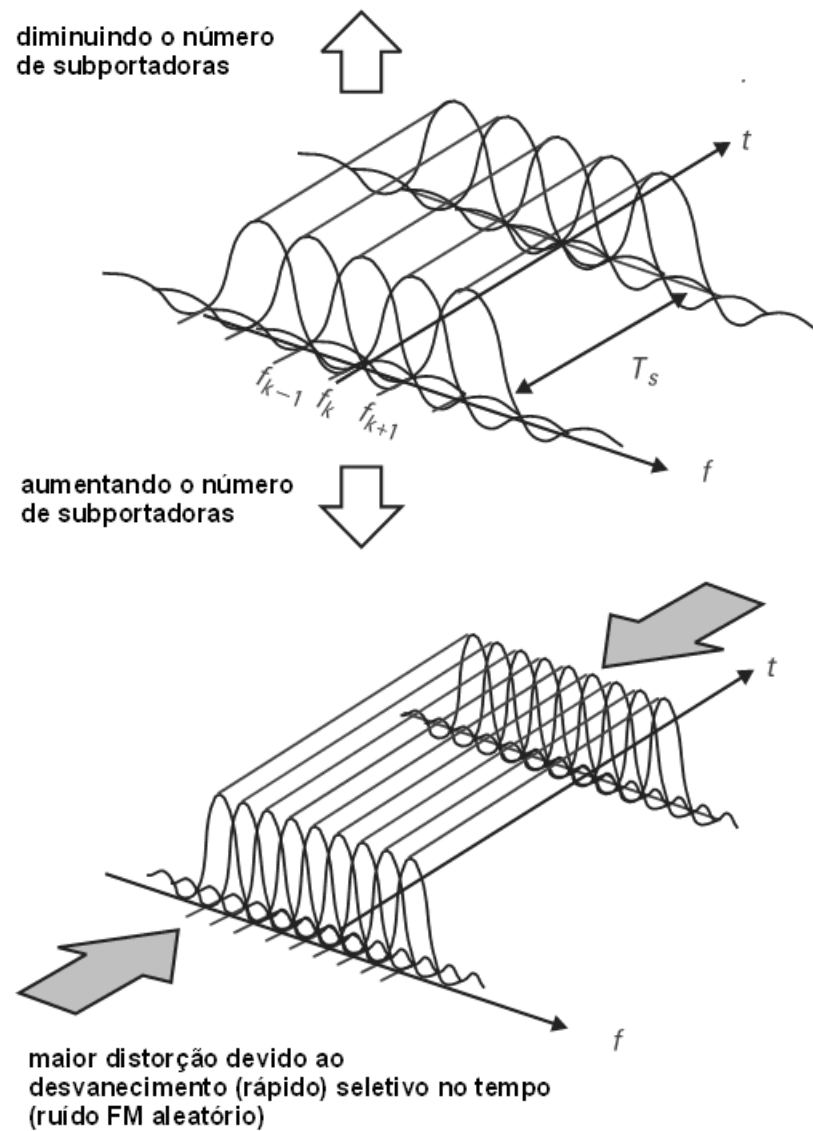
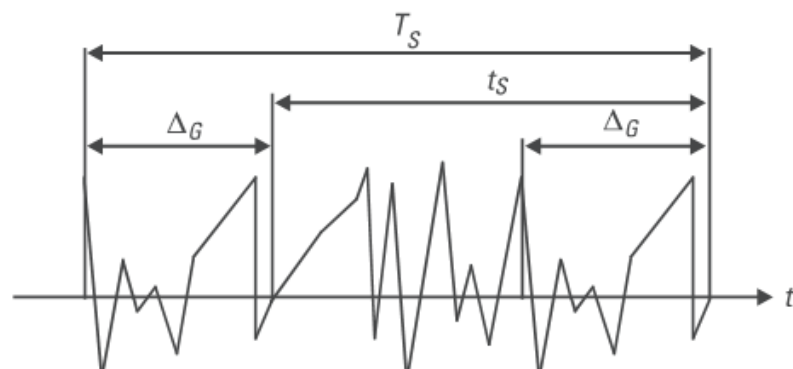


Figura 4.2 - Relação entre número de subportadoras, seletividade em frequência e seletividade no tempo.



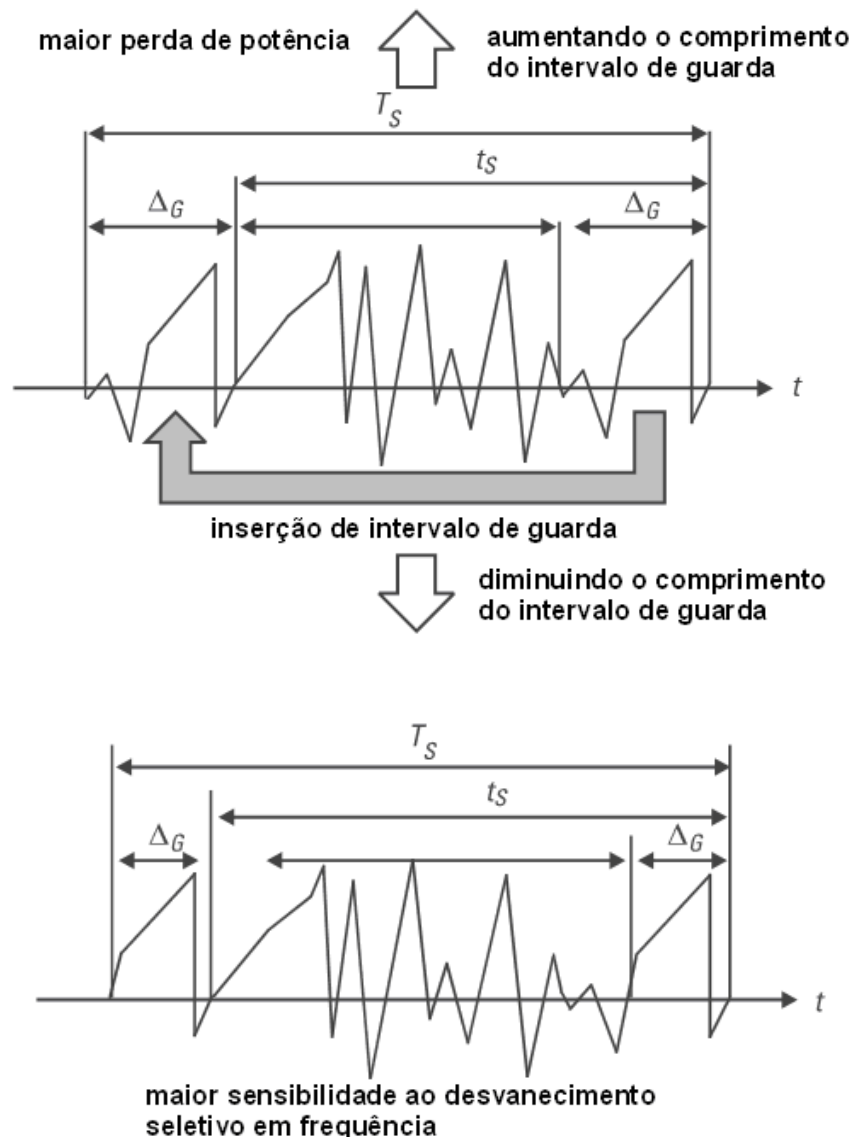


Figura 4.3 - Relação entre o comprimento do intervalo de guarda e seletividade em frequência.

4.4. Robustez contra Desvanecimento Seletivo em Frequência

Conforme visto anteriormente, uma das principais razões para usar OFDM é o aumento da robustez (eficiência) contra o desvanecimento seletivo em frequência.

Interferência entre Símbolos (ISI): No capítulo anterior foi visto que a ISI produz uma forma de distorção de um sinal que faz com que os símbolos transmitidos anteriormente possam ter um efeito sobre o símbolo atualmente recebido. A ISI ocorre quando um sinal é transmitido através de múltiplos caminhos de propagação de diferentes comprimentos.

Empregando um intervalo de guarda (sem transmissão de sinal) com comprimento de $\Delta_G > T_{\max}$ ($T_{\max}$: tempo de retardo), pode-se perfeitamente eliminar a ISI.

4.5. Robustez contra Ruídos Artificiais

Nos últimos anos, uma variedade de ruídos artificiais, que são, na sua maioria, de natureza não-gaussiana, tem drasticamente deteriorado o desempenho dos sistemas de comunicações digitais, tais como linha digital assimétrica de assinante (ADSL - *Asymmetric Digital Subscriber Line*), a radiodifusão digital (*digital broadcasting*) e LANs sem-fio em 2.4 GHz.

Um sistema OFDM substitui o uso de um banco de filtros passa-banda por DFT, de modo que ele possa ser robusto contra ruídos impulsivos, ao dividir a energia do ruído, espectralmente generalizada, em várias sub-bandas.

4.6. Sensibilidade ao Deslocamento de Frequência

Quando uma frequência deslocada (*offset*) é introduzida no canal, ou quando há um descompasso na frequência da portadora entre o transmissor e o receptor local, a BER degrada drasticamente, uma vez que ocorre uma grave interferência entre subportadoras, por causa da sobreposição de espectros de potência entre subportadoras vizinhas. Esta sensibilidade ao deslocamento de frequência é, muitas vezes, apontada como uma grande desvantagem dos sistemas OFDM.

Supondo-se que uma frequência deslocada (*offset*) f_{off} (ou deslocamento de frequência) e uma fase deslocada (*offset*) Φ_{off} (ou deslocamento de fase) são introduzidas em um canal com desvanecimento *Rayleigh*, seletivo no tempo e em frequência, então, o sinal recebido, em expressão equivalente em banda básica, é escrito como:

(4.3)

$$r(t) = \int_{-\infty}^{\infty} h(\tau; t) s(t - \tau) e^{-j(2\pi f_{\text{off}}t + \phi_{\text{off}})} d\tau + n(t)$$

Comparado com o caso quando ignoradas as interferências entre subportadoras, o sistema OFDM é muito sensível para a frequência deslocada (*offset*), ou seja, piora a BER, mesmo para pequenas frequências *offsets*. Nesse caso, é preciso métodos de compensação de frequência deslocada (*offset*) para manter a BER em níveis aceitáveis, mesmo quando uma grande frequência deslocada (*offset*) é introduzida no canal.

Interferência entre Portadoras (ICI): No capítulo anterior, foi visto que o movimento do usuário em relação à estação base é que provoca, no domínio da frequência, um deslocamento *Doppler*. As subportadoras são, então, deslocadas em frequência e não são mais ortogonais. Devido ser necessária a colocação exata da subportadora no sistema, este deslocamento *Doppler* causa problemas para o sistema OFDM, resultando em uma alta taxa de erro de bit (BER), se não houver uma compensação adequada a ser aplicada.

Para resolver o problema ICI, têm-se diferentes técnicas, das quais uma dessas técnicas é usar um filtro de ajuste para manter a ortogonalidade entre as subportadoras, que foi perdida devido ao efeito *Doppler*.

Foi mostrado, anteriormente, que o sistema OFDM emprega a IDFT e DFT. Ao empregar DFT retangular na janela do receptor, a interferência entre subportadoras pode perfeitamente ser eliminada.

Foi visto que a técnica de inserção de intervalos de guarda com prefixo cíclico pode perfeitamente eliminar a interferência entre subportadoras, onde o símbolo OFDM é ciclicamente estendido no período de guarda.

4.7. Sensibilidade a Amplificadores não-lineares

Em geral, o amplificador não-linear distorce a amplitude do sinal de entrada. As mudanças bruscas da amplitude de entrada geram maiores componentes espectrais no espectro de potência do sinal, por isso resulta em um espalhamento de espectro. O espalhamento do espectro para amplificação não-linear é uma causa da Interferência entre Canais Adjacentes (ACI - *Adjacent Channel Interference* - Interferência entre Canais Adjacentes).

Quando um sinal OFDM de banda estreita é amplificado por um dispositivo não-linear sua envoltória não é constante. Em outras palavras, a forma de onda é igual a de um ruído Gaussiano de banda limitada, portanto, mesmo na tentativa de

reduzir o espalhamento de espectro com um grande *back-off* (recuo) de entrada, não se pode eliminar, perfeitamente, os fatores de produção repentina das amplitudes altas, ou seja, não se pode eliminar todas as mudanças bruscas da amplitude de entrada.

Além disso, a radiação gerada a partir de uma subportadora inicia, também, a interferência entre subportadoras, para as subportadoras vizinhas.

Assim, a elevada interferência entre subportadoras deteriora drasticamente o desempenho da BER. Esta sensibilidade à amplificação não-linear é também apontada como uma das principais desvantagens dos sistemas OFDM, bem como a sensibilidade à frequência deslocada (*offset*) discutida anteriormente.

4.8. Sensibilidade para Conversões A/D e D/A

A característica variável da envoltória de um sinal OFDM introduz uma outra sensibilidade para dispositivos não-lineares, tais como conversores A/D e D/A.

Supondo-se que os processadores de sinal digital no transmissor e receptor tenham a mesma resolução de q bits, ou seja, o transmissor utiliza um conversor D/A de q bits, considerando que o receptor utiliza um conversor A/D de q bits. A Figura 4.4 mostra o modelo do sistema.

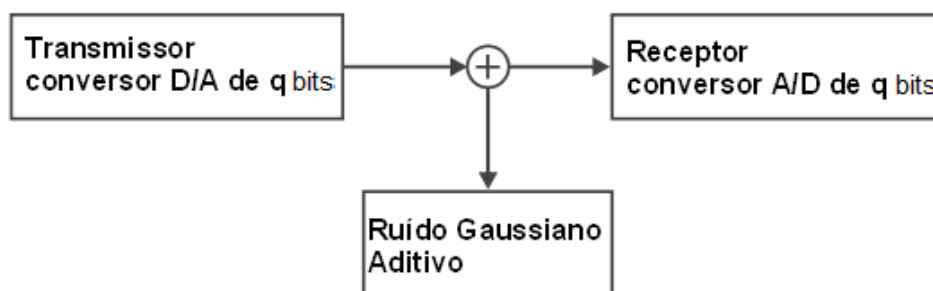


Figura 4.4 - Definição da resolução de conversores A/D e D/A.

Se for possível lidar com o sinal OFDM como um ruído Gaussiano de banda estreita com média zero e variância σ_n^2 , então 99,7% dos valores de amplitude estarão nos intervalos entre $[-3\sigma_s, +3\sigma_s]$ e 99,994% dos valores de amplitude estarão nos intervalos entre $[-4\sigma_s, +4\sigma_s]$.

Aqui, σ_s é chamado de "amplitude eficaz." No transmissor e receptor, distorções não-lineares resultantes da quantização e recorte (*clipping*) são adicionados ao sinal OFDM.

Para o mesmo valor de resolução A/D e D/A, diferentes intervalos de quantização acarretam em diferentes BER.

4.9 Conclusão

Este capítulo discutiu as características dos sistemas OFDM. Apresentou um Modelo do Canal Rádio, onde um sinal transmitido não é somente distorcido pelo desvanecimento multipercurso, mas também, corrompido pelo ruído térmico. Mostrou a relação entre o número de subportadoras, a seletividade em frequência, e a seletividade no tempo. Mostrou, também, a relação entre o comprimento do intervalo de guarda e a seletividade em frequência. Enfatizou a análise teórica e discutiu as vantagens (os prós) e as desvantagens (contras) do sistema OFDM, resumidas a seguir:

Vantagens:

- Robusto contra o desvanecimento seletivo em frequência, causado pela propagação em multipercursos.
- Robusto contra a interferência co-canal em banda estreita.
- Robusto contra ruídos impulsivos artificiais.
- Pode, facilmente, adaptar-se às condições do canal sem equalização complexa.
- Robusto contra interferência entre símbolos (ISI).
- Eficiente implementação usando FFT.
- Baixa sensibilidade para erros de sincronização no tempo.
- Filtros de subcanal no receptor não são necessários (ao contrário do convencional FDM).
- Facilidade em Redes de Frequência Simples, isto é, diversidade na transmissão.

Desvantagens:

- Sensível ao deslocamento de frequência.
- Sensível à amplificação não-linear.
- Sensível ao deslocamento Doppler.
- Sensível aos problemas de sincronização de frequência.

- Ineficiente consumo de potência do transmissor, devido à exigência de potência linear do amplificador.
- Sensibilidade para conversões A/D e D/A.

5. ESTIMAÇÃO DE CANAL E ALOCAÇÃO DE PILOTO

5.1. Introdução

Para que o desempenho de um sistema OFDM possa ser avaliado, o conhecimento do canal precisa ser estimado, já que não está disponível na vida real.

A estimação do canal pode ser feita de várias formas: com ou sem a ajuda de um modelo paramétrico; com a utilização de propriedades de correlação no tempo e/ou na frequência do canal sem fio; baseada em máscara ou formação de piloto; adaptativa ou não.

Os métodos não paramétricos são tentativas de estimar as quantidades de interesse (por exemplo, a resposta em frequência) sem depender de um modelo de canal específico. Por outro lado, a estimação paramétrica assume um determinado modelo de canal, determina os parâmetros deste modelo e deduz as quantidades de interesse.

As correlações no tempo e na frequência são propriedades específicas dos canais que podem ser incorporadas no método de estimação, melhorando a qualidade da estimação.

Os métodos de estimação baseada em pilotos são os métodos mais comumente utilizados, que são aplicáveis em sistemas onde o remetente emite algum sinal conhecido. A estimação baseada em máscara, por outro lado, depende de algumas propriedades do sinal (por exemplo, estacionariedade do sinal) e raramente é usada em sistemas OFDM.

Os métodos adaptativos de estimação de canal são, normalmente, utilizados para canais que variam rapidamente no tempo.

Neste capítulo, serão investigados os métodos de estimação de canais não adaptativos, baseados em piloto. É preciso, no entanto, explorar as propriedades de correlação em um canal.

5.2. Estimação de Canal

A Figura 5.1 apresenta um sistema OFDM, que utiliza o método de estimação de canal baseado em piloto, para a equalização no receptor final.

No transmissor, os dados binários são mapeados para uma modulação específica (QPSK, 16QAM, 64QAM) e os dados modulados sofrem conversão serial para paralelo (S/P), formando um vetor de $(N_0 - N_{\text{piloto}})$ símbolos.

N_0 é o número de subportadoras ocupadas e N_{piloto} é o número de subportadoras piloto.

São inseridos N_{piloto} símbolos pilotos nos dados modulados, formando X_k dados transmitidos de comprimento N_0 , no domínio da frequência.

Zeros de enchimento são inseridos em X_k , aumentando o seu comprimento de N_0 para N_c .

A Transformada (Rápida) Inversa de Fourier (IFFT) é aplicada sobre o vetor X_k , criando o vetor $x(n)$ no domínio no tempo.

A IFFT transforma o espectro de frequência em um sinal no domínio do tempo. A IFFT converte um número de pontos complexos de dados, de comprimento que é uma potência de 2, para o sinal no domínio do tempo, com o mesmo número de pontos.

Então N_{cp} prefixos cíclicos são pré-anexados em $x(n)$ formando o vetor $x_g(n)$ de $N_c + N_{\text{cp}}$ símbolos.

Depois da conversão paralelo para serial (P/S), da conversão digital para analógico (D/A) e filtragem passa-baixa, $x_g(n)$ é transmitido ao longo de um canal AWGN linear e invariante no tempo com função de transferência $h(n)$ e com ruído $z(n)$.

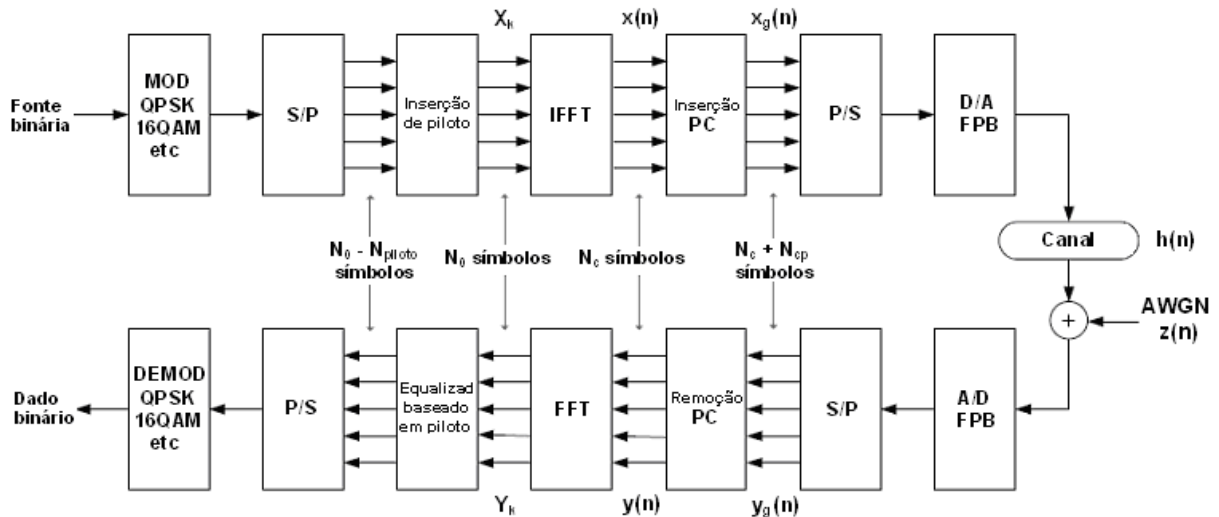


Figura 5.1 - Modelo de um sistema OFDM baseado em piloto.

No receptor, é executado o conjunto oposto das operações. Após a conversão A/D e filtragem passa-baixa, o sinal recebido passa por uma conversão S/P e é armazenado em $y_g(n)$. O prefixo cíclico é, então, removido do $y_g(n)$, criando o vetor de símbolo recebido $y(n)$, de dimensão N_c .

A Transformada Rápida de Fourier (FFT) é aplicada sobre o vetor $y(n)$, criando o vetor Y_k no domínio da frequência.

A FFT transforma o sinal no domínio do tempo em seu espectro de frequência equivalente. Isso é feito encontrando a forma de onda equivalente, gerada por uma soma de componentes senoidais ortogonais. A amplitude e a fase das componentes senoidais representam o espectro de frequência do sinal no domínio do tempo.

Zeros de enchimento são retirados de Y_k , encurtando o seu comprimento de N_c para N_0 . A equalização é aplicada, utilizando um método de estimação de canal baseado em piloto, e os símbolos pilotos são removidos do sinal equalizado. Os dados equalizados passam, em seguida, por uma conversão P/S e pela demodulação, criando as estimativas dos dados binários transmitidos.

A relação entre o sinal transmitido X_k e o recebido Y_k é dada como

(5.1)

$$Y_k = H_k \cdot X_k + Z_k$$

onde:

Z_k é o ruído da k -ésima subportadora no domínio da frequência e

H_k é a função de transferência do canal na k -ésima subportadora.

Para estimar o canal, são necessários símbolos pilotos. Foi assumido que todas as p -ésimas subportadoras contém símbolos pilotos conhecidos (X_{pk}). Usando os símbolos pilotos conhecidos (X_{pk}) e os símbolos recebidos (Y_{pk}) nas subportadoras pilotos, pode-se calcular a estimação de canal (H_{pk}) baseada em pilotos como

(5.2)

$$\hat{H}_{pk} = \frac{Y_{pk}}{X_{pk}} + \frac{Z_{pk}}{X_{pk}} = H_{pk} + Z'_{pk}$$

onde:

Z_{pk} é a contribuição do ruído na pk -ésima subportadora e

Z'_{pk} é a contribuição do ruído medido nesta subportadora.

Diferentes métodos podem ser aplicados para estimar o canal de todas as frequências das subportadoras e não apenas das frequências das subportadoras dos pilotos.

A Figura 5.2 mostra os detalhes do método de estimação de canal descrito acima. Depois da FFT e da remoção dos zeros de enchimento, o sinal dos pilotos locais recebido é extraído do sinal recebido. O canal é estimado nas frequências das subportadoras dos pilotos usando os pilotos conhecidos. Diferentes métodos podem ser aplicados para estimar o canal em todas as frequências H_{pk} .

Nas seções seguintes, será investigado o impacto da alocação de piloto no desempenho da estimação do canal, e serão explorados os diferentes métodos utilizados para a estimação do canal.

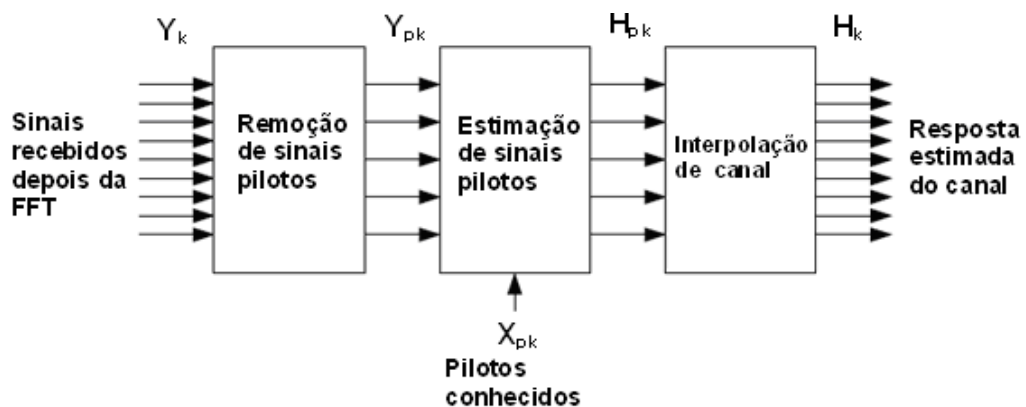


Figura 5.2 - Estimação do canal no domínio da frequência.

5.2.1. Interpretação no Domínio do Tempo-Frequência e Alocação de Piloto

A transmissão OFDM, mostrada na Figura 5.1 (AKRAM, 2007, p. 41), também pode ser descrita usando duas dimensões no domínio do tempo e no domínio da frequência. A Figura 5.3 mostra uma representação no domínio do tempo-frequência, onde símbolos OFDM EUTRAN são transmitidos sequencialmente. Conforme mostrado na Figura 5.1, um símbolo OFDM consiste de N_c subportadoras, simetricamente dispostas em torno da frequência da portadora (0Hz após conversão de descida (*down-conversion*)) ou DC, e não incluem o DC. Note-se que apenas N_o de N_c subportadoras são utilizadas para transmissão de dados; nenhum dado é transmitido em DC e subportadoras de borda (para evitar interferência nas bandas adjacentes). Os valores possíveis para N_o e N_c estão na Tabela 3.1.

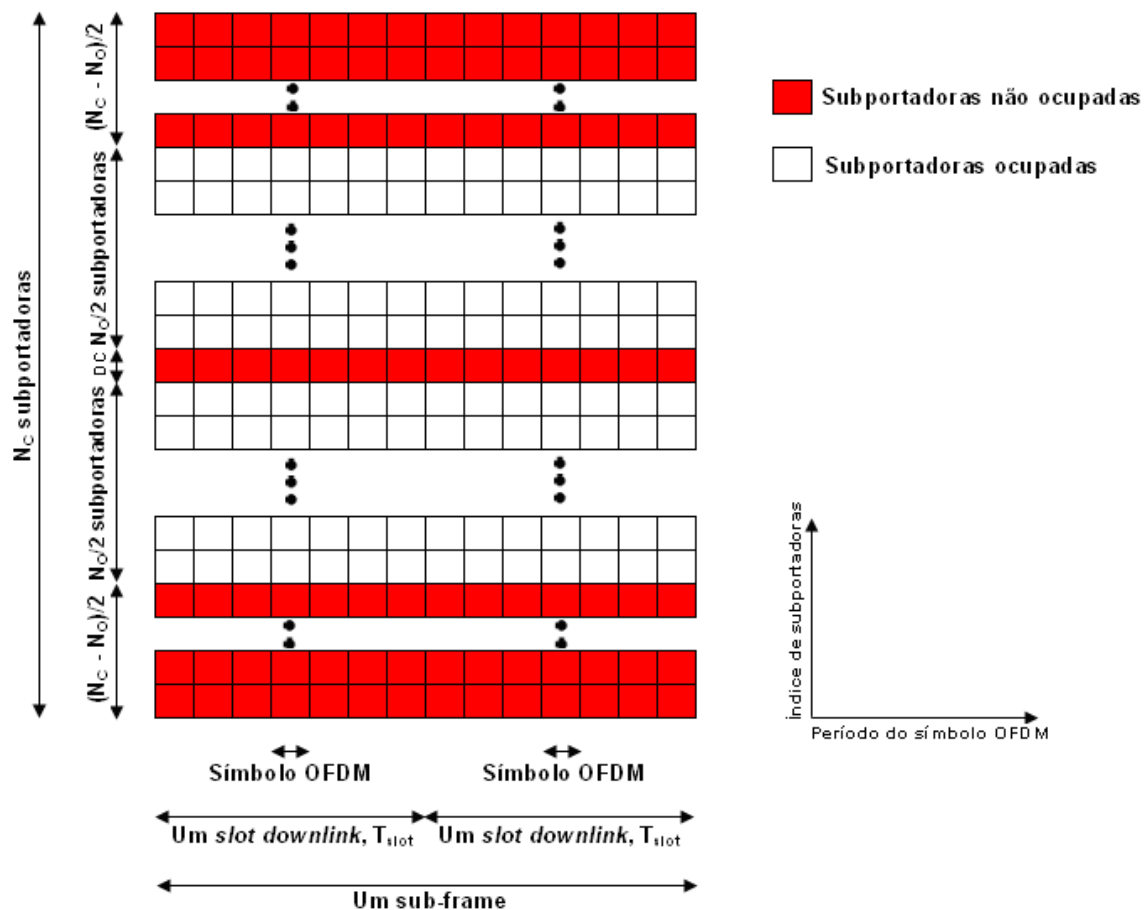
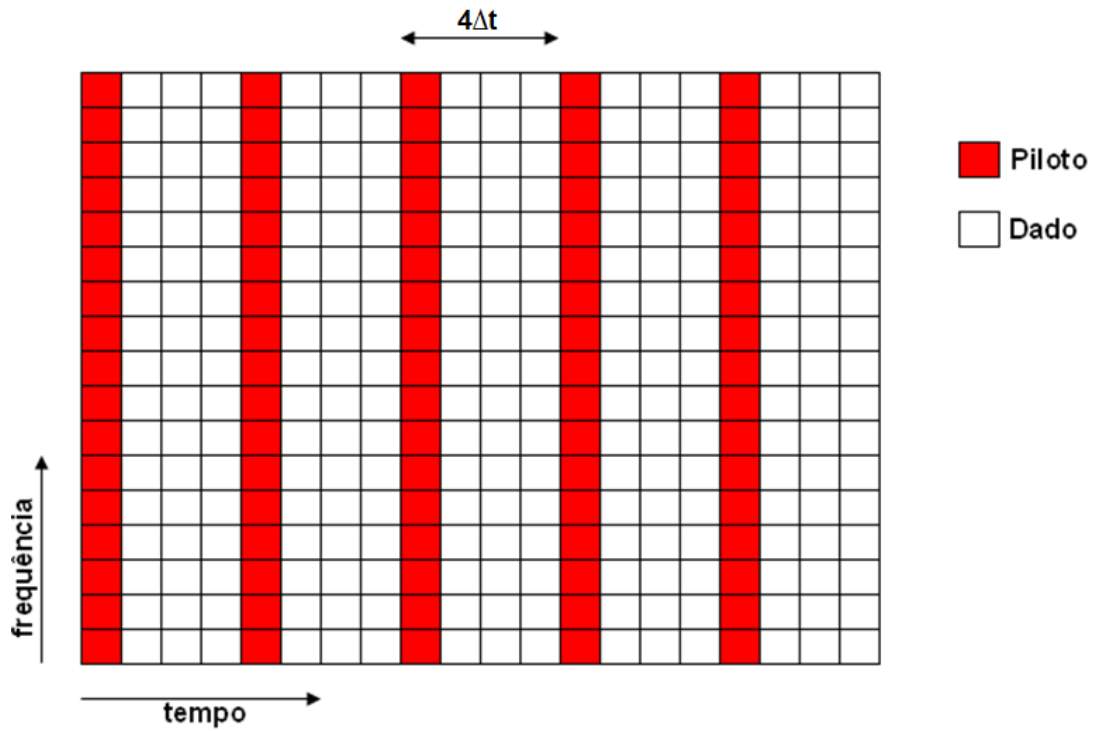


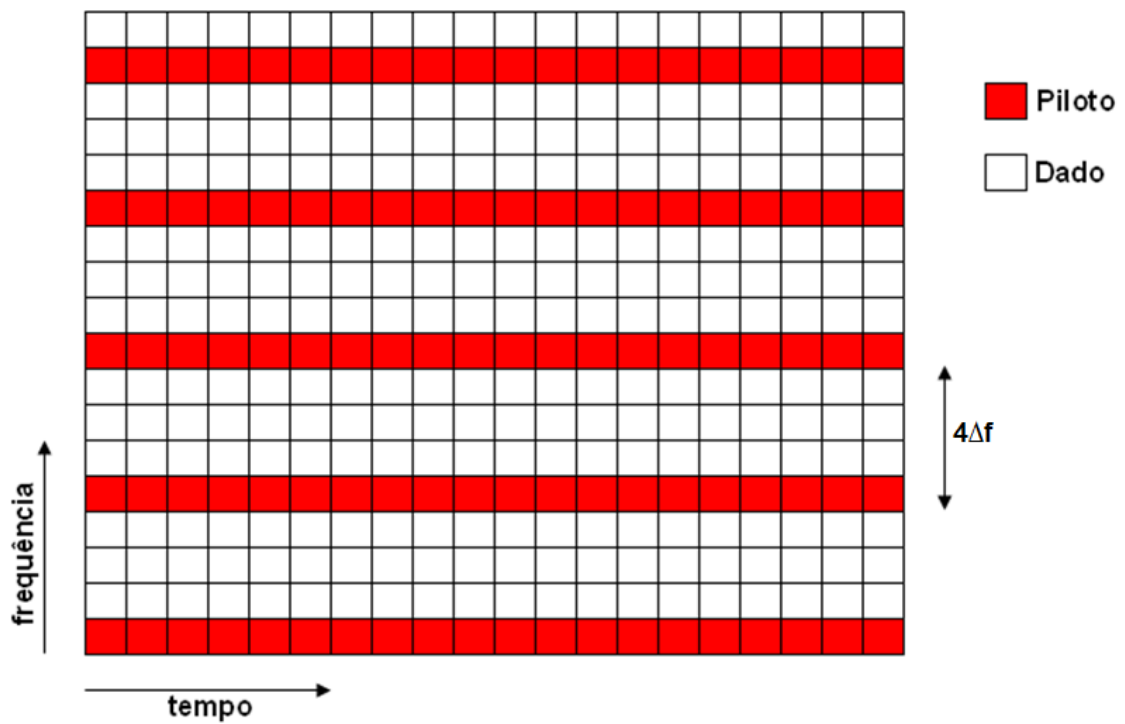
Figura 5.3 - Transmissão OFDM no Domínio do Tempo-Frequência (estrutura genérica do quadro, prefixo cíclico normal).

Existem diferentes possibilidades para a alocação de pilotos no domínio do tempo-frequência de um sistema OFDM (AKRAM, 2007, p. 42). Serão discutidas as três possibilidades, como ilustrado na Figura 5.4.

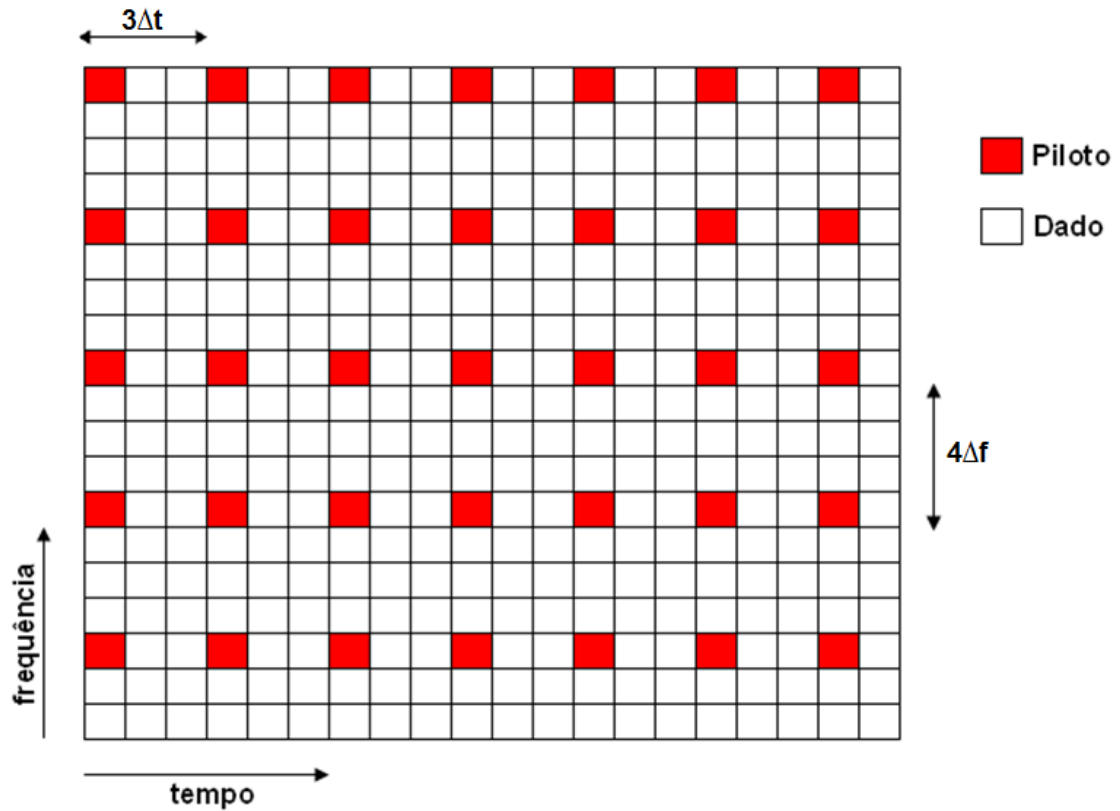
- Um símbolo inteiro OFDM pode ser alocado como piloto, como mostrado na Figura 5.4a (alocação de piloto espaçado no tempo e considerando todas as frequências). Tal alocação será altamente benéfica para a estimação de canal em alta dispersão do sinal no domínio do tempo ($4\Delta t$ - ISI), alto desvanecimento seletivo em frequência, baixa dispersão do sinal no domínio da frequência (Δf nulo - ICI) e baixo desvanecimento seletivo no tempo.
- Os pilotos podem ser transmitidos em subportadoras individuais durante todo o período de transmissão como mostrado na Figura 5.4b (alocação de piloto espaçado na frequência e considerando tempo integral). Essa estratégia é vantajosa para alta dispersão do sinal no domínio da frequência ($4\Delta f$ - ICI), alto desvanecimento seletivo no tempo, baixa dispersão do sinal no domínio do tempo (Δt nulo - ISI) e baixo desvanecimento seletivo em frequência.
- Os pilotos podem ser alocados em intervalos espaçados no tempo e na frequência, conforme ilustrado na Figura 5.4c e na Figura 5.4d (alocação de piloto espaçado no tempo e espaçado na frequência). Dependendo do espaçamento do piloto no tempo-frequência e das propriedades do canal, a estratégia de alocação vai funcionar bem para (Figura 5.4c) alta dispersão do sinal no domínio da frequência ($4\Delta f$ - ICI), alto desvanecimento seletivo no tempo, moderada dispersão do sinal no domínio do tempo ($3\Delta t$ - ISI) e moderado desvanecimento seletivo em frequência ou para (Figura 5.4d) moderada dispersão do sinal no domínio da frequência ($2\Delta f$ - ICI), moderado desvanecimento seletivo no tempo, altíssima dispersão do sinal no domínio do tempo ($6\Delta t$ - ISI) e altíssimo desvanecimento seletivo em frequência.



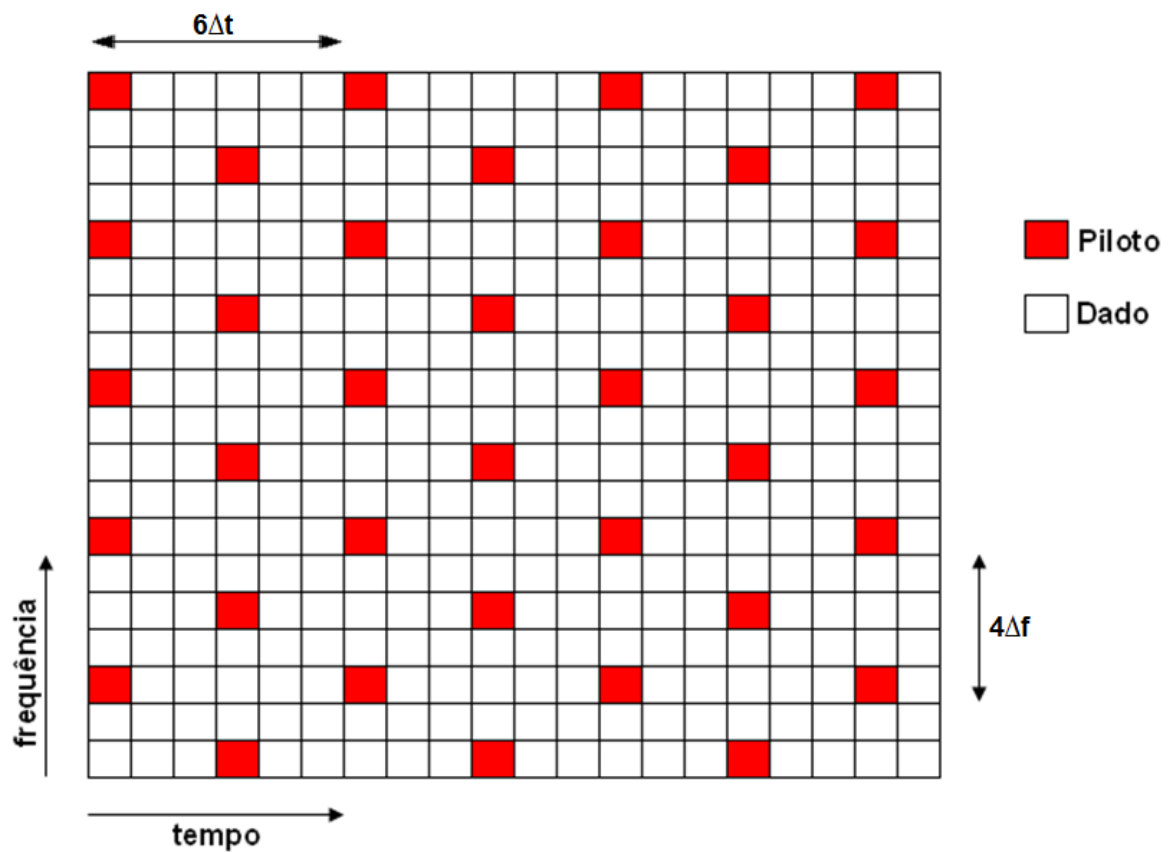
a. Alocação de piloto espaçado no tempo e considerando todas as frequências.



b. Alocação de piloto espaçado na frequência e considerando tempo integral.



c. Alocação de piloto espaçado no tempo e espaçado na frequência.



d. Alocação de piloto espaçado no tempo e espaçado na frequência.

Figura 5.4 - Diferentes possibilidades para alocação de pilotos. (a) Alocação de piloto espaçado no tempo e considerando todas as frequências. (b) Alocação de piloto espaçado na frequência e considerando tempo integral. (c) Alocação de piloto espaçado no tempo e espaçado na frequência. (d) Alocação de piloto espaçado no tempo e espaçado na frequência.

Note-se que o espaçamento na frequência, M_f (Δf), e o espaçamento no tempo, M_t (Δt), entre os pilotos devem ser razoavelmente suficientes para estimar totalmente o canal. Note-se que, para a reconstrução ideal, o espaçamento da frequência M_f entre os pilotos devem cumprir os requisitos do teorema de amostragem 2-D (AKRAM, 2007, p. 43).

(5.3)

$$M_f < \frac{N_c}{2W\tau_{\max}}$$

onde:

W : é a largura de banda do sistema OFDM,

$\tau_{\max}$: é o retardo máximo do canal e

N_c : é o número de subportadoras no sistema OFDM.

Da mesma forma, o espaçamento no tempo M_t (Δt), entre os pilotos devem cumprir (AKRAM, 2007, p. 43):

(5.4)

$$M_t < \frac{W}{2(N_c + W \cdot T_{cp})(f_d + v)}$$

onde:

T_{cp} : é o comprimento do prefixo cíclico,

f_d : é o máximo espalhamento Doppler e

v : é o deslocamento Doppler.

A Figura 5.5 mostra a alocação de piloto em estrutura de quadro genérica EUTRAN e configuração de prefixo normal cíclico. O espaçamento de frequência entre os pilotos é de 6 subportadoras, e o espaçamento de tempo entre os pilotos é

de três dentro de um intervalo de tempo e dois com o símbolo OFDM de entrada, como mostrado na Figura 5.5.

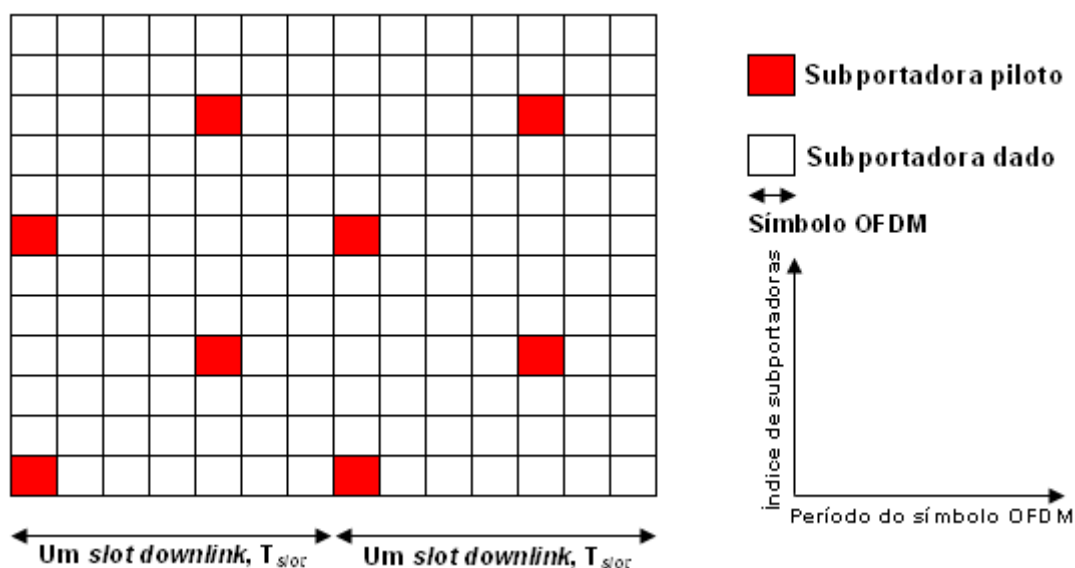


Figura 5.5 - Alocação de piloto em EUTRAN (estrutura de quadro genérica, prefixo cíclico normal).

5.3. OFDM com símbolos piloto para a estimação de canal

A demodulação coerente exige o conhecimento do canal, ou seja, dos coeficientes c_{kl} no modelo de tempo discreto para transmissão OFDM em um canal com desvanecimento. A estrutura bidimensional do sinal OFDM faz uma grade piloto bidimensional especialmente atraente para os canais de medição e estimação.

Um exemplo deste tipo de grade está representado na Figura 5.6. Estes pilotos são normalmente chamados de pilotos espalhados.

Em certas posições no tempo e na frequência, a modulação de símbolos s_{kl} será substituída por símbolos pilotos conhecidos. Nessas situações, o canal pode ser medido.

A Figura 5.6 mostra uma grade retangular com símbolos pilotos a cada três espaçamentos de frequência e a cada quatro intervalos de tempo. A densidade de piloto é, portanto, $1/12$, ou seja, $1/12$ da capacidade total é utilizada para a estimação do canal. Isso diminui não só a taxa de dados, mas, também, a energia disponível por bit (E_b).

A densidade da grade tem que ser compensada com a não coerência do canal, ou seja, para as flutuações tempo-frequência descritas pela função de dispersão.

Para ilustrar isso, através de um exemplo numérico, foi considerada a grade, da Figura 5.6, para um sistema OFDM com espaçamento entre as portadoras $\Delta f = 1/T = 1 \text{ kHz}$ e duração de símbolo $T_S = 1250 \text{ } \mu\text{s}$ (Δt). A cada três frequências, o canal será medido uma vez no tempo $4T_S = 5 \text{ ms}$, isto é, o sinal desconhecido (o canal variante no tempo) é amostrado à uma frequência de amostragem de 200 Hz. Para uma canal, livre de ruído, pode-se concluir a partir do teorema da amostragem que o sinal pode ser recuperado das amostras, se a máxima frequência Doppler $\nu_{\max}$ satisfaz a condição:

(5.5)

$$\nu_{\max} < 100 \text{ Hz}$$

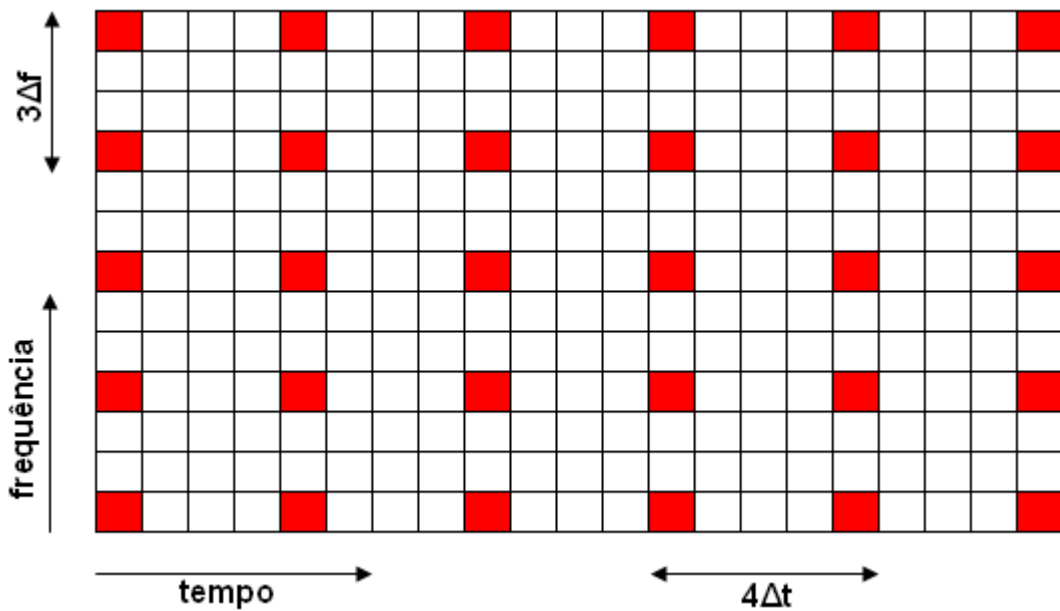


Figura 5.6 - Exemplo de uma grade piloto retangular.

Mais genericamente, para um piloto com espaçamento de $4T_S$, a condição

(5.6)

$$\nu_{\max} T_S < 1/8$$

deve ser cumprida.

Na direção da frequência, o espaçamento da amostra é de 3 kHz. A partir do teorema da amostragem (domínio da frequência), pode-se concluir que o espectro da potência de retardo deve estar dentro de um intervalo de comprimento de 333 μ s. Uma vez que o intervalo de guarda já tem o comprimento 250 μ s, esta condição é automaticamente atendida se puder assumir que todos os ecos se encontram dentro do intervalo de guarda. Agora pode-se começar a interpolação (de acordo com o teorema da amostragem), ou na direção do tempo ou na direção da frequência e, em seguida, calcular os valores interpolados para a outra direção. Interpolações mais simples são possíveis e poderão ser utilizadas, na prática, para um canal mais coerente, por exemplo, a interpolação linear ou aproximação constante. No entanto, para uma mesma variação no tempo e canais seletivos em frequência, estes métodos não são adequados. Para um canal com ruído, a mesma interpolação dada pelo teorema da amostragem não é a melhor escolha porque o ruído não é levado em conta.

Em alguns sistemas, os símbolos pilotos são transmitidos com uma energia mais elevada do que os símbolos de modulação. Nesse caso, uma grade retangular, como mostrada na Figura 5.6 causaria uma maior potência média para cada quatro símbolos OFDM, o que não é desejável por razões de implementação do transmissor. Nesse caso, deve ser escolhida uma grade diagonal.

A Figura 5.7 - mostra como uma grade diagonal é usada para o sistema DVB-T.

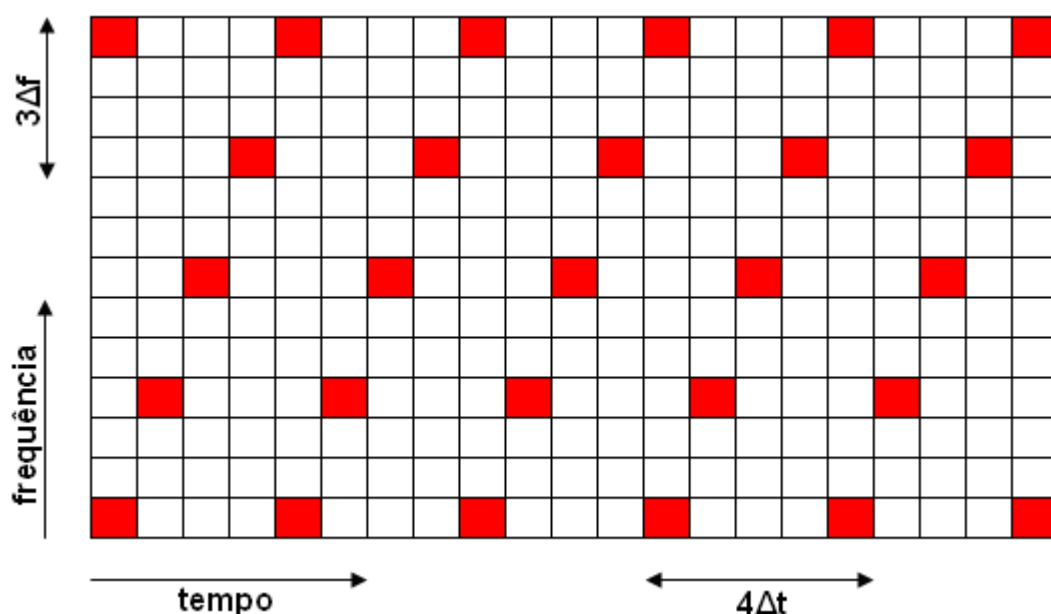


Figura 5.7 - Exemplo de uma grade piloto diagonal.

5.4. Conclusão

Foram apresentados um Método de Estimação de Canal e as diferentes possibilidades para alocação de pilotos, tais como, alocação de piloto espaçado no tempo e considerando todas as frequências, alocação de piloto espaçado na frequência e considerando tempo integral, alocação de piloto espaçado no tempo e espaçado na frequência.

Para finalizar, foi tratado OFDM com símbolos piloto para a estimação de canal.

6. RESULTADOS DA SIMULAÇÃO DA PLATAFORMA OFDM NO MATLAB (SIMULINK)

6.1. Introdução

Este capítulo tem, como objetivo, apresentar os resultados das simulações realizadas com a plataforma OFDM, apresentada no Apêndice A.

A Seção 6.2 apresenta os gráficos BER vs. E_b/N_0 , com a melhor curva de interpolação. A Seção 6.3 apresenta os gráficos Vazão vs. E_b/N_0 . A Seção 6.4 apresenta os resultados BER e Vazão vs. E_b/N_0 . A Seção 6.5 apresenta os gráficos BER vs. Pilotos. A Seção 6.6 apresenta os gráficos Vazão vs. Pilotos. A Seção 6.7 apresenta os resultados BER e Vazão vs. Pilotos. A Seção 6.8. apresenta os resultados Vazão Efetiva vs. E_b/N_0 . A Seção 6.9 apresenta o gráfico BER vs. E_b/N_0 com 16 Pilotos e considerando aproveitamento da redundância dos bits de prefixo cíclico. A Seção 6.10 apresenta proposições de testes e simulações que podem ser realizadas com a plataforma OFDM. A Seção 6.11 apresenta a conclusão.

Os gráficos BER vs. E_b/N_0 , com a melhor curva de interpolação, foram obtidos a partir das simulações feitas na plataforma OFDM apresentada no capítulo anterior. Todos os programas referentes à plotagem dos demais gráficos, encontram-se disponíveis no Apêndice C.

6.2. BER vs. E_b/N_0 com a melhor curva de interpolação

As Figuras 6.1 e 6.2 e as Tabelas 6.1 e 6.2 apresentam, respectivamente, as curvas e os valores das Taxas de Erros (BER) em função do E_b/N_0 , considerando as seguintes quantidades de pilotos: 2, 3, 4, 5, 6, 7, 11, 13, 16, 21, 26, 31, 51, 61, 76, 101, 151.

Foram considerados os seguintes valores em decibel (dB) para o E_b/N_0 : 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15.

Foram traçados dois gráficos BER vs. E_b/N_0 , com a melhor curva de interpolação: um com a BER em escala logarítmica e o outro com a BER em escala linear.

Ao analisar a curva e os dados de cada piloto separadamente, conclui-se que quanto maior o E_b/N_0 menor será a Taxa de Erro. A curva que mostra esta redução gradativa da Taxa de Erro em função do E_b/N_0 é uma exponencial decrescente.

Observa-se que a Taxa de Erro decresce lentamente para os seguintes pilotos: 2, 3, 4, 5, 6 e 7.

Observa-se que a Taxa de Erro decresce um pouco mais rápido para os seguintes pilotos: 51, 61, 76, 101 e 151.

Observa-se que a Taxa de Erro decresce com uma maior rapidez para os seguintes pilotos: 11, 13, 16, 21, 26 e 31.

Ao analisar a curva e os dados de todos os pilotos, conclui-se que a Taxa de Erro decresce, considerando a seguinte sequência de pilotos: 5, 3, 2, 4, 6, 7, 61, 101, 76, 51, 151, 16, 26, 11, 13, 31, 21.

A partir disso, conclui-se que para trabalhar com uma menor Taxa de Erro, o desenvolvedor de um Projeto de Comunicações com Sistemas Móveis, utilizando a plataforma OFDM, deverá considerar as seguintes quantidades de pilotos 16, 26, 11, 13, 31 ou 21, sendo que a opção com a menor Taxa de Erro possível é a de 21 pilotos.

Sendo o foco da investigação, a influência da densidade de pilotos no desempenho do sistema OFDM, conclui-se que o desempenho da BER melhora consistentemente com a inserção de mais pilotos, mas a vazão reduz, devido à inserção de pilotos.

Para uma pequena quantidade de pilotos inseridos (2, 3, 4, 5, 6 e 7), a Taxa de Erro é elevada. Aumentado-se a quantidade de pilotos inseridos (11, 13, 16, 21, 26 e 31), a Taxa de Erro diminui, porém, resultados da simulação mostram também que, se a quantidade de pilotos inseridos (51, 61, 76, 101 e 151) aumentar consideravelmente, o desempenho do sistema degrada, atingindo valores intermediários da BER. Fica como sugestão para trabalhos futuros, conforme escrito no item 6.10, a mudança do estimador do canal, para análise do desempenho do

sistema OFDM, quando a quantidade de pilotos inseridos (51, 61, 76, 101 e 151) aumentar consideravelmente.

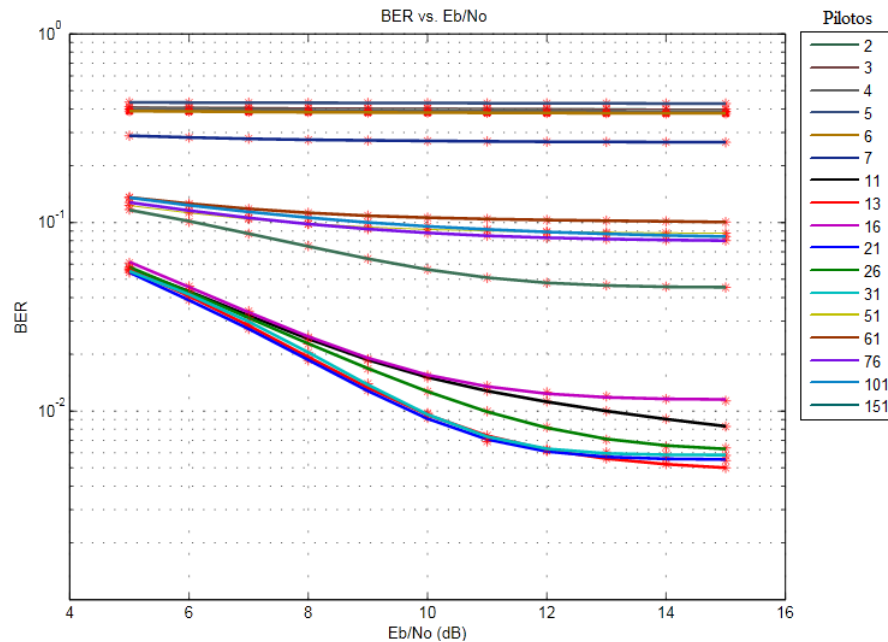


Figura 6.1 - BER vs. Eb/No com a melhor curva de interpolação (Pilotos 5, 3, 2, 4, 6, 7, 61, 101, 76, 51, 151, 16, 26, 11, 13, 31, 21). Ordem decrescente da taxa de erro. Escala logarítmica.

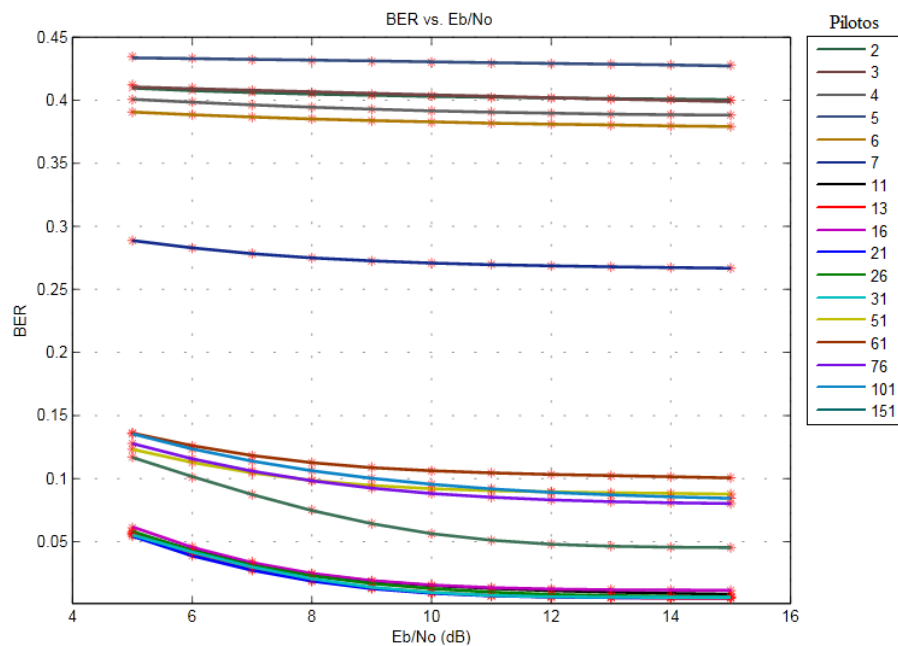


Figura 6.2 - BER vs. Eb/No com a melhor curva de interpolação (Pilotos 5, 3, 2, 4, 6, 7, 61, 101, 76, 51, 151, 16, 26, 11, 13, 31, 21). Ordem decrescente da taxa de erro. Escala linear.

Tabela 6.1 - BER vs. Eb/No com a melhor curva de interpolação (Pilotos 2, 3, 4, 5, 6, 7, 11, 13, 16, 21, 26, 31, 51, 61, 76, 101, 151).

Símbolos Pilotos	Símbolos Dados	Pilotos + Dados	5	6	7	8	9	10	11	12	13	14	15	Eb/No
2	300	302	0.409751	0.407664	0.406128	0.404926	0.403952	0.403148	0.402471	0.40189	0.401377	0.400909	0.400465	Taxa de Erro
3	300	303	0.410601	0.409288	0.408004	0.40675	0.405528	0.40434	0.403188	0.402073	0.400996	0.399959	0.398963	Taxa de Erro
4	300	304	0.400808	0.398488	0.396415	0.394591	0.393014	0.391682	0.390585	0.389713	0.389054	0.388592	0.38831	Taxa de Erro
5	300	305	0.433751	0.433108	0.432466	0.431824	0.431182	0.430541	0.429899	0.429257	0.428615	0.427974	0.427332	Taxa de Erro
6	300	306	0.390701	0.388559	0.386756	0.385228	0.383934	0.382835	0.381899	0.381093	0.380386	0.379749	0.379153	Taxa de Erro
7	300	307	0.288716	0.282891	0.278412	0.275078	0.27266	0.270934	0.269693	0.268768	0.268034	0.267403	0.266824	Taxa de Erro
11	300	311	0.057235	0.043201	0.032201	0.024166	0.018679	0.015109	0.012795	0.011208	0.010023	0.009079	0.008313	Taxa de Erro
13	300	313	0.056547	0.040769	0.028394	0.019373	0.01331	0.009562	0.007414	0.006239	0.005595	0.005227	0.005008	Taxa de Erro
16	300	316	0.061718	0.045492	0.033399	0.024842	0.01913	0.015561	0.01349	0.012384	0.011847	0.011612	0.01152	Taxa de Erro
21	300	321	0.054406	0.038794	0.027302	0.018696	0.012779	0.009103	0.007076	0.006107	0.005715	0.005585	0.005551	Taxa de Erro
26	300	326	0.058103	0.042838	0.031244	0.022803	0.016831	0.012699	0.009927	0.008164	0.007124	0.006565	0.006296	Taxa de Erro
31	300	331	0.0553	0.041616	0.029824	0.020479	0.013799	0.009593	0.007326	0.006318	0.005963	0.005869	0.005852	Taxa de Erro
51	300	351	0.123187	0.112722	0.104555	0.098599	0.094556	0.091993	0.090436	0.089471	0.0888	0.088251	0.087742	Taxa de Erro
61	300	361	0.136074	0.126048	0.118248	0.112576	0.108709	0.106187	0.104528	0.103337	0.102356	0.101449	0.100564	Taxa de Erro
76	300	376	0.127794	0.115727	0.105934	0.098249	0.092425	0.088171	0.085172	0.083124	0.081753	0.080834	0.080196	Taxa de Erro
101	300	401	0.135504	0.123459	0.113838	0.106204	0.100196	0.095514	0.091902	0.089144	0.087059	0.085498	0.084335	Taxa de Erro
151	300	451	0.116704	0.101663	0.087422	0.074742	0.064271	0.056385	0.051072	0.047939	0.046347	0.045644	0.045343	Taxa de Erro

Tabela 6.2 - BER vs. Eb/No com a melhor curva de interpolação (Pilotos 5, 3, 2, 4, 6, 7, 61, 101, 76, 51, 151, 16, 26, 11, 13, 31, 21). Ordem decrescente da taxa de erro.

Simbolos Pilotos	Simbolos Dados	Pilotos + Dados	5	6	7	8	9	10	11	12	13	14	15	Eb/No
5	300	305	0.433751	0.433108	0.432466	0.431824	0.431182	0.430541	0.429899	0.429257	0.428615	0.427974	0.427332	Taxa de Erro
3	300	303	0.410601	0.409288	0.408004	0.40675	0.405528	0.40434	0.403188	0.402073	0.400996	0.399959	0.398963	Taxa de Erro
2	300	302	0.409751	0.407664	0.406128	0.404926	0.403952	0.403148	0.402471	0.40189	0.401377	0.400909	0.400465	Taxa de Erro
4	300	304	0.400808	0.398488	0.396415	0.394591	0.393014	0.391682	0.390585	0.389713	0.389054	0.388592	0.38831	Taxa de Erro
6	300	306	0.390701	0.388559	0.386756	0.385228	0.383934	0.382835	0.381899	0.381093	0.380386	0.379749	0.379153	Taxa de Erro
7	300	307	0.288716	0.282891	0.278412	0.275078	0.27266	0.270934	0.269693	0.268768	0.268034	0.267403	0.266824	Taxa de Erro
61	300	361	0.136074	0.126048	0.118248	0.112576	0.108709	0.106187	0.104528	0.103337	0.102356	0.101449	0.100564	Taxa de Erro
101	300	401	0.135504	0.123459	0.113838	0.106204	0.100196	0.095514	0.091902	0.089144	0.087059	0.085498	0.084335	Taxa de Erro
76	300	376	0.127794	0.115727	0.105934	0.098249	0.092425	0.088171	0.085172	0.083124	0.081753	0.080834	0.080196	Taxa de Erro
51	300	351	0.123187	0.112722	0.104555	0.098599	0.094556	0.091993	0.090436	0.089471	0.0888	0.088251	0.087742	Taxa de Erro
151	300	451	0.116704	0.101663	0.087422	0.074742	0.064271	0.056385	0.051072	0.047939	0.046347	0.045644	0.045343	Taxa de Erro
16	300	316	0.061718	0.045492	0.033399	0.024842	0.01913	0.015561	0.01349	0.012384	0.011847	0.011612	0.011152	Taxa de Erro
26	300	326	0.058103	0.042838	0.031244	0.022803	0.016831	0.012699	0.009927	0.008164	0.007124	0.006565	0.006296	Taxa de Erro
11	300	311	0.057235	0.043201	0.032201	0.024166	0.018679	0.015109	0.012795	0.011208	0.010023	0.009079	0.008313	Taxa de Erro
13	300	313	0.056547	0.040769	0.028394	0.019373	0.01331	0.009562	0.007414	0.006239	0.005595	0.005227	0.005008	Taxa de Erro
31	300	331	0.0553	0.041616	0.029824	0.020479	0.013799	0.009593	0.007326	0.006318	0.005963	0.005869	0.005852	Taxa de Erro
21	300	321	0.054406	0.038794	0.027302	0.018696	0.012779	0.009103	0.007076	0.006107	0.005715	0.005585	0.005551	Taxa de Erro

Para uma melhor visualização das curvas, as Figuras 6.3 e 6.4 apresentam, respectivamente em escala logarítmica e linear, as curvas BER vs. E_b/N_0 with Best Curve Fit, considerando os seguintes pilotos: 2, 3, 4, 5 e 6.

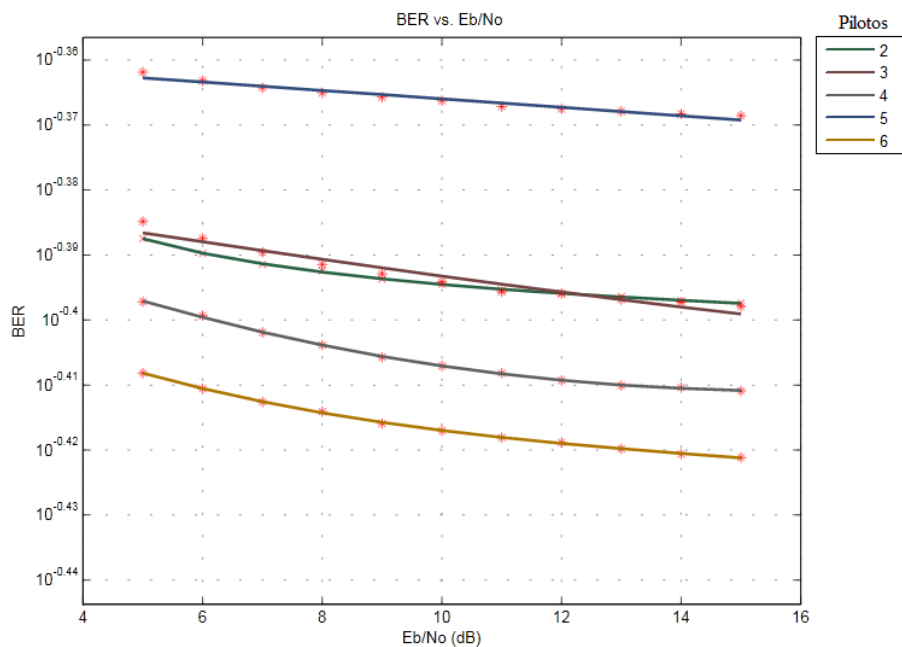


Figura 6.3 - BER vs. E_b/N_0 com a melhor curva de interpolação (Pilotos 5, 3, 2, 4, 6). Ordem decrescente da taxa de erro. Escala logarítmica.

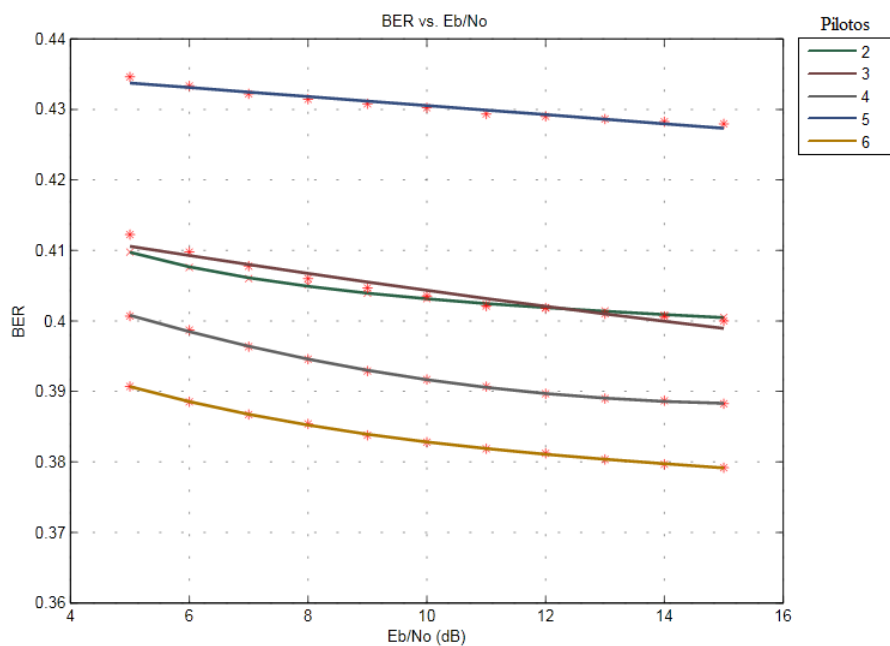


Figura 6.4 - BER vs. E_b/N_0 com a melhor curva de interpolação (Pilotos 5, 3, 2, 4, 6). Ordem decrescente da taxa de erro. Escala linear.

Para uma melhor visualização da curva, as Figuras 6.5 e 6.6 apresentam, respectivamente, em escala logarítmica e linear, a curva BER vs. E_b/N_0 with Best Curve Fit, considerando o piloto 7.

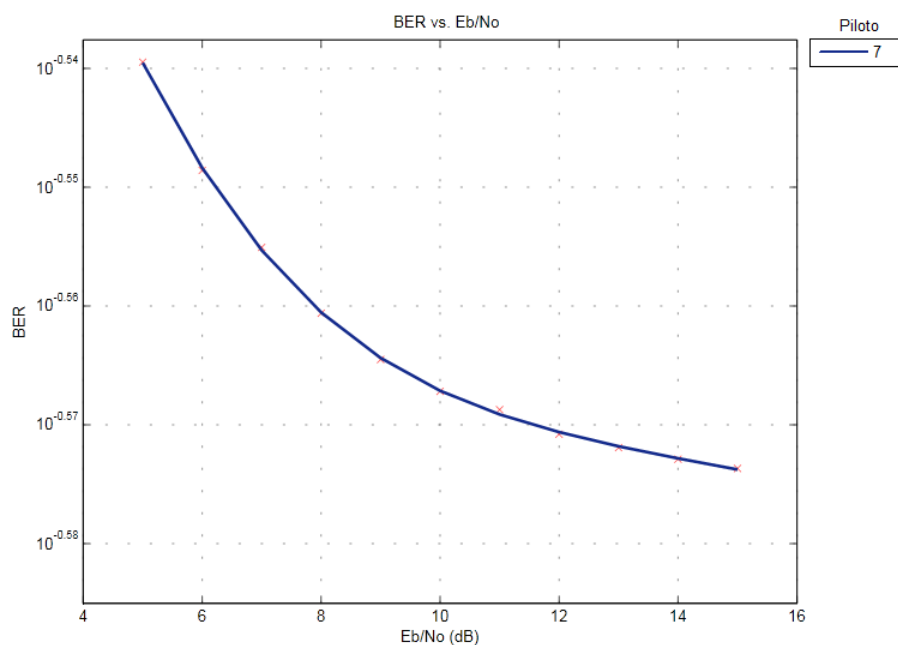


Figura 6.5 - BER vs. E_b/N_0 com a melhor curva de interpolação (Piloto 7). Escala logarítmica.

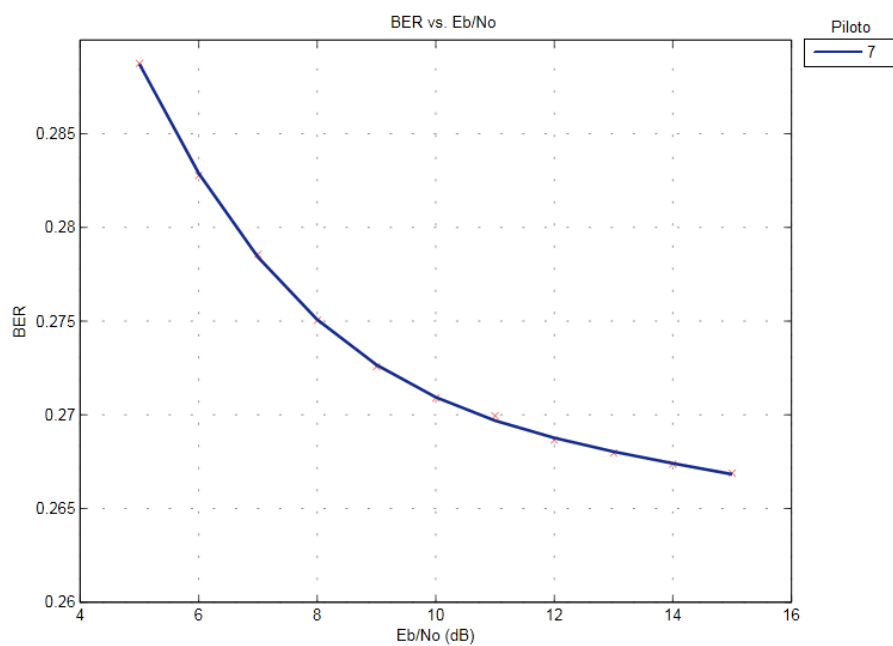


Figura 6.6 - BER vs. E_b/N_0 com a melhor curva de interpolação (Piloto 7). Escala linear.

Para uma melhor visualização das curvas, as Figuras 6.7 e 6.8 apresentam, respectivamente, em escala logarítmica e linear, as curvas BER vs. E_b/N_0 with Best Curve Fit, considerando os seguintes pilotos: 11, 13, 16, 21, 26, 31.

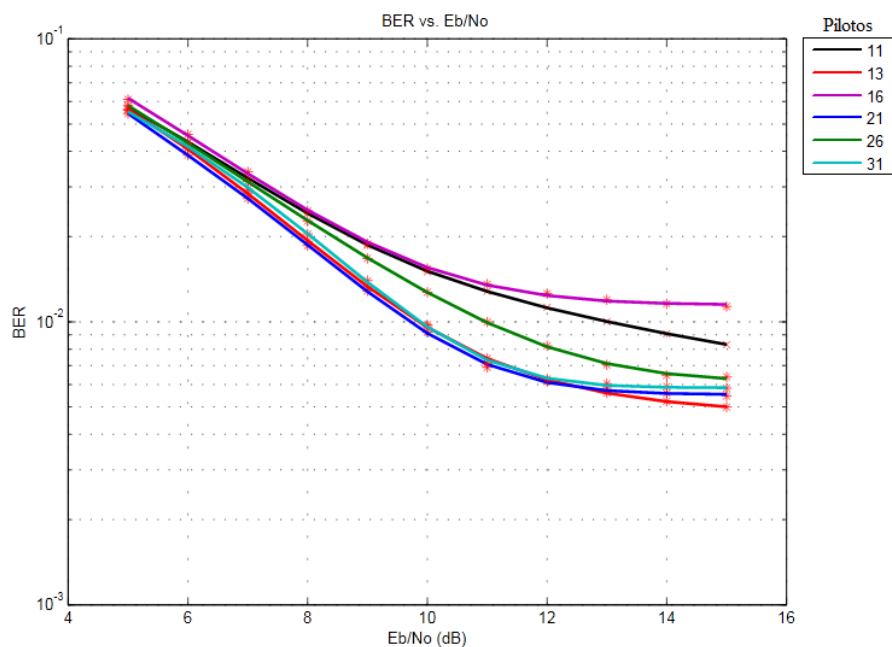


Figura 6.7 - BER vs. E_b/N_0 com a melhor curva de interpolação (Pilotos 16, 26, 11, 13, 31, 21). Ordem decrescente da taxa de erro. Escala logarítmica.

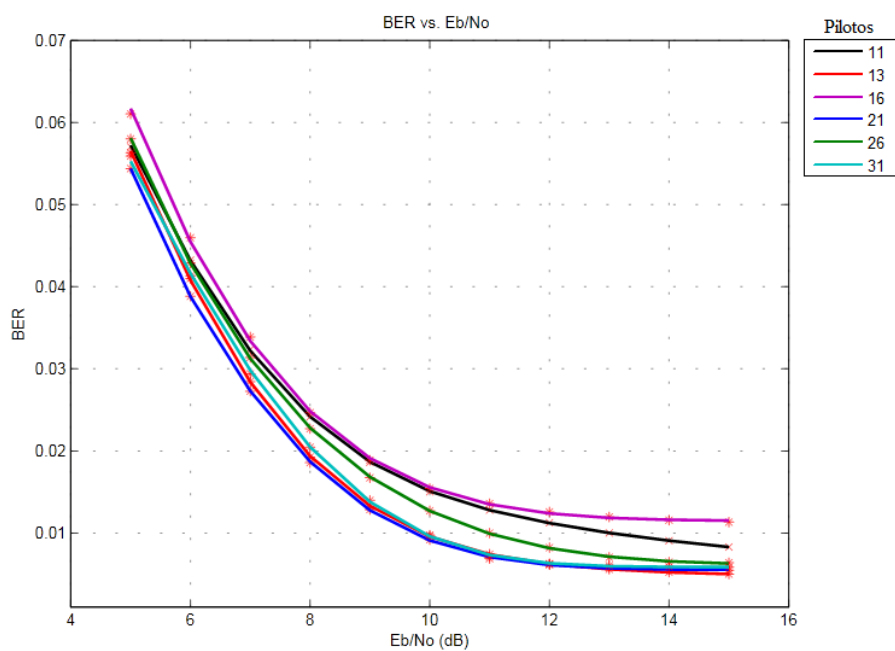


Figura 6.8 - BER vs. E_b/N_0 com a melhor curva de interpolação (Pilotos 16, 26, 11, 13, 31, 21). Ordem decrescente da taxa de erro. Escala linear.

Para uma melhor visualização das curvas, as Figuras 6.9 e 6.10 apresentam, respectivamente, em escala logarítmica e linear, as curvas BER vs. E_b/N_0 with Best Curve Fit, considerando os seguintes pilotos: 51, 61, 76, 101, 151.

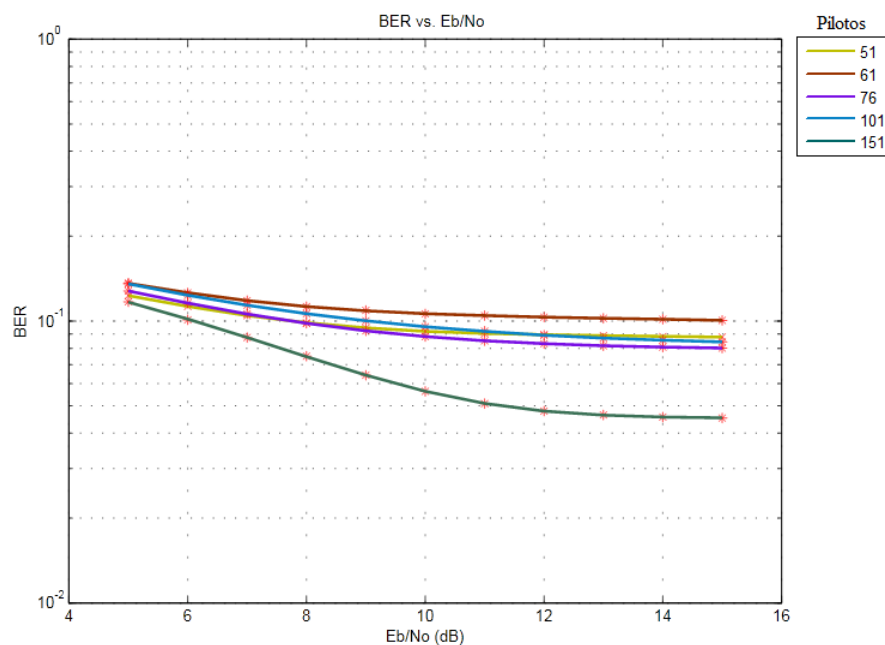


Figura 6.9 - BER vs. E_b/N_0 com a melhor curva de interpolação (Pilotos 61, 101, 76, 51, 151). Ordem decrescente da taxa de erro. Escala logarítmica.

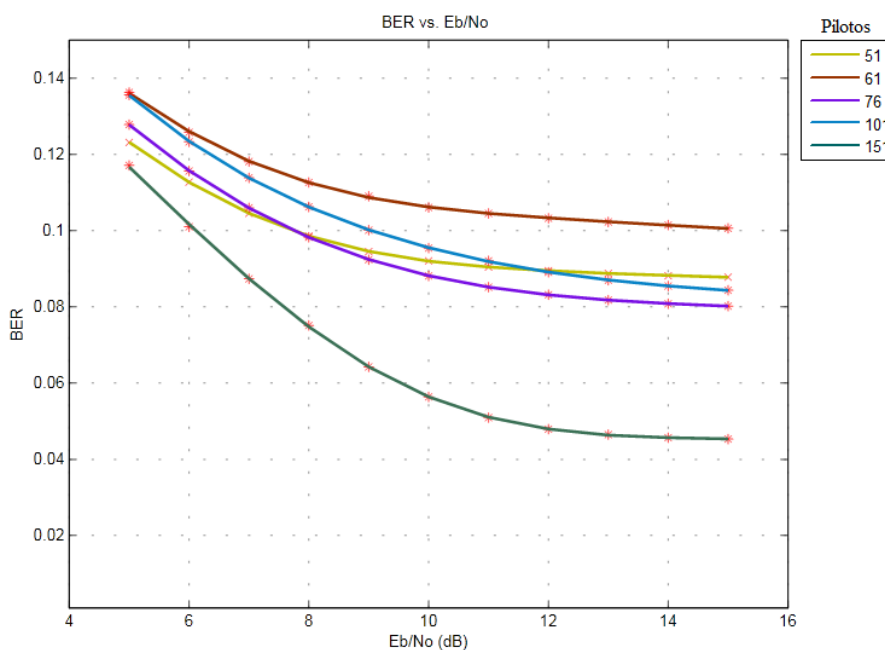


Figura 6.10 - BER vs. E_b/N_0 com a melhor curva de interpolação (Pilotos 61, 101, 76, 51, 151). Ordem decrescente da taxa de erro. Escala linear.

6.3. Vazão vs. Eb/No

As Figuras 6.11 e 6.12 e as Tabelas 6.3 e 6.4 apresentam, respectivamente, as curvas e os valores de Vazão em função do Eb/No, considerando as seguintes quantidades de pilotos: 2, 3, 4, 5, 6, 7, 11, 13, 16, 21, 26, 31, 51, 61, 76, 101, 151.

Foram considerados os seguintes valores, em decibel (dB), para o Eb/No: 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15.

Pode-se definir a vazão através dos seguintes modos:

$$\text{a) Vazão} = \frac{(1 - \text{Taxa de Erro}) \times \text{Bits Dados}}{(\text{Bits Dados} + \text{Bits Pilotos})} \quad [\text{Vazão}] = \%$$

$$\text{b) Vazão} = \frac{(1 - \text{Taxa de Erro}) \times \text{Bits Dados}}{(\text{Bits Dados} + \text{Bits Pilotos})} \times (\text{Bits Dados} + \text{Bits Pilotos})$$

Simplificando, tem-se:

$$\text{Vazão} = (1 - \text{Taxa de Erro}) \times \text{Bits Dados} \quad [\text{Vazão}] = \text{bits/portadora}$$

$$\text{c) Vazão Efetiva} = \frac{[(1 - \text{Taxa de Erro}) \times (\text{Bits Dados} \times \text{CM}_I)]}{[(\text{Bits Dados} \times \text{CM}_I) + \text{Bits Pilotos}]} \quad [\text{Vazão}] = \%$$

onde,

CM_I = Índice de Codificação e Modulação = Número de bits por símbolo = $(1/2) \times 4$ = (Codificador Turbo) * Modulador.

$$\text{d) Vazão Efetiva} = \frac{[(1 - \text{Taxa de Erro}) \times (\text{Bits Dados} \times \text{CM}_I)]}{[(\text{Bits Dados} \times \text{CM}_I) + \text{Bits Pilotos}]}$$

$$\times [(\text{Bits Dados} \times \text{CM}_I) + \text{Bits Pilotos}]$$

Simplificando, tem-se:

$$\text{Vazão Efetiva} = [(1 - \text{Taxa de Erro}) \times (\text{Bits Dados} \times \text{CM}_I)] \quad [\text{Vazão}] = \text{bits/portadora}$$

Nos gráficos e nas tabelas foi considerado a definição da alternativa a.

Foram traçados dois gráficos Vazão vs. Eb/No: um com a Vazão em escala logarítmica e o outro, com a Vazão em escala linear.

Ao analisar a curva e os dados de cada piloto separadamente, conclui-se que quanto maior o Eb/No maior será a Vazão. As Figuras 6.11 e 6.12 mostram as curvas com um crescimento gradativo da Vazão em função do Eb/No.

Observa-se que a Vazão cresce lentamente para os seguintes pilotos 2, 3, 4, 5, 6.

Observa-se que a Vazão cresce um pouco mais rápido para os seguintes pilotos 7, 51, 61, 76, 101, 151.

Observa-se que a Vazão cresce com uma maior rapidez para os seguintes pilotos 11, 13, 16, 21, 26, 31.

Analisando as Figuras 6.11 e 6.12 e os valores nas Tabelas 6.3 e 6.4, pode-se concluir, também, que a partir de um Eb/No igual a 13 dB, o valor da Vazão fica praticamente constante, portanto as curvas assumem o formato de uma reta horizontal.

Ao analisar a curva e os dados de todos os pilotos, conclui-se que a Vazão decresce considerando a seguinte sequência de pilotos: 11, 13, 16, 21, 26, 31, 51, 61, 76, 7, 101, 6, 4, 151, 2, 3, 5. Deste resultado, conclui-se que para trabalhar com uma maior Vazão, o desenvolvedor de um Projeto de Comunicações com Sistemas Móveis, utilizando a plataforma OFDM, deverá considerar as seguintes quantidades de pilotos: 11, 13, 16, 21, 26 ou 31, sendo de 11 pilotos, a opção com a maior Vazão possível.

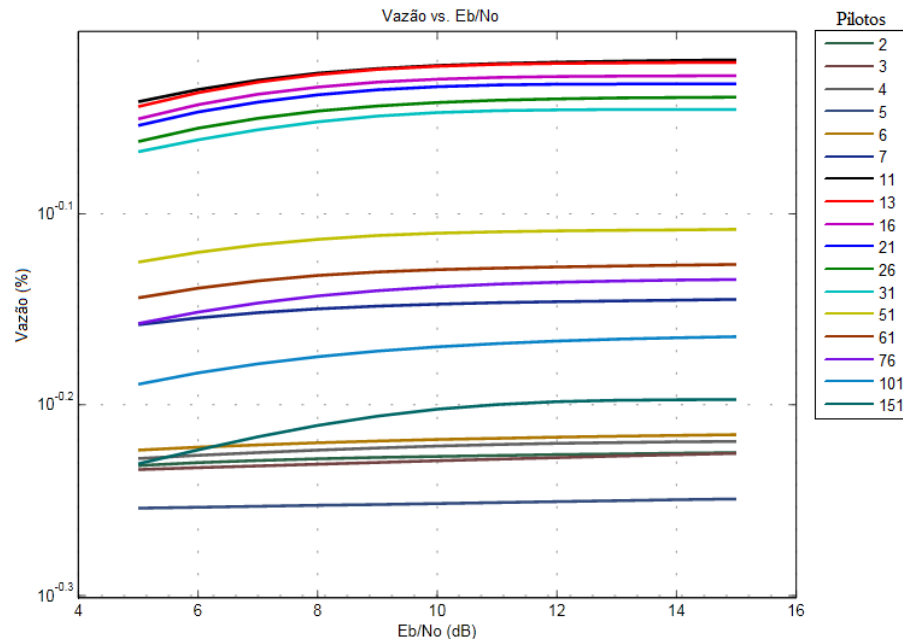


Figura 6.11 - Vazão vs. Eb/No (Pilotos 11, 13, 16, 21, 26, 31, 51, 61, 76, 7, 101, 6, 4, 151, 2, 3, 5). Ordem Decrescente da Vazão. Escala logarítmica.

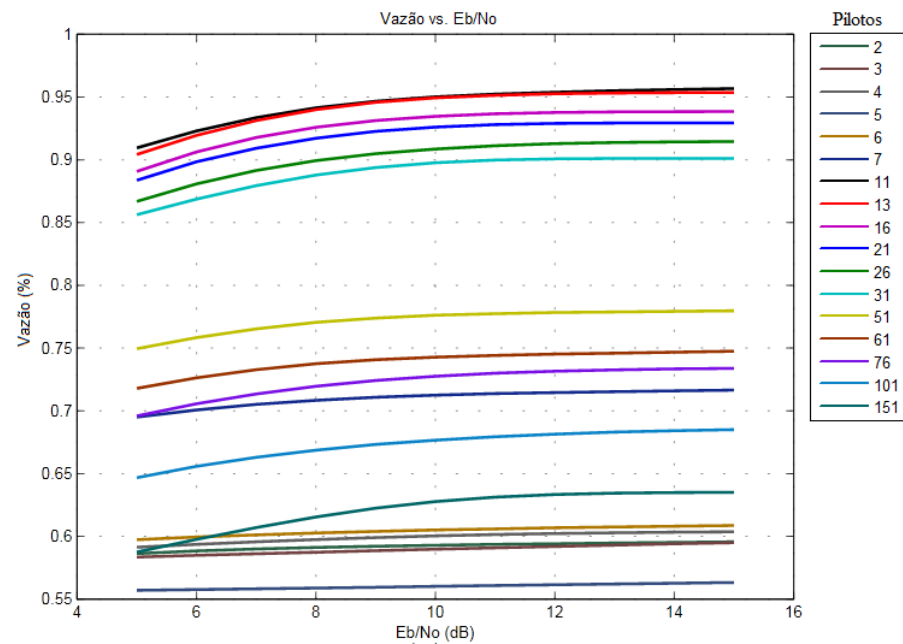


Figura 6.12 - Vazão vs. Eb/No (Pilotos 11, 13, 16, 21, 26, 31, 51, 61, 76, 7, 101, 6, 4, 151, 2, 3, 5). Ordem Decrescente da Vazão. Escala linear.

Tabela 6.3 - Vazão vs. Eb/No (Pilotos 2, 3, 4, 5, 6, 7, 11, 13, 16, 21, 26, 31, 51, 61, 76, 101, 151).

Símbolos Pilotos	Símbolos Dados	Pilotos + Dados	5	6	7	8	9	10	11	12	13	14	15	Eb/No
2	300	302	0.58634	0.588413	0.589939	0.591134	0.5921	0.592899	0.593572	0.594149	0.594659	0.595123	0.595655	Vazão
3	300	303	0.583563	0.584863	0.586135	0.587376	0.588586	0.589762	0.590903	0.592007	0.593073	0.5941	0.595086	Vazão
4	300	304	0.591308	0.593598	0.595643	0.597444	0.598999	0.600314	0.601397	0.602257	0.602907	0.603363	0.603642	Vazão
5	300	305	0.556967	0.557598	0.55823	0.558861	0.559493	0.560124	0.560755	0.561387	0.562018	0.562649	0.56328	Vazão
6	300	306	0.597352	0.599452	0.60122	0.602717	0.603987	0.605063	0.605981	0.606771	0.607465	0.608089	0.608673	Vazão
7	300	307	0.695065	0.700758	0.705135	0.708393	0.710755	0.712443	0.713655	0.714559	0.715277	0.715893	0.716459	Vazão
11	300	311	0.90942	0.922957	0.933568	0.941319	0.946612	0.950055	0.952288	0.953818	0.954962	0.955872	0.956612	Vazão
13	300	313	0.904268	0.91939	0.931252	0.939898	0.94571	0.949301	0.95136	0.952487	0.953104	0.953457	0.953666	Vazão
16	300	316	0.890774	0.906178	0.917659	0.925783	0.931205	0.934594	0.93656	0.93761	0.93812	0.938343	0.93843	Vazão
21	300	321	0.883733	0.898324	0.909064	0.917107	0.922637	0.926072	0.927966	0.928872	0.929238	0.92936	0.929392	Vazão
26	300	326	0.866777	0.880824	0.891493	0.899261	0.904757	0.90856	0.91111	0.912733	0.91369	0.914204	0.914452	Vazão
31	300	331	0.856224	0.868626	0.879313	0.887783	0.893838	0.89765	0.899704	0.900618	0.90094	0.901025	0.901041	Vazão
51	300	351	0.749413	0.758357	0.765337	0.770428	0.773883	0.776074	0.777405	0.77823	0.778803	0.779272	0.779708	Vazão
61	300	361	0.717944	0.726276	0.732758	0.737472	0.740685	0.742781	0.74416	0.745149	0.745965	0.746718	0.747454	Vazão
76	300	376	0.695909	0.705537	0.71335	0.719482	0.724129	0.727523	0.729916	0.73155	0.732644	0.733377	0.733886	Vazão
101	300	401	0.646755	0.655766	0.662964	0.668676	0.67317	0.676673	0.679375	0.681439	0.682998	0.684166	0.685036	Vazão
151	300	451	0.587559	0.597564	0.607036	0.615471	0.622436	0.627682	0.631216	0.6333	0.634359	0.634826	0.635027	Vazão

Tabela 6.4 - Vazão vs. Eb/No (Pilotos 11, 13, 16, 21, 26, 31, 51, 61, 76, 7, 101, 6, 4, 151, 2, 3, 5). Ordem Decrescente da Vazão.

Símbolos Pilotos	Símbolos Dados	Pilotos + Dados	5	6	7	8	9	10	11	12	13	14	15	Eb/No
11	300	311	0.90942	0.922957	0.933568	0.941319	0.946612	0.950055	0.952288	0.953818	0.954962	0.955872	0.956612	Vazão
13	300	313	0.904268	0.91939	0.931252	0.939898	0.94571	0.949301	0.95136	0.952487	0.953104	0.953457	0.953666	Vazão
16	300	316	0.890774	0.906178	0.917659	0.925783	0.931205	0.934594	0.93656	0.93761	0.93812	0.938343	0.93843	Vazão
21	300	321	0.883733	0.898324	0.909064	0.917107	0.922637	0.926072	0.927966	0.928872	0.929238	0.92936	0.929392	Vazão
26	300	326	0.866777	0.880824	0.891493	0.899261	0.904757	0.90856	0.91111	0.912733	0.91369	0.914204	0.914452	Vazão
31	300	331	0.856224	0.868626	0.879313	0.887783	0.893838	0.89765	0.899704	0.900618	0.90094	0.901025	0.901041	Vazão
51	300	351	0.749413	0.758357	0.765337	0.770428	0.773883	0.776074	0.777405	0.77823	0.778803	0.779272	0.779708	Vazão
61	300	361	0.717944	0.726276	0.732758	0.737472	0.740685	0.742781	0.74416	0.745149	0.745965	0.746718	0.747454	Vazão
76	300	376	0.695909	0.705537	0.71335	0.719482	0.724129	0.727523	0.729916	0.73155	0.732644	0.733377	0.733886	Vazão
7	300	307	0.695065	0.700758	0.705135	0.708393	0.710755	0.712443	0.713655	0.714559	0.715277	0.715893	0.716459	Vazão
101	300	401	0.646755	0.655766	0.662964	0.668676	0.67317	0.676673	0.679375	0.681439	0.682998	0.684166	0.685036	Vazão
6	300	306	0.597352	0.599452	0.60122	0.602717	0.603987	0.605063	0.605981	0.606771	0.607465	0.608089	0.608673	Vazão
4	300	304	0.591308	0.593598	0.595643	0.597444	0.598999	0.600314	0.601397	0.602257	0.602907	0.603363	0.603642	Vazão
151	300	451	0.587559	0.597564	0.607036	0.615471	0.622436	0.627682	0.631216	0.6333	0.634359	0.634826	0.635027	Vazão
2	300	302	0.58634	0.588413	0.589939	0.591134	0.5921	0.592899	0.593572	0.594149	0.594659	0.595123	0.595565	Vazão
3	300	303	0.583563	0.584863	0.586135	0.587376	0.588586	0.589762	0.590903	0.592007	0.593073	0.5941	0.595086	Vazão
5	300	305	0.556967	0.557598	0.55823	0.558861	0.559493	0.560124	0.560755	0.561387	0.562018	0.562649	0.56328	Vazão

Para uma melhor visualização das curvas, as Figuras 6.13 e 6.14 apresentam, respectivamente, em escala logarítmica e linear, as curvas Vazão vs. E_b/N_0 , considerando os seguintes pilotos: 11, 13, 16, 21, 26, 31.

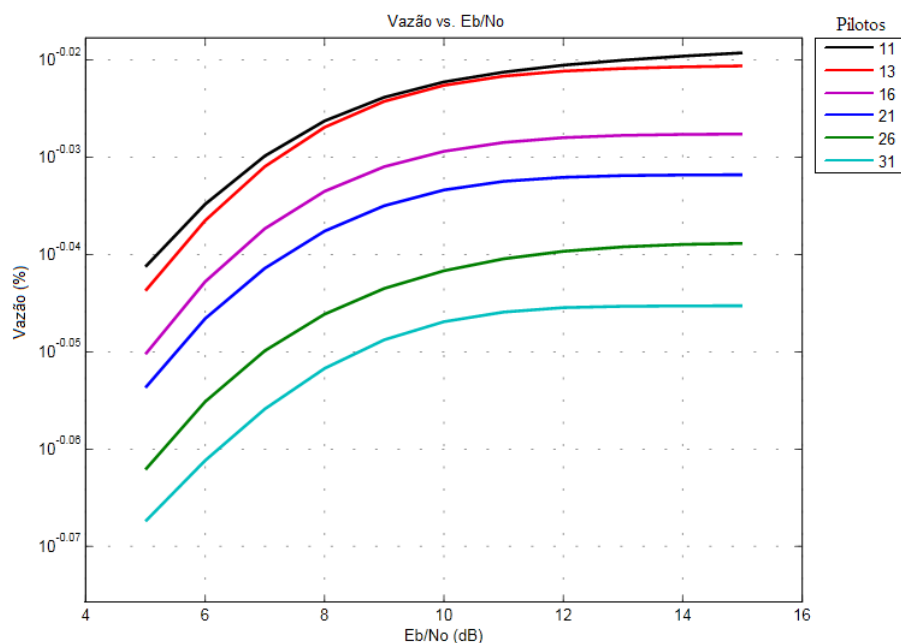


Figura 6.13 - Vazão vs. E_b/N_0 (Pilotos 11, 13, 16, 21, 26, 31). Ordem Decrescente da Vazão. Escala logarítmica.

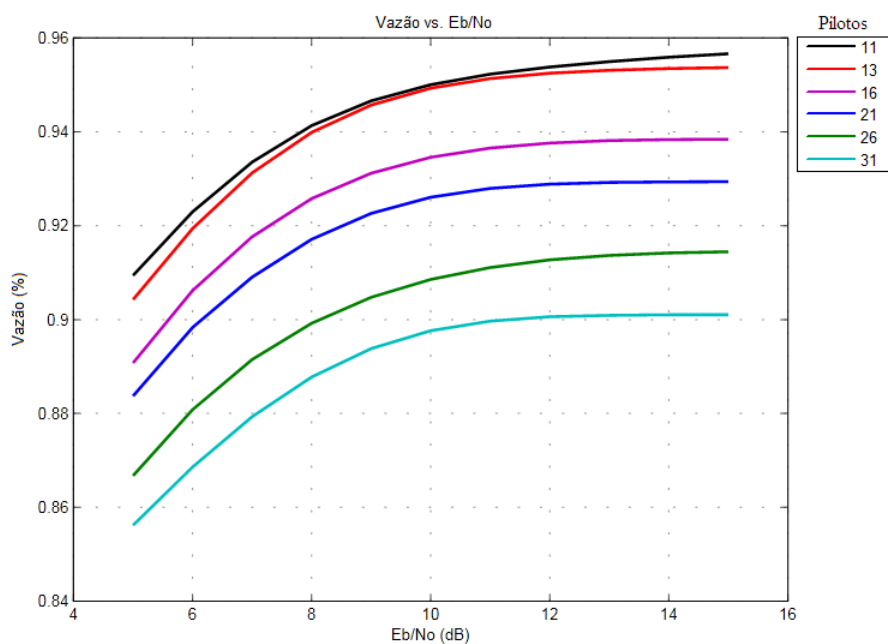


Figura 6.14 - Vazão vs. E_b/N_0 (Pilotos 11, 13, 16, 21, 26, 31). Ordem Decrescente da Vazão. Escala linear.

Para uma melhor visualização das curvas, as Figuras 6.15 e 6.16 apresentam, respectivamente, em escala logarítmica e linear, as curvas Vazão vs. E_b/N_0 , considerando os seguintes pilotos: 51, 61, 76, 7, 101.

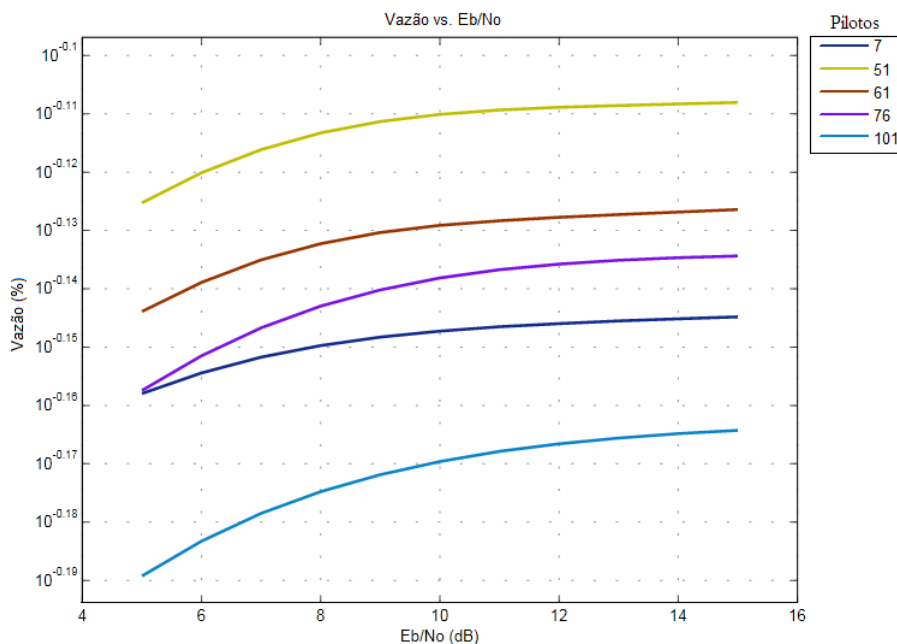


Figura 6.15 - Vazão vs. E_b/N_0 (Pilotos 51, 61, 76, 7, 101). Ordem Decrescente da Vazão. Escala logarítmica.

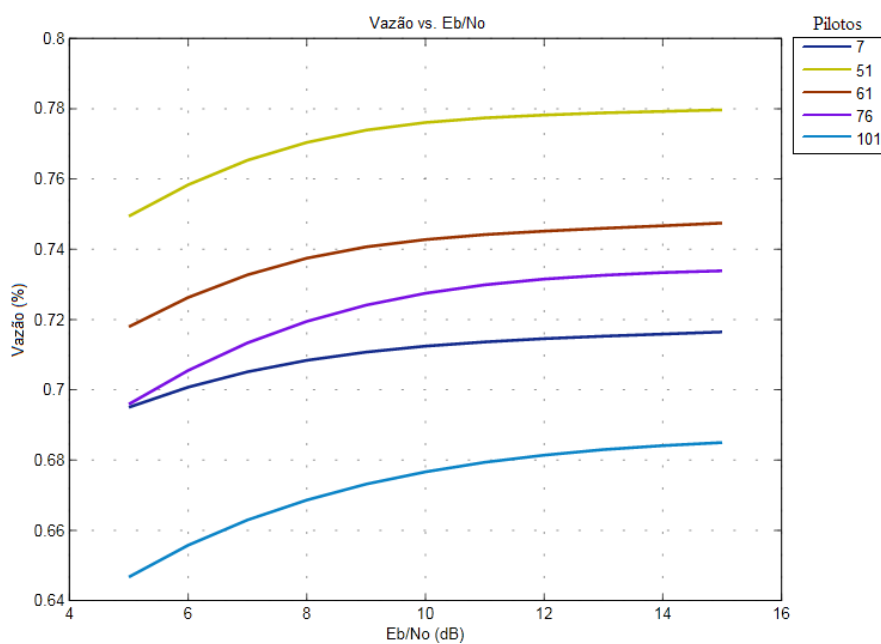


Figura 6.16 - Vazão vs. E_b/N_0 (Pilotos 51, 61, 76, 7, 101). Ordem Decrescente da Vazão. Escala linear.

Para uma melhor visualização das curvas, as Figuras 6.17 e 6.18 apresentam, respectivamente, em escala logarítmica e linear, as curvas Vazão vs. E_b/N_0 , considerando os seguintes pilotos: 6, 4, 151, 2, 3, 5.

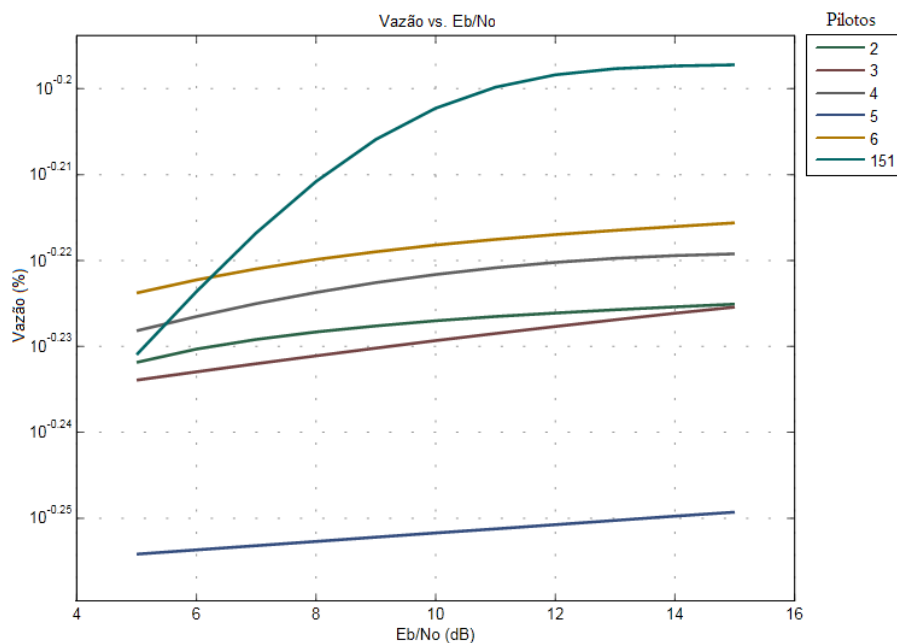


Figura 6.17 - Vazão vs. E_b/N_0 (Pilotos 6, 4, 151, 2, 3, 5). Ordem Decrescente da Vazão. Escala logarítmica.

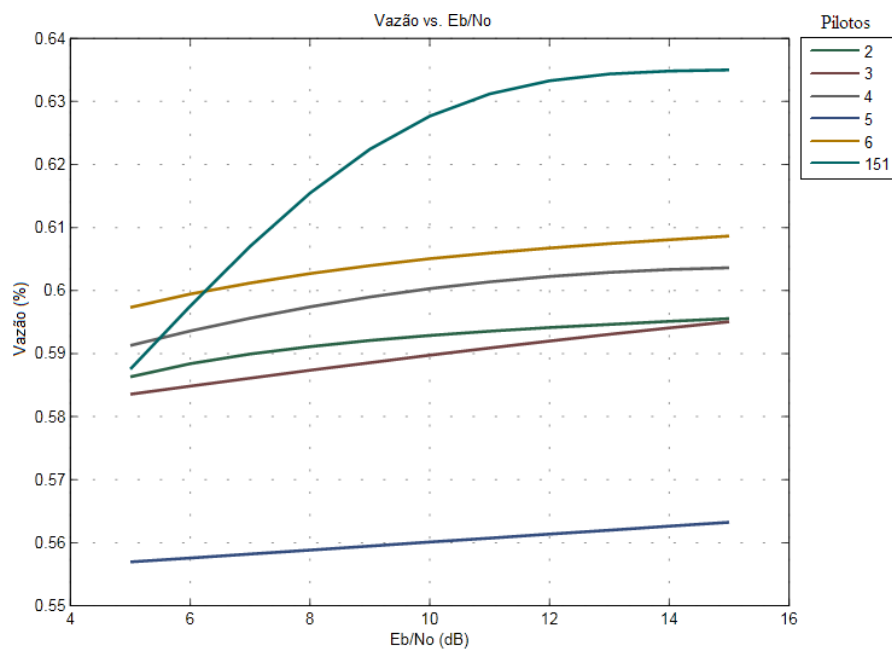


Figura 6.18 - Vazão vs. E_b/N_0 (Pilotos 6, 4, 151, 2, 3, 5). Ordem Decrescente da Vazão. Escala linear.

6.4. Taxa de Erro e Vazão vs. Eb/No

Ao analisar a curva e os dados de todos os pilotos, conclui-se que a sequência de pilotos considerando a ordem de uma melhor Taxa de Erro para uma pior Taxa de Erro fica:

BER: melhor para pior

21, 31, 13, 11, 26, 16, 151, 51, 76, 101, 61, 7, 6, 4, 2, 3, 5.

Ao analisar a curva e os dados de todos os pilotos, conclui-se que a sequência de pilotos considerando a ordem de uma melhor Vazão para uma pior Vazão fica:

Vazão: melhor para pior

11, 13, 16, 21, 26, 31, 51, 61, 76, 7, 101, 6, 4, 151, 2, 3, 5.

No Tópico 6.2 concluiu-se que, para trabalhar com uma menor Taxa de Erro, o desenvolvedor de um Projeto de Comunicações com Sistemas Móveis, utilizando a plataforma OFDM, deveria considerar as seguintes quantidades de pilotos: 16, 26, 11, 13, 31 ou 21. A alternativa de trabalhar com uma menor Taxa de Erro é ideal para os casos em que existirem dificuldades de retransmissão. A opção com a menor Taxa de Erro possível é a de 21 pilotos.

Assim, caso o desenvolvedor opte pela menor Taxa de Erro possível, ele deverá trabalhar com 21 pilotos. Esta opção não representa a opção de maior Vazão.

No tópico 6.3 concluiu-se que, para trabalhar com uma maior Vazão, o desenvolvedor de um Projeto de Comunicações com Sistemas Móveis, utilizando a plataforma OFDM, deveria considerar as seguintes quantidades de pilotos: 11, 13, 16, 21, 26 ou 31, sendo de 11 pilotos, a opção com a maior Vazão possível. Caso o desenvolvedor opte pela maior Vazão possível, ele deverá trabalhar com 11 pilotos. Esta opção não representa a opção de menor Taxa de Erro possível.

Destas 6 opções de pilotos (11, 13, 16, 21, 26, 31) , tem-se:

BER: melhor para pior

21, 31, 13, 11, 26, 16

Vazão: melhor para pior

11, 13, 16, 21, 26, 31

Assim, o desenvolvedor poderia descartar as seguintes alternativas:

- 16 pilotos: por apresentar a pior Taxa de Erro.
- 31 pilotos: por apresentar a pior Vazão.
- 26 pilotos: por apresentar Taxa de Erro e Vazão pior que a Taxa de Erro e Vazão dos pilotos 21 e 11.

O desenvolvedor pode optar por um meio termo, ou seja considerar uma Taxa de Erro média e uma Vazão média. Neste caso ele tem a seguinte alternativa: trabalhar com 13 pilotos. Embora a opção com 13 pilotos apresente uma Vazão pior que a opção com 11 pilotos e apresente uma Taxa de Erro pior que a opção com 21 pilotos, ela apresenta uma Vazão melhor que a opção com 21 pilotos e apresenta uma Taxa de Erro melhor que a opção com 11 pilotos.

Os dados de BER e Vazão em função do Eb/No estão reunidos na Tabela 6.5.

Tabela 6.5 - BER vs. Eb/No com a melhor curva de interpolação e Vazão vs. Eb/No (Pilotos 2, 3, 4, 5, 6, 7, 11, 13, 16, 21, 26, 31, 51, 61, 76, 101, 151).

Símbolos Pilotos	Símbolos Dados	Pilotos + Dados	5	6	7	8	9	10	11	12	13	14	15	Eb/No
2	300	302	0.409751	0.407664	0.406128	0.404926	0.403952	0.403148	0.402471	0.40189	0.401377	0.400909	0.400465	Taxa de Erro
			0.58634	0.588413	0.589939	0.591134	0.5921	0.592899	0.593572	0.594149	0.594659	0.595123	0.595565	Vazão
3	300	303	0.410601	0.409288	0.408004	0.40675	0.405528	0.40434	0.403188	0.402073	0.400996	0.399959	0.398963	Taxa de Erro
			0.583563	0.584863	0.586135	0.587376	0.588586	0.589762	0.590903	0.592007	0.593073	0.5941	0.595086	Vazão
4	300	304	0.400808	0.398488	0.396415	0.394591	0.393014	0.391682	0.390585	0.389713	0.389054	0.388592	0.38831	Taxa de Erro
			0.591308	0.593598	0.595643	0.597444	0.598999	0.600314	0.601397	0.602257	0.602907	0.603363	0.603642	Vazão
5	300	305	0.433751	0.433108	0.432466	0.431824	0.431182	0.430541	0.429899	0.429257	0.428615	0.427974	0.427332	Taxa de Erro
			0.556967	0.557598	0.55823	0.558861	0.559493	0.560124	0.560755	0.561387	0.562018	0.562649	0.56328	Vazão
6	300	306	0.390701	0.388559	0.386756	0.385228	0.383934	0.382835	0.381899	0.381093	0.380386	0.379749	0.379153	Taxa de Erro
			0.597352	0.599452	0.60122	0.602717	0.603987	0.605063	0.605981	0.606771	0.607465	0.608089	0.608673	Vazão
7	300	307	0.288716	0.282891	0.278412	0.275078	0.27266	0.270934	0.269693	0.268768	0.268034	0.267403	0.266824	Taxa de Erro
			0.695065	0.700758	0.705135	0.708393	0.710755	0.712443	0.713655	0.714559	0.715277	0.715893	0.716459	Vazão
11	300	311	0.057235	0.043201	0.032201	0.024166	0.018679	0.015109	0.012795	0.011208	0.010023	0.009079	0.008313	Taxa de Erro
			0.90942	0.922957	0.933568	0.941319	0.946612	0.950055	0.952288	0.953818	0.954962	0.955872	0.956612	Vazão
13	300	313	0.056547	0.040769	0.028394	0.019373	0.01331	0.009562	0.007414	0.006239	0.005595	0.005227	0.005008	Taxa de Erro
			0.904268	0.91939	0.931252	0.939898	0.94571	0.949301	0.95136	0.952487	0.953104	0.953457	0.953666	Vazão
16	300	316	0.061718	0.045492	0.033399	0.024842	0.01913	0.015561	0.01349	0.012384	0.011847	0.011612	0.01152	Taxa de Erro
			0.890774	0.906178	0.917659	0.925783	0.931205	0.934594	0.93656	0.93761	0.93812	0.938343	0.93843	Vazão
21	300	321	0.054406	0.038794	0.027302	0.018696	0.012779	0.009103	0.007076	0.006107	0.005715	0.005585	0.005551	Taxa de Erro
			0.883733	0.898324	0.909064	0.917107	0.922637	0.926072	0.927966	0.928872	0.929238	0.92936	0.929392	Vazão
26	300	326	0.058103	0.042838	0.031244	0.022803	0.016831	0.012699	0.009927	0.008164	0.007124	0.006565	0.006296	Taxa de Erro
			0.866777	0.880824	0.891493	0.899261	0.904757	0.90856	0.91111	0.912733	0.91369	0.914204	0.914452	Vazão
31	300	331	0.0553	0.041616	0.029824	0.020479	0.013799	0.009593	0.007326	0.006318	0.005963	0.005869	0.005852	Taxa de Erro
			0.856224	0.868626	0.879313	0.887783	0.893838	0.89765	0.899704	0.900618	0.90094	0.901025	0.901041	Vazão
51	300	351	0.123187	0.112722	0.104555	0.098599	0.094556	0.091993	0.090436	0.089471	0.0888	0.088251	0.087742	Taxa de Erro
			0.749413	0.758357	0.765337	0.770428	0.773883	0.776074	0.777405	0.77823	0.778803	0.779272	0.779708	Vazão
61	300	361	0.136074	0.126048	0.118248	0.112576	0.108709	0.106187	0.104528	0.103337	0.102356	0.101449	0.100564	Taxa de Erro
			0.717944	0.726276	0.732758	0.737472	0.740685	0.742781	0.74416	0.745149	0.745965	0.746718	0.747454	Vazão
76	300	376	0.127794	0.115727	0.105934	0.098249	0.092425	0.088171	0.085172	0.083124	0.081753	0.080834	0.080196	Taxa de Erro
			0.695909	0.705537	0.71335	0.719482	0.724129	0.727523	0.729916	0.73155	0.732644	0.733377	0.733886	Vazão
101	300	401	0.135504	0.123459	0.113838	0.106204	0.100196	0.095514	0.091902	0.089144	0.087059	0.085498	0.084335	Taxa de Erro
			0.646755	0.655766	0.662964	0.668676	0.67317	0.676673	0.679375	0.681439	0.682998	0.684166	0.685036	Vazão
151	300	451	0.116704	0.101663	0.087422	0.074742	0.064271	0.056385	0.051072	0.047939	0.046347	0.045644	0.045343	Taxa de Erro
			0.587559	0.597564	0.607036	0.615471	0.622436	0.627682	0.631216	0.6333	0.634359	0.634826	0.635027	Vazão

6.5. BER vs. Pilotos

Como foi verificado na Seção 6.3, ao analisar as Figuras 6.11 e 6.12 e os valores nas Tabelas 6.3 e 6.4, pôde-se concluir que a partir de um E_b/N_0 igual a 13 dB, o valor da Vazão fica praticamente constante, portanto as curvas assumem o formato de uma reta horizontal.

Em razão disso, será analisado, nesta seção, a Taxa de Erro em função da quantidade de pilotos, considerando-se um valor fixo de E_b/N_0 igual a 13 dB.

Na Seção 6.6 será analisada a Vazão em função da quantidade de pilotos, considerando-se um valor fixo de E_b/N_0 igual a 13 dB.

A Tabela 6.6 apresenta os valores de BER e Vazão em função da quantidade de pilotos, considerando um E_b/N_0 igual a 13 dB.

Tabela 6.6 - BER e Vazão vs. Pilotos, com E_b/N_0 igual a 13dB.

Símbolos Pilotos	Símbolos Dados	Pilotos + Dados	13	E_b/N_0
2	300	302	0.401376974	Taxa de Erro
			0.594658635	Vazão
3	300	303	0.400996491	Taxa de Erro
			0.593072782	Vazão
4	300	304	0.389053959	Taxa de Erro
			0.602907277	Vazão
5	300	305	0.428615286	Taxa de Erro
			0.562017751	Vazão
6	300	306	0.380386085	Taxa de Erro
			0.607464622	Vazão
7	300	307	0.268033636	Taxa de Erro
			0.715276577	Vazão
11	300	311	0.010022916	Taxa de Erro
			0.954961817	Vazão
13	300	313	0.00559486	Taxa de Erro
			0.953103968	Vazão
16	300	316	0.011846648	Taxa de Erro
			0.93812027	Vazão
21	300	321	0.005715171	Taxa de Erro
			0.929238158	Vazão
26	300	326	0.007123688	Taxa de Erro
			0.913689858	Vazão
31	300	331	0.00596291	Taxa de Erro
			0.900939961	Vazão
51	300	351	0.088800292	Taxa de Erro
			0.778803169	Vazão
61	300	361	0.102355907	Taxa de Erro
			0.74596462	Vazão
76	300	376	0.081753197	Taxa de Erro
			0.732643726	Vazão
101	300	401	0.087059077	Taxa de Erro
			0.682998196	Vazão
151	300	451	0.046347481	Taxa de Erro
			0.63435866	Vazão

As Figuras 6.19 e 6.20 e as Tabelas 6.7 e 6.8 apresentam, respectivamente, as curvas e os valores das Taxas de Erros (BER) versus a quantidade de pilotos, considerando E_b/N_0 igual a 13dB.

Foram consideradas as seguintes quantidades de pilotos: 2, 3, 4, 5, 6, 7, 11, 13, 16, 21, 26, 31, 51, 61, 76, 101, 151; foram traçados dois gráficos BER vs. Pilotos: um com a BER em escala logarítmica e o outro com a BER em escala linear.

Ao analisar a curva e os dados, observa-se que os pilotos 5, 2, 3, 4, 6, 7 apresentam as maiores Taxas de Erros. Os pilotos 61, 51, 101, 76, 151 apresentam Taxas de Erros intermediárias e observa-se que os pilotos 16, 11, 26, 31, 21, 13 apresentam as menores Taxas de erros.

Ao analisar a curva e os dados de todos os pilotos, conclui-se que a Taxa de Erro decresce, considerando a seguinte sequência de pilotos: 5, 2, 3, 4, 6, 7, 61, 51, 101, 76, 151, 16, 11, 26, 31, 21, 13.

A partir disso, conclui-se que, considerando E_b/N_0 igual a 13 dB, para trabalhar com uma menor Taxa de Erro, o desenvolvedor de um Projeto de Comunicações com Sistemas Móveis, utilizando a plataforma OFDM, deverá considerar as seguintes quantidades de pilotos: 16, 11, 26, 31, 21 ou 13, sendo 13 pilotos, a opção com a menor Taxa de Erro possível.

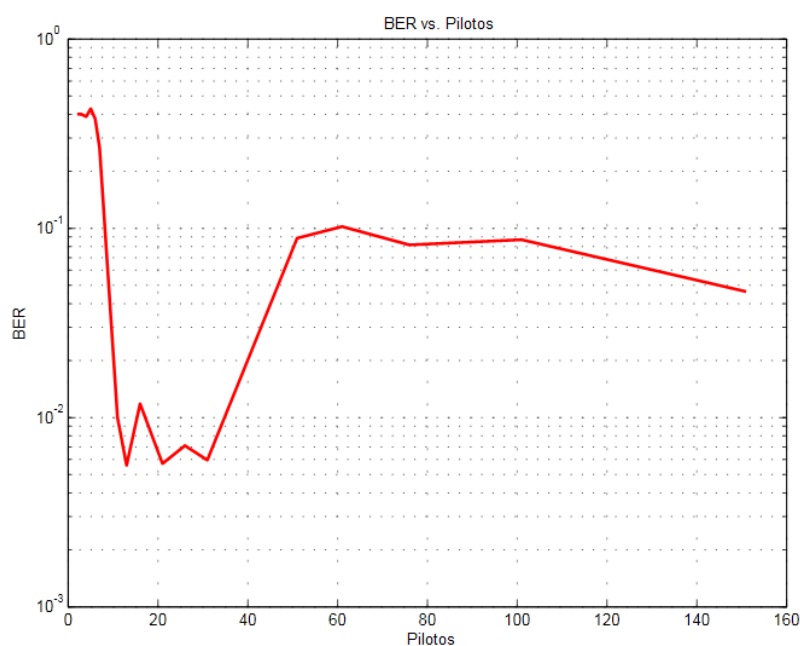


Figura 6.19 - BER vs. Pilotos, com E_b/N_0 igual a 13 dB (Pilotos 5, 2, 3, 4, 6, 7, 61, 51, 101, 76, 151, 16, 11, 26, 31, 21, 13). Ordem decrescente da taxa de erro. Escala logarítmica.

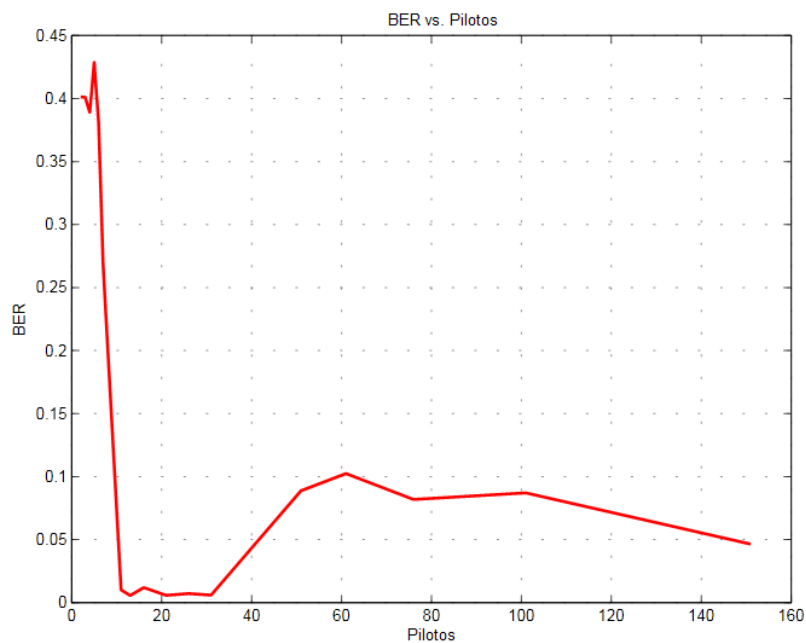


Figura 6.20 - BER vs. Pilotos, com E_b/N_0 igual a 13 dB (Pilotos 5, 2, 3, 4, 6, 7, 61, 51, 101, 76, 151, 16, 11, 26, 31, 21, 13). Ordem decrescente da taxa de erro. Escala linear.

Tabela 6.7 - BER vs. Pilotos, com E_b/N_0 igual a 13 dB (Pilotos 2, 3, 4, 5, 6, 7, 11, 13, 16, 21, 26, 31, 51, 61, 76, 101, 151).

Pilotos	BER
2	0,401377
3	0,400996
4	0,389054
5	0,428615
6	0,380386
7	0,268034
11	0,010023
13	0,005595
16	0,011847
21	0,005715
26	0,007124
31	0,005963
51	0,0888
61	0,102356
76	0,081753
101	0,087059
151	0,046347

Tabela 6.8 - BER vs. Pilotos, com E_b/N_0 igual a 13 dB (Pilotos 5, 2, 3, 4, 6, 7, 61, 51, 101, 76, 151, 16, 11, 26, 31, 21, 13). Ordem decrescente da taxa de erro.

Pilotos	BER
5	0,428615
2	0,401377
3	0,400996
4	0,389054
6	0,380386
7	0,268034
61	0,102356
51	0,0888
101	0,087059
76	0,081753
151	0,046347
16	0,011847
11	0,010023
26	0,007124
31	0,005963
21	0,005715
13	0,005595

Para uma melhor visualização das curvas, as Figuras 6.21 e 6.22 apresentam, respectivamente, em escala logarítmica e linear, as curvas BER vs. Pilotos, considerando os seguintes pilotos: 2, 3, 4, 5, 6 e 7.

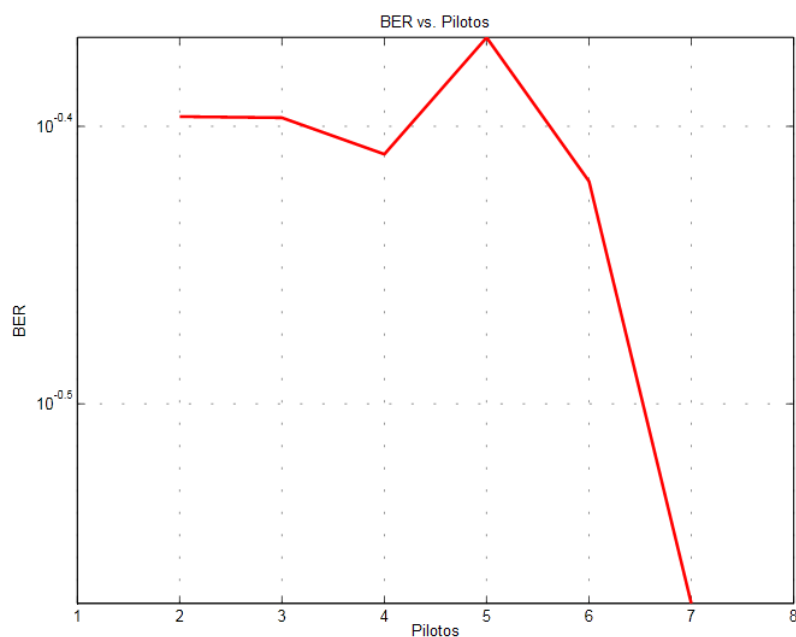


Figura 6.21 - BER vs. Pilotos, com E_b/N_0 igual a 13 dB (Pilotos 5, 2, 3, 4, 6, 7). Ordem decrescente da taxa de erro. Escala logarítmica.

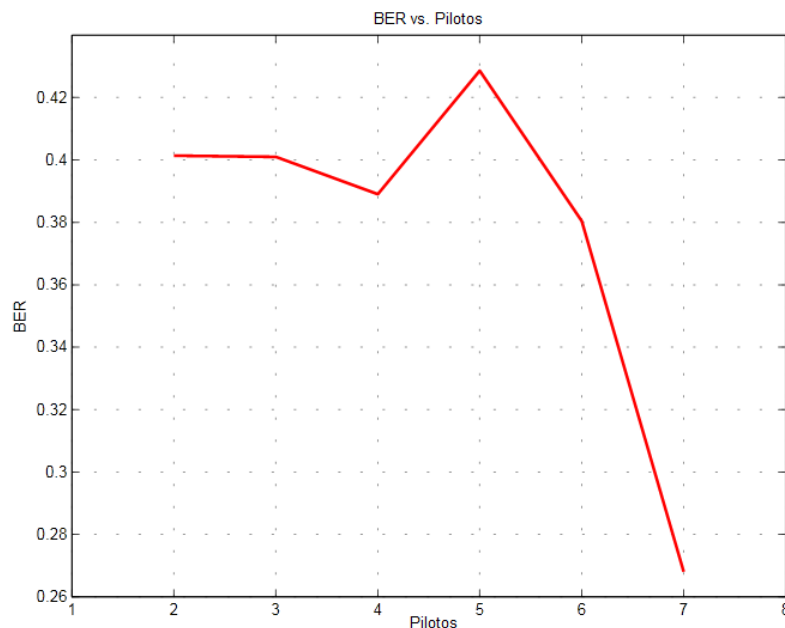


Figura 6.22 - BER vs. Pilotos, com E_b/N_0 igual a 13 dB (Pilotos 5, 2, 3, 4, 6, 7). Ordem decrescente da taxa de erro. Escala linear.

Para uma melhor visualização das curvas, as Figuras 6.23 e 6.24 apresentam, respectivamente, em escala logarítmica e linear, as curvas BER vs. Pilotos, considerando os seguintes pilotos: 11, 13, 16, 21, 26, 31.

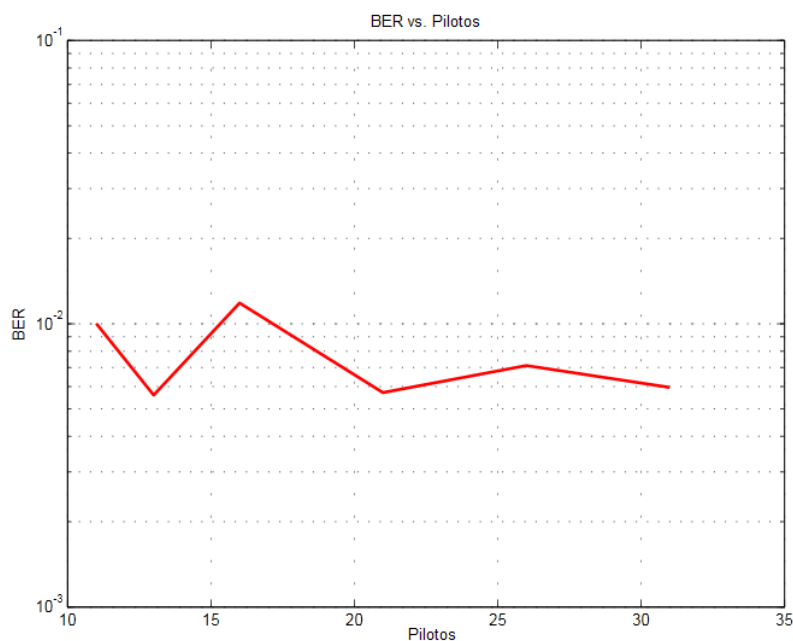


Figura 6.23 - BER vs. Pilotos, com E_b/N_0 igual a 13 dB (Pilotos 16, 11, 26, 31, 21, 13). Ordem decrescente da taxa de erro. Escala logarítmica.

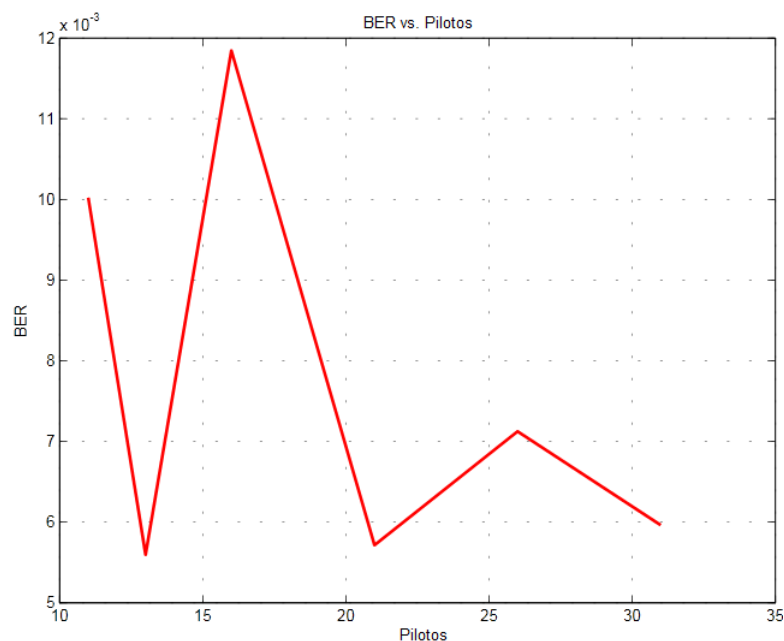


Figura 6.24 - BER vs. Pilotos, com E_b/N_0 igual a 13 dB (Pilotos 16, 11, 26, 31, 21, 13). Ordem decrescente da taxa de erro. Escala linear.

Para uma melhor visualização das curvas, as Figuras 6.25 e 6.26 apresentam, respectivamente, em escala logarítmica e linear, as curvas BER vs. Pilotos, considerando os seguintes pilotos: 51, 61, 76, 101, 151.

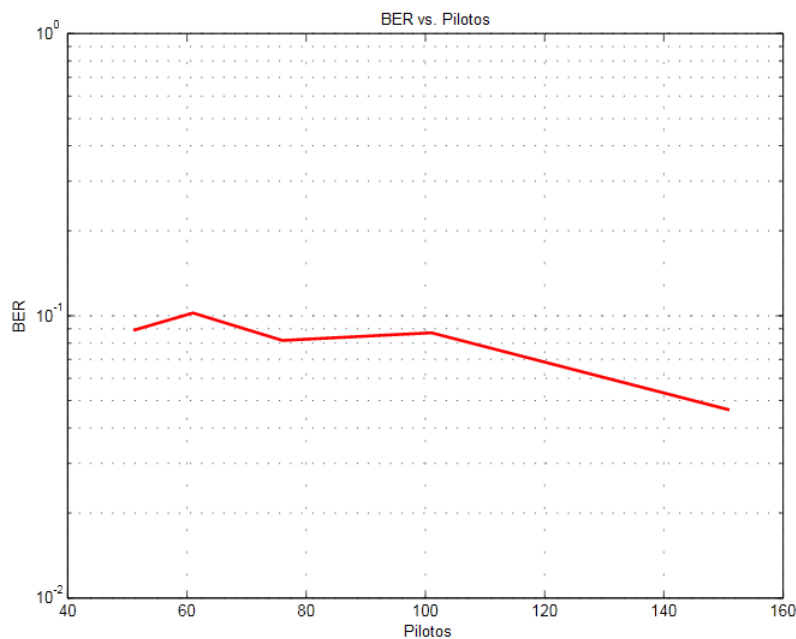


Figura 6.25 - BER vs. Pilotos, com E_b/N_0 igual a 13 dB (Pilotos 61, 51, 101, 76, 151). Ordem decrescente da taxa de erro. Escala logarítmica.

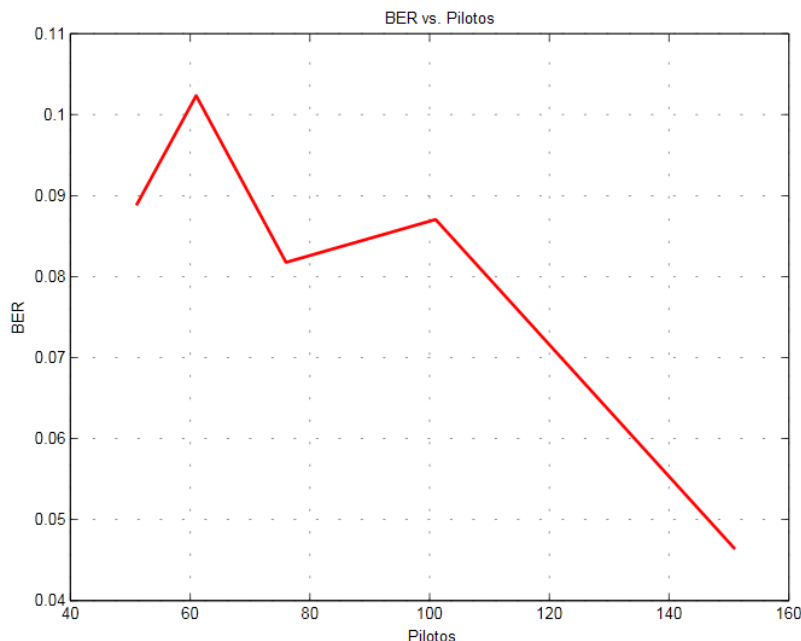


Figura 6.26 - BER vs. Pilotos, com E_b/N_0 igual a 13 dB (Pilotos 61, 51, 101, 76, 151). Ordem decrescente da taxa de erro. Escala linear.

6.6. Vazão vs. Pilotos

Nesta seção será analisado a Vazão em função da quantidade de pilotos, considerando-se um valor fixo de E_b/N_0 igual a 13 dB.

As Figuras 6.27 e 6.28 e as Tabelas 6.9 e 6.10 apresentam, respectivamente, as curvas e os valores de Vazão em função da quantidade de pilotos, considerando E_b/N_0 igual a 13 dB.

Foram consideradas as seguintes quantidades de pilotos: 2, 3, 4, 5, 6, 7, 11, 13, 16, 21, 26, 31, 51, 61, 76, 101, 151.

Conforme visto anteriormente, pode-se definir a vazão através dos seguintes modos:

$$\text{a) Vazão (\%)} = \frac{(1 - \text{Taxa de Erro}) \times \text{Bits Dados}}{(\text{Bits Dados} + \text{Bits Pilotos})} \quad [\text{Vazão}] = \%$$

$$\text{b) Vazão} = \frac{(1 - \text{Taxa de Erro}) \times \text{Bits Dados}}{(\text{Bits Dados} + \text{Bits Pilotos})} \times (\text{Bits Dados} + \text{Bits Pilotos})$$

Simplificando, tem-se:

Vazão = $(1 - \text{Taxa de Erro}) \times \text{Bits Dados}$

[Vazão] = bits/portadora

$$\text{c) Vazão Efetiva (\%)} = \frac{[(1 - \text{Taxa de Erro}) \times (\text{Bits Dados} \times \text{CM}_I)]}{[(\text{Bits Dados} \times \text{CM}_I) + \text{Bits Pilotos}]} \quad [\text{Vazão}] = \%$$

onde,

CM_I = Índice de Codificação e Modulação = Número de bits por símbolo = $(1/2) \times 4$ = (Codificador Turbo) * Modulador.

$$\text{d) Vazão Efetiva} = \frac{[(1 - \text{Taxa de Erro}) \times (\text{Bits Dados} \times \text{CM}_I)]}{[(\text{Bits Dados} \times \text{CM}_I) + \text{Bits Pilotos}]} \times$$

$$\times [(\text{Bits Dados} \times \text{CM}_I) + \text{Bits Pilotos}]$$

Simplificando, tem-se:

$$\text{Vazão Efetiva} = [(1 - \text{Taxa de Erro}) \times (\text{Bits Dados} \times \text{CM}_I)] \quad [\text{Vazão}] = \text{bits/portadora}$$

Nos gráficos e nas tabelas foi considerada a definição da alternativa a.

Foram traçados dois gráficos Vazão vs. Pilotos: um com a Vazão em escala logarítmica e o outro com a Vazão em escala linear.

Ao analisar a curva e os dados, observa-se que os pilotos 11, 13, 16, 21, 26, 31 apresentam os maiores valores de Vazão; os pilotos 51, 61, 76, 7, 101 apresentam valores de Vazão intermediários e os pilotos 151, 6, 4, 2, 3, 5 apresentam os menores valores de Vazão.

Ao analisar a curva e os dados de todos os pilotos, conclui-se que a Vazão decresce, considerando a seguinte sequência de pilotos: 11, 13, 16, 21, 26, 31, 51, 61, 76, 7, 101, 151, 6, 4, 2, 3, 5. A partir disso, conclui-se, considerando E_b/N_0 igual a 13 dB, que para trabalhar com uma maior Vazão, o desenvolvedor de um Projeto de Comunicações com Sistemas Móveis, utilizando a plataforma OFDM, deverá considerar as seguintes quantidades de pilotos: 11, 13, 16, 21, 26 ou 31, sendo 11 pilotos, a opção com a maior Vazão possível.

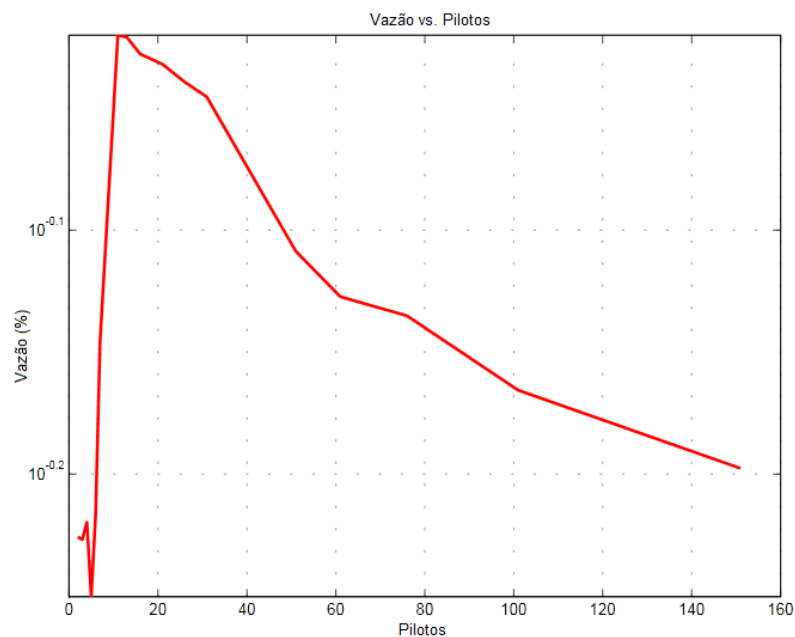


Figura 6.27 - Vazão vs. Pilotos, com E_b/N_0 igual a 13 dB (Pilotos 11, 13, 16, 21, 26, 31, 51, 61, 76, 7, 101, 151, 6, 4, 2, 3, 5). Ordem decrescente de Vazão. Escala logarítmica.

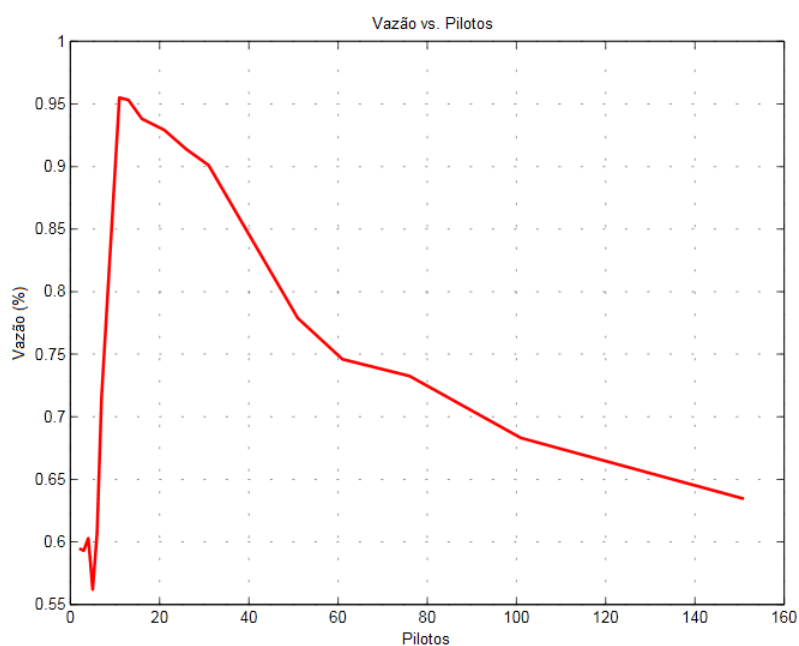


Figura 6.28 - Vazão vs. Pilotos, com E_b/N_0 igual a 13 dB (Pilotos 11, 13, 16, 21, 26, 31, 51, 61, 76, 7, 101, 151, 6, 4, 2, 3, 5). Ordem decrescente de Vazão. Escala linear.

Tabela 6.9 - Vazão vs. Pilotos, com E_b/N_0 igual a 13 dB (Pilotos 2, 3, 4, 5, 6, 7, 11, 13, 16, 21, 26, 31, 51, 61, 76, 101, 151).

Pilotos	Vazão
2	0,594659
3	0,593073
4	0,602907
5	0,562018
6	0,607465
7	0,715277
11	0,954962
13	0,953104
16	0,93812
21	0,929238
26	0,91369
31	0,90094
51	0,778803
61	0,745965
76	0,732644
101	0,682998
151	0,634359

Tabela 6.10 - Vazão vs. Pilotos, com E_b/N_0 igual a 13 dB (Pilotos 11, 13, 16, 21, 26, 31, 51, 61, 76, 7, 101, 151, 6, 4, 2, 3, 5). Ordem decrescente de Vazão.

Pilotos	Vazão
11	0,954962
13	0,953104
16	0,93812
21	0,929238
26	0,91369
31	0,90094
51	0,778803
61	0,745965
76	0,732644
7	0,715277
101	0,682998
151	0,634359
6	0,607465
4	0,602907
2	0,594659
3	0,593073
5	0,562018

Para uma melhor visualização das curvas, as Figuras 6.29 e 6.30 apresentam, respectivamente, em escala logarítmica e linear, as curvas Vazão vs. Pilotos, considerando os seguintes pilotos: 2, 3, 4, 5, 6, 7.

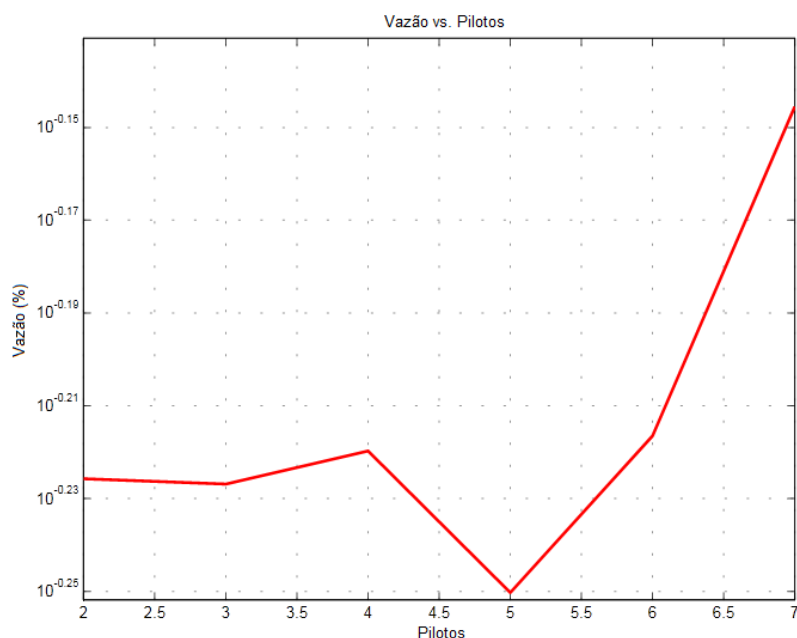


Figura 6.29 - Vazão vs. Pilotos, com E_b/N_0 igual a 13 dB (Pilotos 7, 6, 4, 2, 3, 5). Ordem decrescente de Vazão. Escala logarítmica.

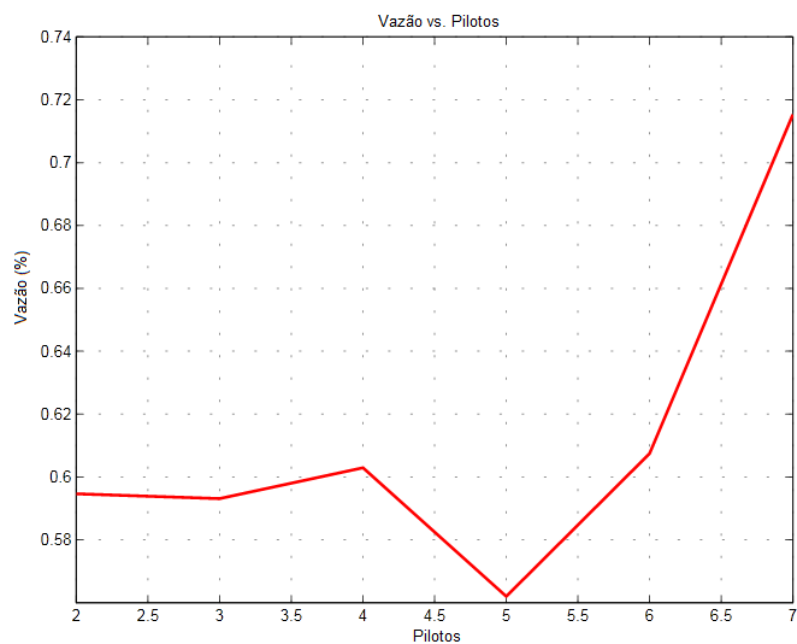


Figura 6.30 - Vazão vs. Pilotos, com E_b/N_0 igual a 13 dB (Pilotos 7, 6, 4, 2, 3, 5). Ordem decrescente de Vazão. Escala linear.

Para uma melhor visualização das curvas, as Figuras 6.31 e 6.32 apresentam, respectivamente, em escala logarítmica e linear, as curvas Vazão vs. Pilotos, considerando os seguintes pilotos: 11, 13, 16, 21, 26, 31.

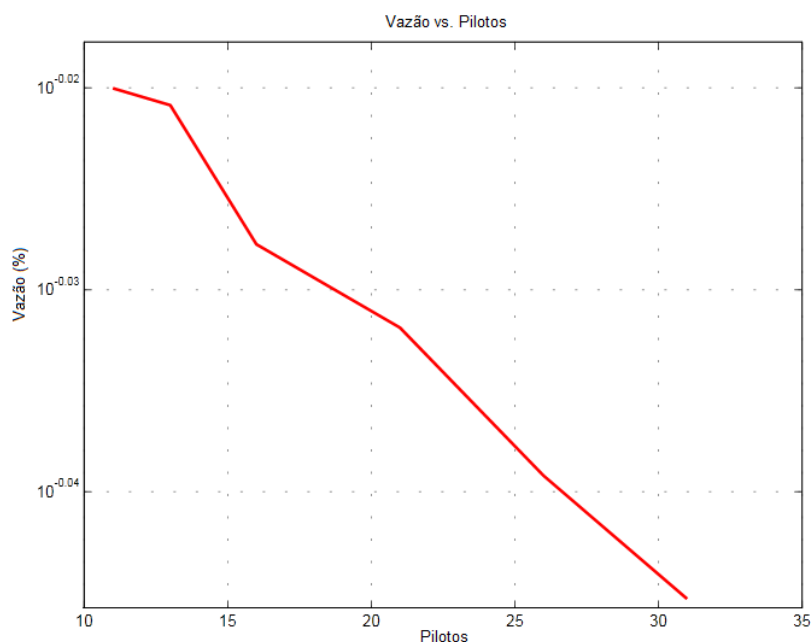


Figura 6.31 - Vazão vs. Pilotos, com E_b/N_0 igual a 13 dB (Pilotos 11, 13, 16, 21, 26, 31). Ordem decrescente de Vazão. Escala logarítmica.

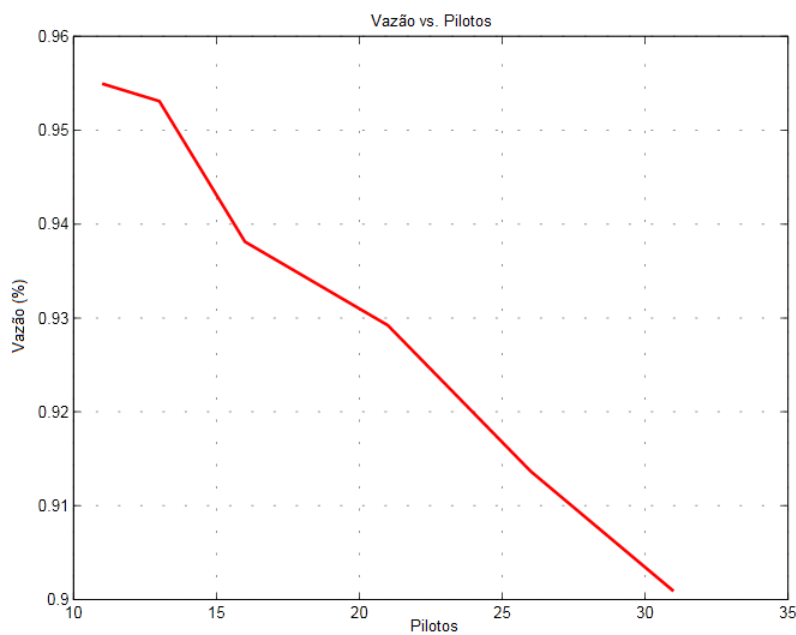


Figura 6.32 - Vazão vs. Pilotos, com E_b/N_0 igual a 13 dB (Pilotos 11, 13, 16, 21, 26, 31). Ordem decrescente de Vazão. Escala linear.

Para uma melhor visualização das curvas, as Figuras 6.33 e 6.34 apresentam, respectivamente, em escala logarítmica e linear, as curvas Vazão vs. Pilotos, considerando os seguintes pilotos: 51, 61, 76, 101, 151.

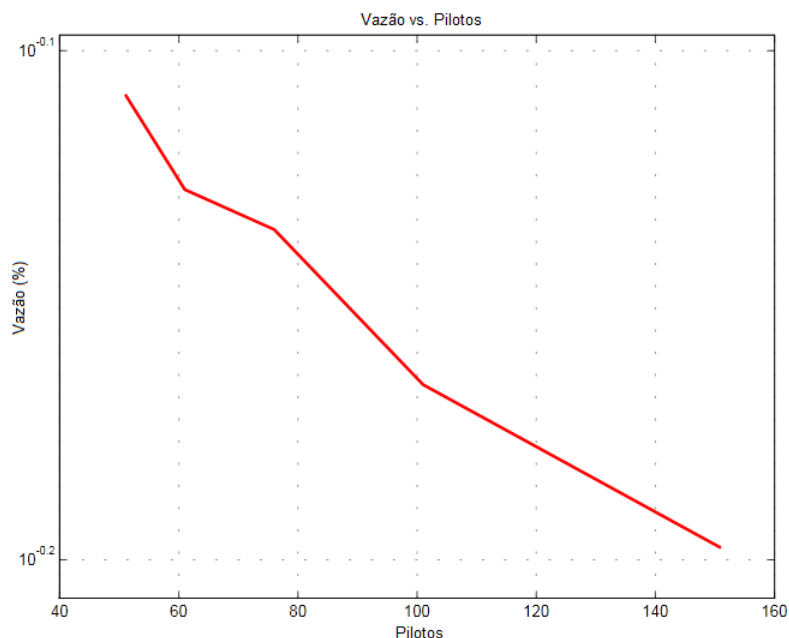


Figura 6.33 - Vazão vs. Pilotos, com E_b/N_0 igual a 13 dB (Pilotos 51, 61, 76, 101, 151). Ordem decrescente de Vazão. Escala logarítmica.

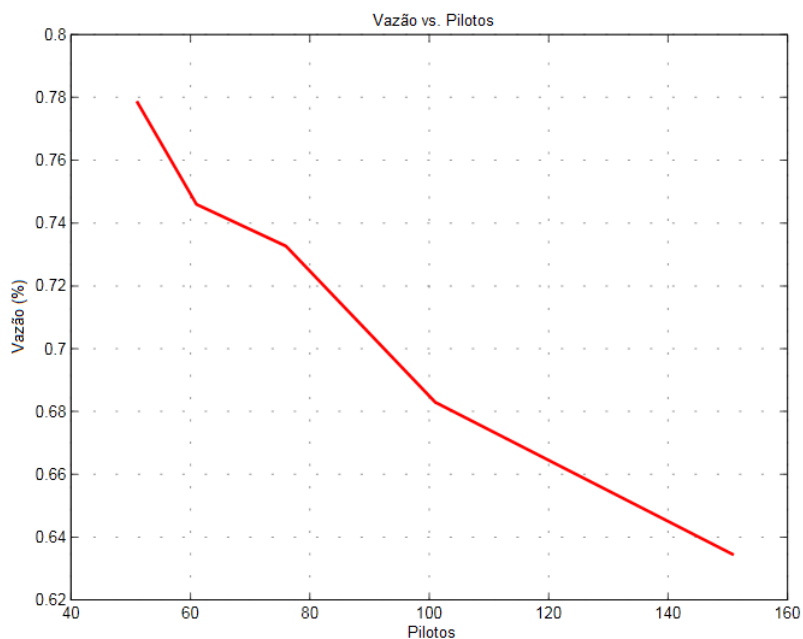


Figura 6.34 - Vazão vs. Pilotos, com E_b/N_0 igual a 13 dB (Pilotos 51, 61, 76, 101, 151). Ordem decrescente de Vazão. Escala linear.

6.7. Taxa de Erro e Vazão vs. Pilotos

Ao analisar a curva e os dados de todos os pilotos, com E_b/N_0 igual a 13 dB, conclui-se que a sequência de pilotos, considerando a ordem de uma melhor Taxa de Erro para uma pior Taxa de Erro, fica:

BER: melhor para pior

13, 21, 31, 26, 11, 16, 151, 76, 101, 51, 61, 7, 6, 4, 3, 2, 5.

Ao analisar a curva e os dados de todos os pilotos, com E_b/N_0 igual a 13 dB, conclui-se que a sequência de pilotos, considerando a ordem de uma melhor Vazão para uma pior Vazão, fica:

Vazão: melhor para pior

11, 13, 16, 21, 26, 31, 51, 61, 76, 7, 101, 151, 6, 4, 2, 3, 5.

No Tópico 6.4, concluiu-se, considerando E_b/N_0 igual a 13 dB, que para trabalhar com uma menor Taxa de Erro, o desenvolvedor de um Projeto de Comunicações com Sistemas Móveis, utilizando a plataforma OFDM, deveria considerar as seguintes quantidades de pilotos: 16, 11, 26, 31, 21 ou 13, sendo 13 pilotos, a opção com a menor Taxa de Erro possível. A alternativa de trabalhar com uma menor Taxa de Erro é ideal para os casos em que existirem dificuldades de retransmissão.

Caso o desenvolvedor opte pela menor Taxa de Erro possível, ele deverá trabalhar com 13 pilotos, mas esta opção não representa a opção de maior Vazão.

No tópico 6.5, concluiu-se, considerando E_b/N_0 igual a 13 dB, que para trabalhar com uma maior Vazão, o desenvolvedor de um Projeto de Comunicações com Sistemas Móveis, utilizando a plataforma OFDM, deveria considerar as seguintes quantidades de pilotos: 11, 13, 16, 21, 26 ou 31, sendo 11 pilotos, a opção com a maior Vazão possível.

Caso o desenvolvedor opte pela maior Vazão possível, ele deverá trabalhar com 11 pilotos, mas esta opção não representa a opção de menor Taxa de Erro possível.

Destas 6 opções de pilotos (11, 13, 16, 21, 26, 31) , temos

BER: melhor para pior

13, 21, 31, 26, 11, 16

Vazão: melhor para pior

11, 13, 16, 21, 26, 31

e o desenvolvedor poderia descartar as seguintes alternativas:

- 16 pilotos: por apresentar a pior Taxa de Erro.
- 31 pilotos: por apresentar a pior Vazão.
- 26 pilotos: embora apresente uma Taxa de Erro um pouco melhor que dos 11 pilotos, apresenta uma Taxa de Erro muito pior que dos 13 pilotos e apresenta uma Vazão muito pior que dos 11 e 13 pilotos.

O desenvolvedor neste caso pode optar por um meio termo, ou seja pode considerar a melhor Taxa de Erro e uma Vazão média. Neste caso ele tem a seguinte alternativa: trabalhar com 13 pilotos.

A opção com 13 pilotos apresenta a melhor Taxa de Erro e uma Vazão um pouco pior que a opção com 11 pilotos. Na média, 13 pilotos é a melhor das opções. Embora a opção com 11 pilotos apresente a melhor Vazão, ela na média não é recomendada, pois apresenta uma Taxa de Erro muito pior que dos 13 pilotos. Embora a opção com 21 pilotos apresente uma Taxa de Erro muito melhor que a opção com 11 pilotos, ela apresenta uma Vazão pior que as opções com 11 e 13 pilotos. Mais uma vez a opção com 13 pilotos é a melhor das opções.

Os dados de BER e Vazão em função da quantidade de pilotos estão reunidos na Tabela 6.11.

Tabela 6.11 - BER e Vazão vs Pilotos.

Pilotos	BER	Vazão
2	0,401377	0,594659
3	0,400996	0,593073
4	0,389054	0,602907
5	0,428615	0,562018
6	0,380386	0,607465
7	0,268034	0,715277
11	0,010023	0,954962
13	0,005595	0,953104
16	0,011847	0,93812
21	0,005715	0,929238
26	0,007124	0,91369
31	0,005963	0,90094
51	0,0888	0,778803
61	0,102356	0,745965
76	0,081753	0,732644
101	0,087059	0,682998
151	0,046347	0,634359

6.8. Vazão Efetiva vs. Eb/No

Se tivesse sido considerada a definição de Vazão Efetiva (alternativa c do item 6.3) ao invés de ter sido considerada a definição de Vazão (alternativa a), ter-se-ia:

$$\text{Vazão Efetiva (\%)} = \frac{[(1 - \text{Taxa de Erro}) \times (\text{Bits Dados} \times \text{CM}_I)]}{[(\text{Bits Dados} \times \text{CM}_I) + \text{Bits Pilotos}]}$$

$$[\text{Vazão}] = \%$$

onde,

CM_I = Índice de Codificação e Modulação = Número de bits por símbolo = $(1/2) \times 4$ = (Codificador Turbo) * Modulador.

Comparando-se os valores de Vazão Efetiva (Tabela 6.12) com os valores de Vazão (Tabela 6.3), percebe-se que os valores de Vazão Efetiva são levemente maiores que os valores de Vazão, além de representarem valores mais realísticos e efetivos, como o próprio nome sugere.

A Tabela 6.12 apresenta os valores de Vazão Efetiva em função do Eb/No, considerando as seguintes quantidades de pilotos: 2, 3, 4, 5, 6, 7, 11, 13, 16, 21, 26, 31, 51, 61, 76, 101, 151. Foram considerados os seguintes valores em decibel (dB) para o Eb/No: 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15.

Ao analisar os dados de cada piloto, separadamente, conclui-se que quanto maior o Eb/No, maior será a Vazão Efetiva. As Tabelas 6.12 e 6.13 mostram os resultados com um crescimento gradativo da Vazão Efetiva com o Eb/No. Observa-se que a Vazão Efetiva cresce lentamente para os seguintes pilotos: 2, 3, 4, 5, 6; a Vazão Efetiva cresce um pouco mais rápido para os seguintes pilotos: 7, 51, 61, 76, 101, 151 e a Vazão Efetiva cresce, com uma maior rapidez, para os seguintes pilotos: 11, 13, 16, 21, 26, 31.

Analisando os valores nas Tabelas 6.12 e 6.13, pode-se concluir, também, que a partir de um Eb/No igual a 13 dB, o valor da Vazão Efetiva fica praticamente constante, portanto, se as curvas fossem plotadas, elas assumiriam o formato de uma reta horizontal.

Ao analisar os dados de todos os pilotos (Tabela 6.13), conclui-se que a Vazão Efetiva decresce, considerando a seguinte sequência de pilotos:

11, 13, 16, 21, 26, 31, 51, 61, 76, 101, 151, 7, 6, 4, 2, 3, 5

Já o Vazão (Tabela 6.3), analisado anteriormente, decrescia considerando a seguinte sequência de pilotos:

11, 13, 16, 21, 26, 31, 51, 61, 76, 7, 101, 6, 4, 151, 2, 3, 5.

Comparando ambos os resultados, pouco mudou (a vazão para 101 e 151 pilotos aumentou e a vazão para 4, 6 e 7 pilotos diminuiu). Como a quantidade de pilotos recomendada, para ambos os casos, é a mesma (11, 13, 16, 21, 26, 31), todas as análises feitas anteriormente permanecem fidedignas.

A partir dos resultados acima, conclui-se que para trabalhar com uma maior Vazão Efetiva, o desenvolvedor de um Projeto de Comunicações com Sistemas Móveis, utilizando a plataforma OFDM, deverá considerar as seguintes quantidades de pilotos: 11, 13, 16, 21, 26 ou 31, sendo 11 pilotos, a opção com a maior Vazão possível.

Sendo o foco da investigação, a influência da densidade de pilotos no desempenho do sistema OFDM, conclui-se que o desempenho da BER melhora consistentemente com a inserção de mais pilotos, mas a Vazão Efetiva reduz devido à inserção de pilotos.

Para uma quantidade média de pilotos inseridos (11, 13, 16, 21, 26 e 31), a Vazão Efetiva é elevada. Aumentado-se consideravelmente a quantidade de pilotos inseridos (51, 61, 76, 101 e 151), a Vazão Efetiva diminui, como era de se esperar, porém resultados da simulação mostram que, se a quantidade de pilotos inseridos (7, 6, 4, 2, 3, 5) abaixar consideravelmente, o desempenho do sistema degrada, atingindo valores muito baixos de Vazão Efetiva.

Tabela 6.12 - Vazão Efetiva vs. Eb/No (Pilotos 2, 3, 4, 5, 6, 7, 11, 13, 16, 21, 26, 31, 51, 61, 76, 101, 151).

Símbolos Pilotos	Símbolos Dados	Pilotos + Dados	5	6	7	8	9	10	11	12	13	14	15	Eb/No
2	600	602	0.588288	0.590368	0.591899	0.593097	0.594067	0.594869	0.595544	0.596123	0.596634	0.5971	0.597543	Vazão Efetiva
3	600	603	0.586467	0.587773	0.589051	0.590299	0.591514	0.592696	0.593842	0.594952	0.596023	0.597055	0.598046	Vazão Efetiva
4	600	604	0.595224	0.597529	0.599588	0.6014	0.602966	0.60429	0.605379	0.606245	0.6069	0.607359	0.607639	Vazão Efetiva
5	600	605	0.56157	0.562207	0.562843	0.56348	0.564117	0.564753	0.56539	0.566026	0.566663	0.567299	0.567935	Vazão Efetiva
6	600	606	0.603266	0.605387	0.607173	0.608685	0.609967	0.611054	0.611981	0.612779	0.613479	0.61411	0.6147	Vazão Efetiva
7	600	607	0.703081	0.70884	0.713267	0.716562	0.718952	0.720659	0.721885	0.722799	0.723525	0.724149	0.724721	Vazão Efetiva
11	600	611	0.925792	0.939573	0.950375	0.958266	0.963654	0.967159	0.969432	0.97099	0.972154	0.973081	0.973834	Vazão Efetiva
13	600	613	0.923445	0.938888	0.951001	0.959831	0.965766	0.969433	0.971536	0.972686	0.973317	0.973677	0.973891	Vazão Efetiva
16	600	616	0.913911	0.929715	0.941494	0.94983	0.955393	0.958869	0.960886	0.961963	0.962487	0.962716	0.962805	Vazão Efetiva
21	600	621	0.913617	0.928702	0.939805	0.94812	0.953837	0.957389	0.959346	0.960283	0.960662	0.960787	0.96082	Vazão Efetiva
26	600	626	0.902777	0.917408	0.92852	0.93661	0.942334	0.946295	0.948952	0.950642	0.951639	0.952174	0.952432	Vazão Efetiva
31	600	631	0.898288	0.9113	0.922512	0.931398	0.93775	0.94175	0.943905	0.944864	0.945202	0.945291	0.945307	Vazão Efetiva
51	600	651	0.808123	0.817768	0.825295	0.830784	0.83451	0.836873	0.838308	0.839198	0.839815	0.840321	0.840791	Vazão Efetiva
61	600	661	0.784199	0.7933	0.80038	0.805529	0.809039	0.811328	0.812834	0.813915	0.814806	0.815629	0.816432	Vazão Efetiva
76	600	676	0.774147	0.784858	0.793549	0.800371	0.80554	0.809316	0.811977	0.813795	0.815012	0.815828	0.816394	Vazão Efetiva
101	600	701	0.739939	0.750249	0.758484	0.765018	0.77016	0.774168	0.77726	0.77962	0.781404	0.782741	0.783736	Vazão Efetiva
151	600	751	0.705696	0.717713	0.72909	0.73922	0.747586	0.753887	0.758131	0.760635	0.761906	0.762468	0.762709	Vazão Efetiva

Tabela 6.13 - Vazão Efetiva vs. Eb/No (Pilotos 11, 13, 16, 21, 26, 31, 51, 61, 76, 101, 151, 7, 6, 4, 2, 3, 5). Ordem Decrescente da Vazão.

Simbolos Pilotos	Simbolos Dados	Pilotos + Dados	5	6	7	8	9	10	11	12	13	14	15	Eb/No
11	600	611	0.925792	0.939573	0.950375	0.958266	0.963654	0.967159	0.969432	0.97099	0.972154	0.973081	0.973834	Vazão Efetiva
13	600	613	0.923445	0.938888	0.951001	0.959831	0.965766	0.969433	0.971536	0.972686	0.973317	0.973677	0.973891	Vazão Efetiva
16	600	616	0.913911	0.929715	0.941494	0.94983	0.955393	0.958869	0.960886	0.961963	0.962487	0.962716	0.962805	Vazão Efetiva
21	600	621	0.913617	0.928702	0.939805	0.94812	0.953837	0.957389	0.959346	0.960283	0.960662	0.960787	0.96082	Vazão Efetiva
26	600	626	0.902777	0.917408	0.92852	0.93661	0.942334	0.946295	0.948952	0.950642	0.951639	0.952174	0.952432	Vazão Efetiva
31	600	631	0.898288	0.9113	0.922512	0.931398	0.93775	0.94175	0.943905	0.944864	0.945202	0.945291	0.945307	Vazão Efetiva
51	600	651	0.808123	0.817768	0.825295	0.830784	0.83451	0.836873	0.838308	0.839198	0.839815	0.840321	0.840791	Vazão Efetiva
61	600	661	0.784199	0.7933	0.80038	0.805529	0.809039	0.811328	0.812834	0.813915	0.814806	0.815629	0.816432	Vazão Efetiva
76	600	676	0.774147	0.784858	0.793549	0.800371	0.80554	0.809316	0.811977	0.813795	0.815012	0.815828	0.816394	Vazão Efetiva
101	600	701	0.739939	0.750249	0.758484	0.765018	0.77016	0.774168	0.77726	0.77962	0.781404	0.782741	0.783736	Vazão Efetiva
151	600	751	0.705696	0.717713	0.72909	0.73922	0.747586	0.753887	0.758131	0.760635	0.761906	0.762468	0.762709	Vazão Efetiva
7	600	607	0.703081	0.70884	0.713267	0.716562	0.718952	0.720659	0.721885	0.722799	0.723525	0.724149	0.724721	Vazão Efetiva
6	600	606	0.603266	0.605387	0.607173	0.608685	0.609967	0.611054	0.611981	0.612779	0.613479	0.61411	0.6147	Vazão Efetiva
4	600	604	0.595224	0.597529	0.599588	0.6014	0.602966	0.60429	0.605379	0.606245	0.6069	0.607359	0.607639	Vazão Efetiva
2	600	602	0.588288	0.590368	0.591899	0.593097	0.594067	0.594869	0.595544	0.596123	0.596634	0.5971	0.597543	Vazão Efetiva
3	600	603	0.586467	0.587773	0.589051	0.590299	0.591514	0.592696	0.593842	0.594952	0.596023	0.597055	0.598046	Vazão Efetiva
5	600	605	0.56157	0.562207	0.562843	0.56348	0.564117	0.564753	0.56539	0.566026	0.566663	0.567299	0.567935	Vazão Efetiva

Tabela 6.14 - BER vs. Eb/No com a melhor curva de interpolação e Vazão Efetiva vs. Eb/No (Pilotos 2, 3, 4, 5, 6, 7, 11, 13, 16, 21, 26, 31, 51, 61, 76, 101, 151).

Símbolos Pilotos	Símbolos Dados	Pilotos + Dados	5	6	7	8	9	10	11	12	13	14	15	Eb/No
2	600	602	0.409751	0.407664	0.406128	0.404926	0.403952	0.403148	0.402471	0.40189	0.401377	0.400909	0.400465	Taxa de Erro
			0.588288	0.590368	0.591899	0.593097	0.594067	0.594869	0.595544	0.596123	0.596634	0.5971	0.597543	Vazão Efetiva
3	600	603	0.410601	0.409288	0.408004	0.40675	0.405528	0.40434	0.403188	0.402073	0.400996	0.399959	0.398963	Taxa de Erro
			0.586467	0.587773	0.589051	0.590299	0.591514	0.592696	0.593842	0.594952	0.596023	0.597055	0.598046	Vazão Efetiva
4	600	604	0.400808	0.398488	0.396415	0.394591	0.393014	0.391682	0.390585	0.389713	0.389054	0.388592	0.38831	Taxa de Erro
			0.595224	0.597529	0.599588	0.6014	0.602966	0.60429	0.605379	0.606245	0.6069	0.607359	0.607639	Vazão Efetiva
5	600	605	0.433751	0.433108	0.432466	0.431824	0.431182	0.430541	0.429899	0.429257	0.428615	0.427974	0.427332	Taxa de Erro
			0.56157	0.562207	0.562843	0.56348	0.564117	0.564753	0.56539	0.566026	0.566663	0.567299	0.567935	Vazão Efetiva
6	600	606	0.390701	0.388559	0.386756	0.385228	0.383934	0.382835	0.381899	0.381093	0.380386	0.379749	0.379153	Taxa de Erro
			0.603266	0.605387	0.607173	0.608685	0.609967	0.611054	0.611981	0.612779	0.613479	0.61411	0.6147	Vazão Efetiva
7	600	607	0.288716	0.282891	0.278412	0.275078	0.27266	0.270934	0.269693	0.268768	0.268034	0.267403	0.266824	Taxa de Erro
			0.703081	0.70884	0.713267	0.716562	0.718952	0.720659	0.721885	0.722799	0.723525	0.724149	0.724721	Vazão Efetiva
11	600	611	0.057235	0.043201	0.032201	0.024166	0.018679	0.015109	0.012795	0.011208	0.010023	0.009079	0.008313	Taxa de Erro
			0.925792	0.939573	0.950375	0.958266	0.963654	0.967159	0.969432	0.97099	0.972154	0.973081	0.973834	Vazão Efetiva
13	600	613	0.056547	0.040769	0.028394	0.019373	0.01331	0.009562	0.007414	0.006239	0.005595	0.005227	0.005008	Taxa de Erro
			0.923445	0.938888	0.951001	0.959831	0.965766	0.969433	0.971536	0.972686	0.973317	0.973677	0.973891	Vazão Efetiva
16	600	616	0.061718	0.045492	0.033399	0.024842	0.01913	0.015561	0.01349	0.012384	0.011847	0.011612	0.01152	Taxa de Erro
			0.913911	0.929715	0.941494	0.94983	0.955393	0.958869	0.960886	0.961963	0.962487	0.962716	0.962805	Vazão Efetiva
21	600	621	0.054406	0.038794	0.027302	0.018696	0.012779	0.009103	0.007076	0.006107	0.005715	0.005585	0.005551	Taxa de Erro
			0.913617	0.928702	0.939805	0.94812	0.953837	0.957389	0.959346	0.960283	0.960662	0.960787	0.96082	Vazão Efetiva
26	600	626	0.058103	0.042838	0.031244	0.022803	0.016831	0.012699	0.009927	0.008164	0.007124	0.006565	0.006296	Taxa de Erro
			0.902777	0.917408	0.92852	0.93661	0.942334	0.946295	0.948952	0.950642	0.951639	0.952174	0.952432	Vazão Efetiva
31	600	631	0.0553	0.041616	0.029824	0.020479	0.013799	0.009593	0.007326	0.006318	0.005963	0.005869	0.005852	Taxa de Erro
			0.898288	0.9113	0.922512	0.931398	0.93775	0.94175	0.943905	0.944864	0.945202	0.945291	0.945307	Vazão Efetiva
51	600	651	0.123187	0.112722	0.104555	0.098599	0.094556	0.091993	0.090436	0.089471	0.0888	0.088251	0.087742	Taxa de Erro
			0.808123	0.817768	0.825295	0.830784	0.83451	0.836873	0.838308	0.839198	0.839815	0.840321	0.840791	Vazão Efetiva
61	600	661	0.136074	0.126048	0.118248	0.112576	0.108709	0.106187	0.104528	0.103337	0.102356	0.101449	0.100564	Taxa de Erro
			0.784199	0.7933	0.80038	0.805529	0.809039	0.811328	0.812834	0.813915	0.814806	0.815629	0.816432	Vazão Efetiva
76	600	676	0.127794	0.115727	0.105934	0.098249	0.092425	0.088171	0.085172	0.083124	0.081753	0.080834	0.080196	Taxa de Erro
			0.774147	0.784858	0.793549	0.800371	0.80554	0.809316	0.811977	0.813795	0.815012	0.815828	0.816394	Vazão Efetiva
101	600	701	0.135504	0.123459	0.113838	0.106204	0.100196	0.095514	0.091902	0.089144	0.087059	0.085498	0.084335	Taxa de Erro
			0.739939	0.750249	0.758484	0.765018	0.77016	0.774168	0.77726	0.77962	0.781404	0.782741	0.783736	Vazão Efetiva
151	600	751	0.116704	0.101663	0.087422	0.074742	0.064271	0.056385	0.051072	0.047939	0.046347	0.045644	0.045343	Taxa de Erro
			0.705696	0.717713	0.72909	0.73922	0.747586	0.753887	0.758131	0.760635	0.761906	0.762468	0.762709	Vazão Efetiva

6.9. BER vs. Eb/No com 16 Pilotos e considerando aproveitamento da redundância dos bits de prefixo cíclico

A Figura 6.29 apresenta as curvas de Taxa de Erro x Eb/No considerando o aproveitamento da redundância dos bits de prefixo cíclico em enlaces OFDM.

Conforme a figura 6.29 e a Tabela 6.8, conclui-se que, para valores de Eb/No entre 5 e 12 dB, consegue-se menores taxas de erro, se considerado o aproveitamento da redundância dos bits de prefixo cíclico.

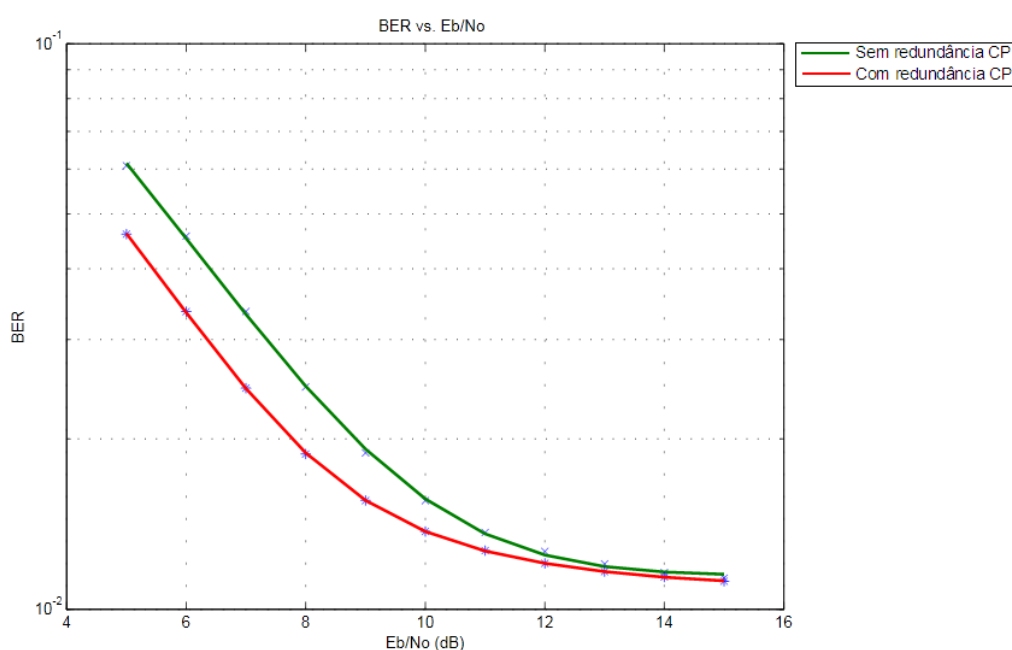


Figura 6.35 - BER vs. Eb/No considerando aproveitamento da redundância dos bits de prefixo cíclico (16 Pilotos).

Tabela 6.15 - BER vs. Eb/No considerando aproveitamento da redundância dos bits de prefixo cíclico (16 Pilotos).

16 Pilotos	Taxa de Erro	
Eb/No	Sem redundância de PC	Com redundância de PC
5	0,060849	0,046062
6	0,045718	0,033618
7	0,033644	0,024602
8	0,024721	0,018824
9	0,01891	0,015573
10	0,01556	0,013724
11	0,013667	0,012704
12	0,012658	0,012059
13	0,012027	0,011659
14	0,011616	0,011401
15	0,011351	0,011207

6.10. Sugestões de simulações e testes futuros que podem ser realizados com a Plataforma OFDM

No bloco *Bernoulli Binary*, a quantidade de amostras empregadas, por quadro (*Samples per frame*), foi igual a 600 e o período da amostra, (*Sample time*) $1/7.68 \times 10^6$ s. Uma proposição é trabalhar com quadros maiores e com outros períodos de amostra. A mudança da quantidade de amostras por quadro acarretará mudanças em outros blocos da plataforma (na maioria dos sub-blocos do *OFDM Transmitter*, no *symbol period* do AWGN, na maioria dos blocos do *OFDM Receiver*, no *LMS Channel Estimator*, no Interpolador, no *Receive delay* do *Error Rate Calculation*) e a mudança no período da amostra (*Sample time*) acarretará mudança no período do símbolo (*symbol period*) do bloco AWGN.

No bloco *Rectangular QAM*, a modulação (*M-ary number*) escolhida foi 16QAM. Uma proposição é trabalhar com outras modulações. A mudança da modulação acarretará mudanças em outros blocos da plataforma (na maioria dos sub-blocos do *OFDM Transmitter*, no *symbol period* do AWGN, na maioria dos blocos do *OFDM Receiver*, no *LMS Channel Estimator*, no Interpolador, no *Receive delay* do *Error Rate Calculation*).

No bloco *OFDM Transmitter – Zero pad for OFDM – Zero Pad* o tamanho do quadro (*Column Size*) escolhido foi 512 bits, de forma que 300 bits de dados mais a quantidade de pilotos era inferior a 512, e era preenchido por zeros de enchimento. Uma proposição é que se trabalhe, também, com uma quantidade de bits de dados e com bits de pilotos que sejam iguais ao tamanho do quadro, assim o quadro será totalmente preenchido, sem a necessidade de zeros de enchimento. A idéia, aqui, é mostrar que se aumentar muito a quantidade de pilotos, o tráfego de informações úteis, ou seja, os dados irá diminuir.

No bloco *OFDM Transmitter*, a inserção dos pilotos foi feita no domínio da frequência. Uma proposição é inserir os pilotos no domínio do tempo, depois do bloco IFFT.

No bloco *OFDM Transmitter – Zero pad for OFDM – Add Cyclic Prefix* a quantidade de bits para o prefix cíclico foi igual a 36. Uma proposição é trabalhar com outras quantidades. Quanto maior o valor para o intervalo de guarda, maior será a proteção contra os efeitos dos multipercursos e menor a capacidade de rede.

No bloco *Rayleigh Fading* trabalhou-se com máximo espalhamento Doppler (maximum Doppler shift) de 0.0213 Hz, com 4 multipercursos, com vetor de retardo (delay vector) [0 110e-9 190e-9 410e-9]s e com vetor ganho [0 -9.7 -19.2 -22.8]dB e com *Normalize gain vector to 0 dB overall gain*. Uma proposição é alterar estes parâmetros. Uma outra sugestão é variar o espalhamento Doppler e plotar curvas da BER em função do espalhamento Doppler.

No bloco AWGN, os parâmetros escolhidos foram número de bits por símbolos (*Number of bits per symbol*) $(1/2)*4$, potência do sinal de entrada (*input signal power*) igual a 1 watt e período do símbolo (*symbol period*) igual a $(1/7.68e+06)*(600/548)$ s. Uma proposição é alterar estes parâmetros.

No bloco *LMS Channel Estimator*, no sub-bloco *LMS Filter - Desired* (parâmetro: *Algorithm*), que adapta os pesos do filtro baseado no algoritmo escolhido para a filtragem do sinal de entrada, mudar o algoritmo escolhido. A idéia é mudar o estimador do canal. O Filtro LMS compara o sinal piloto recebido com o sinal piloto desejado, calcula a diferença entre estes dois sinais e, a partir daí, fornece pesos, que serão úteis ao interpolador, para que ele calcule o sinal de dados desejável e realize a equalização do sinal de dados recebido.

No bloco *S_OFDM-Interpolator* mudar o algoritmo de interpolação.

É possível mudar os parâmetros *Target number of erros* e *Maximum number of symbols* do bloco *Error Rate Calculation*.

Todas as mudanças de parâmetros na transmissão, conseqüentemente acarretarão mudanças na recepção.

Há dois principais problemas na concepção de estimadores de canal, para sistemas sem fio OFDM. O primeiro problema é a disposição das informações do piloto, ou seja, o sinal piloto de referência é usado por ambos, os transmissores e receptores. O segundo problema é a concepção de um estimador de baixa complexidade e, também, com um bom monitoramento de canal. Os dois problemas estão interligados. Em geral, o canal de sistemas OFDM, com desvanecimento, pode ser visto como um sinal bidimensional (2D) (tempo e frequência). O estimador ideal de canal em termos de erro médio quadrático (MSE - *Mean-Square Error*) é baseado em interpolação através do filtro 2D de *Wiener*. Infelizmente, uma estrutura de estimador 2D é demasiadamente complexa, para sua aplicação prática. A combinação de altas taxas de dados e baixas taxas de erros de bits, em sistemas OFDM, necessita de uso de estimadores, que tenham baixa complexidade e, também, alta precisão. As duas estimações básicas 1D de canais são estimação de canal piloto block-type e estimação de canal piloto comb-type, no qual os pilotos são inseridos no domínio da frequência e no domínio do tempo, respectivamente.

Uma sugestão é estender o estudo, para estimação de canal por piloto:

- no domínio da frequência (1D - *Block-type*):
 - *LS Estimator - Least Square Estimator* (Estimador de Quadrados Mínimos).
 - *MMSE Estimator - Minimum Mean-Square Error Estimator* (Estimador de Erro Quadrático Médio Mínimo).
 - *Modified MMSE Estimator* (Estimador MMSE Modificado).
 - *Estimation with Decision Feedback* (Estimação com Decisão de Feedback).
- no domínio do tempo (1D - *Comb-type*):
 - *LS Estimator with 1D Interpolation* (Estimador LS com Interpolação 1D):
 - *LI - Linear Interpolation* (Interpolação Linear).

- *SOI - Second-Order Interpolation* (Interpolação de Segunda Ordem).
- *LPI - Low-Pass Interpolation* (Interpolação Passa-Baixa).
- *SCI - Spline Cubic Interpolation* (Interpolação Spline Cúbico).
- *TDI - Time Domain Interpolation* (Interpolação no Domínio do Tempo).
- *ML Estimator - Maximum Likelihood Estimator* (Estimador de Possibilidade Máxima).
- *PCMB Estimator - Parametric Channel Modeling-Based Estimator* (Estimador Baseado em Modelagem de Canal Paramétrico).
- *Other Pilot-Aided Channel Estimations* (Outras Estimções de Canais Assistidas por Pilotos).
 - *Simplified 2D Estimators* (Estimadores 2D Simplificados).
 - *Iterative Channel Estimators* (Estimadores de Canais Iterativos).
 - *Channel Estimators for OFDM with Multiple Antennas* (Estimadores de Canais para OFDM com Múltiplas Antenas).

mostrando o desempenho (BER vs. E_b/N_0 - variando, também, a quantidade de pilotos) destes diversos métodos.

6.11. Conclusão

Foram apresentados e analisados os resultados das simulações, feitas na plataforma OFDM.

Foram apresentados os gráficos BER vs. E_b/N_0 , com a melhor curva de interpolação, os gráficos Vazão vs. E_b/N_0 , os resultados BER e Vazão vs. E_b/N_0 , os gráficos BER vs. Pilotos, os gráficos Vazão vs. Pilotos, os resultados BER e Vazão vs. Pilotos, os resultados Vazão Efetiva vs. E_b/N_0 e o gráfico BER vs. E_b/N_0 com 16 Pilotos e considerando aproveitamento da redundância dos bits de prefixo cíclico.

Ao analisar a curva BER vs. E_b/N_0 , com a melhor curva de interpolação, e os dados de cada piloto, separadamente, concluiu-se que quanto maior o E_b/N_0 , menor é a Taxa de Erro. A curva que mostra esta redução gradativa da Taxa de Erro, em função do E_b/N_0 , é uma exponencial decrescente.

Ao analisar a curva Vazão vs. Eb/No e os dados de cada piloto, separadamente, concluiu-se que quanto maior o Eb/No, maior é a Vazão.

Concluiu-se também que a partir de um Eb/No igual à 13 dB, o valor da Vazão fica praticamente constante, portanto as curvas assumem o formato de uma reta horizontal.

Caso o desenvolvedor de um projeto de um sistema de comunicações, que utilize a tecnologia OFDM, opte pela menor Taxa de Erro possível, ideal quando se tem dificuldades de retransmissão, ele deverá trabalhar com 21 pilotos. Esta opção não representa a opção de maior Vazão.

Caso o desenvolvedor opte pela maior Vazão possível, ele deverá trabalhar com 11 pilotos. Esta opção não representa a opção de menor Taxa de Erro possível.

O desenvolvedor pode optar por um meio termo, ou seja considerar uma Taxa de Erro média e uma Vazão média. Neste caso ele tem a seguinte alternativa: trabalhar com 13 pilotos.

Através do estudos, concluiu-se também que, para valores de Eb/No entre 5 e 12 dB, consegue-se menores taxas de erro, se for considerado o aproveitamento da redundância dos bits de prefixo cíclico.

7. CONCLUSÃO FINAL

O objetivo desta dissertação, ou seja, a análise do desempenho da tecnologia OFDM, para os sistemas móveis de quarta geração (influência da densidade de pilotos no desempenho do sistema OFDM), foi alcançado.

Todos os estudos, até o presente momento realizados, focavam no desempenho dos sistemas OFDM em função das modulações. Estudos que focassem no desempenho de sistemas OFDM em função da variação da quantidade de pilotos inserida ainda eram bastante vagos e superficiais.

O objetivo primaz deste trabalho foi justamente preencher esta lacuna. Para a concretização deste objetivo, o estudo iniciou com uma abordagem teórica, a fim de que se alcançasse o pleno entendimento da tecnologia OFDM. Feito isso, foi apresentada e explicada, detalhadamente, a plataforma OFDM, desenvolvida no MATLAB, e em seguida foram apresentados os resultados das diversas simulações realizadas nesta plataforma. Conforme visto, foram transmitidos quadros de 300 bits e a quantidade de pilotos inseridos variou consideravelmente (2, 3, 4, 5, 6, 7, 11, 13, 16, 21, 26, 31, 51, 61, 76, 101 e 151 pilotos) a fim de retratar, plenamente os efeitos da variação da quantidade de pilotos inseridos, no desempenho dos sistemas OFDM.

Os resultados obtidos foram:

- BER vs. E_b/N_0 com a melhor curva de interpolação (variando também a quantidade de pilotos).
- Vazão vs. E_b/N_0 (variando também a quantidade de pilotos).
- BER e Vazão vs. E_b/N_0 (variando também a quantidade de pilotos).
- BER vs. Pilotos.
- Vazão vs. Pilotos.
- BER e Vazão vs. Pilotos.

- Vazão Efetiva vs. E_b/N_0 (variando também a quantidade de pilotos).
- BER vs. E_b/N_0 com 16 Pilotos e considerando aproveitamento da redundância dos bits de prefixo cíclico.

A abordagem teórica abrangeu todos os elementos necessários para a compreensão da técnica baseada em multiportadoras OFDM.

- **Características dos Canais com Desvanecimento em Ambientes com Multipercursos de Raios:**

Foi vista, também, a caracterização do fenômeno de multipercurso: A dispersão do sinal rádio móvel pode ocorrer tanto no tempo quanto na frequência e tem origem, principalmente, no fenômeno de multipercurso do sinal transmitido.

Devido aos diversos espalhadores encontrados no ambiente entre o móvel e a estação transmissora, réplicas atenuadas do sinal transmitido chegam ao móvel com diferentes retardos, causando a dispersão do sinal no tempo (esta dispersão provoca interferência entre símbolos). Esta é caracterizada por parâmetros como retardo médio, espalhamento de retardo e banda de coerência. Este fenômeno é chamado de desvanecimento seletivo em frequência.

Devido à mobilidade do meio, ou da estação receptora, associada à variação dos ângulos de chegada dos multipercursos, o efeito *Doppler* resulta como a dispersão do sinal na frequência (esta dispersão provoca interferência entre portadoras) , sendo caracterizado pelos parâmetros de deslocamento *Doppler*, espalhamento *Doppler* e tempo de coerência. Este fenômeno é chamado de desvanecimento seletivo no tempo.

- **Princípios do OFDM:**

OFDM é uma forma especial de Modulação por Multiportadoras (MCM - *Multicarrier Modulation*), onde um único fluxo de dados é transmitido através de um número de subportadoras, a uma taxa mais baixa. OFDM pode ser visto tanto como uma técnica de modulação ou como uma técnica de multiplexação.

OFDM (*Orthogonal Frequency Division Multiplexing*) é uma forma de modulação do sinal por diferentes subportadoras e OFDMA (*Orthogonal Frequency Division Multiple Access*) é uma forma de agrupar sinais de diferentes fontes, utilizando a tecnologia OFDM.

Uma das principais razões para se usar OFDM é o aumento da robustez (eficiência) contra o desvanecimento seletivo em frequência (tolerância à multipercursos), eficiência contra a interferência em banda estreita e a sua alta eficiência espectral.

O aumento da robustez contra o desvanecimento seletivo em frequência ocorre pois, no OFDM, as taxas de símbolos das subportadoras são baixas, com isso a largura de banda do sinal transmitido é pequena: próxima ou menor que a banda de coerência $(\Delta f)_c$ do canal. Como o período do símbolo é o inverso do espaçamento entre frequências e sendo este espaçamento entre frequências pequeno, o período do símbolo será longo, o que permitirá a inserção de intervalo de guarda, que, como será visto adiante, é eficiente para evitar a interferência entre símbolos.

A eficiência contra a interferência em banda estreita é devido à ortogonalidade entre as subportadoras.

A alta eficiência espectral ocorre já que as formas de onda dos espectros de frequência são amplamente difundidas e sobrepostas (graças à ortogonalidade) e por isso requerem menos largura de banda. Esta alta eficiência espectral é devido, também, à possibilidade das potências das portadoras e do regime de modulação serem controlados individualmente para cada portadora.

O Sistema OFDM é muito eficaz em ambientes de multipercursos de raios. Com um único sistema transmissor (uma portadora), desvanecimento ou interferência podem causar a falha de todo o *link* (enlace), mas em um sistema com multiportadoras, apenas uma pequena percentagem de subportadoras será afetada. Um Código de Correção de Erro pode, então, ser usado para corrigir as poucas subportadoras erradas.

- **Interferências:**

- Interferência entre Portadoras (ICI)

- Causa:

- Desvanecimento Seletivo no Tempo
 - Dispersão do Sinal na Frequência

- Solução:

- IDFT e DFT
 - Prefixo Cíclico

- $\Delta t > T_c$ e $t_s < T_c$

- Interferência entre Símbolos

- Causa:
 - Desvanecimento Seletivo em Frequência
 - Dispersão do Sinal no Tempo
- Solução:
 - Modulação Multiportadoras
 - Intervalo de Guarda e Prefixo Cíclico
 - $\Delta_G > T_{\max}$ e $\Delta_G < t_s$
 - $BW < B_c$
 - $t_s > \sigma_T$
 - $\Delta f > BW_c$

- **Relações:**

Relação entre o número de subportadoras, a seletividade em frequência, e a seletividade no tempo:

- Quando o número de subportadoras aumenta, ocorre uma maior distorção devido ao desvanecimento (rápido) seletivo no tempo (ruído FM aleatório), isso por causa da maior duração do símbolo ($\Delta t < T_c$ e $t_s > T_c$). Ocorre uma maior dispersão do sinal na frequência. O espaçamento entre as subportadoras diminui, ocasionando ICI.
- Quando o número de subportadoras diminui, ocorre uma maior distorção devido ao desvanecimento seletivo em frequência, isso por causa do maior espectro de potência de cada subportadora ($\Delta f < BW_c$ e $BW > B_c$). Ocorre uma maior dispersão do sinal no tempo. A duração do símbolo diminui ($t_s < \sigma_T$), ocasionando ISI.
- No OFDM, o desempenho da transmissão é mais sensível ao desvanecimento seletivo no tempo e à dispersão do sinal na frequência.

Relação entre o comprimento do intervalo de guarda e a seletividade em frequência:

- Quando o comprimento do intervalo de guarda aumenta, o desempenho da transmissão piora, por causa da maior perda de potência (menor eficiência espectral). O ideal é que $\Delta G < t_S$.
- Quando o comprimento do intervalo de guarda diminui, o desempenho da transmissão torna-se mais sensível ao desvanecimento seletivo em frequência (dispersão do sinal no tempo), por causa do espalhamento do retardo. O ideal é que $t_S > \sigma_T$, $\Delta G < t_S$ e $\Delta G > T_{\max}$ a fim de minimizar a ISI.

- **Estimação de Canal e Alocação de Piloto:**

Para que o desempenho de um sistema OFDM possa ser avaliado, o conhecimento do canal precisa ser estimado, já que não está disponível na vida real.

Os métodos de estimação baseada em pilotos são os métodos mais comumente utilizados, que são aplicáveis em sistemas onde o remetente emite algum sinal conhecido.

Foi apresentado um modelo de sistema OFDM, que utiliza o método de estimação de canal baseado em piloto, para a equalização no receptor final.

A transmissão OFDM, também, pode ser descrita usando duas dimensões no domínio do tempo e no domínio da frequência.

Existem diferentes possibilidades para a alocação de pilotos no domínio do tempo-frequência de um sistema OFDM. Foram discutidas as três possibilidades, como ilustrado na Figura 5.4.

- Um símbolo inteiro OFDM pode ser alocado como piloto, como mostrado na Figura 5.4a (alocação de piloto espaçado no tempo e considerando todas as frequências). Tal alocação será altamente benéfica para a estimação de canal em alta dispersão do sinal no domínio do tempo ($4\Delta t$ - ISI), alto desvanecimento seletivo em frequência, baixa dispersão do sinal no domínio da frequência (Δf nulo - ICI) e baixo desvanecimento seletivo no tempo.
- Os pilotos podem ser transmitidos em subportadoras individuais durante todo o período de transmissão como mostrado na Figura 5.4b (alocação de piloto espaçado na frequência e considerando tempo integral). Essa estratégia é vantajosa para alta dispersão do sinal no domínio da frequência ($4\Delta f$ - ICI), alto desvanecimento seletivo no

tempo, baixa dispersão do sinal no domínio do tempo (Δt nulo - ISI) e baixo desvanecimento seletivo em frequência.

- Os pilotos podem ser alocados em intervalos espaçados no tempo e na frequência, conforme ilustrado na Figura 5.4c e na Figura 5.4d (alocação de piloto espaçado no tempo e espaçado na frequência). Dependendo do espaçamento do piloto no tempo-frequência e das propriedades do canal, a estratégia de alocação vai funcionar bem para (Figura 5.4c) alta dispersão do sinal no domínio da frequência ($4\Delta f$ - ICI), alto desvanecimento seletivo no tempo, moderada dispersão do sinal no domínio do tempo ($3\Delta t$ - ISI) e moderado desvanecimento seletivo em frequência ou para (Figura 5.4d) moderada dispersão do sinal no domínio da frequência ($2\Delta f$ - ICI), moderado desvanecimento seletivo no tempo, altíssima dispersão do sinal no domínio do tempo ($6\Delta t$ - ISI) e altíssimo desvanecimento seletivo em frequência.

- **Resultados da Simulação da Plataforma OFDM no MATLAB:**

Foram apresentados e analisados os resultados das simulações, feitas na plataforma OFDM.

Foram apresentados os gráficos BER vs. E_b/N_0 , com a melhor curva de interpolação, os gráficos Vazão vs. E_b/N_0 , os resultados BER e Vazão vs. E_b/N_0 , os gráficos BER vs. Pilotos, os gráficos Vazão vs. Pilotos, os resultados BER e Vazão vs. Pilotos, os resultados Vazão Efetiva vs. E_b/N_0 e o gráfico BER vs. E_b/N_0 com 16 Pilotos e considerando aproveitamento da redundância dos bits de prefixo cíclico.

Ao analisar a curva BER vs. E_b/N_0 , com a melhor curva de interpolação, e os dados de cada piloto, separadamente, concluiu-se que quanto maior o E_b/N_0 , menor é a Taxa de Erro. A curva que mostra esta redução gradativa da Taxa de Erro, em função do E_b/N_0 , é uma exponencial decrescente.

Ao analisar a curva Vazão vs. E_b/N_0 e os dados de cada piloto, separadamente, concluiu-se que quanto maior o E_b/N_0 , maior é a Vazão.

Concluiu-se também que a partir de um E_b/N_0 igual à 13 dB, o valor da Vazão fica praticamente constante, portanto as curvas assumem o formato de uma reta horizontal.

Caso o desenvolvedor de um projeto de um sistema de comunicações, que utilize a tecnologia OFDM, opte pela menor Taxa de Erro possível, ideal quando se tem dificuldades de retransmissão, ele deverá trabalhar com 21 pilotos. Esta opção não representa a opção de maior Vazão.

Caso o desenvolvedor opte pela maior Vazão possível, ele deverá trabalhar com 11 pilotos. Esta opção não representa a opção de menor Taxa de Erro possível.

O desenvolvedor pode optar por um meio termo, ou seja considerar uma Taxa de Erro média e uma Vazão média. Neste caso ele tem a seguinte alternativa: trabalhar com 13 pilotos.

Através do estudos, concluiu-se também que, para valores de E_b/N_0 entre 5 e 12 dB, consegue-se menores taxas de erro, se for considerado o aproveitamento da redundância dos bits de prefixo cíclico.

8. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] R. van Nee, and R. Prasad, *OFDM for Wireless Multimedia Communications*, Boston: Artech House, 2000.
- [2] 3GPP TR 25.892 V6.0.0, 3rd Generation Partnership Project, *Technical Specification Group Radio Access Network, Feasibility Study for Orthogonal Frequency Division Multiplexing (OFDM) for UTRAN enhancement (Release 6)*, Valbonne - France: 2004-2006.
- [3] Antonios D. Valkanas, *Adaptive space-frequency coding for MIMO-OFDM systems*, Alabama: ProQuest Information and Learning Company, 2004.
- [4] Henrik Schulze, and Christian Lüders, *Theory and Applications of OFDM and CDMA - Wideband Wireless Communications*, Germany: John Wiley & Sons, 2005.
- [5] Xin Yuan, *Comb-Type Pilot-Aided OFDM Channel Estimation for Underground WLAN Communications*, Québec, 2007.
- [6] Muhammad Saad Akram, *Pilot-based Channel Estimation in OFDM Systems*, Copenhagen: Nokia Mobile Phones, 2007.
- [7] Mary Ann Ingram, and Guillermo Acosta, *OFDM Simulation Using Matlab*, Smart Antenna Research Laboratory, 2000.
- [8] Louis Litwin and Michael Pugel, *The principles of OFDM*, Indianapolis, 2001.

- [9] Ramjee Prasad, *OFDM for Wireless Communications Systems*, Boston: Artech House, Universal Personal Communications Series, 2004.
- [10] Juha Heiskala, and John Terry, *OFDM Wireless LANs: A Theoretical and Practical Guide*, Sams.
- [11] NengWang, and Steven D. Blostein, *Adaptive Zero-Padding OFDM over Frequency-Selective Multipath Channels*, Canada: Hindawi Publishing Corporation, 2003.
- [12] Moray Rumney, *De-mystifying Single Carrier FDMA The New LTE Uplink*, Agilent Technologies, 2008.
- [13] John Kerl, *The DFT and the FFT*, 2008.
- [14] Shinsuke Hara, and Ramjee Prasad, *Multicarrier Techniques for 4g Mobile Communications*, Boston: Artech House, 2003.
- [15] Hui Liu, and Guoqing Li, *OFDM-Based Broadband Wireless Networks, Design and Optimization*, USA: Wiley Interscience, 2005.
- [16] *Some aspects of single-carrier transmission for E-UTRA*, London: Ericsson, 2005.

UNIVERSIDADE FEDERAL FLUMINENSE
ESCOLA DE ENGENHARIA
MESTRADO EM ENGENHARIA DE TELECOMUNICAÇÕES

ONÉSIMO FERREIRA

ANÁLISE DO DESEMPENHO DA TECNOLOGIA OFDM PARA OS SISTEMAS
MÓVEIS DE QUARTA GERAÇÃO

Influência da Densidade de Pilotos no Desempenho do Sistema OFDM

Vol. 2

NITERÓI
2010

ONÉSIMO FERREIRA

ANÁLISE DO DESEMPENHO DA TECNOLOGIA OFDM PARA OS SISTEMAS
MÓVEIS DE QUARTA GERAÇÃO

Influência da Densidade de Pilotos no Desempenho do Sistema OFDM

Dissertação apresentada ao Curso de Pós-Graduação em Engenharia de Telecomunicações da Universidade Federal Fluminense, como requisito parcial para obtenção do Grau de Mestre. Área de Concentração: Sistemas de Telecomunicações.

Orientador: Prof. Dr. EDUARDO RODRIGUES VALE

NITERÓI

2010

Ficha Catalográfica elaborada pela Biblioteca da Escola de Engenharia e Instituto de Computação da UFF

F383 Ferreira, Onésimo.

Análise do desempenho da tecnologia OFDM para os sistemas móveis de quarta geração: influência da densidade de pilotos no desempenho do sistema OFDM / Onésimo Ferreira. – Niterói, RJ: [s.n.], 2010.
310 f.

Dissertação (Mestrado em Engenharia de Telecomunicações) - Universidade Federal Fluminense, 2010.
Orientador: Eduardo Rodrigues Vale.

1. Sistema de comunicação móvel. 2. Multiplexação por divisão de frequência ortogonal. 3. Telefonia celular. 4. Engenharia de telecomunicação. I. Título.

CDD 621.38456

ONÉSIMO FERREIRA

ANÁLISE DO DESEMPENHO DA TECNOLOGIA OFDM PARA OS SISTEMAS
MÓVEIS DE QUARTA GERAÇÃO

Influência da Densidade de Pilotos no Desempenho do Sistema OFDM

Dissertação apresentada ao Curso de Pós-Graduação em Engenharia de Telecomunicações da Universidade Federal Fluminense, como requisito parcial para obtenção do Grau de Mestre. Área de Concentração: Sistemas de Telecomunicações.

Aprovada em Agosto de 2010.

BANCA EXAMINADORA

Prof. Dr. Eduardo Rodrigues Vale - Orientador
UFF

Prof^a. Dr^a. Leni Joaquim de Matos
UFF

Prof. Dr. Julio Cesar Rodrigues Dal Bello
UFF

Prof. Dr. José Santo Guiscafré Panaro
UFF

Prof. Dr. Eduardo Javier Arancibia Vasquez
CLARO

Prof. Dr. Marco Antonio Grivet Mattoso Maia
PUC - RJ

NITERÓI
2010

Aos meus pais.

AGRADECIMENTOS

Agradeço ao Prof. Dr. Eduardo Rodrigues Vale pela sua colaboração.

RESUMO

O objetivo desta dissertação é a análise do desempenho da tecnologia OFDM para os sistemas móveis de quarta geração. Composta de 7 capítulos, abrange todos os elementos necessários para a compreensão da técnica baseada em multiportadoras OFDM. O Capítulo 1 é introdutório, e apresenta uma breve visão geral das tecnologias de acesso celular à Internet, já existentes e emergentes. O Capítulo 2 mostra, sucintamente, as características dos canais com desvanecimento em ambientes com multipercursos de sinal. É o conhecimento básico para compreensão dos fenômenos que ocorrem nestes canais, e é essencial para a análise teórica e para a avaliação do desempenho, em canais de rádio, das técnicas baseadas em multiportadoras. O Capítulo 3 mostra a história e o princípio das técnicas baseadas em multiportadoras, incluindo a técnica OFDM. Inclui a história da origem até a forma atual. O estudo aborda o uso da transformada discreta de Fourier, a inserção de intervalo de guarda, a inserção de prefixo cíclico para a atual forma do OFDM e finaliza, mostrando os parâmetros do sistema OFDM em interface aérea LTE (*Long Term Evolution*). O Capítulo 4 discute as características dos sistemas OFDM. Apresenta um modelo do canal rádio, onde um sinal transmitido não é somente distorcido pelo desvanecimento multipercurso, mas também corrompido pelo ruído térmico. Mostra a relação entre o número de subportadoras, a seletividade em frequência e a seletividade no tempo. Mostra, também, a relação entre o comprimento do intervalo de guarda e a seletividade em frequência. Enfatiza a análise teórica e discute as vantagens e as desvantagens do sistema OFDM, incluindo a robustez contra o desvanecimento seletivo em frequência, a robustez contra os ruídos impulsivos, a sensibilidade ao deslocamento de frequência, a sensibilidade à amplificação não-linear e a sensibilidade para conversões A/D (analógico para digital) e D/A (digital para analógico). O Capítulo 5 apresenta um método de estimação de canal e as diferentes possibilidades para alocação de pilotos, tais como, alocação de piloto espaçado no tempo e considerando todas as frequências, alocação de piloto espaçado na frequência e considerando tempo integral e alocação de piloto espaçado no tempo e espaçado na frequência. Para finalizar é mostrado o sistema OFDM com símbolos pilotos utilizados para estimação de canal. O Capítulo 6 trata da parte de simulação da plataforma OFDM, explorando aspectos tais como alternativas para uma melhoria da relação sinal-ruído do sistema, e apresenta sugestões para trabalhos futuros. O Capítulo 7 apresenta as conclusões.

Palavras-chave: Multipercursos, Desvanecimento, OFDM, Estimação de Canal, Pilotos, Desempenho, BER, Vazão.

ABSTRACT

The objective of this dissertation is to analyze the performance of OFDM technology for the fourth generation mobile systems. Consisting of seven chapters, it covers all the elements necessary to understand the technique based on OFDM multicarrier. The chapter 1 is introductory, and it gives a brief overview of the technologies of mobile internet access. The chapter 2 briefly shows the characteristics of channels with fading in environments with ray multipath. It is the basic knowledge to understand the phenomena that occur in these channels, and is essential for the theoretical analysis and for the evaluation of the performance, in radio channels, of the techniques based on multicarrier. The chapter 3 shows the history and the principle of multicarrier-based techniques, including OFDM technique. Includes the history from the origin until the present shape. The study discusses the use of the discrete Fourier transform, the insertion of the guard interval, the insertion of the cyclic prefix for the current form of OFDM and ends showing the parameters of OFDM system in LTE (Long Term Evolution) air interface. The chapter 4 discusses the characteristics of OFDM systems. Presents a model of the radio channel, where a transmitted signal is not only distorted by multipath fading, but also corrupted by thermal noise. Shows the relationship between the number of subcarriers, the frequency selectivity and the selectivity in time. It also shows the relationship between the length of guard interval and the frequency selectivity. Emphasizes the theoretical analysis and discusses the advantages and the disadvantages of OFDM system, including the robustness against the frequency selective fading, the robustness against the impulsive noise, the sensitivity to frequency offset, the sensitivity to nonlinear amplifications and the sensitivity to conversions A / D (analog to digital) and D / A (digital to analog). The chapter 5 presents a method to estimate the channel and the different possibilities for allocating pilots, such as allocation of pilot spaced in time and considering all the frequencies, the allocation of pilot spaced in frequency and considering full time and the allocation of the pilot spaced in time and spaced in frequency. To finish is showed the OFDM system with pilot symbols used for channel estimation. The chapter 6 discusses the simulation of OFDM platform, exploring issues such as alternatives for improving the signal-noise ratio of the system, and presents suggestions for future works. The chapter 7 presents the conclusions.

Keywords: *Multipath, Fading, OFDM, Channel Estimation, Pilots, Performance, BER, Throughput.*

JUSTIFICATIVA E OBJETIVO

Alguns estudos têm sido feitos para analisar o desempenho de sistemas OFDM, por exemplo, Athanasios Doukas e Grigorios Kalivas (*Department of Electrical and Computer Engineering, Campus of Rion, Achaia, 26500, Greece, 2006*) analisaram a estimação do canal, em sistemas OFDM, utilizando interpolação linear. A estimação do canal e a análise do desempenho foram feitas através dos parâmetros MSE (*Mean Square Error*) e BER (*Bit Error Rate*) e caracterizadas pelo número e distância de pilotos inseridos e pelo espalhamento *Doppler*. Eles examinaram a BER e MSE, utilizando espalhamento *Doppler* de 0Hz, modulação BPSK e QPSK com 4, 5 e 6 pilotos e relação sinal-ruído (SNR) variando de 4 a 18 dB. A análise mostrou que MSE e BER são impactados pela seletividade em frequência dos canais. Mostrou, também, a importância da inserção de pilotos na eliminação de muitos erros na interpolação.

Dongxu Shen, Kai-Kit Wong, Zhengang Pan e Victor O. K. (*Department of Electrical & Electronic Engineering, the University of Hong Kong, Pokfulam Road, Hong Kong, China, 2003*) examinaram o *throughput* (vazão) em sistemas OFDM, com estimação de canais assistidos por pilotos. Especificamente, o foco da investigação era a influência da densidade de pilotos e do número de usuários no desempenho do sistema OFDM. Eles mostraram que o desempenho da BER melhora consistentemente com a inserção de mais pilotos, mas a vazão reduz devido à inserção de pilotos. O número ótimo de pilotos pode ser determinado pelo esquema de modulação, para maximizar a vazão efetiva, e em função da relação sinal-ruído (SNR). Resultados da simulação mostraram que o desempenho do sistema degrada com muitos usuários. As simulações feitas foram BER e vazão, em função das modulações QPSK, 16QAM, e 64-QAM, com espaçamento entre frequências $M_f = 4$ ou 8 e espaçamento no tempo $M_t = 3, 4$ ou 5, quantidade de usuários $U = 3$ e E_b/N_0 variando de 5 a 30 dB.

Sinem Coleri, Mustafa Ergen, Anuj Puri e Ahmad Bahai (*Department of Electrical Engineering and Computer Science at University of California, Berkeley, USA, 2002*) estudaram estimação de canal em sistemas OFDM com técnicas baseadas em pilotos (*Least Square - LS* e *Minimum Mean-Square-Error - MMSE*). Inicialmente, apresentaram um sistema OFDM em banda base com inserção de pilotos e inserção de intervalo de guarda. Abordaram inserção de pilotos no domínio do tempo e no domínio da frequência. As simulações feitas foram BER em função da relação sinal-ruído (SNR), que variava de 10 a 40 dB, e em função da modulação BPSK, QPSK, 16QAM e DPSK, com desvanecimento Rayleigh, e passando por um canal AWGN e com frequência *Doppler* de 70 Hz. Para a modulação 16QAM, foi feita a análise da BER em função da frequência *Doppler*, variando de 20 a 100Hz. Os resultados das simulações mostraram que o desempenho da interpolação com estimação de canal baseado em pilotos é muito melhor que a estimação baseada em algoritmos. As simulações também mostraram que a degradação do desempenho pode ser tolerada para altas taxas de dados com baixos valores de espalhamento *Doppler*.

Chi-Min Li, Hui-Che Chen, Po-Jen Wang, Jia-Chyi Wu e I-Tseng Tang (*Department of Electrical & Electronic Engineering, the University of Hong Kong, Pokfulam Road, Hong Kong, China, 2008*) também propuseram métodos baseados em pilotos para sincronização no tempo e estimação de canal em sistemas OFDM.

Yushi Shen e Ed Martinez (2006) apresentaram um modelo em banda base de um sistema OFDM, com divisão do espectro de frequências disponível em várias

subportadoras. O modelo consistia, basicamente, de modulação/demodulação, inserção/remoção de banda de guarda, uso de IDFT/DFT, inserção/remoção de prefixo cíclico, canal com desvanecimento por multipercursos e introdução de ruído gaussiano (AWGN). O estudo abordou estimação de canal por piloto no domínio da frequência (*Block-type*: LS, MMSE e MMSE modificado) e no domínio do tempo (*Comb-type*: LS, LI, SOI, LPI, SCI e TDI), mostrando o desempenho (SER vs. SNR) destes diversos métodos (SER - *Symbol Error Rate*).

Ali Ramadan, Tariq J. Khanzada e Abbas Omar (*Faculty of Electrical Engineering, University of Magdeburg, Germany, 2009*) apresentaram proposta para cancelamento de interferências entre subportadoras e estimação de canal utilizando interpolação de Newton e filtro de Wiener.

Todos os estudos citados acima focam no desempenho dos sistemas OFDM em função das modulações. Estudos que focam no desempenho de sistemas OFDM em função da variação da quantidade de pilotos inserida ainda são bastante vagos e superficiais. Por exemplo, nos estudos acima, apenas um deles varia a quantidade de pilotos utilizada e varia muito pouco (apenas 4, 5 e 6 pilotos), não mostrando, detalhadamente, o desempenho dos sistemas OFDM.

O objetivo deste trabalho é preencher esta lacuna, ou seja, mostrar o desempenho de sistemas OFDM em função da variação da quantidade de pilotos inserida; conforme dito no Resumo acima, a abordagem inicial será teórica, a fim de alcançar o pleno entendimento da tecnologia OFDM. Feito isso, será apresentada e explicada detalhadamente a plataforma OFDM feita no MATLAB (Apêndices A, B e C) e, em seguida, serão apresentados os resultados das diversas simulações realizadas nesta plataforma. Conforme será visto adiante, serão transmitidos quadros de 300 bits e a quantidade de pilotos inseridos variará consideravelmente (2, 3, 4, 5, 6, 7, 11, 13, 16, 21, 26, 31, 51, 61, 76, 101 e 151 pilotos), com o objetivo de retratar plenamente os efeitos da variação das quantidades de pilotos inseridos no desempenho dos sistemas OFDM.

Os resultados esperados são:

- BER vs. E_b/N_0 com a melhor curva de interpolação (variando, também, a quantidade de pilotos).
- Vazão vs. E_b/N_0 (variando, também, a quantidade de pilotos).
- BER e Vazão vs. E_b/N_0 (variando, também, a quantidade de pilotos).
- BER vs. Pilotos.
- Vazão vs. Pilotos.
- BER e Vazão vs. Pilotos.
- Vazão Efetiva vs. E_b/N_0 (variando, também, a quantidade de pilotos).
- BER vs. E_b/N_0 com 16 Pilotos e considerando aproveitamento da redundância dos bits de prefixo cíclico.

Para um aprofundamento das diversas simulações realizadas na plataforma OFDM, recomenda-se a leitura dos Apêndices, cujos objetivos são:

- Apêndice A: apresentar a plataforma OFDM desenvolvida no MATLAB (*Simulink*).
- Apêndice B: apresentar os *OFDM Transmitter* (Transmissores OFDM) e os *LMS Channel Estimator* (Estimadores de Canal LMS) das 17 Plataformas OFDM desenvolvidas, bem como o *Received Signal* (Sinal Recebido) e o *Spectrum Scope* (Escopo do Espectro), referentes a 13 pilotos. Trata-se de um aprofundamento do Apêndice A.
- Apêndice C: disponibilizar todos os programas feitos para as 17 plataformas OFDM.

SUMÁRIO

1. INTRODUÇÃO	19
1.1. Sistemas de Comunicações Móveis	19
1.2. Evolução para os Sistemas 4G	22
1.3. Técnicas de Multiportadoras para Sistemas 4G	24
1.4. 3GPP - Evolução a Longo Prazo (LTE - <i>Long Term Evolution</i>) e OFDM	25
1.4.1. <i>EUTRAN Interface de Rádio</i>	26
1.5. Conclusão	27
2. CARACTERÍSTICAS DOS CANAIS COM DESVANECIMENTO EM AMBIENTES COM MULTIPERCURSOS DE RAIOS	29
2.1. Introdução	29
2.2. Caracterização do Fenômeno de Multipercurso	29
2.3. Canais com Desvanecimento <i>Rayleigh e Rice</i>	31
2.4. Canais com Desvanecimento Seletivo em Frequência e Não Seletivo em Frequência	31
2.5. Canais com Desvanecimento Seletivo no Tempo e Não Seletivo no Tempo	31
2.6. Exemplos de Canais com Desvanecimento por Multipercursos	32
2.7. Conclusão	34
3. PRINCÍPIOS E HISTÓRIA DA MODULAÇÃO POR MULTIPORTADORAS (MCM - MULTICARRIER MODULATION) E OFDM (ORTHOGONAL FREQUENCY DIVISION MULTIPLEXING)	35
3.1. Introdução	35
3.2. Origem do OFDM	38
3.3. Uso da Transformada Discreta de Fourier	44
3.4. Inserção de prefixo cíclico para a atual forma de OFDM	48
3.5. Parâmetros do Sistema OFDM em Interface Aérea LTE (<i>Long Term Evolution</i>)	58
3.6. Conclusão	60
4. CARACTERÍSTICAS DO OFDM	62
4.1. Introdução	62
4.2. Modelo do Canal Rádio	64
4.3. Número Ótimo de Subportadoras e Comprimento Ótimo do Intervalo de Guarda	65
4.4. Robustez contra Desvanecimento Seletivo em Frequência	68
4.5. Robustez contra Ruídos Artificiais	69
4.6. Sensibilidade ao Deslocamento de Frequência	69
4.7. Sensibilidade a Amplificadores não-lineares	70
4.8. Sensibilidade para Conversões A/D e D/A	71
4.9 Conclusão	72
5. ESTIMAÇÃO DE CANAL E ALOCAÇÃO DE PILOTO	74
5.1. Introdução	74
5.2. Estimação de Canal	75
5.2.1. <i>Interpretação no Domínio do Tempo-Frequência e Alocação de Piloto</i>	78
5.3. OFDM com símbolos piloto para a estimação de canal	83
5.4. Conclusão	86
6. RESULTADOS DA SIMULAÇÃO DA PLATAFORMA OFDM NO MATLAB (SIMULINK)	87

6.1. Introdução	87
6.2. BER vs. Eb/No com a melhor curva de interpolação	87
6.3. Vazão vs. Eb/No	96
6.4. Taxa de Erro e Vazão vs. Eb/No.....	104
6.5. BER vs. Pilotos	107
6.6. Vazão vs. Pilotos.....	113
6.7. Taxa de Erro e Vazão vs. Pilotos	120
6.8. Vazão Efetiva vs. Eb/No	122
6.9. BER vs. Eb/No com 16 Pilotos e considerando aproveitamento da redundância dos bits de prefixo cíclico.....	128
6.10. Sugestões de simulações e testes futuros que podem ser realizados com a Plataforma OFDM	129
6.11. Conclusão	132
7. CONCLUSÃO FINAL.....	134
8. REFERÊNCIAS BIBLIOGRÁFICAS	141
A. APÊNDICE - PLATAFORMA OFDM NO MATLAB (<i>SIMULINK</i>)	161
B. APÊNDICE - TRANSMISSOR OFDM, ESTIMADOR DE CANAL LMS, SINAL RECEBIDO E ESCOPO DO ESPECTRO.....	236
C. APÊNDICE - PROGRAMAS FEITOS PARA A PLATAFORMA OFDM	254

LISTA DE ILUSTRAÇÕES

Figura 2.1 - Exemplo de uma transmissão com multipercursos de sinal.	30
Figura 2.2 - Exemplos de perfis de retardos de multipercursos: (a) Perfil de atraso de multipercurso com um decaimento exponencial, e (b) Perfil de atraso de multipercurso tipo i.i.d.	33
Figura 3.1 - Comparação entre SCM e MCM: (a) espectro de frequências dos sinais transmitidos; e (b) espectro de frequências dos sinais recebidos.	39
Figura 3.2 - Comparação das formas de onda transmitidas: (a) Sistema de transmissão de dados seriais de banda básica, e (b) o sistema OFDM.	43
Figura 3.3 - Comparação dos espectros de frequência: sinais ortogonais não sobrepostos de banda limitada (MCM), (b) sinais ortogonais sobrepostos limitados no tempo (MCM), (c) sinais ortogonais sobrepostos de banda limitada (MCM) e (d) um sinal limitado no tempo (SCM).	44
Figura 3.4 - Sistema OFDM: (a) transmissor e (b) receptor.	46
Figura 3.5 - FFT Básica, Transmissor e Receptor OFDM.	48
Figura 3.6 - Resposta ao impulso instantânea de um canal com desvanecimento seletivo de frequência.	49
Figura 3.7 - Sinais transmitidos: (a) não inserção de nenhum intervalo de guarda, (b) inserção de intervalo de guarda; e (c) inserção de intervalo de guarda com prefixo cíclico.	51
Figura 3.8 - Sinais Recebidos: (a) sem inserção de nenhum intervalo de guarda, (b) inserção de intervalo de guarda; e (c) inserção de intervalo de guarda com prefixo cíclico.	52
Figura 3.9 - Forma atual do OFDM: (a) e (b) técnica de extensão cíclica, (c) espectro de frequência do pulso de onda, e (d) espectro de frequência do sinal OFDM.	54
Figura 3.10 - Espectros de potência de sinais OFDM transmitidos: (a) 16 subportadoras, (b) 64 subportadoras e (c) 256 subportadoras.	56
Figura 3.11 - Estrutura do quadro genérico EUTRAN.	58
Figura 3.12 - Grade de Recursos <i>downlink</i> (estrutura do quadro genérico, prefixo cíclico normal).	59
Figura 4.1 - Modelo de Canal Rádio.	65
Figura 4.2 - Relação entre número de subportadoras, seletividade em frequência e seletividade no tempo.	67
Figura 4.3 - Relação entre o comprimento do intervalo de guarda e seletividade em frequência.	68
Figura 4.4 - Definição da resolução de conversores A/D e D/A.	71
Figura 5.1 - Modelo de um sistema OFDM baseado em piloto.	76
Figura 5.2 - Estimação do canal no domínio da frequência.	77
Figura 5.3 - Transmissão OFDM no Domínio do Tempo-Frequência (estrutura genérica do quadro, prefixo cíclico normal).	78
Figura 5.4 - Diferentes possibilidades para alocação de pilotos. (a) Alocação de piloto espaçado no tempo e considerando todas as frequências. (b) Alocação de piloto espaçado na frequência e considerando tempo integral. (c) Alocação de piloto espaçado no tempo e espaçado na frequência. (d) Alocação de piloto espaçado no tempo e espaçado na frequência.	82
Figura 5.5 - Alocação de piloto em EUTRAN (estrutura de quadro genérica, prefixo cíclico normal).	83
Figura 5.6 - Exemplo de uma grade piloto retangular.	84
Figura 5.7 - Exemplo de uma grade piloto diagonal.	85

Figura 6.1 - BER vs. Eb/No com a melhor curva de interpolação (Pilotos 5, 3, 2, 4, 6, 7, 61, 101, 76, 51, 151, 16, 26, 11, 13, 31, 21). Ordem decrescente da taxa de erro. Escala logarítmica.	89
Figura 6.2 - BER vs. Eb/No com a melhor curva de interpolação (Pilotos 5, 3, 2, 4, 6, 7, 61, 101, 76, 51, 151, 16, 26, 11, 13, 31, 21). Ordem decrescente da taxa de erro. Escala linear.	89
Figura 6.3 - BER vs. Eb/No com a melhor curva de interpolação (Pilotos 5, 3, 2, 4, 6). Ordem decrescente da taxa de erro. Escala logarítmica.	92
Figura 6.4 - BER vs. Eb/No com a melhor curva de interpolação (Pilotos 5, 3, 2, 4, 6). Ordem decrescente da taxa de erro. Escala linear.	92
Figura 6.5 - BER vs. Eb/No com a melhor curva de interpolação (Piloto 7). Escala logarítmica.	93
Figura 6.6 - BER vs. Eb/No com a melhor curva de interpolação (Piloto 7). Escala linear.	93
Figura 6.7 - BER vs. Eb/No com a melhor curva de interpolação (Pilotos 16, 26, 11, 13, 31, 21). Ordem decrescente da taxa de erro. Escala logarítmica.	94
Figura 6.8 - BER vs. Eb/No com a melhor curva de interpolação (Pilotos 16, 26, 11, 13, 31, 21). Ordem decrescente da taxa de erro. Escala linear.	94
Figura 6.9 - BER vs. Eb/No com a melhor curva de interpolação (Pilotos 61, 101, 76, 51, 151). Ordem decrescente da taxa de erro. Escala logarítmica.	95
Figura 6.10 - BER vs. Eb/No com a melhor curva de interpolação (Pilotos 61, 101, 76, 51, 151). Ordem decrescente da taxa de erro. Escala linear.	95
Figura 6.11 - Vazão vs. Eb/No (Pilotos 11, 13, 16, 21, 26, 31, 51, 61, 76, 7, 101, 6, 4, 151, 2, 3, 5). Ordem Decrescente da Vazão. Escala logarítmica.	98
Figura 6.12 - Vazão vs. Eb/No (Pilotos 11, 13, 16, 21, 26, 31, 51, 61, 76, 7, 101, 6, 4, 151, 2, 3, 5). Ordem Decrescente da Vazão. Escala linear.	98
Figura 6.13 - Vazão vs. Eb/No (Pilotos 11, 13, 16, 21, 26, 31). Ordem Decrescente da Vazão. Escala logarítmica.	101
Figura 6.14 - Vazão vs. Eb/No (Pilotos 11, 13, 16, 21, 26, 31). Ordem Decrescente da Vazão. Escala linear.	101
Figura 6.15 - Vazão vs. Eb/No (Pilotos 51, 61, 76, 7, 101). Ordem Decrescente da Vazão. Escala logarítmica.	102
Figura 6.16 - Vazão vs. Eb/No (Pilotos 51, 61, 76, 7, 101). Ordem Decrescente da Vazão. Escala linear.	102
Figura 6.17 - Vazão vs. Eb/No (Pilotos 6, 4, 151, 2, 3, 5). Ordem Decrescente da Vazão. Escala logarítmica.	103
Figura 6.18 - Vazão vs. Eb/No (Pilotos 6, 4, 151, 2, 3, 5). Ordem Decrescente da Vazão. Escala linear.	103
Figura 6.19 - BER vs. Pilotos, com Eb/No igual a 13 dB (Pilotos 5, 2, 3, 4, 6, 7, 61, 51, 101, 76, 151, 16, 11, 26, 31, 21, 13). Ordem decrescente da taxa de erro. Escala logarítmica.	108
Figura 6.20 - BER vs. Pilotos, com Eb/No igual a 13 dB (Pilotos 5, 2, 3, 4, 6, 7, 61, 51, 101, 76, 151, 16, 11, 26, 31, 21, 13). Ordem decrescente da taxa de erro. Escala linear.	109
Figura 6.21 - BER vs. Pilotos, com Eb/No igual a 13 dB (Pilotos 5, 2, 3, 4, 6, 7). Ordem decrescente da taxa de erro. Escala logarítmica.	110
Figura 6.22 - BER vs. Pilotos, com Eb/No igual a 13 dB (Pilotos 5, 2, 3, 4, 6, 7). Ordem decrescente da taxa de erro. Escala linear.	111
Figura 6.23 - BER vs. Pilotos, com Eb/No igual a 13 dB (Pilotos 16, 11, 26, 31, 21, 13). Ordem decrescente da taxa de erro. Escala logarítmica.	111

Figura 6.24 - BER vs. Pilotos, com E_b/N_0 igual a 13 dB (Pilotos 16, 11, 26, 31, 21, 13). Ordem decrescente da taxa de erro. Escala linear.	112
Figura 6.25 - BER vs. Pilotos, com E_b/N_0 igual a 13 dB (Pilotos 61, 51, 101, 76, 151). Ordem decrescente da taxa de erro. Escala logarítmica.	112
Figura 6.26 - BER vs. Pilotos, com E_b/N_0 igual a 13 dB (Pilotos 61, 51, 101, 76, 151). Ordem decrescente da taxa de erro. Escala linear.	113
Figura 6.27 - Vazão vs. Pilotos, com E_b/N_0 igual a 13 dB (Pilotos 11, 13, 16, 21, 26, 31, 51, 61, 76, 7, 101, 151, 6, 4, 2, 3, 5). Ordem decrescente de Vazão. Escala logarítmica.	115
Figura 6.28 - Vazão vs. Pilotos, com E_b/N_0 igual a 13 dB (Pilotos 11, 13, 16, 21, 26, 31, 51, 61, 76, 7, 101, 151, 6, 4, 2, 3, 5). Ordem decrescente de Vazão. Escala linear.	115
Figura 6.29 - Vazão vs. Pilotos, com E_b/N_0 igual a 13 dB (Pilotos 7, 6, 4, 2, 3, 5). Ordem decrescente de Vazão. Escala logarítmica.	117
Figura 6.30 - Vazão vs. Pilotos, com E_b/N_0 igual a 13 dB (Pilotos 7, 6, 4, 2, 3, 5). Ordem decrescente de Vazão. Escala linear.	117
Figura 6.31 - Vazão vs. Pilotos, com E_b/N_0 igual a 13 dB (Pilotos 11, 13, 16, 21, 26, 31). Ordem decrescente de Vazão. Escala logarítmica.	118
Figura 6.32 - Vazão vs. Pilotos, com E_b/N_0 igual a 13 dB (Pilotos 11, 13, 16, 21, 26, 31). Ordem decrescente de Vazão. Escala linear.	118
Figura 6.33 - Vazão vs. Pilotos, com E_b/N_0 igual a 13 dB (Pilotos 51, 61, 76, 101, 151). Ordem decrescente de Vazão. Escala logarítmica.	119
Figura 6.34 - Vazão vs. Pilotos, com E_b/N_0 igual a 13 dB (Pilotos 51, 61, 76, 101, 151). Ordem decrescente de Vazão. Escala linear.	119
Figura 6.35 - BER vs. E_b/N_0 considerando aproveitamento da redundância dos bits de prefixo cíclico (16 Pilotos).	128

LISTA DE TABELAS

Tabela 1.1 - Técnicas de acesso à interface rádio.	21
Tabela 1.2 - Padrões e tecnologias celulares.	22
Tabela 1.3 - Evolução dos sistemas de comunicações móveis em 3GPP ao longo do tempo.	26
Tabela 1.4 - Especificações da Interface Rádio LTE.	27
Tabela 3.1. Parâmetros do Sistema OFDM em EUTRAN	60
Tabela 6.1 - BER vs. Eb/No com a melhor curva de interpolação (Pilotos 2, 3, 4, 5, 6, 7, 11, 13, 16, 21, 26, 31, 51, 61, 76, 101, 151).	90
Tabela 6.2 - BER vs. Eb/No com a melhor curva de interpolação (Pilotos 5, 3, 2, 4, 6, 7, 61, 101, 76, 51, 151, 16, 26, 11, 13, 31, 21). Ordem decrescente da taxa de erro.	91
Tabela 6.3 - Vazão vs. Eb/No (Pilotos 2, 3, 4, 5, 6, 7, 11, 13, 16, 21, 26, 31, 51, 61, 76, 101, 151).	99
Tabela 6.4 - Vazão vs. Eb/No (Pilotos 11, 13, 16, 21, 26, 31, 51, 61, 76, 7, 101, 6, 4, 151, 2, 3, 5). Ordem Decrescente da Vazão.	100
Tabela 6.5 - BER vs. Eb/No com a melhor curva de interpolação e Vazão vs. Eb/No (Pilotos 2, 3, 4, 5, 6, 7, 11, 13, 16, 21, 26, 31, 51, 61, 76, 101, 151).	106
Tabela 6.6 - BER e Vazão vs. Pilotos, com Eb/No igual a 13dB.	107
Tabela 6.7 - BER vs. Pilotos, com Eb/No igual a 13 dB (Pilotos 2, 3, 4, 5, 6, 7, 11, 13, 16, 21, 26, 31, 51, 61, 76, 101, 151).	109
Tabela 6.8 - BER vs. Pilotos, com Eb/No igual a 13 dB (Pilotos 5, 2, 3, 4, 6, 7, 61, 51, 101, 76, 151, 16, 11, 26, 31, 21, 13). Ordem decrescente da taxa de erro.	110
Tabela 6.9 - Vazão vs. Pilotos, com Eb/No igual a 13 dB (Pilotos 2, 3, 4, 5, 6, 7, 11, 13, 16, 21, 26, 31, 51, 61, 76, 101, 151).	116
Tabela 6.10 - Vazão vs. Pilotos, com Eb/No igual a 13 dB (Pilotos 11, 13, 16, 21, 26, 31, 51, 61, 76, 7, 101, 151, 6, 4, 2, 3, 5). Ordem decrescente de Vazão.	116
Tabela 6.11 - BER e Vazão vs Pilotos.	122
Tabela 6.12 - Vazão Efetiva vs. Eb/No (Pilotos 2, 3, 4, 5, 6, 7, 11, 13, 16, 21, 26, 31, 51, 61, 76, 101, 151).	125
Tabela 6.13 - Vazão Efetiva vs. Eb/No (Pilotos 11, 13, 16, 21, 26, 31, 51, 61, 76, 101, 151, 7, 6, 4, 2, 3, 5). Ordem Decrescente da Vazão.	126
Tabela 6.14 - BER vs. Eb/No com a melhor curva de interpolação e Vazão Efetiva vs. Eb/No (Pilotos 2, 3, 4, 5, 6, 7, 11, 13, 16, 21, 26, 31, 51, 61, 76, 101, 151).	127
Tabela 6.15 - BER vs. Eb/No considerando aproveitamento da redundância dos bits de prefixo cíclico (16 Pilotos).	129

LISTA DE SIGLAS

- ACI (*Adjacent Channel Interference* - Interferência entre Canais Adjacentes)
- ADSL (*Asymmetric Digital Subscriber Line* - Linha Digital Assimétrica de Assinante)
- AWGN (*Additive White Gaussian Noise* - Ruído Gaussiano Branco Aditivo)
- BDMA (*Band Division Multiple Access* - Acesso Múltiplo por Divisão de Banda)
- CDMA (*Code Division Multiple Access* - Acesso Múltiplo por Divisão de Código)
- COFDM (*Code Orthogonal Frequency Division Multiplexing* - Multiplexação Ortogonal Codificada por Divisão de Frequência)
- CRM (Canal Rádio Móvel)
- DAB (*Digital Audio Broadcasting* - Radiodifusão de Áudio Digital)
- DVB (*Digital Video Broadcasting* - Radiodifusão de Vídeo Digital)
- DFT (*Discrete Fourier Transform* - Transformada Discreta de Fourier)
- DS-WCDMA (*Direct Sequence-Wideband CDMA* - CDMA de Banda Larga de Sequência Direta)
- EDGE (*Enhanced Data Rates for Global Evolution* - Melhores Taxas de Dados para Evolução Global).
- EUTRAN (*Evolved UTRAN* - UTRAN Evoluída)
- FDM (*Frequency Division Multiplexing* - Multiplexação por Divisão de Frequência)
- FDMA (*Frequency Division Multiple Access* - Acesso Múltiplo por Divisão de Frequência)
- FDD (*Frequency Division Duplex* - Duplexação por Divisão de Frequência)
- FDP (*Frequency Domain Pilot* - Piloto no Domínio da Frequência)
- FEC (*Forward Error Correction* - Correção de Erro em Repasse)
- FFT (*Fast Fourier Transform* - Transformada Rápida de Fourier)
- GPRS (*General Packet Radio Service* - Serviço de Rádio de Pacote Geral)
- GSM (*Global System for Mobile Communications* - Sistema Global para Comunicações Móveis)
- HDR (*High Data Rate* - Alta Taxa de Dados)
- HSDPA (*High Speed Downlink Packet Access* - Acesso Pacote *Downlink* de Alta Velocidade)
- HSPA (*High Speed Packet Access* - Acesso Pacote de Alta Velocidade)
- HSUPA (*High Speed Uplink Packet Access* - Acesso Pacote *Uplink* de Alta Velocidade)
- IDFT (*Inverse Discrete Fourier Transform* - Transformada Discreta de Fourier Inversa)
- IFDMA (*Interleaved FDMA* - FDMA Intercalado)
- IMT (*International Mobile Telecommunications* - Telecomunicações Móveis Internacional)
- IP (*Internet Protocol* - Protocolo de Internet)
- ISI (*Interference Intersymbol* - Interferência Intersimbólica)
- ITU (*International Telecommunication Union* - União Internacional de Telecomunicações)
- LANs (*Local Area Networks* - Redes Locais)
- LFDMA (*Localized FDMA* - FDMA Localizado)
- LO (*Local Oscillator* - Oscilador Local)
- LTE (*Long Term Evolution* - Evolução à Longo Prazo)
- ME (*Mobile Equipment* - Equipamento Móvel)
- MSE (*Mean Square Error* - Erro Quadrado Médio)
- MCM (*Multicarrier Modulation* - Modulação de Multiportadoras)

- MIMO (*Multiple Input Multiple Output* - Múltiplas Entradas Múltiplas Saídas)
- MMAC (*Multimedia Mobile Access Communications* - Comunicação Multimídia de Acesso Móvel)
- MSC (*Mobile Switching Center* - Central de Comutação de Unidade Móvel)
- OFDM (*Orthogonal Frequency Division Multiplexing* - Multiplexação Ortogonal por Divisão de Frequência)
- PANs (*Personal Area Networks* - Redes Pessoais)
- PAR (*Peak to Average Ratio* - Relação Média de Pico)
- PAPR (*Peak-to-Average Power Ratio* - Relação Média de Pico de Potência)
- PBX (*Private Branch Exchange* - Troca de Ramais Privados)
- PN (*Pseudo Noise* - Pseudo Ruído)
- PSK (*Phase Shift Keying* - Modulação por Chaveamento de Frequência)
- PSTN (*Public Switched Telephone Network* - Rede Telefônica Pública Comutada)
- PTS (*Partial Transmit Sequences* - Sequência de Transmissão Parcial)
- QAM (*Quadrature Amplitude Modulation* - Modulação de Amplitude em Quadratura)
- SC-FDMA (*Single-Carrier Frequency Division Multiple Access* - Acesso Múltiplo por Divisão de Frequência com Portadora Única)
- SCM (*Singlecarrier Modulation* - Modulação com Portadora Única)
- SER (*Symbol Error Rate* - Taxa de Erro de Símbolos)
- SLM (*Selective Mapping* - Mapeamento Seletivo)
- SOFDMA (*Scalable Orthogonal Frequency Division Multiplexing* - Multiplexação Ortogonal Escalável por Divisão de Frequência)
- STDCC (*Swept Time-Delay Cross-Correlation* - Correlação Cruzada por Varredura de Tempo de Retardo)
- SIP (*Session Initiation Protocol* - Protocolo de Iniciação de Sessão)
- SU MIMO (*Single-User Multiple Input/Multiple Output* - Múltiplas Entradas Múltiplas Saídas com Usuário Único)
- TCP (*Transmission Control Protocol* - Protocolo de Controle de Transmissão)
- TDD (*Time Division Duplex* - Duplexação por Divisão no Tempo)
- TDM (*Time Division Multiplexing* - Multiplexação por Divisão no Tempo)
- TDMA (*Time Division Multiple Access* - Acesso Múltiplo por Divisão no Tempo)
- TDP (*Time Domain Pilot* - Piloto no Domínio do Tempo)
- TE (*Terminal Equipment* - Equipamento Terminal)
- TRs (*Technical Requirements* - Requisitos Técnicos)
- TSs (*Technical Specifications* - Especificações Técnicas)
- UE (*User Equipment* - Equipamento do Usuário)
- UMTS (*Universal Mobile Telecommunications Service* - Serviço Universal de Telecomunicação Móvel)
- USIM (*Universal Subscriber Identity Module* - Módulo de Identificação Universal do Assinante)
- UTRA (*Universal Terrestrial Radio Access* - Acesso Rádio Terrestre Universal)
- WCDMA (*Wideband Code Division Multiple Access* - Acesso Múltiplo por Divisão de Código de Banda Larga)
- WLANs (*Wireless Local Area Networks* - Redes Locais sem Fio)
- WSSUS (*Wide-Sense Stationary Uncorrelated Scatering* - Canal estacionário no sentido amplo no domínio do tempo e com espalhamento descorrelacionado no domínio de retardo)
- 3GPP (*3rd Generation Partnership Project*)

A. APÊNDICE - PLATAFORMA OFDM NO MATLAB (*SIMULINK*)

A.1. Introdução

Este Capítulo tem por objetivo apresentar detalhadamente a plataforma OFDM, desenvolvida no MATLAB (*Simulink*). É apresentado um modelo em banda base de um sistema OFDM, com divisão do espectro de frequências disponível em várias subportadoras.

Os cinco elementos principais da plataforma OFDM são: Fonte de Símbolos, Transmissor OFDM, Canal Rádio Móvel, Receptor OFDM e Medidor de Taxa de Erro.

O modelo consiste, basicamente, na geração do sinal binário que será transmitido, codificação e decodificação, modulação e demodulação, inserção e remoção de pilotos no domínio da frequência, inserção e remoção dos zeros de enchimento, uso de IFFT e FFT, inserção e remoção de prefixo cíclico, canal com desvanecimento por multipercursos *Rayleigh* e introdução de ruído gaussiano (AWGN).

Ao todo, foram desenvolvidas 17 plataformas OFDM, cada uma referente aos diferentes números de pilotos (2, 3, 4, 5, 6, 7, 11, 13, 16, 21, 26, 31, 51, 61, 76, 101 e 151) considerados. Este capítulo mostra e explica a plataforma OFDM considerando 2 pilotos. As demais plataformas estão disponíveis no Apêndice B.

Foi feito um programa (disponível no Apêndice C) que solicita, para ser executada, cada uma das 17 plataformas OFDM. Cada uma destas plataformas, formada pelos blocos mostrados na Figura A.1, executa os blocos e sub-blocos disponíveis ao longo deste Capítulo e no Apêndice B e executa, também, o programa do Interpolador (disponível no Apêndice C). Cada plataforma foi executada

11 vezes, já que foram considerados no bloco AWGN, 11 valores de E_b/N_0 , de 5 a 15 dB.

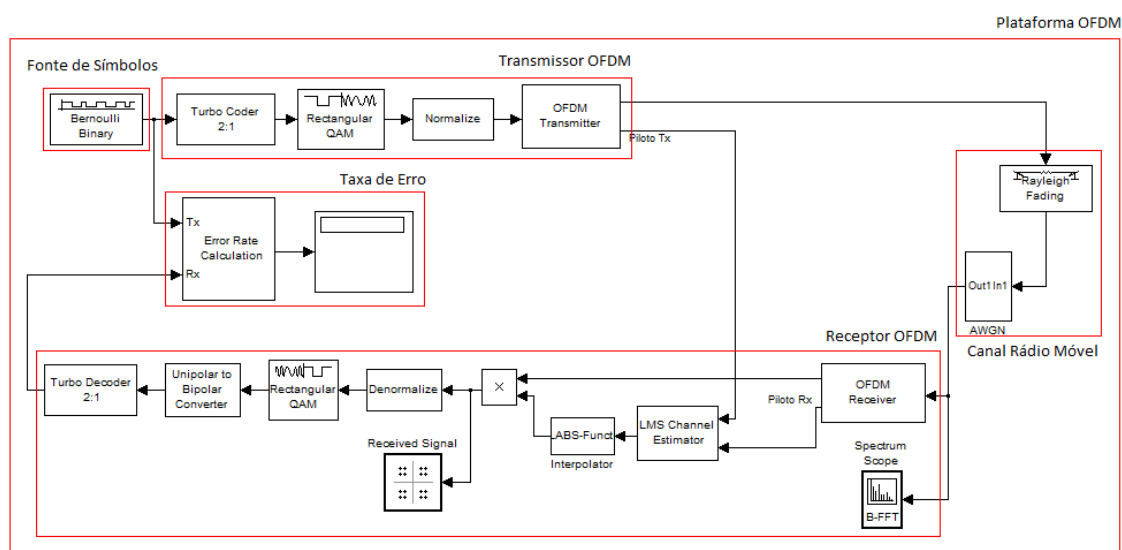


Figura A.1 - Plataforma OFDM.

A Figura A.2 mostra a simulação da plataforma OFDM considerando 2 Pilotos. Pela figura percebe-se a geração de quadros com 600 amostras. O sinal binário é codificado, e o tamanho dos quadros passa a ser de 1200 amostras. O sinal é modulado (16QAM). Como na modulação 16QAM, a quantidade de bits por símbolo são 4, o tamanho dos quadros passa a ser um quarto (1/4) de 1200 amostras, ou seja, o tamanho do quadro passa a ser de 300 amostras.

No transmissor, são inseridos 2 pilotos no domínio da frequência, portanto, os quadros passam a ter 302 amostras (dados e pilotos concatenados).

Para dois pilotos tem-se quadros com 302 amostras $\{[Dados],[Pilotos]\} = \{[2:301],[1\ 302]\}$.

Para três pilotos tem-se quadros com 303 amostras $\{[Dados],[Pilotos]\} = \{[2:151\ 153:302],[1\ 152\ 303]\}$.

Para quatro pilotos tem-se quadros com 304 amostras $\{[Dados],[Pilotos]\} = \{[2:101\ 103:202\ 204:303],[1\ 102\ 203\ 304]\}$. E assim, sucessivamente, para as demais quantidades de pilotos.

Como no transmissor o tamanho do quadro definido é de 512 amostras, a diferença entre 512 e 302, ou seja, 210, será preenchida por zeros de enchimento.

Ainda no transmissor, a IFFT transforma o espectro de frequência em um sinal no domínio do tempo, ou seja, transforma o sinal do domínio da frequência para o domínio do tempo.

Ainda no transmissor, é feita a inserção do prefixo cíclico, ou seja as últimas 36 amostras são repetidas no início de cada quadro, e os quadros passam a ter 548 amostras.

Quadros com 548 amostras serão transmitidos no Canal Rádio Móvel e sofrerão os efeitos do desvanecimento por multipercursos *Rayleigh* e da introdução de ruído gaussiano (AWGN).

No bloco *Rayleigh Fading* foram considerados 4 percursos e os seguintes parâmetros recomendados pelo LTE:

- Máximo deslocamento Doppler (*Maximum Doppler shift*) (Hz): 0.0213
- Vetor de retardo (*Delay vector*) (s): [0 110e-9 190e-9 410e-9]
- Vetor Ganho (*Gain vector*) (dB): [0 -9.7 -19.2 -22.8]
- *Normalize gain vector to 0dB overall gain*. O ganho efetivo do canal (considerando todos os caminhos) é 0 decibéis.

Já no bloco AWGN, onde é feita a introdução do ruído gaussiano, o Eb/No varia entre 5 e 15 dB.

No receptor, o quadro com 548 amostras é recebido. As 36 amostras de prefixo cíclico são removidas, portanto, o quadro passa a ter 512 amostras.

Ainda no receptor, a FFT transforma o sinal do domínio do tempo para o domínio da frequência. Os zeros de enchimento são removidos, e os quadros passam a ter 302 amostras (300 amostras de dados e 2 pilotos). Os pilotos são removidos, restando quadros com 300 amostras de dados.

No *LMS Channel Estimator* o filtro LMS irá comparar o sinal piloto recebido (*input*) com o sinal piloto desejado. Calculará a diferença entre estes dois sinais e, a partir daí, fornecerá pesos que serão úteis ao interpolador para que ele calcule o sinal de dados desejável.

O interpolador utilizará os pesos fornecidos pelo filtro LMS e, a partir disso, calculará o sinal de dados (quadros com 300 bits) desejável.

Equalização: o bloco *Product* multiplicará o sinal de dados desejável (estimado no interpolador) e o sinal de dados recebido (sinal que passou por um

canal com Desvanecimento de Multipercursos *Rayleigh* e foi adicionado Ruído Gaussiano Branco). Na saída tem-se quadros com 300 bits de dados.

Finalmente, estes quadros com 300 amostras serão demodulados (16QAM), passando a ter 1200 amostras. Os quadros com 1200 amostras serão decodificados, passando a ter 600 amostras.

O bloco *Error Rate Calculation* calcula a taxa de erro dos dados recebidos.

A quantidade de quadros é dada em função de parâmetros, que estão definidos no bloco *Error Rate Calculation*, que calcula a taxa de erro dos dados recebidos, comparando-os com uma versão atrasada da transmissão dos dados. À saída do bloco tem-se elementos, constituídos pela taxa de erro, seguido pelo número de erros detectados e o número total de símbolos comparados. Esse vetor pode ser enviado para qualquer espaço de trabalho ou porta de saída. No caso da plataforma OFDM, o vetor constituído pela taxa de erro está sendo enviado para o *Display* (mostrador numérico) e para o *Sink Block Parameters (Signal To Workspace)* a fim de possibilitar a plotagem dos gráficos (que serão disponibilizados adiante).

O parâmetro “*Stop simulation*” (do bloco *Error Rate Calculation*) interrompe a simulação ao detectar um número alvo pré-definido de erros ou um número máximo de símbolos.

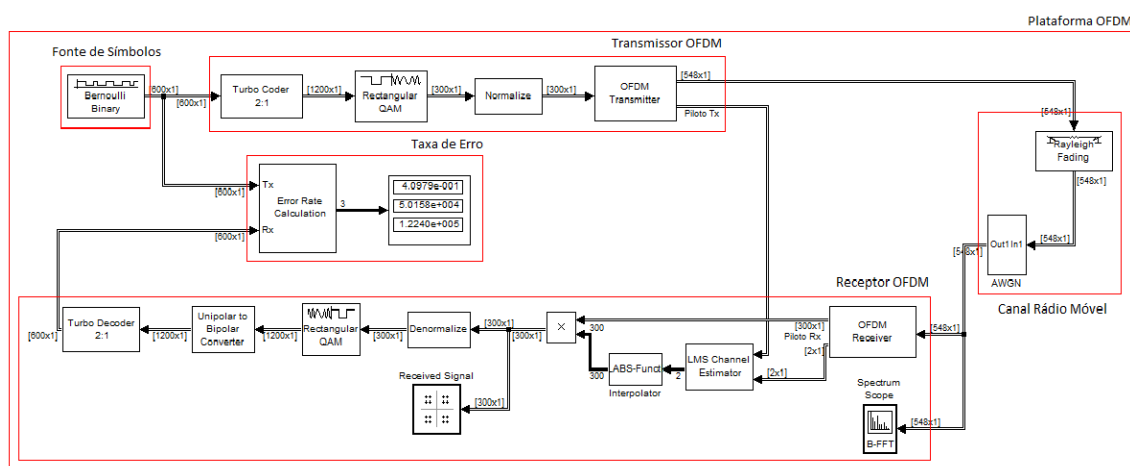


Figura A.2 - Simulação da plataforma OFDM com 2 Pilotos.

Para o completo entendimento dos resultados alcançados, nas diversas simulações realizadas nas 17 plataformas OFDM, e que serão apresentados no

próximo capítulo, não se faz necessária a leitura deste capítulo. Portanto, torna-se facultativo para o leitor a leitura deste capítulo.

A.1.1. Bernoulli Binary

Gerador Binário de Bernoulli: Gera um número binário aleatório (da distribuição) de Bernoulli.

O bloco Gerador Binário de Bernoulli gera um número binário aleatório usando a distribuição de Bernoulli. A distribuição de Bernoulli com parâmetro p produz zero com probabilidade p e um com probabilidade $1-p$. A distribuição de Bernoulli tem valor médio $1-p$ e variância $p(1-p)$. A probabilidade de um parâmetro zero especifica p , e pode ser algum número real entre zero e um.

Parâmetros:

- Probabilidade de ocorrer um zero (*Probability of a zero*): 0,5
- Valor inicial (*Initial seed*): 12343
- Período da amostra (*Sample time*): $1/7.68e+06$
- Saídas baseadas em quadro (*Frame-based outputs*)
- Amostras por quadro (*Samples per frame*): 600
- Tipo do dado de saída (*Output data type*): *double*

O número de elementos em *Initial seed* e o parâmetro Probabilidade de ocorrer um zero iniciam (definem) o número de colunas numa saída baseada em quadro ou o número de elementos em um vetor de saída baseado em amostra. Também o formato (linha ou coluna) dos parâmetros *Initial seed* e Probabilidade de ocorrer um zero definem o formato de um sinal de saída bidimensional baseado em amostra (*sample-based*).

Probabilidade de ocorrer um zero (*Probability of a zero*): a probabilidade com que uma saída zero ocorre.

Valor Inicial (*Initial seed*): O valor inicial para o gerador do número aleatório. O valor inicial pode ser tanto um vetor de mesmo comprimento como a probabilidade do parâmetro zero, ou um escalar.

Período da amostra (*Sample time*): O período de cada vetor baseado em amostra ou cada linha de uma matriz baseada em quadro.

Saídas baseadas em quadro (*Frame-based outputs*): Determina se a saída é baseada em quadro ou baseada em amostra.

Amostras por quadro (*Samples per frame*): O número de amostras em cada coluna de um sinal de saída baseado em quadro. Este campo só está ativo se forem verificadas saídas baseadas em quadro.

Tipo do dado de saída (*Output data type*): O tipo de saída do bloco pode ser especificado como *boolean*, *int8*, *uint8*, *int16*, *uint16*, *int32*, *uint32*, *single*, or *double*. Por padrão, o bloco está definido como *double*.

A Tabela A.1 e a Figura A.3 mostram a quantidade de *frames* utilizada em função do número de pilotos.

A quantidade de frames é dada em função de parâmetros, que estão definidos no bloco *Error Rate Calculation*, que calcula a taxa de erro dos dados recebidos, comparando-os com uma versão atrasada da transmissão dos dados. A saída do bloco são elementos, constituídos pela taxa de erro, seguido pelo número de erros detectados e o número total de símbolos comparados. Esse vetor pode ser enviado para qualquer espaço de trabalho ou porta de saída. No caso da plataforma OFDM, o vetor constituído pela taxa de erro está sendo enviado para o *Display* (mostrador numérico) e para o *Sink Block Parameters (Signal To Workspace)* a fim de que possibilite a sua visualização e plotagem dos gráficos que utilizam como dado a taxa de erro (que serão disponibilizados adiante).

O parâmetro “*Stop simulation*” (do bloco *Error Rate Calculation*) interrompe a simulação ao detectar um número alvo pré-definido de erros ou um número máximo de símbolos.

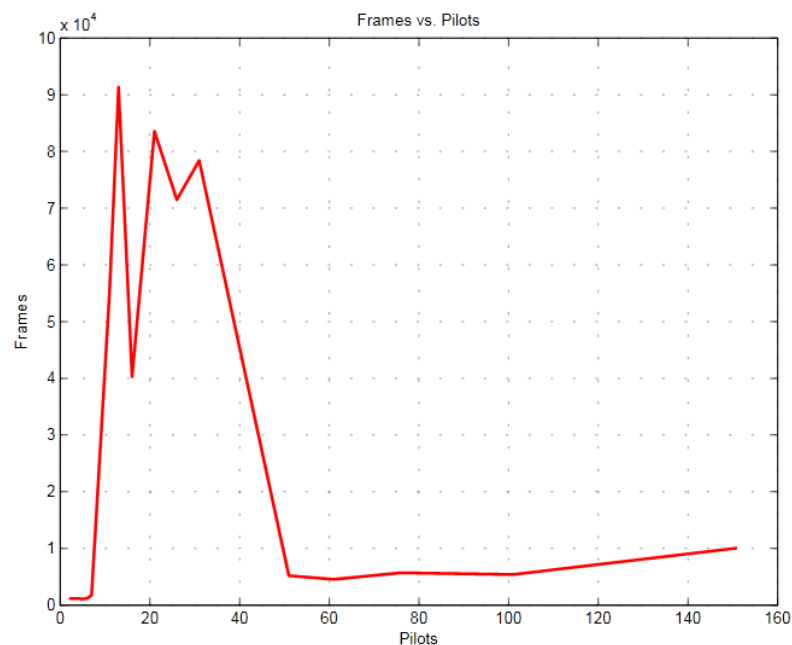
Parâmetros:

- Número de erros estabelecido (*Target number of errors*): 50.000
- Número máximo de símbolos (*Maximum number of symbols*): 10^8

A interrupção da simulação é dada quando qualquer um dos dois parâmetros acima for atingido primeiro.

Tabela A.1 - Quantidade de *frames* utilizada em função do número de pilotos.

Pilotos	Frames
2	1146
3	1146
4	1179
5	1069
6	1206
7	1716
11	55031
13	91358
16	40311
21	83543
26	71493
31	78414
51	5207
61	4543
76	5694
101	5415
151	10084

Figura A.3 - Quantidade de *frames* utilizada em função do número de pilotos.

Conforme observado na Tabela A.1 e na Figura A.3, as plataformas OFDM com quantidades de pilotos iguais à 11, 13, 16, 21, 26 e 31 apresentam uma maior quantidade de *frames*, isso significa que estas quantidades de pilotos precisam de uma quantidade maior de *frames* para chegar ao número de erros ou ao número total de símbolos pré-estabelecidos e comparados.

Sinal de Saída: quadros com 600 amostras (número binário aleatório de Bernoulli).

A.1.2. Turbo Coder 2:1

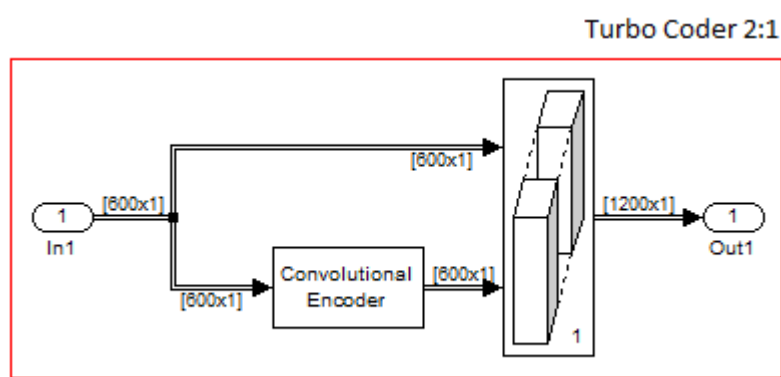


Figura A.4 - Turbo Coder 2:1.

A.1.2.1 In 1

Inport: Fornece uma porta de entrada para o subsistema *Turbo Coder 2:1*.

Parâmetros podem ser usados para especificar explicitamente os atributos do sinal de entrada.

Parâmetro:

- Número da porta (*Port number*): 1

Sinal de Entrada: quadros com 600 amostras (número binário aleatório de Bernoulli).

Sinal de Saída: quadros com 600 amostras (número binário aleatório de Bernoulli).

A.1.2.2. Convolutional Encoder

Codificador convolucional (*Convolutional Encoder*): codifica dados binários convolucionalmente. Usa a função *poly2trellis* (comprimento de restrição, polinômio gerador de código (em número octal) e polinômio de conexão de *feedback* (em octal)).

O bloco do codificador convolucional codifica uma sequência de vetores binários de entrada para produzir uma sequência de vetores binários de saída. Este bloco pode processar múltiplos símbolos ao mesmo tempo.

Parâmetros:

- *Trellis structure: poly2trellis(4,[15],13)*
- Modo de Operação (*Operation mode*): *Truncated (reset every frame)*.

Trellis structure: estrutura do MATLAB que contém a descrição do codificador convolucional.

Truncated reset every frame mode: o bloco trata cada quadro independentemente, isto é, o estado do codificador é “resetado” (zerado) no início de cada quadro.

Sinal de Entrada: quadros com 600 amostras (número binário aleatório de Bernoulli).

Sinal de Saída: quadros com 600 amostras codificados convolucionalmente.

A.1.2.3. 1-Matrix Concatenate

Concatenar: Concatena sinais de entrada com mesmo tipo de dados para criar um sinal de saída contíguo.

O bloco Concatenar concatena os sinais em suas entradas para criar um sinal de saída cujos elementos residem em locais contíguos na memória. Este bloco opera em vetor ou modo de concatenação de matriz multidimensional, dependendo da configuração do seu parâmetro *Mode*. Em ambos os casos, as entradas são concatenadas de cima para baixo ou da esquerda para a direita, portas de entrada.

Parâmetro:

- Número de entradas (*Number of inputs*): 2
- Modo (*Mode*): Matriz multidimensional (*Multidimensional array*)

- Dimensão de concatenação (*Cocanetate dimension*): 1

Dimensão de concatenação (*Cocanetate dimension*): Especifica a dimensão de saída ao longo do qual concatena as matrizes de entrada. Por exemplo, para concatenar as matrizes de entrada vertical ou horizontalmente, deve-se especificar 1 ou 2, respectivamente. Esta opção só aparece quando selecionada a matriz multidimensional como o parâmetro *mode*.

Sinal de Entrada: quadros com 600 amostras (número binário aleatório de Bernoulli) e quadros com 600 amostras codificados convolucionalmente.

Sinal de Saída: quadros com 1200 amostras (sendo 600 amostras números binários aleatórios de Bernoulli e 600 amostras codificadas convolucionalmente).

A.1.2.4. Out1

Outport: Fornece uma porta de saída para o subsistema *Turbo Coder 2:1*.

Parâmetros podem ser usados para especificar explicitamente os atributos do sinal de entrada.

Parâmetro:

- Número da porta (*Port number*): 1

Sinal de Entrada: quadros com 1200 amostras (sendo 600 amostras números binários aleatórios de Bernoulli e 600 amostras codificadas convolucionalmente).

Sinal de Saída: quadros com 1200 amostras (sendo 600 amostras números binários aleatórios de Bernoulli e 600 amostras codificadas convolucionalmente).

A.1.3. Rectangular QAM

Modulador Banda Base QAM Retangular: Modula o sinal de entrada usando o método de modulação em amplitude de quadratura retangular.

O valor do número *M*-ary deve ser uma potência inteira de dois.

A entrada pode ser bits ou inteiros. No caso de entrada de bit baseada em quadro, a largura da entrada deve ser igual a um múltiplo inteiro do número de bits por símbolos. (No caso de entrada de bit baseada em amostra, a largura da entrada deve ser igual ao número de bits por símbolo).

Para entrada de inteiros baseada em quadro, a entrada deve ser um vetor coluna. (Para entrada de inteiros baseada em amostra, a entrada deve ser um escalar).

No caso de entrada baseada em quadro, a largura do quadro de entrada deve ser igual ao produto do número de símbolos e o valor de amostras por símbolo.

(No caso de entrada baseada em amostra, o tempo da amostra de saída deve ser igual ao período do símbolo dividido pelo valor de amostras por símbolo.)

Modulação: 16 QAM.

Parâmetros:

- *M-ary number*: 16 (Número de bits por símbolo: 4)
- Tipo de Entrada (*Input type*): Bit (igual à amostra)
- *Constellation ordering*: Gray
- Entradas baseadas em quadro (*Frame-based inputs*)
- Método de normalização (*Normalization method*): *Min. Distance between symbols*
- Distância mínima (*Minimum distance*): 1

Sinal de entrada: quadros com 1200 amostras (sendo 600 amostras números binários aleatórios de Bernoulli e 600 amostras codificadas convolucionalmente).

Sinal de saída: Sinal modulado, quadros com 300 amostras.

A.1.4. Normalize

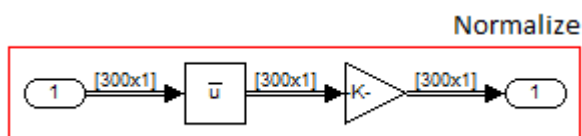


Figura A.5 - Normalizador (*Normalize*).

A.1.4.1. 1-Inport

Inport: Fornece uma porta de entrada para o subsistema *Normalize*.

Sinal de entrada: Sinal modulado, quadros com 300 amostras.

Sinal de saída: Sinal modulado, quadros com 300 amostras.

A.1.4.2. Math - u

Math: Funções matemáticas, incluindo logaritmo, exponencial, potência e módulo.

Parâmetro:

- *Function: conj*

conj: complexo conjugado.

Sinal de entrada: Sinal modulado, quadros com 300 amostras.

Sinal de saída: Sinal modulado, quadros com 300 amostras.

A.1.4.3. Gain - k

Ganho (*Gain*): *Element – wise gain* ($y = K.*u$) ou *matrix gain* ($y = K*u$ or $y = u*K$).

Multiplica a entrada por uma constante. A entrada e o ganho podem ser um escalar, vetor ou matriz.

A entrada e o ganho são multiplicados, e o resultado é convertido para o tipo de dados de saída utilizando o arredondamento especificado.

Parâmetros:

- Ganho (*Gain*): $1/\text{sqrt}(10)$
- Multiplicação (*Multiplication*): *Element – wise gain* ($y = K.*u$)

Multiplicação: Especifica o modo de multiplicação (elemento ou matriz). Elemento-wise ($K.*u$) - Cada elemento de entrada é multiplicado por cada elemento do ganho. Se necessário, o bloco realiza expansões, de modo que a entrada e o ganho tenham as mesmas dimensões.

sqrt: *square root* – raiz quadrada.

Sinal de entrada: Sinal modulado, quadros com 300 amostras.

Sinal de saída: Sinal modulado, quadros com 300 amostras.

A.1.4.4. Outport – 1

Outport: Fornece uma porta de saída para o subsistema *Normalize*.

Sinal de entrada: Sinal modulado, quadros com 300 amostras.

Sinal de saída: Sinal modulado, quadros com 300 amostras.

A.1.5. OFDM Transmitter (2 Pilotos)

A Figura A.6 representa um transmissor OFDM (*OFDM Transmitter*) considerando 2 pilotos, os transmissores OFDM para as demais quantidades de pilotos (3, 4, 5, 6, 7, 11, 13, 16, 21, 26, 31, 51, 61, 76, 101, 151) encontram-se no Apêndice B.

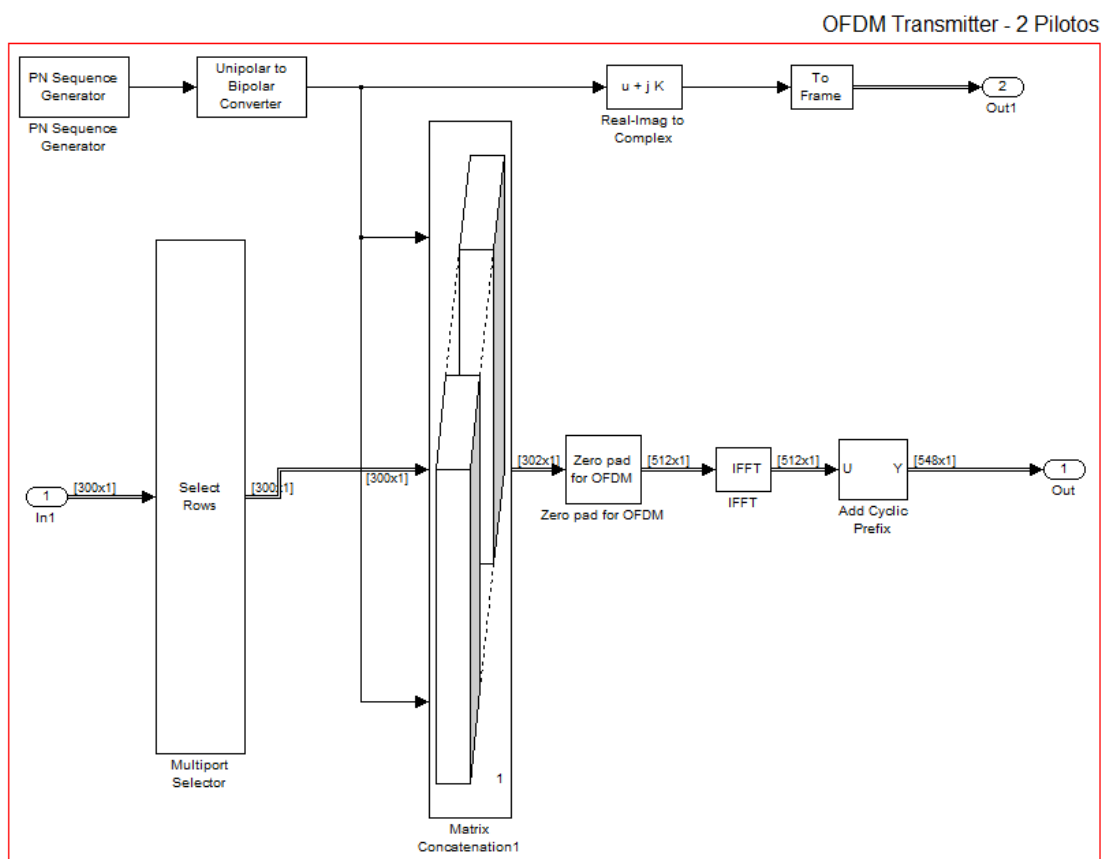


Figura A.6 - Transmissor OFDM (*OFDM Transmitter*) considerando 2 pilotos.

A.1.5.1. PN Sequence Generator (para geração de pilotos)

PN Sequence Generator: Gera uma sequência de ruído pseudo-aleatório (PN), utilizando um registrador de deslocamento com realimentação linear (LFSR - *Linear Feedback Shift Register*). O LFSR é implementado usando um gerador de registro de deslocamento simples (SSRG - *Simple Shift Register Generator*, ou Fibonacci - XOR).

Parâmetros:

- Polinômio gerador (*Generator polynomial*): [1 0 0 1 0 0 0 1]

- Estado inicial (*Initial states*): [1 1 1 1 1 1 1]

Polinômio gerador (*Generator polynomial*): Polinômio que determina as conexões de feedback dos registradores de deslocamento.

Estado inicial (*Initial states*): Vetor de estados iniciais dos registradores de deslocamento.

Sinal de saída: Uma sequência de ruído pseudo-aleatório (PN).

A.1.5.2. Unipolar to Bipolar Converter

Conversor Unipolar para Bipolar: Converte um sinal unipolar no intervalo [0, M-1], onde M é o M-ário número, em um sinal bipolar.

O bloco Conversor Unipolar para Bipolar mapeia o sinal de entrada unipolar para um sinal de saída bipolar. Se a entrada consiste de números inteiros entre 0 e M-1, onde M é o parâmetro M-ário número, então a saída é composto de números inteiros entre - (M-1) e M-1. Se M é par, então a saída é ímpar. Se M é ímpar, então a saída é par. Este bloco é projetado para funcionar apenas quando o valor de entrada está dentro do conjunto (0,1,2 ... (M-1)), onde M é o parâmetro número M-ário. Se o valor da entrada está fora deste conjunto de números inteiros a saída não pode ser válida.

Parâmetros:

- *M-ary number*: 2
- Polaridade (*Polarity*): Negativa (*Negative*)
- Tipo do dado de saída (*Output data type*): Igual ao dado de entrada (*Same as input*)

M-ary number: O número de símbolos no alfabeto bipolar ou unipolar.

Polaridade: Um valor positivo faz com que o bloco mantenha a ordem relativa dos símbolos no alfabeto. Um valor negativo faz com que o bloco inverta a ordem relativa dos símbolos no alfabeto.

A Tabela A.2 mostra como o mapeamento do bloco depende do parâmetro de polaridade.

Tabela A.2 - Mapeamento do bloco em função do parâmetro de polaridade.

Valor do Parâmetro Polaridade	Saída correspondente para o valor de entrada k
Positivo	$2k-(M-1)$
Negativo	$-2k+(M-1)$

Sinal de entrada: Uma sequência de ruído pseudo-aleatório (PN).

Sinal de saída: Sinal bipolar.

A.1.5.3. $u + jK$ – *Real-Imag to Complex*

Real – Imaginário para Complexo: constrói uma saída complexa a partir de uma entrada real e/ou imaginária.

Parâmetros:

- Entrada (*Input*): real
- Parte imaginária (*Imag part*): 0

Sinal de entrada: Sinal bipolar real.

Sinal de saída: Sinal bipolar complexo.

A.1.5.4. *To Frame*

Frame Status Conversion: Especifica o status do quadro do sinal de saída como baseado em quadro ou baseado em amostra.

Parâmetro:

- Sinal de saída (*Output signal*): Baseado em quadro (*Frame-based*).

Sinal de entrada: Sinal bipolar complexo.

Sinal de saída: Sinal bipolar complexo baseado em quadro.

A.1.5.5. 2 - Out1

Outport: Fornece uma porta de saída para o subsistema que gerou os pilotos.

Sinal de entrada: Sinal bipolar complexo baseado em quadro (pilotos).

Sinal de saída: Sinal bipolar complexo baseado em quadro (pilotos).

Este piloto (*desired*) será comparado com o piloto (*input*) que chegará na recepção. Esta comparação será feita no LMS Channel Estimator. Maiores detalhes será visto posteriormente.

A.1.5.6. In1

Inport: Fornece uma porta de entrada para o subsistema Transmissor.

Sinal de entrada: Sinal modulado, quadros com 300 amostras.

Sinal de saída: Sinal modulado, quadros com 300 amostras.

A.1.5.7. Select Rows – Multiport Selector

Seletor de Múltiplas Portas (*Multiport Selector*): saída especificada de linhas ou colunas para uma ou mais portas de saída. O número de portas de saída é determinado pelo número de vetores de índice, cada um especificado como um vetor separado de entrada em uma matriz de células.

Distribui subconjuntos arbitrários de linhas ou colunas de entrada para várias portas de saída.

Parâmetros:

- Seleção (*Select*): Linhas (*Rows*)
- Índices de saída (*Indices to output*): {1:300}
- Índice inválido (*Invalid index*): Gerador de erro (*Generator Error*).

Seleção (*Select*): Seleciona a dimensão da entrada, linhas ou colunas.

Índices de saída (*Indices to output*): Uma matriz de células especificando subconjuntos de linha ou coluna para propagar a cada uma das portas de saída. O número de células na matriz determina o número de portas de saída do bloco.

Índice inválido: Resposta a um valor de índice inválido (linha inexistente).

Com 2 pilotos tem-se índice de saída (*Indices to output*) igual à {1:300}, que representa uma matriz de uma célula, que gera uma saída independente (com uma porta distinta). O parâmetro {1:300} também representa as linhas 1, 2, ..., 300 da matriz de entrada e comprimento de saída (*output size*) 300-by-N.

Com 3 pilotos tem-se índices de saída (*Indices to output*) igual à {1:150,151:300}, que representam uma matriz de duas células, que gera duas

saídas independentes (cada uma em uma porta distinta). O parâmetro {1:150,151:300} também representa as linhas 1, 2, ..., 150 e 151, 152, ..., 300 da matriz de entrada e comprimento de saída (*output size*) 300-by-N.

Com 4 pilotos tem-se índices de saída (*Indices to output*) igual à {1:100,101:200, 201:300}, que representam uma matriz de três células, que gera três saídas independentes (cada uma em uma porta distinta). O parâmetro {1:100,101:200, 201:300} também representa as linhas 1, 2, ..., 100; 101, 102, ..., 200 e 201, 202, ..., 300 da matriz de entrada e comprimento de saída (*output size*) 300-by-N.

O raciocínio vale para as demais quantidades de pilotos (5, 6, 7, 11, 13, 16, 21, 26, 31, 51, 61, 76, 101, 151).

A Tabela A.3 apresenta a quantidade de portas de saída do bloco em função da quantidade de pilotos.

Tabela A.3 - *Select Rows*.

Divisores	Select Rows	Pilotos	300/Select Rows
2	150	151	2
3	100	101	3
4	75	76	4
5	60	61	5
6	50	51	6
10	30	31	10
12	25	26	12
15	20	21	15
20	15	16	20
25	12	13	25
30	10	11	30
50	6	7	50
60	5	6	60
75	4	5	75
100	3	4	100
150	2	3	150
300	1	2	300

Na Tabela A.3 a coluna “*Select Rows*” representa o número de células na matriz, que determina o número de portas de saída do bloco (número de subportadoras).

Já a coluna “*300/Select Rows*” representa a quantidade de bits que serão transmitidos em cada uma das portas.

A Tabela A.4 apresenta o parâmetro “Índices de saída” (*Indices to output*) para todas as diferentes quantidades de pilotos consideradas.

Tabela A.4 - Índices de saída (*Indices to output*).

Pilotos	OFDM Transmitter
	Select Rows
Pilotos	Indices to Output
2	{1:300}
3	{1:150,151:300}
4	{1:100,101:200,201:300}
5	{1:75,76:150,151:225,226:300}
6	{1:60,61:120,121:180,181:240,241:300}
7	{1:50,51:100,101:150,151:200,201:250,251:300}
11	{1:30,31:60,61:90,91:120,121:150,151:180,181:210,211:240,241:270,271:300}
13	{1:25,26:50,51:75,76:100,101:125,126:150,151:175,176:200,201:225,226:250,251:275,276:300}
16	{1:20,21:40,41:60,61:80,81:100,101:120,121:140,141:160,161:180,181:200,201:220,221:240,241:260,261:280,281:300}
21	{1:15,16:30,31:45,46:60,61:75,76:90,91:105,106:120,121:135,136:150,151:165,166:180,181:195,196:210,211:225,226:240,241:255,256:270,271:285,286:300}
26	{1:12,13:24,25:36,37:48,49:60,61:72,73:84,85:96,97:108,109:120,121:132,133:144,145:156,157:168,169:180,181:192,193:204,205:216,217:228,229:240,241:252,253:264,265:276,277:288,289:300}

31	{1:10,11:20,21:30,31:40,41:50,51:60,61:70,71:80,81:90,91:100,101:110,111:120,121:130,131:140,141:150,151:160,161:170,171:180,181:190,191:200,201:210,211:220,221:230,231:240,241:250,251:260,261:270,271:280,281:290,291:300}
51	{1:6,7:12,13:18,19:24,25:30,31:36,37:42,43:48,49:54,55:60,61:66,67:72,73:78,79:84,85:90,91:96,97:102,103:108,109:114,115:120,121:126,127:132,133:138,139:144,145:150,151:156,157:162,163:168,169:174,175:180,181:186,187:192,193:198,199:204,205:210,211:216,217:222,223:228,229:234,235:240,241:246,247:252,253:258,259:264,265:270,271:276,277:282,283:288,289:294,295:300}
61	{1:5,6:10,11:15,16:20,21:25,26:30,31:35,36:40,41:45,46:50,51:55,56:60,61:65,66:70,71:75,76:80,81:85,86:90,91:95,96:100,101:105,106:110,111:115,116:120,121:125,126:130,131:135,136:140,141:145,146:150,151:155,156:160,161:165,166:170,171:175,176:180,181:185,186:190,191:195,196:200,201:205,206:210,211:215,216:220,221:225,226:230,231:235,236:240,241:245,246:250,251:255,256:260,261:265,266:270,271:275,276:280,281:285,286:290,291:295,296:300}
76	{1:4,5:8,9:12,13:16,17:20,21:24,25:28,29:32,33:36,37:40,41:44,45:48,49:52,53:56,57:60,61:64,65:68,69:72,73:76,77:80,81:84,85:88,89:92,93:96,97:100,101:104,105:108,109:112,113:116,117:120,121:124,125:128,129:132,133:136,137:140,141:144,145:148,149:152,153:156,157:160,161:164,165:168,169:172,173:176,177:180,181:184,185:188,189:192,193:196,197:200,201:204,205:208,209:212,213:216,217:220,221:224,225:228,229:232,233:236,237:240,241:244,245:248,249:252,253:256,257:260,261:264,265:268,269:272,273:276,277:280,281:284,285:288,289:292,293:296,297:300}
101	{1:3,4:6,7:9,10:12,13:15,16:18,19:21,22:24,25:27,28:30,31:33,34:36,37:39,40:42,43:45,46:48,49:51,52:54,55:57,58:60,61:63,64:66,67:69,70:72,73:75,76:78,79:81,82:84,85:87,88:90,91:93,94:96,97:99,100:102,103:105,106:108,109:111,112:114,115:117,118:120,121:123,124:126,127:129,130:132,133:135,136:138,139:141,142:144,145:147,148:150,151:153,154:156,157:159,160:162,163:165,166:168,169:171,172:174,175:177,178:180,181:183,184:186,187:189,190:192,193:195,196:198,199:201,202:204,205:207,208:210,211:213,214:216,217:219,220:222,223:225,226:228,229:231,232:234,235:237,238:240,241:243,244:246,247:249,250:252,253:255,256:258,259:261,262:264,265:267,268:270,271:273,274:276,277:279,280:282,283:285,286:288,289:291,292:294,295:297,298:300}
151	{1:2,3:4,5:6,7:8,9:10,11:12,13:14,15:16,17:18,19:20,21:22,23:24,25:26,27:28,29:30,31:32,33:34,35:36,37:38,39:40,41:42,43:44,45:46,47:48,49:50,51:52,53:54,55:56,57:58,59:60,61:62,63:64,65:66,67:68,69:70,71:72,73:74,75:76,77:78,79:80,81:82,83:84,85:86,87:88,89:90,91:92,93:94,95:96,97:98,99:100,101:102,103:104,105:106,107:108,109:110,111:112,113:114,115:116,117:118,

	119:120,121:122,123:124,125:126,127:128,129:130,131:132,133:134,135:136, 137:138,139:140,141:142,143:144,145:146,147:148,149:150,151:152,153:154, 155:156,157:158,159:160,161:162,163:164,165:166,167:168,169:170,171:172, 173:174,175:176,177:178,179:180,181:182,183:184,185:186,187:188,189:190, 191:192,193:194,195:196,197:198,199:200,201:202,203:204,205:206,207:208, 209:210,211:212,213:214,215:216,217:218,219:220,221:222,223:224,225:226, 227:228,229:230,231:232,233:234,235:236,237:238,239:240,241:242,243:244, 245:246,247:248,249:250,251:252,253:254,255:256,257:258,259:260,261:262, 263:264,265:266,267:268,269:270,271:272,273:274,275:276,277:278,279:280, 281:282,283:284,285:286,287:288,289:290,291:292,293:294,295:296,297:298, 299:300}
--	--

Sinal de entrada: Sinal modulado, quadros com 300 amostras (bits).

Sinal de saída: Sinal modulado, quadros com 300 amostras. A coluna “300/Select Rows” da Tabela A.3 apresenta a quantidade de bits que serão transmitidos em cada uma das portas.

6.1.5.8. Matrix Concatenation 1 (para inserção de pilotos no domínio da frequência)

Concatenar: Concatena sinais de entrada do mesmo tipo de dados para criar um sinal de saída contíguo.

O bloco Concatenar concatena os sinais em suas entradas para criar um sinal de saída cujos elementos residem em locais contíguos na memória. Este bloco opera em modo vetor ou modo de concatenação de matriz multidimensional, dependendo da configuração do seu parâmetro *Mode*. Em ambos os casos, as entradas são concatenados de cima para baixo ou da esquerda para a direita.

Parâmetros:

- Número de entradas (*Number of inputs*): 3 (Para 2 Pilotos)
- Modo (*Mode*): Matriz multidimensional (*Multidimensional array*)
- Dimensão da concatenação (*Concatenate dimension*): 1

Número de entradas (*Number of inputs*): Número de entradas do bloco.

Modo (*Mode*): Especifica o tipo de concatenação realizada por este bloco. As opções são Matriz multidimensional (*Multidimensional array*) ou Vetor (*Vector*).

Dimensão da concatenação (*Concatenate dimension*): Especifica a dimensão de saída ao longo do qual concatena as matrizes de entrada. Por exemplo, para

concatenar as matrizes de entrada vertical ou horizontalmente, deve-se especificar 1 ou 2, respectivamente. Esta opção só aparece se for selecionado matriz Multidimensional para o parâmetro modo.

Tabela A.5 - Dados do Transmissor.

Divisores	Select Rows	Pilotos	Dados	Dados + Pilotos	(Dados + Pilotos)/2	Arred ((Dados + Pilotos / 2))	(Column size = 512) Column size - Arred((Dados + Pilotos / 2))
300	1	2	300	302	151	151	361
150	2	3	300	303	151.5	152	360
100	3	4	300	304	152	152	360
75	4	5	300	305	152.5	153	359
60	5	6	300	306	153	153	359
50	6	7	300	307	153.5	154	358
30	10	11	300	311	155.5	156	356
25	12	13	300	313	156.5	157	355
20	15	16	300	316	158	158	354
15	20	21	300	321	160.5	161	351
12	25	26	300	326	163	163	349
10	30	31	300	331	165.5	166	346
6	50	51	300	351	175.5	176	336
5	60	61	300	361	180.5	181	331
4	75	76	300	376	188	188	324
3	100	101	300	401	200.5	201	311
2	150	151	300	451	225.5	226	286

A Tabela A.5, na coluna “Dados + Pilotos”, mostra a quantidade de Dados e Pilotos concatenados, para as diferentes quantidades de pilotos consideradas.

Tabela A.6 - Dados do Transmissor.

	OFDM Transmitter	OFDM Transmitter	OFDM Transmitter	OFDM Transmitter
	Zero pad for OFDM	Zero pad for OFDM	Zero pad for OFDM	Matriz Concatenation 1
	Zero pad	Selector - Index vector	Selector 1 - Index vector	Number of Inputs
Pilotos	Column size		[Dado+P Zero D+Piloto]	Select Rows + Pilotos
2	512	[152:512 1:151]	[362:512 152:361 1:151]	3
3	512	[153:512 1:152]	[361:512 153:360 1:152]	5
4	512	[153:512 1:152]	[361:512 153:360 1:152]	7
5	512	[154:512 1:153]	[360:512 154:359 1:153]	9
6	512	[154:512 1:153]	[360:512 154:359 1:153]	11
7	512	[155:512 1:154]	[359:512 155:358 1:154]	13
11	512	[157:512 1:156]	[357:512 157:356 1:156]	21
13	512	[158:512 1:157]	[356:512 158:355 1:157]	25
16	512	[159:512 1:158]	[355:512 159:354 1:158]	31
21	512	[162:512 1:161]	[352:512 162:351 1:161]	41
26	512	[164:512 1:163]	[350:512 164:349 1:163]	51
31	512	[167:512 1:166]	[347:512 167:346 1:166]	61
51	512	[177:512 1:176]	[337:512 177:336 1:176]	101
61	512	[182:512 1:181]	[332:512 182:331 1:181]	121
76	512	[189:512 1:188]	[325:512 189:324 1:188]	151
101	512	[202:512 1:201]	[312:512 202:311 1:201]	201
151	512	[227:512 1:226]	[287:512 227:286 1:226]	301

A Tabela A.6, na coluna “OFDM Transmitter - Matriz Concatenation 1- Number of Inputs - Select Rows + Pilotos”, apresenta o número de entradas do bloco Matriz concatenation, para as diferentes quantidades de pilotos consideradas. Quanto maior a quantidade de pilotos, maior será a quantidade de entradas do bloco.

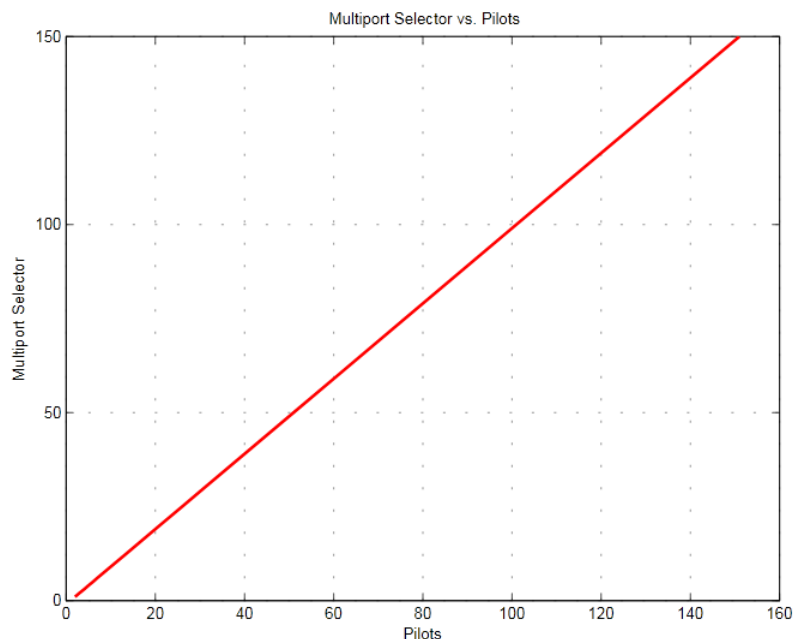


Figura A.7 - Multiport selector vs. Pilots.

Considerando a inserção de dois pilotos tem-se $\{[Dados],[Pilotos]\} = \{[2:301],[1\ 302]\}$.

Considerando a inserção de três pilotos tem-se $\{[Dados],[Pilotos]\} = \{[2:151\ 153:302],[1\ 152\ 303]\}$.

Considerando a inserção de quatro pilotos tem-se $\{[Dados],[Pilotos]\} = \{[2:101\ 103:202\ 204:303],[1\ 102\ 203\ 304]\}$.

O raciocínio vale para as demais quantidades de pilotos (5, 6, 7, 11, 13, 16, 21, 26, 31, 51, 61, 76, 101, 151).

Sinal de entrada: Sinal modulado, quadros com 300 amostras.

Sinal de saída: Sinal modulado, quadros com 300 amostras mais 2 Pilotos. Dados e Pilotos concatenados.

A.1.5.9. Zero pad for OFDM (para inserção de zeros de enchimento)

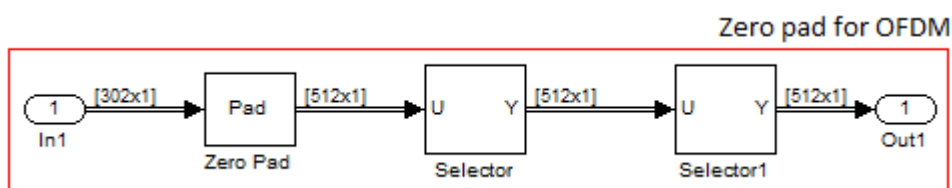


Figura A.8 - Zero pad for OFDM.

A.1.5.9.1. 1 - In1

Inport: Fornece uma porta de entrada para o subsistema *Zero pad for OFDM*.

Sinal de entrada: Sinal modulado, quadros com 302 amostras. Dados e Pilotos concatenados.

Sinal de saída: : Sinal modulado, quadros com 302 amostras. Dados e Pilotos concatenados.

A.1.5.9.2. Pad – Zero Pad

Pad: Especifica um valor constante para a entrada ao longo das dimensões especificadas.

O bloco Pad estende (aumenta) ou encurta (diminui) as dimensões da entrada através de enchimento ou truncagem ao longo de suas colunas, linhas, colunas e linhas, ou de qualquer dimensão(s) que for especificada. Truncamento ocorre quando as dimensões de saída forem especificadas mais curtas do que as dimensões de entrada correspondente. A entrada e saída podem também ter o mesmo comprimento (*pass-through*).

O parâmetro sinal Pad (*Pad signal*) controla se as dimensões de entrada especificadas são preenchidas ou truncadas em seu início, fim, ou ambos.

Parâmetros:

- *Pad over: Columns*
- Valor Pad (*Pad value*): 0
- Comprimento da Coluna (*Column size*): 512
- *Pad signal at: end*

A Tabela A.7, na coluna “Zeros de enchimento” mostra a quantidade de zeros de enchimento em função da quantidade de pilotos.

Tabela A.7 - Zeros de enchimento.

Divisores	Select Rows	Pilotos	Dados	Dados + Pilotos	Column size	Zeros de Enchimento
300	1	2	300	302	512	210
150	2	3	300	303	512	209
100	3	4	300	304	512	208
75	4	5	300	305	512	207
60	5	6	300	306	512	206
50	6	7	300	307	512	205
30	10	11	300	311	512	201
25	12	13	300	313	512	199
20	15	16	300	316	512	196
15	20	21	300	321	512	191
12	25	26	300	326	512	186
10	30	31	300	331	512	181
6	50	51	300	351	512	161
5	60	61	300	361	512	151
4	75	76	300	376	512	136
3	100	101	300	401	512	111
2	150	151	300	451	512	61

Pela Tabela A.7, conclui-se que quanto maior a quantidade de pilotos, menor será a quantidade de zeros de enchimento.

Notar que a quantidade de bits de dados mais pilotos é menor que 512. A diferença entre 512 e a quantidade de bits de dados mais pilotos são os zeros de enchimento. Se a quantidade de bits de dados mais pilotos fosse igual à 512, e se ainda aumentasse a quantidade de pilotos, deveria-se diminuir a quantidade de dados, para que os 512 bits pudessem suportar todos os bits de dados e pilotos, o que implicaria numa redução da transmissão de bits úteis (de dados). Para que a taxa de bits de dados não diminuísse, uma solução seria aumentar o parâmetro Column size de 512 para 1024 bits.

Tabela A.8 - Pilotos, dados, zeros de enchimento, prefixo cíclico, quadro total.

Pilotos	Dados	Zeros de Enchimento	CP	Column size	Pilotos (%)	Dados (%)	Zeros de Enchimento (%)	CP (%)	Column size (%)
2	300	210	36	548	0,0036496	0,547445255	0,383211679	0,0657	1
3	300	209	36	548	0,0054745	0,547445255	0,381386861	0,0657	1
4	300	208	36	548	0,0072993	0,547445255	0,379562044	0,0657	1
5	300	207	36	548	0,0091241	0,547445255	0,377737226	0,0657	1
6	300	206	36	548	0,0109489	0,547445255	0,375912409	0,0657	1
7	300	205	36	548	0,0127737	0,547445255	0,374087591	0,0657	1
11	300	201	36	548	0,020073	0,547445255	0,366788321	0,0657	1
13	300	199	36	548	0,0237226	0,547445255	0,363138686	0,0657	1
16	300	196	36	548	0,0291971	0,547445255	0,357664234	0,0657	1
21	300	191	36	548	0,0383212	0,547445255	0,348540146	0,0657	1
26	300	186	36	548	0,0474453	0,547445255	0,339416058	0,0657	1
31	300	181	36	548	0,0565693	0,547445255	0,330291971	0,0657	1
51	300	161	36	548	0,0930657	0,547445255	0,29379562	0,0657	1
61	300	151	36	548	0,1113139	0,547445255	0,275547445	0,0657	1
76	300	136	36	548	0,1386861	0,547445255	0,248175182	0,0657	1
101	300	111	36	548	0,1843066	0,547445255	0,202554745	0,0657	1
151	300	61	36	548	0,2755474	0,547445255	0,111313869	0,0657	1

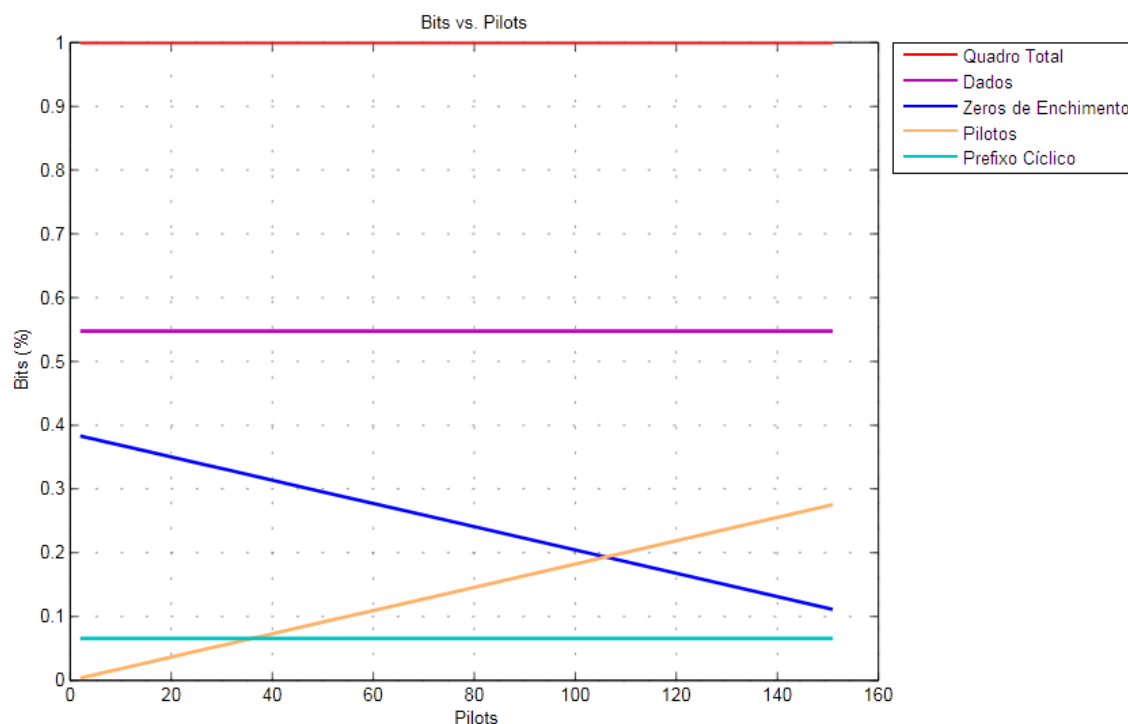


Figura A.9 - Pilotos, dados, zeros de enchimento, prefixo cíclico, quadro total.

Sinal de entrada: Sinal modulado, quadros com 302 amostras. Dados e pilotos concatenados.

Sinal de saída: Sinal modulado, quadros com 512 amostras, sendo 302 amostras dados e pilotos concatenados e 210 amostras zeros de enchimento.

A.1.5.9.3. Selector

Selector: Seleciona ou reordena elementos específicos de um sinal de entrada multidimensional. O índice de cada elemento é identificado por uma porta de entrada ou caixa de diálogo.

Parâmetros:

- Dimensão do número de entrada (*Number of input dimensions*): 1
- Modo de indexação (*Index mode*): *One-based*
- Opção de Índice (*Index option*): *Index vector*
- Índice (*Index*): [152:512 1:151] (2 Pilotos)
- *Input port size*: 512

Para 2 pilotos, tem-se como índice: [Dados+Pilotos+Zeros Dados+Pilotos] = [152:512 1:151].

Para 3 pilotos, tem-se como índice: [Dados+Pilotos+Zeros Dados+Pilotos] = [153:512 1:152]

Para 4 pilotos, tem-se como índice: [Dados+Pilotos+Zeros Dados+Pilotos] = [153:512 1:152]

A Tabela A.6, na coluna “OFDM Transmitter - Zero pad for OFDM - Selector - Index vector”, tem-se os índices em função da quantidade de pilotos.

Sinal de entrada: Sinal modulado, quadros com 512 amostras, sendo 302 amostras dados e pilotos concatenados e 210 amostras zeros de enchimento.

Sinal de saída: Sinal modulado reordenado, quadros com 512 amostras, sendo 302 amostras dados e pilotos concatenados e 210 amostras zeros de enchimento.

A.1.5.9.4. Selector 1

Selector: Seleciona ou reordena elementos específicos de um sinal de entrada multidimensional. O índice de cada elemento é identificado por uma porta de entrada ou caixa de diálogo.

Parâmetros:

- Dimensão do número de entrada (*Number of input dimensions*): 1
- Modo de indexação (*Index mode*): *One-based*
- Opção de Índice (*Index option*): *Index vector*
- Índice (Index): [362:512 152:361 1:151] (2 Pilotos)
- *Input port size*: 512

Para 2 pilotos, tem-se como índice: [Dados+Pilotos Zeros Dados+Pilotos] = [362:512 152:361 1:151]

Para 3 pilotos, tem-se como índice: [Dados+Pilotos Zeros Dados+Pilotos] = [361:512 153:360 1:152]

Para 4 pilotos, tem-se como índice: [Dados+Pilotos Zeros Dados+Pilotos] = [361:512 153:360 1:152]

A Tabela A.6, na coluna “OFDM Transmitter - Zero pad for OFDM - Selector 1 - Index vector”, tem-se os índices em função da quantidade de pilotos.

Sinal de entrada: Sinal modulado reordenado, quadros com 512 amostras, sendo 302 amostras dados e pilotos concatenados e 210 amostras zeros de enchimento.

Sinal de saída: : Sinal modulado novamente reordenado, quadros com 512 amostras, sendo 302 amostras dados e pilotos concatenados e 210 amostras zeros de enchimento.

A.1.5.9.5. 1 - Out 1

Outport: Fornece uma porta de saída para o subsistema *Zero pad for OFDM*.

Sinal de entrada: : Sinal modulado novamente reordenado, quadros com 512 amostras, sendo 302 amostras dados e pilotos concatenados e 210 amostras zeros de enchimento.

Sinal de saída: : Sinal modulado novamente reordenado, quadros com 512 amostras, sendo 302 amostras dados e pilotos concatenados e 210 amostras zeros de enchimento.

A.1.5.10. IFFT

IFFT: A Saída é a *Inverse Fast Fourier Transform* (IFFT) de uma entrada real ou complexa. As saídas são reais, se selecionado '*Input is conjugate symmetric*', caso contrário, as saídas são complexas.

Calcula a transformada inversa (rápida) de Fourier (IFFT) da entrada.

Transforma o sinal do domínio da frequência para o domínio do tempo. A IFFT transforma o espectro de frequência em um sinal no domínio do tempo.

Sinal de entrada: : Sinal modulado novamente reordenado, quadros com 512 amostras, sendo 302 amostras dados e pilotos concatenados e 210 amostras zeros de enchimento. Domínio da frequência.

Sinal de saída: : Sinal modulado novamente reordenado, quadros com 512 amostras, sendo 302 amostras dados e pilotos concatenados e 210 amostras zeros de enchimento. Domínio do tempo.

A.1.5.11. Add Cyclic Prefix

Selector: Seleciona ou reordena elementos específicos de um sinal de entrada multidimensional. O índice de cada elemento é identificado por uma porta de entrada ou caixa de diálogo. Inserção de prefixo cíclico.

Parâmetros:

- Dimensão do número de entrada (*Number of input dimensions*): 1
- Modo de indexação (*Index mode*): *One-based*
- Opção de Índice (*Index option*): *Index vector*
- Índice (*Index*): [477:512 1:512]
- *Input port size*: 512

Índice (*Index*): [Prefixo cíclico Dados+Zeros+Pilotos] = [477:512 1:512]

Tabela A.9 - Prefixo Cíclico.

OFDM Transmitter	512	512
Add Cyclic Prefix	476	36
[477:512 1:512]	36	548

Sinal de entrada: Sinal modulado novamente reordenado, quadros com 512 amostras, sendo 302 amostras dados e pilotos concatenados e 210 amostras zeros de enchimento. Domínio do tempo.

Sinal de saída: Sinal modulado novamente reordenado, quadros com 548 amostras, sendo 302 amostras dados e pilotos concatenados, 210 amostras zeros de enchimento e 36 amostras prefixo cíclico. Domínio do tempo.

A.1.5.12. 1 Out

Outport: Fornece uma porta de saída para o subsistema Transmissor OFDM.

Sinal de entrada: : Sinal modulado novamente reordenado, quadros com 512 amostras, sendo 302 amostras dados e pilotos concatenados e 210 amostras zeros de enchimento. Domínio do tempo.

Sinal de saída: : Sinal modulado novamente reordenado, quadros com 548 amostras, sendo 302 amostras dados e pilotos concatenados, 210 amostras zeros de enchimento e 36 amostras prefixo cíclico. Domínio do tempo.

A.1.6. Rayleigh Fading

Canal com Desvanecimento de Multipercursos *Rayleigh* (*Multipath Rayleigh Fading Channel*): Canal com Desvanecimento de Multipercursos *Rayleigh* para sinais complexos de banda base.

O número de caminhos é igual ao comprimento de cada parâmetro 'Delay vector' ou 'Gain vector'. Foram considerados quatro percursos.

O bloco canal com multipercursos e desvanecimento Rayleigh implementa uma simulação em banda base de um canal de propagação com multipercursos. Este bloco é útil para a modelagem de sistemas de comunicações móveis sem fio.

Este bloco só aceita em sua entrada sinais complexos baseados em quadro.

O sinal de entrada deve ter um período de amostragem discreta superior a 0.

O movimento relativo entre o transmissor e o receptor causa deslocamentos Doppler na frequência do sinal. O PSD Jakes (densidade espectral de potência) determina o espectro do processo de Rayleigh.

Uma vez que um canal com multipercursos reflete sinais em vários lugares, um sinal transmitido viaja para o receptor por diferentes caminhos que podem ter diferentes comprimentos e, portanto, diferentes intervalos de tempo associados. O desvanecimento ocorre quando os sinais viajam por caminhos diferentes e interferem uns com os outros. O parâmetro vetor *Delay* especifica o tempo de atraso para cada caminho. Se o vetor de ganho de 0 dB normalizado para ganho global estiver desmarcado, então o vetor de ganho especifica o ganho de cada caminho. Se a caixa estiver marcada, então o bloco usa um múltiplo do vetor de ganho, em vez do vetor de ganho próprio, é escolhido um fator de escala de modo que o ganho efetivo do canal, considerando todos os caminhos é 0 dB.

O número de percursos é o comprimento do vetor *Delay* ou vetor Ganho, o que for maior. Se ambos os parâmetros são vetores, então eles devem ter o mesmo comprimento.

O bloco multiplica o sinal de entrada por amostras de um processo aleatório complexo de uma distribuição Rayleigh. O valor escalar inicial é o parâmetro para o gerador de números aleatórios. Os números aleatórios são gerados usando o método Ziguarte, que é o mesmo método usado pela função randn MATLAB.

Parâmetros recomendados pelo LTE:

- Máximo deslocamento Doppler (*Maximum Doppler shift*) (Hz): 0.0213
- Vetor de retardo (*Delay vector*) (s): [0 110e-9 190e-9 410e-9]
- Vetor Ganho (*Gain vector*) (dB): [0 -9.7 -19.2 -22.8]
- *Normalize gain vector to 0dB overall gain*
- Valor Inicial (*Initial seed*): 73

Máximo deslocamento Doppler (*Maximum Doppler shift*): Um escalar positivo que indica o desvio Doppler máximo.

Vetor de retardo (*Delay vector*): Um vetor que especifica o atraso de propagação para cada caminho.

Vetor Ganho (*Gain vector*): Um vetor que especifica o ganho de cada caminho.

Normalize gain vector to 0dB overall gain: Normaliza o vetor ganho de 0 dB como ganho global. O ganho efetivo do canal (considerando todos os caminhos) é 0 decibéis.

Valor Inicial (*Initial seed*): Valor escalar para o gerador de ruído gaussiano.

Sinal de entrada: Sinal modulado novamente reordenado, quadros com 548 amostras, sendo 302 amostras dados e pilotos concatenados, 210 amostras zeros de enchimento e 36 amostras prefixo cíclico. Domínio do tempo.

Sinal de saída: Sinal modulado novamente reordenado, quadros com 548 amostras, sendo 302 amostras dados e pilotos concatenados, 210 amostras zeros de enchimento e 36 amostras prefixo cíclico. Domínio do tempo. O sinal passou por um canal com Desvanecimento de Multipercursos Rayleigh.

Para caracterizar o desvanecimento por multipercursos, foram definidos 4 percursos.

As Figuras A.10 e A.22 apresentam a resposta ao impulso, a magnitude e o retardo em cada um destes 4 percursos, para amostras com 548 e 1060 bits, respectivamente.

As Figuras A.15 e A.27 apresentam o ganho (perda) nos multipercursos, para amostras com 548 e 1060 bits, respectivamente.

Amostras com 548 bits.

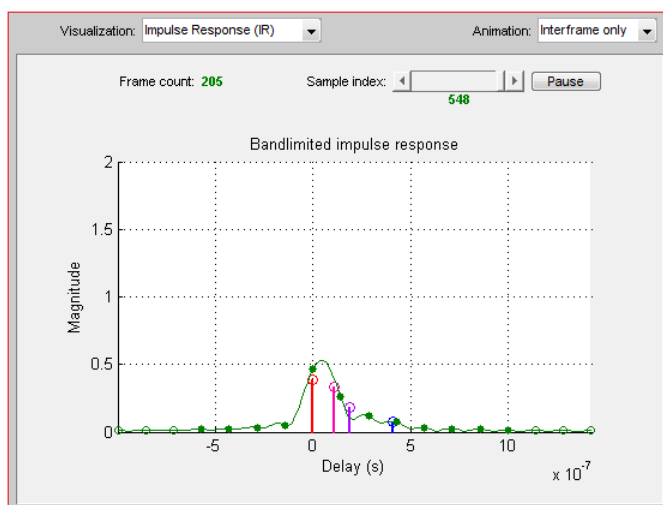


Figura A.10 - Resposta ao Impulso (IR - *Impulse Response*)

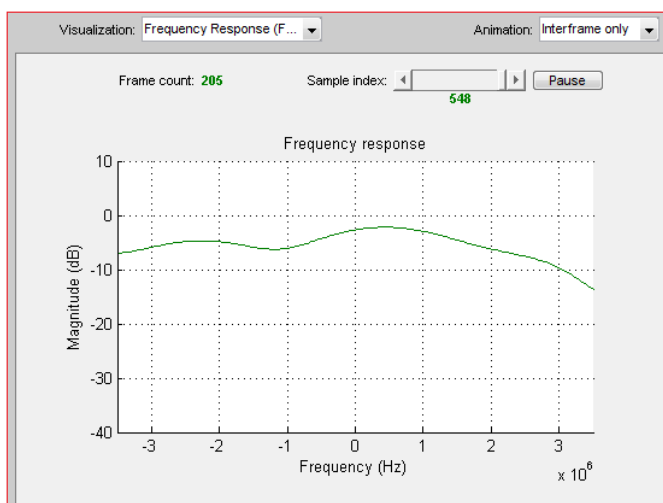


Figura A.11 - Resposta a Frequência (FR - *Frequency Response*)

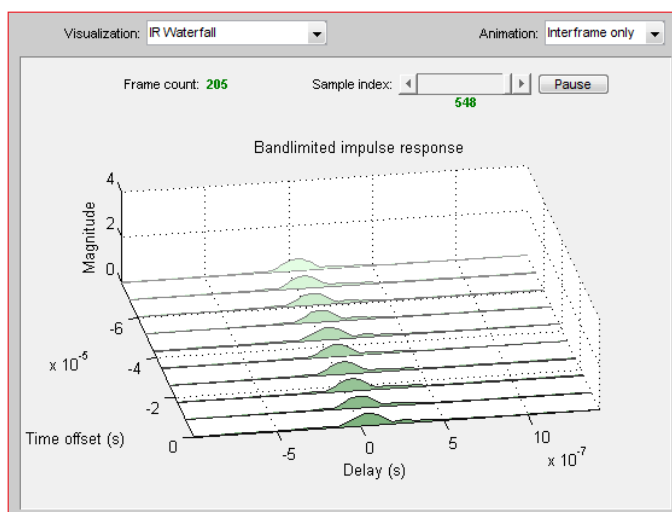


Figura A.12 - IR Waterfall.

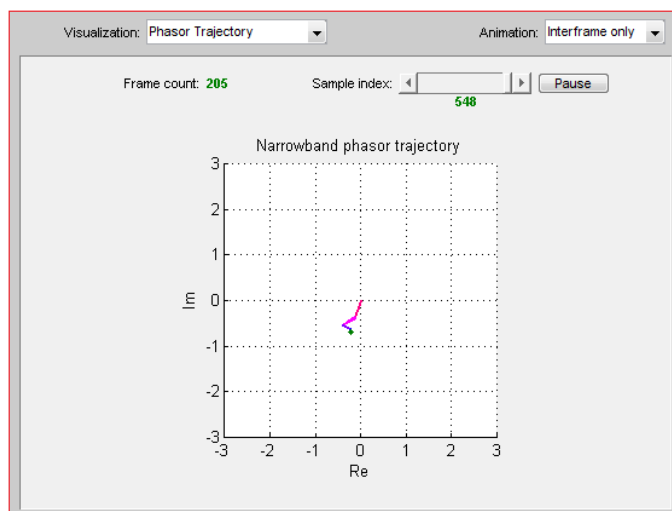


Figura A.13 - Trajetória do Fasor (*Phasor Trajectory*).

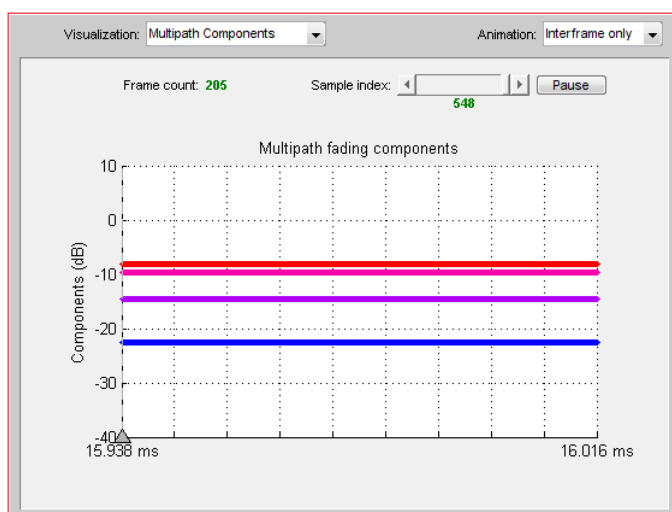


Figura A.14 - Componentes Multipercursos (*Multipath Component*).

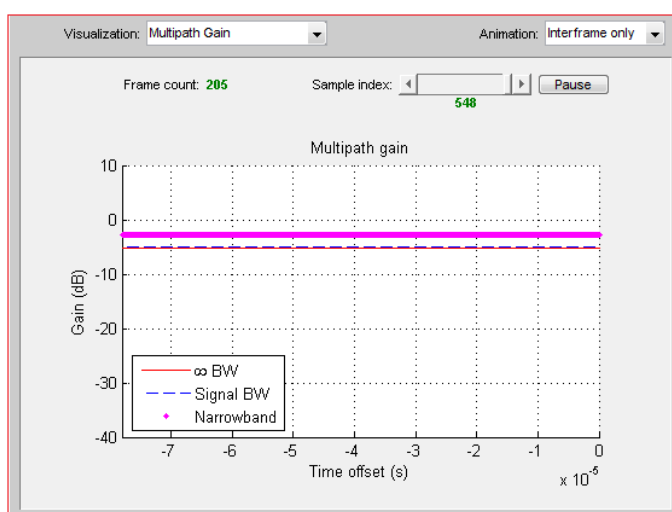


Figura A.15 - Ganho nos Multipercursos (*Multipath Gain*).

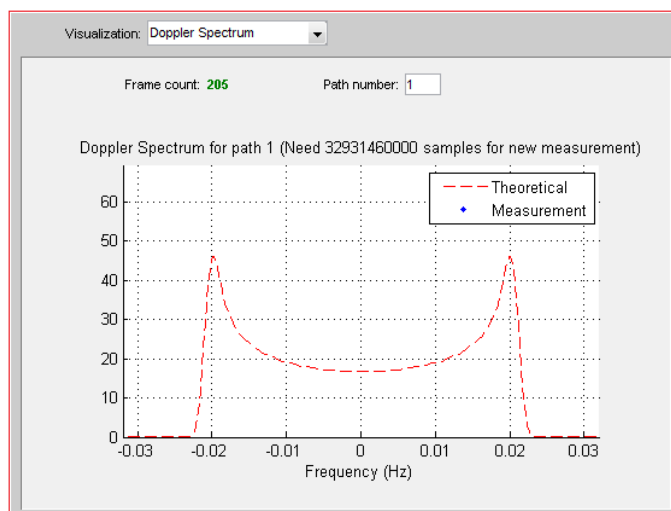


Figura A.16 - Espectro *Doppler* (*Doppler Spectrum*).

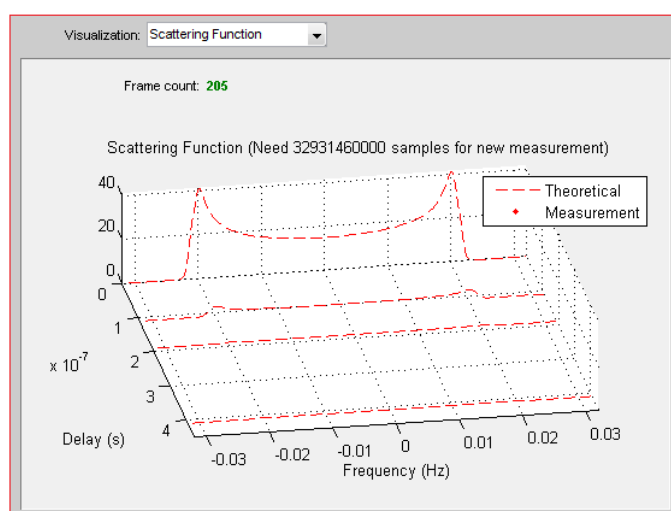


Figura A.17 - Função Espalhamento (*Scattering Function*).

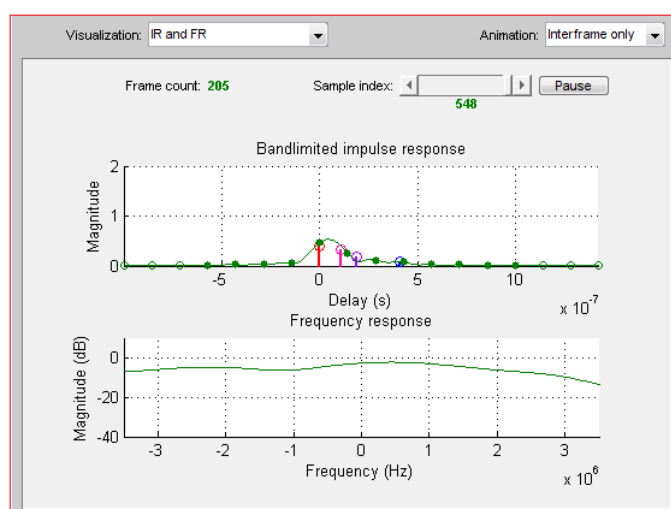


Figura A.18 - *IR and FR*.

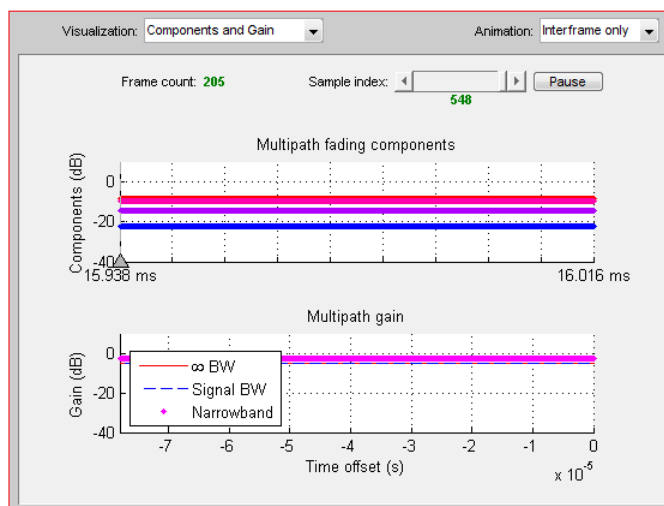


Figura A.19 - Componentes e Ganhos (*Components and Gain*).

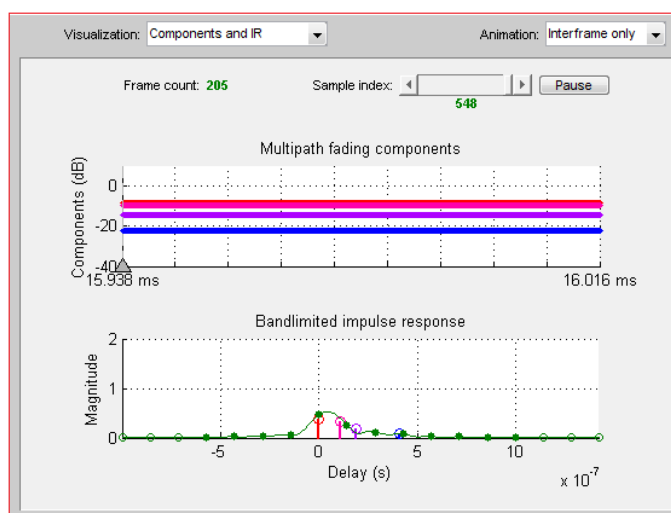


Figura A.20 - Componentes e IR (*Components and IR*).

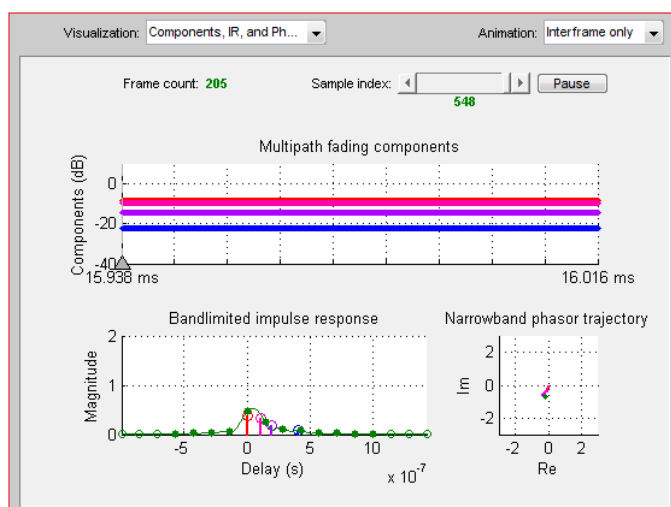


Figura A.21 - Componentes, IR e Fazor (*Components, IR and Phasor*).

Amostras com 1060 bits.

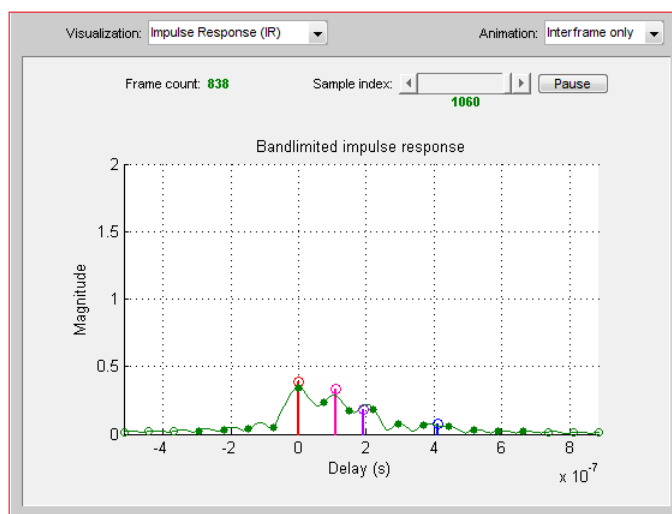


Figura A.22 - Resposta ao Impulso (IR - *Impulse Response*).

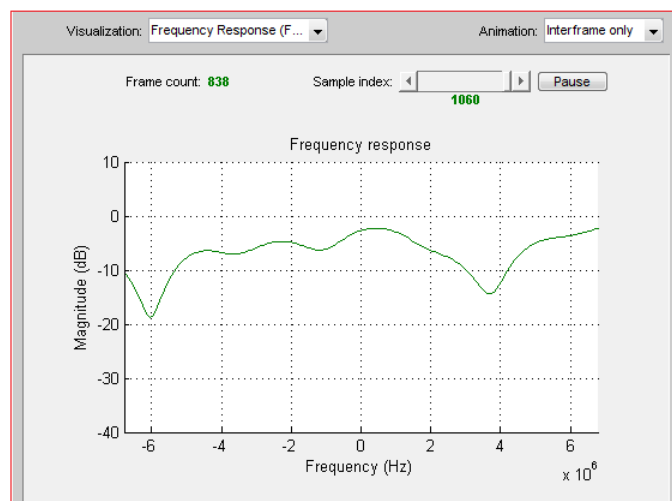


Figura A.23 - Resposta a frequência (FR - *Frequency Response*).

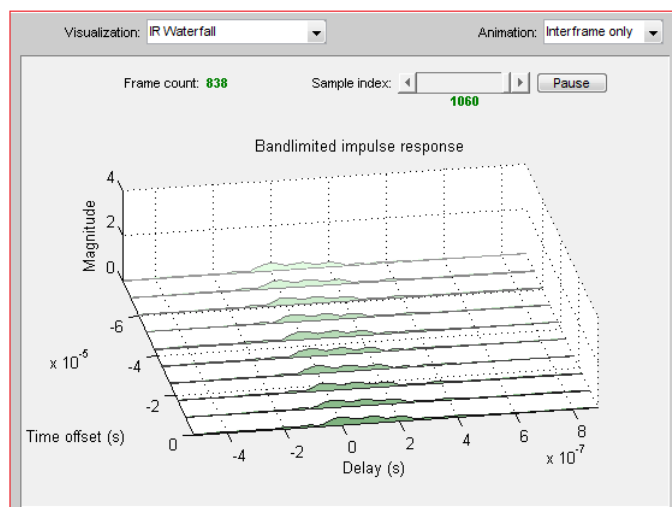


Figura A.24 - IR Waterfall.

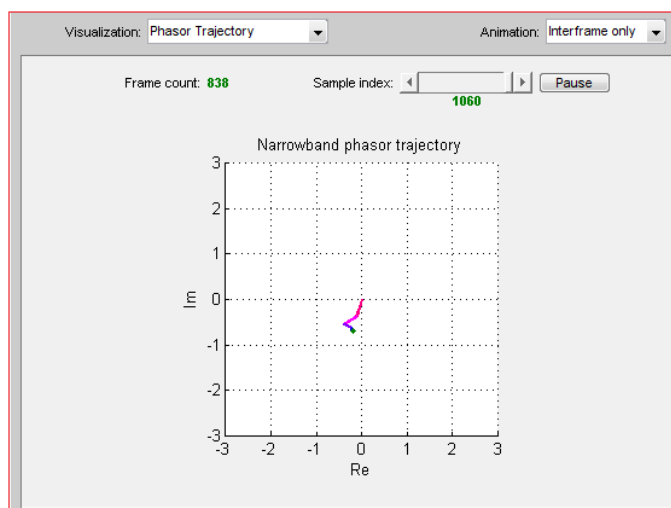


Figura A.25 - Trajetória do Fasor (*Phasor Trajectory*).

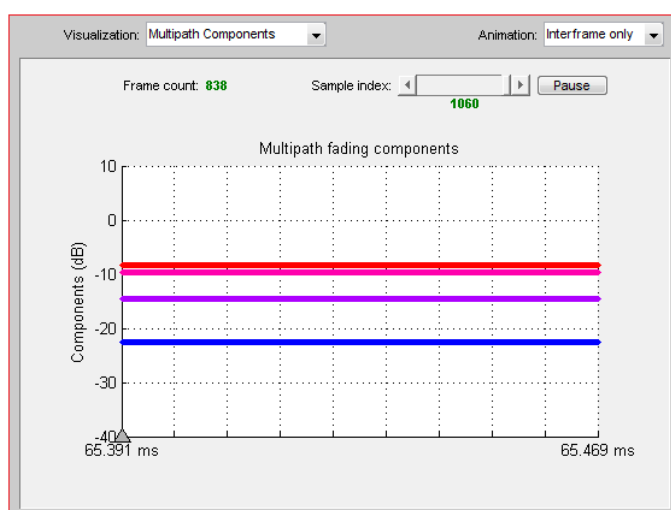


Figura A.26 - Componentes Multipercursos (*Multipath Components*).

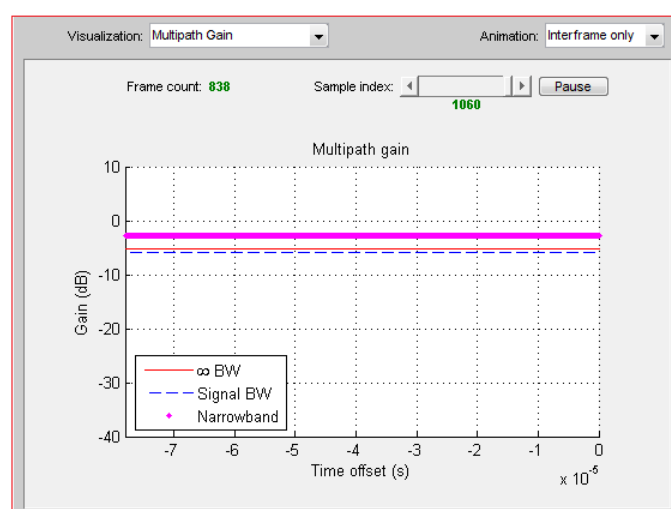


Figura A.27 - Ganho nos Multipercursos (*Multipath Gain*).

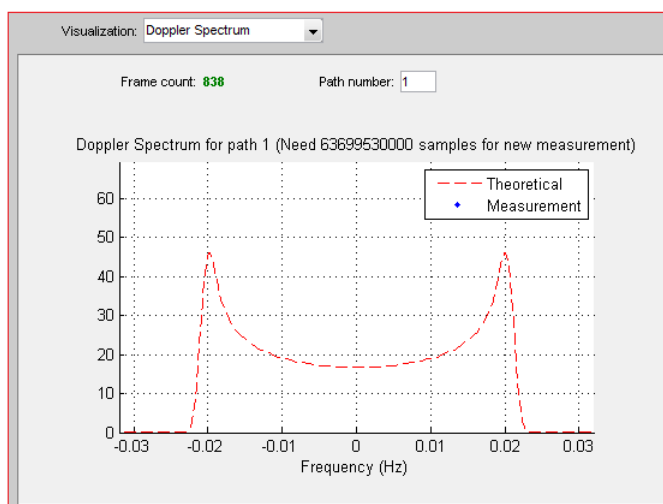


Figura A.28 - Espectro *Doppler* (*Doppler Spectrum*).

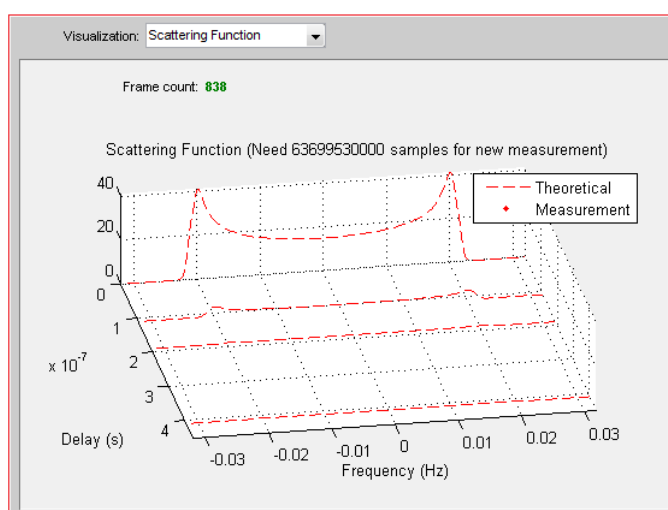


Figura A.29 - Função Espalhamento (*Scattering Function*).

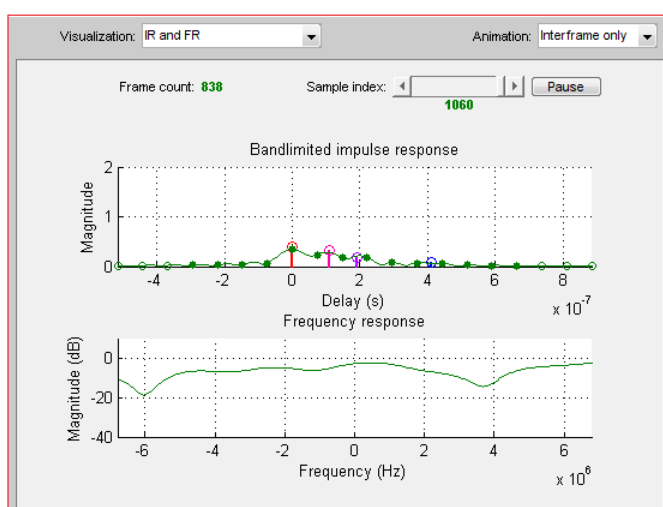


Figura A.30 - IR e FR (*IR and FR*).

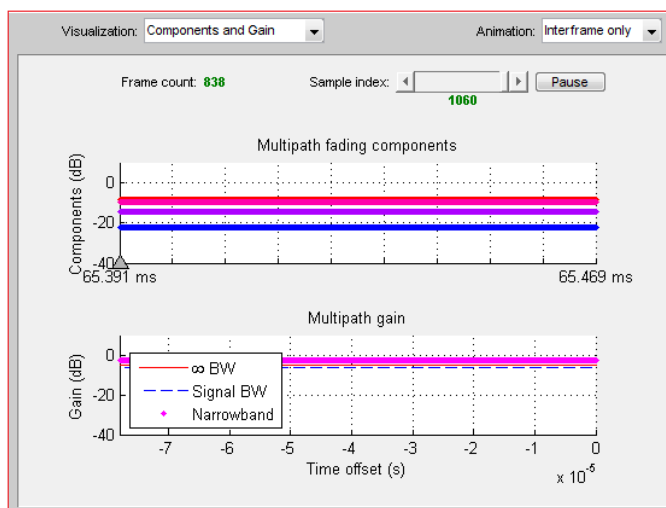


Figura A.31 - Componentes e Ganhos (*Components and Gain*).

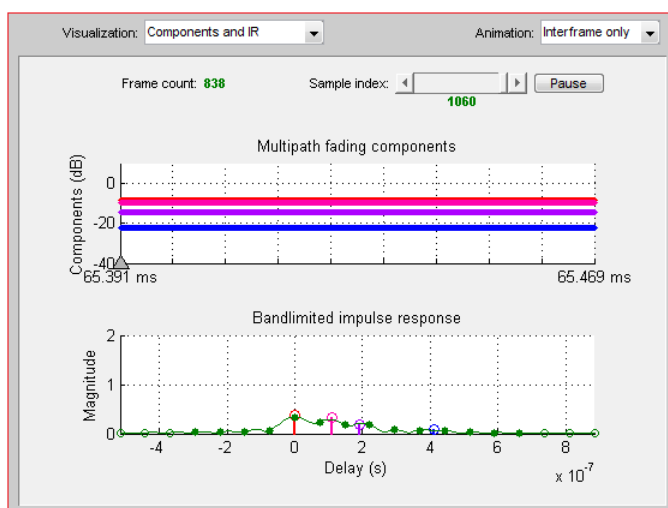


Figura A.32 - Componentes e IR (*Components and IR*).

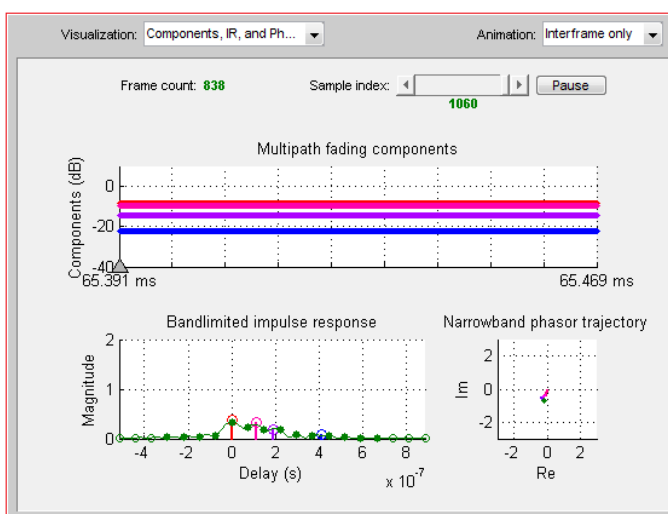


Figura A.33 - Componentes, IR e Fazor (*Components, IR and Phasor*).

6.1.7. AWGN

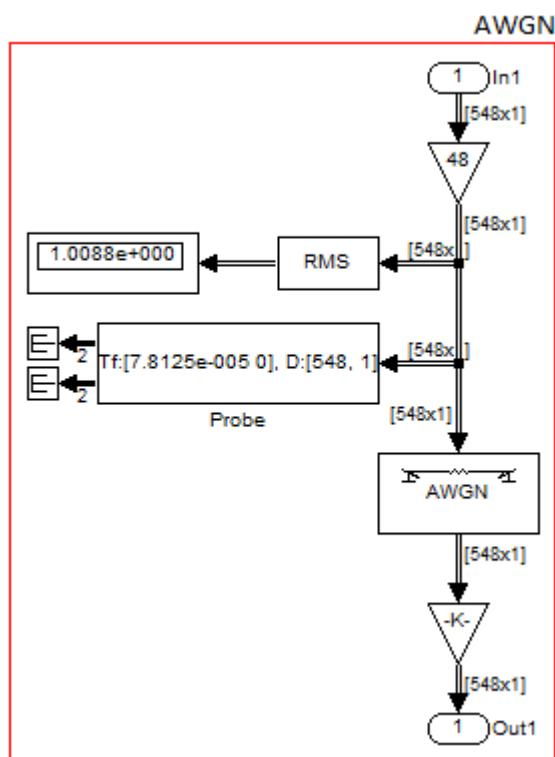


Figura A.34 - AWGN.

A.1.7.1. 1 - In1

Inport: Fornece uma porta de entrada para o subsistema AWGN.

Sinal de entrada: Sinal modulado novamente reordenado, quadros com 548 amostras, sendo 302 amostras dados e pilotos concatenados, 210 amostras zeros de enchimento e 36 amostras prefixo cíclico. Domínio do tempo. O sinal passou por um canal com Desvanecimento de Multipercursos Rayleigh.

Sinal de saída: Sinal modulado novamente reordenado, quadros com 548 amostras, sendo 302 amostras dados e pilotos concatenados, 210 amostras zeros de enchimento e 36 amostras prefixo cíclico. Domínio do tempo. O sinal passou por um canal com Desvanecimento de Multipercursos Rayleigh.

A.1.7.2. 48

Gain: Element – wise gain ($y = K.*u$) ou matrix gain ($y = K*u$ or $y = u*K$).

Parâmetro:

- Ganho (*Gain*): 48

Sinal de entrada: Sinal modulado novamente reordenado, quadros com 548 amostras, sendo 302 amostras dados e pilotos concatenados, 210 amostras zeros de enchimento e 36 amostras prefixo cíclico. Domínio do tempo. O sinal passou por um canal com Desvanecimento de Multipercursos Rayleigh.

Sinal de saída: Sinal modulado novamente reordenado, quadros com 548 amostras, sendo 302 amostras dados e pilotos concatenados, 210 amostras zeros de enchimento e 36 amostras prefixo cíclico. Domínio do tempo. O sinal passou por um canal com Desvanecimento de Multipercursos Rayleigh.

A.1.7.3. RMS

RMS: Comuta a raiz quadrada média (*Root-mean-square*) dos elementos do vetor de entrada.

Parâmetros:

- *Find the RMS value over: Each column*
- Valor Inicial (*Initial seed*): 73

Sinal de entrada: Sinal modulado novamente reordenado, quadros com 548 amostras, sendo 302 amostras dados e pilotos concatenados, 210 amostras zeros de enchimento e 36 amostras prefixo cíclico. Domínio do tempo. O sinal passou por um canal com Desvanecimento de Multipercursos Rayleigh.

A.1.7.4. Display

Display: Mostrador numérico do valor de entrada (RMS). (Valor RMS desejado = 1). Este valor é fixo, mesmo com o aumento da quantidade de pilotos inseridos.

A.1.7.5. Probe

Sonda (*Probe*): Examina uma linha e fornece a sua largura, tempo da amostra, e dimensões.

Sinal de entrada: Sinal modulado novamente reordenado, quadros com 548 amostras, sendo 302 amostras dados e pilotos concatenados, 210 amostras zeros de enchimento e 36 amostras prefixo cíclico. Domínio do tempo. O sinal passou por um canal com Desvanecimento de Multipercursos Rayleigh.

A.1.7.6. Terminate

Terminator: Utilizado para mostrar os sinais de saída (Período da amostra, dimensão do sinal) .

A.1.7.7. AWGN

Canal AWGN: Adiciona Ruído Gaussiano Branco para o sinal de entrada. Os sinais de entrada e de saída podem ser reais ou complexos. O bloco suporta entrada de multicanais.

O bloco Canal AWGN adiciona ruído Gaussiano branco para um sinal de entrada real ou complexo. Quando o sinal de entrada é real, este bloco adiciona ruído Gaussiano real e produz um sinal de saída real. Quando o sinal de entrada é complexo, este bloco adiciona ruído Gaussiano complexo e produz um sinal de saída complexo. Este bloco herda o período de amostragem do sinal de entrada.

Parâmetros:

- *Initial seed*: 12345
- *Mode: Signal to noise ratio* (E_b/N_0)
- $E_b/N_0 = 5$ à 15 dB
- *Number of bits per symbol*: $(1/2)*4 = (\text{Turbo coder}) * \text{Modulador}$
- *Input signal power (watts)*: 1
- *Symbol period(s)*: $(1/7.68e+06)*(600/548)$

E_b/N_0 : a razão entre a energia do bit e a densidade espectral de potência de ruído do símbolo, em decibéis.

Number of bits per symbol: $(1/2)*4 = (\text{Turbo coder}) * \text{Modulador}$

Symbol period(s): $(1/7.68e+06)*(600/548) = (\text{Período da amostra}) * (\text{amostras por quadro/amostras real por quadro})$

Sinal de entrada: Sinal modulado novamente reordenado, quadros com 548 amostras, sendo 302 amostras dados e pilotos concatenados, 210 amostras zeros de enchimento e 36 amostras prefixo cíclico. Domínio do tempo. O sinal passou por um canal com Desvanecimento de Multipercursos Rayleigh.

Sinal de saída: Sinal modulado novamente reordenado, quadros com 548 amostras, sendo 302 amostras dados e pilotos concatenados, 210 amostras zeros de enchimento e 36 amostras prefixo cíclico. Domínio do tempo. O sinal passou por um canal com Desvanecimento de Multipercursos Rayleigh e foi adicionado Ruído Gaussiano Branco.

A.1.7.8. K

*Gain: Element – wise gain ($y = K.*u$) or matrix gain ($y = K*u$ or $y = u*K$).*

Parâmetro:

- Ganho (*Gain*): 1/48

Sinal de entrada: Sinal modulado novamente reordenado, quadros com 548 amostras, sendo 302 amostras dados e pilotos concatenados, 210 amostras zeros de enchimento e 36 amostras prefixo cíclico. Domínio do tempo. O sinal passou por um canal com Desvanecimento de Multipercursos Rayleigh e foi adicionado Ruído Gaussiano Branco.

Sinal de saída: Sinal modulado novamente reordenado, quadros com 548 amostras, sendo 302 amostras dados e pilotos concatenados, 210 amostras zeros de enchimento e 36 amostras prefixo cíclico. Domínio do tempo. O sinal passou por um canal com Desvanecimento de Multipercursos Rayleigh e foi adicionado Ruído Gaussiano Branco.

A.1.7.9. 1 Out

Outport: Fornece uma porta de saída para o subsistema AWGN.

Sinal de entrada: Sinal modulado novamente reordenado, quadros com 548 amostras, sendo 302 amostras dados e pilotos concatenados, 210 amostras zeros de enchimento e 36 amostras prefixo cíclico. Domínio do tempo. O sinal passou por um canal com Desvanecimento de Multipercursos Rayleigh e foi adicionado Ruído Gaussiano Branco.

Sinal de saída: Sinal modulado novamente reordenado, quadros com 548 amostras, sendo 302 amostras dados e pilotos concatenados, 210 amostras zeros de enchimento e 36 amostras prefixo cíclico. Domínio do tempo. O sinal passou por um canal com Desvanecimento de Multipercursos Rayleigh e foi adicionado Ruído Gaussiano Branco.

A.1.8. B-FFT - Spectrum Scope

Spectrum Scope: Calcula e exibe o periodograma de cada sinal de entrada.

A.1.9. OFDM Receiver

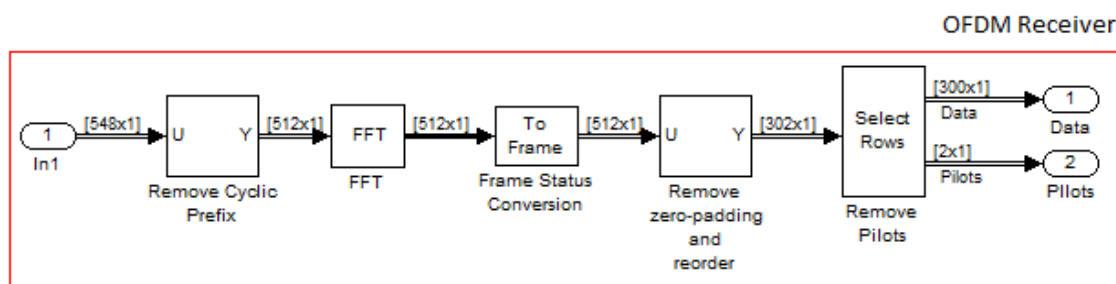


Figura A.35 - Receptor OFDM (*OFDM Receiver*).

A.1.9.1. 1 - In1

Inport: Fornece uma porta de entrada para o subsistema receptor OFDM (*OFDM receiver*).

Sinal de entrada: Sinal modulado novamente reordenado, quadros com 548 amostras, sendo 302 amostras dados e pilotos concatenados, 210 amostras zeros de enchimento e 36 amostras prefixo cíclico. Domínio do tempo. O sinal passou por um canal com Desvanecimento de Multipercursos Rayleigh e foi adicionado Ruído Gaussiano Branco.

Sinal de saída: Sinal modulado novamente reordenado, quadros com 548 amostras, sendo 302 amostras dados e pilotos concatenados, 210 amostras zeros de enchimento e 36 amostras prefixo cíclico. Domínio do tempo. O sinal passou por um canal com Desvanecimento de Multipercursos Rayleigh e foi adicionado Ruído Gaussiano Branco.

A.1.9.2. Remove Cyclic Prefix

Selector: Seleciona ou reordena elementos específicos de um sinal de entrada multidimensional. O índice de cada elemento é identificado por uma porta de entrada ou caixa de diálogo. Inserção de prefixo cíclico.

Remoção das 36 amostras de prefixo cíclico.

Parâmetros:

- Dimensão do número de entrada (*Number of input dimensions*): 1
- Modo de indexação (*Index mode*): *One-based*
- Opção de Índice (*Index option*): *Index vector*
- Índice (*Index*): [37:548]
- *Input port size*: 548

Índice (*Index*): [Dados+Zeros+Pilotos] = [37:548]

Tabela A.10 - Prefixo Cíclico.

OFDM Receiver	548
Remove Cyclic Prefix	36
[37:548]	512

Sinal de entrada: Sinal modulado novamente reordenado, quadros com 548 amostras, sendo 302 amostras dados e pilotos concatenados, 210 amostras zeros de enchimento e 36 amostras prefixo cíclico. Domínio do tempo. O sinal passou por um canal com Desvanecimento de Multipercursos Rayleigh e foi adicionado Ruído Gaussiano Branco.

Sinal de saída: Sinal modulado novamente reordenado, quadros com 512 amostras, sendo 302 amostras dados e pilotos concatenados e 210 amostras zeros de enchimento. Domínio do tempo. O sinal passou por um canal com Desvanecimento de Multipercursos Rayleigh e foi adicionado Ruído Gaussiano Branco. Remoção das 36 amostras de prefixo cíclico.

A.1.9.3. FFT

FFT: Saída complexa *fast Fourier transform (FFT)* de uma entrada real ou complexa.

Calcula a transformada rápida de Fourier (FFT) da entrada.

Transforma o sinal do domínio do tempo para o domínio da frequência. A FFT transforma um sinal no domínio do tempo no espectro de frequência.

Sinal de entrada: Sinal modulado novamente reordenado, quadros com 512 amostras, sendo 302 amostras dados e pilotos concatenados e 210 amostras zeros de enchimento. Domínio do tempo. O sinal passou por um canal com Desvanecimento de Multipercursos Rayleigh e foi adicionado Ruído Gaussiano Branco. Remoção das 36 amostras de prefixo cíclico.

Sinal de saída: Sinal modulado novamente reordenado, quadros com 512 amostras, sendo 302 amostras dados e pilotos concatenados e 210 amostras zeros de enchimento. O sinal passou por um canal com Desvanecimento de Multipercursos Rayleigh e foi adicionado Ruído Gaussiano Branco. Remoção das 36 amostras de prefixo cíclico. Domínio da frequência.

A.1.9.4. To Frame - Frame Status Conversion

Frame Status Conversion: Especifica o status de quadro do sinal de saída.

A.1.9.5. Remove zero-padding and reorder

Selector: Seleciona ou reordena elementos específicos de um sinal de entrada multidimensional. O índice de cada elemento é identificado por uma porta de entrada ou caixa de diálogo.

Remoção dos zeros de enchimento.

Para 2 pilotos e com os zeros de enchimentos, tinha-se como índice: [Dados+Pilotos Zeros Dados+Pilotos] = [362:512 152:361 1:151]. Com a remoção dos zeros de enchimentos ficou [Dados+Pilotos Dados+Pilotos] = [1:151 362:512].

Para 3 pilotos e com os zeros de enchimentos, tinha-se como índice: [Dados+Pilotos Zeros Dados+Pilotos] = [361:512 153:360 1:152]. Com a remoção dos zeros de enchimentos ficou [Dados+Pilotos Dados+Pilotos] =

[1:152 361:512]

Para 4 pilotos e com os zeros de enchimentos, tinha-se como índice: [Dados+Pilotos Zeros Dados+Pilotos] = [361:512 153:360 1:152]. Com a remoção dos zeros de enchimentos ficou [Dados+Pilotos Dados+Pilotos] = [1:152 361:512]

A Tabela A.10, na coluna “OFDM Receiver – Remove Zero pad and reorder - Selector Index vector”, tem-se os índices em função da quantidade de pilotos, após a remoção dos zeros de enchimento.

Tabela A.11 - Remoção dos zeros de enchimento.

						OFDM Receiver
						Remove Zero pad and reorder
						Selector Index vector
Divisores	Select Rows	Pilotos	Dados + Pilotos	(Dados + Pilotos)/Select Rows	Arred((Dados + Pilotos)/Select Rows)	Dados + Pilotos
300	1	2	302	302.0	302	[1:151 362:512]
150	2	3	303	151.5	152	[1:152 361:512]
100	3	4	304	101.3	101	[1:152 361:512]
75	4	5	305	76.3	76	[1:153 360:512]
60	5	6	306	61.2	61	[1:153 360:512]
50	6	7	307	51.2	51	[1:154 359:512]
30	10	11	311	31.1	31	[1:156 357:512]
25	12	13	313	26.1	26	[1:157 356:512]
20	15	16	316	21.1	21	[1:158 355:512]
15	20	21	321	16.1	16	[1:161 352:512]
12	25	26	326	13.0	13	[1:163 350:512]
10	30	31	331	11.0	11	[1:166 347:512]
6	50	51	351	7.0	7	[1:176 337:512]
5	60	61	361	6.0	6	[1:181 332:512]
4	75	76	376	5.0	5	[1:188 325:512]
3	100	101	401	4.0	4	[1:201 312:512]
2	150	151	451	3.0	3	[1:226 287:512]

Sinal de entrada: Sinal modulado novamente reordenado, quadros com 512 amostras, sendo 302 amostras dados e pilotos concatenados e 210 amostras zeros de enchimento. O sinal passou por um canal com Desvanecimento de Multipercursos Rayleigh e foi adicionado Ruído Gaussiano Branco. Remoção das 36 amostras de prefixo cíclico. Domínio da frequência.

Sinal de saída: Sinal modulado novamente reordenado, quadros com 302 amostras de dados e pilotos concatenados. O sinal passou por um canal com Desvanecimento de Multipercursos Rayleigh e foi adicionado Ruído Gaussiano Branco. Remoção das 36 amostras de prefixo cíclico. Domínio da frequência. Remoção das 210 amostras de zeros de enchimento.

A.1.9.6. Remove Pilots – Select Rows

Multiport Selector: Saída especificada de linhas ou colunas para uma ou mais portas de saída. O número de portas de saída é determinado pelo número de vetores de índice, cada um é especificado como um vetor separado de entrada em uma matriz de células.

Remoção dos pilotos.

Considerando a remoção de dois pilotos tem-se $\{[Dados],[Pilotos]\} = \{[2:301],[1\ 302]\}$.

Considerando a remoção de três pilotos tem-se $\{[Dados],[Pilotos]\} = \{[2:151\ 153:302],[1\ 152\ 303]\}$.

Considerando a remoção de quatro pilotos tem-se $\{[Dados],[Pilotos]\} = \{[2:101\ 103:202\ 204:303],[1\ 102\ 203\ 304]\}$.

O raciocínio vale para as demais quantidades de pilotos (5, 6, 7, 11, 13, 16, 21, 26, 31, 51, 61, 76, 101, 151).

A Tabela A.12 apresenta a remoção dos pilotos.

Tabela A.12 - Remoção dos pilotos.

Pilotos	OFDM
	Receiver
	Multipoint
	Selector
Pilotos	Indices to
	output
	{[Select
	Rows],[Pilots]}
2	{[2:301],[1 302]}
3	{[2:151 153:302],[1 152 303]}
4	{[2:101 103:202 204:303],[1 102 203 304]}
5	{[2:76 78:152 154:228 230:304],[1 77 153 229 305]}
6	{[2:61 63:122 124:183 185:244 246:305],[1 62 123 184 245 306]}
7	{[2:51 53:102 104:153 155:204 206:255 257:306],[1 52 103 154 205 256 307]}
11	{[2:31 33:62 64:93 95:124 126:155 157:186 188:217 219:248 250:279 281:310], [1 32 63 94 125 156 187 218 249 280 311]}
13	{[2:26 28:52 54:78 80:104 106:130 132:156 158:182 184:208 210:234 236:260 262:286 288:312],[1 27 53 79 105 131 157 183 209 235 261 287 313]}
16	{[2:21 23:42 44:63 65:84 86:105 107:126 128:147 149:168 170:189 191:210 212:231 233:252 254:273 275:294 296:315],[1 22 43 64 85 106 127 148 169 190 211 232 253 274 295 316]}
21	{[2:16 18:32 34:48 50:64 66:80 82:96 98:112 114:128 130:144 146:160 162:176 178:192 194:208 210:224 226:240 242:256 258:272 274:288 290:304 306:320], [1 17 33 49 65 81 97 113 129 145 161 177 193 209 225 241 257 273 289 305 321]}
26	{[2:13 15:26 28:39 41:52 54:65 67:78 80:91 93:104 106:117 119:130 132:143 145:156 158:169 171:182 184:195 197:208 210:221 223:234 236:247 249:260 262:273 275:286 288:299 301:312 314:325],[1 14 27 40 53 66 79 92 105 118 131 144 157 170 183 196 209 222 235 248 261 274 287 300 313 326]}
31	{[2:11 13:22 24:33 35:44 46:55 57:66 68:77 79:88 90:99 101:110 112:121 123:132 134:143 145:154 156:165 167:176 178:187 189:198 200:209 211:220 222:231

	233:242 244:253 255:264 266:275 277:286 288:297 299:308 310:319 321:330], [1 12 23 34 45 56 67 78 89 100 111 122 133 144 155 166 177 188 199 210 221 232 243 254 265 276 287 298 309 320 331]]
51	{{2:7 9:14 16:21 23:28 30:35 37:42 44:49 51:56 58:63 65:70 72:77 79:84 86:91 93:98 100:105 107:112 114:119 121:126 128:133 135:140 142:147 149:154 156:161 163:168 170:175 177:182 184:189 191:196 198:203 205:210 212:217 219:224 226:231 233:238 240:245 247:252 254:259 261:266 268:273 275:280 282:287 289:294 296:301 303:308 310:315 317:322 324:329 331:336 338:343 345:350],[1 8 15 22 29 36 43 50 57 64 71 78 85 92 99 106 113 120 127 134 141 148 155 162 169 176 183 190 197 204 211 218 225 232 239 246 253 260 267 274 281 288 295 302 309 316 323 330 337 344 351]]}
61	{{2:6 8:12 14:18 20:24 26:30 32:36 38:42 44:48 50:54 56:60 62:66 68:72 74:78 80:84 86:90 92:96 98:102 104:108 110:114 116:120 122:126 128:132 134:138 140:144 146:150 152:156 158:162 164:168 170:174 176:180 182:186 188:192 194:198 200:204 206:210 212:216 218:222 224:228 230:234 236:240 242:246 248:252 254:258 260:264 266:270 272:276 278:282 284:288 290:294 296:300 302:306 308:312 314:318 320:324 326:330 332:336 338:342 344:348 350:354 356:360],[1 7 13 19 25 31 37 43 49 55 61 67 73 79 85 91 97 103 109 115 121 127 133 139 145 151 157 163 169 175 181 187 193 199 205 211 217 223 229 235 241 247 253 259 265 271 277 283 289 295 301 307 313 319 325 331 337 343 349 355 361]]}
76	{{2:5 7:10 12:15 17:20 22:25 27:30 32:35 37:40 42:45 47:50 52:55 57:60 62:65 67:70 72:75 77:80 82:85 87:90 92:95 97:100 102:105 107:110 112:115 117:120 122:125 127:130 132:135 137:140 142:145 147:150 152:155 157:160 162:165 167:170 172:175 177:180 182:185 187:190 192:195 197:200 202:205 207:210 212:215 217:220 222:225 227:230 232:235 237:240 242:245 247:250 252:255 257:260 262:265 267:270 272:275 277:280 282:285 287:290 292:295 297:300 302:305 307:310 312:315 317:320 322:325 327:330 332:335 337:340 342:345 347:350 352:355 357:360 362:365 367:370 372:375],[1 6 11 16 21 26 31 36 41 46 51 56 61 66 71 76 81 86 91 96 101 106 111 116 121 126 131 136 141 146 151 156 161 166 171 176 181 186 191 196 201 206 211 216 221 226 231 236 241 246 251 256 261 266 271 276 281 286 291 296 301 306 311 316 321 326 331 336 341 346 351 356 361 366 371 376]]}
101	{{2:4 6:8 10:12 14:16 18:20 22:24 26:28 30:32 34:36 38:40 42:44 46:48 50:52 54:56 58:60 62:64 66:68 70:72 74:76 78:80 82:84 86:88 90:92 94:96 98:100 102:104 106:108 110:112 114:116 118:120 122:124 126:128 130:132 134:136 138:140 142:144 146:148 150:152 154:156 158:160 162:164 166:168 170:172 174:176 178:180 182:184 186:188 190:192 194:196 198:200 202:204 206:208 210:212 214:216 218:220 222:224 226:228 230:232 234:236 238:240 242:244 246:248 250:252 254:256 258:260 262:264 266:268 270:272 274:276 278:280

	282:284 286:288 290:292 294:296 298:300 302:304 306:308 310:312 314:316 318:320 322:324 326:328 330:332 334:336 338:340 342:344 346:348 350:352 354:356 358:360 362:364 366:368 370:372 374:376 378:380 382:384 386:388 390:392 394:396 398:400],[1 5 9 13 17 21 25 29 33 37 41 45 49 53 57 61 65 69 73 77 81 85 89 93 97 101 105 109 113 117 121 125 129 133 137 141 145 149 153 157 161 165 169 173 177 181 185 189 193 197 201 205 209 213 217 221 225 229 233 237 241 245 249 253 257 261 265 269 273 277 281 285 289 293 297 301 305 309 313 317 321 325 329 333 337 341 345 349 353 357 361 365 369 373 377 381 385 389 393 397 401]]
151	{{2:3 5:6 8:9 11:12 14:15 17:18 20:21 23:24 26:27 29:30 32:33 35:36 38:39 41:42 44:45 47:48 50:51 53:54 56:57 59:60 62:63 65:66 68:69 71:72 74:75 77:78 80:81 83:84 86:87 89:90 92:93 95:96 98:99 101:102 104:105 107:108 110:111 113:114 116:117 119:120 122:123 125:126 128:129 131:132 134:135 137:138 140:141 143:144 146:147 149:150 152:153 155:156 158:159 161:162 164:165 167:168 170:171 173:174 176:177 179:180 182:183 185:186 188:189 191:192 194:195 197:198 200:201 203:204 206:207 209:210 212:213 215:216 218:219 221:222 224:225 227:228 230:231 233:234 236:237 239:240 242:243 245:246 248:249 251:252 254:255 257:258 260:261 263:264 266:267 269:270 272:273 275:276 278:279 281:282 284:285 287:288 290:291 293:294 296:297 299:300 302:303 305:306 308:309 311:312 314:315 317:318 320:321 323:324 326:327 329:330 332:333 335:336 338:339 341:342 344:345 347:348 350:351 353:354 356:357 359:360 362:363 365:366 368:369 371:372 374:375 377:378 380:381 383:384 386:387 389:390 392:393 395:396 398:399 401:402 404:405 407:408 410:411 413:414 416:417 419:420 422:423 425:426 428:429 431:432 434:435 437:438 440:441 443:444 446:447 449:450],[1 4 7 10 13 16 19 22 25 28 31 34 37 40 43 46 49 52 55 58 61 64 67 70 73 76 79 82 85 88 91 94 97 100 103 106 109 112 115 118 121 124 127 130 133 136 139 142 145 148 151 154 157 160 163 166 169 172 175 178 181 184 187 190 193 196 199 202 205 208 211 214 217 220 223 226 229 232 235 238 241 244 247 250 253 256 259 262 265 268 271 274 277 280 283 286 289 292 295 298 301 304 307 310 313 316 319 322 325 328 331 334 337 340 343 346 349 352 355 358 361 364 367 370 373 376 379 382 385 388 391 394 397 400 403 406 409 412 415 418 421 424 427 430 433 436 439 442 445 448 451]]

Sinal de entrada: Sinal modulado novamente reordenado, quadros com 302 amostras de dados e pilotos concatenados. O sinal passou por um canal com Desvanecimento de Multipercursos Rayleigh e foi adicionado Ruído Gaussiano Branco. Remoção das 36 amostras de prefixo cíclico. Domínio da frequência. Remoção das 210 amostras de zeros de enchimento.

Sinal de saída: Sinal modulado novamente reordenado, quadros com 300 amostras de dados e 2 amostras de pilotos não concatenados. O sinal passou por um canal com Desvanecimento de Multipercursos Rayleigh e foi adicionado Ruído

Gaussiano Branco. Remoção das 36 amostras de prefixo cíclico. Domínio da frequência. Remoção das 210 amostras de zeros de enchimento. Remoção dos Pilotos.

A.1.9.7. 1 Data

Outport: Fornece uma porta de saída para um subsistema ou modelo.

Sinal de entrada: Sinal modulado novamente reordenado, quadros com 300 amostras de dados. O sinal passou por um canal com Desvanecimento de Multipercursos Rayleigh e foi adicionado Ruído Gaussiano Branco. Remoção das 36 amostras de prefixo cíclico. Domínio da frequência. Remoção das 210 amostras de zeros de enchimento. Remoção definitiva dos Pilotos. Só os dados estão sendo considerados.

A.1.9.8. 2 Pilots

Outport: Fornece uma porta de saída para um subsistema ou modelo.

Sinal de entrada: Sinal modulado novamente reordenado, quadros com 2 amostras de pilotos. O sinal passou por um canal com Desvanecimento de Multipercursos Rayleigh e foi adicionado Ruído Gaussiano Branco. Remoção das 36 amostras de prefixo cíclico. Domínio da frequência. Remoção das 210 amostras de zeros de enchimento. Remoção dos dados. Só os Pilotos estão sendo considerados.

A.1.10. LMS Channel Estimator

A Figura A.36 representa um *LMS Channel Estimator* considerando 2 pilotos, os *LMS Channel Estimator* para as demais quantidades de pilotos (3, 4, 5, 6, 7, 11, 13, 16, 21, 26, 31, 51, 61, 76, 101, 151) encontram-se no Apêndice B.

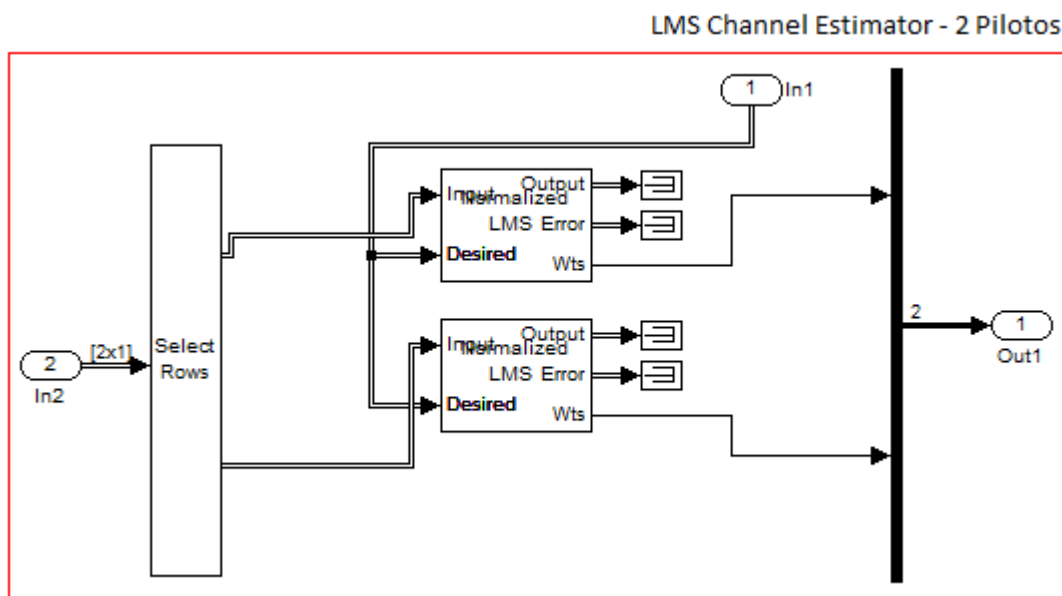


Figura A.36 - *LMS Channel Estimator*.

A.1.10.1. 2 - In2

Inport: Fornece uma porta de entrada para um subsistema ou modelo.

Sinal de entrada: Sinal modulado novamente reordenado, quadros com 2 amostras de pilotos. O sinal passou por um canal com Desvanecimento de Multipercursos Rayleigh e foi adicionado Ruído Gaussiano Branco. Remoção das 36 amostras de prefixo cíclico. Domínio da frequência. Remoção das 210 amostras de zeros de enchimento. Remoção dos dados. Só os Pilotos estão sendo considerados.

Sinal de saída: Sinal modulado novamente reordenado, quadros com 2 amostras de pilotos. O sinal passou por um canal com Desvanecimento de Multipercursos Rayleigh e foi adicionado Ruído Gaussiano Branco. Remoção das 36 amostras de prefixo cíclico. Domínio da frequência. Remoção das 210 amostras de zeros de enchimento. Remoção dos dados. Só os Pilotos estão sendo considerados.

A.1.10.2. Select Rows

Seletor de Múltiplas Portas (*Multiport Selector*): saída especificada de linhas ou colunas para uma ou mais portas de saída. O número de portas de saída é determinado pelo número de vetores de índice, cada um especificado como um vetor separado de entrada em uma matriz de células.

Distribui subconjuntos arbitrários de linhas ou colunas de entrada para várias portas de saída.

Parâmetros:

- Seleção (*Select*): Linhas (*Rows*)
- Índices de saída (*Indices to output*): {[1],[2]}
- Índice inválido (*Invalid index*): Gerador de erro (*Generator Error*).

Seleção (*Select*): Selecciona a dimensão da entrada, linhas ou colunas. Índices de saída (*Indices to output*): Uma matriz de células especificando subconjuntos de linha ou coluna para propagar a cada uma das portas de saída. O número de células na matriz determina o número de portas de saída do bloco.

Índice inválido: Resposta a um valor de índice inválido (linha inexistente).

Com 2 pilotos tem-se índice de saída (*Indices to output*) igual à {[1],[2]}, que representa uma matriz de duas células, que gera duas saídas independentes (com duas portas distintas). O parâmetro {[1],[2]} também representa as linhas 1, 2 da matriz de entrada e comprimento de saída (output size) 2-by-N.

Com 3 pilotos tem-se índices de saída (*Indices to output*) igual à {[1],[2],[3]}, que representam uma matriz de três células, que gera três saídas independentes (cada uma em uma porta distinta). O parâmetro {[1],[2],[3]} também representa as linhas 1, 2, 3 da matriz de entrada e comprimento de saída (output size) 3-by-N.

Com 4 pilotos tem-se índices de saída (*Indices to output*) igual à {[1],[2],[3],[4]}, que representam uma matriz de quatro células, que gera quatro saídas independentes (cada uma em uma porta distinta). O parâmetro {[1],[2],[3],[4]} também representa as linhas 1, 2, 3, 4 da matriz de entrada e comprimento de saída (output size) 4-by-N.

O raciocínio vale para as demais quantidades de pilotos (5, 6, 7, 11, 13, 16, 21, 26, 31, 51, 61, 76, 101, 151).

A Tabela A.12, na coluna “LMS Channel Estimator - Select Rows - Indices to Output” apresenta a quantidade de portas de saída do bloco em função da quantidade de pilotos.

A Tabela A.13, na coluna “LMS Channel Estimator - Select Rows - Indices to Output” apresenta os índices de saída do bloco em função da quantidade de pilotos.

Tabela A.13 - LMS Channel Estimator.

			LMS Channel Estimator	LMS Channel Estimator
			Mux	Select Rows
Divisores	Select Rows	Pilotos	Number of Inputs	Indices to Output
300	1	2	2	2
150	2	3	3	3
100	3	4	4	4
75	4	5	5	5
60	5	6	6	6
50	6	7	7	7
30	10	11	11	11
25	12	13	13	13
20	15	16	16	16
15	20	21	21	21
12	25	26	26	26
10	30	31	31	31
6	50	51	51	51
5	60	61	61	61
4	75	76	76	76
3	100	101	101	101
2	150	151	151	151

Tabela A.14 - LMS Channel Estimator.

LMS Channel Estimator	
Select Rows	
Pilotos	Indices to Output
2	{[1],[2]}
3	{[1],[2],[3]}
4	{[1],[2],[3],[4]}
5	{[1],[2],[3],[4],[5]}

6	{[1],[2],[3],[4],[5],[6]}
7	{[1],[2],[3],[4],[5],[6],[7]}
11	{[1],[2],[3],[4],[5],[6],[7],[8],[9],[10],[11]}
13	{[1],[2],[3],[4],[5],[6],[7],[8],[9],[10],[11],[12],[13]}
16	{[1],[2],[3],[4],[5],[6],[7],[8],[9],[10],[11],[12],[13],[14],[15],[16]}
21	{[1],[2],[3],[4],[5],[6],[7],[8],[9],[10],[11],[12],[13],[14],[15],[16],[17],[18],[19],[20],[21]}
26	{[1],[2],[3],[4],[5],[6],[7],[8],[9],[10],[11],[12],[13],[14],[15],[16],[17],[18],[19],[20],[21],[22],[23],[24],[25],[26]}
31	{[1],[2],[3],[4],[5],[6],[7],[8],[9],[10],[11],[12],[13],[14],[15],[16],[17],[18],[19],[20],[21],[22],[23],[24],[25],[26],[27],[28],[29],[30],[31]}
51	{[1],[2],[3],[4],[5],[6],[7],[8],[9],[10],[11],[12],[13],[14],[15],[16],[17],[18],[19],[20],[21],[22],[23],[24],[25],[26],[27],[28],[29],[30],[31],[32],[33],[34],[35],[36],[37],[38],[39],[40],[41],[42],[43],[44],[45],[46],[47],[48],[49],[50],[51]}
61	{[1],[2],[3],[4],[5],[6],[7],[8],[9],[10],[11],[12],[13],[14],[15],[16],[17],[18],[19],[20],[21],[22],[23],[24],[25],[26],[27],[28],[29],[30],[31],[32],[33],[34],[35],[36],[37],[38],[39],[40],[41],[42],[43],[44],[45],[46],[47],[48],[49],[50],[51],[52],[53],[54],[55],[56],[57],[58],[59],[60],[61]}
76	{[1],[2],[3],[4],[5],[6],[7],[8],[9],[10],[11],[12],[13],[14],[15],[16],[17],[18],[19],[20],[21],[22],[23],[24],[25],[26],[27],[28],[29],[30],[31],[32],[33],[34],[35],[36],[37],[38],[39],[40],[41],[42],[43],[44],[45],[46],[47],[48],[49],[50],[51],[52],[53],[54],[55],[56],[57],[58],[59],[60],[61],[62],[63],[64],[65],[66],[67],[68],[69],[70],[71],[72],[73],[74],[75],[76]}
101	{[1],[2],[3],[4],[5],[6],[7],[8],[9],[10],[11],[12],[13],[14],[15],[16],[17],[18],[19],[20],[21],[22],[23],[24],[25],[26],[27],[28],[29],[30],[31],[32],[33],[34],[35],[36],[37],[38],[39],[40],[41],[42],[43],[44],[45],[46],[47],[48],[49],[50],[51],[52],[53],[54],[55],[56],[57],[58],[59],[60],[61],[62],[63],[64],[65],[66],[67],[68],[69],[70],[71],[72],[73],[74],[75],[76],[77],[78],[79],[80],[81],[82],[83],[84],[85],[86],[87],[88],[89],[90],[91],[92],[93],[94],[95],[96],[97],[98],[99],[100],[101]}
151	{[1],[2],[3],[4],[5],[6],[7],[8],[9],[10],[11],[12],[13],[14],[15],[16],[17],[18],[19],[20],[21],[22],[23],[24],[25],[26],[27],[28],[29],[30],[31],[32],[33],[34],[35],[36],[37],[38],[39],[40],[41],[42],[43],[44],[45],[46],[47],[48],[49],[50],[51],[52],[53],[54],[55],[56],[57],[58],[59],[60],[61],[62],[63],[64],[65],[66],[67],[68],[69],[70],[71],[72],[73],[74],[75],[76],[77],[78],[79],[80],[81],[82],[83],[84],[85],[86],[87],[88],[89],[90],[91],[92],[93],[94],[95],[96],[97],[98],[99],[100],[101],[102],[103],[104],[105],[106],[107],[108],[109],[110],[111],[112],[113],[114],[115],[116],[117],[118],[119],[120],[121],[122],[123],[124],[125],[126],[127],[128],[129],[130],[131],[132],[133],[134],[135],[136],[137],[138],[139],[140],[141],[142],[143],[144],[145],[146],[147],[148],[149],[150],[151]}

	[57],[58],[59],[60],[61],[62],[63],[64],[65],[66],[67],[68],[69],[70],[71],[72],[73],[74], [75],[76],[77],[78],[79],[80],[81],[82],[83],[84],[85],[86],[87],[88],[89],[90],[91],[92], [93],[94],[95],[96],[97],[98],[99],[100],[101],[102],[103],[104],[105],[106],[107],[108], [109],[110],[111],[112],[113],[114],[115],[116],[117],[118],[119],[120],[121],[122], [123],[124],[125],[126],[127],[128],[129],[130],[131],[132],[133],[134],[135],[136], [137],[138],[139],[140],[141],[142],[143],[144],[145],[146],[147],[148],[149],[150], [151]}
--	---

Sinal de entrada: Sinal modulado novamente reordenado, quadros com 2 amostras de pilotos. O sinal passou por um canal com Desvanecimento de Multipercursos Rayleigh e foi adicionado Ruído Gaussiano Branco. Remoção das 36 amostras de prefixo cíclico. Domínio da frequência. Remoção das 210 amostras de zeros de enchimento. Remoção dos dados. Só os Pilotos estão sendo considerados.

Sinal de saída: Sinal modulado novamente reordenado, quadros com 2 amostras de pilotos. O sinal passou por um canal com Desvanecimento de Multipercursos Rayleigh e foi adicionado Ruído Gaussiano Branco. Remoção das 36 amostras de prefixo cíclico. Domínio da frequência. Remoção das 210 amostras de zeros de enchimento. Remoção dos dados. Só os Pilotos estão sendo considerados e separadamente.

A.1.10.3. 2 - In1

Inport: Fornece uma porta de entrada para o subsistema *LMS Channel Estimator*.

Sinal de Entrada: Sinal bipolar complexo baseado em quadro (pilotos).

Sinal de Saída: Sinal bipolar complexo baseado em quadro (pilotos).

A.1.10.4. LMS Filter - Desired

LMS Filter: Adapta os pesos do filtro baseado no algoritmo escolhido para a filtragem do sinal de entrada.

O bloco filtro LMS pode implementar um filtro adaptativo FIR com cinco algoritmos diferentes. O bloco de estimativas dos pesos do filtro, ou coeficientes, necessários para minimizar o erro, $e(n)$, entre o sinal de saída $y(n)$ e o sinal desejado, $d(n)$. Conecta o sinal que é desejável filtrar para a porta de entrada. Este

signal de entrada pode ser um escalar baseado em amostra ou um sinal de canal único baseado em quadro. Conecta o sinal desejado para a porta desejada. O sinal desejado deve ter o mesmo tipo de dados, quadro de status, complexidade e dimensões que o sinal de entrada. A porta de saída emite o sinal de entrada filtrado, que é a estimativa do sinal desejado. A saída da porta de saída tem o mesmo quadro de status do sinal de entrada. A porta de erro gera o resultado da subtração do sinal de saída do sinal desejado.

Parâmetros:

- *Algorithm: Normalized LMS*
- *Output filter weights*

Algoritmo: Escolhe o algoritmo usado para calcular os pesos do filtro.

Pesos dos filtros de saída: Selecionar este item para exportar os pesos do filtro para a porta Wts.

Sinal de entrada:

- Input: Sinal modulado novamente reordenado, quadros com 2 amostras de pilotos. O sinal passou por um canal com Desvanecimento de Multipercursos Rayleigh e foi adicionado Ruído Gaussiano Branco. Remoção das 36 amostras de prefixo cíclico. Domínio da frequência. Remoção das 210 amostras de zeros de enchimento. Remoção dos dados. Só os Pilotos estão sendo considerados e separadamente. Sinal Piloto recebido.
- Desired: Sinal bipolar complexo baseado em quadro (pilotos). O sinal desejado não passou por um canal com Desvanecimento de Multipercursos Rayleigh e nem foi adicionado Ruído Gaussiano Branco. Sinal Piloto desejado.

Sinal de saída:

- Output: A porta de saída emite o sinal de entrada filtrado, que é a estimativa do sinal desejado.
- Error: gera o resultado da subtração do sinal de saída do sinal desejado. O Filtro LMS irá comparar o sinal piloto recebido (input) com o sinal piloto desejado. Calculará a diferença entre estes dois sinais e a partir daí fornecerá pesos que serão úteis ao interpolador para que ele calcule o sinal de dados desejável.
- Wts: pesos do filtro.

A.1.10.5. Terminate

Terminator: Utilizado para mostrar os sinais de saída (*Output*, *Error*).

A.1.10.6. Mux

Mux: Multiplexa sinais escalares ou vetoriais. Multiplexa os pesos do filtro. (saídas das portas Wts).

Parâmetros:

- *Number of inputs*: 2
- *Display option*: bar

A Tabela A.12, na coluna “LMS Channel Estimator - Mux - Number of Inputs” mostra todos os parâmetros *Number of inputs* em função do número de pilotos.

Sinal de entrada: Wts: pesos do filtro.

Sinal de saída: Wts: pesos do filtro multiplexados.

A.1.10.7. 1 - Out1

Outport: Fornece uma porta de saída para o subsistema LMS Channel Estimator.

Sinal de entrada: Wts: pesos do filtro multiplexados.

Sinal de saída: Wts: pesos do filtro multiplexados.

A.1.11. S_OFDM – Interpolator

M-S_Function: Bloco definido pelo usuário escrito usando o MATLAB S-Function API.

O interpolador utilizará os pesos fornecidos pelo filtro LMS e a partir disso calculará o sinal de dados desejável.

Parâmetros:

- *Função S_OFDM* (detalhes no Apêndice C)

Sinal de entrada: Wts: pesos do filtro multiplexados.

Sinal de saída: quadros com 300 bits de dados. Sinal de dados desejável.

A.1.12. X-Product

Produto: multiplica ou divide entradas.

Parâmetro:

- *Number of inputs: 2*

Sinal de entrada: Sinal de dados desejável (estimado no interpolador) e sinal de dados recebido (sinal que passou por um canal com Desvanecimento de Multipercursos Rayleigh e foi adicionado Ruído Gaussiano Branco). Quadros com 300 bits de dados.

Sinal de saída: equalização - multiplicação dos sinais de dados estimado e recebido. Quadros com 300 bits de dados.

A.1.13. Denormalize

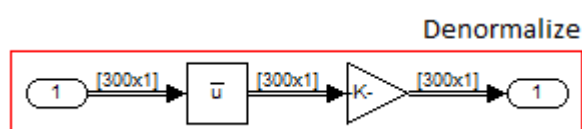


Figura A.37 - Denormalizador (*Denormalize*).

A.1.13.1. 1-Inport

Inport: Fornece uma porta de entrada para um subsistema ou modelo.

Sinal de entrada: sinais de dados estimado e recebido multiplicados. Quadros com 300 bits de dados.

Sinal de saída: sinais de dados estimado e recebido multiplicados. Quadros com 300 bits de dados.

A.1.13.2. U-Math

Math: Funções matemáticas, incluindo logaritmo, exponencial, potência e módulo.

Parâmetro:

- *Function: conj*
conj: complexo conjugado.

Sinal de entrada: sinais de dados estimado e recebido multiplicados. Quadros com 300 bits de dados.

Sinal de saída: sinais de dados estimado e recebido multiplicados. Quadros com 300 bits de dados.

A.1.13.3. K-Gain

Ganho (*Gain*): *Element – wise gain* ($y = K.*u$) ou *matrix gain* ($y = K*u$ or $y = u*K$).

Multiplica a entrada por uma constante. A entrada e o ganho podem ser um escalar, vetor ou matriz.

A entrada e o ganho são multiplicados, e o resultado é convertido para o tipo de dados de saída utilizando o arredondamento especificado.

Parâmetros:

- Ganho (*Gain*): $\text{sqrt}(10)$
- Multiplicação (*Multiplication*): *Element – wise gain* ($y = K.*u$)

Multiplicação: Especifica o modo de multiplicação (elemento ou matriz). Elemento-wise ($K \cdot u$) - Cada elemento de entrada é multiplicado por cada elemento do ganho. Se necessário, o bloco realiza expansões, de modo que a entrada e o ganho tenham as mesmas dimensões.

sqrt : *square root* – raiz quadrada.

Sinal de entrada: sinais de dados estimado e recebido multiplicados. Quadros com 300 bits de dados.

Sinal de saída: sinais de dados estimado e recebido multiplicados. Quadros com 300 bits de dados.

A.1.13.4. 1-Out1

Outport: Fornece uma porta de saída para o subsistema Denormalizador (*Denormalize*).

Sinal de entrada: sinais de dados estimado e recebido multiplicados. Quadros com 300 bits de dados.

Sinal de saída: sinais de dados estimado e recebido multiplicados. Quadros com 300 bits de dados.

A.1.14. Received Signal

Discrete-Time Scatter Plot Scope: A Aplicação de Plotagem de Dispersão Discreta no Tempo é usada para exibir uma constelação de sinais modulados em seu espaço de sinal, plotando a componente em fase versus a componente em quadratura.

Use '*Samples per Symbol*' definido para 1 para visualizar sinais após a detecção ou para sinais sem *oversampling*. Para sinais *oversampled*, a entrada é dizimada por '*Samples per Symbol*' saltando para amostras iniciais '*Offset*'.

Parâmetros:

- *Samples per symbol*: 1
- *Offset (samples)*: 0
- *Points displayed*: 160
- *New points per display*: 80

A.1.15. Rectangular QAM

Rectangular QAM Demodulator Baseband (Demodulador Banda Base QAM Retangular): Demodula o sinal de entrada usando o método de modulação em amplitude de quadratura retangular.

O valor do número *M*-ary deve ser uma potência inteira de dois.

A entrada pode ser bits ou inteiros. No caso de entrada de bit baseada em quadro, a largura da entrada deve ser igual a um múltiplo inteiro do número de bits por símbolos. (No caso de entrada de bit baseada em amostra, a largura da entrada deve ser igual ao número de bits por símbolo).

Para entrada de inteiros baseada em quadro, a entrada deve ser um vetor coluna. (Para entrada de inteiros baseada em amostra, a entrada deve ser um escalar).

No caso de entrada baseada em quadro, a largura do quadro de entrada deve ser igual ao produto do número de símbolos e o valor de amostras por símbolo.

(No caso de entrada baseada em amostra, o tempo da amostra de saída deve ser igual ao período do símbolo dividido pelo valor de amostras por símbolo.)

Modulação: 16 QAM.

Parâmetros:

- *M-ary number*: 16 (Número de bits por símbolo: 4)
- Tipo de Entrada (*Input type*): Bit (igual à amostra)
- *Constellation ordering*: Gray
- Entradas baseadas em quadro (*Frame-based inputs*)
- Método de normalização (*Normalization method*): *Min. Distance between symbols*
- Distância mínima (*Minimum distance*): 2

Sinal de entrada: sinais de dados estimado e recebido multiplicados. Quadros com 300 bits de dados. Sinal modulado.

Sinal de saída: Sinal demodulado, quadros com 1200 amostras.

A.1.16. Unipolar to Bipolar Converter

Conversor Unipolar para Bipolar: converte um sinal unipolar no intervalo [0, M-1], onde M é o M-ário número, em um sinal bipolar.

Conversor Unipolar para Bipolar: Converte um sinal unipolar no intervalo [0, M-1], onde M é o M-ário número, em um sinal bipolar.

O bloco Conversor Unipolar para Bipolar mapeia o sinal de entrada unipolar para um sinal de saída bipolar. Se a entrada consiste de números inteiros entre 0 e M-1, onde M é o parâmetro M-ário número, então a saída é composto de números inteiros entre - (M-1) e M-1. Se M é par, então a saída é ímpar. Se M é ímpar, então a saída é par. Este bloco é projetado para funcionar apenas quando o valor de entrada está dentro do conjunto (0,1,2 ... (M-1)), onde M é o parâmetro número M-ário. Se o valor da entrada está fora deste conjunto de números inteiros a saída não pode ser válida.

Parâmetros:

- *M-ary number*: 2
- Polaridade (*Polarity*): Positiva (*Positive*)

A.1.17.2. Zero-Order Hold

Manter ordem zero.

A.1.17.3. Demux

Demultiplexador: Demultiplexa sinais vetoriais em escalares ou em vetores menores.

A.1.17.4. Sum

Soma: Adiciona ou subtrai entradas.

A.1.17.5. APP Decoder

APP Decoder: Um decodificador de probabilidade a posteriori (APP).

Decodificador convolucional (*Deconvolucional Encoder*): decodifica dados binários convolucionalmente. Usa a função *poly2trellis* (comprimento de restrição, polinômio gerador de código (em número octal) e polinômio de conexão de *feedback* (em octal)).

O bloco do decodificador convolucional decodifica uma sequência de vetores binários de entrada para produzir uma sequência de vetores binários de saída. Este bloco pode processar múltiplos símbolos ao mesmo tempo.

Parâmetros:

- *Trellis structure*: *poly2trellis*(4,[15],13)
- Modo de Operação (*Operation mode*): *Truncated (reset every frame)*.
- *Algorithm*: *Max*
- *Number of scaling bits*: 3

Trellis structure: estrutura do MATLAB que contém a descrição do decodificador convolucional.

Truncated reset every frame mode: o bloco trata cada quadro independentemente, isto é, o estado do codificador é “resetado” (zerado) no início de cada quadro.

Sinal de Entrada: Sinal bipolar, quadros com 1200 amostras.

Sinal de Saída: quadros com 600 amostras decodificados convolucionalmente.

A.1.17.6. Terminator

Terminator: Utilizado para mostrar os sinais de saída.

A.1.17.7. Sum

Soma: Adiciona ou subtrai entradas.

A.1.17.8. Discrete Pulse Generator

Gerador de Pulso Discreto:

Pulsos de saída:

If (t >=PhaseDelay)&&Pulse is on

Y(t) = Amplitude

Else

Y(t) = 0

End

A.1.17.9. X-Product

Produto: multiplica ou divide entradas.

A.1.17.10. z^{-600} – Delay

Delay: Atraso na entrada discreta no tempo por um determinado número de amostras (ou quadros).

A.1.17.11. Buffer

Buffer: Converte amostras escalares para uma saída de quadros a uma taxa mais baixa que das amostrass. Pode-se converter um quadro para um tamanho menor ou maior.

A.1.17.12. Selector

Selector: Seleciona ou reordena elementos específicos de um sinal de entrada multidimensional. O índice de cada elemento é identificado de uma porta de entrada ou caixa de diálogo.

A.1.17.13. Hard Decision

Probabilidade de transformação binária.

A.1.17.14.1-Out1

Outport: Fornece uma porta de saída para o subsistema Turbo Decoder 2:1.

Sinal de Saída: quadros com 600 amostras decodificados convolucionalmente.

A.1.18. Error Rate Calculation

Cálculo da Taxa de erro: Calcula a taxa de erro dos dados recebidos, comparando-os com uma versão atrasada da transmissão dos dados. A saída do bloco são elementos, constituídos pela taxa de erro, seguido pelo número de erros detectados e o número total de símbolos comparados. Esse vetor pode ser enviada para qualquer espaço de trabalho ou porta de saída.

O bloco cálculo da taxa de erro compara os dados de entrada de um transmissor com os dados de entrada de um receptor. Ele calcula a taxa de erro como uma estatística dinâmica, dividindo o número total de pares desiguais dos elementos de dados pelo número total de elementos de dados da entrada de uma fonte.

Pode-se usar este bloco para calcular taxa de erro de símbolos ou taxa de erro de bits, porque ele não considera a magnitude da diferença entre os elementos de dados de entrada. Se as entradas são bits, em seguida, o bloco calcula a taxa de erro de bit. Se as entradas são símbolos, em seguida, ele calcula a taxa de erro de símbolo.

A quantidade de frames é dada em função de parâmetros, que estão definidos no bloco *Error Rate Calculation*, que calcula a taxa de erro dos dados recebidos, comparando-os com uma versão atrasada da transmissão dos dados. A saída do bloco são elementos, constituídos pela taxa de erro, seguido pelo número de erros detectados e o número total de símbolos comparados. Esse vetor pode ser enviado para qualquer espaço de trabalho ou porta de saída. No caso da plataforma OFDM, o vetor constituído pela taxa de erro está sendo enviado para o *Display* (mostrador numérico) e para o *Sink Block Parameters (Signal To Workspace)* a fim de que possibilite a plotagem dos gráficos (que serão disponibilizados adiante).

O parâmetro “*Stop simulation*” (do bloco *Error Rate Calculation*) interrompe a simulação ao detectar um número alvo pré-definido de erros ou um número máximo de símbolos.

Parâmetros:

- *Receive delay*: 600
- *Computation delay*: 0
- *Computation mode*: *Entire frame*.
- *Output data*: *Port*
- *Stop Simulation*.
- *Target number of errors*: 50.000
- *Maximum number of symbols*: 10^8

Receive delay: Número de amostras em que os dados são recebidos em atraso em relação aos dados transmitidos.

Computation delay: Número de amostras que o bloco deve ignorar, no início da comparação.

A interrupção da simulação é dada quando qualquer um dos dois parâmetros acima for atingido primeiro.

O Bloco *Error Rate Calculation* compara o sinal de dados recebido com o sinal de dados desejado. Calculará a diferença entre estes dois sinais e a partir daí fornecerá a taxa de erro.

A.1.19. Display

Display: Mostrador numérico de valor de entrada.

A saída do bloco são elementos, constituídos pela taxa de erro, seguido pelo número de erros detectados e o número total de símbolos comparados.

A.1.20. Sink Block Parameters: Signal To Workspace



Figura A.39 - *Sink Block Parameters: Signal To Workspace*.

Escreve entrada para especificar vetor na área de trabalho principal do MATLAB. Os dados não estão disponíveis até a simulação ser parada ou pausada.

O vetor taxa de erro pode ser enviado para qualquer espaço de trabalho ou porta de saída. No caso da plataforma OFDM, o vetor constituído pela taxa de erro está sendo enviado para o *Display* (mostrador numérico) e para o *Sink Block Parameters (Signal To Workspace)* a fim de que possibilite a plotagem dos gráficos (que serão disponibilizados adiante).

A.2. Plataforma OFDM com CP – 16 Pilotos

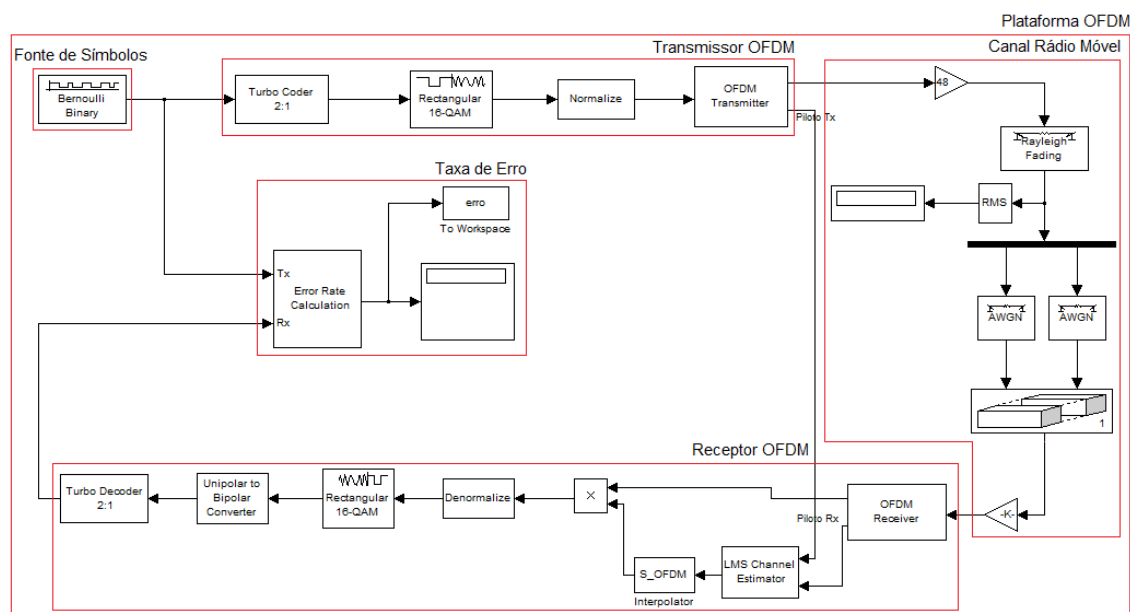


Figura A.40 - Plataforma OFDM com CP - 16 Pilotos.

Plataforma para o levantamento de curvas de Taxa de Erro x E_b/N_0 com consideração de aproveitamento da redundância dos bits de "cyclic prefix" em enlaces OFDM.

A.2.1. Demux

Demultiplexador: Demultiplexa sinais vetoriais em escalares ou em vetores menores.

Parâmetros:

- *Number of outputs:* [rep 548-rep].

Repv_{et} = [1 248].

Rep = repvet(i).

A.2.2. AWGN 1 e 2

Canal AWGN: Adiciona Ruído Gaussiano Branco para o sinal de entrada. Os sinais de entrada e de saída podem ser reais ou complexos. O bloco suporta entrada de multicanais.

O bloco Canal AWGN adiciona ruído Gaussiano branco para um sinal de entrada real ou complexo. Quando o sinal de entrada é real, este bloco adiciona ruído Gaussiano real e produz um sinal de saída real. Quando o sinal de entrada é complexo, este bloco adiciona ruído Gaussiano complexo e produz um sinal de saída complexo. Este bloco herda o período de amostragem do sinal de entrada.

AWGN 1

Parâmetros:

- *Initial seed*: 12345
- *Mode*: *Signal to noise ratio* (E_b/N_0)
- $E_b/N_0 = 5$ à 15 dB
- $E_b/N_0 = E_b/N_0 + 3$ dB
- *Number of bits per symbol*: $(1/2)*4 = (\text{Turbo coder}) * \text{Modulador}$
- *Input signal power (watts)*: 1
- *Symbol period(s)*: $(1/7.68e+06)*(600/\text{rep})$

Repvect = [1 248].

Rep = repvect(i).

E_b/N_0 : a razão entre a energia do bit e a densidade espectral de potência de ruído do símbolo, em decibéis.

Number of bits per symbol: $(1/2)*4 = (\text{Turbo coder}) * \text{Modulador}$.

Symbol period(s): $(1/7.68e+06)*(600/\text{rep}) = (\text{Período da amostra}) * (\text{amostras por quadro/amostras real por quadro})$.

AWGN 2

Parâmetros:

- *Initial seed*: 12345
- *Mode*: *Signal to noise ratio* (E_b/N_0)
- $E_b/N_0 = 5$ à 15 dB

- $E_b/N_0 = E_b/N_0$
- *Number of bits per symbol*: $(1/2)*4 = (\text{Turbo coder}) * \text{Modulador}$
- *Input signal power (watts)*: 1
- *Symbol period(s)*: $(1/7.68e+06)*(600/(548-\text{rep}))$

Repvect = [1 248].

Rep = repvect(i).

E_b/N_0 : a razão entre a energia do bit e a densidade espectral de potência de ruído do símbolo, em decibéis.

Number of bits per symbol: $(1/2)*4 = (\text{Turbo coder}) * \text{Modulador}$.

Symbol period(s): $(1/7.68e+06)*(600/(548-\text{rep})) = (\text{Período da amostra}) * (\text{amostras por quadro}/\text{amostras real por quadro})$.

Obs.: Com $\text{rep} = 1$, a quantidade de bits que passava no AWGN 1 era 1 bit e a quantidade de bits que passava no AWGN 2 era 547 bits.

Com $\text{rep} = 248$, a quantidade de bits que passava no AWGN 1 era 248 bits e a quantidade de bits que passava no AWGN 2 era 300 bits.

A.2.3. Matrix Concatenation 1 ()

Concatenar: Concatena sinais de entrada do mesmo tipo de dados para criar um sinal de saída contíguo.

O bloco Concatenar concatena os sinais em suas entradas para criar um sinal de saída cujos elementos residem em locais contíguos na memória. Este bloco opera em modo vetor ou modo de concatenação de matriz multidimensional, dependendo da configuração do seu parâmetro *Mode*. Em ambos os casos, as entradas são concatenados de cima para baixo ou da esquerda para a direita.

Parâmetros:

- Número de entradas (*Number of inputs*): 2
- Modo (*Mode*): Matriz multidimensional (*Multidimensional array*)

- Dimensão da concatenação (*Concatenate dimension*): 1

Número de entradas (*Number of inputs*): Número de entradas do bloco.

Modo (*Mode*): Especifica o tipo de concatenação realizada por este bloco. As opções são Matriz multidimensional (*Multidimensional array*) ou Vetor (*Vector*).

Dimensão da concatenação (*Concatenate dimension*): Especifica a dimensão de saída ao longo do qual concatena as matrizes de entrada. Por exemplo, para concatenar as matrizes de entrada vertical ou horizontalmente, deve-se especificar 1 ou 2, respectivamente. Esta opção só aparece se for selecionado matriz Multidimensional para o parâmetro modo.

A.3. Conclusão

Foi estudado detalhadamente a plataforma OFDM.

Foi apresentado um modelo em banda base de um sistema OFDM, com divisão do espectro de frequências disponível em várias subportadoras.

O modelo consiste basicamente na geração do sinal binário que será transmitido, codificação e decodificação, modulação e demodulação, inserção e remoção de pilotos no domínio da frequência, inserção e remoção dos zeros de enchimento, uso de IFFT e FFT, inserção e remoção de prefixo cíclico, canal com desvanecimento por multipercursos *Rayleigh* e introdução de ruído gaussiano (AWGN).

O objetivo da Simulação da plataforma OFDM era a obtenção dos valores e gráficos de Taxa de Erro em função do E_b/N_0 e da quantidade de pilotos inserida.

O bloco *Error Rate Calculation* é que calcula a taxa de erro dos dados recebidos.

Conforme visto, a quantidade de frames é dada em função de parâmetros, que estão definidos no bloco *Error Rate Calculation*, que calcula a taxa de erro dos dados recebidos, comparando-os com uma versão atrasada da transmissão dos dados. A saída do bloco são elementos, constituídos pela taxa de erro, seguido pelo número de erros detectados e o número total de símbolos comparados. Esse vetor pode ser enviado para qualquer espaço de trabalho ou porta de saída. No caso da plataforma OFDM, o vetor constituído pela taxa de erro está sendo enviado para o *Display* (mostrador numérico) e para o *Sink Block Parameters* (*Signal To*

Workspace) a fim de que possibilite a plotagem dos gráficos (que serão disponibilizados adiante).

O parâmetro “*Stop simulation*” (do bloco *Error Rate Calculation*) interrompe a simulação ao detectar um número alvo pré-definido de erros ou um número máximo de símbolos.

B. APÊNDICE - TRANSMISSOR OFDM, ESTIMADOR DE CANAL LMS, SINAL RECEBIDO E ESCOPO DO ESPECTRO

B.1. Introdução

Ao todo, foram desenvolvidas 17 plataformas OFDM, cada uma referente aos diferentes números de pilotos (2, 3, 4, 5, 6, 7, 11, 13, 16, 21, 26, 31, 51, 61, 76, 101 e 151) considerados.

Este Apêndice apresenta todos os Transmissores OFDM, os Estimadores de Canal LMS, e apresenta, também, o Sinal Recebido e o Escopo do Espectro, considerando 13 pilotos.

B.2. Transmissor OFDM e Estimador de Canal LMS

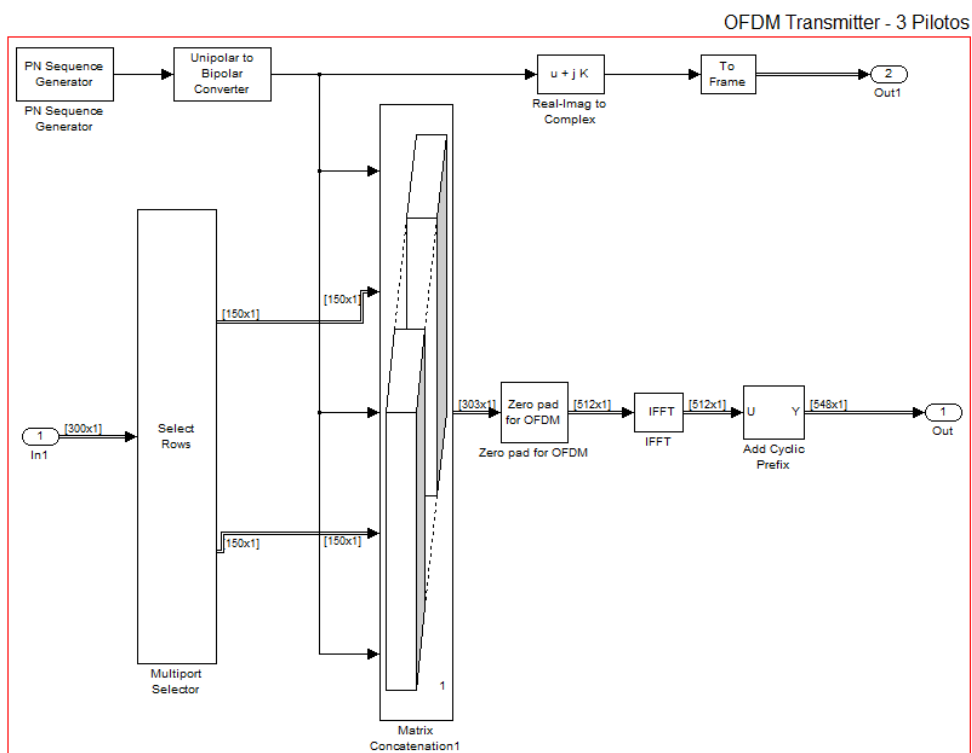


Figura B.1 - Transmissor OFDM - 3 Pilotos.

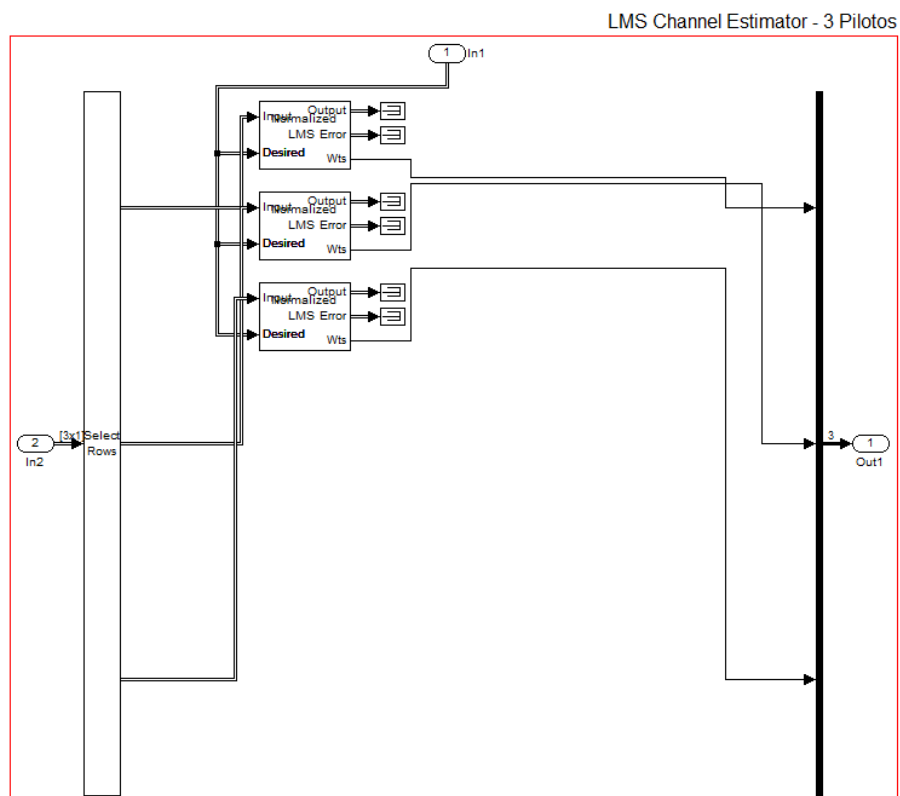


Figura B.2 - Estimador de Canal LMS - 3 Pilotos.

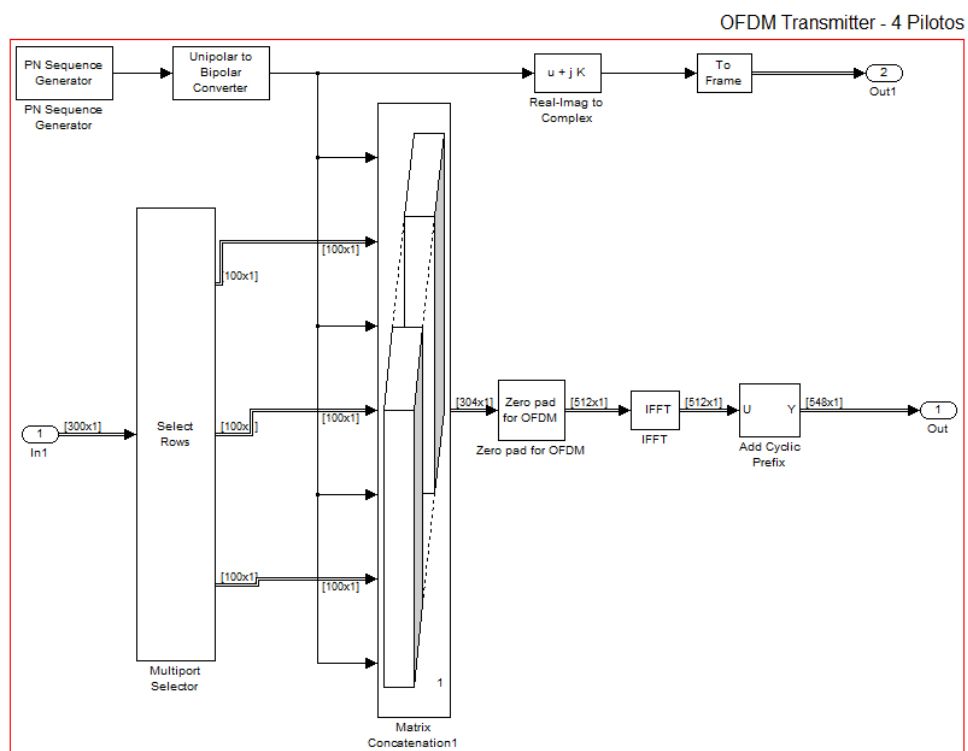


Figura B.3 - Transmissor OFDM - 4 Pilotos.

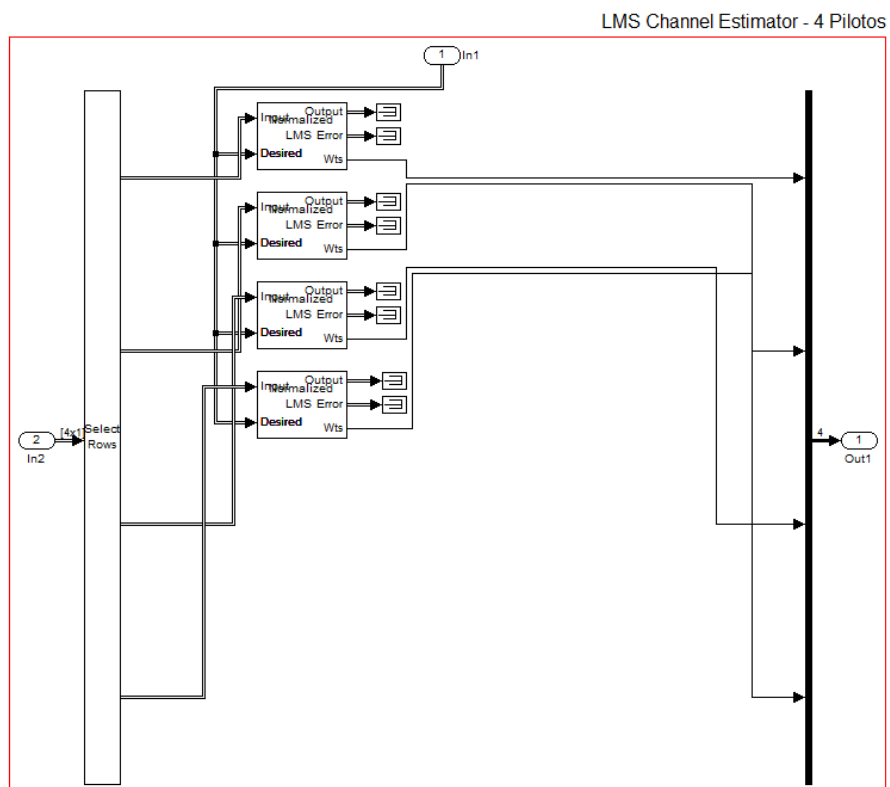


Figura B.4 - Estimador de Canal LMS - 4 Pilotos.

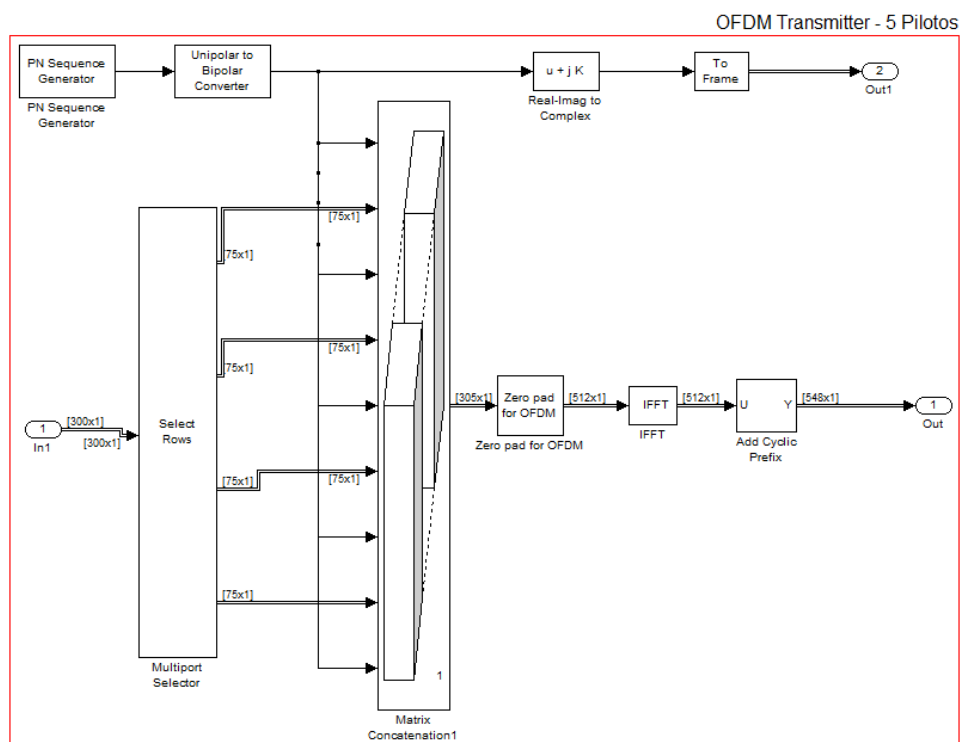


Figura B.5 - Transmissor OFDM - 5 Pilotos.

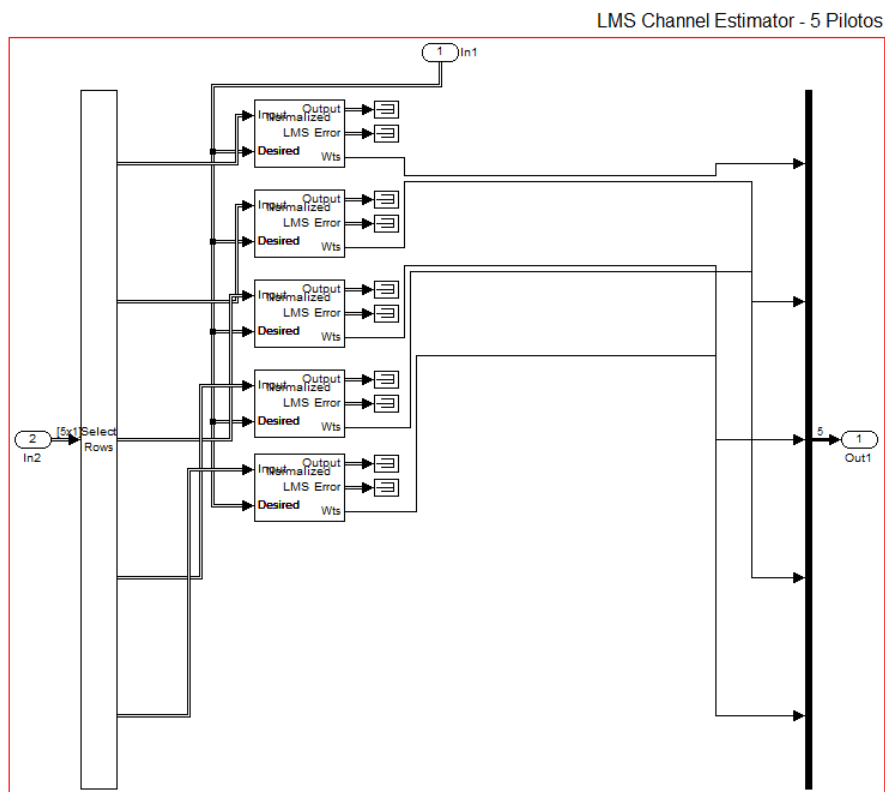


Figura B.6 - Estimador de Canal LMS - 5 Pilotos.

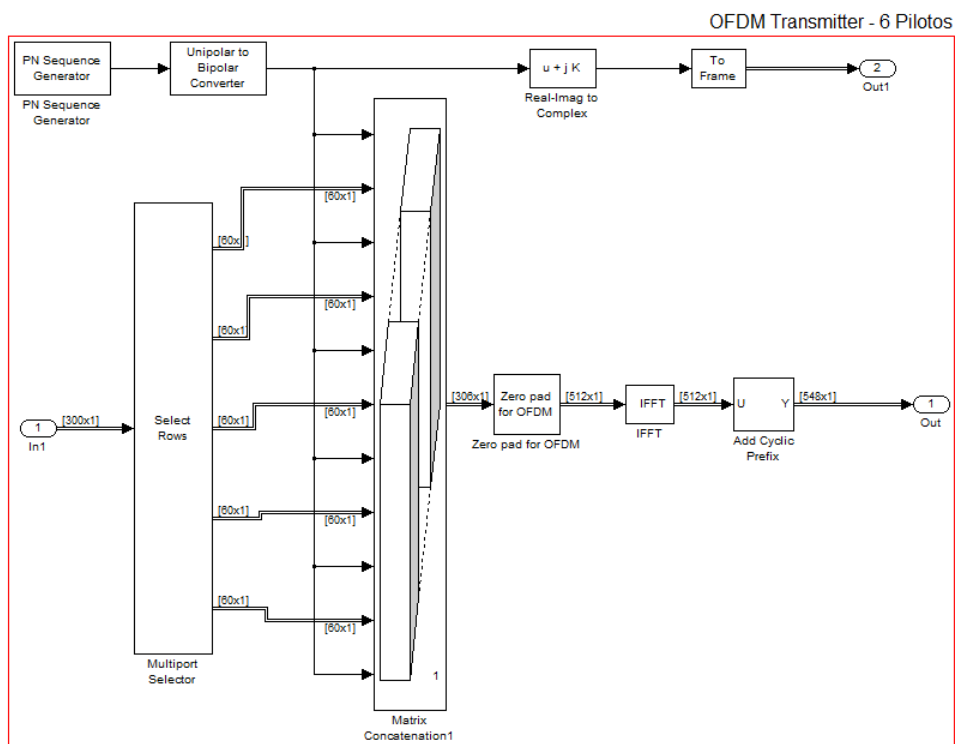


Figura B.7- Transmissor OFDM - 6 Pilotos.

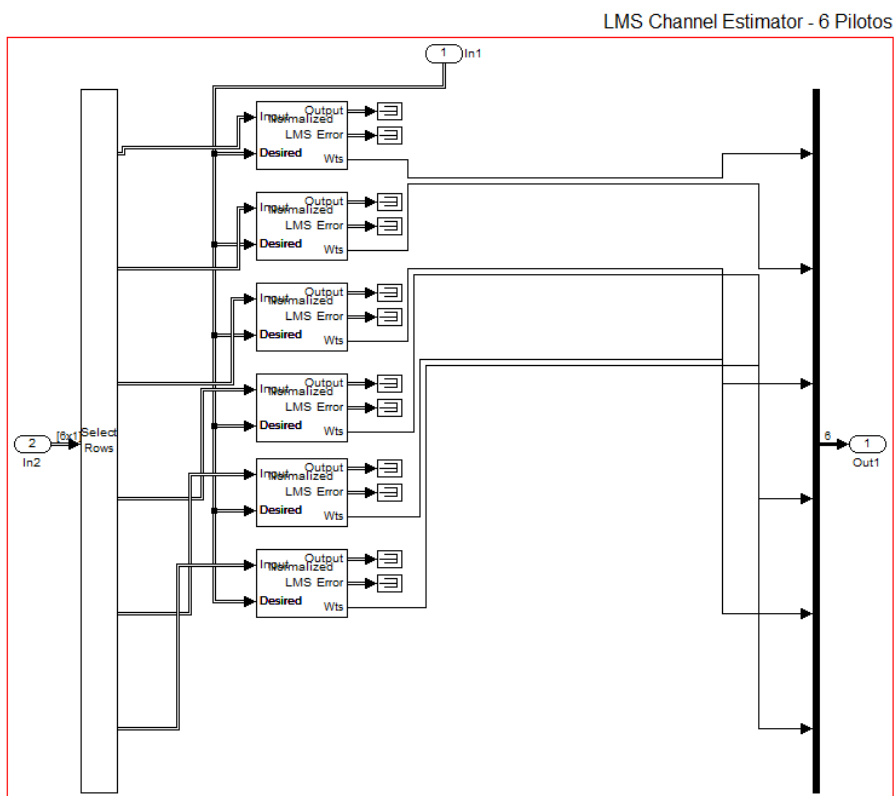


Figura B.8 - Estimador de Canal LMS - 6 Pilotos.

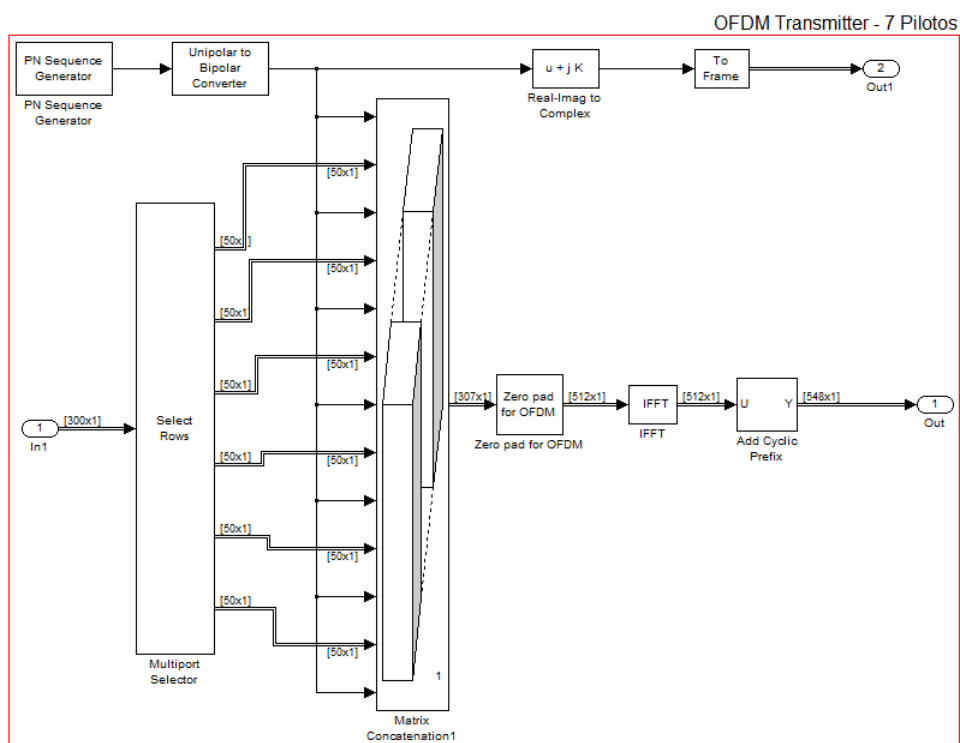


Figura B.9 - Transmissor OFDM - 7 Pilotos.

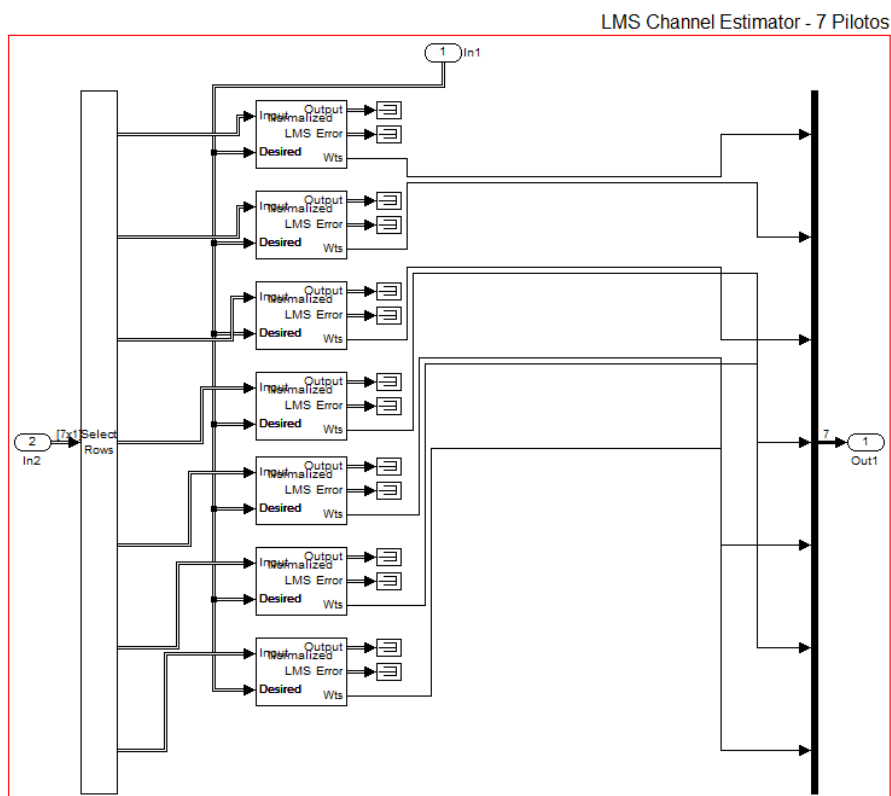


Figura B.10 - Estimador de Canal LMS - 7 Pilotos.

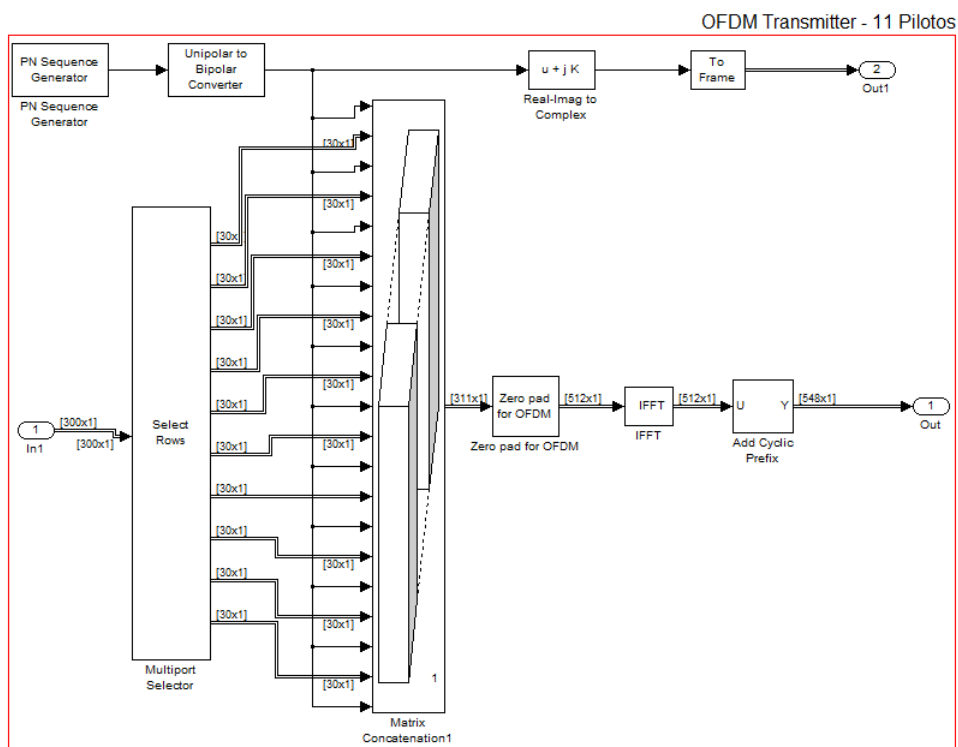


Figura B.11- Transmissor OFDM - 11 Pilotos.

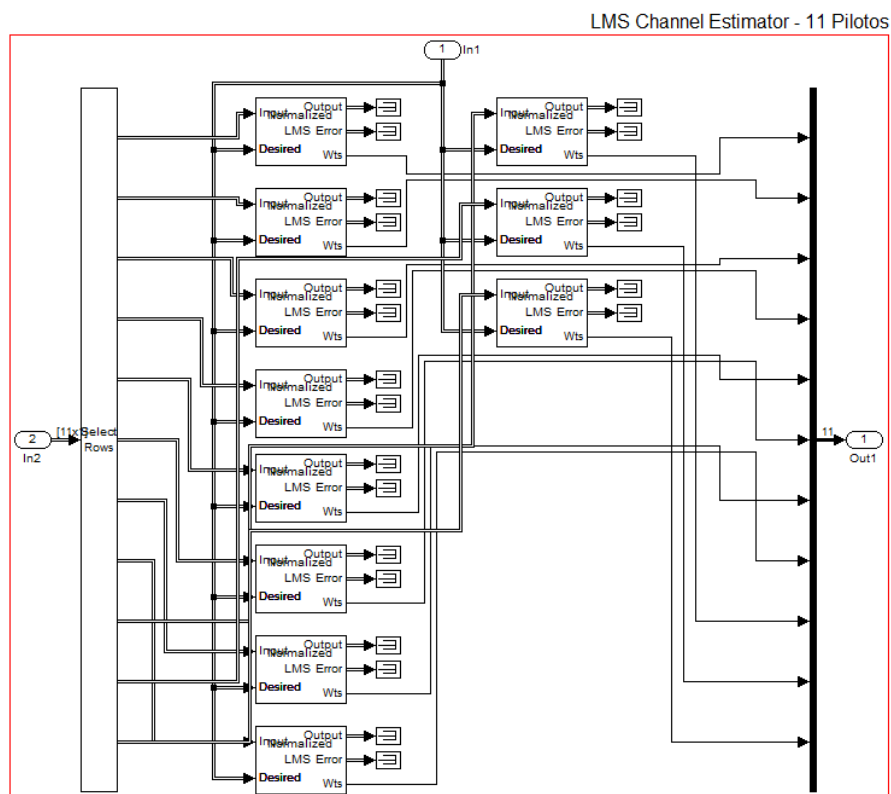


Figura B.12 - Estimador de Canal LMS - 11 Pilotos.

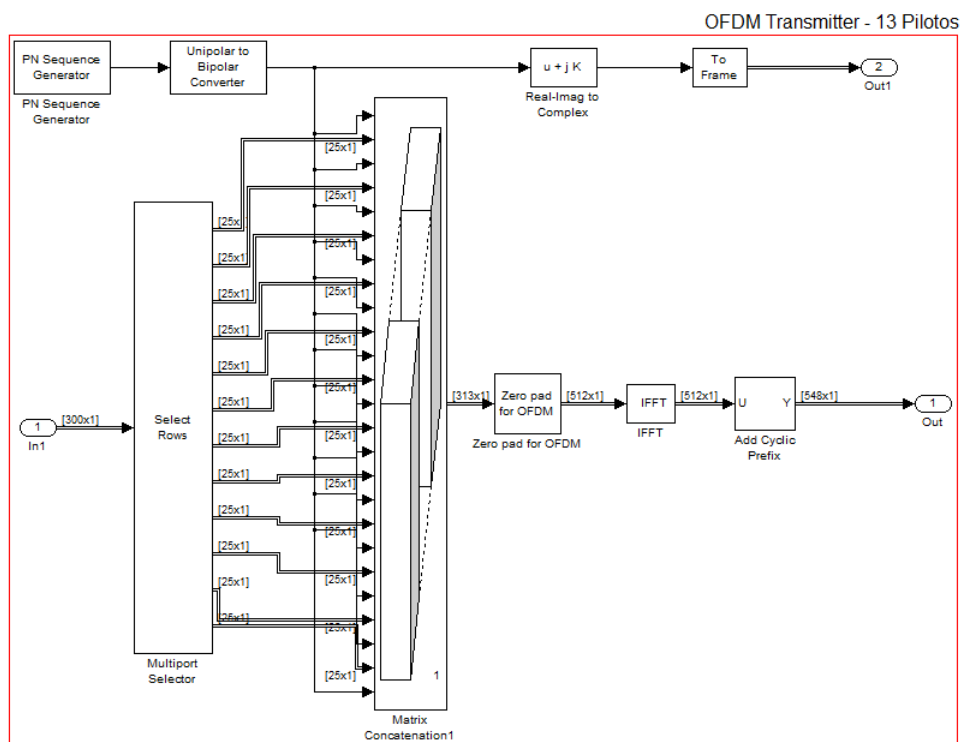


Figura B.13 - Transmissor OFDM - 13 Pilotos.

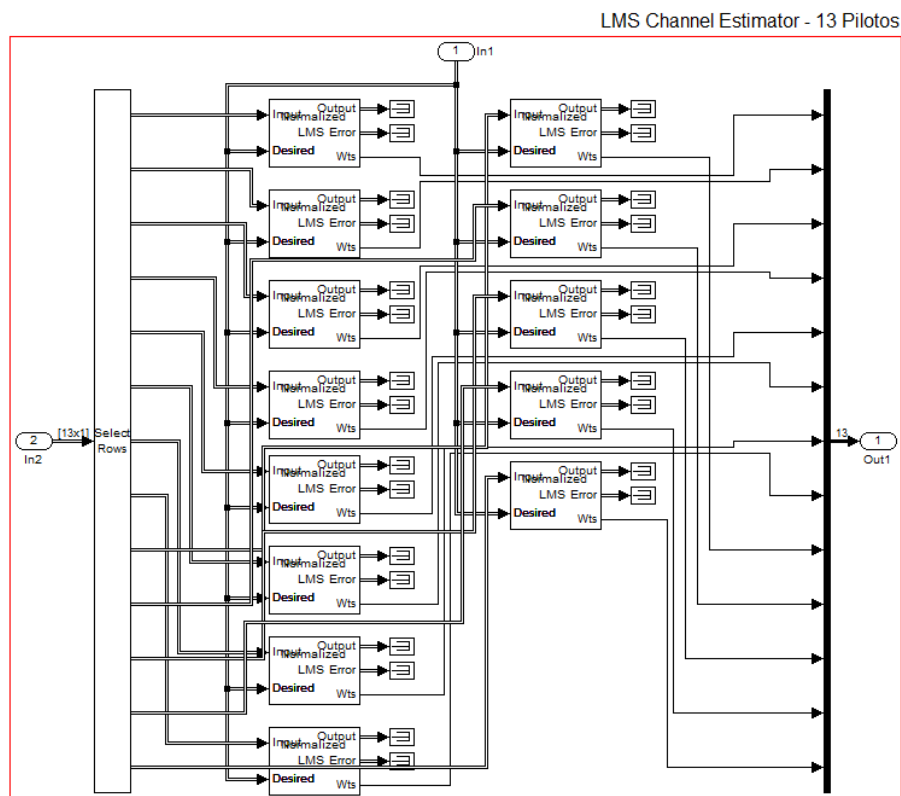


Figura B.14 - Estimador de Canal LMS - 13 Pilotos.

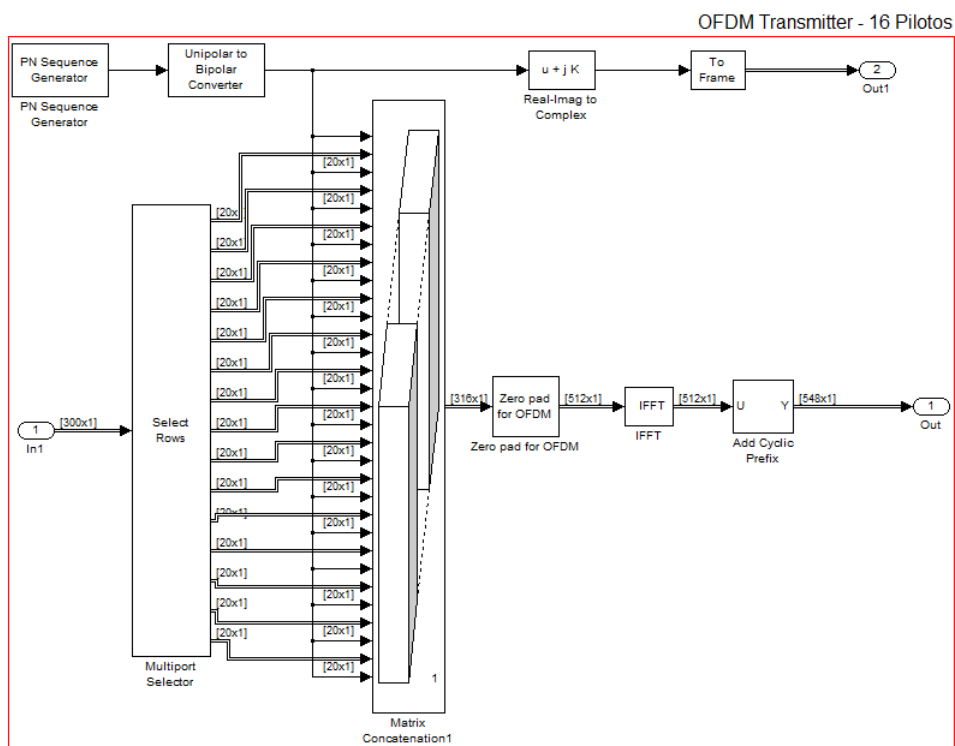


Figura B.15 - Transmissor OFDM - 16 Pilotos.

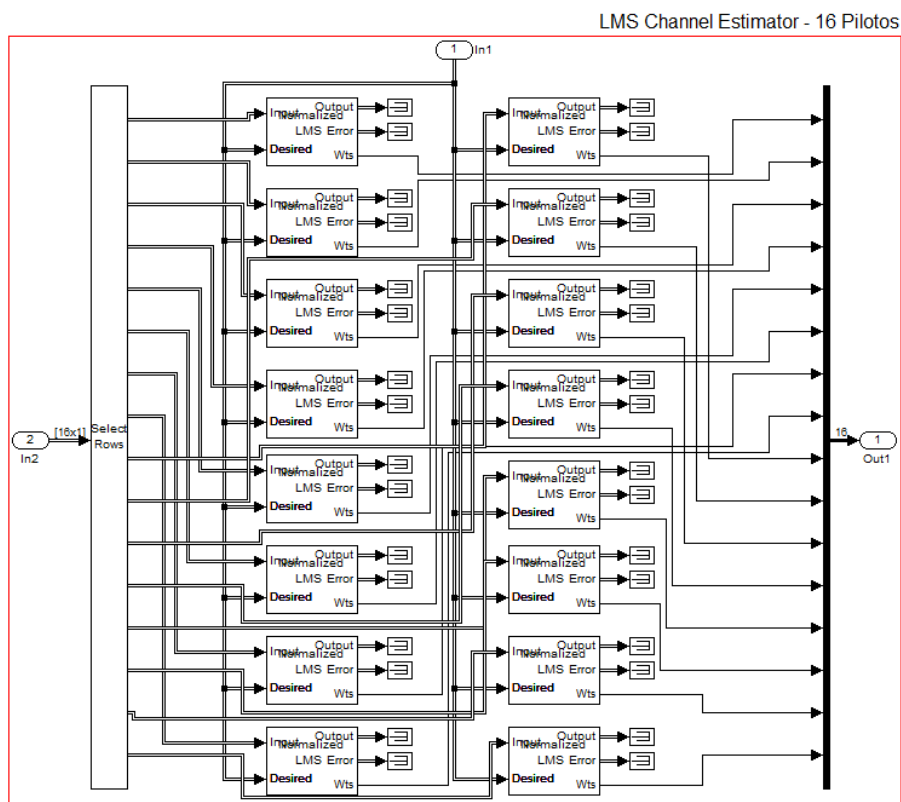


Figura B.16 - Estimador de Canal LMS - 16 Pilotos.

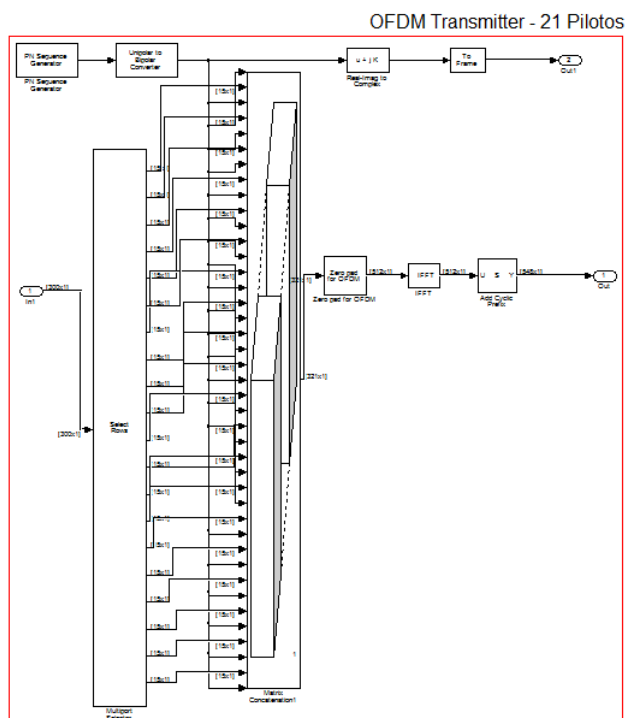


Figura B.17 - Transmissor OFDM - 21Pilotos.

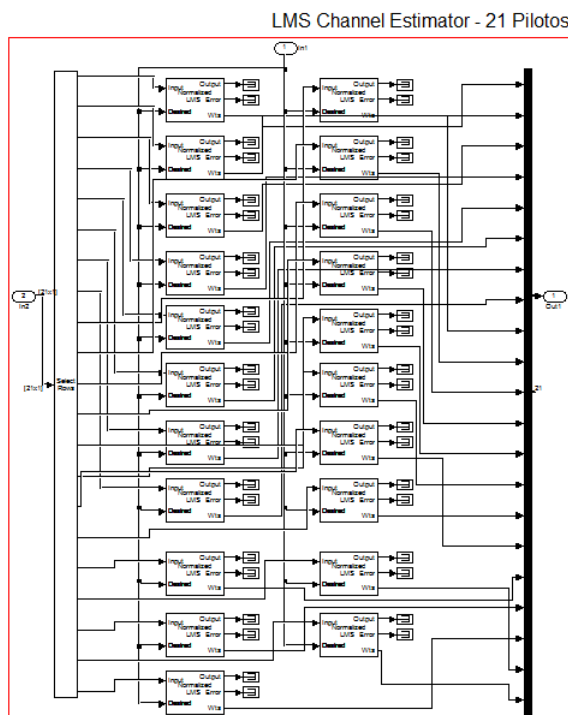


Figura B.18 - Estimador de Canal LMS - 21 Pilotos.

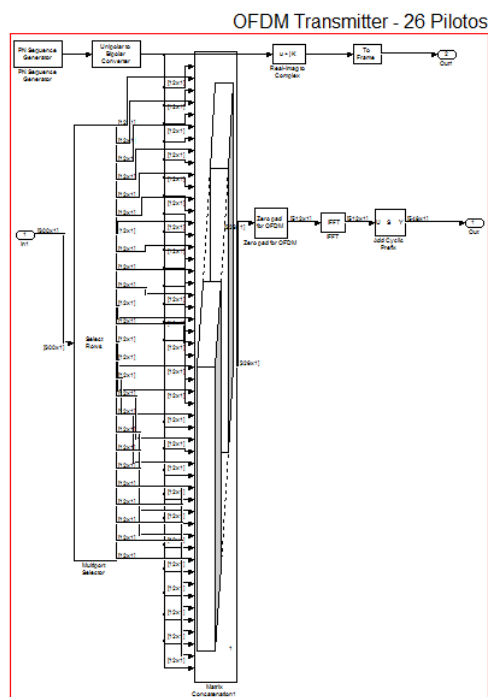


Figura B.19 - Transmissor OFDM - 26 Pilotos.

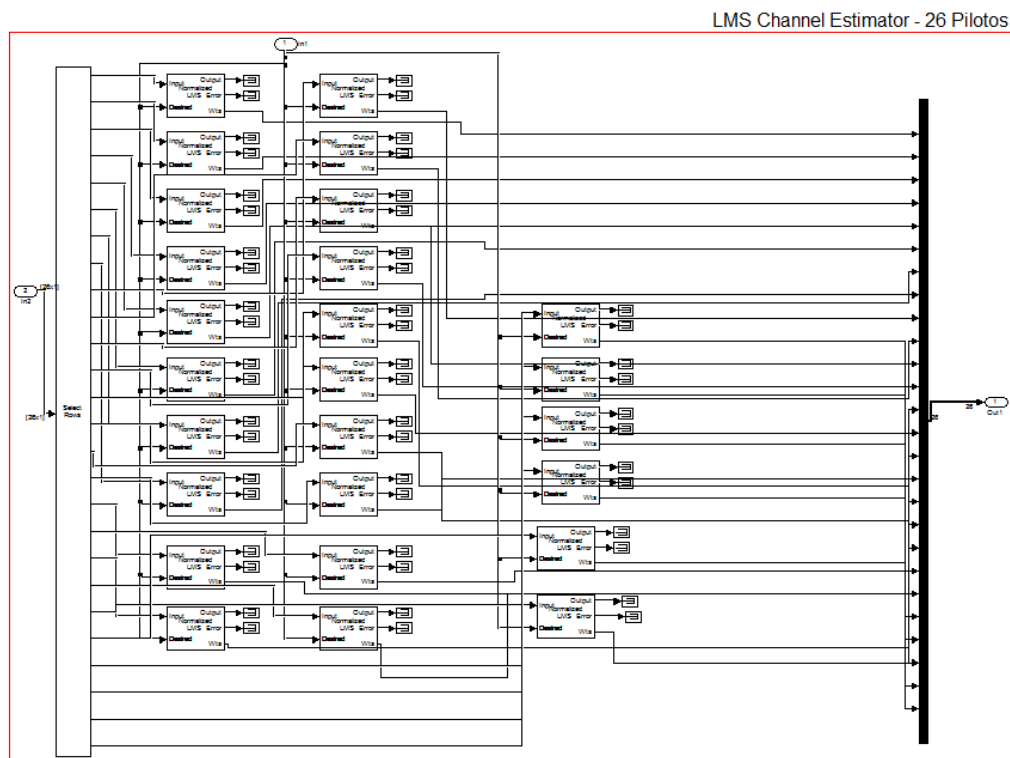


Figura B.20 - Estimador de Canal LMS - 26 Pilotos.

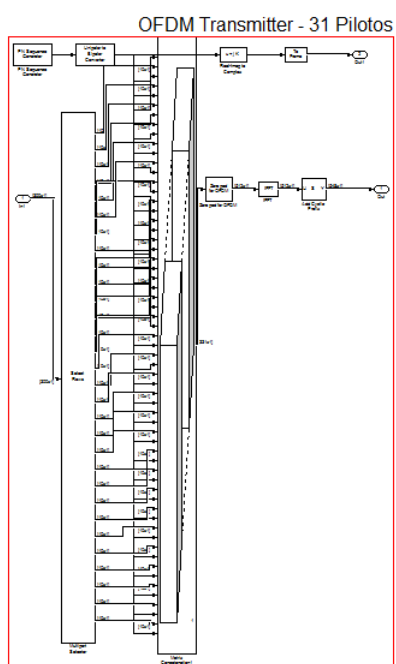


Figura B.21- Transmissor OFDM - 31 Pilotos.

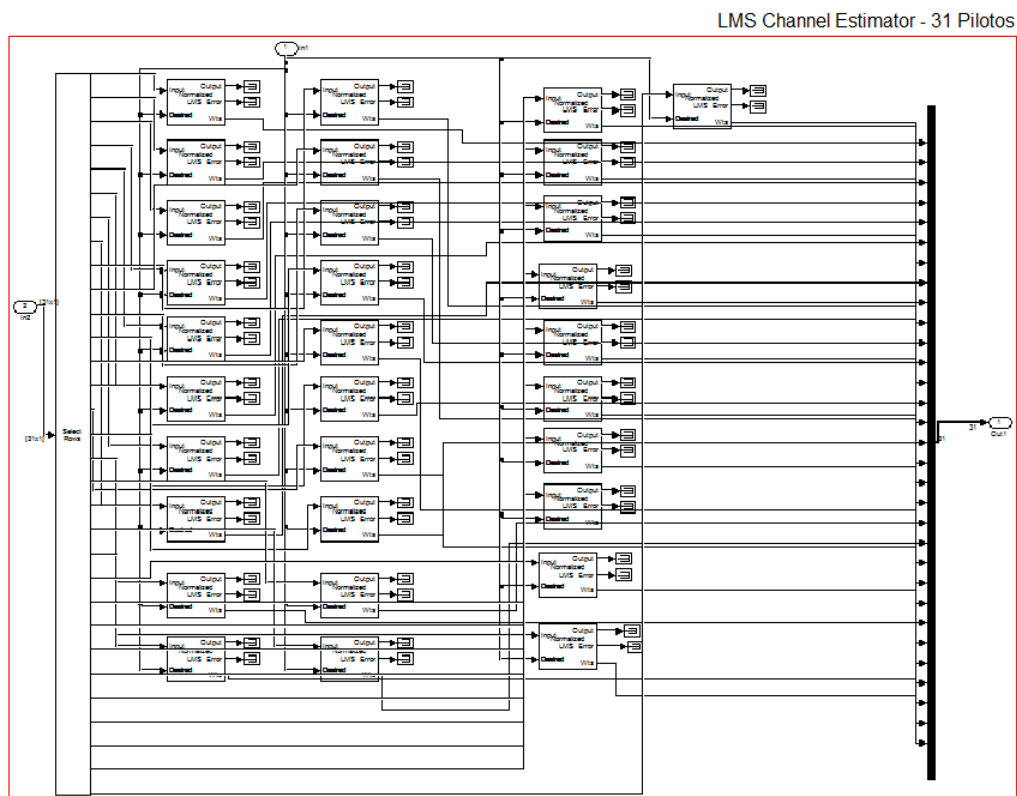


Figura B.22 - Estimador de Canal LMS - 31 Pilotos.

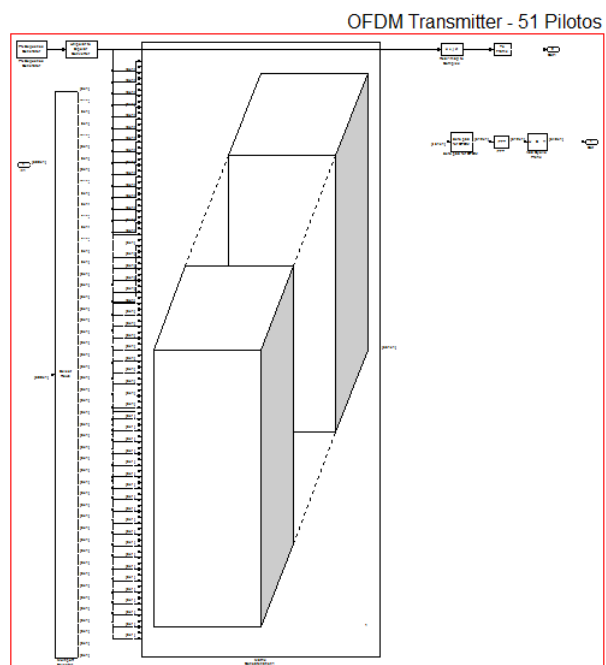


Figura B.23 - Transmissor OFDM - 51 Pilotos.

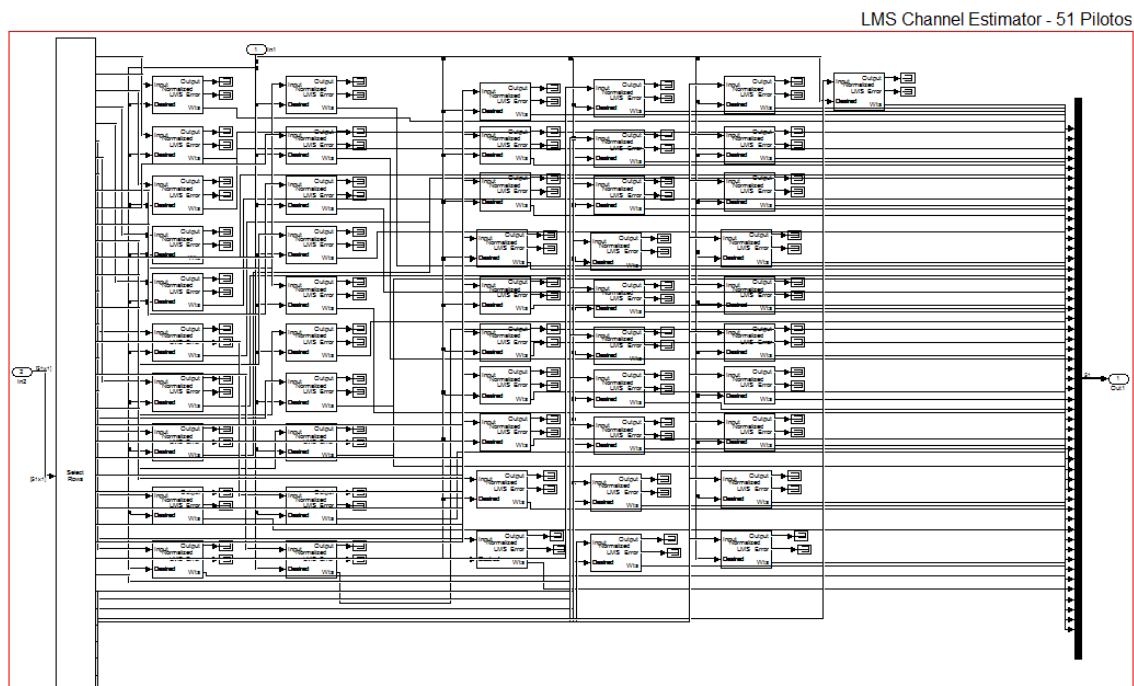


Figura B.24 - Estimador de Canal LMS - 51 Pilotos.

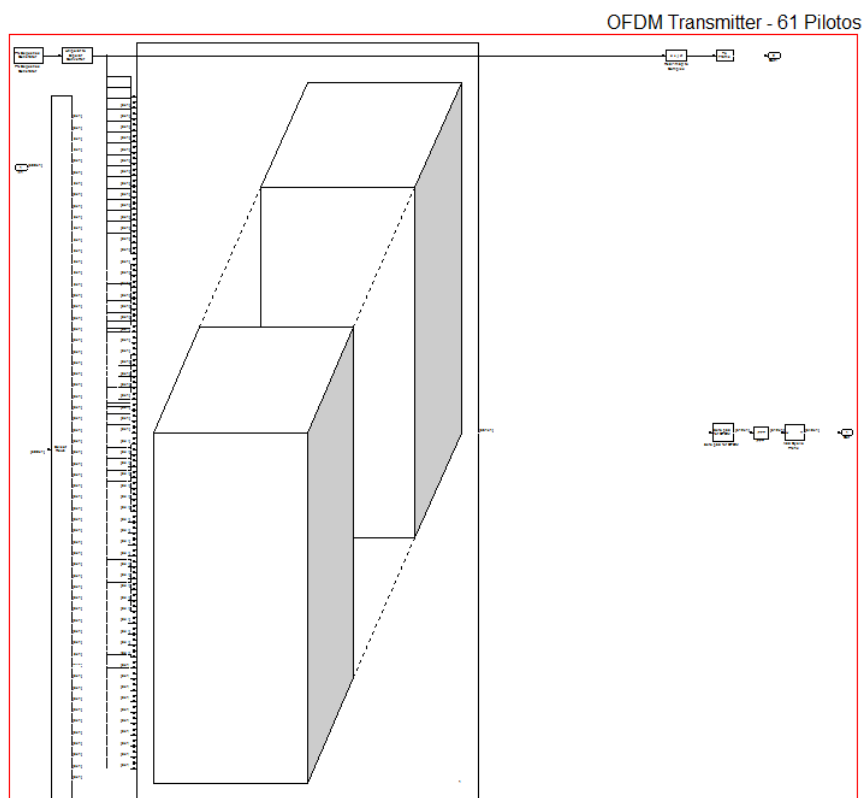


Figura B.25 - Transmissor OFDM - 61Pilotos.

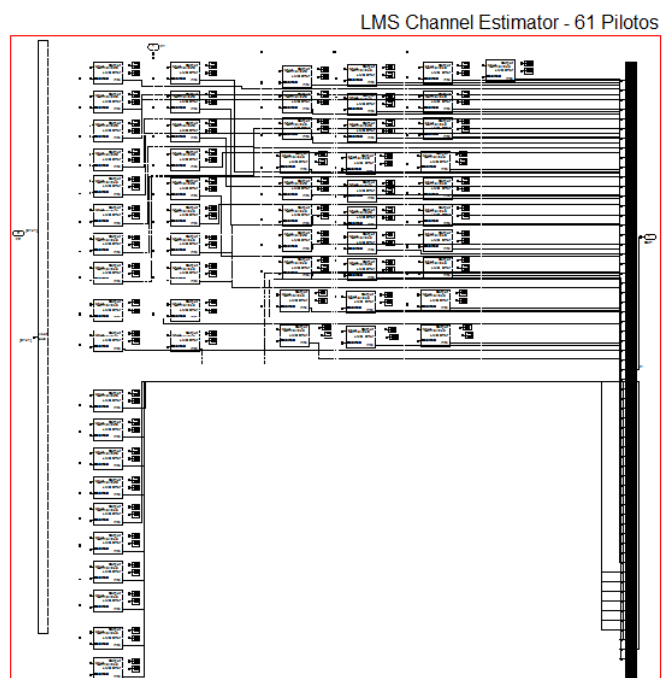


Figura B.26 - Estimador de Canal LMS - 61 Pilotos.

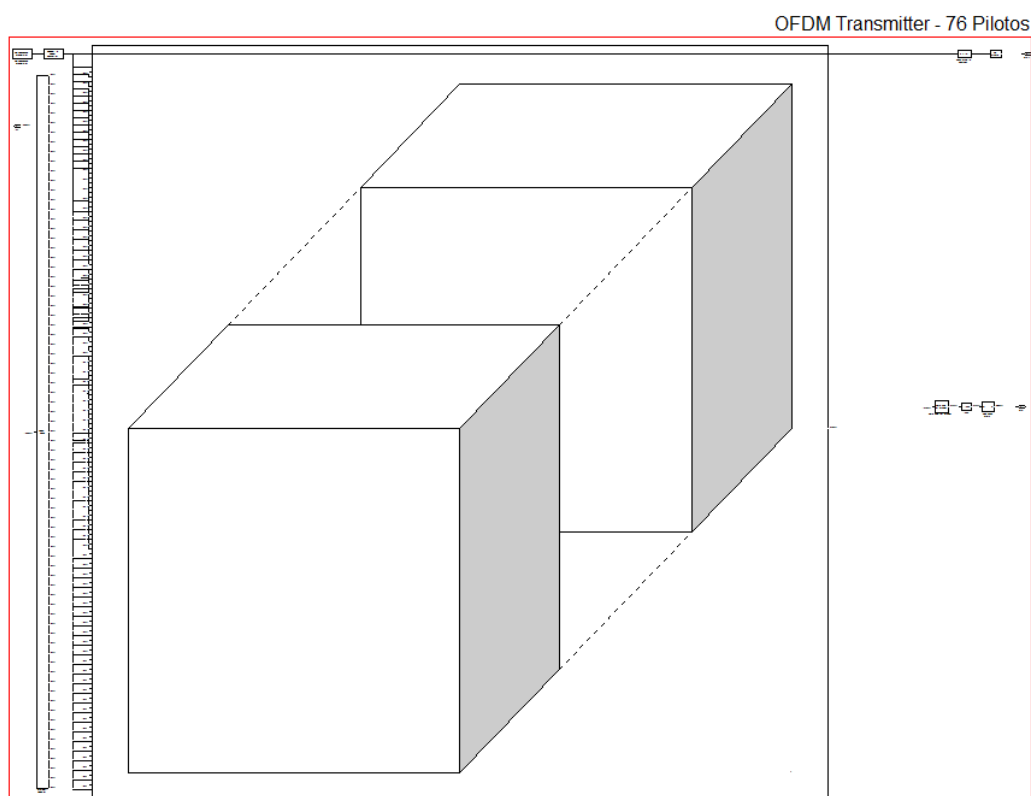


Figura B.27- Transmissor OFDM - 76 Pilotos.

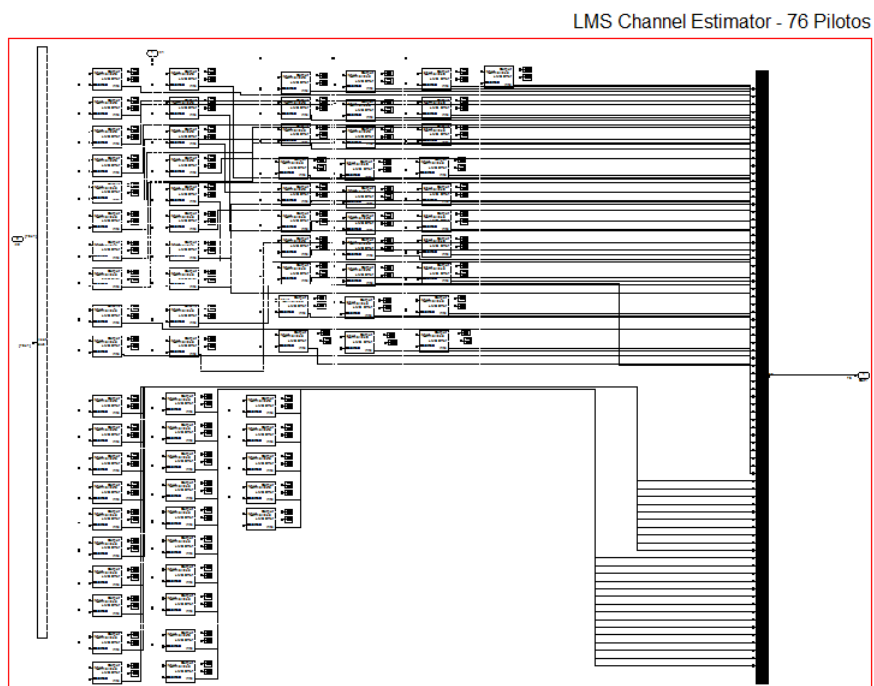


Figura B.28 - Estimador de Canal LMS - 76 Pilotos.

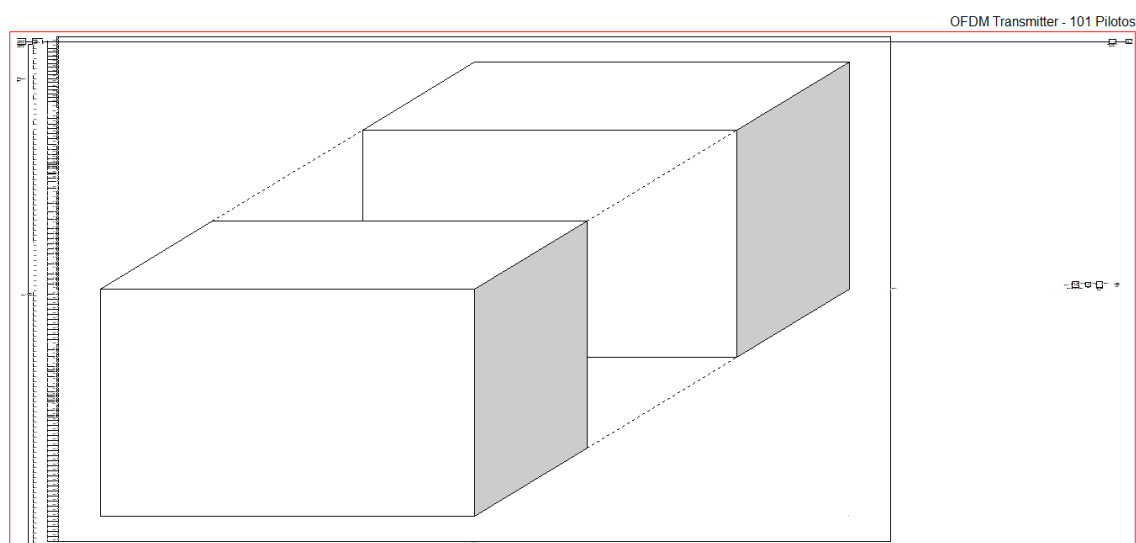


Figura B.29 - Transmissor OFDM - 101Pilotos.

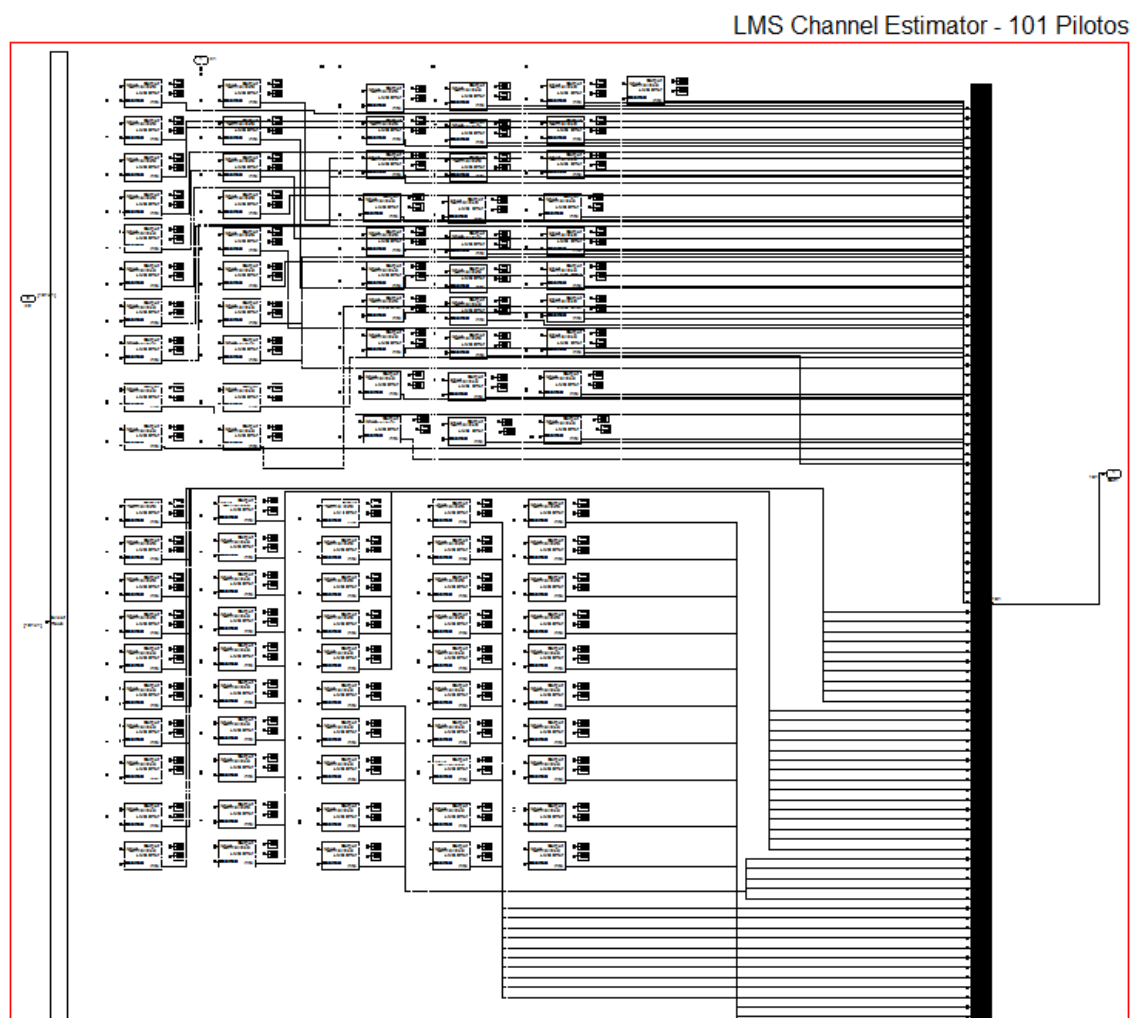


Figura B.30 - Estimador de Canal LMS - 101 Pilotos.

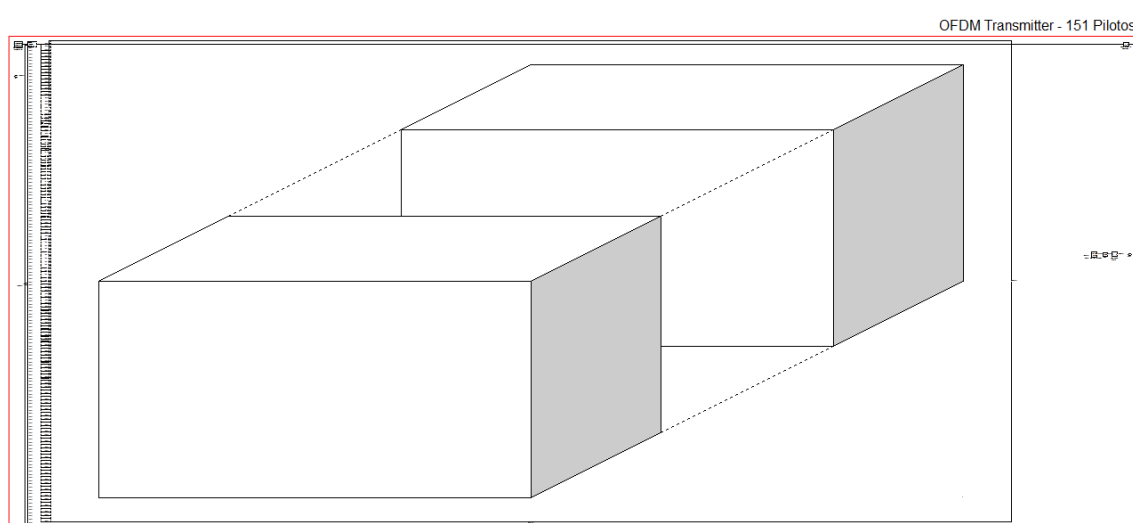


Figura B.31- Transmissor OFDM - 151Pilotos.

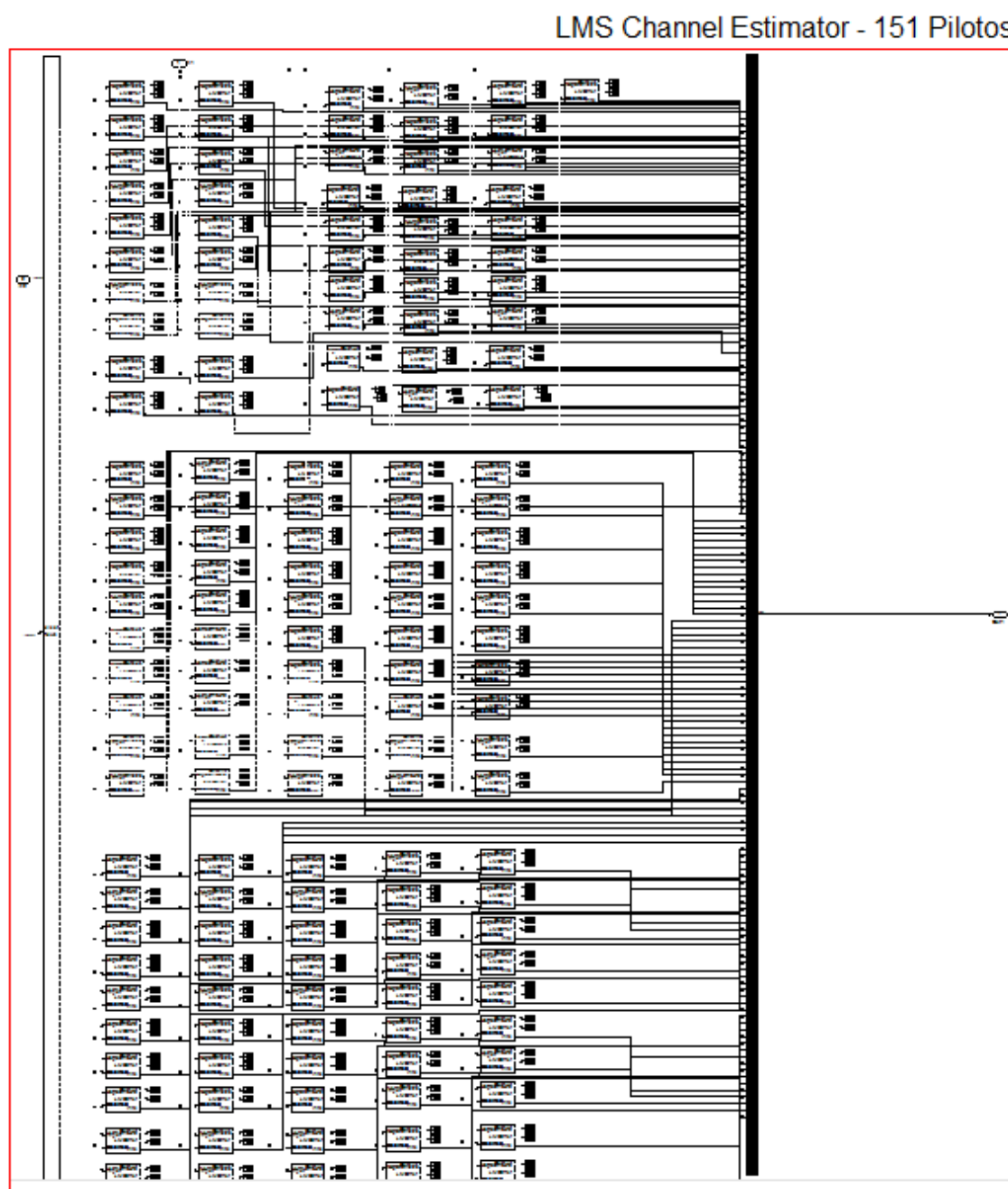


Figura B.32 - Estimador de Canal LMS - 151 Pilotos.

B.3. Sinal Recebido e Escopo do Espectro

OFDM P13

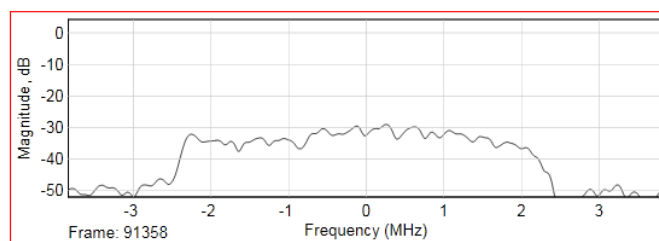
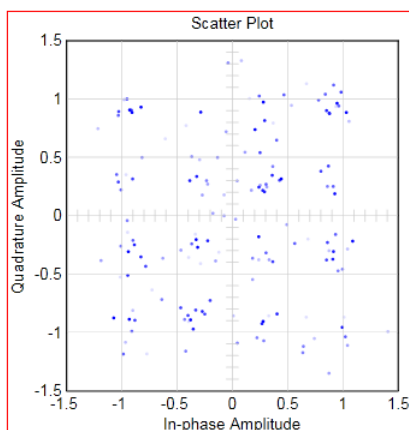


Figura B.33 - Sinal Recebido - 13 Pilotos.

Figura B.34 - Escopo do Espectro - 13 Pilotos.

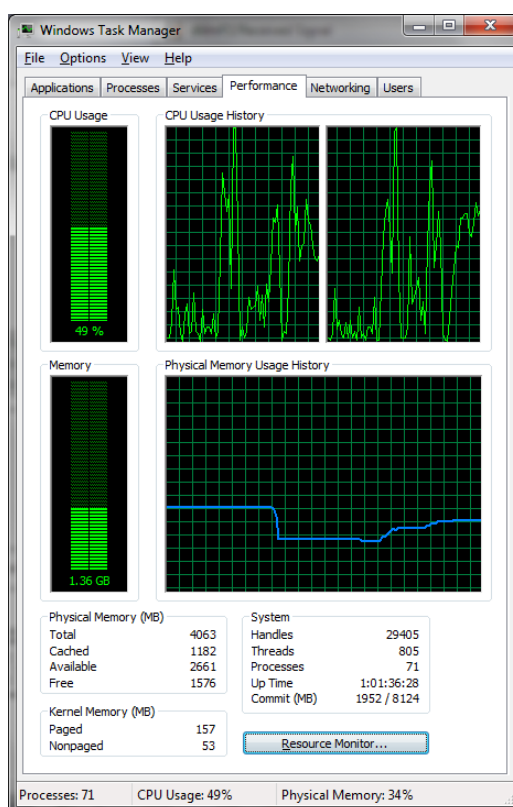


Figura B.35 - CPU e Memória Física.

B.4. Conclusão

Este Apêndice apresentou todos os Transmissores OFDM, os Estimadores de Canal LMS, e apresentou, também, o Sinal Recebido e o Escopo do Espectro, considerando 13 pilotos.

C. APÊNDICE - PROGRAMAS FEITOS PARA A PLATAFORMA OFDM

C.1. Introdução

Este apêndice apresenta todos os Programas feitos para a plataforma OFDM.

C.2. Programas

S_OFDMP2

```
function msfcn_S_OFDM(block)
% Level-2 M file S-Function for times two demo.
% Copyright 1990-2004 The MathWorks, Inc.
% $Revision: 1.1.6.1 $
    setup(block);
%endfunction
function setup(block)
    %% Register number of input and output ports
    block.NumInputPorts = 1;
    block.NumOutputPorts = 1;
    %% Setup functional port properties to dynamically
    %% inherited.
    block.SetPreCompInpPortInfoToDynamic;
    block.SetPreCompOutPortInfoToDynamic;
    % Override input port properties
    block.InputPort(1).DatatypeID = 0; % double
    block.InputPort(1).Complexity = 'complex';
    % Override output port properties
    block.OutputPort(1).DatatypeID = 0; % double
    block.OutputPort(1).Complexity = 'complex';
    block.InputPort(1).DirectFeedthrough = true;
    %% SetInputPortDimensions:
    %%     Functionality      : Check and set input and optionally output
    %%                         port dimensions
    %%     C-Mex counterpart: mdlSetInputPortDimensionInfo
    %%
    block.RegBlockMethod('SetInputPortDimensions', @SetInpPortDims);
    %%
    %% SetOutputPortDimensions:
    %%     Functionality      : Check and set output and optionally input
    %%                         port dimensions
    %%     C-Mex counterpart: mdlSetOutputPortDimensionInfo
```

```

%%
block.RegBlockMethod('SetOutputPortDimensions', @SetOutPortDims);
%% Set block sample time to inherited
block.SampleTimes = [-1 0];
%% Run accelerator on TLC
block.SetAccelRunOnTLC(true);
%% Register methods
block.RegBlockMethod('Outputs', @Output);
%endfunction
function SetInpPortDims(block, idx, di)
    block.InputPort(1).Dimensions = 2;
    block.OutputPort(1).Dimensions = 300;
%endfunction
function SetOutPortDims(block, idx, di)
    block.OutputPort(1).Dimensions = 300;
    block.InputPort(1).Dimensions = 2;
%endfunction
function Output(block)
x=[1 2]';
xi=[1+1/301 1+2/301 1+3/301 1+4/301 1+5/301 1+6/301 1+7/301 1+8/301 1+9/301
1+10/301 ...
1+11/301 1+12/301 1+13/301 1+14/301 1+15/301 1+16/301 1+17/301 1+18/301
1+19/301 1+20/301 ...
1+21/301 1+22/301 1+23/301 1+24/301 1+25/301 1+26/301 1+27/301 1+28/301
1+29/301 1+30/301 ...
1+31/301 1+32/301 1+33/301 1+34/301 1+35/301 1+36/301 1+37/301 1+38/301
1+39/301 1+40/301 ...
1+41/301 1+42/301 1+43/301 1+44/301 1+45/301 1+46/301 1+47/301 1+48/301
1+49/301 1+50/301 ...
1+51/301 1+52/301 1+53/301 1+54/301 1+55/301 1+56/301 1+57/301 1+58/301
1+59/301 1+60/301 ...
1+61/301 1+62/301 1+63/301 1+64/301 1+65/301 1+66/301 1+67/301 1+68/301
1+69/301 1+70/301 ...
1+71/301 1+72/301 1+73/301 1+74/301 1+75/301 1+76/301 1+77/301 1+78/301
1+79/301 1+80/301 ...
1+81/301 1+82/301 1+83/301 1+84/301 1+85/301 1+86/301 1+87/301 1+88/301
1+89/301 1+90/301 ...
1+91/301 1+92/301 1+93/301 1+94/301 1+95/301 1+96/301 1+97/301 1+98/301
1+99/301 1+100/301 ...
1+101/301 1+102/301 1+103/301 1+104/301 1+105/301 1+106/301 1+107/301
1+108/301 1+109/301 1+110/301 ...
1+111/301 1+112/301 1+113/301 1+114/301 1+115/301 1+116/301 1+117/301
1+118/301 1+119/301 1+120/301 ...
1+121/301 1+122/301 1+123/301 1+124/301 1+125/301 1+126/301 1+127/301
1+128/301 1+129/301 1+130/301 ...
1+131/301 1+132/301 1+133/301 1+134/301 1+135/301 1+136/301 1+137/301
1+138/301 1+139/301 1+140/301 ...
1+141/301 1+142/301 1+143/301 1+144/301 1+145/301 1+146/301 1+147/301
1+148/301 1+149/301 1+150/301 ...
1+151/301 1+152/301 1+153/301 1+154/301 1+155/301 1+156/301 1+157/301
1+158/301 1+159/301 1+160/301 ...
1+161/301 1+162/301 1+163/301 1+164/301 1+165/301 1+166/301 1+167/301
1+168/301 1+169/301 1+170/301 ...
1+171/301 1+172/301 1+173/301 1+174/301 1+175/301 1+176/301 1+177/301
1+178/301 1+179/301 1+180/301 ...
1+181/301 1+182/301 1+183/301 1+184/301 1+185/301 1+186/301 1+187/301
1+188/301 1+189/301 1+190/301 ...
1+191/301 1+192/301 1+193/301 1+194/301 1+195/301 1+196/301 1+197/301
1+198/301 1+199/301 1+200/301 ...
1+201/301 1+202/301 1+203/301 1+204/301 1+205/301 1+206/301 1+207/301
1+208/301 1+209/301 1+210/301 ...

```

```

1+211/301 1+212/301 1+213/301 1+214/301 1+215/301 1+216/301 1+217/301
1+218/301 1+219/301 1+220/301 ...
1+221/301 1+222/301 1+223/301 1+224/301 1+225/301 1+226/301 1+227/301
1+228/301 1+229/301 1+230/301 ...
1+231/301 1+232/301 1+233/301 1+234/301 1+235/301 1+236/301 1+237/301
1+238/301 1+239/301 1+240/301 ...
1+241/301 1+242/301 1+243/301 1+244/301 1+245/301 1+246/301 1+247/301
1+248/301 1+249/301 1+250/301 ...
1+251/301 1+252/301 1+253/301 1+254/301 1+255/301 1+256/301 1+257/301
1+258/301 1+259/301 1+260/301 ...
1+261/301 1+262/301 1+263/301 1+264/301 1+265/301 1+266/301 1+267/301
1+268/301 1+269/301 1+270/301 ...
1+271/301 1+272/301 1+273/301 1+274/301 1+275/301 1+276/301 1+277/301
1+278/301 1+279/301 1+280/301 ...
1+281/301 1+282/301 1+283/301 1+284/301 1+285/301 1+286/301 1+287/301
1+288/301 1+289/301 1+290/301 ...
1+291/301 1+292/301 1+293/301 1+294/301 1+295/301 1+296/301 1+297/301
1+298/301 1+299/301 1+300/301]';
block.OutputPort(1).Data = interp1(x,block.InputPort(1).Data,xi,'spline');
%endfunction

```

S_OFDMP3

```

function msfcn_S_OFDM(block)
% Level-2 M file S-Function for times two demo.
% Copyright 1990-2004 The MathWorks, Inc.
% $Revision: 1.1.6.1 $
setup(block);
%endfunction
function setup(block)
%% Register number of input and output ports
block.NumInputPorts = 1;
block.NumOutputPorts = 1;
%% Setup functional port properties to dynamically
%% inherited.
block.SetPreCompInPortInfoToDynamic;
block.SetPreCompOutPortInfoToDynamic;
% Override input port properties
block.InputPort(1).DatatypeID = 0; % double
block.InputPort(1).Complexity = 'complex';
% Override output port properties
block.OutputPort(1).DatatypeID = 0; % double
block.OutputPort(1).Complexity = 'complex';
block.InputPort(1).DirectFeedthrough = true;
%% SetInputPortDimensions:
%% Functionality : Check and set input and optionally output
%% port dimensions
%% C-Mex counterpart: mdlSetInputPortDimensionInfo
%%
block.RegBlockMethod('SetInputPortDimensions', @SetInPortDims);
%%
%% SetOutputPortDimensions:
%% Functionality : Check and set output and optionally input
%% port dimensions
%% C-Mex counterpart: mdlSetOutputPortDimensionInfo
%%
block.RegBlockMethod('SetOutputPortDimensions', @SetOutPortDims);
%% Set block sample time to inherited
block.SampleTimes = [-1 0];
%% Run accelerator on TLC
block.SetAccelRunOnTLC(true);

```



```

%% Register methods
block.RegBlockMethod('Outputs',@Output);
%endfunction
function SetInpPortDims(block, idx, di)
    block.InputPort(1).Dimensions = 3;
    block.OutputPort(1).Dimensions = 300;
%endfunction
function SetOutPortDims(block, idx, di)
    block.OutputPort(1).Dimensions = 300;
    block.InputPort(1).Dimensions = 3;
%endfunction
function Output(block)
x=[1 2 3]';
xi=[1+1/151 1+2/151 1+3/151 1+4/151 1+5/151 1+6/151 1+7/151 1+8/151 1+9/151
1+10/151 ...
1+11/151 1+12/151 1+13/151 1+14/151 1+15/151 1+16/151 1+17/151 1+18/151
1+19/151 1+20/151 ...
1+21/151 1+22/151 1+23/151 1+24/151 1+25/151 1+26/151 1+27/151 1+28/151
1+29/151 1+30/151 ...
1+31/151 1+32/151 1+33/151 1+34/151 1+35/151 1+36/151 1+37/151 1+38/151
1+39/151 1+40/151 ...
1+41/151 1+42/151 1+43/151 1+44/151 1+45/151 1+46/151 1+47/151 1+48/151
1+49/151 1+50/151 ...
1+51/151 1+52/151 1+53/151 1+54/151 1+55/151 1+56/151 1+57/151 1+58/151
1+59/151 1+60/151 ...
1+61/151 1+62/151 1+63/151 1+64/151 1+65/151 1+66/151 1+67/151 1+68/151
1+69/151 1+70/151 ...
1+71/151 1+72/151 1+73/151 1+74/151 1+75/151 1+76/151 1+77/151 1+78/151
1+79/151 1+80/151 ...
1+81/151 1+82/151 1+83/151 1+84/151 1+85/151 1+86/151 1+87/151 1+88/151
1+89/151 1+90/151 ...
1+91/151 1+92/151 1+93/151 1+94/151 1+95/151 1+96/151 1+97/151 1+98/151
1+99/151 1+100/151 ...
1+101/151 1+102/151 1+103/151 1+104/151 1+105/151 1+106/151 1+107/151
1+108/151 1+109/151 1+110/151 ...
1+111/151 1+112/151 1+113/151 1+114/151 1+115/151 1+116/151 1+117/151
1+118/151 1+119/151 1+120/151 ...
1+121/151 1+122/151 1+123/151 1+124/151 1+125/151 1+126/151 1+127/151
1+128/151 1+129/151 1+130/151 ...
1+131/151 1+132/151 1+133/151 1+134/151 1+135/151 1+136/151 1+137/151
1+138/151 1+139/151 1+140/151 ...
1+141/151 1+142/151 1+143/151 1+144/151 1+145/151 1+146/151 1+147/151
1+148/151 1+149/151 1+150/151 ...
2+1/151 2+2/151 2+3/151 2+4/151 2+5/151 2+6/151 2+7/151 2+8/151 2+9/151
2+10/151 ...
2+11/151 2+12/151 2+13/151 2+14/151 2+15/151 2+16/151 2+17/151 2+18/151
2+19/151 2+20/151 ...
2+21/151 2+22/151 2+23/151 2+24/151 2+25/151 2+26/151 2+27/151 2+28/151
2+29/151 2+30/151 ...
2+31/151 2+32/151 2+33/151 2+34/151 2+35/151 2+36/151 2+37/151 2+38/151
2+39/151 2+40/151 ...
2+41/151 2+42/151 2+43/151 2+44/151 2+45/151 2+46/151 2+47/151 2+48/151
2+49/151 2+50/151 ...
2+51/151 2+52/151 2+53/151 2+54/151 2+55/151 2+56/151 2+57/151 2+58/151
2+59/151 2+60/151 ...
2+61/151 2+62/151 2+63/151 2+64/151 2+65/151 2+66/151 2+67/151 2+68/151
2+69/151 2+70/151 ...
2+71/151 2+72/151 2+73/151 2+74/151 2+75/151 2+76/151 2+77/151 2+78/151
2+79/151 2+80/151 ...
2+81/151 2+82/151 2+83/151 2+84/151 2+85/151 2+86/151 2+87/151 2+88/151
2+89/151 2+90/151 ...

```

```

2+91/151 2+92/151 2+93/151 2+94/151 2+95/151 2+96/151 2+97/151 2+98/151
2+99/151 2+100/151 ...
2+101/151 2+102/151 2+103/151 2+104/151 2+105/151 2+106/151 2+107/151
2+108/151 2+109/151 2+110/151 ...
2+111/151 2+112/151 2+113/151 2+114/151 2+115/151 2+116/151 2+117/151
2+118/151 2+119/151 2+120/151 ...
2+121/151 2+122/151 2+123/151 2+124/151 2+125/151 2+126/151 2+127/151
2+128/151 2+129/151 2+130/151 ...
2+131/151 2+132/151 2+133/151 2+134/151 2+135/151 2+136/151 2+137/151
2+138/151 2+139/151 2+140/151 ...
2+141/151 2+142/151 2+143/151 2+144/151 2+145/151 2+146/151 2+147/151
2+148/151 2+149/151 2+150/151]';
block.OutputPort(1).Data = interp1(x,block.InputPort(1).Data,xi,'spline');
%endfunction

```

S_OFDMP4

```

function msfcn_S_OFDM(block)
% Level-2 M file S-Function for times two demo.
% Copyright 1990-2004 The MathWorks, Inc.
% $Revision: 1.1.6.1 $
setup(block);
%endfunction
function setup(block)
%% Register number of input and output ports
block.NumInputPorts = 1;
block.NumOutputPorts = 1;
%% Setup functional port properties to dynamically
%% inherited.
block.SetPreCompInPortInfoToDynamic;
block.SetPreCompOutPortInfoToDynamic;
% Override input port properties
block.InputPort(1).DatatypeID = 0; % double
block.InputPort(1).Complexity = 'complex';
% Override output port properties
block.OutputPort(1).DatatypeID = 0; % double
block.OutputPort(1).Complexity = 'complex';
block.InputPort(1).DirectFeedthrough = true;
%% SetInputPortDimensions:
%% Functionality : Check and set input and optionally output
%% port dimensions
%% C-Mex counterpart: mdlSetInputPortDimensionInfo
%%
block.RegBlockMethod('SetInputPortDimensions', @SetInpPortDims);
%%
%% SetOutputPortDimensions:
%% Functionality : Check and set output and optionally input
%% port dimensions
%% C-Mex counterpart: mdlSetOutputPortDimensionInfo
%%
block.RegBlockMethod('SetOutputPortDimensions', @SetOutPortDims);
%% Set block sample time to inherited
block.SampleTimes = [-1 0];
%% Run accelerator on TLC
block.SetAccelRunOnTLC(true);
%% Register methods
block.RegBlockMethod('Outputs', @Output);
%endfunction
function SetInpPortDims(block, idx, di)
block.InputPort(1).Dimensions = 4;
block.OutputPort(1).Dimensions = 300;

```

```

%endfunction
function SetOutPortDims(block, idx, di)
    block.OutputPort(1).Dimensions = 300;
    block.InputPort(1).Dimensions = 4;
%endfunction
function Output(block)
x=[1 2 3 4]';
xi=[1+1/101 1+2/101 1+3/101 1+4/101 1+5/101 1+6/101 1+7/101 1+8/101 1+9/101
1+10/101 ...
1+11/101 1+12/101 1+13/101 1+14/101 1+15/101 1+16/101 1+17/101 1+18/101
1+19/101 1+20/101 ...
1+21/101 1+22/101 1+23/101 1+24/101 1+25/101 1+26/101 1+27/101 1+28/101
1+29/101 1+30/101 ...
1+31/101 1+32/101 1+33/101 1+34/101 1+35/101 1+36/101 1+37/101 1+38/101
1+39/101 1+40/101 ...
1+41/101 1+42/101 1+43/101 1+44/101 1+45/101 1+46/101 1+47/101 1+48/101
1+49/101 1+50/101 ...
1+51/101 1+52/101 1+53/101 1+54/101 1+55/101 1+56/101 1+57/101 1+58/101
1+59/101 1+60/101 ...
1+61/101 1+62/101 1+63/101 1+64/101 1+65/101 1+66/101 1+67/101 1+68/101
1+69/101 1+70/101 ...
1+71/101 1+72/101 1+73/101 1+74/101 1+75/101 1+76/101 1+77/101 1+78/101
1+79/101 1+80/101 ...
1+81/101 1+82/101 1+83/101 1+84/101 1+85/101 1+86/101 1+87/101 1+88/101
1+89/101 1+90/101 ...
1+91/101 1+92/101 1+93/101 1+94/101 1+95/101 1+96/101 1+97/101 1+98/101
1+99/101 1+100/101 ...
2+1/101 2+2/101 2+3/101 2+4/101 2+5/101 2+6/101 2+7/101 2+8/101 2+9/101
2+10/101 ...
2+11/101 2+12/101 2+13/101 2+14/101 2+15/101 2+16/101 2+17/101 2+18/101
2+19/101 2+20/101 ...
2+21/101 2+22/101 2+23/101 2+24/101 2+25/101 2+26/101 2+27/101 2+28/101
2+29/101 2+30/101 ...
2+31/101 2+32/101 2+33/101 2+34/101 2+35/101 2+36/101 2+37/101 2+38/101
2+39/101 2+40/101 ...
2+41/101 2+42/101 2+43/101 2+44/101 2+45/101 2+46/101 2+47/101 2+48/101
2+49/101 2+50/101 ...
2+51/101 2+52/101 2+53/101 2+54/101 2+55/101 2+56/101 2+57/101 2+58/101
2+59/101 2+60/101 ...
2+61/101 2+62/101 2+63/101 2+64/101 2+65/101 2+66/101 2+67/101 2+68/101
2+69/101 2+70/101 ...
2+71/101 2+72/101 2+73/101 2+74/101 2+75/101 2+76/101 2+77/101 2+78/101
2+79/101 2+80/101 ...
2+81/101 2+82/101 2+83/101 2+84/101 2+85/101 2+86/101 2+87/101 2+88/101
2+89/101 2+90/101 ...
2+91/101 2+92/101 2+93/101 2+94/101 2+95/101 2+96/101 2+97/101 2+98/101
2+99/101 2+100/101 ...
3+1/101 3+2/101 3+3/101 3+4/101 3+5/101 3+6/101 3+7/101 3+8/101 3+9/101
3+10/101 ...
3+11/101 3+12/101 3+13/101 3+14/101 3+15/101 3+16/101 3+17/101 3+18/101
3+19/101 3+20/101 ...
3+21/101 3+22/101 3+23/101 3+24/101 3+25/101 3+26/101 3+27/101 3+28/101
3+29/101 3+30/101 ...
3+31/101 3+32/101 3+33/101 3+34/101 3+35/101 3+36/101 3+37/101 3+38/101
3+39/101 3+40/101 ...
3+41/101 3+42/101 3+43/101 3+44/101 3+45/101 3+46/101 3+47/101 3+48/101
3+49/101 3+50/101 ...
3+51/101 3+52/101 3+53/101 3+54/101 3+55/101 3+56/101 3+57/101 3+58/101
3+59/101 3+60/101 ...
3+61/101 3+62/101 3+63/101 3+64/101 3+65/101 3+66/101 3+67/101 3+68/101
3+69/101 3+70/101 ...

```

```

3+71/101 3+72/101 3+73/101 3+74/101 3+75/101 3+76/101 3+77/101 3+78/101
3+79/101 3+80/101 ...
3+81/101 3+82/101 3+83/101 3+84/101 3+85/101 3+86/101 3+87/101 3+88/101
3+89/101 3+90/101 ...
3+91/101 3+92/101 3+93/101 3+94/101 3+95/101 3+96/101 3+97/101 3+98/101
3+99/101 3+100/101]';
block.OutputPort(1).Data = interp1(x,block.InputPort(1).Data,xi,'spline');
%endfunction

```

S_OFDMP5

```

function msfcn_S_OFDM(block)
% Level-2 M file S-Function for times two demo.
% Copyright 1990-2004 The MathWorks, Inc.
% $Revision: 1.1.6.1 $
setup(block);
%endfunction
function setup(block)
%% Register number of input and output ports
block.NumInputPorts = 1;
block.NumOutputPorts = 1;
%% Setup functional port properties to dynamically
%% inherited.
block.SetPreCompInPortInfoToDynamic;
block.SetPreCompOutPortInfoToDynamic;
% Override input port properties
block.InputPort(1).DatatypeID = 0; % double
block.InputPort(1).Complexity = 'complex';
% Override output port properties
block.OutputPort(1).DatatypeID = 0; % double
block.OutputPort(1).Complexity = 'complex';
block.InputPort(1).DirectFeedthrough = true;
%% SetInputPortDimensions:
%% Functionality : Check and set input and optionally output
%% port dimensions
%% C-Mex counterpart: mdlSetInputPortDimensionInfo
%%
block.RegBlockMethod('SetInputPortDimensions', @SetInpPortDims);
%%
%% SetOutputPortDimensions:
%% Functionality : Check and set output and optionally input
%% port dimensions
%% C-Mex counterpart: mdlSetOutputPortDimensionInfo
%%
block.RegBlockMethod('SetOutputPortDimensions', @SetOutPortDims);
%% Set block sample time to inherited
block.SampleTimes = [-1 0];
%% Run accelerator on TLC
block.SetAccelRunOnTLC(true);
%% Register methods
block.RegBlockMethod('Outputs',@Output);
%endfunction
function SetInpPortDims(block, idx, di)
block.InputPort(1).Dimensions = 5;
block.OutputPort(1).Dimensions = 300;
%endfunction
function SetOutPortDims(block, idx, di)
block.OutputPort(1).Dimensions = 300;
block.InputPort(1).Dimensions = 5;
%endfunction
function Output(block)

```

```

x=[1 2 3 4 5]';
xi=[1+1/76 1+2/76 1+3/76 1+4/76 1+5/76 1+6/76 1+7/76 1+8/76 1+9/76 1+10/76
...
1+11/76 1+12/76 1+13/76 1+14/76 1+15/76 1+16/76 1+17/76 1+18/76 1+19/76
1+20/76 ...
1+21/76 1+22/76 1+23/76 1+24/76 1+25/76 1+26/76 1+27/76 1+28/76 1+29/76
1+30/76 ...
1+31/76 1+32/76 1+33/76 1+34/76 1+35/76 1+36/76 1+37/76 1+38/76 1+39/76
1+40/76 ...
1+41/76 1+42/76 1+43/76 1+44/76 1+45/76 1+46/76 1+47/76 1+48/76 1+49/76
1+50/76 ...
1+51/76 1+52/76 1+53/76 1+54/76 1+55/76 1+56/76 1+57/76 1+58/76 1+59/76
1+60/76 ...
1+61/76 1+62/76 1+63/76 1+64/76 1+65/76 1+66/76 1+67/76 1+68/76 1+69/76
1+70/76 ...
1+71/76 1+72/76 1+73/76 1+74/76 1+75/76 ...
2+1/76 2+2/76 2+3/76 2+4/76 2+5/76 2+6/76 2+7/76 2+8/76 2+9/76 2+10/76 ...
2+11/76 2+12/76 2+13/76 2+14/76 2+15/76 2+16/76 2+17/76 2+18/76 2+19/76
2+20/76 ...
2+21/76 2+22/76 2+23/76 2+24/76 2+25/76 2+26/76 2+27/76 2+28/76 2+29/76
2+30/76 ...
2+31/76 2+32/76 2+33/76 2+34/76 2+35/76 2+36/76 2+37/76 2+38/76 2+39/76
2+40/76 ...
2+41/76 2+42/76 2+43/76 2+44/76 2+45/76 2+46/76 2+47/76 2+48/76 2+49/76
2+50/76 ...
2+51/76 2+52/76 2+53/76 2+54/76 2+55/76 2+56/76 2+57/76 2+58/76 2+59/76
2+60/76 ...
2+61/76 2+62/76 2+63/76 2+64/76 2+65/76 2+66/76 2+67/76 2+68/76 2+69/76
2+70/76 ...
2+71/76 2+72/76 2+73/76 2+74/76 2+75/76 ...
3+1/76 3+2/76 3+3/76 3+4/76 3+5/76 3+6/76 3+7/76 3+8/76 3+9/76 3+10/76 ...
3+11/76 3+12/76 3+13/76 3+14/76 3+15/76 3+16/76 3+17/76 3+18/76 3+19/76
3+20/76 ...
3+21/76 3+22/76 3+23/76 3+24/76 3+25/76 3+26/76 3+27/76 3+28/76 3+29/76
3+30/76 ...
3+31/76 3+32/76 3+33/76 3+34/76 3+35/76 3+36/76 3+37/76 3+38/76 3+39/76
3+40/76 ...
3+41/76 3+42/76 3+43/76 3+44/76 3+45/76 3+46/76 3+47/76 3+48/76 3+49/76
3+50/76 .....
3+51/76 3+52/76 3+53/76 3+54/76 3+55/76 3+56/76 3+57/76 3+58/76 3+59/76
3+60/76 ...
3+61/76 3+62/76 3+63/76 3+64/76 3+65/76 3+66/76 3+67/76 3+68/76 3+69/76
3+70/76 ...
3+71/76 3+72/76 3+73/76 3+74/76 3+75/76 ...
4+1/76 4+2/76 4+3/76 4+4/76 4+5/76 4+6/76 4+7/76 4+8/76 4+9/76 4+10/76 ...
4+11/76 4+12/76 4+13/76 4+14/76 4+15/76 4+16/76 4+17/76 4+18/76 4+19/76
4+20/76 ...
4+21/76 4+22/76 4+23/76 4+24/76 4+25/76 4+26/76 4+27/76 4+28/76 4+29/76
4+30/76 ...
4+31/76 4+32/76 4+33/76 4+34/76 4+35/76 4+36/76 4+37/76 4+38/76 4+39/76
4+40/76 ...
4+41/76 4+42/76 4+43/76 4+44/76 4+45/76 4+46/76 4+47/76 4+48/76 4+49/76
4+50/76 ...
4+51/76 4+52/76 4+53/76 4+54/76 4+55/76 4+56/76 4+57/76 4+58/76 4+59/76
4+60/76 ...
4+61/76 4+62/76 4+63/76 4+64/76 4+65/76 4+66/76 4+67/76 4+68/76 4+69/76
4+70/76 ...
4+71/76 4+72/76 4+73/76 4+74/76 4+75/76]';
block.OutputPort(1).Data = interp1(x,block.InputPort(1).Data,xi,'spline');
%endfunction

```

S_OFDMP6

```

function msfcn_S_OFDM(block)
% Level-2 M file S-Function for times two demo.
% Copyright 1990-2004 The MathWorks, Inc.
% $Revision: 1.1.6.1 $
setup(block);
%endfunction
function setup(block)
    %% Register number of input and output ports
    block.NumInputPorts = 1;
    block.NumOutputPorts = 1;
    %% Setup functional port properties to dynamically
    %% inherited.
    block.SetPreCompInpPortInfoToDynamic;
    block.SetPreCompOutPortInfoToDynamic;
    % Override input port properties
    block.InputPort(1).DatatypeID = 0; % double
    block.InputPort(1).Complexity = 'complex';
    % Override output port properties
    block.OutputPort(1).DatatypeID = 0; % double
    block.OutputPort(1).Complexity = 'complex';
    block.InputPort(1).DirectFeedthrough = true;
    %% SetInputPortDimensions:
    %% Functionality : Check and set input and optionally output
    %% port dimensions
    %% C-Mex counterpart: mdlSetInputPortDimensionInfo
    %%
    block.RegBlockMethod('SetInputPortDimensions', @SetInpPortDims);
    %%
    %% SetOutputPortDimensions:
    %% Functionality : Check and set output and optionally input
    %% port dimensions
    %% C-Mex counterpart: mdlSetOutputPortDimensionInfo
    %%
    block.RegBlockMethod('SetOutputPortDimensions', @SetOutPortDims);
    %% Set block sample time to inherited
    block.SampleTimes = [-1 0];
    %% Run accelerator on TLC
    block.SetAccelRunOnTLC(true);
    %% Register methods
    block.RegBlockMethod('Outputs', @Output);
%endfunction
function SetInpPortDims(block, idx, di)
    block.InputPort(1).Dimensions = 6;
    block.OutputPort(1).Dimensions = 300;
%endfunction
function SetOutPortDims(block, idx, di)
    block.OutputPort(1).Dimensions = 300;
    block.InputPort(1).Dimensions = 6;
%endfunction
function Output(block)
x=[1 2 3 4 5 6]';
xi=[1+1/61 1+2/61 1+3/61 1+4/61 1+5/61 1+6/61 1+7/61 1+8/61 1+9/61 1+10/61
...
1+11/61 1+12/61 1+13/61 1+14/61 1+15/61 1+16/61 1+17/61 1+18/61 1+19/61
1+20/61 ...
1+21/61 1+22/61 1+23/61 1+24/61 1+25/61 1+26/61 1+27/61 1+28/61 1+29/61
1+30/61 ...
1+31/61 1+32/61 1+33/61 1+34/61 1+35/61 1+36/61 1+37/61 1+38/61 1+39/61
1+40/61 ...

```

```

1+41/61 1+42/61 1+43/61 1+44/61 1+45/61 1+46/61 1+47/61 1+48/61 1+49/61
1+50/61 ...
1+51/61 1+52/61 1+53/61 1+54/61 1+55/61 1+56/61 1+57/61 1+58/61 1+59/61
1+60/61 ...
2+1/61 2+2/61 2+3/61 2+4/61 2+5/61 2+6/61 2+7/61 2+8/61 2+9/61 2+10/61 ...
2+11/61 2+12/61 2+13/61 2+14/61 2+15/61 2+16/61 2+17/61 2+18/61 2+19/61
2+20/61 ...
2+21/61 2+22/61 2+23/61 2+24/61 2+25/61 2+26/61 2+27/61 2+28/61 2+29/61
2+30/61 ...
2+31/61 2+32/61 2+33/61 2+34/61 2+35/61 2+36/61 2+37/61 2+38/61 2+39/61
2+40/61 ...
2+41/61 2+42/61 2+43/61 2+44/61 2+45/61 2+46/61 2+47/61 2+48/61 2+49/61
2+50/61 ...
2+51/61 2+52/61 2+53/61 2+54/61 2+55/61 2+56/61 2+57/61 2+58/61 2+59/61
2+60/61 ...
3+1/61 3+2/61 3+3/61 3+4/61 3+5/61 3+6/61 3+7/61 3+8/61 3+9/61 3+10/61 ...
3+11/61 3+12/61 3+13/61 3+14/61 3+15/61 3+16/61 3+17/61 3+18/61 3+19/61
3+20/61 ...
3+21/61 3+22/61 3+23/61 3+24/61 3+25/61 3+26/61 3+27/61 3+28/61 3+29/61
3+30/61 ...
3+31/61 3+32/61 3+33/61 3+34/61 3+35/61 3+36/61 3+37/61 3+38/61 3+39/61
3+40/61 ...
3+41/61 3+42/61 3+43/61 3+44/61 3+45/61 3+46/61 3+47/61 3+48/61 3+49/61
3+50/61 ...
3+51/61 3+52/61 3+53/61 3+54/61 3+55/61 3+56/61 3+57/61 3+58/61 3+59/61
3+60/61 ...
4+1/61 4+2/61 4+3/61 4+4/61 4+5/61 4+6/61 4+7/61 4+8/61 4+9/61 4+10/61 ...
4+11/61 4+12/61 4+13/61 4+14/61 4+15/61 4+16/61 4+17/61 4+18/61 4+19/61
4+20/61 ...
4+21/61 4+22/61 4+23/61 4+24/61 4+25/61 4+26/61 4+27/61 4+28/61 4+29/61
4+30/61 ...
4+31/61 4+32/61 4+33/61 4+34/61 4+35/61 4+36/61 4+37/61 4+38/61 4+39/61
4+40/61 ...
4+41/61 4+42/61 4+43/61 4+44/61 4+45/61 4+46/61 4+47/61 4+48/61 4+49/61
4+50/61 ...
4+51/61 4+52/61 4+53/61 4+54/61 4+55/61 4+56/61 4+57/61 4+58/61 4+59/61
4+60/61 ...
5+1/61 5+2/61 5+3/61 5+4/61 5+5/61 5+6/61 5+7/61 5+8/61 5+9/61 5+10/61 ...
5+11/61 5+12/61 5+13/61 5+14/61 5+15/61 5+16/61 5+17/61 5+18/61 5+19/61
5+20/61 ...
5+21/61 5+22/61 5+23/61 5+24/61 5+25/61 5+26/61 5+27/61 5+28/61 5+29/61
5+30/61 ...
5+31/61 5+32/61 5+33/61 5+34/61 5+35/61 5+36/61 5+37/61 5+38/61 5+39/61
5+40/61 ...
5+41/61 5+42/61 5+43/61 5+44/61 5+45/61 5+46/61 5+47/61 5+48/61 5+49/61
5+50/61 ...
5+51/61 5+52/61 5+53/61 5+54/61 5+55/61 5+56/61 5+57/61 5+58/61 5+59/61
5+60/61]';
block.OutputPort(1).Data = interp1(x,block.InputPort(1).Data,xi,'spline');
%endfunction

```

S_OFDMP7

```

function msfcn_S_OFDM(block)
% Level-2 M file S-Function for times two demo.
% Copyright 1990-2004 The MathWorks, Inc.
% $Revision: 1.1.6.1 $
setup(block);
%endfunction
function setup(block)
%% Register number of input and output ports
block.NumInputPorts = 1;

```

```

block.NumOutputPorts = 1;
%% Setup functional port properties to dynamically
%% inherited.
block.SetPreCompInpPortInfoToDynamic;
block.SetPreCompOutPortInfoToDynamic;
% Override input port properties
block.InputPort(1).DatatypeID = 0; % double
block.InputPort(1).Complexity = 'complex';
% Override output port properties
block.OutputPort(1).DatatypeID = 0; % double
block.OutputPort(1).Complexity = 'complex';
block.InputPort(1).DirectFeedthrough = true;
%% SetInputPortDimensions:
%%   Functionality      : Check and set input and optionally output
%%                       port dimensions
%%   C-Mex counterpart: mdlSetInputPortDimensionInfo
%%
block.RegBlockMethod('SetInputPortDimensions', @SetInpPortDims);
%%
%% SetOutputPortDimensions:
%%   Functionality      : Check and set output and optionally input
%%                       port dimensions
%%   C-Mex counterpart: mdlSetOutputPortDimensionInfo
%%
block.RegBlockMethod('SetOutputPortDimensions', @SetOutPortDims);
%% Set block sample time to inherited
block.SampleTimes = [-1 0];
%% Run accelerator on TLC
block.SetAccelRunOnTLC(true);
%% Register methods
block.RegBlockMethod('Outputs', @Output);
%endfunction
function SetInpPortDims(block, idx, di)
    block.InputPort(1).Dimensions = 7;
    block.OutputPort(1).Dimensions = 300;
%endfunction
function SetOutPortDims(block, idx, di)
    block.OutputPort(1).Dimensions = 300;
    block.InputPort(1).Dimensions = 7;
%endfunction
function Output(block)
x=[1 2 3 4 5 6 7]';
xi=[1+1/51 1+2/51 1+3/51 1+4/51 1+5/51 1+6/51 1+7/51 1+8/51 1+9/51 1+10/51
...
1+11/51 1+12/51 1+13/51 1+14/51 1+15/51 1+16/51 1+17/51 1+18/51 1+19/51
1+20/51 ...
1+21/51 1+22/51 1+23/51 1+24/51 1+25/51 1+26/51 1+27/51 1+28/51 1+29/51
1+30/51 ...
1+31/51 1+32/51 1+33/51 1+34/51 1+35/51 1+36/51 1+37/51 1+38/51 1+39/51
1+40/51 ...
1+41/51 1+42/51 1+43/51 1+44/51 1+45/51 1+46/51 1+47/51 1+48/51 1+49/51
1+50/51 ...
2+1/51 2+2/51 2+3/51 2+4/51 2+5/51 2+6/51 2+7/51 2+8/51 2+9/51 2+10/51 ...
2+11/51 2+12/51 2+13/51 2+14/51 2+15/51 2+16/51 2+17/51 2+18/51 2+19/51
2+20/51 ...
2+21/51 2+22/51 2+23/51 2+24/51 2+25/51 2+26/51 2+27/51 2+28/51 2+29/51
2+30/51 ...
2+31/51 2+32/51 2+33/51 2+34/51 2+35/51 2+36/51 2+37/51 2+38/51 2+39/51
2+40/51 ...
2+41/51 2+42/51 2+43/51 2+44/51 2+45/51 2+46/51 2+47/51 2+48/51 2+49/51
2+50/51 ...

```



```

3+1/51 3+2/51 3+3/51 3+4/51 3+5/51 3+6/51 3+7/51 3+8/51 3+9/51 3+10/51 ...
3+11/51 3+12/51 3+13/51 3+14/51 3+15/51 3+16/51 3+17/51 3+18/51 3+19/51
3+20/51 ...
3+21/51 3+22/51 3+23/51 3+24/51 3+25/51 3+26/51 3+27/51 3+28/51 3+29/51
3+30/51 ...
3+31/51 3+32/51 3+33/51 3+34/51 3+35/51 3+36/51 3+37/51 3+38/51 3+39/51
3+40/51 ...
3+41/51 3+42/51 3+43/51 3+44/51 3+45/51 3+46/51 3+47/51 3+48/51 3+49/51
3+50/51 ...
4+1/51 4+2/51 4+3/51 4+4/51 4+5/51 4+6/51 4+7/51 4+8/51 4+9/51 4+10/51 ...
4+11/51 4+12/51 4+13/51 4+14/51 4+15/51 4+16/51 4+17/51 4+18/51 4+19/51
4+20/51 ...
4+21/51 4+22/51 4+23/51 4+24/51 4+25/51 4+26/51 4+27/51 4+28/51 4+29/51
4+30/51 ...
4+31/51 4+32/51 4+33/51 4+34/51 4+35/51 4+36/51 4+37/51 4+38/51 4+39/51
4+40/51 ...
4+41/51 4+42/51 4+43/51 4+44/51 4+45/51 4+46/51 4+47/51 4+48/51 4+49/51
4+50/51 ...
5+1/51 5+2/51 5+3/51 5+4/51 5+5/51 5+6/51 5+7/51 5+8/51 5+9/51 5+10/51 ...
5+11/51 5+12/51 5+13/51 5+14/51 5+15/51 5+16/51 5+17/51 5+18/51 5+19/51
5+20/51 ...
5+21/51 5+22/51 5+23/51 5+24/51 5+25/51 5+26/51 5+27/51 5+28/51 5+29/51
5+30/51 ...
5+31/51 5+32/51 5+33/51 5+34/51 5+35/51 5+36/51 5+37/51 5+38/51 5+39/51
5+40/51 ...
5+41/51 5+42/51 5+43/51 5+44/51 5+45/51 5+46/51 5+47/51 5+48/51 5+49/51
5+50/51 ...
6+1/51 6+2/51 6+3/51 6+4/51 6+5/51 6+6/51 6+7/51 6+8/51 6+9/51 6+10/51 ...
6+11/51 6+12/51 6+13/51 6+14/51 6+15/51 6+16/51 6+17/51 6+18/51 6+19/51
6+20/51 ...
6+21/51 6+22/51 6+23/51 6+24/51 6+25/51 6+26/51 6+27/51 6+28/51 6+29/51
6+30/51 ...
6+31/51 6+32/51 6+33/51 6+34/51 6+35/51 6+36/51 6+37/51 6+38/51 6+39/51
6+40/51 ...
6+41/51 6+42/51 6+43/51 6+44/51 6+45/51 6+46/51 6+47/51 6+48/51 6+49/51
6+50/51]';
block.OutputPort(1).Data = interp1(x,block.InputPort(1).Data,xi,'spline');
%endfunction

```

S_OFDMP11

```

function msfcn_S_OFDM(block)
% Level-2 M file S-Function for times two demo.
% Copyright 1990-2004 The MathWorks, Inc.
% $Revision: 1.1.6.1 $
    setup(block);
%endfunction
function setup(block)
    %% Register number of input and output ports
    block.NumInputPorts = 1;
    block.NumOutputPorts = 1;
    %% Setup functional port properties to dynamically
    %% inherited.
    block.SetPreCompInPortInfoToDynamic;
    block.SetPreCompOutPortInfoToDynamic;
    % Override input port properties
    block.InputPort(1).DatatypeID = 0; % double
    block.InputPort(1).Complexity = 'complex';
    % Override output port properties
    block.OutputPort(1).DatatypeID = 0; % double
    block.OutputPort(1).Complexity = 'complex';
    block.InputPort(1).DirectFeedthrough = true;

```

```

%% SetInputPortDimensions:
%%   Functionality      : Check and set input and optionally output
%%                        port dimensions
%%   C-Mex counterpart: mdlSetInputPortDimensionInfo
%%
block.RegBlockMethod('SetInputPortDimensions', @SetInpPortDims);
%%
%% SetOutputPortDimensions:
%%   Functionality      : Check and set output and optionally input
%%                        port dimensions
%%   C-Mex counterpart: mdlSetOutputPortDimensionInfo
%%
block.RegBlockMethod('SetOutputPortDimensions', @SetOutPortDims);
%% Set block sample time to inherited
block.SampleTimes = [-1 0];
%% Run accelerator on TLC
block.SetAccelRunOnTLC(true);
%% Register methods
block.RegBlockMethod('Outputs', @Output);
%endfunction
function SetInpPortDims(block, idx, di)
    block.InputPort(1).Dimensions = 11;
    block.OutputPort(1).Dimensions = 300;
%endfunction
function SetOutPortDims(block, idx, di)
    block.OutputPort(1).Dimensions = 300;
    block.InputPort(1).Dimensions = 11;
%endfunction
function Output(block)
x=[1 2 3 4 5 6 7 8 9 10 11]';
xi=[1+1/31 1+2/31 1+3/31 1+4/31 1+5/31 1+6/31 1+7/31 1+8/31 1+9/31 1+10/31
...
1+11/31 1+12/31 1+13/31 1+14/31 1+15/31 1+16/31 1+17/31 1+18/31 1+19/31
1+20/31 ...
1+21/31 1+22/31 1+23/31 1+24/31 1+25/31 1+26/31 1+27/31 1+28/31 1+29/31
1+30/31 ...
2+1/31 2+2/31 2+3/31 2+4/31 2+5/31 2+6/31 2+7/31 2+8/31 2+9/31 2+10/31 ...
2+11/31 2+12/31 2+13/31 2+14/31 2+15/31 2+16/31 2+17/31 2+18/31 2+19/31
2+20/31 ...
2+21/31 2+22/31 2+23/31 2+24/31 2+25/31 2+26/31 2+27/31 2+28/31 2+29/31
2+30/31 ...
3+1/31 3+2/31 3+3/31 3+4/31 3+5/31 3+6/31 3+7/31 3+8/31 3+9/31 3+10/31 ...
3+11/31 3+12/31 3+13/31 3+14/31 3+15/31 3+16/31 3+17/31 3+18/31 3+19/31
3+20/31 ...
3+21/31 3+22/31 3+23/31 3+24/31 3+25/31 3+26/31 3+27/31 3+28/31 3+29/31
3+30/31 ...
4+1/31 4+2/31 4+3/31 4+4/31 4+5/31 4+6/31 4+7/31 4+8/31 4+9/31 4+10/31 ...
4+11/31 4+12/31 4+13/31 4+14/31 4+15/31 4+16/31 4+17/31 4+18/31 4+19/31
4+20/31 ...
4+21/31 4+22/31 4+23/31 4+24/31 4+25/31 4+26/31 4+27/31 4+28/31 4+29/31
4+30/31 ...
5+1/31 5+2/31 5+3/31 5+4/31 5+5/31 5+6/31 5+7/31 5+8/31 5+9/31 5+10/31 ...
5+11/31 5+12/31 5+13/31 5+14/31 5+15/31 5+16/31 5+17/31 5+18/31 5+19/31
5+20/31 ...
5+21/31 5+22/31 5+23/31 5+24/31 5+25/31 5+26/31 5+27/31 5+28/31 5+29/31
5+30/31 ...
6+1/31 6+2/31 6+3/31 6+4/31 6+5/31 6+6/31 6+7/31 6+8/31 6+9/31 6+10/31 ...
6+11/31 6+12/31 6+13/31 6+14/31 6+15/31 6+16/31 6+17/31 6+18/31 6+19/31
6+20/31 ...
6+21/31 6+22/31 6+23/31 6+24/31 6+25/31 6+26/31 6+27/31 6+28/31 6+29/31
6+30/31 ...

```

```

7+1/31 7+2/31 7+3/31 7+4/31 7+5/31 7+6/31 7+7/31 7+8/31 7+9/31 7+10/31 ...
7+11/31 7+12/31 7+13/31 7+14/31 7+15/31 7+16/31 7+17/31 7+18/31 7+19/31
7+20/31 ...
7+21/31 7+22/31 7+23/31 7+24/31 7+25/31 7+26/31 7+27/31 7+28/31 7+29/31
7+30/31 ...
8+1/31 8+2/31 8+3/31 8+4/31 8+5/31 8+6/31 8+7/31 8+8/31 8+9/31 8+10/31 ...
8+11/31 8+12/31 8+13/31 8+14/31 8+15/31 8+16/31 8+17/31 8+18/31 8+19/31
8+20/31 ...
8+21/31 8+22/31 8+23/31 8+24/31 8+25/31 8+26/31 8+27/31 8+28/31 8+29/31
8+30/31 ...
9+1/31 9+2/31 9+3/31 9+4/31 9+5/31 9+6/31 9+7/31 9+8/31 9+9/31 9+10/31 ...
9+11/31 9+12/31 9+13/31 9+14/31 9+15/31 9+16/31 9+17/31 9+18/31 9+19/31
9+20/31 ...
9+21/31 9+22/31 9+23/31 9+24/31 9+25/31 9+26/31 9+27/31 9+28/31 9+29/31
9+30/31 ...
10+1/31 10+2/31 10+3/31 10+4/31 10+5/31 10+6/31 10+7/31 10+8/31 10+9/31
10+10/31 ...
10+11/31 10+12/31 10+13/31 10+14/31 10+15/31 10+16/31 10+17/31 10+18/31
10+19/31 10+20/31 ...
10+21/31 10+22/31 10+23/31 10+24/31 10+25/31 10+26/31 10+27/31 10+28/31
10+29/31 10+30/31]';
block.OutputPort(1).Data = interp1(x,block.InputPort(1).Data,xi,'spline');
%endfunction

```

S_OFDMP13

```

function msfcn_S_OFDM(block)
% Level-2 M file S-Function for times two demo.
% Copyright 1990-2004 The MathWorks, Inc.
% $Revision: 1.1.6.1 $
    setup(block);
%endfunction
function setup(block)
    %% Register number of input and output ports
    block.NumInputPorts = 1;
    block.NumOutputPorts = 1;
    %% Setup functional port properties to dynamically
    %% inherited.
    block.SetPreCompInPortInfoToDynamic;
    block.SetPreCompOutPortInfoToDynamic;
    % Override input port properties
    block.InputPort(1).DatatypeID = 0; % double
    block.InputPort(1).Complexity = 'complex';
    % Override output port properties
    block.OutputPort(1).DatatypeID = 0; % double
    block.OutputPort(1).Complexity = 'complex';
    block.InputPort(1).DirectFeedthrough = true;
    %% SetInputPortDimensions:
    %% Functionality : Check and set input and optionally output
    %% port dimensions
    %% C-Mex counterpart: mdlSetInputPortDimensionInfo
    %%
    block.RegBlockMethod('SetInputPortDimensions', @SetInpPortDims);
    %%
    %% SetOutputPortDimensions:
    %% Functionality : Check and set output and optionally input
    %% port dimensions
    %% C-Mex counterpart: mdlSetOutputPortDimensionInfo
    %%
    block.RegBlockMethod('SetOutputPortDimensions', @SetOutPortDims);
    %% Set block sample time to inherited
    block.SampleTimes = [-1 0];

```

```

%% Run accelerator on TLC
block.SetAccelRunOnTLC(true);
%% Register methods
block.RegBlockMethod('Outputs',@Output);
%endfunction
function SetInpPortDims(block, idx, di)
    block.InputPort(1).Dimensions = 13;
    block.OutputPort(1).Dimensions = 300;
%endfunction
function SetOutPortDims(block, idx, di)
    block.OutputPort(1).Dimensions = 300;
    block.InputPort(1).Dimensions = 13;
%endfunction
function Output(block)
x=[1 2 3 4 5 6 7 8 9 10 11 12 13]';
xi=[1+1/26 1+2/26 1+3/26 1+4/26 1+5/26 1+6/26 1+7/26 1+8/26 1+9/26 1+10/26
...
1+11/26 1+12/26 1+13/26 1+14/26 1+15/26 1+16/26 1+17/26 1+18/26 1+19/26
1+20/26 ...
1+21/26 1+22/26 1+23/26 1+24/26 1+25/26 ...
2+1/26 2+2/26 2+3/26 2+4/26 2+5/26 2+6/26 2+7/26 2+8/26 2+9/26 2+10/26 ...
2+11/26 2+12/26 2+13/26 2+14/26 2+15/26 2+16/26 2+17/26 2+18/26 2+19/26
2+20/26 ...
2+21/26 2+22/26 2+23/26 2+24/26 2+25/26 ...
3+1/26 3+2/26 3+3/26 3+4/26 3+5/26 3+6/26 3+7/26 3+8/26 3+9/26 3+10/26 ...
3+11/26 3+12/26 3+13/26 3+14/26 3+15/26 3+16/26 3+17/26 3+18/26 3+19/26
3+20/26 ...
3+21/26 3+22/26 3+23/26 3+24/26 3+25/26 ...
4+1/26 4+2/26 4+3/26 4+4/26 4+5/26 4+6/26 4+7/26 4+8/26 4+9/26 4+10/26 ...
4+11/26 4+12/26 4+13/26 4+14/26 4+15/26 4+16/26 4+17/26 4+18/26 4+19/26
4+20/26 ...
4+21/26 4+22/26 4+23/26 4+24/26 4+25/26 ...
5+1/26 5+2/26 5+3/26 5+4/26 5+5/26 5+6/26 5+7/26 5+8/26 5+9/26 5+10/26 ...
5+11/26 5+12/26 5+13/26 5+14/26 5+15/26 5+16/26 5+17/26 5+18/26 5+19/26
5+20/26 ...
5+21/26 5+22/26 5+23/26 5+24/26 5+25/26 ...
6+1/26 6+2/26 6+3/26 6+4/26 6+5/26 6+6/26 6+7/26 6+8/26 6+9/26 6+10/26 ...
6+11/26 6+12/26 6+13/26 6+14/26 6+15/26 6+16/26 6+17/26 6+18/26 6+19/26
6+20/26 ...
6+21/26 6+22/26 6+23/26 6+24/26 6+25/26 ...
7+1/26 7+2/26 7+3/26 7+4/26 7+5/26 7+6/26 7+7/26 7+8/26 7+9/26 7+10/26 ...
7+11/26 7+12/26 7+13/26 7+14/26 7+15/26 7+16/26 7+17/26 7+18/26 7+19/26
7+20/26 ...
7+21/26 7+22/26 7+23/26 7+24/26 7+25/26 ...
8+1/26 8+2/26 8+3/26 8+4/26 8+5/26 8+6/26 8+7/26 8+8/26 8+9/26 8+10/26 ...
8+11/26 8+12/26 8+13/26 8+14/26 8+15/26 8+16/26 8+17/26 8+18/26 8+19/26
8+20/26 ...
8+21/26 8+22/26 8+23/26 8+24/26 8+25/26 ...
9+1/26 9+2/26 9+3/26 9+4/26 9+5/26 9+6/26 9+7/26 9+8/26 9+9/26 9+10/26 ...
9+11/26 9+12/26 9+13/26 9+14/26 9+15/26 9+16/26 9+17/26 9+18/26 9+19/26
9+20/26 ...
9+21/26 9+22/26 9+23/26 9+24/26 9+25/26 ...
10+1/26 10+2/26 10+3/26 10+4/26 10+5/26 10+6/26 10+7/26 10+8/26 10+9/26
10+10/26 ...
10+11/26 10+12/26 10+13/26 10+14/26 10+15/26 10+16/26 10+17/26 10+18/26
10+19/26 10+20/26 ...
10+21/26 10+22/26 10+23/26 10+24/26 10+25/26 ...
11+1/26 11+2/26 11+3/26 11+4/26 11+5/26 11+6/26 11+7/26 11+8/26 11+9/26
11+10/26 ...
11+11/26 11+12/26 11+13/26 11+14/26 11+15/26 11+16/26 11+17/26 11+18/26
11+19/26 11+20/26 ...

```

```

11+21/26 11+22/26 11+23/26 11+24/26 11+25/26 ...
12+1/26 12+2/26 12+3/26 12+4/26 12+5/26 12+6/26 12+7/26 12+8/26 12+9/26
12+10/26 ...
12+11/26 12+12/26 12+13/26 12+14/26 12+15/26 12+16/26 12+17/26 12+18/26
12+19/26 12+20/26 ...
12+21/26 12+22/26 12+23/26 12+24/26 12+25/26]';
block.OutputPort(1).Data = interp1(x,block.InputPort(1).Data,xi,'spline');
%endfunction

```

S_OFDMP16

```

function msfcn_S_OFDM(block)
% Level-2 M file S-Function for times two demo.
% Copyright 1990-2004 The MathWorks, Inc.
% $Revision: 1.1.6.1 $
    setup(block);
%endfunction
function setup(block)
    %% Register number of input and output ports
    block.NumInputPorts = 1;
    block.NumOutputPorts = 1;
    %% Setup functional port properties to dynamically
    %% inherited.
    block.SetPreCompInpPortInfoToDynamic;
    block.SetPreCompOutPortInfoToDynamic;
    % Override input port properties
    block.InputPort(1).DatatypeID = 0; % double
    block.InputPort(1).Complexity = 'complex';
    % Override output port properties
    block.OutputPort(1).DatatypeID = 0; % double
    block.OutputPort(1).Complexity = 'complex';
    block.InputPort(1).DirectFeedthrough = true;
    %% SetInputPortDimensions:
    %% Functionality : Check and set input and optionally output
    %% port dimensions
    %% C-Mex counterpart: mdlSetInputPortDimensionInfo
    %%
    block.RegBlockMethod('SetInputPortDimensions', @SetInpPortDims);
    %%
    %% SetOutputPortDimensions:
    %% Functionality : Check and set output and optionally input
    %% port dimensions
    %% C-Mex counterpart: mdlSetOutputPortDimensionInfo
    %%
    block.RegBlockMethod('SetOutputPortDimensions', @SetOutPortDims);
    %% Set block sample time to inherited
    block.SampleTimes = [-1 0];
    %% Run accelerator on TLC
    block.SetAccelRunOnTLC(true);
    %% Register methods
    block.RegBlockMethod('Outputs',@Output);
%endfunction
function SetInpPortDims(block, idx, di)
    block.InputPort(1).Dimensions = 16;
    block.OutputPort(1).Dimensions = 300;
%endfunction
function SetOutPortDims(block, idx, di)
    block.OutputPort(1).Dimensions = 300;
    block.InputPort(1).Dimensions = 16;
%endfunction
function Output(block)
x=[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16]';

```

```

xi=[1+1/21 1+2/21 1+3/21 1+4/21 1+5/21 1+6/21 1+7/21 1+8/21 1+9/21 1+10/21
1+11/21 1+12/21 1+13/21 1+14/21 1+15/21 1+16/21 1+17/21 1+18/21 1+19/21
1+20/21 ...
2+1/21 2+2/21 2+3/21 2+4/21 2+5/21 2+6/21 2+7/21 2+8/21 2+9/21 2+10/21
2+11/21 2+12/21 2+13/21 2+14/21 2+15/21 2+16/21 2+17/21 2+18/21 2+19/21
2+20/21 ...
3+1/21 3+2/21 3+3/21 3+4/21 3+5/21 3+6/21 3+7/21 3+8/21 3+9/21 3+10/21
3+11/21 3+12/21 3+13/21 3+14/21 3+15/21 3+16/21 3+17/21 3+18/21 3+19/21
3+20/21 ...
4+1/21 4+2/21 4+3/21 4+4/21 4+5/21 4+6/21 4+7/21 4+8/21 4+9/21 4+10/21
4+11/21 4+12/21 4+13/21 4+14/21 4+15/21 4+16/21 4+17/21 4+18/21 4+19/21
4+20/21 ...
5+1/21 5+2/21 5+3/21 5+4/21 5+5/21 5+6/21 5+7/21 5+8/21 5+9/21 5+10/21
5+11/21 5+12/21 5+13/21 5+14/21 5+15/21 5+16/21 5+17/21 5+18/21 5+19/21
5+20/21 ...
6+1/21 6+2/21 6+3/21 6+4/21 6+5/21 6+6/21 6+7/21 6+8/21 6+9/21 6+10/21
6+11/21 6+12/21 6+13/21 6+14/21 6+15/21 6+16/21 6+17/21 6+18/21 6+19/21
6+20/21 ...
7+1/21 7+2/21 7+3/21 7+4/21 7+5/21 7+6/21 7+7/21 7+8/21 7+9/21 7+10/21
7+11/21 7+12/21 7+13/21 7+14/21 7+15/21 7+16/21 7+17/21 7+18/21 7+19/21
7+20/21 ...
8+1/21 8+2/21 8+3/21 8+4/21 8+5/21 8+6/21 8+7/21 8+8/21 8+9/21 8+10/21
8+11/21 8+12/21 8+13/21 8+14/21 8+15/21 8+16/21 8+17/21 8+18/21 8+19/21
8+20/21 ...
9+1/21 9+2/21 9+3/21 9+4/21 9+5/21 9+6/21 9+7/21 9+8/21 9+9/21 9+10/21
9+11/21 9+12/21 9+13/21 9+14/21 9+15/21 9+16/21 9+17/21 9+18/21 9+19/21
9+20/21 ...
10+1/21 10+2/21 10+3/21 10+4/21 10+5/21 10+6/21 10+7/21 10+8/21 10+9/21
10+10/21 10+11/21 10+12/21 10+13/21 10+14/21 10+15/21 10+16/21 10+17/21
10+18/21 10+19/21 10+20/21 ...
11+1/21 11+2/21 11+3/21 11+4/21 11+5/21 11+6/21 11+7/21 11+8/21 11+9/21
11+10/21 11+11/21 11+12/21 11+13/21 11+14/21 11+15/21 11+16/21 11+17/21
11+18/21 11+19/21 11+20/21 ...
12+1/21 12+2/21 12+3/21 12+4/21 12+5/21 12+6/21 12+7/21 12+8/21 12+9/21
12+10/21 12+11/21 12+12/21 12+13/21 12+14/21 12+15/21 12+16/21 12+17/21
12+18/21 12+19/21 12+20/21 ...
13+1/21 13+2/21 13+3/21 13+4/21 13+5/21 13+6/21 13+7/21 13+8/21 13+9/21
13+10/21 13+11/21 13+12/21 13+13/21 13+14/21 13+15/21 13+16/21 13+17/21
13+18/21 13+19/21 13+20/21 ...
14+1/21 14+2/21 14+3/21 14+4/21 14+5/21 14+6/21 14+7/21 14+8/21 14+9/21
14+10/21 14+11/21 14+12/21 14+13/21 14+14/21 14+15/21 14+16/21 14+17/21
14+18/21 14+19/21 14+20/21 ...
15+1/21 15+2/21 15+3/21 15+4/21 15+5/21 15+6/21 15+7/21 15+8/21 15+9/21
15+10/21 15+11/21 15+12/21 15+13/21 15+14/21 15+15/21 15+16/21 15+17/21
15+18/21 15+19/21 15+20/21]';
block.OutputPort(1).Data = interp1(x,block.InputPort(1).Data,xi,'spline');
%endfunction

```

S_OFDMP21

```

function msfcn_S_OFDM(block)
% Level-2 M file S-Function for times two demo.
% Copyright 1990-2004 The MathWorks, Inc.
% $Revision: 1.1.6.1 $
    setup(block);
%endfunction
function setup(block)
    %% Register number of input and output ports
    block.NumInputPorts = 1;
    block.NumOutputPorts = 1;
    %% Setup functional port properties to dynamically

```

```

%% inherited.
block.SetPreCompInpPortInfoToDynamic;
block.SetPreCompOutPortInfoToDynamic;
% Override input port properties
block.InputPort(1).DatatypeID = 0; % double
block.InputPort(1).Complexity = 'complex';
% Override output port properties
block.OutputPort(1).DatatypeID = 0; % double
block.OutputPort(1).Complexity = 'complex';
block.InputPort(1).DirectFeedthrough = true;
%% SetInputPortDimensions:
%%     Functionality      : Check and set input and optionally output
%%                          port dimensions
%%     C-Mex counterpart: mdlSetInputPortDimensionInfo
%%
block.RegBlockMethod('SetInputPortDimensions', @SetInpPortDims);
%%
%% SetOutputPortDimensions:
%%     Functionality      : Check and set output and optionally input
%%                          port dimensions
%%     C-Mex counterpart: mdlSetOutputPortDimensionInfo
%%
block.RegBlockMethod('SetOutputPortDimensions', @SetOutPortDims);
%% Set block sample time to inherited
block.SampleTimes = [-1 0];
%% Run accelerator on TLC
block.SetAccelRunOnTLC(true);
%% Register methods
block.RegBlockMethod('Outputs', @Output);
%endfunction
function SetInpPortDims(block, idx, di)
    block.InputPort(1).Dimensions = 21;
    block.OutputPort(1).Dimensions = 300;
%endfunction
function SetOutPortDims(block, idx, di)
    block.OutputPort(1).Dimensions = 300;
    block.InputPort(1).Dimensions = 21;
%endfunction
function Output(block)
x=[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21]';
xi=[1+1/16 1+2/16 1+3/16 1+4/16 1+5/16 1+6/16 1+7/16 1+8/16 1+9/16 1+10/16
1+11/16 1+12/16 1+13/16 1+14/16 1+15/16 ...
2+1/16 2+2/16 2+3/16 2+4/16 2+5/16 2+6/16 2+7/16 2+8/16 2+9/16 2+10/16
2+11/16 2+12/16 2+13/16 2+14/16 2+15/16 ...
3+1/16 3+2/16 3+3/16 3+4/16 3+5/16 3+6/16 3+7/16 3+8/16 3+9/16 3+10/16
3+11/16 3+12/16 3+13/16 3+14/16 3+15/16 ...
4+1/16 4+2/16 4+3/16 4+4/16 4+5/16 4+6/16 4+7/16 4+8/16 4+9/16 4+10/16
4+11/16 4+12/16 4+13/16 4+14/16 4+15/16 ...
5+1/16 5+2/16 5+3/16 5+4/16 5+5/16 5+6/16 5+7/16 5+8/16 5+9/16 5+10/16
5+11/16 5+12/16 5+13/16 5+14/16 5+15/16 ...
6+1/16 6+2/16 6+3/16 6+4/16 6+5/16 6+6/16 6+7/16 6+8/16 6+9/16 6+10/16
6+11/16 6+12/16 6+13/16 6+14/16 6+15/16 ...
7+1/16 7+2/16 7+3/16 7+4/16 7+5/16 7+6/16 7+7/16 7+8/16 7+9/16 7+10/16
7+11/16 7+12/16 7+13/16 7+14/16 7+15/16 ...
8+1/16 8+2/16 8+3/16 8+4/16 8+5/16 8+6/16 8+7/16 8+8/16 8+9/16 8+10/16
8+11/16 8+12/16 8+13/16 8+14/16 8+15/16 ...
9+1/16 9+2/16 9+3/16 9+4/16 9+5/16 9+6/16 9+7/16 9+8/16 9+9/16 9+10/16
9+11/16 9+12/16 9+13/16 9+14/16 9+15/16 ...
10+1/16 10+2/16 10+3/16 10+4/16 10+5/16 10+6/16 10+7/16 10+8/16 10+9/16
10+10/16 10+11/16 10+12/16 10+13/16 10+14/16 10+15/16 ...

```

```

11+1/16 11+2/16 11+3/16 11+4/16 11+5/16 11+6/16 11+7/16 11+8/16 11+9/16
11+10/16 11+11/16 11+12/16 11+13/16 11+14/16 11+15/16 ...
12+1/16 12+2/16 12+3/16 12+4/16 12+5/16 12+6/16 12+7/16 12+8/16 12+9/16
12+10/16 12+11/16 12+12/16 12+13/16 12+14/16 12+15/16 ...
13+1/16 13+2/16 13+3/16 13+4/16 13+5/16 13+6/16 13+7/16 13+8/16 13+9/16
13+10/16 13+11/16 13+12/16 13+13/16 13+14/16 13+15/16 ...
14+1/16 14+2/16 14+3/16 14+4/16 14+5/16 14+6/16 14+7/16 14+8/16 14+9/16
14+10/16 14+11/16 14+12/16 14+13/16 14+14/16 14+15/16 ...
15+1/16 15+2/16 15+3/16 15+4/16 15+5/16 15+6/16 15+7/16 15+8/16 15+9/16
15+10/16 15+11/16 15+12/16 15+13/16 15+14/16 15+15/16 ...
16+1/16 16+2/16 16+3/16 16+4/16 16+5/16 16+6/16 16+7/16 16+8/16 16+9/16
16+10/16 16+11/16 16+12/16 16+13/16 16+14/16 16+15/16 ...
17+1/16 17+2/16 17+3/16 17+4/16 17+5/16 17+6/16 17+7/16 17+8/16 17+9/16
17+10/16 17+11/16 17+12/16 17+13/16 17+14/16 17+15/16 ...
18+1/16 18+2/16 18+3/16 18+4/16 18+5/16 18+6/16 18+7/16 18+8/16 18+9/16
18+10/16 18+11/16 18+12/16 18+13/16 18+14/16 18+15/16 ...
19+1/16 19+2/16 19+3/16 19+4/16 19+5/16 19+6/16 19+7/16 19+8/16 19+9/16
19+10/16 19+11/16 19+12/16 19+13/16 19+14/16 19+15/16 ...
20+1/16 20+2/16 20+3/16 20+4/16 20+5/16 20+6/16 20+7/16 20+8/16 20+9/16
20+10/16 20+11/16 20+12/16 20+13/16 20+14/16 20+15/16]';
block.OutputPort(1).Data = interp1(x,block.InputPort(1).Data,xi,'spline');
%endfunction

```

S_OFDMP26

```

function msfcn_S_OFDM(block)
% Level-2 M file S-Function for times two demo.
% Copyright 1990-2004 The MathWorks, Inc.
% $Revision: 1.1.6.1 $
    setup(block);
%endfunction
function setup(block)
    %% Register number of input and output ports
    block.NumInputPorts = 1;
    block.NumOutputPorts = 1;
    %% Setup functional port properties to dynamically
    %% inherited.
    block.SetPreCompInPortInfoToDynamic;
    block.SetPreCompOutPortInfoToDynamic;
    % Override input port properties
    block.InputPort(1).DatatypeID = 0; % double
    block.InputPort(1).Complexity = 'complex';
    % Override output port properties
    block.OutputPort(1).DatatypeID = 0; % double
    block.OutputPort(1).Complexity = 'complex';
    block.InputPort(1).DirectFeedthrough = true;
    %% SetInputPortDimensions:
    %% Functionality : Check and set input and optionally output
    %% port dimensions
    %% C-Mex counterpart: mdlSetInputPortDimensionInfo
    %%
    block.RegBlockMethod('SetInputPortDimensions', @SetInpPortDims);
    %%
    %% SetOutputPortDimensions:
    %% Functionality : Check and set output and optionally input
    %% port dimensions
    %% C-Mex counterpart: mdlSetOutputPortDimensionInfo
    %%
    block.RegBlockMethod('SetOutputPortDimensions', @SetOutPortDims);
    %% Set block sample time to inherited
    block.SampleTimes = [-1 0];
    %% Run accelerator on TLC

```



```

block.SetAccelRunOnTLC(true);
%% Register methods
block.RegBlockMethod('Outputs',@Output);
%endfunction
function SetInpPortDims(block, idx, di)
    block.InputPort(1).Dimensions = 26;
    block.OutputPort(1).Dimensions = 300;
%endfunction
function SetOutPortDims(block, idx, di)
    block.OutputPort(1).Dimensions = 300;
    block.InputPort(1).Dimensions = 26;
%endfunction
function Output(block)
x=[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26]';
xi=[1+1/13 1+2/13 1+3/13 1+4/13 1+5/13 1+6/13 1+7/13 1+8/13 1+9/13 1+10/13
1+11/13 1+12/13 ...
2+1/13 2+2/13 2+3/13 2+4/13 2+5/13 2+6/13 2+7/13 2+8/13 2+9/13 2+10/13
2+11/13 2+12/13 ...
3+1/13 3+2/13 3+3/13 3+4/13 3+5/13 3+6/13 3+7/13 3+8/13 3+9/13 3+10/13
3+11/13 3+12/13 ...
4+1/13 4+2/13 4+3/13 4+4/13 4+5/13 4+6/13 4+7/13 4+8/13 4+9/13 4+10/13
4+11/13 4+12/13 ...
5+1/13 5+2/13 5+3/13 5+4/13 5+5/13 5+6/13 5+7/13 5+8/13 5+9/13 5+10/13
5+11/13 5+12/13 ...
6+1/13 6+2/13 6+3/13 6+4/13 6+5/13 6+6/13 6+7/13 6+8/13 6+9/13 6+10/13
6+11/13 6+12/13 ...
7+1/13 7+2/13 7+3/13 7+4/13 7+5/13 7+6/13 7+7/13 7+8/13 7+9/13 7+10/13
7+11/13 7+12/13 ...
8+1/13 8+2/13 8+3/13 8+4/13 8+5/13 8+6/13 8+7/13 8+8/13 8+9/13 8+10/13
8+11/13 8+12/13 ...
9+1/13 9+2/13 9+3/13 9+4/13 9+5/13 9+6/13 9+7/13 9+8/13 9+9/13 9+10/13
9+11/13 9+12/13 ...
10+1/13 10+2/13 10+3/13 10+4/13 10+5/13 10+6/13 10+7/13 10+8/13 10+9/13
10+10/13 10+11/13 10+12/13 ...
11+1/13 11+2/13 11+3/13 11+4/13 11+5/13 11+6/13 11+7/13 11+8/13 11+9/13
11+10/13 11+11/13 11+12/13 ...
12+1/13 12+2/13 12+3/13 12+4/13 12+5/13 12+6/13 12+7/13 12+8/13 12+9/13
12+10/13 12+11/13 12+12/13 ...
13+1/13 13+2/13 13+3/13 13+4/13 13+5/13 13+6/13 13+7/13 13+8/13 13+9/13
13+10/13 13+11/13 13+12/13 ...
14+1/13 14+2/13 14+3/13 14+4/13 14+5/13 14+6/13 14+7/13 14+8/13 14+9/13
14+10/13 14+11/13 14+12/13 ...
15+1/13 15+2/13 15+3/13 15+4/13 15+5/13 15+6/13 15+7/13 15+8/13 15+9/13
15+10/13 15+11/13 15+12/13 ...
16+1/13 16+2/13 16+3/13 16+4/13 16+5/13 16+6/13 16+7/13 16+8/13 16+9/13
16+10/13 16+11/13 16+12/13 ...
17+1/13 17+2/13 17+3/13 17+4/13 17+5/13 17+6/13 17+7/13 17+8/13 17+9/13
17+10/13 17+11/13 17+12/13 ...
18+1/13 18+2/13 18+3/13 18+4/13 18+5/13 18+6/13 18+7/13 18+8/13 18+9/13
18+10/13 18+11/13 18+12/13 ...
19+1/13 19+2/13 19+3/13 19+4/13 19+5/13 19+6/13 19+7/13 19+8/13 19+9/13
19+10/13 19+11/13 19+12/13 ...
20+1/13 20+2/13 20+3/13 20+4/13 20+5/13 20+6/13 20+7/13 20+8/13 20+9/13
20+10/13 20+11/13 20+12/13 ...
21+1/13 21+2/13 21+3/13 21+4/13 21+5/13 21+6/13 21+7/13 21+8/13 21+9/13
21+10/13 21+11/13 21+12/13 ...
22+1/13 22+2/13 22+3/13 22+4/13 22+5/13 22+6/13 22+7/13 22+8/13 22+9/13
22+10/13 22+11/13 22+12/13 ...
23+1/13 23+2/13 23+3/13 23+4/13 23+5/13 23+6/13 23+7/13 23+8/13 23+9/13
23+10/13 23+11/13 23+12/13 ...

```

```

24+1/13 24+2/13 24+3/13 24+4/13 24+5/13 24+6/13 24+7/13 24+8/13 24+9/13
24+10/13 24+11/13 24+12/13 ...
25+1/13 25+2/13 25+3/13 25+4/13 25+5/13 25+6/13 25+7/13 25+8/13 25+9/13
25+10/13 25+11/13 25+12/13]';

```

```

block.OutputPort(1).Data = interp1(x,block.InputPort(1).Data,xi,'spline');
%endfunction

```

S_OFDMP31

```

function msfcn_S_OFDM(block)
% Level-2 M file S-Function for times two demo.
% Copyright 1990-2004 The MathWorks, Inc.
% $Revision: 1.1.6.1 $
    setup(block);
%endfunction
function setup(block)
    %% Register number of input and output ports
    block.NumInputPorts = 1;
    block.NumOutputPorts = 1;
    %% Setup functional port properties to dynamically
    %% inherited.
    block.SetPreCompInpPortInfoToDynamic;
    block.SetPreCompOutPortInfoToDynamic;
    % Override input port properties
    block.InputPort(1).DatatypeID = 0; % double
    block.InputPort(1).Complexity = 'complex';
    % Override output port properties
    block.OutputPort(1).DatatypeID = 0; % double
    block.OutputPort(1).Complexity = 'complex';
    block.InputPort(1).DirectFeedthrough = true;
    %% SetInputPortDimensions:
    %% Functionality : Check and set input and optionally output
    %% port dimensions
    %% C-Mex counterpart: mdlSetInputPortDimensionInfo
    %%
    block.RegBlockMethod('SetInputPortDimensions', @SetInpPortDims);
    %%
    %% SetOutputPortDimensions:
    %% Functionality : Check and set output and optionally input
    %% port dimensions
    %% C-Mex counterpart: mdlSetOutputPortDimensionInfo
    %%
    block.RegBlockMethod('SetOutputPortDimensions', @SetOutPortDims);
    %% Set block sample time to inherited
    block.SampleTimes = [-1 0];
    %% Run accelerator on TLC
    block.SetAccelRunOnTLC(true);
    %% Register methods
    block.RegBlockMethod('Outputs',@Output);
%endfunction
function SetInpPortDims(block, idx, di)
    block.InputPort(1).Dimensions = 31;
    block.OutputPort(1).Dimensions = 300;
%endfunction
function SetOutPortDims(block, idx, di)
    block.OutputPort(1).Dimensions = 300;
    block.InputPort(1).Dimensions = 31;
%endfunction
function Output(block)

```

```

x=[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27
28 29 30 31]';
xi=[1+1/11 1+2/11 1+3/11 1+4/11 1+5/11 1+6/11 1+7/11 1+8/11 1+9/11 1+10/11
...
2+1/11 2+2/11 2+3/11 2+4/11 2+5/11 2+6/11 2+7/11 2+8/11 2+9/11 2+10/11 ...
3+1/11 3+2/11 3+3/11 3+4/11 3+5/11 3+6/11 3+7/11 3+8/11 3+9/11 3+10/11 ...
4+1/11 4+2/11 4+3/11 4+4/11 4+5/11 4+6/11 4+7/11 4+8/11 4+9/11 4+10/11 ...
5+1/11 5+2/11 5+3/11 5+4/11 5+5/11 5+6/11 5+7/11 5+8/11 5+9/11 5+10/11 ...
6+1/11 6+2/11 6+3/11 6+4/11 6+5/11 6+6/11 6+7/11 6+8/11 6+9/11 6+10/11 ...
7+1/11 7+2/11 7+3/11 7+4/11 7+5/11 7+6/11 7+7/11 7+8/11 7+9/11 7+10/11 ...
8+1/11 8+2/11 8+3/11 8+4/11 8+5/11 8+6/11 8+7/11 8+8/11 8+9/11 8+10/11 ...
9+1/11 9+2/11 9+3/11 9+4/11 9+5/11 9+6/11 9+7/11 9+8/11 9+9/11 9+10/11 ...
10+1/11 10+2/11 10+3/11 10+4/11 10+5/11 10+6/11 10+7/11 10+8/11 10+9/11
10+10/11 ...
11+1/11 11+2/11 11+3/11 11+4/11 11+5/11 11+6/11 11+7/11 11+8/11 11+9/11
11+10/11 ...
12+1/11 12+2/11 12+3/11 12+4/11 12+5/11 12+6/11 12+7/11 12+8/11 12+9/11
12+10/11 ...
13+1/11 13+2/11 13+3/11 13+4/11 13+5/11 13+6/11 13+7/11 13+8/11 13+9/11
13+10/11 ...
14+1/11 14+2/11 14+3/11 14+4/11 14+5/11 14+6/11 14+7/11 14+8/11 14+9/11
14+10/11 ...
15+1/11 15+2/11 15+3/11 15+4/11 15+5/11 15+6/11 15+7/11 15+8/11 15+9/11
15+10/11 ...
16+1/11 16+2/11 16+3/11 16+4/11 16+5/11 16+6/11 16+7/11 16+8/11 16+9/11
16+10/11 ...
17+1/11 17+2/11 17+3/11 17+4/11 17+5/11 17+6/11 17+7/11 17+8/11 17+9/11
17+10/11 ...
18+1/11 18+2/11 18+3/11 18+4/11 18+5/11 18+6/11 18+7/11 18+8/11 18+9/11
18+10/11 ...
19+1/11 19+2/11 19+3/11 19+4/11 19+5/11 19+6/11 19+7/11 19+8/11 19+9/11
19+10/11 ...
20+1/11 20+2/11 20+3/11 20+4/11 20+5/11 20+6/11 20+7/11 20+8/11 20+9/11
20+10/11 ...
21+1/11 21+2/11 21+3/11 21+4/11 21+5/11 21+6/11 21+7/11 21+8/11 21+9/11
21+10/11 ...
22+1/11 22+2/11 22+3/11 22+4/11 22+5/11 22+6/11 22+7/11 22+8/11 22+9/11
22+10/11 ...
23+1/11 23+2/11 23+3/11 23+4/11 23+5/11 23+6/11 23+7/11 23+8/11 23+9/11
23+10/11 ...
24+1/11 24+2/11 24+3/11 24+4/11 24+5/11 24+6/11 24+7/11 24+8/11 24+9/11
24+10/11 ...
25+1/11 25+2/11 25+3/11 25+4/11 25+5/11 25+6/11 25+7/11 25+8/11 25+9/11
25+10/11 ...
26+1/11 26+2/11 26+3/11 26+4/11 26+5/11 26+6/11 26+7/11 26+8/11 26+9/11
26+10/11 ...
27+1/11 27+2/11 27+3/11 27+4/11 27+5/11 27+6/11 27+7/11 27+8/11 27+9/11
27+10/11 ...
28+1/11 28+2/11 28+3/11 28+4/11 28+5/11 28+6/11 28+7/11 28+8/11 28+9/11
28+10/11 ...
29+1/11 29+2/11 29+3/11 29+4/11 29+5/11 29+6/11 29+7/11 29+8/11 29+9/11
29+10/11 ...
30+1/11 30+2/11 30+3/11 30+4/11 30+5/11 30+6/11 30+7/11 30+8/11 30+9/11
30+10/11]';
block.OutputPort(1).Data = interp1(x,block.InputPort(1).Data,xi,'spline');
%endfunction

```

S_OFDMP51

```

function msfcn_S_OFDM(block)
% Level-2 M file S-Function for times two demo.

```

```

% Copyright 1990-2004 The MathWorks, Inc.
% $Revision: 1.1.6.1 $
setup(block);
%endfunction
function setup(block)
    %% Register number of input and output ports
    block.NumInputPorts = 1;
    block.NumOutputPorts = 1;
    %% Setup functional port properties to dynamically
    %% inherited.
    block.SetPreCompInPortInfoToDynamic;
    block.SetPreCompOutPortInfoToDynamic;
    % Override input port properties
    block.InputPort(1).DatatypeID = 0; % double
    block.InputPort(1).Complexity = 'complex';
    % Override output port properties
    block.OutputPort(1).DatatypeID = 0; % double
    block.OutputPort(1).Complexity = 'complex';
    block.InputPort(1).DirectFeedthrough = true;
    %% SetInputPortDimensions:
    %%   Functionality : Check and set input and optionally output
    %%                   port dimensions
    %%   C-Mex counterpart: mdlSetInputPortDimensionInfo
    %%
    block.RegBlockMethod('SetInputPortDimensions', @SetInpPortDims);
    %%
    %% SetOutputPortDimensions:
    %%   Functionality : Check and set output and optionally input
    %%                   port dimensions
    %%   C-Mex counterpart: mdlSetOutputPortDimensionInfo
    %%
    block.RegBlockMethod('SetOutputPortDimensions', @SetOutPortDims);
    %% Set block sample time to inherited
    block.SampleTimes = [-1 0];
    %% Run accelerator on TLC
    block.SetAccelRunOnTLC(true);
    %% Register methods
    block.RegBlockMethod('Outputs', @Output);
%endfunction
function SetInpPortDims(block, idx, di)
    block.InputPort(1).Dimensions = 51;
    block.OutputPort(1).Dimensions = 300;
%endfunction
function SetOutPortDims(block, idx, di)
    block.OutputPort(1).Dimensions = 300;
    block.InputPort(1).Dimensions = 51;
%endfunction
function Output(block)
x=[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27
28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51]';
xi=[1+1/7 1+2/7 1+3/7 1+4/7 1+5/7 1+6/7 ...
2+1/7 2+2/7 2+3/7 2+4/7 2+5/7 2+6/7 ...
3+1/7 3+2/7 3+3/7 3+4/7 3+5/7 3+6/7 ...
4+1/7 4+2/7 4+3/7 4+4/7 4+5/7 4+6/7 ...
5+1/7 5+2/7 5+3/7 5+4/7 5+5/7 5+6/7 ...
6+1/7 6+2/7 6+3/7 6+4/7 6+5/7 6+6/7 ...
7+1/7 7+2/7 7+3/7 7+4/7 7+5/7 7+6/7 ...
8+1/7 8+2/7 8+3/7 8+4/7 8+5/7 8+6/7 ...
9+1/7 9+2/7 9+3/7 9+4/7 9+5/7 9+6/7 ...
10+1/7 10+2/7 10+3/7 10+4/7 10+5/7 10+6/7 ...
11+1/7 11+2/7 11+3/7 11+4/7 11+5/7 11+6/7 ...

```

```

12+1/7 12+2/7 12+3/7 12+4/7 12+5/7 12+6/7 ...
13+1/7 13+2/7 13+3/7 13+4/7 13+5/7 13+6/7 ...
14+1/7 14+2/7 14+3/7 14+4/7 14+5/7 14+6/7 ...
15+1/7 15+2/7 15+3/7 15+4/7 15+5/7 15+6/7 ...
16+1/7 16+2/7 16+3/7 16+4/7 16+5/7 16+6/7 ...
17+1/7 17+2/7 17+3/7 17+4/7 17+5/7 17+6/7 ...
18+1/7 18+2/7 18+3/7 18+4/7 18+5/7 18+6/7 ...
19+1/7 19+2/7 19+3/7 19+4/7 19+5/7 19+6/7 ...
20+1/7 20+2/7 20+3/7 20+4/7 20+5/7 20+6/7 ...
21+1/7 21+2/7 21+3/7 21+4/7 21+5/7 21+6/7 ...
22+1/7 22+2/7 22+3/7 22+4/7 22+5/7 22+6/7 ...
23+1/7 23+2/7 23+3/7 23+4/7 23+5/7 23+6/7 ...
24+1/7 24+2/7 24+3/7 24+4/7 24+5/7 24+6/7 ...
25+1/7 25+2/7 25+3/7 25+4/7 25+5/7 25+6/7 ...
26+1/7 26+2/7 26+3/7 26+4/7 26+5/7 26+6/7 ...
27+1/7 27+2/7 27+3/7 27+4/7 27+5/7 27+6/7 ...
28+1/7 28+2/7 28+3/7 28+4/7 28+5/7 28+6/7 ...
29+1/7 29+2/7 29+3/7 29+4/7 29+5/7 29+6/7 ...
30+1/7 30+2/7 30+3/7 30+4/7 30+5/7 30+6/7 ...
31+1/7 31+2/7 31+3/7 31+4/7 31+5/7 31+6/7 ...
32+1/7 32+2/7 32+3/7 32+4/7 32+5/7 32+6/7 ...
33+1/7 33+2/7 33+3/7 33+4/7 33+5/7 33+6/7 ...
34+1/7 34+2/7 34+3/7 34+4/7 34+5/7 34+6/7 ...
35+1/7 35+2/7 35+3/7 35+4/7 35+5/7 35+6/7 ...
36+1/7 36+2/7 36+3/7 36+4/7 36+5/7 36+6/7 ...
37+1/7 37+2/7 37+3/7 37+4/7 37+5/7 37+6/7 ...
38+1/7 38+2/7 38+3/7 38+4/7 38+5/7 38+6/7 ...
39+1/7 39+2/7 39+3/7 39+4/7 39+5/7 39+6/7 ...
40+1/7 40+2/7 40+3/7 40+4/7 40+5/7 40+6/7 ...
41+1/7 41+2/7 41+3/7 41+4/7 41+5/7 41+6/7 ...
42+1/7 42+2/7 42+3/7 42+4/7 42+5/7 42+6/7 ...
43+1/7 43+2/7 43+3/7 43+4/7 43+5/7 43+6/7 ...
44+1/7 44+2/7 44+3/7 44+4/7 44+5/7 44+6/7 ...
45+1/7 45+2/7 45+3/7 45+4/7 45+5/7 45+6/7 ...
46+1/7 46+2/7 46+3/7 46+4/7 46+5/7 46+6/7 ...
47+1/7 47+2/7 47+3/7 47+4/7 47+5/7 47+6/7 ...
48+1/7 48+2/7 48+3/7 48+4/7 48+5/7 48+6/7 ...
49+1/7 49+2/7 49+3/7 49+4/7 49+5/7 49+6/7 ...
50+1/7 50+2/7 50+3/7 50+4/7 50+5/7 50+6/7]';
block.OutputPort(1).Data = interp1(x,block.InputPort(1).Data,xi,'spline');
%endfunction

```

S_OFDMP61

```

function msfcn_S_OFDM(block)
% Level-2 M file S-Function for times two demo.
% Copyright 1990-2004 The MathWorks, Inc.
% $Revision: 1.1.6.1 $
    setup(block);
%endfunction
function setup(block)
    %% Register number of input and output ports
    block.NumInputPorts = 1;
    block.NumOutputPorts = 1;
    %% Setup functional port properties to dynamically
    %% inherited.
    block.SetPreCompInpPortInfoToDynamic;
    block.SetPreCompOutPortInfoToDynamic;
    % Override input port properties

```

```

block.InputPort(1).DatatypeID = 0; % double
block.InputPort(1).Complexity = 'complex';
% Override output port properties
block.OutputPort(1).DatatypeID = 0; % double
block.OutputPort(1).Complexity = 'complex';
block.InputPort(1).DirectFeedthrough = true;
%% SetInputPortDimensions:
%%   Functionality      : Check and set input and optionally output
%%                        port dimensions
%%   C-Mex counterpart: mdlSetInputPortDimensionInfo
%%
block.RegBlockMethod('SetInputPortDimensions', @SetInpPortDims);
%%
%% SetOutputPortDimensions:
%%   Functionality      : Check and set output and optionally input
%%                        port dimensions
%%   C-Mex counterpart: mdlSetOutputPortDimensionInfo
%%
block.RegBlockMethod('SetOutputPortDimensions', @SetOutPortDims);
%% Set block sample time to inherited
block.SampleTimes = [-1 0];
%% Run accelerator on TLC
block.SetAccelRunOnTLC(true);
%% Register methods
block.RegBlockMethod('Outputs', @Output);
%endfunction
function SetInpPortDims(block, idx, di)
    block.InputPort(1).Dimensions = 61;
    block.OutputPort(1).Dimensions = 300;
%endfunction
function SetOutPortDims(block, idx, di)
    block.OutputPort(1).Dimensions = 300;
    block.InputPort(1).Dimensions = 61;
%endfunction
function Output(block)
x=[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27
28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52
53 54 55 56 57 58 59 60 61]';
xi=[1+1/6 1+2/6 1+3/6 1+4/6 1+5/6 ...
2+1/6 2+2/6 2+3/6 2+4/6 2+5/6 ...
3+1/6 3+2/6 3+3/6 3+4/6 3+5/6 ...
4+1/6 4+2/6 4+3/6 4+4/6 4+5/6 ...
5+1/6 5+2/6 5+3/6 5+4/6 5+5/6 ...
6+1/6 6+2/6 6+3/6 6+4/6 6+5/6 ...
7+1/6 7+2/6 7+3/6 7+4/6 7+5/6 ...
8+1/6 8+2/6 8+3/6 8+4/6 8+5/6 ...
9+1/6 9+2/6 9+3/6 9+4/6 9+5/6 ...
10+1/6 10+2/6 10+3/6 10+4/6 10+5/6 ...
11+1/6 11+2/6 11+3/6 11+4/6 11+5/6 ...
12+1/6 12+2/6 12+3/6 12+4/6 12+5/6 ...
13+1/6 13+2/6 13+3/6 13+4/6 13+5/6 ...
14+1/6 14+2/6 14+3/6 14+4/6 14+5/6 ...
15+1/6 15+2/6 15+3/6 15+4/6 15+5/6 ...
16+1/6 16+2/6 16+3/6 16+4/6 16+5/6 ...
17+1/6 17+2/6 17+3/6 17+4/6 17+5/6 ...
18+1/6 18+2/6 18+3/6 18+4/6 18+5/6 ...
19+1/6 19+2/6 19+3/6 19+4/6 19+5/6 ...
20+1/6 20+2/6 20+3/6 20+4/6 20+5/6 ...
21+1/6 21+2/6 21+3/6 21+4/6 21+5/6 ...
22+1/6 22+2/6 22+3/6 22+4/6 22+5/6 ...
23+1/6 23+2/6 23+3/6 23+4/6 23+5/6 ...

```

```

24+1/6 24+2/6 24+3/6 24+4/6 24+5/6 ...
25+1/6 25+2/6 25+3/6 25+4/6 25+5/6 ...
26+1/6 26+2/6 26+3/6 26+4/6 26+5/6 ...
27+1/6 27+2/6 27+3/6 27+4/6 27+5/6 ...
28+1/6 28+2/6 28+3/6 28+4/6 28+5/6 ...
29+1/6 29+2/6 29+3/6 29+4/6 29+5/6 ...
30+1/6 30+2/6 30+3/6 30+4/6 30+5/6 ...
31+1/6 31+2/6 31+3/6 31+4/6 31+5/6 ...
32+1/6 32+2/6 32+3/6 32+4/6 32+5/6 ...
33+1/6 33+2/6 33+3/6 33+4/6 33+5/6 ...
34+1/6 34+2/6 34+3/6 34+4/6 34+5/6 ...
35+1/6 35+2/6 35+3/6 35+4/6 35+5/6 ...
36+1/6 36+2/6 36+3/6 36+4/6 36+5/6 ...
37+1/6 37+2/6 37+3/6 37+4/6 37+5/6 ...
38+1/6 38+2/6 38+3/6 38+4/6 38+5/6 ...
39+1/6 39+2/6 39+3/6 39+4/6 39+5/6 ...
40+1/6 40+2/6 40+3/6 40+4/6 40+5/6 ...
41+1/6 41+2/6 41+3/6 41+4/6 41+5/6 ...
42+1/6 42+2/6 42+3/6 42+4/6 42+5/6 ...
43+1/6 43+2/6 43+3/6 43+4/6 43+5/6 ...
44+1/6 44+2/6 44+3/6 44+4/6 44+5/6 ...
45+1/6 45+2/6 45+3/6 45+4/6 45+5/6 ...
46+1/6 46+2/6 46+3/6 46+4/6 46+5/6 ...
47+1/6 47+2/6 47+3/6 47+4/6 47+5/6 ...
48+1/6 48+2/6 48+3/6 48+4/6 48+5/6 ...
49+1/6 49+2/6 49+3/6 49+4/6 49+5/6 ...
50+1/6 50+2/6 50+3/6 50+4/6 50+5/6 ...
51+1/6 51+2/6 51+3/6 51+4/6 51+5/6 ...
52+1/6 52+2/6 52+3/6 52+4/6 52+5/6 ...
53+1/6 53+2/6 53+3/6 53+4/6 53+5/6 ...
54+1/6 54+2/6 54+3/6 54+4/6 54+5/6 ...
55+1/6 55+2/6 55+3/6 55+4/6 55+5/6 ...
56+1/6 56+2/6 56+3/6 56+4/6 56+5/6 ...
57+1/6 57+2/6 57+3/6 57+4/6 57+5/6 ...
58+1/6 58+2/6 58+3/6 58+4/6 58+5/6 ...
59+1/6 59+2/6 59+3/6 59+4/6 59+5/6 ...
60+1/6 60+2/6 60+3/6 60+4/6 60+5/6]';
block.OutputPort(1).Data = interp1(x,block.InputPort(1).Data,xi,'spline');
%endfunction

```

S_OFDMP76

```

function msfcn_S_OFDM(block)
% Level-2 M file S-Function for times two demo.
% Copyright 1990-2004 The MathWorks, Inc.
% $Revision: 1.1.6.1 $
    setup(block);
%endfunction
function setup(block)
    %% Register number of input and output ports
    block.NumInputPorts = 1;
    block.NumOutputPorts = 1;
    %% Setup functional port properties to dynamically
    %% inherited.
    block.SetPreCompInPortInfoToDynamic;
    block.SetPreCompOutPortInfoToDynamic;
    % Override input port properties
    block.InputPort(1).DatatypeID = 0; % double
    block.InputPort(1).Complexity = 'complex';
    % Override output port properties
    block.OutputPort(1).DatatypeID = 0; % double

```

```

block.OutputPort(1).Complexity = 'complex';
block.InputPort(1).DirectFeedthrough = true;
%% SetInputPortDimensions:
%%   Functionality      : Check and set input and optionally output
%%                        port dimensions
%%   C-Mex counterpart: mdlSetInputPortDimensionInfo
%%
block.RegBlockMethod('SetInputPortDimensions', @SetInpPortDims);
%%
%% SetOutputPortDimensions:
%%   Functionality      : Check and set output and optionally input
%%                        port dimensions
%%   C-Mex counterpart: mdlSetOutputPortDimensionInfo
%%
block.RegBlockMethod('SetOutputPortDimensions', @SetOutPortDims);
%% Set block sample time to inherited
block.SampleTimes = [-1 0];
%% Run accelerator on TLC
block.SetAccelRunOnTLC(true);
%% Register methods
block.RegBlockMethod('Outputs', @Output);
%endfunction
function SetInpPortDims(block, idx, di)
    block.InputPort(1).Dimensions = 76;
    block.OutputPort(1).Dimensions = 300;
%endfunction
function SetOutPortDims(block, idx, di)
    block.OutputPort(1).Dimensions = 300;
    block.InputPort(1).Dimensions = 76;
%endfunction
function Output(block)
x=[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27
28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52
53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76]';
xi=[1+1/5 1+2/5 1+3/5 1+4/5 ...
2+1/5 2+2/5 2+3/5 2+4/5 ...
3+1/5 3+2/5 3+3/5 3+4/5 ...
4+1/5 4+2/5 4+3/5 4+4/5 ...
5+1/5 5+2/5 5+3/5 5+4/5 ...
6+1/5 6+2/5 6+3/5 6+4/5 ...
7+1/5 7+2/5 7+3/5 7+4/5 ...
8+1/5 8+2/5 8+3/5 8+4/5 ...
9+1/5 9+2/5 9+3/5 9+4/5 ...
10+1/5 10+2/5 10+3/5 10+4/5 ...
11+1/5 11+2/5 11+3/5 11+4/5 ...
12+1/5 12+2/5 12+3/5 12+4/5 ...
13+1/5 13+2/5 13+3/5 13+4/5 ...
14+1/5 14+2/5 14+3/5 14+4/5 ...
15+1/5 15+2/5 15+3/5 15+4/5 ...
16+1/5 16+2/5 16+3/5 16+4/5 ...
17+1/5 17+2/5 17+3/5 17+4/5 ...
18+1/5 18+2/5 18+3/5 18+4/5 ...
19+1/5 19+2/5 19+3/5 19+4/5 ...
20+1/5 20+2/5 20+3/5 20+4/5 ...
21+1/5 21+2/5 21+3/5 21+4/5 ...
22+1/5 22+2/5 22+3/5 22+4/5 ...
23+1/5 23+2/5 23+3/5 23+4/5 ...
24+1/5 24+2/5 24+3/5 24+4/5 ...
25+1/5 25+2/5 25+3/5 25+4/5 ...
26+1/5 26+2/5 26+3/5 26+4/5 ...
27+1/5 27+2/5 27+3/5 27+4/5 ...

```



```

28+1/5 28+2/5 28+3/5 28+4/5 ...
29+1/5 29+2/5 29+3/5 29+4/5 ...
30+1/5 30+2/5 30+3/5 30+4/5 ...
31+1/5 31+2/5 31+3/5 31+4/5 ...
32+1/5 32+2/5 32+3/5 32+4/5 ...
33+1/5 33+2/5 33+3/5 33+4/5 ...
34+1/5 34+2/5 34+3/5 34+4/5 ...
35+1/5 35+2/5 35+3/5 35+4/5 ...
36+1/5 36+2/5 36+3/5 36+4/5 ...
37+1/5 37+2/5 37+3/5 37+4/5 ...
38+1/5 38+2/5 38+3/5 38+4/5 ...
39+1/5 39+2/5 39+3/5 39+4/5 ...
40+1/5 40+2/5 40+3/5 40+4/5 ...
41+1/5 41+2/5 41+3/5 41+4/5 ...
42+1/5 42+2/5 42+3/5 42+4/5 ...
43+1/5 43+2/5 43+3/5 43+4/5 ...
44+1/5 44+2/5 44+3/5 44+4/5 ...
45+1/5 45+2/5 45+3/5 45+4/5 ...
46+1/5 46+2/5 46+3/5 46+4/5 ...
47+1/5 47+2/5 47+3/5 47+4/5 ...
48+1/5 48+2/5 48+3/5 48+4/5 ...
49+1/5 49+2/5 49+3/5 49+4/5 ...
50+1/5 50+2/5 50+3/5 50+4/5 ...
51+1/5 51+2/5 51+3/5 51+4/5 ...
52+1/5 52+2/5 52+3/5 52+4/5 ...
53+1/5 53+2/5 53+3/5 53+4/5 ...
54+1/5 54+2/5 54+3/5 54+4/5 ...
55+1/5 55+2/5 55+3/5 55+4/5 ...
56+1/5 56+2/5 56+3/5 56+4/5 ...
57+1/5 57+2/5 57+3/5 57+4/5 ...
58+1/5 58+2/5 58+3/5 58+4/5 ...
59+1/5 59+2/5 59+3/5 59+4/5 ...
60+1/5 60+2/5 60+3/5 60+4/5 ...
61+1/5 61+2/5 61+3/5 61+4/5 ...
62+1/5 62+2/5 62+3/5 62+4/5 ...
63+1/5 63+2/5 63+3/5 63+4/5 ...
64+1/5 64+2/5 64+3/5 64+4/5 ...
65+1/5 65+2/5 65+3/5 65+4/5 ...
66+1/5 66+2/5 66+3/5 66+4/5 ...
67+1/5 67+2/5 67+3/5 67+4/5 ...
68+1/5 68+2/5 68+3/5 68+4/5 ...
69+1/5 69+2/5 69+3/5 69+4/5 ...
70+1/5 70+2/5 70+3/5 70+4/5 ...
71+1/5 71+2/5 71+3/5 71+4/5 ...
72+1/5 72+2/5 72+3/5 72+4/5 ...
73+1/5 73+2/5 73+3/5 73+4/5 ...
74+1/5 74+2/5 74+3/5 74+4/5 ...
75+1/5 75+2/5 75+3/5 75+4/5]';
block.OutputPort(1).Data = interp1(x,block.InputPort(1).Data,xi,'spline');
%endfunction

```

S_OFDMP101

```

function msfcn_S_OFDM(block)
% Level-2 M file S-Function for times two demo.
% Copyright 1990-2004 The MathWorks, Inc.
% $Revision: 1.1.6.1 $
    setup(block);
%endfunction
function setup(block)
    %% Register number of input and output ports

```

```

block.NumInputPorts = 1;
block.NumOutputPorts = 1;
%% Setup functional port properties to dynamically
%% inherited.
block.SetPreCompInpPortInfoToDynamic;
block.SetPreCompOutPortInfoToDynamic;
% Override input port properties
block.InputPort(1).DatatypeID = 0; % double
block.InputPort(1).Complexity = 'complex';
% Override output port properties
block.OutputPort(1).DatatypeID = 0; % double
block.OutputPort(1).Complexity = 'complex';
block.InputPort(1).DirectFeedthrough = true;
%% SetInputPortDimensions:
%%   Functionality      : Check and set input and optionally output
%%                        port dimensions
%%   C-Mex counterpart: mdlSetInputPortDimensionInfo
%%
block.RegBlockMethod('SetInputPortDimensions', @SetInpPortDims);
%%
%% SetOutputPortDimensions:
%%   Functionality      : Check and set output and optionally input
%%                        port dimensions
%%   C-Mex counterpart: mdlSetOutputPortDimensionInfo
%%
block.RegBlockMethod('SetOutputPortDimensions', @SetOutPortDims);
%% Set block sample time to inherited
block.SampleTimes = [-1 0];
%% Run accelerator on TLC
block.SetAccelRunOnTLC(true);
%% Register methods
block.RegBlockMethod('Outputs', @Output);
%endfunction
function SetInpPortDims(block, idx, di)
    block.InputPort(1).Dimensions = 101;
    block.OutputPort(1).Dimensions = 300;
%endfunction
function SetOutPortDims(block, idx, di)
    block.OutputPort(1).Dimensions = 300;
    block.InputPort(1).Dimensions = 101;
%endfunction
function Output(block)
x=[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27
28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52
53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77
78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 ...
    101]';
xi=[1+1/4 1+2/4 1+3/4 ...
2+1/4 2+2/4 2+3/4 ...
3+1/4 3+2/4 3+3/4 ...
4+1/4 4+2/4 4+3/4 ...
5+1/4 5+2/4 5+3/4 ...
6+1/4 6+2/4 6+3/4 ...
7+1/4 7+2/4 7+3/4 ...
8+1/4 8+2/4 8+3/4 ...
9+1/4 9+2/4 9+3/4 ...
10+1/4 10+2/4 10+3/4 ...
11+1/4 11+2/4 11+3/4 ...
12+1/4 12+2/4 12+3/4 ...
13+1/4 13+2/4 13+3/4 ...
14+1/4 14+2/4 14+3/4 ...

```

15+1/4	15+2/4	15+3/4	...
16+1/4	16+2/4	16+3/4	...
17+1/4	17+2/4	17+3/4	...
18+1/4	18+2/4	18+3/4	...
19+1/4	19+2/4	19+3/4	...
20+1/4	20+2/4	20+3/4	...
21+1/4	21+2/4	21+3/4	...
22+1/4	22+2/4	22+3/4	...
23+1/4	23+2/4	23+3/4	...
24+1/4	24+2/4	24+3/4	...
25+1/4	25+2/4	25+3/4	...
26+1/4	26+2/4	26+3/4	...
27+1/4	27+2/4	27+3/4	...
28+1/4	28+2/4	28+3/4	...
29+1/4	29+2/4	29+3/4	...
30+1/4	30+2/4	30+3/4	...
31+1/4	31+2/4	31+3/4	...
32+1/4	32+2/4	32+3/4	...
33+1/4	33+2/4	33+3/4	...
34+1/4	34+2/4	34+3/4	...
35+1/4	35+2/4	35+3/4	...
36+1/4	36+2/4	36+3/4	...
37+1/4	37+2/4	37+3/4	...
38+1/4	38+2/4	38+3/4	...
39+1/4	39+2/4	39+3/4	...
40+1/4	40+2/4	40+3/4	...
41+1/4	41+2/4	41+3/4	...
42+1/4	42+2/4	42+3/4	...
43+1/4	43+2/4	43+3/4	...
44+1/4	44+2/4	44+3/4	...
45+1/4	45+2/4	45+3/4	...
46+1/4	46+2/4	46+3/4	...
47+1/4	47+2/4	47+3/4	...
48+1/4	48+2/4	48+3/4	...
49+1/4	49+2/4	49+3/4	...
50+1/4	50+2/4	50+3/4	...
51+1/4	51+2/4	51+3/4	...
52+1/4	52+2/4	52+3/4	...
53+1/4	53+2/4	53+3/4	...
54+1/4	54+2/4	54+3/4	...
55+1/4	55+2/4	55+3/4	...
56+1/4	56+2/4	56+3/4	...
57+1/4	57+2/4	57+3/4	...
58+1/4	58+2/4	58+3/4	...
59+1/4	59+2/4	59+3/4	...
60+1/4	60+2/4	60+3/4	...
61+1/4	61+2/4	61+3/4	...
62+1/4	62+2/4	62+3/4	...
63+1/4	63+2/4	63+3/4	...
64+1/4	64+2/4	64+3/4	...
65+1/4	65+2/4	65+3/4	...
66+1/4	66+2/4	66+3/4	...
67+1/4	67+2/4	67+3/4	...
68+1/4	68+2/4	68+3/4	...
69+1/4	69+2/4	69+3/4	...
70+1/4	70+2/4	70+3/4	...
71+1/4	71+2/4	71+3/4	...
72+1/4	72+2/4	72+3/4	...
73+1/4	73+2/4	73+3/4	...
74+1/4	74+2/4	74+3/4	...
75+1/4	75+2/4	75+3/4	...

```

76+1/4 76+2/4 76+3/4 ...
77+1/4 77+2/4 77+3/4 ...
78+1/4 78+2/4 78+3/4 ...
79+1/4 79+2/4 79+3/4 ...
80+1/4 80+2/4 80+3/4 ...
81+1/4 81+2/4 81+3/4 ...
82+1/4 82+2/4 82+3/4 ...
83+1/4 83+2/4 83+3/4 ...
84+1/4 84+2/4 84+3/4 ...
85+1/4 85+2/4 85+3/4 ...
86+1/4 86+2/4 86+3/4 ...
87+1/4 87+2/4 87+3/4 ...
88+1/4 88+2/4 88+3/4 ...
89+1/4 89+2/4 89+3/4 ...
90+1/4 90+2/4 90+3/4 ...
91+1/4 91+2/4 91+3/4 ...
92+1/4 92+2/4 92+3/4 ...
93+1/4 93+2/4 93+3/4 ...
94+1/4 94+2/4 94+3/4 ...
95+1/4 95+2/4 95+3/4 ...
96+1/4 96+2/4 96+3/4 ...
97+1/4 97+2/4 97+3/4 ...
98+1/4 98+2/4 98+3/4 ...
99+1/4 99+2/4 99+3/4 ...
100+1/4 100+2/4 100+3/4]';
block.OutputPort(1).Data = interp1(x,block.InputPort(1).Data,xi,'spline');
%endfunction

```

S_OFDMP151

```

function msfcn_S_OFDM(block)
% Level-2 M file S-Function for times two demo.
% Copyright 1990-2004 The MathWorks, Inc.
% $Revision: 1.1.6.1 $
    setup(block);
%endfunction
function setup(block)
    %% Register number of input and output ports
    block.NumInputPorts = 1;
    block.NumOutputPorts = 1;
    %% Setup functional port properties to dynamically
    %% inherited.
    block.SetPreCompInpPortInfoToDynamic;
    block.SetPreCompOutPortInfoToDynamic;
    % Override input port properties
    block.InputPort(1).DatatypeID = 0; % double
    block.InputPort(1).Complexity = 'complex';
    % Override output port properties
    block.OutputPort(1).DatatypeID = 0; % double
    block.OutputPort(1).Complexity = 'complex';
    block.InputPort(1).DirectFeedthrough = true;
    %% SetInputPortDimensions:
    %% Functionality : Check and set input and optionally output
    %% port dimensions
    %% C-Mex counterpart: mdlSetInputPortDimensionInfo
    %%
    block.RegBlockMethod('SetInputPortDimensions', @SetInpPortDims);
    %%
    %% SetOutputPortDimensions:
    %% Functionality : Check and set output and optionally input
    %% port dimensions

```

```

%% C-Mex counterpart: mdlSetOutputPortDimensionInfo
%%
block.RegBlockMethod('SetOutputPortDimensions', @SetOutPortDims);
%% Set block sample time to inherited
block.SampleTimes = [-1 0];
%% Run accelerator on TLC
block.SetAccelRunOnTLC(true);
%% Register methods
block.RegBlockMethod('Outputs', @Output);
%endfunction
function SetInpPortDims(block, idx, di)
    block.InputPort(1).Dimensions = 151;
    block.OutputPort(1).Dimensions = 300;
%endfunction
function SetOutPortDims(block, idx, di)
    block.OutputPort(1).Dimensions = 300;
    block.InputPort(1).Dimensions = 151;
%endfunction
function Output(block)
x=[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27
28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52
53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77
78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 ...
    101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118
119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137
138 139 140 141 142 143 144 145 146 147 148 149 150 151]';
xi=[1+1/3 1+2/3 ...
2+1/3 2+2/3 ...
3+1/3 3+2/3 ...
4+1/3 4+2/3 ...
5+1/3 5+2/3 ...
6+1/3 6+2/3 ...
7+1/3 7+2/3 ...
8+1/3 8+2/3 ...
9+1/3 9+2/3 ...
10+1/3 10+2/3 ...
11+1/3 11+2/3 ...
12+1/3 12+2/3 ...
13+1/3 13+2/3 ...
14+1/3 14+2/3 ...
15+1/3 15+2/3 ...
16+1/3 16+2/3 ...
17+1/3 17+2/3 ...
18+1/3 18+2/3 ...
19+1/3 19+2/3 ...
20+1/3 20+2/3 ...
21+1/3 21+2/3 ...
22+1/3 22+2/3 ...
23+1/3 23+2/3 ...
24+1/3 24+2/3 ...
25+1/3 25+2/3 ...
26+1/3 26+2/3 ...
27+1/3 27+2/3 ...
28+1/3 28+2/3 ...
29+1/3 29+2/3 ...
30+1/3 30+2/3 ...
31+1/3 31+2/3 ...
32+1/3 32+2/3 ...
33+1/3 33+2/3 ...
34+1/3 34+2/3 ...
35+1/3 35+2/3 ...

```

36+1/3 36+2/3 ...
37+1/3 37+2/3 ...
38+1/3 38+2/3 ...
39+1/3 39+2/3 ...
40+1/3 40+2/3 ...
41+1/3 41+2/3 ...
42+1/3 42+2/3 ...
43+1/3 43+2/3 ...
44+1/3 44+2/3 ...
45+1/3 45+2/3 ...
46+1/3 46+2/3 ...
47+1/3 47+2/3 ...
48+1/3 48+2/3 ...
49+1/3 49+2/3 ...
50+1/3 50+2/3 ...
51+1/3 51+2/3 ...
52+1/3 52+2/3 ...
53+1/3 53+2/3 ...
54+1/3 54+2/3 ...
55+1/3 55+2/3 ...
56+1/3 56+2/3 ...
57+1/3 57+2/3 ...
58+1/3 58+2/3 ...
59+1/3 59+2/3 ...
60+1/3 60+2/3 ...
61+1/3 61+2/3 ...
62+1/3 62+2/3 ...
63+1/3 63+2/3 ...
64+1/3 64+2/3 ...
65+1/3 65+2/3 ...
66+1/3 66+2/3 ...
67+1/3 67+2/3 ...
68+1/3 68+2/3 ...
69+1/3 69+2/3 ...
70+1/3 70+2/3 ...
71+1/3 71+2/3 ...
72+1/3 72+2/3 ...
73+1/3 73+2/3 ...
74+1/3 74+2/3 ...
75+1/3 75+2/3 ...
76+1/3 76+2/3 ...
77+1/3 77+2/3 ...
78+1/3 78+2/3 ...
79+1/3 79+2/3 ...
80+1/3 80+2/3 ...
81+1/3 81+2/3 ...
82+1/3 82+2/3 ...
83+1/3 83+2/3 ...
84+1/3 84+2/3 ...
85+1/3 85+2/3 ...
86+1/3 86+2/3 ...
87+1/3 87+2/3 ...
88+1/3 88+2/3 ...
89+1/3 89+2/3 ...
90+1/3 90+2/3 ...
91+1/3 91+2/3 ...
92+1/3 92+2/3 ...
93+1/3 93+2/3 ...
94+1/3 94+2/3 ...
95+1/3 95+2/3 ...
96+1/3 96+2/3 ...

```

97+1/3 97+2/3 ...
98+1/3 98+2/3 ...
99+1/3 99+2/3 ...
100+1/3 100+2/3 ...
101+1/3 101+2/3 ...
102+1/3 102+2/3 ...
103+1/3 103+2/3 ...
104+1/3 104+2/3 ...
105+1/3 105+2/3 ...
106+1/3 106+2/3 ...
107+1/3 107+2/3 ...
108+1/3 108+2/3 ...
109+1/3 109+2/3 ...
110+1/3 110+2/3 ...
111+1/3 111+2/3 ...
112+1/3 112+2/3 ...
113+1/3 113+2/3 ...
114+1/3 114+2/3 ...
115+1/3 115+2/3 ...
116+1/3 116+2/3 ...
117+1/3 117+2/3 ...
118+1/3 118+2/3 ...
119+1/3 119+2/3 ...
120+1/3 120+2/3 ...
121+1/3 121+2/3 ...
122+1/3 122+2/3 ...
123+1/3 123+2/3 ...
124+1/3 124+2/3 ...
125+1/3 125+2/3 ...
126+1/3 126+2/3 ...
127+1/3 127+2/3 ...
128+1/3 128+2/3 ...
129+1/3 129+2/3 ...
130+1/3 130+2/3 ...
131+1/3 131+2/3 ...
132+1/3 132+2/3 ...
133+1/3 133+2/3 ...
134+1/3 134+2/3 ...
135+1/3 135+2/3 ...
136+1/3 136+2/3 ...
137+1/3 137+2/3 ...
138+1/3 138+2/3 ...
139+1/3 139+2/3 ...
140+1/3 140+2/3 ...
141+1/3 141+2/3 ...
142+1/3 142+2/3 ...
143+1/3 143+2/3 ...
144+1/3 144+2/3 ...
145+1/3 145+2/3 ...
146+1/3 146+2/3 ...
147+1/3 147+2/3 ...
148+1/3 148+2/3 ...
149+1/3 149+2/3 ...
150+1/3 150+2/3]';
block.OutputPort(1).Data = interp1(x,block.InputPort(1).Data,xi,'spline');
%endfunction

```

OFDM_BERxEbNo_2x6

```

%bloco=50; %Bernoulli Binary Generator: Samples per Frame = bloco
%repvet=[];

```

```

%repvet=[1 31 61 91 120 135 149];
%for i=1:length(repvet);
%    rep=repvet(i); %Demux: Number of outputs = [3*bloco-rep,rep]
%
%set(0,'DefaultAxesLineStyleOrder',{'-','--',':', '-.'})
set(0,'DefaultAxesColorOrder',[1 0 0])
%
% 2 PILOTOS
EbNoVec=5.0:1.0:15.0;
BER_Vec=[];
EBNOX=[];
for n=1:length(EbNoVec);
    EbNo=EbNoVec(n);
    sim('OFDMP2');
    if erro > 0.00000001;
        BER_Vec(n,:)=erro;
        EBNOX(n,:)=EbNo;
    else
        n=length(EbNoVec);
    end;
end;
semilogy(EBNOX(:,1),BER_Vec(:,1),'+');
%semilogy(EBNOX(:,1),BER_Vec(:,1));
berfit(EBNOX(:,1),BER_Vec(:,1));
hold on;
%end;
% 3 PILOTOS
EbNoVec=5.0:1.0:15.0;
BER_Vec=[];
EBNOX=[];
for n=1:length(EbNoVec);
    EbNo=EbNoVec(n);
    sim('OFDMP3');
    if erro > 0.00000001;
        BER_Vec(n,:)=erro;
        EBNOX(n,:)=EbNo;
    else
        n=length(EbNoVec);
    end;
end;
semilogy(EBNOX(:,1),BER_Vec(:,1),'+');
%semilogy(EBNOX(:,1),BER_Vec(:,1));
berfit(EBNOX(:,1),BER_Vec(:,1));
hold on;
%
% 4 PILOTOS
EbNoVec=5.0:1.0:15.0;
BER_Vec=[];
EBNOX=[];
for n=1:length(EbNoVec);
    EbNo=EbNoVec(n);
    sim('OFDMP4');
    if erro > 0.00000001;
        BER_Vec(n,:)=erro;
        EBNOX(n,:)=EbNo;
    else
        n=length(EbNoVec);
    end;
end;
semilogy(EBNOX(:,1),BER_Vec(:,1),'+');
%semilogy(EBNOX(:,1),BER_Vec(:,1));

```



```

berfit(EBNOX(:,1),BER_Vec(:,1));
hold on;
%
% 5 PILOTOS
EbNoVec=5.0:1.0:15.0;
BER_Vec=[];
EBNOX=[];
for n=1:length(EbNoVec);
    EbNo=EbNoVec(n);
    sim('OFDMP5');
    if erro > 0.00000001;
        BER_Vec(n,:)=erro;
        EBNOX(n,:)=EbNo;
    else
        n=length(EbNoVec);
    end;
end;
semilogy(EBNOX(:,1),BER_Vec(:,1),'+');
%semilogy(EBNOX(:,1),BER_Vec(:,1));
berfit(EBNOX(:,1),BER_Vec(:,1));
hold on;
%
% 6 PILOTOS
EbNoVec=5.0:1.0:15.0;
BER_Vec=[];
EBNOX=[];
for n=1:length(EbNoVec);
    EbNo=EbNoVec(n);
    sim('OFDMP6');
    if erro > 0.00000001;
        BER_Vec(n,:)=erro;
        EBNOX(n,:)=EbNo;
    else
        n=length(EbNoVec);
    end;
end;
semilogy(EBNOX(:,1),BER_Vec(:,1),'+');
%semilogy(EBNOX(:,1),BER_Vec(:,1));
berfit(EBNOX(:,1),BER_Vec(:,1));
hold on;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
xlabel('Eb/No (dB)');
ylabel('BER');
%title('Plataforma OFDM');
%legend('DeltaN/N = 0.0;0.2;0.4;0.6;0.8;0.9;1.0 escrever taxa de erro (%)');
hold off;
%
%set(0,'DefaultAxesLineStyleOrder','remove')
set(0,'DefaultAxesColorOrder','remove')

```

OFDM BERxEbNo 7

[illegible]

```

% 7 PILOTOS
EbNoVec=5.0:1.0:15.0;
BER_Vec=[];
EBNOX=[];
for n=1:length(EbNoVec);
    EbNo=EbNoVec(n);
    sim('OFDMP7');
    if erro > 0.00000001;
        BER_Vec(n,:)=erro;
        EBNOX(n,:)=EbNo;
    else
        n=length(EbNoVec);
    end;
end;
semilogy(EBNOX(:,1),BER_Vec(:,1),'+');
%semilogy(EBNOX(:,1),BER_Vec(:,1));
berfit(EBNOX(:,1),BER_Vec(:,1));
hold on;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
xlabel('Eb/No (dB)');
ylabel('BER');
%title('Plataforma OFDM');
%legend('DeltaN/N = 0.0;0.2;0.4;0.6;0.8;0.9;1.0 escrever taxa de erro (%)');
hold off;
%
%set(0,'DefaultAxesLineStyleOrder','remove')
set(0,'DefaultAxesColorOrder','remove')

```

OFDM_BERxEbNo_11x31

```

%bloco=50; %Bernoulli Binary Generator: Samples per Frame = bloco
%repvet=[];
%repvet=[1 31 61 91 120 135 149];
%for i=1:length(repvet);
%    rep=repvet(i); %Demux: Number of outputs = [3*bloco-rep,rep]
%
%set(0,'DefaultAxesLineStyleOrder',{'-','--',':', '-.'})
set(0,'DefaultAxesColorOrder',[1 0 0])
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 11 PILOTOS
EbNoVec=5.0:1.0:15.0;
BER_Vec=[];
EBNOX=[];
for n=1:length(EbNoVec);
    EbNo=EbNoVec(n);
    sim('OFDMP11');
    if erro > 0.00000001;
        BER_Vec(n,:)=erro;
        EBNOX(n,:)=EbNo;
    else
        n=length(EbNoVec);
    end;
end;
semilogy(EBNOX(:,1),BER_Vec(:,1),'+');
%semilogy(EBNOX(:,1),BER_Vec(:,1));
berfit(EBNOX(:,1),BER_Vec(:,1));
hold on;
%
% 13 PILOTOS
EbNoVec=5.0:1.0:15.0;

```

```

BER_Vec=[];
EBNOX=[];
for n=1:length(EbNoVec);
    EbNo=EbNoVec(n);
    sim('OFDMP13');
    if erro > 0.00000001;
        BER_Vec(n,:)=erro;
        EBNOX(n,:)=EbNo;
    else
        n=length(EbNoVec);
    end;
end;
semilogy(EBNOX(:,1),BER_Vec(:,1),'+');
%semilogy(EBNOX(:,1),BER_Vec(:,1));
berfit(EBNOX(:,1),BER_Vec(:,1));
hold on;
%
% 16 PILOTOS
EbNoVec=5.0:1.0:15.0;
BER_Vec=[];
EBNOX=[];
for n=1:length(EbNoVec);
    EbNo=EbNoVec(n);
    sim('OFDMP16');
    if erro > 0.00000001;
        BER_Vec(n,:)=erro;
        EBNOX(n,:)=EbNo;
    else
        n=length(EbNoVec);
    end;
end;
semilogy(EBNOX(:,1),BER_Vec(:,1),'+');
%semilogy(EBNOX(:,1),BER_Vec(:,1));
berfit(EBNOX(:,1),BER_Vec(:,1));
hold on;
%
% 21 PILOTOS
EbNoVec=5.0:1.0:15.0;
BER_Vec=[];
EBNOX=[];
for n=1:length(EbNoVec);
    EbNo=EbNoVec(n);
    sim('OFDMP21');
    if erro > 0.00000001;
        BER_Vec(n,:)=erro;
        EBNOX(n,:)=EbNo;
    else
        n=length(EbNoVec);
    end;
end;
semilogy(EBNOX(:,1),BER_Vec(:,1),'+');
%semilogy(EBNOX(:,1),BER_Vec(:,1));
berfit(EBNOX(:,1),BER_Vec(:,1));
hold on;
%
% 26 PILOTOS
EbNoVec=5.0:1.0:15.0;
BER_Vec=[];
EBNOX=[];
for n=1:length(EbNoVec);
    EbNo=EbNoVec(n);

```

```

sim('OFDMP26');
if erro > 0.00000001;
BER_Vec(n,:)=erro;
EBNOX(n,:)=EbNo;
else
n=length(EbNoVec);
end;
end;
semilogy(EBNOX(:,1),BER_Vec(:,1),'+');
%semilogy(EBNOX(:,1),BER_Vec(:,1));
berfit(EBNOX(:,1),BER_Vec(:,1));
hold on;
%
% 31 PILOTOS
EbNoVec=5.0:1.0:15.0;
BER_Vec=[];
EBNOX=[];
for n=1:length(EbNoVec);
EbNo=EbNoVec(n);
sim('OFDMP31');
if erro > 0.00000001;
BER_Vec(n,:)=erro;
EBNOX(n,:)=EbNo;
else
n=length(EbNoVec);
end;
end;
semilogy(EBNOX(:,1),BER_Vec(:,1),'+');
%semilogy(EBNOX(:,1),BER_Vec(:,1));
berfit(EBNOX(:,1),BER_Vec(:,1));
hold on;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
xlabel('Eb/No (dB)');
ylabel('BER');
%title('Plataforma OFDM');
%legend('DeltaN/N = 0.0;0.2;0.4;0.6;0.8;0.9;1.0 escrever taxa de erro (%)');
hold off;
%
%set(0,'DefaultAxesLineStyleOrder','remove')
set(0,'DefaultAxesColorOrder','remove')

```

OFDM_BERxEbNo_51x151

```

%bloco=50; %Bernoulli Binary Generator: Samples per Frame = bloco
%repvet=[];
%repvet=[1 31 61 91 120 135 149];
%for i=1:length(repvet);
%    rep=repvet(i); %Demux: Number of outputs = [3*bloco-rep,rep]
%
%set(0,'DefaultAxesLineStyleOrder',{'-','--',':', '-.'})
set(0,'DefaultAxesColorOrder',[1 0 0])
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 51 PILOTOS
EbNoVec=5.0:1.0:15.0;
BER_Vec=[];
EBNOX=[];
for n=1:length(EbNoVec);
EbNo=EbNoVec(n);
sim('OFDMP51');
if erro > 0.00000001;

```

```

        BER_Vec(n,:)=erro;
        EBNOX(n,:)=EbNo;
    else
        n=length(EbNoVec);
    end;
end;
semilogy(EBNOX(:,1),BER_Vec(:,1),'+');
%semilogy(EBNOX(:,1),BER_Vec(:,1));
berfit(EBNOX(:,1),BER_Vec(:,1));
hold on;
%
% 61 PILOTOS
EbNoVec=5.0:1.0:15.0;
BER_Vec=[];
EBNOX=[];
for n=1:length(EbNoVec);
    EbNo=EbNoVec(n);
    sim('OFDMP61');
    if erro > 0.00000001;
        BER_Vec(n,:)=erro;
        EBNOX(n,:)=EbNo;
    else
        n=length(EbNoVec);
    end;
end;
semilogy(EBNOX(:,1),BER_Vec(:,1),'+');
%semilogy(EBNOX(:,1),BER_Vec(:,1));
berfit(EBNOX(:,1),BER_Vec(:,1));
hold on;
%
% 76 PILOTOS
EbNoVec=5.0:1.0:15.0;
BER_Vec=[];
EBNOX=[];
for n=1:length(EbNoVec);
    EbNo=EbNoVec(n);
    sim('OFDMP76');
    if erro > 0.00000001;
        BER_Vec(n,:)=erro;
        EBNOX(n,:)=EbNo;
    else
        n=length(EbNoVec);
    end;
end;
semilogy(EBNOX(:,1),BER_Vec(:,1),'+');
%semilogy(EBNOX(:,1),BER_Vec(:,1));
berfit(EBNOX(:,1),BER_Vec(:,1));
hold on;
%
% 101 PILOTOS
EbNoVec=5.0:1.0:15.0;
BER_Vec=[];
EBNOX=[];
for n=1:length(EbNoVec);
    EbNo=EbNoVec(n);
    sim('OFDMP101');
    if erro > 0.00000001;
        BER_Vec(n,:)=erro;
        EBNOX(n,:)=EbNo;
    else
        n=length(EbNoVec);
    end;
end;

```

```

        end;
    end;
    semilogy(EBNOX(:,1),BER_Vec(:,1),'+');
    %semilogy(EBNOX(:,1),BER_Vec(:,1));
    berfit(EBNOX(:,1),BER_Vec(:,1));
    hold on;
    %
    % 151 PILOTOS
    EbNoVec=5.0:1.0:15.0;
    BER_Vec=[];
    EBNOX=[];
    for n=1:length(EbNoVec);
        EbNo=EbNoVec(n);
        sim('OFDMP151');
        if erro > 0.00000001;
            BER_Vec(n,:)=erro;
            EBNOX(n,:)=EbNo;
        else
            n=length(EbNoVec);
        end;
    end;
    semilogy(EBNOX(:,1),BER_Vec(:,1),'+');
    %semilogy(EBNOX(:,1),BER_Vec(:,1));
    berfit(EBNOX(:,1),BER_Vec(:,1));
    hold on;
    %
    xlabel('Eb/No (dB)');
    ylabel('BER');
    %title('Plataforma OFDM');
    %legend('DeltaN/N = 0.0;0.2;0.4;0.6;0.8;0.9;1.0 escrever taxa de erro (%)');
    hold off;
    %
    %set(0,'DefaultAxesLineStyleOrder','remove')
    set(0,'DefaultAxesColorOrder','remove')

```

OFDM_BERxEbNo_todos

```

%bloco=50; %Bernoulli Binary Generator: Samples per Frame = bloco
%repvet=[];
%repvet=[1 31 61 91 120 135 149];
%for i=1:length(repvet);
%    rep=repvet(i); %Demux: Number of outputs = [3*bloco-rep,rep]
%
%set(0,'DefaultAxesLineStyleOrder',{'-','--',':', '-.'})
set(0,'DefaultAxesColorOrder',[1 0 0])
%
% 2 PILOTOS
EbNoVec=5.0:1.0:15.0;
BER_Vec=[];
EBNOX=[];
for n=1:length(EbNoVec);
    EbNo=EbNoVec(n);
    sim('OFDMP2');
    if erro > 0.00000001;
        BER_Vec(n,:)=erro;
        EBNOX(n,:)=EbNo;
    else
        n=length(EbNoVec);
    end;
end;
end;

```

```

semilogy(EBNOX(:,1),BER_Vec(:,1),'+');
%semilogy(EBNOX(:,1),BER_Vec(:,1));
berfit(EBNOX(:,1),BER_Vec(:,1));
hold on;
%end;
% 3 PILOTOS
EbNoVec=5.0:1.0:15.0;
BER_Vec=[];
EBNOX=[];
for n=1:length(EbNoVec);
    EbNo=EbNoVec(n);
    sim('OFDMP3');
    if erro > 0.00000001;
        BER_Vec(n,:)=erro;
        EBNOX(n,:)=EbNo;
    else
        n=length(EbNoVec);
    end;
end;
semilogy(EBNOX(:,1),BER_Vec(:,1),'+');
%semilogy(EBNOX(:,1),BER_Vec(:,1));
berfit(EBNOX(:,1),BER_Vec(:,1));
hold on;
%
% 4 PILOTOS
EbNoVec=5.0:1.0:15.0;
BER_Vec=[];
EBNOX=[];
for n=1:length(EbNoVec);
    EbNo=EbNoVec(n);
    sim('OFDMP4');
    if erro > 0.00000001;
        BER_Vec(n,:)=erro;
        EBNOX(n,:)=EbNo;
    else
        n=length(EbNoVec);
    end;
end;
semilogy(EBNOX(:,1),BER_Vec(:,1),'+');
%semilogy(EBNOX(:,1),BER_Vec(:,1));
berfit(EBNOX(:,1),BER_Vec(:,1));
hold on;
%
% 5 PILOTOS
EbNoVec=5.0:1.0:15.0;
BER_Vec=[];
EBNOX=[];
for n=1:length(EbNoVec);
    EbNo=EbNoVec(n);
    sim('OFDMP5');
    if erro > 0.00000001;
        BER_Vec(n,:)=erro;
        EBNOX(n,:)=EbNo;
    else
        n=length(EbNoVec);
    end;
end;
semilogy(EBNOX(:,1),BER_Vec(:,1),'+');
%semilogy(EBNOX(:,1),BER_Vec(:,1));
berfit(EBNOX(:,1),BER_Vec(:,1));
hold on;

```

```

%
% 6 PILOTOS
EbNoVec=5.0:1.0:15.0;
BER_Vec=[];
EBNOX=[];
for n=1:length(EbNoVec);
    EbNo=EbNoVec(n);
    sim('OFDMP6');
    if erro > 0.00000001;
        BER_Vec(n,:)=erro;
        EBNOX(n,:)=EbNo;
    else
        n=length(EbNoVec);
    end;
end;
semilogy(EBNOX(:,1),BER_Vec(:,1),'+');
%semilogy(EBNOX(:,1),BER_Vec(:,1));
berfit(EBNOX(:,1),BER_Vec(:,1));
hold on;
%
% 7 PILOTOS
EbNoVec=5.0:1.0:15.0;
BER_Vec=[];
EBNOX=[];
for n=1:length(EbNoVec);
    EbNo=EbNoVec(n);
    sim('OFDMP7');
    if erro > 0.00000001;
        BER_Vec(n,:)=erro;
        EBNOX(n,:)=EbNo;
    else
        n=length(EbNoVec);
    end;
end;
semilogy(EBNOX(:,1),BER_Vec(:,1),'+');
%semilogy(EBNOX(:,1),BER_Vec(:,1));
berfit(EBNOX(:,1),BER_Vec(:,1));
hold on;
%
% 11 PILOTOS
EbNoVec=5.0:1.0:15.0;
BER_Vec=[];
EBNOX=[];
for n=1:length(EbNoVec);
    EbNo=EbNoVec(n);
    sim('OFDMP11');
    if erro > 0.00000001;
        BER_Vec(n,:)=erro;
        EBNOX(n,:)=EbNo;
    else
        n=length(EbNoVec);
    end;
end;
semilogy(EBNOX(:,1),BER_Vec(:,1),'+');
%semilogy(EBNOX(:,1),BER_Vec(:,1));
berfit(EBNOX(:,1),BER_Vec(:,1));
hold on;
%
% 13 PILOTOS
EbNoVec=5.0:1.0:15.0;
BER_Vec=[];

```



```

EBNOX=[];
for n=1:length(EbNoVec);
    EbNo=EbNoVec(n);
    sim('OFDMP13');
    if erro > 0.000000001;
        BER_Vec(n,:)=erro;
        EBNOX(n,:)=EbNo;
    else
        n=length(EbNoVec);
    end;
end;
semilogy(EBNOX(:,1),BER_Vec(:,1),'+');
%semilogy(EBNOX(:,1),BER_Vec(:,1));
berfit(EBNOX(:,1),BER_Vec(:,1));
hold on;
%
% 16 PILOTOS
EbNoVec=5.0:1.0:15.0;
BER_Vec=[];
EBNOX=[];
for n=1:length(EbNoVec);
    EbNo=EbNoVec(n);
    sim('OFDMP16');
    if erro > 0.000000001;
        BER_Vec(n,:)=erro;
        EBNOX(n,:)=EbNo;
    else
        n=length(EbNoVec);
    end;
end;
semilogy(EBNOX(:,1),BER_Vec(:,1),'+');
%semilogy(EBNOX(:,1),BER_Vec(:,1));
berfit(EBNOX(:,1),BER_Vec(:,1));
hold on;
%
% 21 PILOTOS
EbNoVec=5.0:1.0:15.0;
BER_Vec=[];
EBNOX=[];
for n=1:length(EbNoVec);
    EbNo=EbNoVec(n);
    sim('OFDMP21');
    if erro > 0.000000001;
        BER_Vec(n,:)=erro;
        EBNOX(n,:)=EbNo;
    else
        n=length(EbNoVec);
    end;
end;
semilogy(EBNOX(:,1),BER_Vec(:,1),'+');
%semilogy(EBNOX(:,1),BER_Vec(:,1));
berfit(EBNOX(:,1),BER_Vec(:,1));
hold on;
%
% 26 PILOTOS
EbNoVec=5.0:1.0:15.0;
BER_Vec=[];
EBNOX=[];
for n=1:length(EbNoVec);
    EbNo=EbNoVec(n);
    sim('OFDMP26');

```

```

        if erro > 0.00000001;
            BER_Vec(n,:)=erro;
            EBNOX(n,:)=EbNo;
        else
            n=length(EbNoVec);
        end;
    end;
    semilogy(EBNOX(:,1),BER_Vec(:,1),'+');
    %semilogy(EBNOX(:,1),BER_Vec(:,1));
    berfit(EBNOX(:,1),BER_Vec(:,1));
    hold on;
    %
    % 31 PILOTOS
    EbNoVec=5.0:1.0:15.0;
    BER_Vec=[];
    EBNOX=[];
    for n=1:length(EbNoVec);
        EbNo=EbNoVec(n);
        sim('OFDMP31');
        if erro > 0.00000001;
            BER_Vec(n,:)=erro;
            EBNOX(n,:)=EbNo;
        else
            n=length(EbNoVec);
        end;
    end;
    semilogy(EBNOX(:,1),BER_Vec(:,1),'+');
    %semilogy(EBNOX(:,1),BER_Vec(:,1));
    berfit(EBNOX(:,1),BER_Vec(:,1));
    hold on;
    %
    % 51 PILOTOS
    EbNoVec=5.0:1.0:15.0;
    BER_Vec=[];
    EBNOX=[];
    for n=1:length(EbNoVec);
        EbNo=EbNoVec(n);
        sim('OFDMP51');
        if erro > 0.00000001;
            BER_Vec(n,:)=erro;
            EBNOX(n,:)=EbNo;
        else
            n=length(EbNoVec);
        end;
    end;
    semilogy(EBNOX(:,1),BER_Vec(:,1),'+');
    %semilogy(EBNOX(:,1),BER_Vec(:,1));
    berfit(EBNOX(:,1),BER_Vec(:,1));
    hold on;
    %
    % 61 PILOTOS
    EbNoVec=5.0:1.0:15.0;
    BER_Vec=[];
    EBNOX=[];
    for n=1:length(EbNoVec);
        EbNo=EbNoVec(n);
        sim('OFDMP61');
        if erro > 0.00000001;
            BER_Vec(n,:)=erro;
            EBNOX(n,:)=EbNo;
        else

```

```

        n=length(EbNoVec);
        end;
    end;
    semilogy(EBNOX(:,1),BER_Vec(:,1),'+');
    %semilogy(EBNOX(:,1),BER_Vec(:,1));
    berfit(EBNOX(:,1),BER_Vec(:,1));
    hold on;
    %
    % 76 PILOTOS
    EbNoVec=5.0:1.0:15.0;
    BER_Vec=[];
    EBNOX=[];
    for n=1:length(EbNoVec);
        EbNo=EbNoVec(n);
        sim('OFDMP76');
        if erro > 0.00000001;
            BER_Vec(n,:)=erro;
            EBNOX(n,:)=EbNo;
        else
            n=length(EbNoVec);
        end;
    end;
    semilogy(EBNOX(:,1),BER_Vec(:,1),'+');
    %semilogy(EBNOX(:,1),BER_Vec(:,1));
    berfit(EBNOX(:,1),BER_Vec(:,1));
    hold on;
    %
    % 101 PILOTOS
    EbNoVec=5.0:1.0:15.0;
    BER_Vec=[];
    EBNOX=[];
    for n=1:length(EbNoVec);
        EbNo=EbNoVec(n);
        sim('OFDMP101');
        if erro > 0.00000001;
            BER_Vec(n,:)=erro;
            EBNOX(n,:)=EbNo;
        else
            n=length(EbNoVec);
        end;
    end;
    semilogy(EBNOX(:,1),BER_Vec(:,1),'+');
    %semilogy(EBNOX(:,1),BER_Vec(:,1));
    berfit(EBNOX(:,1),BER_Vec(:,1));
    hold on;
    %
    % 151 PILOTOS
    EbNoVec=5.0:1.0:15.0;
    BER_Vec=[];
    EBNOX=[];
    for n=1:length(EbNoVec);
        EbNo=EbNoVec(n);
        sim('OFDMP151');
        if erro > 0.00000001;
            BER_Vec(n,:)=erro;
            EBNOX(n,:)=EbNo;
        else
            n=length(EbNoVec);
        end;
    end;
    semilogy(EBNOX(:,1),BER_Vec(:,1),'+');

```

```

%semilogy(EBNOX(:,1),BER_Vec(:,1));
berfit(EBNOX(:,1),BER_Vec(:,1));
hold on;
%
xlabel('Eb/No (dB)');
ylabel('BER');
%title('Plataforma OFDM');
%legend('DeltaN/N = 0.0;0.2;0.4;0.6;0.8;0.9;1.0 escrever taxa de erro (%)');
hold off;
%
%set(0,'DefaultAxesLineStyleOrder','remove')
set(0,'DefaultAxesColorOrder','remove')

```

OFDM_BERxPilotos_2x7

```

%set(0,'DefaultAxesLineStyleOrder',{'-','--',':', '-.'})
set(0,'DefaultAxesColorOrder',[1 0 0])
%
Pilotos=[2 3 4 5 6 7];
BERVec=[0.401376973927282 0.400996490511249 0.389053959427532
0.42861528628076 0.380386085093648 0.268033636183984];
plot(Pilotos,BERVec);
hold on;
%
xlabel('Pilotos');
ylabel('BER');
%title('Plataforma OFDM');
%legend('DeltaN/N = 0.0;0.2;0.4;0.6;0.8;0.9;1.0 escrever taxa de erro (%)');
hold off;
%
%set(0,'DefaultAxesLineStyleOrder','remove')
set(0,'DefaultAxesColorOrder','remove')

```

OFDM_BERxPilotos_11x31

```

%set(0,'DefaultAxesLineStyleOrder',{'-','--',':', '-.'})
set(0,'DefaultAxesColorOrder',[1 0 0])
%
Pilotos=[11 13 16 21 26 31];
BERVec=[0.010022916005804 0.005594859905376 0.011846648468292
0.005715170698032 0.007123687598943 0.005962909584411];
plot(Pilotos,BERVec);
hold on;
%
xlabel('Pilotos');
ylabel('BER');
%title('Plataforma OFDM');
%legend('DeltaN/N = 0.0;0.2;0.4;0.6;0.8;0.9;1.0 escrever taxa de erro (%)');
hold off;
%
%set(0,'DefaultAxesLineStyleOrder','remove')
set(0,'DefaultAxesColorOrder','remove')

```

OFDM_BERxPilotos_51x151

```

%set(0,'DefaultAxesLineStyleOrder',{'-','--',':', '-.'})

```

```

set(0,'DefaultAxesColorOrder',[1 0 0])
%
Pilotos=[51 61 76 101 151];
BERVec=[0.088800292255878 0.102355907419566 0.081753197364477
0.087059077475243 0.046347480594433];
plot(Pilotos,BERVec);
hold on;
%
xlabel('Pilotos');
ylabel('BER');
%title('Plataforma OFDM');
%legend('DeltaN/N = 0.0;0.2;0.4;0.6;0.8;0.9;1.0 escrever taxa de erro
(%)');
hold off;
%
%set(0,'DefaultAxesLineStyleOrder','remove')
set(0,'DefaultAxesColorOrder','remove')

```

OFDM_BERxPilotos_todos

```

%set(0,'DefaultAxesLineStyleOrder',{'-','--',':', '-.'})
set(0,'DefaultAxesColorOrder',[1 0 0])
%
Pilotos=[2 3 4 5 6 7 11 13 16 21 26 31 51 61 76 101 151];
BERVec=[0.401376973927282 0.400996490511249 0.389053959427532
0.42861528628076 0.380386085093648 0.268033636183984 0.010022916005804...
0.005594859905376 0.011846648468292 0.005715170698032 0.007123687598943
0.005962909584411 0.088800292255878 0.102355907419566...
0.081753197364477 0.087059077475243 0.046347480594433];
plot(Pilotos,BERVec);
hold on;
%
xlabel('Pilotos');
ylabel('BER');
%title('Plataforma OFDM');
%legend('DeltaN/N = 0.0;0.2;0.4;0.6;0.8;0.9;1.0 escrever taxa de erro
(%)');
hold off;
%
%set(0,'DefaultAxesLineStyleOrder','remove')
set(0,'DefaultAxesColorOrder','remove')

```

OFDM_ThroughputxEbNo_todos - Vs 2

```

%set(0,'DefaultAxesLineStyleOrder',{'-','--',':', '-.'})
set(0,'DefaultAxesColorOrder',[1 0 0])
%
EbNoVec=[5 6 7 8 9 10 11 12 13 14 15];
% 2 PILOTOS
VazaoVec=[0.586340032063529 0.588413480631607 0.589939066069932
0.591133554113823 0.592100309528066 0.592899418862838 0.593571927478613...
0.59414948279126 0.594658635171574 0.595123033817227
0.595564779029912];
plot(EbNoVec,VazaoVec);
hold on;
% 3 PILOTOS
VazaoVec=[0.583563334563899 0.584863309732231 0.586134955553339
0.587376427664451 0.58858596668465 0.589761901344871 0.590902652690518...
0.592006739071859 0.593072781672031 0.594099510351109
0.595085769608298];

```

```

plot (EbNoVec,VazaoVec);
hold on;
% 4 PILOTOS
VazaoVec=[0.591308143825585 0.593597624896989 0.595643461298776
0.5974435342476 0.598998990971317 0.600314210651444 0.601396673185042...
0.602256729401687 0.602907276880725 0.603363351364007
0.603641649511078];
plot (EbNoVec,VazaoVec);
hold on;
% 5 PILOTOS
VazaoVec=[0.556966695336093 0.557598299484568 0.55822981871242
0.558861267125458 0.559492655649492 0.560123992685701 0.560755284630566...
0.56138653628848 0.562017751199253 0.56264893189809 0.56328008012202];
plot (EbNoVec,VazaoVec);
hold on;
% 6 PILOTOS
VazaoVec=[0.597351663251975 0.59945203516413 0.601219962241854
0.602717442953995 0.603986588944284 0.605063330911097 0.605981058341814...
0.606771447409069 0.607464622457208 0.608089328918771
0.608673308751179];
plot (EbNoVec,VazaoVec);
hold on;
% 7 PILOTOS
VazaoVec=[0.695065457370252 0.700758449980855 0.705134825824274
0.708393208983616 0.71075536350658 0.712442653899121 0.713655432214644...
0.714558811770714 0.715276577018908 0.71589279701024
0.716458859735194];
plot (EbNoVec,VazaoVec);
hold on;
% 11 PILOTOS
VazaoVec=[0.909419671072975 0.922957155361323 0.933567938132207
0.941319287356539 0.94661162953413 0.95005522860574 0.95228792863862...
0.953818281229431 0.954961817357745 0.955872097047599
0.956611565805566];
plot (EbNoVec,VazaoVec);
hold on;
% 13 PILOTOS
VazaoVec=[0.90426799160111 0.91939029812212 0.931251929611135
0.939898244022239 0.945709681711563 0.94930130230285 0.951360206725669...
0.952486700425225 0.953103968141812 0.953456768970941
0.953666374548716];
plot (EbNoVec,VazaoVec);
hold on;
% 16 PILOTOS
VazaoVec=[0.890774019506559 0.906178021522783 0.917659040557855
0.92578318463385 0.931205471296879 0.934593925805202 0.936560218221412...
0.937609985590971 0.938120270441495 0.938343487955672
0.938430212071208];
plot (EbNoVec,VazaoVec);
hold on;
% 21 PILOTOS
VazaoVec=[0.883732667350861 0.898323755008863 0.909063794830629
0.917106885177495 0.922636636267542 0.926072315231683 0.927965907597691...
0.928871692606412 0.929238158226138 0.929359814184711
0.929391790450354];
plot (EbNoVec,VazaoVec);
hold on;
% 26 PILOTOS
VazaoVec=[0.866776684941314 0.880823822018679 0.891493119438703
0.899260974503516 0.904756639564279 0.908559617526359 0.911110010644954...
0.912732547629874 0.913689858037783 0.91420412294398 0.91445172511584];

```

```

plot(EbNoVec,VazaoVec);
hold on;
% 31 PILOTOS
VazaoVec=[0.856223590668736 0.868625598377364 0.879313171434014
0.887783050507687 0.893837647821128 0.897650104738508 0.899704251389143...
0.900618287385708 0.900939961101742 0.90102490141602
0.901040685975845];
plot(EbNoVec,VazaoVec);
hold on;
% 51 PILOTOS
VazaoVec=[0.749412848304926 0.758357379826744 0.76533740056737
0.77042836106411 0.773883442098948 0.776074396484422 0.777405385469944...
0.778230323084969 0.77880316901207 0.779272403444472
0.779707522249586];
plot(EbNoVec,VazaoVec);
hold on;
% 61 PILOTOS
VazaoVec=[0.717943756907229 0.726276056553386 0.732758180092381
0.737471859589855 0.740685083697756 0.742781180831077 0.744159676291243...
0.745148991117627 0.745964619872937 0.746718098865982
0.747453843654341];
plot(EbNoVec,VazaoVec);
hold on;
% 76 PILOTOS
VazaoVec=[0.69590877814284 0.705537318979625 0.713350168065006
0.719482422791661 0.724128746283957 0.72752313027753 0.729915693598132...
0.731549918800063 0.732643725507066 0.73337694898003
0.733886308681115];
plot(EbNoVec,VazaoVec);
hold on;
% 101 PILOTOS
VazaoVec=[0.646754836972755 0.655766257919912 0.662964281199846
0.668675562464978 0.673169735873254 0.676672771968959 0.679375416914563...
0.681438765430839 0.682998196402561 0.684166414130066 0.68503603531002]
;
plot(EbNoVec,VazaoVec);
hold on;
% 151 PILOTOS
VazaoVec=[0.587558614361401 0.597563615194515 0.607036120275644
0.615470634597639 0.622435978214229 0.627682068811232 0.63121585478372...
0.633300183141572 0.63435866035847 0.634826292783585
0.635026926582605];
plot(EbNoVec,VazaoVec);
hold on;
%
xlabel('Eb/No (dB)');
ylabel('Vazão (%)');
%title('Plataforma OFDM');
%legend('DeltaN/N = 0.0;0.2;0.4;0.6;0.8;0.9;1.0 escrever taxa de erro
(%)');
hold off;
%
%set(0,'DefaultAxesLineStyleOrder','remove')
set(0,'DefaultAxesColorOrder','remove')

```

OFDM_ThroughputxEbNo_todos - Vs 2 – AA – 11, 13, 16, 21, 26, 31

```

%set(0,'DefaultAxesLineStyleOrder',{'-','--',':', '-.'})
set(0,'DefaultAxesColorOrder',[1 0 0])
%
EbNoVec=[5 6 7 8 9 10 11 12 13 14 15];

```

```

% 11 PILOTOS
VazaoVec=[0.909419671072975 0.922957155361323 0.933567938132207
0.941319287356539 0.94661162953413 0.95005522860574 0.95228792863862...
0.953818281229431 0.954961817357745 0.955872097047599
0.956611565805566];
plot(EbNoVec,VazaoVec);
hold on;
% 13 PILOTOS
VazaoVec=[0.90426799160111 0.91939029812212 0.931251929611135
0.939898244022239 0.945709681711563 0.94930130230285 0.951360206725669...
0.952486700425225 0.953103968141812 0.953456768970941
0.953666374548716];
plot(EbNoVec,VazaoVec);
hold on;
% 16 PILOTOS
VazaoVec=[0.890774019506559 0.906178021522783 0.917659040557855
0.92578318463385 0.931205471296879 0.934593925805202 0.936560218221412...
0.937609985590971 0.938120270441495 0.938343487955672
0.938430212071208] ;
plot(EbNoVec,VazaoVec);
hold on;
% 21 PILOTOS
VazaoVec=[0.883732667350861 0.898323755008863 0.909063794830629
0.917106885177495 0.922636636267542 0.926072315231683 0.927965907597691...
0.928871692606412 0.929238158226138 0.929359814184711
0.929391790450354];
plot(EbNoVec,VazaoVec);
hold on;
% 26 PILOTOS
VazaoVec=[0.866776684941314 0.880823822018679 0.891493119438703
0.899260974503516 0.904756639564279 0.908559617526359 0.911110010644954...
0.912732547629874 0.913689858037783 0.91420412294398 0.91445172511584];
plot(EbNoVec,VazaoVec);
hold on;
% 31 PILOTOS
VazaoVec=[0.856223590668736 0.868625598377364 0.879313171434014
0.887783050507687 0.893837647821128 0.897650104738508 0.899704251389143...
0.900618287385708 0.900939961101742 0.90102490141602
0.901040685975845];
plot(EbNoVec,VazaoVec);
hold on;
%
xlabel('Eb/No (dB)');
ylabel('Vazão (%)');
%title('Plataforma OFDM');
%legend('DeltaN/N = 0.0;0.2;0.4;0.6;0.8;0.9;1.0 escrever taxa de erro
(%)');
hold off;
%
%set(0,'DefaultAxesLineStyleOrder','remove')
set(0,'DefaultAxesColorOrder','remove')

```

OFDM_ThroughputxEbNo_todos - Vs 2 – BB – 7, 51, 61, 76, 101

```

%set(0,'DefaultAxesLineStyleOrder',{'-','--',':', '-.'})
set(0,'DefaultAxesColorOrder',[1 0 0])
%
EbNoVec=[5 6 7 8 9 10 11 12 13 14 15];
% 7 PILOTOS
VazaoVec=[0.695065457370252 0.700758449980855 0.705134825824274
0.708393208983616 0.71075536350658 0.712442653899121 0.713655432214644...

```



```

        0.714558811770714 0.715276577018908 0.71589279701024
0.716458859735194];
plot(EbNoVec,VazaoVec);
hold on;

% 51 PILOTOS
VazaoVec=[0.749412848304926 0.758357379826744 0.76533740056737
0.77042836106411 0.773883442098948 0.776074396484422 0.777405385469944...
0.778230323084969 0.77880316901207 0.779272403444472
0.779707522249586];
plot(EbNoVec,VazaoVec);
hold on;

% 61 PILOTOS
VazaoVec=[0.717943756907229 0.726276056553386 0.732758180092381
0.737471859589855 0.740685083697756 0.742781180831077 0.744159676291243...
0.745148991117627 0.745964619872937 0.746718098865982
0.747453843654341];
plot(EbNoVec,VazaoVec);
hold on;

% 76 PILOTOS
VazaoVec=[0.69590877814284 0.705537318979625 0.713350168065006
0.719482422791661 0.724128746283957 0.72752313027753 0.729915693598132...
0.731549918800063 0.732643725507066 0.73337694898003
0.733886308681115];
plot(EbNoVec,VazaoVec);
hold on;

% 101 PILOTOS
VazaoVec=[0.646754836972755 0.655766257919912 0.662964281199846
0.668675562464978 0.673169735873254 0.676672771968959 0.679375416914563...
0.681438765430839 0.682998196402561 0.684166414130066 0.68503603531002]
;
plot(EbNoVec,VazaoVec);
hold on;
%
xlabel('Eb/No (dB)');
ylabel('Vazão (%)');
%title('Plataforma OFDM');
%legend('DeltaN/N = 0.0;0.2;0.4;0.6;0.8;0.9;1.0 escrever taxa de erro
(%)');
hold off;
%
%set(0,'DefaultAxesLineStyleOrder','remove')
set(0,'DefaultAxesColorOrder','remove')

```

OFDM_ThroughputxEbNo_todos - Vs 2 – CC – 2, 3, 4, 5, 6, 151

```

%set(0,'DefaultAxesLineStyleOrder',{'-','--',':', '-.'})
set(0,'DefaultAxesColorOrder',[1 0 0])
%
EbNoVec=[5 6 7 8 9 10 11 12 13 14 15];
% 2 PILOTOS
VazaoVec=[0.586340032063529 0.588413480631607 0.589939066069932
0.591133554113823 0.592100309528066 0.592899418862838 0.593571927478613...
0.59414948279126 0.594658635171574 0.595123033817227
0.595564779029912];
plot(EbNoVec,VazaoVec);
hold on;
% 3 PILOTOS
VazaoVec=[0.583563334563899 0.584863309732231 0.586134955553339
0.587376427664451 0.58858596668465 0.589761901344871 0.590902652690518...

```

```

    0.592006739071859 0.593072781672031 0.594099510351109
0.595085769608298];
plot(EbNoVec,VazaoVec);
hold on;
% 4 PILOTOS
VazaoVec=[0.591308143825585 0.593597624896989 0.595643461298776
0.5974435342476 0.598998990971317 0.600314210651444 0.601396673185042...
0.602256729401687 0.602907276880725 0.603363351364007
0.603641649511078];
plot(EbNoVec,VazaoVec);
hold on;
% 5 PILOTOS
VazaoVec=[0.556966695336093 0.557598299484568 0.55822981871242
0.558861267125458 0.559492655649492 0.560123992685701 0.560755284630566...
0.56138653628848 0.562017751199253 0.56264893189809 0.56328008012202];
plot(EbNoVec,VazaoVec);
hold on;
% 6 PILOTOS
VazaoVec=[0.597351663251975 0.59945203516413 0.601219962241854
0.602717442953995 0.603986588944284 0.605063330911097 0.605981058341814...
0.606771447409069 0.607464622457208 0.608089328918771
0.608673308751179];
plot(EbNoVec,VazaoVec);
hold on;
% 151 PILOTOS
VazaoVec=[0.587558614361401 0.597563615194515 0.607036120275644
0.615470634597639 0.622435978214229 0.627682068811232 0.63121585478372...
0.633300183141572 0.63435866035847 0.634826292783585
0.635026926582605];
plot(EbNoVec,VazaoVec);
hold on;
%
xlabel('Eb/No (dB)');
ylabel('Vazão (%)');
%title('Plataforma OFDM');
%legend('DeltaN/N = 0.0;0.2;0.4;0.6;0.8;0.9;1.0 escrever taxa de erro
(%)');
hold off;
%
%set(0,'DefaultAxesLineStyleOrder','remove')
set(0,'DefaultAxesColorOrder','remove')

```

OFDM_ThroughputxPilotos_2x7

```

%set(0,'DefaultAxesLineStyleOrder',{'-','-',':','-.'})
set(0,'DefaultAxesColorOrder',[1 0 0])
%
%Pilotos=[2 3 4 5 6 7];
Pilotos=2:1:7;
VazaoVec=[0.594658635171574 0.593072781672031 0.602907276880725
0.562017751199253 0.607464622457208 0.715276577018908];
plot(Pilotos,VazaoVec);
hold on;
%
xlabel('Pilotos');
ylabel('Vazão(%)');
%title('Plataforma OFDM');
%legend('DeltaN/N = 0.0;0.2;0.4;0.6;0.8;0.9;1.0 escrever taxa de erro
(%)');
hold off;
%

```

```
%set(0,'DefaultAxesLineStyleOrder','remove')
set(0,'DefaultAxesColorOrder','remove')
```

OFDM_ThroughputXPilotos_11x31

```
%set(0,'DefaultAxesLineStyleOrder',{'-','--',':', '-.'})
set(0,'DefaultAxesColorOrder',[1 0 0])
%
Pilotos=[11 13 16 21 26 31];
VazaoVec=[0.954961817357745 0.953103968141812 0.938120270441495
0.929238158226138 0.913689858037783 0.900939961101742];
plot(Pilotos,VazaoVec);
hold on;
%
xlabel('Pilotos');
ylabel('Vazão(%)');
%title('Plataforma OFDM');
%legend('DeltaN/N = 0.0;0.2;0.4;0.6;0.8;0.9;1.0 escrever taxa de erro
(%)');
hold off;
%
%set(0,'DefaultAxesLineStyleOrder','remove')
set(0,'DefaultAxesColorOrder','remove')
```

OFDM_ThroughputXPilotos_51x151

```
%set(0,'DefaultAxesLineStyleOrder',{'-','--',':', '-.'})
set(0,'DefaultAxesColorOrder',[1 0 0])
%
Pilotos=[51 61 76 101 151];
VazaoVec=[0.77880316901207 0.745964619872937 0.732643725507066
0.682998196402561 0.63435866035847];
plot(Pilotos,VazaoVec);
hold on;
%
xlabel('Pilotos');
ylabel('Vazão(%)');
%title('Plataforma OFDM');
%legend('DeltaN/N = 0.0;0.2;0.4;0.6;0.8;0.9;1.0 escrever taxa de erro
(%)');
hold off;
%
%set(0,'DefaultAxesLineStyleOrder','remove')
set(0,'DefaultAxesColorOrder','remove')
```

OFDM_ThroughputXPilotos_todos

```
%set(0,'DefaultAxesLineStyleOrder',{'-','--',':', '-.'})
set(0,'DefaultAxesColorOrder',[1 0 0])
%
Pilotos=[2 3 4 5 6 7 11 13 16 21 26 31 51 61 76 101 151];
VazaoVec=[0.594658635171574 0.593072781672031 0.602907276880725
0.562017751199253 0.607464622457208 0.715276577018908 0.954961817357745
0.953103968141812...
0.938120270441495 0.929238158226138 0.913689858037783 0.900939961101742
0.77880316901207 0.745964619872937 0.732643725507066 0.682998196402561...
0.63435866035847];
```

```

plot(Pilotos,VazaoVec);
hold on;
%
xlabel('Pilotos');
ylabel('Vazão(%)');
%title('Plataforma OFDM');
%legend('DeltaN/N = 0.0;0.2;0.4;0.6;0.8;0.9;1.0 escrever taxa de erro
(%)');
hold off;
%
%set(0,'DefaultAxesLineStyleOrder','remove')
set(0,'DefaultAxesColorOrder','remove')

```

OFDM_FramesxPilotos

```

%set(0,'DefaultAxesLineStyleOrder',{'-','--',':', '-.'})
set(0,'DefaultAxesColorOrder',[1 0 0])
%
Pilotos=[2 3 4 5 6 7 11 13 16 21 26 31 51 61 76 101 151];
FrameVec=[1146 1146 1179 1069 1206 1716 55031 91358 40311 83543 71493 78414
5207 4543 5694 5415 10084];
plot(Pilotos,FrameVec);
hold on;
%
xlabel('Pilotos');
ylabel('Frames');
%title('Plataforma OFDM');
%legend('DeltaN/N = 0.0;0.2;0.4;0.6;0.8;0.9;1.0 escrever taxa de erro
(%)');
hold off;
%
%set(0,'DefaultAxesLineStyleOrder','remove')
set(0,'DefaultAxesColorOrder','remove')

```

OFDM – CP

```

% Programa para o levantamento de curvas de Taxa de Erro x Eb/No com
% consideração de aproveitamento da redundância dos bits de "cyclic prefix"
% em enlaces OFDM
format short e;
erro=[];
repvet=[];
repvet=[1 248];
for i=1:length(repvet);
    rep=repvet(i);
    EbNoVec=5:1.0:15.0;
    BER_Vec=[];
    EBNÖX=[];
    for n=1:length(EbNoVec);
        EbNo=EbNoVec(n);
        sim('ofdm3');
        aux=size(erro,1); % Retorna o número de linhas do vetor erro
        erro=erro(aux);
        if erro > 0.000000001;
            BER_Vec(n,:)=erro;
            EBNÖX(n,:)=EbNo;
        else
            n=length(EbNoVec);
        end;
    end;
end;

```

```
semilogy(EBNOX(:,1),BER_Vec(:,1),'+');
berfit(EBNOX(:,1),BER_Vec(:,1));
RESULTADO=[EBNOX BER_Vec]
hold on;
end;
xlabel('Eb/No (dB)');
ylabel('P(erro)');
title('OFDM - Ganho de Prefixo');
legend('Ganho Desativado/Ativado');
```

OFDM_MultiportSelectorxPilotos

```
%set(0,'DefaultAxesLineStyleOrder',{'-','--',':', '-.'})
set(0,'DefaultAxesColorOrder',[1 0 0])
%
Pilots=[2 3 4 5 6 7 11 13 16 21 26 31 51 61 76 101 151];
MultiportSelector=[1 2 3 4 5 6 10 12 15 20 25 30 50 60 75 100 150];
plot(Pilots,MultiportSelector);
hold on;
%
xlabel('Pilots');
ylabel(' Multiport Selector ');
title('Plataforma OFDM');
legend('DeltaN/N = 0.0;0.2;0.4;0.6;0.8;0.9;1.0 escrever taxa de erro (%)');
hold off;
%
%set(0,'DefaultAxesLineStyleOrder','remove')
set(0,'DefaultAxesColorOrder','remove')
```

OFDM_BitsxPilots - Vs 2

```
%set(0,'DefaultAxesLineStyleOrder',{'-','--',':', '-.'})
set(0,'DefaultAxesColorOrder',[1 0 0])
%
Pilotos=[2 3 4 5 6 7 11 13 16 21 26 31 51 61 76 101 151];
% Quadro total
PercTotal=[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1];
plot(Pilotos, PercTotal);
hold on;
% DADOS
PercDados=[0.54744525547445300 0.54744525547445300 0.54744525547445300
0.54744525547445300...
0.54744525547445300 0.54744525547445300 0.54744525547445300
0.54744525547445300...
0.54744525547445300 0.54744525547445300 0.54744525547445300
0.54744525547445300...
0.54744525547445300 0.54744525547445300 0.54744525547445300
0.54744525547445300 0.54744525547445300];
plot(Pilotos,PercDados);
hold on;
% ZEROS DE ENCHIMENTO
PercZeros=[0.38321167883211700 0.38138686131386900 0.37956204379562000
0.37773722627737200...
0.37591240875912400 0.37408759124087600 0.36678832116788300
0.36313868613138700 0.35766423357664200...
0.34854014598540100 0.33941605839416100 0.33029197080292000
0.29379562043795600 0.27554744525547400...
0.24817518248175200 0.20255474452554700 0.11131386861313900];
```

```

plot(Pilotos,PercZeros);
hold on;
% PILOTOS
PercPilotos=[0.00364963503649635 0.00547445255474453 0.00729927007299270
0.00912408759124088...
0.01094890510948910 0.01277372262773720 0.02007299270072990
0.02372262773722630 0.02919708029197080...
0.03832116788321170 0.04744525547445260 0.05656934306569340
0.09306569343065690 0.11131386861313900...
0.13868613138686100 0.18430656934306600 0.27554744525547400];
plot(Pilotos,PercPilotos);
hold on;
% PREFIXO CÍCLICO
PercCP=[0.06569343065693430 0.06569343065693430 0.06569343065693430
0.06569343065693430 0.06569343065693430...
0.06569343065693430 0.06569343065693430 0.06569343065693430
0.06569343065693430 0.06569343065693430...
0.06569343065693430 0.06569343065693430 0.06569343065693430
0.06569343065693430 0.06569343065693430...
0.06569343065693430 0.06569343065693430];
plot(Pilotos,PercCP);
hold on;
%
xlabel(' Pilots');
ylabel(' Bits (%)');
title('Bits vs. Pilots');
%legend('DeltaN/N = 0.0;0.2;0.4;0.6;0.8;0.9;1.0 escrever taxa de erro
(%)');
hold off;
%
%set(0,'DefaultAxesLineStyleOrder','remove')
set(0,'DefaultAxesColorOrder','remove')

```

C.3. Conclusão

Este apêndice apresentou todos os Programas feitos para a Plataforma OFDM e para plotagem dos gráficos.