



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CTC - CENTRO TECNOLÓGICO
DEPARTAMENTO DE ENGENHARIA ELÉTRICA E ELETRÔNICA

Estimação e Compensação de Canal em um Sistema OFDM Implementado em Rádio Definido por Software

Winderson Luiz Franzoi Speranzini

Orientador: Prof. Danilo Silva, Ph.D.

Florianópolis
2015

WINDERSON LUIZ FRANZOI SPERANZINI

**ESTIMAÇÃO E COMPENSAÇÃO DE CANAL
EM UM SISTEMA OFDM IMPLEMENTADO EM
RÁDIO DEFINIDO POR SOFTWARE**

Monografia submetida ao curso de Engenharia Elétrica da Universidade Federal de Santa Catarina como requisito para aprovação da disciplina EEL7890 - Projeto Final.
Orientador: Prof. Danilo Silva, Ph.D.

**FLORIANÓPOLIS
2015**

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Speranzini, Winderson

Estimação e Compensação de Canal em um Sistema OFDM
Implementado em Rádio Definido por Software / Winderson
Speranzini ; orientador, Danilo Silva - Florianópolis, SC,
2015.

82 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Centro Tecnológico.
Graduação em Engenharia Elétrica.

Inclui referências

1. Engenharia Elétrica. 2. OFDM. 3. Codificação de Rede
na camada física. 4. Rádio Definido por Software. 5.
802.11. I. Silva, Danilo. II. Universidade Federal de
Santa Catarina. Graduação em Engenharia Elétrica. III.
Título.

Winderson Luiz Franzoi Speranzini

**ESTIMAÇÃO E COMPENSAÇÃO DE CANAL EM UM
SISTEMA OFDM IMPLEMENTADO EM RÁDIO
DEFINIDO POR SOFTWARE**

Esta Monografia foi julgada adequada no contexto da disciplina EEL7890 - Projeto Final, e aprovada em sua forma final pelo Departamento de Engenharia Elétrica da Universidade Federal de Santa Catarina.

Florianópolis, 08 de setembro de 2015.



Prof. Renato Lucas Pacheco, Dr.
Coordenador do Curso

Banca Examinadora:



Prof. Danilo Silva, Ph.D.
Orientador
Universidade Federal de Santa Catarina



Prof. Carlos Aurélio Faria da Rocha, Dr.
Universidade Federal de Santa Catarina



Prof. Bartolomeu Ferreira Uchôa Filho, Ph.D.
Universidade Federal de Santa Catarina

Para meus pais Irvando e Rosani.

Agradecimentos

Desejo expressar meu reconhecimento e gratidão as pessoas que foram essenciais para realização deste trabalho.

ao Professor Danilo Silva por me orientar com maestria, fazendo-me questionar e refletir sempre e até sobre as coisas mais simples, guiando com muita paciência, dando dicas sobre os mais variados assuntos e realizando papel fundamental e inesquecível na minha formação;

ao Professor Carlos Aurélio Faria Da Rocha por todo seu empenho e contribuições na minha formação e que, mesmo em momentos ruins, se propos a ministrar matérias chave na construção deste trabalho;

ao Professor Bartolomeu Ferreira Uchôa Filho que, com sua forma cordial e amigável, aceitou revisar e consequentemente melhorar este trabalho;

aos tantos outros professores do programa de graduação que contribuíram em todos os aspectos e tanto dentro da sala de aula como fora, sempre se propuseram a tirar dúvidas, enriquecer com suas experiências de vida e dedicarem seu tempo para nossa formação;

ao colega e amigo Ricardo Bohaczuk Venturelli pelo enorme apoio, descontraído e tornando melhor a vivência durante o percurso ao mesmo tempo que se disponibilizava a ajudar e ensinar em quaisquer momentos e sendo assim, foi um dos maiores contribuintes para construção deste trabalho.

aos meus pais, Irvando Luiz Seperanzini e Rosani Maria Franzoi por todo o apoio, carinho e amor incondicional durante toda a minha vida;

aos meus irmãos, Larissa Maria Franzoi Speranzini e Gleyson Franzoi Speranzini pela amizade e todos os bons momentos;

a minha querida Larissa Garbari, seus pais Valdir Garbari e Patrícia Quaiatto, seu irmão Angelo Garbari e demais familiares, que já fazem parte da minha família;

as tantas outras amizades feitas durante o curso que colaboraram em muitos e mais variados aspectos, tornando a experiência de realizar a graduação muito mais prazerosa.

RESUMO

Este projeto consiste da investigação e mitigação das não idealidades físicas que surgem em implementações de sistemas de comunicações digitais, com ênfase para sistemas que utilizem a modulação *Orthogonal Frequency-Division Multiplexing* (OFDM). Primeiramente foram elencadas e descritas as não idealidades mais críticas para ocorrer a comunicação. Estas compõem o modelo do canal. Revisamos então os conceitos que baseiam a modulação OFDM e elencamos as especificações do padrão 802.11 que baseiam parte deste trabalho. Utilizando o conjunto de um computador comum com o aparelho *Universal Software Radio Peripheral* (USRP) para atuar como uma plataforma de rádio definido por software, foi possível implementar e testar um conjunto de técnicas para estimação e compensação do canal. Ao fim, obtemos uma implementação que valida duas das técnicas propostas e auxiliará nos primeiros passos para os futuros trabalhos na área.

Palavras-chave: Codificação de Rede na camada física, PNC, Estimação de Canal, Compensação de Canal, PHY, 802.11, OFDM, Rádio Definido por Software, SDR

ABSTRACT

This project consist of the investigation and mitigation of physical nonidealities that shows in digital communication system implementations, with emphasis in systems using OFDM modulations. First were listed and described the most critical nonidealities, that compose the channel model, to occur the communication. We reviewed the concepts that base the ofdm modulation and listed the specifications of the 802.11 standard that base part of the presented work. Using a kit constituted of a personal computer and an USRP device to act like a software defined radio platform, was possible to implement and test a kit of channel estimation and compensation techniques. At the end, we get a implementation that validates two proposed techniques and will assit the first steps of the future works from the area.

Keywords: Physical-Layer Network Coding, PNC, Channel Estimation, Channel Compensation, PHY, 802.11, OFDM, Software Defined Rádio, SDR

Sumário

1	Introdução	1
1.1	Motivação	2
1.2	Objetivos	4
1.3	Itens Fora do Escopo do Trabalho	4
1.4	Organização do Trabalho	5
1.5	Considerações Matemáticas	5
2	Sistemas Baseados em OFDM	7
2.1	Modelo Simplificado do Canal	8
2.2	Princípios de uma Transmissão OFDM	10
2.3	Prefixo Cíclico	14
2.4	Perturbações no Sinal pelo <i>Hardware</i>	16
2.4.1	Desalinhamento de Portadora	16
2.4.2	Desalinhamento de Amostra	17
	Desalinhamento de Instante de Amostragem	18
	Desalinhamento de Frequência de Amostragem	19
2.4.3	Atraso de Frame	20
2.5	Resumo do Modelo Completo do Canal	21
3	Estimação de canal sob o padrão IEEE 802.11	23
3.1	Padrão IEEE 802.11	23

3.1.1	Estrutura do Pacote	25
	Short Training Field	25
	Long Training Field	26
	Signal Field	27
	Data Field	28
3.2	Estimações	29
3.2.1	Desvio de Frequência da Portadora	30
3.2.2	Resposta em Frequência do Canal	33
3.2.3	Desvio de Instante de Amostragem	34
4	Implementação na Prática e Resultados	35
4.1	Rádio Definido por Software	35
4.1.1	Sistema de SDR utilizado	36
	GNU Radio	37
	USRP	37
4.2	Projeto WiME	39
4.3	Organização dos Dispositivos e Softwares	40
4.3.1	Comunicação entre Estimador e Compensador	40
4.4	Desafios Práticos em Aberto	40
4.4.1	Estimação do Atraso de Frame	41
4.4.2	Estimação do Desvio de Frequência de Amostragem	41
4.4.3	Compensação do Atraso Entre Transmissores	41
4.4.4	Compensação do Desvio de Frequência de Amostragem	41
4.4.5	Compensação de Desvio de Instantes de Amostragem	42
4.5	Compensações Implementadas	42
4.5.1	Desvio de Frequência	43
4.5.2	Distorção Linear do Canal	43
4.6	Resultados	45
5	Conclusão	51
5.1	Trabalhos futuros	52
A	Base Teórica do OFDM	53
A.1	Discrete Fourier Transform	53
A.1.1	Convolação Circular e DFT	55
A.2	Demonstração detalhada: Ortogonalidade em Frequência	57

Lista de Figuras

1.1	Modos de troca de mensagens em uma topologia <i>two-way relay</i>	3
2.1	Canal seletivo dividido em sub-bandas	8
2.2	Modelo simplificado do canal	10
2.3	PSD de sinais single carrier e multi carrier	10
2.4	Esquema de geração de símbolos OFDM	11
2.5	Espectro de um sinal OFDM (escala linear)	11
2.6	Sistema OFDM simplificado	13
2.7	Intervalo de guarda entre símbolos	14
2.8	Símbolo precedido do CP	15
2.9	Efeito do desalinhamento de portadora	17
2.10	Desalinhamento de amostra	18
2.11	Perca de uma amostra na recepção	20
3.1	Estrutura do pacote 802.11a	25
3.2	Construção do STF	26
3.3	Estrutura do campo de sinal	28
3.4	Correlação da LTS com o Preâmbulo	30
3.5	Frame OFDM	32
4.1	Sistema SDR	37

4.2	Diagrama a nível de sistema do USRP	38
4.3	Bloco de compensação do desvio de frequência	44
4.4	Bloco de compensação de interferência linear do canal	45
4.5	Sistema de comunicação realizado	46
4.6	Módulo da estimação do tap $k = 26$ do canal em cada realização (transmissão de um frame)	47
4.7	Módulo da média dos $taps$ do canal antes da equalização	47
4.8	Módulo da média dos $taps$ do canal após a equalização	48
4.9	Estimação do Δ_f [kHz] entre transmissor e receptor em cada frame	48
4.10	Diferença média, no domínio da frequência, de fase entre os símbolos OFDM	49
A.1	Espectro DTFT e DFT	54
A.2	Exemplo de <i>zero padding</i>	57
A.3	Computação da convolução linear utilizando DFT	57

Lista de Abreviações

- ADC** Conversor Analógico para Digital, do inglês *Analog to Digital Converter*. 13, 17, 18, 19, 35, 36
- AGC** Controle Automático de Ganho, do inglês *Automatic Gain Control*. 26
- AWGN** Ruído Gaussiano Branco Aditivo, do inglês *Additive White Gaussian Noise*. 13
- BPSK** *Binary Phase-Shift Keying*. 27
- CFO** *Carrier Frequency Offset*. 16, 21
- CNC** Codificação de Rede Convencional, do inglês *Conventional Network Coding*. 3
- CP** Prefixo Cíclico, do inglês *Cyclic Prefix*. 14, 33, 51
- CPO** *Carrier Phase Offset*. 16, 21
- CSV** *Comma-Separated Values*. 40
- DAC** Conversor Digital para Analógico, do inglês *Digital to Analog Converter*. 12, 17, 18, 19, 35, 36

- DDC** *Digital Down-Converter*. 36
- DFT** *Discrete Fourier Transform*. 8, 13, 17, 19, 32, 53, 54, 55, 56
- DSP** *Digital Signal Processor*. 36
- DTFT** *Discrete Time Fourier Transform*. 53
- DUC** *Digital Up-Converter*. 36
- FFT** *Fast Fourier Transform*. 13, 14, 16, 18, 44, 55
- FIR** Resposta ao Impulso Finita, do inglês *Finite Impulse Response*.
9
- FPGA** *Field-Programmable Gate Array*. 37, 46
- GI** Intervalo de Guarda, do inglês *Guard Interval*. 14
- GR** *GNU Radio*. 36, 37, 38, 42, 45, 52
- GRC** *GNU Radio Companion*. 36, 37, 43
- ICI** *Inter Carrier Interference*. 16
- IDFT** *Inverse Discrete Fourier Transform*. 8, 12, 54, 55, 56
- IFFT** *Inverse Fast Fourier Transform*. 12, 13, 44, 55
- ISI** Interferência Intersimbólica, do inglês *Intersymbol Interference*. 7,
14
- LNA** *Low-Noise Amplifier*. 36
- LO** Oscilador Local, do inglês *Local Oscillator*. 36, 37
- LTF** Campo de Treinamento Longo, do inglês *Long Training Field*.
26, 27, 30, 31, 33, 34
- LTS** Sequência de Treinamento Longa, do inglês *Long Training Sequence*. 26, 30, 33
- MAC** *Media Access Control*. 23, 24, 38, 52

- OFDM** *Orthogonal Frequency-Division Multiplexing*. ix, xi, 1, 4, 5, 7, 9, 10, 11, 12, 13, 14, 15, 19, 24, 27, 28, 31, 32, 34, 39, 43, 44, 51, 53
- OSI** *Open Systems Interconnection*. 1, 23
- PAPR** *Peak-to-Average Power Ratio*. 25
- PHY** Camada Física, do inglês *Physical-Layer*. 23, 24, 38, 51, 52
- PLL** *Phase-Locked Loop*. 37
- PNC** Codificação de Rede na camada física, do inglês *Physical-Layer Network Coding*. 2, 3, 4, 20, 40, 41, 42, 51, 52
- PSD** Densidade Espectral de Potência, do inglês *Power Spectral Density*. 9
- QPSK** *Quadrature Phase-Shift Keying*. 39
- SDR** Rádio Definido por Software, do inglês *Software Defined Rádio*. 2, 4, 35, 38
- SFO** *Sampling Frequency Offset*. 17, 21
- SIG** Campo de Sinal, do inglês *Signal Field*. 27
- SPO** *Sampling Phase Offset*. 17, 21
- STF** Campo de Treinamento Curto, do inglês *Short Training Field*. 25, 26, 27, 31
- STS** Sequência de Treinamento Curta, do inglês *Short Training Sequence*. 25
- USRP** *Universal Software Radio Peripheral*. ix, xi, 2, 37, 38, 40, 45, 46
- VCO** *Voltage-Controlled Oscillator*. 37
- XML** *eXtensible Markup Language*. 43

CAPÍTULO 1

Introdução

Comunicações sem fio são parte essencial da sociedade moderna. Seja utilizando a internet, realizando uma ligação ou assistindo televisão, esta tecnologia é presente e realiza papel fundamental no nosso estilo de vida. Com a crescente expansão de aparelhos que necessitam de conexões robustas e rápidas à internet, pesquisas nesta área vem ganhando muito destaque.

Um subgrupo considerável deste conceito, são redes locais sem fio. Este subgrupo se beneficia dos vastos estudos para este tipo de sistema de comunicação. O conceito de camadas de processamento, como a divisão definida pelo modelo *Open Systems Interconnection* (OSI), é um desses exemplos de benefícios o qual abre possibilidades para reutilizar muitas outras técnicas como roteamento e codificação.

Em se tratando de realizar a transmissão e recepção física do sinal, um padrão amplamente aceito e utilizado nestes sistemas é o IEEE 802.11 e suas variantes. Eles preferencialmente utilizam como método de modulação, a técnica OFDM convencional.

O OFDM pode ser considerada tanto como uma técnica de modulação quanto uma técnica de multiplexação e tem provado, através das suas muitas implementações práticas, ser muito efetivo na mitiga-

ção dos efeitos dos multi-percursos de um canal de rádio (interferência intersimbólica), enquanto atingindo altas taxas de transmissão e eficiência espectral quando comparado com sistemas de portadora única [12].

Entretanto, a implementação prática em ambientes mais próximos a realidade atual destas tecnologias é algo relativamente complicado e oneroso. Assim, muitas vezes as universidades acabam se restringindo ao estudo teórico e implementação via simulações computacionais.

Com os avanços das tecnologias de processamento nos últimos anos, grandes mudanças ocorreram na forma como encaramos os problemas existentes. Em particular o conceito de Rádio Definido por Software, do inglês *Software Defined Radio* (SDR), junto com o desenvolvimento de plataformas acessíveis como por exemplo o USRP da *Ettus Research*, tornam muito mais simples este processo de implementação na prática.

Dessa forma, este projeto propõe uma implementação em plataforma SDR de um sistema de comunicação baseado no padrão IEEE 802.11 e focado nas técnicas de estimação e compensação do canal.

1.1 Motivação

A principal motivação para este trabalho foi a escassez de trabalhos os quais implementem a técnica Codificação de Rede na camada física, do inglês *Physical-Layer Network Coding* (PNC). Este é um conceito promissor e novo de codificação o qual foi proposto em 2006 em [13] para aplicações de redes sem fio.

A ideia por trás deste método é explorar a forma de codificação que ocorre naturalmente quando duas onda eletromagnéticas se sobrepõem uma a outra. Em muitas aplicações sem fio contemporâneas, esta interferência é tratada como um fenômeno destrutivo. Entretanto, se feito um bom uso, esta técnica pode chegar até a dobrar a taxa de transferência de dados de um sistema e sem qualquer alterações sensíveis no sinal em banda passante.

Uma topologia onde a técnica pode ser melhor exemplificada é a *two way relay*. Esta topologia consiste de dois nós (A e B) os quais desejam trocar uma mensagem através de um nó intermediário (R / *Relay*) por não possuírem um caminho de comunicação direto entre eles. A forma clássica de comunicação é dada pela Fig. 1.1a onde cada linha significa

um instante de tempo [6].

Na Fig. 1.1b esta está elucidada a técnica Codificação de Rede Convencional, do inglês *Conventional Network Coding* (CNC) a qual realiza a operação xOR (soma módulo 2, representada por \oplus) entre as mensagens, mas que é invertível pelos nós fontes (ex.: nó A realiza $m_A \oplus m_B \oplus m_A$ para obter m_B).

Por fim, o PNC (Fig. 1.1c) consiste da operação de soma física das mensagens. Após realizar uma técnica específica de detecção, é possível obter a combinação xOR das mensagens e reutilizar os mesmos princípios do CNC.

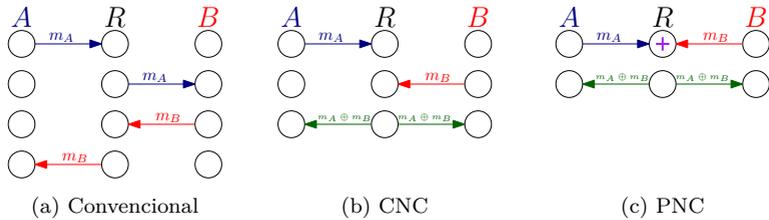


Figura 1.1: Modos de troca de mensagens em uma topologia *two-way relay*

A implementação desta última técnica consiste da realização de algumas etapas. Primeiramente é necessário controlar a sobreposição dos sinais enviados (recebidos no nó R) de forma a obter um resultado o mais próximo possível ao esperado. Após, é necessário realizar uma sequência de processamentos para obtermos a combinação xOR entre as mensagens e por fim, realizar a etapa CNC.

O foco deste trabalho está na implementação da primeira etapa. Um grande obstáculo na aplicação dela é a necessidade de sincronia entre os dois nós que desejam compartilhar a mensagem. Isto é necessário para que os símbolos das mensagens que chegam ao nó *relay* se somem da forma esperada.

O fato dos sinais que chegam no nó R deverem estar sincronizados implica que as compensações, que normalmente são realizadas na recepção do sinal, devem ser realizadas antes mesmo da transmissão pelo canal. Ou seja, os transmissores da etapa PNC são responsáveis por realizar as diversas compensações do sistema de comunicação. Isto é possível somente através de uma estimativa prévia dos desvios.

Para realização de uma estimativa, os nós A e B realizam uma curta

comunicação com o nó R em instantes separados. Este último estima os parâmetros e os retorna aos respectivos nós via cabeçalhos da camada MAC ou superior, visto que os cabeçalhos da camada física são muito limitados em tamanho. Isto pode ser feito de forma semelhante a figura 1.1a ou até mesmo a 1.1b, mas com envio de mensagens curtas.

Uma segunda possibilidade é cada nó destino (A e B) estimar os parâmetros após o recebimento de uma mensagem do nó R , entretanto ocorre um *delay* de processamento diferente entre os nós destino o que pode vir prejudicar na realização da técnica.

Assim sendo, visto que a estimação ocorrerá no nó R , neste trabalho a estimação de cada parâmetro de sincronização e do canal é tratada de forma separada a correção dos mesmos visto que elas devem ocorrer em momentos e lugares separados.

1.2 Objetivos

Como principal objetivo para este trabalho temos a listagem e implementação de técnicas de estimativas e compensações de canal em um sistema OFDM, mas de forma separada uma da outra afim de cumprir alguns requisitos do sistema PNC.

Como objetivo secundário temos um projeto de implementação em plataforma SDR de um sistema de comunicação o qual possa ser compartilhado com a comunidade de forma a socializar o conhecimento.

1.3 Itens Fora do Escopo do Trabalho

Apesar dos seguintes itens complementarem a implementação proposta, eles desviam do foco principal que é direcionado a estimação e compensação de problemas físicos do canal.

- Camada MAC;
- Modulações lineares e codificação de canal;
- Detecção de constelações expandidas (PNC);
- Preocupações referentes aos símbolos e sequências predefinidas do 802.11 (ex.: Preâmbulo) devido ao uso do PNC.

1.4 Organização do Trabalho

Este trabalho foi escrito de forma a iniciar pelos conceitos mais básicos e abrangentes, depois para as suas peculiaridades para então finalizar com a implementação e os resultados.

Desta forma o capítulo 2 introduz o modelo simplificado do canal, segue com o conceito de OFDM que é sucedido pela técnica do prefixo cíclico e por um modelo mais completo de canal.

O capítulo 3 descreve do que consiste a camada física do 802.11 e a forma com que ela é utilizada para realizar as estimações do modelo do canal completo.

Por fim o capítulo 4 introduz a plataforma utilizada, depois quais os problemas que existem para implementação de algumas métodos, como foram implementados os outros métodos e o último tópico mostra os resultados obtidos.

1.5 Considerações Matemáticas

Devido a quantidade de documentos em inglês o qual este trabalho se baseia, foi optado por utilizar o *ponto* (.) como separador decimal em vez da *vírgula* (,) como é previsto pelo padrão brasileiro.

De forma a reduzir a quantidade de elementos matemáticos mas em troca do rigor, todos subconjuntos descritos por *colchetes* ([]) deste trabalho pertencerão ao grupo dos números inteiros (\mathbb{Z}). Ex.: $[-2, 3] = \{-2, -1, 0, 1, 2, 3\}$.

Os sinais discretos os quais são representados por letras maiúsculas, estão no domínio da frequência, não importando a letra do argumento. Ex.: $H[n]$.

Os sinais os quais o argumento está entre parênteses "()", são sinais definidos em qualquer argumento pertencente aos reais mesmo que o sinal seja do tipo $s(nT)$ onde $n \in \mathbb{Z}$ e $T \in \mathbb{R}$ é uma constante. Ainda no caso deste exemplo, os valores intermediários entre os instantes amostrados, são considerados como valores interpolados pela fórmula de interpolação de Whittaker–Shannon (ideal).

Sistemas Baseados em OFDM

Devido aos múltiplos caminhos que um sinal geralmente percorre em meios sem fio, o canal se torna seletivo em frequência fazendo com que ocorra um espalhamento temporal dos símbolos. Por sua vez, este espalhamento pode acabar se estendendo sobre os símbolos subsequentes, fenômeno chamado de Interferência Intersimbólica, do inglês *Intersymbol Interference* (ISI) [8].

Tradicionalmente em sistemas de portadora única, este efeito é combatido com o uso de equalizadores longos que tentam inverter ou atenuar os efeitos do canal. Entretanto, para obter um desempenho aceitável em sistemas muito seletivos, o projeto e implementação destes equalizadores pode tornar-se problemático.

Desta forma surge a Multiplexação por Divisão Ortogonal de Frequência, do inglês, *Orthogonal Frequency-Division Multiplexing* (OFDM). Esta é uma técnica de modulação digital amplamente utilizada nos sistemas de comunicação atuais. A técnica se baseia na ideia de multiplexação em frequência onde são enviados múltiplos símbolos em diferentes portadoras. Uma possível analogia seria com um sistema que antes transmitia dados de forma serial e agora os transmite de forma paralela.

Para superar a seletividade do canal, a modulação OFDM propõe a divisão da banda disponível em várias sub-bandas através do uso de portadoras ortogonais entre si, de forma que em cada uma delas ocorra um desvanecimento plano. A Fig. 2.1 ilustra esta característica. Esta estratégia permite o uso de equalizadores menos complexos para cada subcanal.

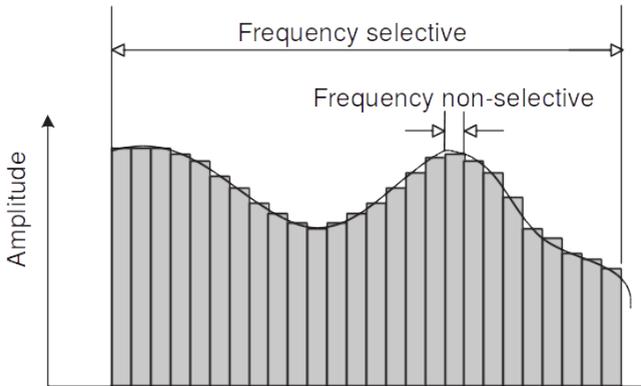


Figura 2.1: Canal seletivo dividido em sub-bandas

Fonte: [5], p. 116

Outras vantagens deste método de modulação vêm das propriedades matemáticas do par de transformações utilizados para sua geração e demodulação. As transformadas em questão são a *Inverse Discrete Fourier Transform* (IDFT) e a *Discrete Fourier Transform* (DFT) respectivamente.

2.1 Modelo Simplificado do Canal

O canal de um sistema de comunicação sem fio, se estudado profundamente, é algo extremamente complexo. Além de todos os efeitos causados pelo meio em que o sinal se propaga, ainda existem os efeitos provenientes das imperfeições dos aparelhos transmissores e receptores, chamados de imperfeições de *hardware*. Assim, desenvolver e aplicar soluções para todos ou grande parte destes problemas seria inviável para este trabalho.

Foram assumidas algumas simplificações e suposições para esta implementação. Nesta seção está exposto um modelo genérico e simplificado do canal e na seção 2.4, há um detalhamento dos problemas de *hardware* que causam efeitos que podem vir a impossibilitar a implementação.

Resumidamente, os comportamentos assumidos para este canal são os seguintes:

- Se comporta como um canal seletivo em frequência;
- Sua resposta ao impulso possui valor de espalhamento temporal τ conhecido;
- É invariante no tempo durante o intervalo que um frame é transmitido, ou seja, possui desvanecimento lento;
- Não possui nulos espectrais dentro da faixa utilizada;
- Possui desvanecimento plano em cada subportadora;
- Realiza adição de ruído gaussiano branco.

Como consequência dessas hipóteses, o canal em banda base pode ser modelado por um filtro discreto do tipo Resposta ao Impulso Finita, do inglês *Finite Impulse Response* (FIR) e com *taps* complexos $h[n]$, seguido por adição de um ruído gaussiano branco, cuja variável aleatória é representada por η . A Fig. 2.2 ilustra o modelo descrito.

Uma primeira observação é que dado o valor do espalhamento temporal do canal τ e o valor do tempo de amostragem no receptor T_a , o número de amostras que compõem a resposta ao impulso do canal será $G = 1 + \lfloor \frac{\tau}{T_a} \rfloor$. Assim o canal é representado por um filtro $h[n]$, $0 \leq n \leq 1 - G$.

Uma outra observação é que, como o ruído é amostrado, ele perde as componentes de mais alta frequência e se torna um ruído colorido sob o ponto de vista de sinais contínuos.

Matematicamente o sinal recebido no domínio do tempo discreto é dado por (2.1).

$$y[n] = x[n] * h[n] + \eta[n]; \quad (2.1)$$

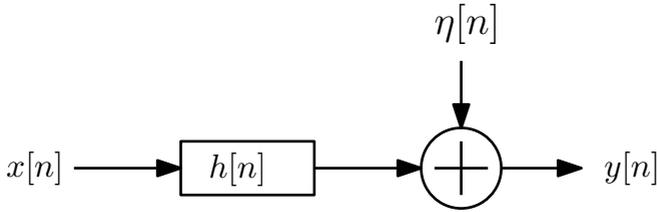


Figura 2.2: Modelo simplificado do canal

2.2 Princípios de uma Transmissão OFDM

Símbolos OFDM consistem de um bloco de dados modulados em uma mesma constelação, os quais estão situados em portadoras ortogonais e distintas. Nesta modulação, cada portadora que carrega informação é chamada de subportadora. A figura 2.3b, que mostra a Densidade Espectral de Potência, do inglês *Power Spectral Density* (PSD) típica de um sinal OFDM, elucida esta característica. Nela, cada pico equivale a um símbolo sendo transmitido em uma das subportadoras.

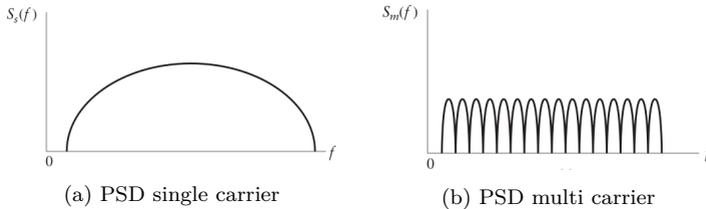


Figura 2.3: PSD de sinais single carrier e multi carrier

Fonte: [9], p. 302

O sistema OFDM divide um canal de banda $B = 1/T_a$ em N sub-canais espaçados de $\Delta_f = B/N$. Significa que uma transmissão que era realizada com uma alta taxa ($R_a = 1/T_a$), agora é feita com uma taxa N vezes menor ($R_s = R_a/N$). Isto é devido ao fato que a transmissão serial dos dados foi trocada por uma transmissão paralela. Assim a duração de um símbolo OFDM deve ser $T_s = NT_a$ mas a taxa líquida de transmissão de dados permanece a mesma.

Para geração deste sinal em banda base, a *stream* de símbolos a ser enviada $X(nT_a)$ é dividida em blocos de N símbolos $X_k(mT_s)$, onde

$k \in [0, N - 1]$. Em cada bloco então, cada símbolo é transladado para dentro de cada uma das sub-bandas.

Todavia, na modulação OFDM, o pulso que formata cada símbolo é retangular. Desta forma, o espectro de cada símbolo tem formato da função sinc a qual é infinita, ou seja, ocorre sobreposição dos espectros. Porém, na posição do espectro onde um símbolo está situado e é amostrado, existe um nulo das sinc de todos os outros símbolos. A figura 2.5 mostra como os espectros se sobrepõem e a figura 2.4 mostra a estrutura básica deste sistema.

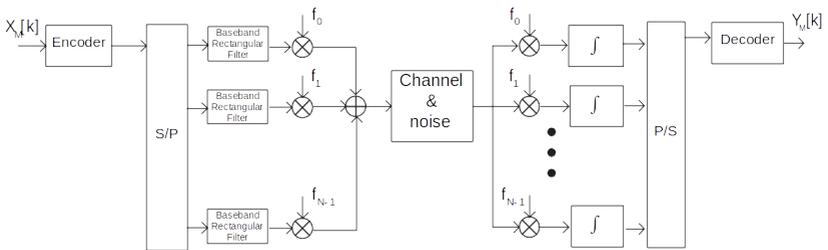


Figura 2.4: Esquema de geração de símbolos OFDM

Fonte: Adaptado de [5], p. 118

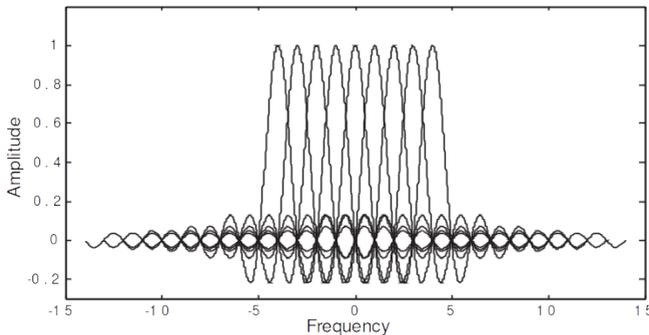


Figura 2.5: Espectro de um sinal OFDM (escala linear)

Fonte: [5], p. 118

A representação matemática de um símbolo OFDM em tempo con-

tínuo e em banda base é dado por (2.2).

$$x(t) = \sum_k X_k e^{i2\pi k \Delta_F t} \quad 0 \leq t < T_s \quad (2.2)$$

onde Δ_F é a diferença de frequência entre as subportadoras, N é o número de subportadoras, X_k são os símbolos em banda base que serão modulados em um símbolo OFDM e $k \in [0, N - 1]$ é o índice que define a frequência de cada subportadora. O valor mínimo da diferença de frequência entre subportadoras que transportem símbolos de qualquer constelação linear e com ocorrência de ortogonalidade é $\Delta_F = \frac{1}{T_s}$. Este valor é obtido em detalhes na seção A.2.

Pode-se amostrar esta expressão em $t = nT_a$ e assim obtemos a seguinte expressão do sinal em banda base.

$$x(nT_a) = \sum_k X_k e^{i2\pi \frac{kn}{N}} \quad 0 \leq t < T_s \quad (2.3)$$

Note que esta expressão é a equação da IDFT que pode ser vista na seção A.1, equação (A.2), e nos mostra que um sinal OFDM pode ser gerado por esta transformada.

Uma comunicação baseada em OFDM então consistirá das seguintes etapas:

- Um sinal, já modulado com uma modulação linear em banda base, é dividido em blocos de N símbolos;

$$X[n] \rightarrow X_k[m], \quad 0 \leq k \leq N - 1$$

- Então em cada bloco é aplicada a *Inverse Fast Fourier Transform* (IFFT) de N pontos;

$$X_k[m] \xrightarrow{IFFT_N} x_k[m]$$

- Após, as amostras são re-enfileiradas;

$$x_k[m] \rightarrow x[n]$$

- O sinal é gerado em banda base através de um Conversor Digital para Analógico, do inglês *Digital to Analog Converter* (DAC) com frequência T_a ;

$$x[n] \xrightarrow{DAC_{T_a}} x(t)$$

- Por fim o sinal é transladado para banda passante e transmitido.

$$x(t) \xrightarrow{LO} s(t)$$

Note que m é o índice do símbolo OFDM e possui taxa de amostragem N vezes menor que o índice n que é associado as amostras. A recepção do sinal é realizada de forma análoga e inversa, consistindo das seguintes etapas:

- O sinal receptado é transladado para banda base.

$$r(t) \xrightarrow{LO} y(t)$$

- O sinal resultante então é amostrado com frequência T_a através de um Conversor Analógico para Digital, do inglês *Analog to Digital Converter* (ADC);

$$y(t) \xrightarrow{ADC T_a} y[n]$$

- Estas amostras são agrupadas em blocos de N amostras;

$$y[n] \rightarrow y_k[m], 0 \leq k \leq N - 1$$

- Então, de forma a inverter a $IFFT_N$, em cada bloco é aplicada a *Fast Fourier Transform* (FFT) de N pontos;

$$y_k[m] \xrightarrow{FFT_N} Y_k[m]$$

- Por fim, os símbolos são re-enfileirados;

$$Y_k[m] \rightarrow Y[n]$$

A Fig. 2.6 elucida estas etapas.

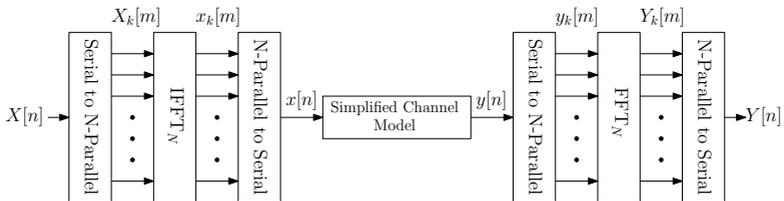


Figura 2.6: Sistema OFDM simplificado

Note que devido ao uso deste par de transformadas, o sinal antes da IFFT ou depois da FFT é dito pertencente ao domínio da frequência,

enquanto o sinal que é transmitido pelo canal é dito no domínio do tempo.

Considerando um caso particular do canal quando não ocorre espalhamento no sinal, ou seja, quando o canal só induz um ganho e uma defasagem constante em todas as amostras, ao realizar uma simples multiplicação pelo inverso desta constante obteremos no receptor exatamente o sinal transmitido. Um outro cenário consiste de um canal Ruído Gaussiano Branco Aditivo, do inglês *Additive White Gaussian Noise* (AWGN). Neste caso, os símbolos recebidos estarão corrompidos pela DFT do ruído que será representada pela variável aleatória $\Gamma[n]$.

Entretanto em um caso que o canal espalhe o sinal no tempo, ocorrem efeitos de interferência os quais não são trivialmente resolvidos através dos métodos clássicos. Para simplificar este problema, utiliza-se o método visto na seção a seguir.

2.3 Prefixo Cíclico

De forma a possibilitar uma equalização simples e assim eliminar a ISI causada pelo espalhamento temporal do sinal, utiliza-se a técnica de inserção do Prefixo Cíclico, do inglês *Cyclic Prefix* (CP). Os efeitos causados por este espalhamento podem ser divididas em dois.

O primeiro é o espalhamento proveniente do símbolo OFDM anterior, ou seja, uma interferência entre símbolos OFDM. Através do uso de um Intervalo de Guarda, do inglês *Guard Interval* (GI), entre estes símbolos (Fig. 2.7) é possível eliminar este problema.

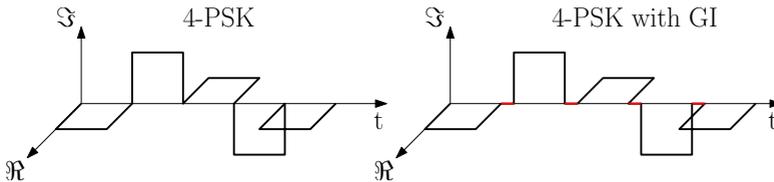


Figura 2.7: Intervalo de guarda entre símbolos

O segundo efeito diz respeito as interferência entre as amostras de um mesmo símbolo OFDM. Este efeito causa perda de ortogonalidade entre as subportadoras e é compensado através da equalização do sinal.

Na equação (2.1) vemos que o canal realiza a convolução linear com

o sinal transmitido. De forma a obter a expressão (2.4) ao realizarmos a FFT do sinal recebido, necessitamos que esta convolução linear (A.3) se comporte como uma circular (A.4).

$$Y[k]_m = X[k]_m \cdot H[k]_m + \Gamma[k]_m; \quad (2.4)$$

onde m é o índice do símbolo OFDM e $\Gamma[k]_m$ é a transformada discreta de Fourier de $\eta[n]$ aplicada ao símbolo m .

Note que nesta expressão, o canal $H[k]$ realiza uma simples multiplicação em cada símbolo $X[k]$, fazendo com que a equalização precise considerar apenas o valor do símbolo atual.

Lembrando que o canal é representado por um vetor $h[n]$, $0 \leq n \leq G - 1$, para então realizar a convolução circular neste cenário, é necessário que cada vetor que representa um símbolo OFDM $x_k[m]$ seja acrescido de G amostras de sua cauda em seu começo, resultando em um símbolo de tamanho $G + N - 1$

Desta modo, em (2.5) está expressado matematicamente o novo símbolo estendido e na Fig. 2.8 ele está representado graficamente.

$$\tilde{x}[l][m] = \begin{cases} x[l + (N - G)][m] & l \leq G - 1; \\ x[l - G][m] & l \geq G; \end{cases} \quad 0 \leq l \leq G + N - 1; \quad (2.5)$$

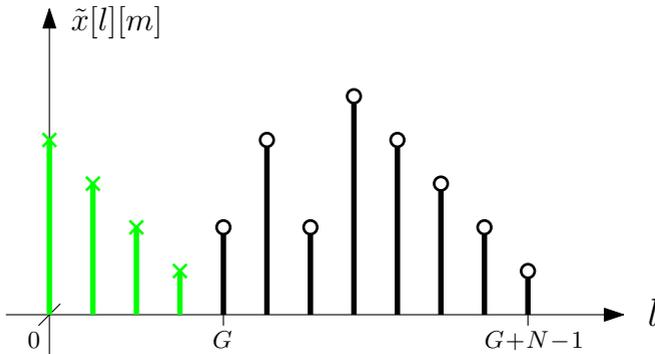


Figura 2.8: Símbolo precedido do CP

2.4 Perturbações no Sinal pelo *Hardware*

Devido a impossibilidade de conseguirmos a perfeita igualdade de características em componentes com as mesmas funções (clonados), existem duas complicações que são esperadas. A primeira é a diferença de frequência e fase entre os osciladores que elevam e abaixam a frequência do sinal. A segunda é a diferença de frequência e fase dos sinais de *clock* que controlam os geradores e amostradores de amostras.

Devido a limitações físicas e de *hardware*, um sinal também possui uma terceira problemática que se refere ao atraso entre início da transmissão e início da recepção. Todos os três problemas são encarados como imperfeições do canal e estão detalhados nesta seção.

2.4.1 Desalinhamento de Portadora

O desalinhamento de portadora pode ser separado em desalinhamento de frequência e desalinhamento de fase. Sobre o desalinhamento de fase, também chamada de *Carrier Phase Offset* (CPO), ela consiste de uma multiplicação por uma constante complexa $e^{i\phi}$ tanto no domínio do tempo, como no domínio da frequência. Desta forma ela pode ser considerada e inserida no modelo de resposta em frequência do canal.

Ocorrendo tanto pelo efeito Doppler como pela dessincronização dos osciladores que geram as portadoras do transmissor e do receptor, o desalinhamento de frequência de portadora, também chamado de *Carrier Frequency Offset* (CFO), causa um deslocamento espectral do sinal recebido. Devido ao fato dos símbolos serem transmitidos em subportadoras cujos espectros se sobrepõem, um erro de alinhamento pode ser crítico. Mais precisamente, ocorrerá uma amostragem do símbolo digital em uma posição deslocada, o que acarreta em uma espécie de interferência entre símbolos. Esta interferência é diferente da que ocorre devido ao espalhamento do canal e é chamada de *Inter Carrier Interference* (ICI). Na Fig. 2.9 está exemplificado este conceito.

No caso deste desalinhamento não ser corrigido, ao realizar a FFT obteremos um deslocamento contínuo e uniforme total no sinal. Temos

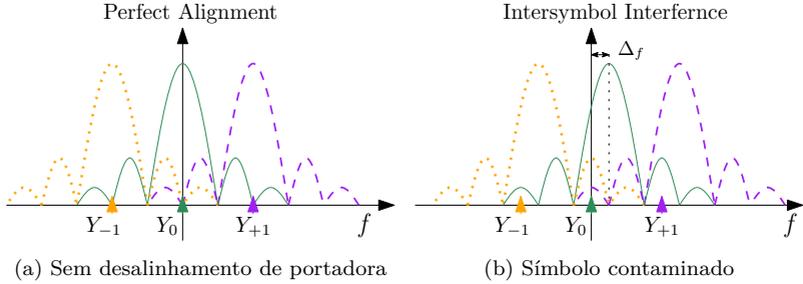


Figura 2.9: Efeito do desalinhamento de portadora

assim:

$$\begin{aligned}
 X[n] &\xrightarrow{\mathcal{F}^{-1}} x(nT_a); \\
 x(nT_a) &\xrightarrow{DAC} x(t); \\
 x(t) \cdot e^{i2\pi t\Delta_f} &= y(t); \\
 y(t) &\xrightarrow{ADC} y[n]; \\
 y[n] &\xrightarrow{\mathcal{F}} X(nF_d - \Delta_f).
 \end{aligned}$$

onde T_a é a taxa de amostragem, \mathcal{F} é a transformada de Fourier, Δ_f é a diferença de frequência entre portadoras e F_d é o passo da frequência discreta e equivale a $F_d = (2T_a N)^{-1}$ onde N é a quantidade de pontos escolhidos da DFT.

Em suma, uma multiplicação por exponencial no tempo resulta em deslocamento na frequência e assim o sinal recebido $Y[n]$ é representado em (2.6).

$$\begin{aligned}
 y(t) &= x(t) \cdot e^{i2\pi t\Delta_f} \\
 Y[n] &= X(nF_d - \Delta_f)
 \end{aligned} \tag{2.6}$$

2.4.2 Desalinhamento de Amostra

De forma semelhante ao desalinhamento de portadora, o desalinhamento de amostra também ocorre por uma dessincronização, entretanto, entre os sinais de *clock* do DAC e do ADC. Esta dessincronização é separada em dois parâmetros que ocorrem por motivos diferentes. O

primeiro consiste da defasagem entre os sinais de *clock*. Este é encontrado na literatura com o termo *Sampling Phase Offset* (SPO), mas neste trabalho é chamado de desalinhamento de instante de amostragem. O segundo é a diferença de frequência (fundamental) entre os sinais de *clock*. Na literatura é chamado *Sampling Frequency Offset* (SFO) mas neste trabalho é chamado de desalinhamento de frequência de amostragem.

Desalinhamento de Instante de Amostragem

Devido a atrasos entre a transmissão e recepção do sinal, inicialização dos pulsos de *clock* em instantes diferentes e entre outros, existirá o desalinhamento Δ_t do instante de amostragem do DAC em relação ao ADC. Para entender a influência deste problema, iremos simplificar o sistema para uma comunicação digital em banda base (somente eixo dos reais) e com um código de linha *ON-OFF* o qual é implementável com apenas um par de conversores. Assumindo perfeita sincronia de frequência entre os conversores, a equação (2.7) descreve o efeito e a Fig. 2.10 exemplifica as alterações no sinal neste cenário.

$$y[n] = x(nT_a - \Delta_t) \quad (2.7)$$

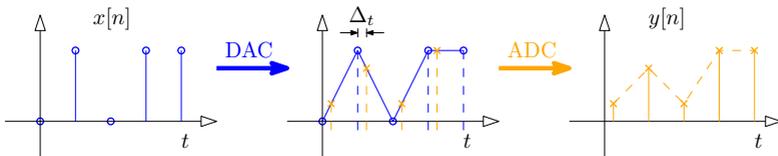


Figura 2.10: Desalinhamento de amostra

Nota-se que neste tipo de modulação ocorre uma espécie de interferência entre símbolos que altera seus valores. No entanto, ao transitar do eixo do tempo para o eixo da frequência através da FFT, os efeitos são outros. Comparando novamente com o efeito do desalinhamento de portadora e usando a ideia da dualidade da transformada de Fourier, espera-se que um deslocamento no tempo deva causar uma multiplica-

ção por exponencial na frequência. Temos assim

$$\begin{aligned} X[n] &\xrightarrow{\mathcal{F}^{-1}} x(nT_a); \\ x(nT_a) &\xrightarrow{DAC} x(t); \\ x(t - \Delta_t) &= y(t); \\ y(t) &\xrightarrow{ADC} y[n]; \\ y[n] &\xrightarrow{\mathcal{F}} X[n] \cdot e^{-i2\pi\Delta_t n F_d}; \end{aligned}$$

Onde T_a é a taxa de amostragem, \mathcal{F} é a transformada de Fourier, e F_d é o passo da frequência discreta e equivale a $F_d = (2T_a N)^{-1}$ onde N é a quantidade de pontos escolhidos da DFT. Logo o sinal recebido $Y[n]$ é representado em (2.8).

$$\begin{aligned} y(t) &= x(t - \Delta_t) \\ Y[n] &= X[n] \cdot e^{-i2\pi\Delta_t n F_d} \end{aligned} \quad (2.8)$$

Desalinhamento de Frequência de Amostragem

O sinal de *clock*, que é o responsável pela ativação do amostrador, geralmente é gerado por osciladores de cristal. Estes são componentes que possuem frequências bem definidas e menor variância de *jitter* em relação a outros métodos de geração. O problema é que, por menor que sejam as diferenças entre dois componentes, eles nunca estarão totalmente sincronizados e dessa forma sempre estará inserido um erro.

Podemos separar este problema em duas situações. A primeira pode ser vista como um desalinhamento no instante de amostragem que varia com o tempo. Temos então

$$\begin{aligned} y[n] &= x(nT_a - \Delta_{t2} \cdot n) \\ \Delta_{t2} &= \frac{f_{c2} - f_{c1}}{f_{c1} f_{c2}} \end{aligned}$$

Onde f_{c1} e f_{c2} são as frequências dos cristais do DAC (transmissor) e do ADC (receptor) respectivamente.

No entanto não foi investigado como esta expressão se comporta no domínio da frequência. Pode-se então aproximar por um desalinha-

pelo fato do atraso ser muito maior que o intervalo de símbolo. Para este trabalho e para maioria das aplicações, este parâmetro é irrelevante, visto que as amostras as quais não possuem as características de um frame são descartadas. Em implementações de PNC, este parâmetro se torna crítico e seu estudo deverá ser mais aprofundado.

2.5 Resumo do Modelo Completo do Canal

Em síntese, as seguintes grandezas são o foco deste trabalho e compõe parte do modelo do canal:

- Resposta em frequência, descrito na seção 2.1;
- CPO, descrito na seção 2.4.1;
- CFO, descrito na seção 2.4.1;
- SPO, descrito na seção 2.4.2;
- SFO, descrito na seção 2.4.2;
- Atraso de frame.

Observe que, como visto em 2.4.1, o CPO é modelado como parte da resposta em frequência do canal.

Estimação de canal sob o padrão IEEE 802.11

Um padrão ou protocolo consiste de um conjunto definido de regras e normas afim de assegurarem a confiabilidade e certas funcionalidades num processo. Neste caso, o processo é a transmissão de dados por um meio sem fio. Já como protocolo, temos o IEEE 802.11, amplamente aceito em ambientes de redes sem fio de computadores. Este protocolo oferece técnicas e serviços para os diversos estágios requeridos por este ambiente. Neste trabalho é utilizado um conjunto de regras específicas, orientadas para transmissão por este canal. Este conjunto também define como realizar as técnicas de estimacões de canal.

No presente capítulo, estão divididas em partes separadas as definições do padrão e a forma com que a estimacão é realizada. O objetivo é auxiliar na abstracão do conteúdo e reduzir a quantidade total de informacão de cada tópico.

3.1 Padrão IEEE 802.11

O padrão IEEE 802.11 é um protocolo que define a camada *Media Access Control* (MAC) e a Camada Física, do inglês *Physical-Layer* (PHY) do modelo OSI. Este último é um modelo conceitual que, sem

levar em conta as estruturas internas e tecnologias, caracteriza e padroniza as funções de comunicação de um sistema de telecomunicação.

Originalmente criado em 1997, o 802.11 foi desenvolvido especificamente para aplicações de redes de computadores sem fio. No entanto, com o passar dos anos, novas tecnologias foram surgindo no ramo de comunicações trazendo diversas vantagens em relação ao padrão original. Desta forma, foram criados padrões adendos. Estes consistiam basicamente de modificações que introduziam os novos métodos, mantendo grande parte do padrão original, na maioria das vezes a camada MAC, intactos. Dos mais comuns podem se citar o 802.11a, primeiro a utilizar modulação OFDM, 802.11b, 802.11g, 802.11n.

Devido a simplicidade, limitações de hardware e ao uso da modulação OFDM, esta implementação é baseada no subtipo a do padrão, mas com frequências dos canais do subtipo g que são mais baixas. Como o escopo deste trabalho é a realização física da comunicação, utilizamos somente a parte do protocolo do 802.11a que se refere a transmissão pelo canal, no caso a PHY.

O padrão define inicialmente os seguintes itens:

- Modulação utilizada deve ser OFDM de 64 subportadoras;
- Os índices das subportadoras vão de -32 a 31 (ver (2.2));
- As subportadoras $k \in [-32, -27] \cup [0] \cup [27, 31]$ são não-populadas, ou seja, são nulas em qualquer instante de tempo;
- Cada símbolo OFDM deve durar $3.2\mu\text{s}$;
- Cada símbolo deve ser precedido por $0.8\mu\text{s}$ de prefixo cíclico (ver seção: 2.3);
- Os possíveis canais para transmissão do sinal estão definidos em (3.1). Obs.: Estes canais são do subtipo g e alguns deles podem não ser disponíveis em alguns países;

$$f_{c_g} \in \{2412, 2417, 2422, 2427, 2432, 2437, 2442, \quad (3.1)$$

$$2447, 2452, 2457, 2462, 2467, 2472, 2484\} \quad [\text{MHz}]$$

3.1.1 Estrutura do Pacote

Como frame é o nome dado ao pacote da PHY e visto que utilizamos o protocolo 802.11a, o pacote a ser transmitido será chamado de frame 802.11a. A estrutura deste frame pode ser vista na Fig. 3.1.

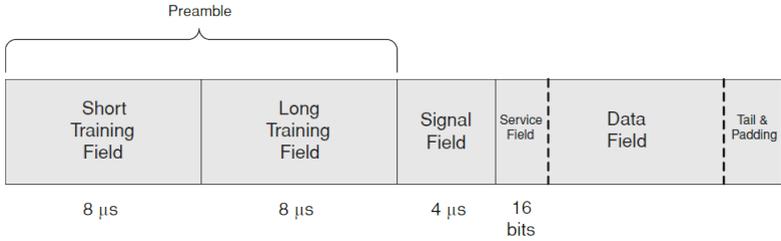


Figura 3.1: Estrutura do pacote 802.11a

Fonte: [11], p. 59

Short Training Field

O Campo de Treinamento Curto, do inglês *Short Training Field* (STF), é o primeiro campo do preâmbulo e consiste de uma seqüência periódica, no domínio do tempo, que é gerada por 10 repetições de um símbolo de $0.8\mu\text{s}$. Este símbolo foi escolhido por possuir boas propriedades de correlação e baixo *Peak-to-Average Power Ratio* (PAPR). Ele é representado no domínio da frequência como (3.2) [1]. Esta representação será chamada de Sequência de Treinamento Curta, do inglês *Short Training Sequence* (STS).

$$\begin{aligned}
 Y_{k \in [-26, 26]} = & \sqrt{13/6} \cdot \{0, 0, 1 + j, 0, 0, 0, -1 - j, 0, 0, 0, 1 + j, \\
 & 0, 0, 0, -1 - j, 0, 0, 0, -1 - j, 0, 0, 0, 1 + j, 0, 0, 0, 0, 0, \\
 & 0, -1 - j, 0, 0, 0, -1 - j, 0, 0, 0, 1 + j, 0, 0, 0, 1 + j, 0, 0, 0, \\
 & 1 + j, 0, 0, 0, 1 + j, 0, 0\}
 \end{aligned} \quad (3.2)$$

Após realizar a $IDFT_{64}$ desta representação, obteremos 4 repetições de um símbolo no domínio do tempo os quais estão exibidos na Fig. 3.2. Estas 4 repetições corresponderão a $3.2\mu\text{s}$ se gerados a uma taxa de

20MSamples/s. Note que cada símbolo deste será precedido do prefixo cíclico que, neste caso, corresponde a exatamente uma repetição do símbolo, totalizando assim $4\mu\text{s}$ e 5 repetições por STS.

O STF pode ser útil para detecção do sinal, estimativa de ganho para correção por um Controle Automático de Ganho, do inglês *Automatic Gain Control* (AGC), e estimação mais grossa de desvio de frequência.

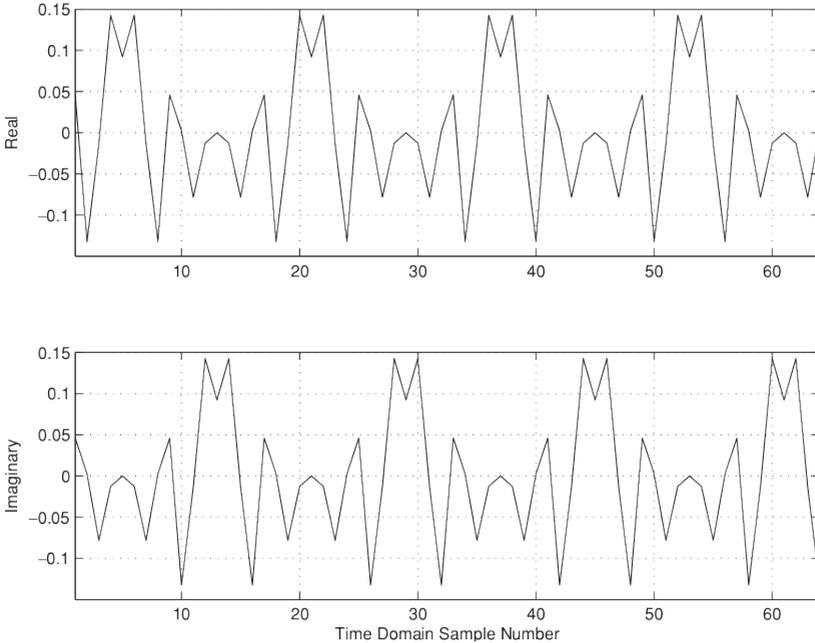


Figura 3.2: Construção do STF

Fonte: [11], p. 60

Long Training Field

O Campo de Treinamento Longo, do inglês *Long Training Field* (LTF), é o segundo campo do preâmbulo e consiste de uma sequência periódica, no domínio do tempo, que é gerada por 2.5 repetições de um símbolo de $3.2\mu\text{s}$. Esta representação será chamada de Sequência de Treinamento Longa, do inglês *Long Training Sequence* (LTS). Ele é representado no

domínio da frequência como (3.3) [1].

$$L_{k \in [-26, 26]} = \{1, 1, -1, -1, 1, 1, -1, 1, -1, 1, 1, 1, 1, 1, 1, -1, -1, 1, 1, -1, 1, -1, 1, 1, 1, 0, 1, -1, -1, 1, 1, -1, 1, -1, 1, -1, -1, -1, -1, -1, 1, 1, -1, -1, 1, -1, 1, 1, 1, 1\} \quad (3.3)$$

Para geração do LTF são gerados 2 símbolos no domínio do tempo (através da $IDFT_{64}\{LTS\}$) em sequência e posto um prefixo cíclico de 32 amostras (equivalentes a $1.6\mu s$ a taxa de 20MS/s). Note que, apesar deste prefixo não obedecer as regras dos outros prefixos, ou seja 1 prefixo antes de cada símbolo, o LTF como um todo ainda satisfaz a condição para convolução circular devido a sua periodicidade.

O LTF tem utilidade tanto no domínio do tempo quanto da frequência. Como o STF, através do LTF é possível realizar uma estimação de desvio de frequência, no entanto de forma mais fina. Por possuir cada subportadora populada, é possível também estimar a resposta em frequência do canal.

Signal Field

O Campo de Sinal, do inglês *Signal Field* (SIG), é um campo que carrega informações digitais. Este é composto por um único símbolo OFDM de dados. Um símbolo de dados, por sua vez, possui em cada subportadora uma certa representação digital, ou seja, uma modulação. Além disso, as subportadoras $k = -21, -7, 7, 21$ carregam um valor de amostra pré-definido, com finalidade de um posterior rastreamento da variação de fase e frequência. Estas são chamadas de subportadoras piloto.

O SIG carrega 24 bits de informação os quais são codificados com taxa $1/2$ e são transmitidos em somente um símbolo OFDM em que cada subportadora é modulada em *Binary Phase-Shift Keying* (BPSK). O motivo desta baixa configuração é pela garantia de recebimento destes importantes dados, tanto para a estação que irá receber a mensagem quanto para as outras estações que estão escutando o canal.

A estrutura do SIG está representada na Fig. 3.3. O primeiro campo indica a modulação e taxa do codificador os quais os bits de dados serão transmitidos. A tabela 3.1 mostra as opções possíveis. O

segundo campo é reservado, ou seja, sem função. O terceiro campo indica qual é a quantidade líquida de bytes (antes da codificação do canal) que serão transmitidos. O quarto campo contém a paridade dos 17 primeiros bits. O último campo consiste de 6 zeros.

Rate 4 bits	Reserved 1 bit	Length 12 bits	Parity 1 bit	Tail 6 bits
----------------	-------------------	-------------------	-----------------	----------------

Figura 3.3: Estrutura do campo de sinal

Fonte: [11], p. 64

Tabela 3.1: Informações de Taxa do Frame 802.11a

Adaptado de [11], p. 64

Rate bit field	Data rate (Mbps)	Modulation	Code rate
1101	6	BPSK	1/2
1111	9	BPSK	3/4
0101	12	QPSK	1/2
0111	18	QPSK	3/4
1001	24	16-QAM	1/2
1011	36	16-QAM	3/4
0001	48	64-QAM	2/3
0011	54	64-QAM	3/4

Data Field

O campo de dados, do inglês *Data Field*, é o campo que efetivamente carrega a mensagem. Escolhendo uma configuração da tabela 3.1, o campo é construído da seguinte forma

- (i) São adicionados 16 zeros binários no início e 6 zeros ao fim da *stream* de dados. Se necessário serão adicionados outros zeros a fim de preencher um símbolo OFDM;
- (ii) A *stream* passa por um *scrambler*;

(mais informações do porque estão na seção 4.4). São eles o desvio de frequência da portadora, o desvio de instante de amostragem e a resposta em frequência do canal sobre cada subportadora.

3.2.1 Desvio de Frequência da Portadora

Para realizar a estimação inicial deste parâmetro, foi utilizado o LTF. Como os símbolos são definidos no domínio da frequência mas o utilizamos no domínio do tempo, utilizaremos a notação LTS_t para enfatizar esta propriedade. Afim de detectarmos e numerarmos corretamente cada amostra do frame recebido, convoluímos a LTS_t , com todo pré-amplulo recebido. Esta operação resulta na Fig. 3.4.

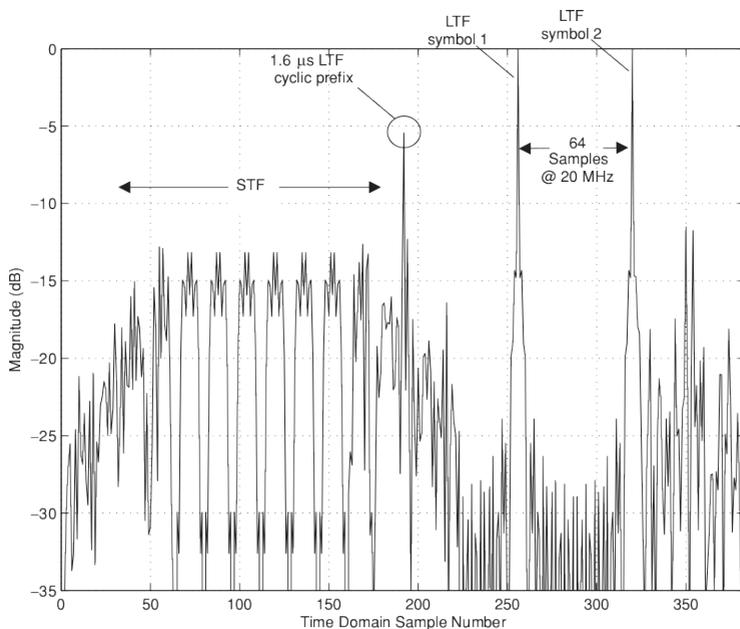


Figura 3.4: Correlação da LTS com o Pré-amplulo

Fonte: [11], p. 63

Os dois maiores picos deste gráfico indicam exatamente quando a LTS esperada está alinhada com as duas LTS recebidas.

A ideia por trás desta estimação é que os símbolos da primeira sequência devem ser iguais aos da segunda a menos de uma rotação

de fase, a qual é associada ao desvio de frequência. Significa que $LTS_{t_2}[n] = LTS_{t_1}[n]e^{i\theta} \rightarrow LTS_{t_2}[n]\overline{LTS_{t_1}[n]} = e^{i\theta}$, onde $\overline{\cdot}$ representa a operação de conjugação. Assim, para uma primeira aproximação, podemos realizar a média de todos os ângulos da seguinte forma:

$$\theta = \frac{1}{64} \sum_{n=0}^{63} \arg(LTS_{t_2}[n]\overline{LTS_{t_1}[n]})$$

No entanto se realizarmos o somatório de $LTS_{t_2}[n]\overline{LTS_{t_1}[n]}$ primeiro e depois calcularmos o argumento, realizaremos uma média ponderada dos ângulos, o que é interessante no sentido que o ruído afeta menos as amostras de maior amplitude. Em (3.6) está representado matematicamente este procedimento.

$$\theta = \arg\left(\sum_{n=0}^{63} LTS_{t_2}[n]\overline{LTS_{t_1}[n]}\right) \quad (3.6)$$

Note que θ é uma defasagem, em radianos, adicionada entre 64 amostras que são a separação entre uma amostra da LTS_{t_2} e a correspondente na LTS_{t_1} . Significa que o deslocamento angular que ocorre entre cada amostra do frame é $\theta/64$ rad. Então a estimação final do desvio de frequência é obtido pela equação (3.7).

$$\begin{aligned} \Delta f &= \frac{f_a \theta}{64} \quad [\text{rad/s}] \\ \Delta f &= \frac{f_a \theta}{2\pi 64} \quad [\text{Hz/s}] \end{aligned} \quad (3.7)$$

Onde $f_a = 1/T_a$ é a frequência/taxa de amostragem (5MS/s neste trabalho).

Note que o STF também pode ser utilizado da mesma forma para o mesmo fim. Este campo contém maior número de períodos, e assim, em vez de 64 amostras separando cada repetição, nele serão somente 16. Esta sequência possibilita uma obtenção de desvios de frequências maiores e de até $\pm 625\text{kHz}$ ($\theta = \pm\pi$) em frequência de amostragem padronizada (20MS/s), em comparação com os $\pm 156.25\text{kHz}$ da LTF. Contudo, esta última permite uma estimação mais precisa e o ideal então é utilizar ambas as sequências para esta estimação.

Por fim, podemos utilizar as portadoras piloto que estão inserida

dentro dos símbolos de dados para poder rastrear o desvio de frequência que ocorre dentro do frame. O diagrama da Fig. 3.5 elucida um frame OFDM e é útil para entender esta técnica.

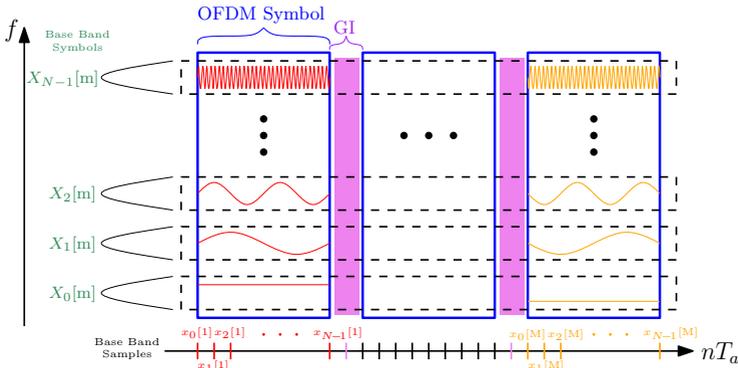


Figura 3.5: Frame OFDM

Considerando que a diferença de frequência dentro de um símbolo OFDM recebido causa uma defasagem irrisória, podemos assumir que cada símbolo deste possui uma defasagem média constante em todas as suas amostras. Ou seja, cada amostra dum mesmo símbolo OFDM é multiplicado por um $e^{i \cdot \phi[m]}$, onde m é o índice deste símbolo. Matematicamente temos que o sinal recebido é $y_k[m] = x_k[m] \cdot e^{i \cdot \phi[m]}$.

Note que como a DFT é performada dentro de um símbolo OFDM, e devido a esta defasagem ser constante neste intervalo, pela propriedade de linearidade da transformação, o sinal transformado possuirá a mesma multiplicação, ou seja $Y_k[m] = X_k[m] \cdot e^{i \cdot \phi[m]}$. Isto é necessário visto que para estimação de fase dentro de um símbolo de dados, precisamos de valores conhecidos e assim, só podemos utilizar o sinal no domínio da frequência e as subportadoras piloto.

Note que o desvio de frequência pode ser obtido através da diferença de fase entre dois símbolos adjacentes visto que $2\pi\Delta_f T_s = \Delta\phi = \phi[m] - \phi[m-1]$, onde T_s corresponde a duração do símbolo.

Para realizar a estimação desta diferença de fase entre cada símbolo OFDM, utilizamos as subportadoras piloto recebidas do símbolo atual e do passado. A operação realizada consiste da média das medições de diferença de fase dos 4 símbolos piloto e é descrita pela equação (3.8).

Note que esta equação leva em conta a polaridade dos símbolos.

$$\begin{aligned} \phi[m] - \phi[m-1] = \arg \left(\right. \\ & p[m]Y_{-21}[m] \cdot \overline{p[m-1]Y_{-21}[m-1]} + \\ & p[m]Y_{-7}[m] \cdot \overline{p[m-1]Y_{-7}[m-1]} + \\ & p[m]Y_7[m] \cdot \overline{p[m-1]Y_7[m-1]} + \\ & \left. p[m]Y_{21}[m] \cdot \overline{p[m-1]Y_{21}[m-1]} \right) \end{aligned} \quad (3.8)$$

onde $p[m]$ é a polaridade das subportadoras piloto no símbolo m .

3.2.2 Resposta em Frequência do Canal

Conforme visto na seção 2.1, o canal é modelado por um filtro discreto e de tamanho finito. Devido ao uso do CP o canal induz uma multiplicação por um valor complexo em cada amostra do domínio da frequência. Desta forma, a resposta em frequência do canal é representada por $H_k \in \mathbb{C}$, onde $k \in [-26, -1] \cup [1, 26]$ visto que os outros *taps* são não populados.

Para sua estimação, é utilizado os símbolos da LTF no domínio da frequência. Primeiramente é realizada a média entre as duas LTS recebidas afim de obter uma $\widehat{LTS}[k]$ com nível do ruído reduzido em 3dB. Como pode ser visto em (3.3), os símbolos possuem um valor unitário e conhecido em cada subportadora relevante. Isto significa que simplesmente multiplicar a LTS recebida pela esperada, retorna o valor de cada *tap* $\hat{H}[k]$ da estimação da resposta em frequência do canal.

$$\begin{aligned} \widehat{LTS}[k] &= LTS[k] \cdot H[k] + \frac{\Gamma[k]}{2} = \frac{LTS_1[k] \cdot H[k] + LTS_2[k] \cdot H[k] + \Gamma[k]}{2} \\ LTS[k] \cdot LTS[k] &= \begin{cases} 1 & k \in [-26, -1] \cup [1, 26] \\ 0 & \text{Caso contrário} \end{cases} \\ \hat{H}[k] &= H[k] + \frac{\Gamma[k] \cdot LTS[k]}{2} = \widehat{LTS}[k] \cdot LTS[k] \quad k \in [-26, -1] \cup [1, 26] \end{aligned}$$

Note que é utilizado a multiplicação dos vetores em vez da divisão pois poupa processamento. Note também que esta técnica só funciona devido a forma com que a LTS foi projetada.

3.2.3 Desvio de Instante de Amostragem

De forma semelhante a estimação de desvio de frequência, esta estimação também mede uma diferença de ângulo, no entanto no domínio da frequência. Para realizar esta estimação, utilizamos o LTF e as subportadoras piloto.

Como dentro de um frame é comum existir uma variação lenta deste desvio, é necessário realizar constantemente a estimação. Isto justifica o uso das subportadoras piloto que cada símbolo OFDM carrega.

O método consiste da obtenção da diferença de fase entre as portadoras piloto de um símbolo de dados recebido $Y_k[m]$. As três primeiras foram definidas para possuir os mesmos valores enquanto a última possui o negativo deles. Lembrando que as subportadoras piloto estão separadas por 14 amostras da frequência, o argumento inserido entre amostras $\varphi[m]$ é calculado em (3.9).

$$\varphi[m] = \frac{1}{14} \arg(Y_{-21}[m] \cdot \overline{Y_{-7}[m]} + Y_{-7}[m] \cdot \overline{Y_7[m]} + Y_7[m] \cdot Y_{21}[m]) \quad (3.9)$$

Este ângulo é associada com o parâmetro buscado. Em (2.8) temos a expressão a qual os associa. Entretanto não é necessário conhecer o valor propriamente dito. Isto porque para correção também é utilizado o ângulo φ e assim o valor real do parâmetro acaba não tendo aplicação para este trabalho.

Implementação na Prática e Resultados

Devido a complexidade e ao grande número de tecnologias que um sistema de telecomunicação sem fio precisa para funcionar, sua implementação sempre foi uma tarefa árdua e custosa. Quase sempre quando se desejava aplicar um novo conceito ou teoria, se fazia necessário adaptar ou recriar algum protocolo e um hardware que o implementasse. Com o avanço das tecnologias de processamento nos últimos anos, se tornou possível realizarmos grande parte destes componentes de hardware em software. Em particular, a tecnologia de SDR que se beneficiou destes avanços, torna quase todo sistema implementável em software. Deste modo, a implementação deste trabalho é baseada neste conceito.

4.1 Rádio Definido por Software

Rádio definido por software é um paradigma relativamente antigo mas promissor da área de telecomunicações. Ele idealmente consiste da criação de uma estação de rádio qualquer através de um software implementado numa plataforma de processamento digital, um DAC ou um ADC, e uma antena. Isto significa que, com um mesmo aparelho, podemos realizar qualquer tipo de comunicação sem fio, desde analógi-

cas até aquelas que utilizam protocolos complexos como os utilizados em redes de computadores. Isto tudo é obtido apenas por trocar um software, revelando o quanto este sistema é flexível.

No entanto, existem várias dificuldades técnicas na implementação deste conceito. Estas impõem certas restrições e acabam exigindo uma mesclagem entre processamento digital e analógico (em hardware). O principal problema é referente a dificuldade de manipulação e processamento de sinais em alta frequência. Outro grande problema é referente as restrições de banda que ocorrem em qualquer antena.

Entretanto, mesmo com estas dificuldades, ainda é possível realizar o conceito. O que existe hoje são aparelhos intermediários que estão projetados para operar em uma certa faixa de frequências e que realizam a conversão entre banda base (ou intermediária) e banda passante. O processamento do sinal de rádio pode ser realizado virtualmente em qualquer processador que possua uma interface apta a se comunicar com o aparelho intermediário. Comumente este processador será de um computador comum o qual executa um sistema operacional e um programa de processamento de sinais.

A Fig. 4.1 mostra uma versão condensada do funcionamento do sistema. Nesta figura existe um dispositivo de processamento a mais, um *Digital Signal Processor* (DSP), o qual é bem comum, mas não necessário, neste tipo de aparelho. Este dispositivo é utilizada para converter as taxas de amostragem entre o computador e o ADC/DAC e também é geralmente utilizado para implementar os *Digital Down-Converter* (DDC) e *Digital Up-Converter* (DUC) que realizam uma conversão entre banda intermediária e base e centralizam o espectro. Note que o bloco que realiza a conversão entre banda passante e intermediária é o Oscilador Local, do inglês *Local Oscillator* (LO), junto aos filtros de sintonia e limitação de banda (*anti-aliasing*). Na parte receptora convém utilizar um *Low-Noise Amplifier* (LNA), que é um amplificador com atenuação de ruído, em vez de um amplificador comum como o utilizado na parte transmissora.

4.1.1 Sistema de SDR utilizado

Para realização deste trabalho utilizamos o conjunto de uma plataforma de rádio definido por software da *Ettus Research*, o *USRP N200*, e um computador comum o qual que executa o software *GNU Radio*

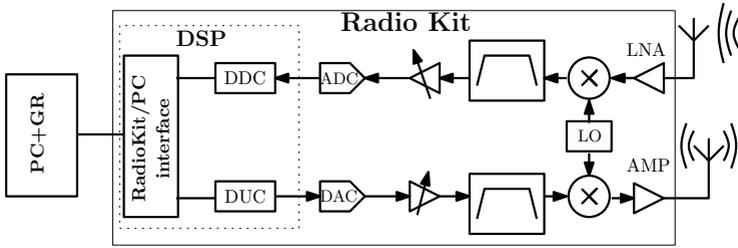


Figura 4.1: Sistema SDR

Companion (GRC) e *GNU Radio* (GR).

GNU Radio

O GRC é uma plataforma gráfica utilizada para desenvolver aplicações do software GR. O GR por sua vez, é um *toolkit* de código aberto que provê blocos de acesso/manipulação de hardware e de processamento de sinais com foco na implementação de sistemas de SDR. O GR é um *GNU Project* oficial, publicado pela primeira vez em 2001, que iniciou como um *fork* do código *Pspectra* criado no *Massachusetts Institute of Technology* (MIT). Ao longo destes anos e por ser um projeto aberto, ele já ganhou centenas de contribuições de pessoas de todo o mundo.

Este programa permite usar e cascatear vários objetos de processamento de sinais prontos a fim de facilitar a construção de um sistema mais sofisticado. Ele inclui desde blocos aritméticos e de geração de sinais, até blocos de acesso ao hardware, modulações digitais e algoritmos de codificação e decodificação. Além disso, é possível criar e adicionar novos blocos, entretanto, são necessários conhecimentos mais aprofundados nas linguagens C++ e Python.

Apesar desta plataforma se assemelhar ao *Simulink*[®], seu funcionamento interno é diferente devido a preocupação com a eficiência computacional e a performance em tempo real do sistema. A versão utilizada atualmente do GNU Radio é a 3.7.5.

USRP

Os hardwares USRP da *Ettus Research* (uma empresa da *National Instruments* desde 2010), fazem o papel do aparelho intermediário citado.

Estas plataformas foram desenvolvidos com o intuito de realizarem a interface entre sinal em banda base e banda passante. Elas nos permitem trabalhar com uma ampla faixa de frequências e considerável banda. Seu diagrama interno está representada na Fig. 4.2. Importante notar que nesta arquitetura estão implementados, tanto no transmissor quanto no receptor, duas portadoras em quadratura. Para isso, o aparelho se utiliza do *Phase-Locked Loop* (PLL) e do *Voltage-Controlled Oscillator* (VCO) que juntos atuam como dois LO com garantia de 90° de defasagem entre si. Outro detalhe é que o USRP utiliza um *Field-Programmable Gate Array* (FPGA) para realizar os pré-processamentos digitais.

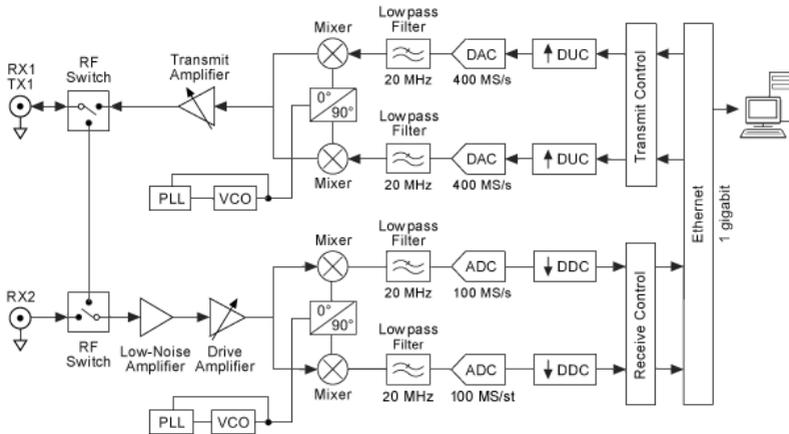


Figura 4.2: Diagrama a nível de sistema do USRP

Fonte: [2]

Da família de equipamentos USRP, o modelo *USRP N200 Rev 4.0* foi o utilizado para o desenvolvimento das aplicações. Este aparelho necessita de uma placa filha que realiza a interface rádio frequência (RF) do equipamento. É nela que o sinal é captado das altas frequências e transladado para banda intermediária ou base e vice-versa, enquanto o aparelho em si se preocupa com os pré-processamentos e a comunicação com o computador. A placa filha utilizada no trabalho é o modelo *SBX Rev 5.1* cujo é capaz de atingir frequências de até 4 GHz com até 40 MHz de banda.

4.2 Projeto WiME

Por consistir de uma grande quantidade de etapas, a implementação de um sistema que realiza a PHY do 802.11 demandaria muito tempo. Assim não iniciamos este trabalho do zero, o iniciamos a partir de um software do projeto WiME [3]. Este projeto consiste de um *framework* baseado em SDR para experimentação e medição de ambientes de redes sem fio. Dele utilizamos o software *gr-ieee802-11* o qual possui a implementação de um transmissor e de um receptor de frames 802.11a na plataforma GR e USRP. Esta implementação realiza o que foi citado na seção 3.1 e uma parte da camada MAC. Uma visão geral da implementação do receptor e dos blocos que o compõe pode ser encontrada em [4].

Para utilizar este software, precisamos definir alguns parâmetros referentes ao protocolo e outros parâmetros do próprio sistema. O modo de operação do 802.11 utilizado e que melhor se adaptou ao ambiente testado foi o seguinte:

- Frequência de centro: 2.484 GHz (outras possibilidades no conjunto (3.1));
- Modulação/Taxa: *Quadrature Phase-Shift Keying* (QPSK) $1/2$, ver tabela 3.1;

Como configuração de sistema, devemos definir alguns parâmetros necessários para seu correto funcionamento e cujos valores variam de acordo com o ambiente. São eles:

- Ganho do receptor e transmissor;
- Mensagem que será transmitida repetidamente;
- Período entre envios da mensagem;
- Tipo de equalizador do canal;

Por fim, utilizamos uma configuração que desobedece o padrão 802.11a, mas permite um melhor funcionamento na plataforma. Esta é a taxa de amostragem do aparelho. Devendo ser definida para 20 MS/s para que cada símbolo OFDM correspondesse a $3.2\mu\text{s}$, esta foi definida em 5 MS/s, o que resulta em um símbolo quatro vezes mais

comprido e subportadoras quatro vezes mais próximas. Contudo, além de não desobedecer nenhum limite da teoria, o processamento é reduzido em quatro vezes. Assim pudemos inserir novos processos dentro do software sem atingir o limite de processamento disponível.

4.3 Organização dos Dispositivos e Softwares

Para esta implementação, utilizamos um aparelho USRP para atuar como o transmissor e outro aparelho para atuar como receptor. Os testes mostraram que o software do transmissor consome muito pouco processamento em relação ao software do receptor, assim pudemos manter os aparelhos conectados ao mesmo computador o qual executa ambos os softwares. Esta prática tornou mais simples a comunicação necessária entre os programas.

4.3.1 Comunicação entre Estimador e Compensador

Atualmente todas as estimações que acontecem no receptor são gravadas em um arquivo de *log* no disco rígido. Cada estimacão é identificada por uma *tag* que indica qual o tipo da estimacão. Este arquivo então é processado e cada tipo de estimacão é separada em novos arquivos no formato *Comma-Separated Values* (CSV). O software de computacão numérica *GNU Octave* é utilizado para ler os arquivos e processar estes valores. Um *script* remove erros de gravacão, faz a média das leituras e retorna uma string no formato de tupla do Python. Finalmente, as tuplas são inseridas nos blocos de compensacão.

Nota-se que este serviço é praticamente todo manual. Em trabalhos futuros é previsto melhorias desta tarefa.

4.4 Desafios Práticos em Aberto

Limitacões do hardware, complicacões no uso da plataforma, problemas devido a modelagem simplificada e novos problemas não modelados são alguns exemplos de incômodos que aparecem na transiçãõ entre teoria e implementacão.

Sumarizando, do total de estimacões propostas neste trabalho, duas não puderam ser obtidas, e do total de compensacões, três não puderam ser implementadas.

4.4.1 Estimação do Atraso de Frame

Esta requer uma medida de tempo precisa (na casa dos microssegundos) entre a transmissão e recepção do frame. O padrão nos fornece bons meios de detectar a chegada de um frame. No entanto, na plataforma utilizada, além das dificuldades para criação do programa, é particularmente complicado obter valores precisos pois irão depender muito do estado atual do computador, visto que existe um sistema operacional controlando vários processos. Para a maioria das aplicações, incluindo a feita neste trabalho, ela é dispensável. Entretanto, para aplicações em sistemas PNC ela é importante.

4.4.2 Estimação do Desvio de Frequência de Amostragem

Este parâmetro foi estimado manualmente na plataforma e mostrou-se baixo. Não foi possível criar um método satisfatório para a implementação da estimação deste parâmetro e também sabe-se que é bem complicado de realizar a compensação. Deste modo, é um problema que não será abordado e não comprometerá o desenvolvimento deste trabalho.

4.4.3 Compensação do Atraso Entre Transmissores

Sobre as compensações, esta é específica para sistemas PNC e se refere a sincronia de frame entre dois transmissores. Para solucionar este problema, basta incluir um atraso num dos programas para que os frames cheguem em sincronia no receptor. No entanto, da mesma forma que na estimação, este atraso é intimamente ligado ao estado atual do computador e assim se torna complicado obter precisão. Também se utilizarmos dois computadores separados para cada transmissão, teremos uma segunda tarefa que seria criar alguma forma de acionar a transmissão sincronizadamente. Todas estas complicações, para serem tratadas, requererão conhecimentos aprofundados em áreas da computação e da plataforma que é utilizada.

4.4.4 Compensação do Desvio de Frequência de Amostragem

O desvio de tempo de amostragem é dependente do cristal que gera o pulso de *clock* em cada aparelho. Modificar a frequência dos pulsos não

é factível. Assim, quando o *clock* do receptor estiver suficientemente atrasado/adiantado do *clock* do transmissor, perderemos/duplicaremos uma amostra e conseqüentemente obteremos perda total do frame, visto que os códigos corretores não são projetados para corrigir este tipo de problema. Basicamente possuímos três possíveis soluções para contornar o problema. A primeira é utilizar um pulso de *clock* externo para todos os aparelhos o que neste caso eliminaria totalmente o problema. Entretanto esta opção desvia do conceito da implementação real de sistemas sem fio e da técnica PNC. A segunda é a utilização de uma família de códigos corretores mais sofisticados que possam identificar e corrigir este tipo de problema. Esta solução foge um pouco da simplicidade provida pelo protocolo. Também requererá um estudo mais aprofundado de viabilidade. A terceira é utilizar sobre-amostragem do sinal no lado do receptor e utilizar um outro conjunto de técnicas para identificação e correção do problema. Qualquer uma das três soluções necessitaria de tempo para pesquisa e implementação e, desta forma, não fazem parte do escopo deste trabalho.

4.4.5 Compensação de Desvio de Instantes de Amostragem

Esta última não pode ser implementada, visto os prazos de entrega do trabalho. O receptor do projeto WiME realiza esta compensação logo após a estimação do mesmo.

4.5 Compensações Implementadas

Visto as dificuldades descritas em 4.4, neste trabalho foram implementadas somente as compensações do canal e da diferença de frequência entre portadoras. Como realizamos uma estimação para cada realização do sistema (transmissão de múltiplos frames), compensação das mudanças rápidas dos parâmetros, como as que ocorrem dentro de um frame, não foram implementadas. Outro detalhe é que os blocos de compensação foram utilizados pelo programa de transmissão afim de validar alguns dos requerimentos do PNC.

As implementações feitas consistem de dois novos programas para o GR. Para gerar-los é necessário seguir um padrão. Primeiramente é necessário gerar um esqueleto que conterà os blocos que executam estes programas. Este é chamado de módulo. Depois são criados os blocos os

quais devem realizar operações bem definidas. No caso deste trabalho, o módulo é o conjunto de blocos de compensação do canal e cada bloco por sua vez realiza uma das compensações.

Para implementar um programa num bloco, geralmente precisamos editar três arquivos. O primeiro e segundo são programados em C++ e contêm a implementação e o cabeçalho. O terceiro já, é um arquivo no formato *eXtensible Markup Language* (XML) e descreve a representação gráfica do bloco no GRC. Instruções mais detalhadas deste processo podem ser vistas em [7].

4.5.1 Desvio de Frequência

Para realizar esta compensação, criamos um bloco o qual utiliza a estimação de desvio de frequência (Δ_f) para criar uma exponencial complexa com frequência invertida. Este bloco multiplica as amostras de sua entrada com esta exponencial de forma a compensar a rotação do canal. Por isto, ele deve estar localizado antes do bloco de transmissão ou depois do bloco de recepção. A Fig. 4.3 mostra os parâmetros de entrada e um exemplo de posicionamento do bloco.

Esta exponencial está representada em (4.1).

$$c[n] = e^{-i2\pi\Delta_f n T_a} \quad (4.1)$$

Onde T_a é o intervalo entre amostras.

De forma a permitir um ajuste em tempo real da frequência sem inserir descontinuidade de fase, utilizamos a seguinte técnica para geração desta exponencial.

$$\theta_e[n] = \theta_e[n-1] + 2\pi\Delta_f T_a \quad (4.2)$$

$$c[n] = e^{-i\theta_e[n]} \quad (4.3)$$

Note que é necessário armazenar o valor do ângulo utilizado na amostra anterior.

4.5.2 Distorção Linear do Canal

Como visto na equação (2.4), o canal no domínio da frequência realiza uma multiplicação por um escalar complexo em cada amostra. De posse

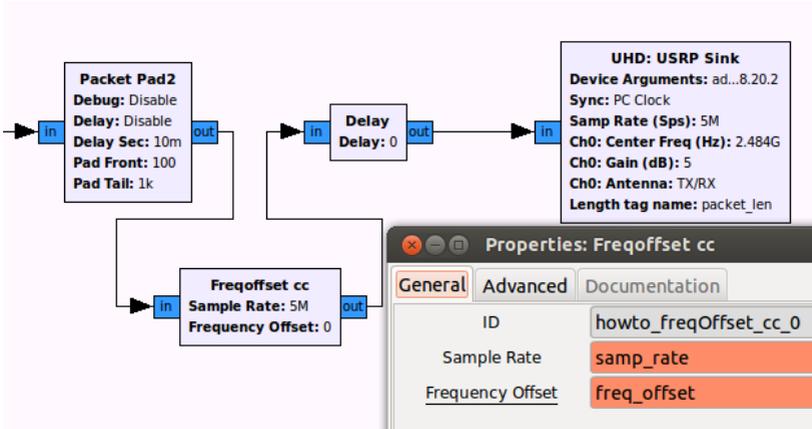


Figura 4.3: Bloco de compensação do desvio de frequência

da estimação do canal ($\hat{H}[k]$), para realizar a compensação é suficiente inverter cada elemento deste vetor e multiplicá-lo elemento a elemento com cada símbolo OFDM recebido ($Y[k]$). Este tipo de compensação é conhecida como equalização de força zero e está representada em (4.4)

$$\hat{X}[k] = \frac{Y[k]}{\hat{H}[k]} \quad (4.4)$$

Devido a inversão de elementos que são menores que um, este tipo de equalização, quando realizada na recepção, tem como consequência um ganho do nível do ruído. Para evitar o problema, podemos aplicar a equalização no lado do transmissor o qual possui o sinal íntegro. Entretanto, vale notar que nos casos com ocorrência de nulos espectrais, o transmissor realizará um alto ganho no sinal o que resultará em um gasto excessivo de potência e também, devido ao fato da própria estimação ser corrompida pelo ruído, obteremos um grande erro de imprecisão.

Para criar o programa que realiza a compensação, foi necessário programar um bloco que processa vetores. Isto porque as possíveis localizações deste bloco são antes da entrada do bloco da IFFT (transmissor) ou depois da saída do bloco FFT (receptor). Este bloco recebe como parâmetro o vetor de estimação do canal já invertido, ou seja,

$\hat{H}^{-1}[k]$, e como entrada ele recebe os símbolos OFDM no domínio da frequência. Então ele realiza a multiplicação de elemento a elemento descrita e entrega o vetor resultante na saída. Na Fig.4.4 temos a imagem do bloco e um exemplo de seu posicionamento.

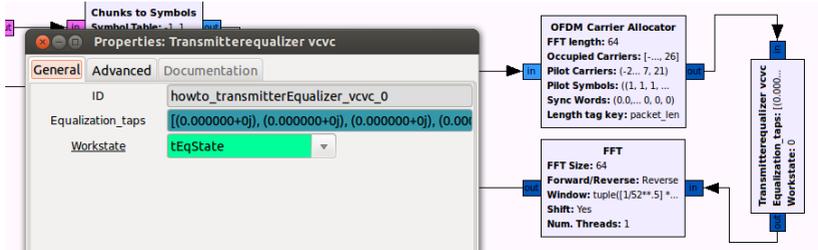


Figura 4.4: Bloco de compensação de interferência linear do canal

4.6 Resultados

Ao fim do trabalho, obtivemos uma implementação em GR e USRP, adaptada do software *gr-ieee802-11*, a qual implementa um sistema de comunicação baseado no padrão 802.11 e montada com a topologia ilustrada pela Fig. 4.5.

Através de ajustes nos parâmetros mencionados na seção 4.2, adaptamos a aplicação ao ambiente de teste, podendo então gerar algumas leituras. Estas leituras foram realizadas através do envio de múltiplos frames.

Na Fig. 4.6, está representado o módulo da estimação de um *tap* da resposta em frequência do canal em função das realizações (envio de um frame). Note que no instante do frame em destaque (48), o canal foi modificado. Esta descontinuidade foi resultado de uma deslocação proposital do aparelho transmissor. Também é perceptível a interferência que o ruído induz nesta medição. Os primeiros frames são erros provenientes da inicialização do software e assim não estão presentes na imagem.

Para plotar a Fig. 4.7, primeiro foi realizado a transmissão de uma sequência de frames. Em cada um, foi estimado e armazenado o módulo dos *taps* do canal. Por último foi calculado a média de cada *tap* entre os frames. Note que os *taps* zerados são das portadoras não populadas

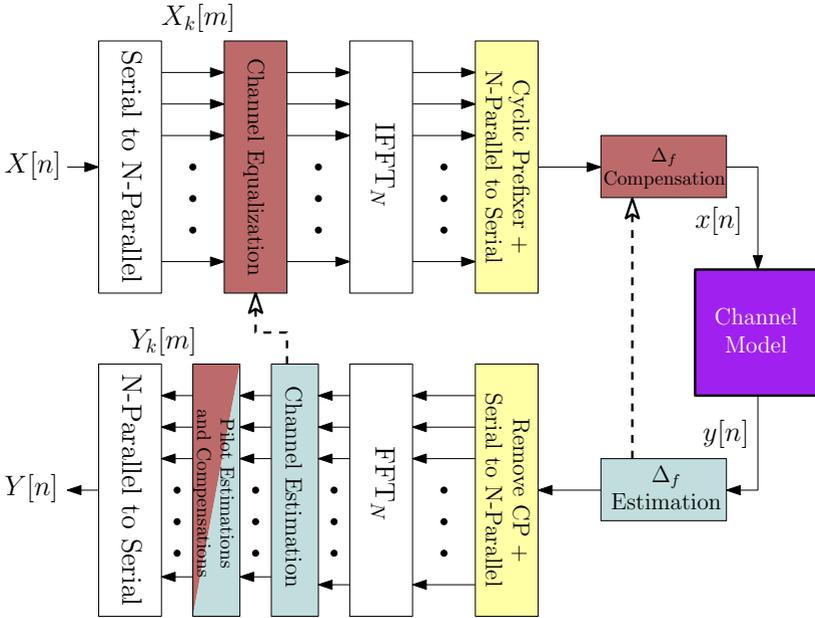


Figura 4.5: Sistema de comunicação realizado

e assim são valores não estimados.

Para plotar a Fig. 4.8, foi realizado o mesmo procedimento anterior, mas agora com o equalizador corrigindo os símbolos através de uma estimativa prévia.

Apesar da equalização se comportar quase como o esperado, os dois últimos *taps* não se comportaram como os outros. Em testes separados foi constatado que eles são relacionados entre si, ou seja, a compensação que é realizada em um interfere na compensação do outro. Isto é um indicio de não linearidade. No entanto não pôde ser confirmado durante a pesquisa.

Na Fig. 4.9 está representada a estimativa da diferença de frequência de portadora em kHz entre o transmissor e o receptor ao longo das várias realizações.

Diferentemente da implementação anterior, a implementação da compensação do desvio de frequência não gerou resultados. Isto é devido ao fato que o USRP, na recepção, realiza uma centralização automática de banda a qual está programada em seu FPGA. Não foi

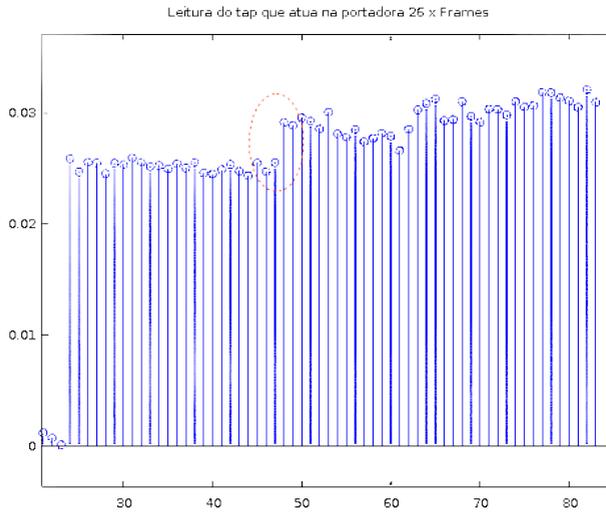


Figura 4.6: Módulo da estimação do $\text{tap } k = 26$ do canal em cada realização (transmissão de um frame)

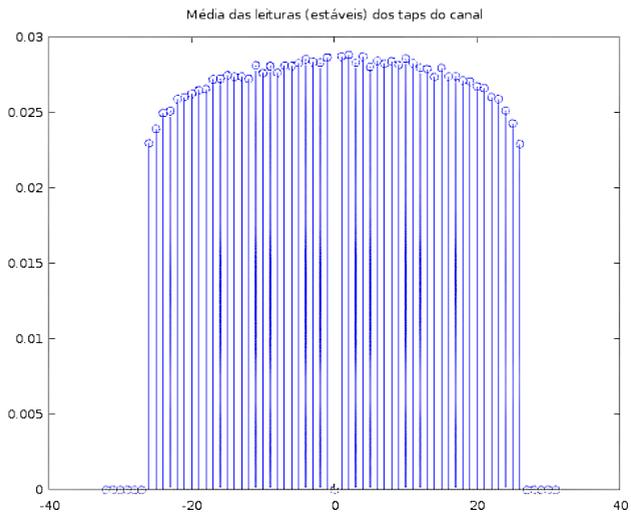


Figura 4.7: Módulo da média dos taps do canal antes da equalização

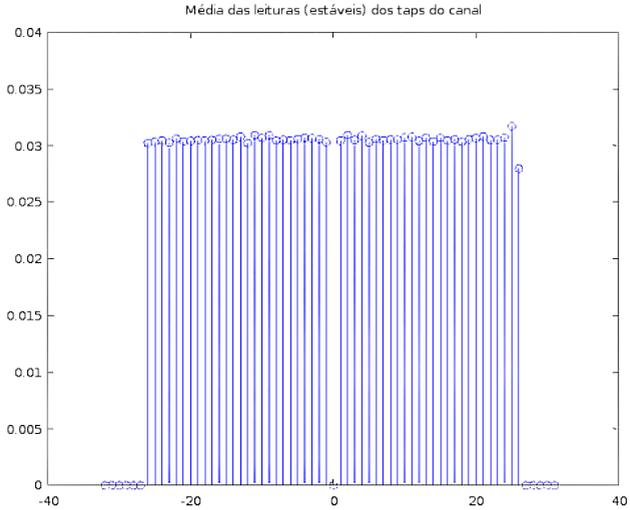


Figura 4.8: Módulo da média dos *taps* do canal após a equalização

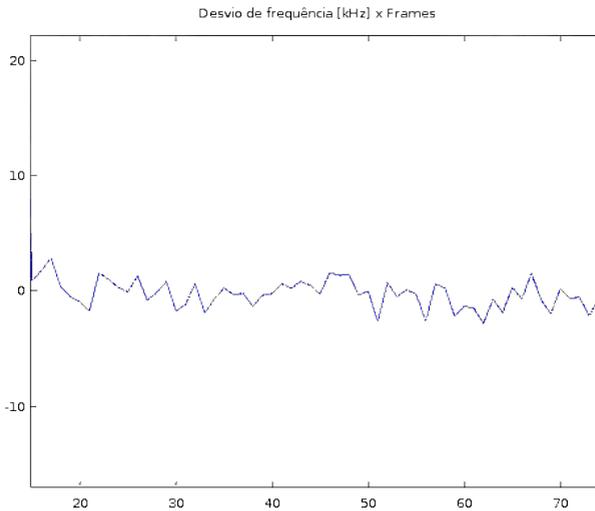


Figura 4.9: Estimação do Δ_f [kHz] entre transmissor e receptor em cada frame

Valor Médio: -446 Hz

possível desabilitar esta função e assim também não foi possível comprovar a compensação realizada.

Entretanto, para testar o funcionamento do programa, transmitimos o sinal com frequência central deslocada o suficiente para sair da faixa de correção do centralizador. Isto faz com que o próprio frame 802.11 não seja detectado pelo receptor. Ao ativar a implementação, o software de recepção volta a funcionar, mostrando que de certa forma ele compensa o desvio. Todavia não temos a garantia de precisão.

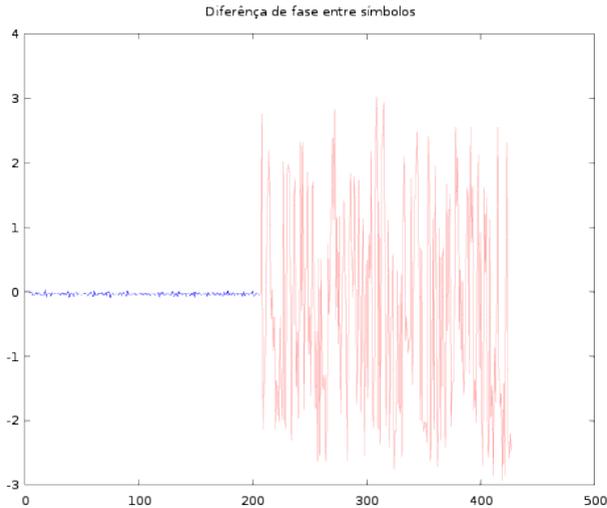


Figura 4.10: Diferença média, no domínio da frequência, de fase entre os símbolos OFDM

Na Fig. 4.10 está plotada a leitura da diferença de fase entre as pilotos de símbolos adjacentes versus o índice do símbolo. Esta estimação foi feita conforme explica a seção 3.2.1. Nota-se que esta diferença em média é próxima a zero e desta forma ela só é utilizada para correção do ruído de fase que ocorre entre os símbolos. Um detalhe é que a compensação deste parâmetro é feita logo após a estimação, pois este parâmetro varia muito rápido para ser compensado em um momento futuro. As amostras em vermelho são estimações falsas provenientes do fato do programa não identificar o fim do frame.

CAPÍTULO 5

Conclusão

Em virtude do que foi mencionado, conseguimos elencar os desafios físicos mais comuns de implementação de sistemas de comunicação sem fio. Foram revisados os conceitos básicos da modulação OFDM convencional junto a técnica de CP. Descrevemos um modelo do canal o qual leva em conta a plataforma utilizada e estes efeitos físicos mais comuns na implementação. Também expomos como este modelo afeta na transmissão de sinais OFDM.

Apresentamos as padronizações da PHY do protocolo 802.11 e sua variante a a qual utiliza a modulação OFDM. Com base no padrão, derivamos um conjunto de métodos para estimação dos parâmetros de interesse do canal.

Foi exposto a plataforma de implementação utilizada e algumas de suas peculiaridades. Citamos o ponto de partida do trabalho e descrevemos e implementamos técnicas de compensação de dois efeitos do canal. Por fim mostramos que é exequível realizar estas compensações antes mesmo da transmissão do sinal, e com resultados satisfatórios.

5.1 Trabalhos futuros

Apesar de citarmos vários desafios físicos, nem todos puderam ser implementados da forma desejada. Em função da implementação, ainda existem alguns pontos a serem aperfeiçoados. Tendo em vista a implementação da técnica PNC, existem outros desafios em aberto. As sugestões para os futuros trabalhos são:

- realizar a implantação da comunicação entre programas do GR e em uma segunda etapa, deverão ser implementadas as camadas de maior nível (MAC, IP) adaptadas de forma a se encarregarem desta intercomunicação de parâmetros;
- criar soluções e as implementar para os tópicos citados na seção 4.4;
- inserir novas funcionalidades e tornar mais eficientes os programas criados e utilizados;
- estudo e implementação das técnicas de detecção de sinais digitais somados;
- realizar modificações na camada PHY do protocolo de forma a possibilitar a implementação do PNC.

APÊNDICE A

Base Teórica do OFDM

Para compreendermos as propriedades dum método, é necessário entender a teoria por trás dele. Desta forma, para compreender algumas decisões que foram adotadas para o sistema OFDM, é importante ter conhecimento dos dois seguintes conteúdos.

A.1 Discrete Fourier Transform

Para obter a representação de Fourier exata de um sinal $x[n]$ finito e discreto, utilizamos a transformação da *Discrete Time Fourier Transform* (DTFT). No entanto, se quisermos uma representação discreta, podemos aplicar a transformada DFT de N pontos e conseguiremos a mesma representação espectral só que amostrada com N pontos igualmente espaçados. Isto é importante pois este espectro amostrado pode ser gravado e manipulado por máquinas digitais. Note que a DFT necessita saber a quantidade de pontos que ela irá processar. Assim utilizaremos a seguinte notação DFT_N . A função e o porquê deste número de pontos será esclarecido posteriormente.

Um detalhe importante desta transformada é que ela assume que os N pontos transformados fazem parte de um sinal periódico com

período N . Este sinal será chamado de sequência periódica associada e sua notação é $\tilde{x}[n]$. A figura A.1 elucida os comentários anteriores.

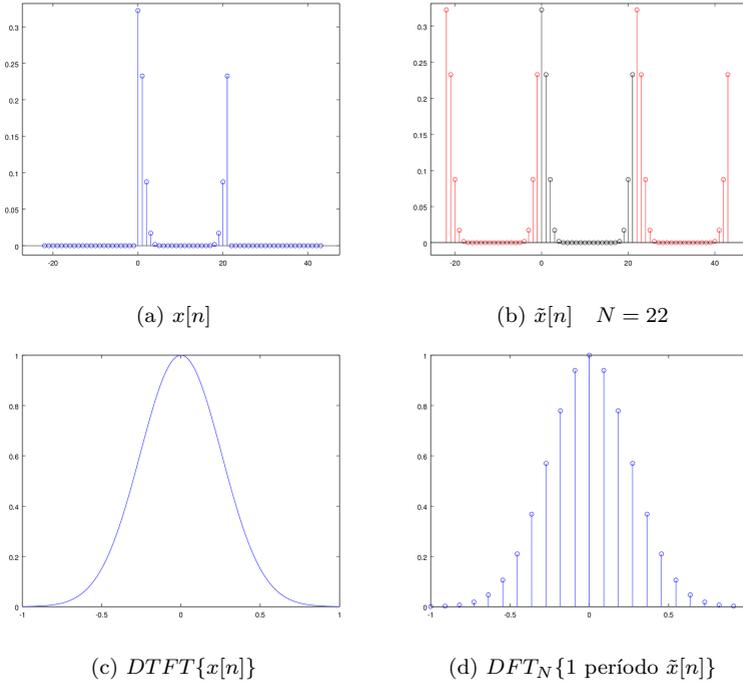


Figura A.1: Espectro DTFT e DFT

A definição matemática da DFT_N sobre um sinal discreto e limitado $x[n]$ esta dada em (A.1) [10].

$$X[k] \triangleq \sum_{n=0}^{N-1} x[n] e^{-j \frac{2\pi}{N} kn} \quad (\text{A.1})$$

Apesar da definição, vale a pena notar que $k \in \mathbb{Z}$, entretanto este sinal se repete em $k = N$ e por isto o domínio de k é restringido em $0 \leq k \leq N-1$. Mas note também que o domínio de k poderia ser restringido em um domínio deslocado, como o que está definido na equação (2.2), pois continuaria abrangendo todas as amostras.

A DFT também é invertível, e sua expressão, chamada de IDFT $_N$,

é dada por (A.2) [10].

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j \frac{2\pi}{N} kn} \quad (\text{A.2})$$

Nesta transformada já, k é restringido mas $n \in \mathbb{Z}$. De forma semelhante ao caso anterior, em $n = N$ ocorre periodicidade e por isso o domínio de n também é restringido em $0 \leq n \leq N-1$. Note a que as mesmas considerações de deslocamento podem ser feitas a este domínio. Desta modo, tanto o sinal no domínio do tempo como no domínio da frequência são restringidos por serem considerados periódicos.

Este par de transformadas possuem uma implementação através de um algoritmo computacionalmente eficiente. São eles a FFT para a transformada DFT e IFFT para a IDFT. Preferencialmente, o número de pontos destes algoritmos deverá ser $N = 2^p$, $p \in \mathbb{Z}$ para um ganho de velocidade computacional maior ainda.

A.1.1 Convolução Circular e DFT

Apesar dos enormes benefícios que a DFT provê, existe um problema em específico que deve ser levado em consideração. Este é a convolução circular.

Normalmente, quando queremos aplicar uma filtragem digital em um sinal finito $x[n]$, $0 \leq n \leq L-1$, implementamos a convolução linear deste com uma sequência $h[n]$, $0 \leq n \leq M-1$, cujo espectro já é conhecido. Os valores de $h[n]$ e $x[n]$ que estão fora do domínio, são assumidos como 0. Esta operação retorna um vetor $y[n]$, $0 \leq n \leq L+M-2$ que corresponde ao sinal filtrado. Esta operação é chamada de convolução linear e matematicamente está representada em (A.3).

$$y[n] = h[n] * x[n] = \sum_{g=-\infty}^{\infty} h[g] x[n-g]; \quad (\text{A.3})$$

Se realizarmos a $DFT_N\{x[n]\} = X[k]$, em seguida a $DFT_N\{h[n]\} = H[k]$, depois realizarmos a $IDFT_N\{X[k] \cdot H[k]\}$, nós não obteremos a mesma sequência $y[n]$ que é obtida pela convolução linear anterior, ou seja, $X[k] \cdot H[k] \neq Y[k]$. Isto se deve as suposições que fundamentam a DFT.

Como a DFT espera receber uma sequência periódica associada, ou seja, uma sequência truncada que corresponde a um período de um sinal infinito, ela assume que as amostras que não estão presentes antes do início e após o fim, são as amostras do sinal periódico. Assim, ela realiza uma convolução linear entre sinais que são diferentes dos originais. Esta operação, que é diferente da anterior, é chamada de convolução circular e está representada em (A.4) e deduzida em [10].

$$y[n] = IDFT_N\{H[k] \cdot X[k]\} = \sum_{m=0}^{N-1} h[m]x[\langle n - m \rangle_N]; \quad 0 \leq n \leq N - 1 \quad (\text{A.4})$$

Onde $\langle a \rangle_N$ representa modulo N sobre o valor a e que nesta aplicação realiza o chamado deslocamento circular.

Note que a convolução circular funciona sobre e retorna um número N finito de elementos. Para que não ocorram percas nesta operação, é necessário que todos os pontos dos dois vetores h e x sejam considerados. Desta forma, $N \geq \max(M, L)$. A operação anterior de convolução circular de N pontos será denotada por:

$$y[n] = h[n] \textcircled{N} x[n] \quad (\text{A.5})$$

É possível entretanto realizar a convolução linear através da convolução circular. Para isso é preciso inserir 0's (zeros) em uma das extremidades do sinal. A quantidade de zeros que deve ser inserida num sinal é dependente do tamanho do sinal que irá convoluir com ele e neste sinal que irá convoluir, deve ser realizada a mesma técnica.

Possuímos então 2 sinais $x[n]$, $0 \leq n \leq L - 1$ e $h[n]$, $0 \leq n \leq M - 1$ os quais desejamos realizar a convolução linear através do par DFT e IDFT. Note que $x[n]$ possui tamanho L e $h[n]$ possui tamanho M . Primeiramente preenchemos de zeros o fim ou o início de cada sinal, operação esta chamada de *zero padding*. A quantidade de zeros que $x[n]$ recebe deve ser pelo menos $M - 1$ (ex Fig. A.2a) e a quantidade que $h[n]$ recebe deve ser pelo menos $L - 1$ (ex Fig. A.2b).

Depois aplicamos a transformação DFT_N com $N \geq M + L - 1$ em ambos os sinais. Agora realizamos a multiplicação entre eles e por fim realizar a $IDFT_N$. Assim obteremos um sinal $y[n]$, $0 \leq n \leq L + M - 1$ que corresponde a convolução linear de $x[n]$ com $h[n]$. A Fig. A.3

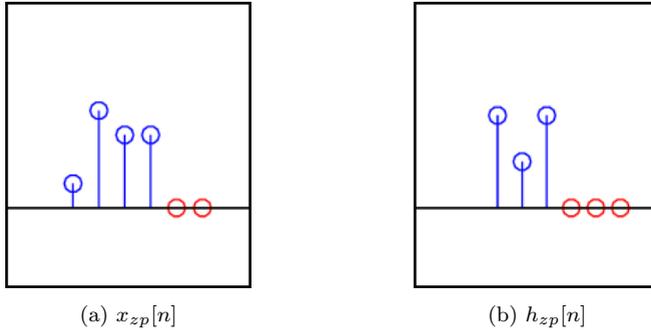


Figura A.2: Exemplo de zero padding

mostra um diagrama de bloco destas operações.

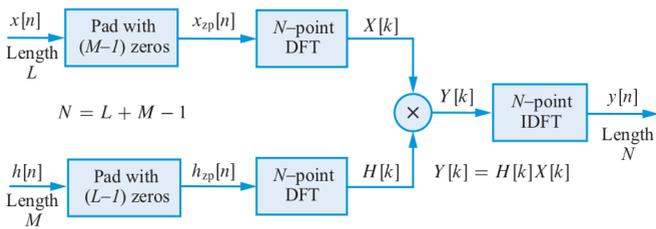


Figura A.3: Computação da convolução linear utilizando DFT

Fonte: [10], p. 392

A.2 Demonstração detalhada: Ortogonalidade em Frequência

Dados os 2 seguintes sinais em banda passante, com modulação digital, linear e gerados com pulso de formatação retangular com duração T_s (ou seja $0 \leq t \leq T_s$):

$$s_1(t) = a \cdot \cos(2\pi f_1 t) + b \cdot \sin(2\pi f_1 t)$$

$$s_2(t) = c \cdot \cos(2\pi f_2 t) + d \cdot \sin(2\pi f_2 t)$$

Para ocorrer ortogonalidade entre os dois sinais s_1 e s_2 , a seguinte

relação deve ser obedecida:

$$\int_0^{T_s} s_1(t)s_2(t) dt = 0 \quad (\text{A.6})$$

Para simplificar os primeiros equacionamentos,

- $\alpha = 2\pi f_1 t$
- $\beta = 2\pi f_2 t$

então,

$$s_1 \cdot s_2 =$$

$$ac \cdot \cos(\alpha) \cos(\beta) + ad \cdot \cos(\alpha) \sin(\beta) + bc \cdot \sin(\alpha) \cos(\beta) + bd \cdot \sin(\alpha) \sin(\beta)$$

Expandindo as multiplicações em identidades de produto temos

$$\begin{aligned} s_1 \cdot s_2 = & \frac{ac}{2} \left(\cos(\alpha - \beta) + \cos(\alpha + \beta) \right) + \\ & \frac{ad}{2} \left(\sin(\alpha + \beta) - \sin(\alpha - \beta) \right) + \\ & \frac{bc}{2} \left(\sin(\alpha + \beta) + \sin(\alpha - \beta) \right) + \\ & \frac{bd}{2} \left(\cos(\alpha - \beta) - \cos(\alpha + \beta) \right) \end{aligned}$$

$$\begin{aligned} s_1 \cdot s_2 = & \frac{ac - bd}{2} \left(\cos(\alpha + \beta) \right) + \frac{ac + bd}{2} \left(\cos(\alpha - \beta) \right) + \\ & \frac{ad + bc}{2} \left(\sin(\alpha + \beta) \right) + \frac{bc - ad}{2} \left(\sin(\alpha - \beta) \right) \end{aligned}$$

Uma segunda simplificação,

$$\frac{ac - bd}{2} = k_1; \quad \frac{ac + bd}{2} = k_2; \quad \frac{ad + bc}{2} = k_3; \quad \frac{bc - ad}{2} = k_4;$$

Levando em conta que,

$$\alpha + \beta = 2\pi(f_1 + f_2)t; \quad \alpha - \beta = 2\pi(f_1 - f_2)t;$$

temos que

$$\int_0^{T_s} s_1(t)s_2(t) dt = \left(\begin{aligned} &\frac{k_1}{2\pi(f_1 + f_2)} \sin(2\pi(f_1 + f_2)t) + \\ &\frac{k_2}{2\pi(f_1 - f_2)} \sin(2\pi(f_1 - f_2)t) + \\ &\frac{-k_3}{2\pi(f_1 + f_2)} \cos(2\pi(f_1 + f_2)t) + \\ &\frac{-k_4}{2\pi(f_1 - f_2)} \cos(2\pi(f_1 - f_2)t) \end{aligned} \right) \Bigg|_0^{T_s}$$

Quando $t = 0$, $\sin(kt) = 0$ e $\cos(kt) = 1$, assim

$$\int_0^{T_s} s_1(t)s_2(t) dt = \begin{aligned} &\frac{k_1}{2\pi(f_1 + f_2)} \sin(2\pi(f_1 + f_2)T_s) + \\ &\frac{k_2}{2\pi(f_1 - f_2)} \sin(2\pi(f_1 - f_2)T_s) + \\ &\frac{-k_3}{2\pi(f_1 + f_2)} \cos(2\pi(f_1 + f_2)T_s) + \\ &\frac{-k_4}{2\pi(f_1 - f_2)} \cos(2\pi(f_1 - f_2)T_s) + \\ &\frac{k_3}{2\pi(f_1 + f_2)} + \frac{k_4}{2\pi(f_1 - f_2)} \end{aligned}$$

A fim de zerarmos esta expressão, devemos escolher argumentos tais que o valor dos cossenos seja 1, para que k_3 e k_4 se anulem com $-k_3$ e $-k_4$, e que os senos sejam nulos. Temos assim as seguintes relações

- $2\pi(f_1 + f_2)T_s \in \mathbb{Z} \cdot 2\pi$ para anular parcelas com k_3
- $2\pi(f_1 - f_2)T_s \in \mathbb{Z} \cdot 2\pi$ para anular parcelas com k_4
- $2\pi(f_1 + f_2)T_s \in \mathbb{Z} \cdot \pi$ para anular parcelas com k_1
- $2\pi(f_1 - f_2)T_s \in \mathbb{Z} \cdot \pi$ para anular parcelas com k_2

e logo,

- $f_1 + f_2 \in \frac{\mathbb{Z}}{T_s}$
- $f_1 + f_2 \in \frac{\mathbb{Z}}{2T_s}$
- $f_1 - f_2 \in \frac{\mathbb{Z}}{T_s}$
- $f_1 - f_2 \in \frac{\mathbb{Z}}{2T_s}$

Temos que assegurar então que todas estas condições sejam obedecidas. Para isto realizamos a intersecção dos conjuntos referentes a $(f_1 + f_2)$ e os referentes a $(f_1 - f_2)$. Por fim, para dois sinais com modulações digitais e lineares genéricas serem ortogonais entre si, suas frequências centrais devem obedecer as seguintes relações

- $f_1 + f_2 = \frac{m}{T_s} \quad m \in \mathbb{Z}$
- $f_1 - f_2 = \frac{n}{T_s} \quad n \in \mathbb{Z} \quad m \neq n$

Destas duas expressões, isolamos alguns parâmetros importantes para as implementações deste trabalho. São eles:

- Diferença mínima de frequência entre portadoras ortogonais:

$$\Delta_F = (f_1 - f_2)|_{n=1} = \frac{1}{T_s} \quad (\text{A.7})$$

- Frequência de centro:

$$f_c = \frac{f_1 + f_2}{2} = \frac{m}{2T_s} \quad m \in \mathbb{Z} \quad (\text{A.8})$$

- Frequência individual de cada portadora com diferença mínima entre elas:

$$f_1 = \frac{m+1}{2T_s} \quad m \in \mathbb{Z}; \quad f_2 = \frac{m-1}{2T_s} \quad m \in \mathbb{Z}; \quad (\text{A.9})$$

Referências bibliográficas

- [1] “IEEE Standard for Information technology–Telecommunications and information exchange between systems local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,” *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007)*, pp. 1–2793, March 2012.
- [2] (2015, 08) What Is NI USRP Hardware? National Instruments. [Online]. Available: <http://www.ni.com/white-paper/12985/en/>
- [3] (2015, 08) WiME. [Online]. Available: <http://www.ccs-labs.org/projects/wime/>
- [4] B. Bloessl, M. Segata, C. Sommer, and F. Dressler, “An IEEE 802.11a/g/p OFDM Receiver for GNU Radio,” in *ACM SIGCOMM 2013, 2nd ACM SIGCOMM Workshop of Software Radio Implementation Forum (SRIF 2013)*. Hong Kong, China: ACM, August 2013, pp. 9–16.
- [5] Y. S. Cho, J. Kim, W. Y. Yang, and C. G. Kang, *MIMO-OFDM Wireless Communications with MATLAB*. John Wiley & Sons (Asia) Pte Ltd, 2010.

- [6] R. Farias and D. Silva, “Proposta de Extensão do Protocolo 802.11 MAC para Codificação de Rede na Camada Física,” Sep 2015, pp. 1–5.
- [7] GNU Radio Website. (2015, 08). [Online]. Available: <http://www.gnuradio.org>
- [8] A. Goldsmith, *Wireless communications*. Cambridge University Press, 2005.
- [9] S. Haykin and M. Moher, *Introduction to Analog and Digital Communications*, 2nd ed. John Wiley & Sons, Inc, 2007.
- [10] D. G. Manolakis and V. K. Ingle, *Applied Digital Signal Processing*. Cambridge University Press, 2011.
- [11] E. Perahia and R. Stacey, *Next Generation Wireless LANs*. Cambridge University Press, 2008.
- [12] A. E. Stanciu, L.-M. Nemțoi, and I. M. Moise, “Considerations regarding the Spectral Efficiency of Orthogonal Frequency Division Multiplexing,” *Development and Application Systems International Conference*, pp. 1–7, May 2012.
- [13] S. Zhang, S. C. Liew, and P. P. Lam, “Hot topic: Physical-layer network coding,” in *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom ’06. New York, NY, USA: ACM, 2006, pp. 358–365. [Online]. Available: <http://doi.acm.org/10.1145/1161089.1161129>