



DISSERTAÇÃO

Mestrado em Engenharia Eletrotécnica

Sincronização em OFDM

CRISTÓVÃO RODRIGUES

Leiria, Novembro de 2013



DISSERTAÇÃO

Mestrado em Engenharia Eletrotécnica

Sincronização em OFDM

CRISTÓVÃO RODRIGUES

Dissertação de Mestrado realizada sob a orientação do professor doutor Carlos Miguel Nogueira Gaspar Ribeiro, Professor da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria.

Leiria, Novembro de 2013

Agradecimentos

Gostaria de agradecer ao meu orientador professor doutor Carlos Miguel Nogueira Gaspar Ribeiro por todo o apoio e dedicação prestados.

Esta tese é dedicada aos meus pais, pois foi com o seu apoio e dedicação que eu cheguei aqui. Sem eles eu nunca teria conseguido.

Não esquecendo também de agradecer a todos os meus colegas e amigos, que de alguma forma contribuíram para que fosse possível atingir esta etapa da minha formação acadêmica.

Abstract

This dissertation presents a work on synchronization for OFDM systems. The intention of this work is the integration of the synchronization architecture in the research project “CROWN - Co-operative Radio over Fibre for Wireless Networks” currently underway at the Institute of Telecommunications.

This architecture was implemented on a platform based on programmable FPGA devices, using the development tools Matlab, System Generator and ISE. The implemented system has the particularity of having an asynchronous operation principle and uses the algorithms of Van de Beek [1] and Carlos Ribeiro [2] to perform to the estimation and subsequent synchronization. Both algorithms were used for estimation of the carrier frequency offset (CFO), while the algorithm of Van de Beek was also used to estimate the beginning of the frame.

Analyses of system performance were conducted for different conditions, being the aim to observe the performance of the implemented estimators. The performance was then analyzed according to the resulting BER and the precision on the estimation of the beginning of the frame and the value of the CFO. In addition to the individual analysis, it is also made a comparison between the accuracy of both estimators.

Keywords: OFDM synchronization, System Generator, FPGA, CFO

Resumo

Esta dissertação apresenta o trabalho sobre sincronização de recepção para sistemas OFDM. Tendo como objetivo a integração da arquitetura desenvolvida no projeto de investigação “CROWN - Co-operative Radio over Fibre for Wireless Networks” atualmente em curso no Instituto de Telecomunicações.

Esta arquitetura de recepção foi implementada numa plataforma de desenvolvimento baseada em dispositivos programáveis FPGA, recorrendo às ferramentas de desenvolvimento MatLab, System Generator e ISE. O sistema implementado tem a particularidade de ter um princípio de funcionamento assíncrono e recorre aos algoritmos de Van de Beek [1] e Carlos Ribeiro [2] para proceder à estimação e conseqüente sincronização. Ambos os algoritmos foram utilizados para estimação do CFO, tendo o algoritmo de Van de Beek sido também utilizado para estimação do início de trama.

Foram realizadas análises do desempenho do sistema para diferentes condições, sendo o objectivo de analisar o desempenho dos estimadores implementados. A performance foi então analisada de acordo com BER resultante e do erro de estimação do início de trama e do valor do CFO. Para além da análise individual dos resultados, é também feita uma comparação da precisão de ambos os estimadores.

Palavras-chave: Sincronização em OFDM, System Generator, FPGA, CFO

Índice

Agradecimentos	i
Abstract	iii
Resumo	v
Índice	ix
Lista de Figuras	xi
Lista de Tabelas	xv
Lista de Acrónimos	xvii
1 Introdução	1
1.1 Contexto e motivação	1
1.2 Objetivos	2
1.3 Organização do documento	3
2 Introdução de conceitos relevantes para o trabalho	5
2.1 Canal rádio multi-percurso	5
2.2 Esquemas de Multiplexação	8
2.2.1 Multiplexação FDD e TDD	8
2.2.2 Acesso múltiplo TDMA, FDMA e CDMA	8
2.2.3 OFDM	9
2.2.3.1 Múltiplas sub-portadoras e ortogonalidade	9
2.2.3.2 Prefixo Cíclico	11
2.2.3.3 Sincronização, Equalização do canal e Modulação das sub-portadoras	12

2.2.4	OFDMA	13
2.2.5	SC-FDMA	14
2.3	Long Term Evolution (LTE)	16
2.3.1	LTE: Estrutura das <i>frames</i> e <i>sub-frames</i>	16
2.3.1.1	Estrutura dos recursos	17
2.4	FPGAs	19
2.4.1	Xilinx System Generator	19
2.4.2	Algoritmo CORDIC	20
2.4.2.1	CORDIC: Modo rotação	20
2.4.2.2	CORDIC: Modo translação	20
2.4.3	Direct Digital Synthesizer (DDS)	22
2.4.4	Revisão de técnicas de simplificação da DFT	22
2.4.4.1	FFT <i>radix-2</i> e <i>radix-4</i>	23
3	Sincronização em OFDM	27
3.1	Limitações do OFDM	27
3.2	Compensação do CFO	29
3.3	Algoritmos de Sincronização	31
3.3.1	Algoritmos baseados em símbolos <i>training</i> ou pilotos	31
3.3.1.1	Revisão do algoritmo proposto por C. Ribeiro em [2]	32
3.3.1.2	Sequências <i>training</i> : <i>Zadoff-Chu</i>	36
3.3.2	Algoritmos <i>Non-Data Aided</i>	37
3.3.3	Algoritmos de estimação blind	39
4	Sistema OFDM Assíncrono	41
4.1	Processamento assíncrono	41
4.1.1	Transmissor	42
4.1.2	Recetor	43
4.1.3	Constituição do Sistema	44
4.2	Implementação do algoritmo ML de van de Beek na plataforma <i>System Generator</i>	46
4.2.1	Energia e Correlação	46
4.2.2	Decisor da detecção da sequência <i>Zadoff-Chu</i>	49
4.2.2.1	Estimação temporal recorrendo a memórias FIFO	52

4.2.3	Sincronização temporal e estimação do CFO	53
4.3	Implementação do algoritmo de C. Ribeiro na plataforma <i>System Generator</i>	56
4.3.1	Implementação do processamento do estimador	58
4.3.2	Definição da gama da janela de estimação de CFO	59
4.4	Bloco <i>CFO Correction</i>	62
5	Resultados	65
5.1	Estimador ML de van de Beek	65
5.1.1	Desempenho em Ruído AWGN	65
5.1.2	Desempenho em Ruído AWGN e canal rádio <i>outdoor</i>	67
5.2	Estimador com base nos pilotos no TD de C. Ribeiro	70
5.2.1	Desempenho em Ruído AWGN	70
5.2.2	Desempenho em Ruído AWGN e canal rádio <i>outdoor</i>	71
5.3	Comparação do desempenho dos estimadores	72
6	Conclusão e trabalho futuro	75
6.1	Conclusão	75
6.1.1	Conclusões sobre o desempenho dos estimadores	75
6.1.2	Conclusões sobre o assincronismo do sistema	76
6.2	Trabalho futuro	77
	Bibliografia	79

Lista de Figuras

2.1	Diagrama de blocos que ilustra os efeitos da propagação em canais sem fios	6
2.2	Esquemas de multiplexação FDD e TDD.	8
2.3	Esquemas de multiplexação TDMA, FDMA e CDMA.	9
2.4	Espectro de 5 sub-portadoras de um sinal OFDM.	10
2.5	Estrutura de um sistema OFDM.	10
2.6	Estrutura do Prefixo Cíclico	11
2.7	Aumento da robustez contra ISI devido ao uso do CP.	12
2.8	Alocação de sub-portadoras a diferentes utilizadores, no caso de sistemas OFDM e OFDMA.	13
2.9	Métodos possíveis de distribuição das sub-portadoras pelos diferentes utilizadores, em sistemas OFDMA.	14
2.10	Estruturas dos sistemas OFDMA e SC_FDMA.	15
2.11	Aspecto dos símbolos OFDMA e SC-FDMA no domínio do tempo.	15
2.12	Aspecto dos símbolos SC-FDMA, no domínio do tempo e frequência, no caso da multiplexação ser implementada de forma localizada ou distribuída.	15
2.13	Estrutura da <i>frame</i> FD-LTE.	16
2.14	Estrutura da <i>frame</i> TD-LTE.	17
2.15	Rotação do vector de entrada por um ângulo α	21
2.16	Translação de um vector (coordenadas cartesianas para polares).	21
2.17	<i>Butterflies</i> radix-2 e radix-4 [3].	24
3.1	Consequências da presença de CFO.	28
3.2	Constituição dos símbolos de referência propostos por Schimdl & Cox para estimação do início de trama e CFO.	32
3.3	Estrutura do estimador ML de van de Beek [1].	38
4.1	Sistemas síncronos: o fluxo de dados funciona com ritmo constante.	41

4.2	Sistemas assíncronos: o único bloco que funciona com um ritmo constante é o ADC/DAC.	42
4.3	Sistemas assíncronos: Constituição geral dos módulos do transmissor.	43
4.4	Sistemas assíncronos: Constituição geral dos módulos do recetor.	44
4.5	Implementação do sistema OFDM desenvolvido na plataforma <i>Xilinx System Generator</i>	45
4.6	Consequências da remoção do CP antes da compensação do CFO.	46
4.7	Implementação do algoritmo de estimação ML de van de Beek na plataforma <i>Xilinx System Generator</i>	47
4.8	Implementação do cálculo da Energia na plataforma <i>Xilinx System Generator</i>	47
4.9	Resultado do cálculo da energia para duas tramas.	48
4.10	Implementação do cálculo da correlação e estimação do desvio de fase na plataforma <i>Xilinx System Generator</i>	48
4.11	Resultados do algoritmo de estimação para duas tramas.	49
4.12	Implementação do bloco <i>Frame_Detection</i>	50
4.13	Resultado da estimador ML de van de Beek (b) em relação ao conteúdo contido nos símbolos OFDM. Encontrando-se apenas representada a componente imaginária dos símbolos OFDM (a).	50
4.14	Diagrama de estados da máquina de estados <i>Peak_Detector</i>	51
4.15	Janela de procura do pico de correlação resultante da sequência <i>Zadoff-Chu</i>	52
4.16	Diagrama de estados da máquina de estados <i>WE_Out_Ctrl</i>	54
4.17	Comportamento das saídas do bloco <i>Frame_Detection</i>	55
4.18	Implementação do estimador de CFO de C. Ribeiro, na plataforma <i>Xilinx System Generator</i>	57
4.19	Implementação do bloco <i>Estimator</i> , na plataforma <i>Xilinx System Generator</i>	58
4.20	Implementação dos blocos <i>Estimation</i> , na plataforma <i>Xilinx System Generator</i>	59
4.21	Implementação do bloco <i>Define_candidate_CFO</i> , na plataforma <i>Xilinx System Generator</i>	61
4.22	Implementação do bloco <i>CFO_Correction</i> , na plataforma <i>Xilinx System Generator</i>	62
4.23	Compensação do CFO: Correspondência entre a fase e as amostras a ser compensadas por esta.	63
5.1	Módulo do erro de estimação de CFO (dB) ao longo do tempo.	68

5.2	Desvios médios das estimações de CFO, para 2 kHz, para diferentes níveis SNR.	74
-----	---	----

Lista de Tabelas

2.1	Alocação espectral do esquema TD-LTE.	17
2.2	Parametrização do <i>uplink/downlink</i> do LTE [4].	18
2.3	Larguras de banda do LTE [4].	18
5.1	Estimador ML de van de Beek: Desvio médio da estimação de CFO, em Hertz, para diferentes níveis SNR.	66
5.2	Estimador ML de van de Beek: Desvio médio da estimação de início de trama efetuada, em amostras, para diferentes níveis SNR.	66
5.3	Estimador ML de van de Beek: BER resultante para os vários CFOs e níveis SNR.	67
5.4	Estimador ML de van de Beek: Desvio médio da estimação de CFO, em Hertz, para diferentes níveis SNR e um canal rádio <i>outdoor</i> (250ns rms delay spread).	67
5.5	Estimador ML de van de Beek: Desvio médio da estimação de início de trama, em amostras, para diferentes níveis SNR e um canal rádio <i>outdoor</i> (250ns rms delay spread).	68
5.6	Estimador ML de van de Beek: Desvio médio da estimação de CFO, em Hertz, ignorando diferenças superiores a 1500 Hz.	69
5.7	Estimador ML de van de Beek: Percentagem das estimações de CFO com uma diferença superior a 1500 Hz.	69
5.8	Estimador de C. Ribeiro (Sincronização temporal ideal): Desvio médio da estimação de CFO, em Hertz, para diferentes níveis SNR.	70
5.9	Estimador de C. Ribeiro (Sincronização temporal de van de Beek): Desvio médio da estimação de CFO, em Hertz, para diferentes níveis SNR.	71
5.10	Estimador de C. Ribeiro (Sincronização temporal ideal): Desvio médio da estimação de CFO, em Hertz, para diferentes níveis SNR e um canal rádio <i>outdoor</i> (250ns rms delay spread).	71

5.11	Estimador de C. Ribeiro (Sincronização temporal de van de Beek): Desvio médio da estimação de CFO, em Hertz, para diferentes níveis SNR e um canal rádio <i>outdoor</i> (250ns rms delay spread).	72
5.12	MSE, em dB, entre o desvio médio de CFO do estimador de C. Ribeiro (Sincronização temporal ideal) e do estimador de van de Beek, para diferentes níveis SNR.	73
5.13	MSE, em dB, entre o desvio médio de CFO do estimador de C. Ribeiro (Sincronização temporal ML de van de Beek) e do estimador de van de Beek, para diferentes níveis SNR.	73
5.14	MSE, em dB, entre o desvio médio de CFO do estimador de C. Ribeiro (Sincronização temporal ideal) e do estimador de van de Beek, para diferentes níveis SNR e um canal rádio <i>outdoor</i> (250ns rms delay spread).	74

Lista de Acrónimos

ADC Analog to Digital Converter

ADSL Asymmetric Digital Subscriber Line

AGC Automatic Gain Control

AWGN Additive White Gaussian Noise

BER Bit Error Rate

BPSK Binary Phase-Shift Keying

CAS Carrier Assignment Scheme

CAZAC Constant Amplitude Zero Autocorrelation Waveform

CDMA Code Division Multiple Access

CFO Carrier Frequency Offset

CIR Channel Impulse Response

CORDIC COordinate Rotation DIgital Computer

CP Cyclic Prefix

DAC Digital to Analog Converter

DDS Direct Digital Synthesizers

DFT Discrete Fourier Transform

DSP Digital Signal Processing

DVB Digital Video Broadcasting

DwPTS Downlink Pilot Time Slot

FD Frequency-Domain

FD-LTE Frequency-Division Long-Term Evolution

FDD Frequency Division Duplex

FDMA Frequency Division Multiple Access

FEC Forward Error Correction

FIFO First In First Out)

FIR Finite Impulse Response

FFT Fast Fourier Transform

FPGA Field-programmable gate array

FSK Frequency Shift Keying

GNU Radio Software Defined Radio Platform, (GNU Open Source)

GP Guard Period

HF High Frequency

ICI Inter-Carrier Interference

IDFT Inverse Discrete Fourier Transform

IFFT Inverse Fast Fourier Transform

IIR Infinite Impulse Response

ISE Integrated Software Environment

ISI Inter-Symbol Interference

LAN Local Area Network

LOS Line of Sighth

LTE Long Term Evolution

LTE-Advanced Advancements for Evolved Universal Terrestrial Radio Access

LUT Look-Up Table

MAC Medium Access Control

MAI Multiple Access Interference

MAN Metropolitan Area Network

MIMO Multiple-Input Multiple-Output

ML Maximum Likelihood

MMSE Minimum Mean Square Error

MSE Mean Square Error

MSK Minimum Shift Keying

OFDM Orthogonal Frequency Division Multiplexing

OFDMA Orthogonal Frequency Division Multiple Access

OSI Open Systems Interconnection

PAR Peak to Average Ratio

PDP Power Delay Profile

PLL Phase-locked loop

PLL Phase-locked loop

PHY Physical Layer

PN Pseudo-Noise

PSD Power Spectral Density

PSK Phase-Shift Keying

QAM Quadrature-Amplitude Modulation

QPSK Quadrature Phase-Shift Keying

RADAR Radio Detection And Ranging

RAM Random Access Memory

RB Resource Block

RF Radio-Frequency

RS reed Solomon

RTL Register Transfer Level

SC-FDMA Single Carrier Frequency Division Multiple Access

SDR Software Defined Radio

SISO Single-Input Single-Output

SNR Signal to Noise Ratio

TD Time-Domain

TD-LTE Time-Division Long-Term Evolution

TDD Time Division Duplex

TDMA Time Division Multiple Access

UMTS Universal Mobile Telecommunications System

UpPTS Uplink Pilot Time Slot

VHDL VHSIC Hardware Description Language

VHSIC Very High Speed Integrated Circuits

WARP Wireless Open-Access Research Platform

Wi-Fi Wireless Fidelity

WiMAX Worldwide Interoperability for Microwave Access

Capítulo 1

Introdução

Este capítulo apresenta uma introdução ao trabalho desenvolvido nesta dissertação. É feita uma descrição da motivação que levou à realização deste trabalho e são apresentados os objetivos e a organização da tese.

1.1 Contexto e motivação

Nas últimas décadas tem-se assistido a uma explosão no crescimento da indústria das comunicações sem fios. Os sistemas da primeira e segunda geração focaram-se apenas em comunicações de voz, enquanto as redes de terceira geração (3GPP e 3GPP2) começaram a focar-se também na transmissão de dados, recorrendo à técnica *Code Division Multiple Access* (CDMA). Tendo em conta estas considerações, as tecnologias 4G como o LTE (*Long Term Evolution*) e o WiMAX têm evoluído muito rapidamente, com o objectivo de proporcionar velocidades de transmissão até antes não imaginadas. É de realçar que ambas estas tecnologias recorrem ao uso do esquema de multiplexação *Orthogonal Frequency Division Multiplexing* (OFDM). A proliferação de sistemas de transmissão sem fios baseados nesta tecnologia, de que se destacam o Wi-Fi (802.11g), DVB, WiMAX e LTE, tem demonstrado a potencialidade desta técnica. Esta permite atingir velocidades de transmissão muito elevadas mesmo em ambientes interiores com interferência de outros sistemas. A utilização de um elevado número de portadoras com uma pequena separação na frequência e sem banda de guarda, torna estes sistemas muito sensíveis a imperfeições de sincronismo na receção. Assim, uma das características inerentes aos sistemas baseados nesta tecnologia é requerer uma elevada precisão na sincronização do recetor.

O objetivo dos sistemas das próximas gerações é fornecer altos débitos, baixa latência, alta-fidelidade e elevada robustez. Para conseguir atingir todos estes requisitos, é necessário fazer uma reestruturação significativa tanto da camada física (PHY) como da

arquitetura de rede. Assim, os sistemas da próxima geração não terão apenas de suportar uma capacidade enorme de processamento, mas também lhes será requerido uma grande flexibilidade, permitindo-os suportar vários modos de funcionamento. Com vista a proceder a melhorias nas arquiteturas de rede, tem-se começado a focar na escolha do hardware utilizado. Quanto a esta questão, as FPGAs têm mostrado ser a melhor escolha, devido à estrutura de paralelismo inerente e à sua capacidade de rápida reconfiguração, conseguindo assim corresponder aos requisitos de processamento e flexibilidade requeridos. Para além da característica de rápida reconfiguração, as FPGAs sofreram também uma grande revolução a nível da implementação de algoritmos de processamento de sinais. Esta evolução torna-as, assim, candidatas de peso, substituindo os sistemas de *hardware* tradicionais, isto tanto para comunicações com fios como sem fios [5].

A expectativa quanto à tendência do futuro dos sistemas de comunicação é que estes sejam baseados na tecnologia OFDM e recorram à utilização de FPGAs na camada física [5]. Portanto, esta dissertação apresenta o trabalho sobre sincronização de receção para sistemas OFDM, tendo como objetivo final a implementação em *hardware* FPGA. Pretende-se com o trabalho desenvolvido a integração da arquitetura de sincronização no projeto de investigação “CROWN - Co-operative Radio over Fibre for Wireless Networks” atualmente em curso no Instituto de Telecomunicações.

1.2 Objetivos

O objectivo do trabalho é a implementação de um sistema com um princípio de funcionamento assíncrono, cuja sincronização do recetor é feita recorrendo aos algoritmos de van de Beek [1] e C. Ribeiro [2]. Ambos os algoritmos foram utilizados para estimação do CFO, tendo o algoritmo de van de Beek sido também utilizado para estimação do início de trama. Esta arquitetura de receção foi implementada numa plataforma de desenvolvimento baseada em dispositivos programáveis FPGA, recorrendo às ferramentas de desenvolvimento MatLab, System Generator e ISE.

Para a concretização dos objetivos foram estabelecidas as seguintes fases:

1. Estado-da-arte sobre sistemas OFDM e algoritmos de sincronização.
2. Definição da arquitetura de sincronização com identificação dos algoritmos a implementar para cada tarefa.
3. Exploração das ferramentas de desenvolvimento MatLab, System Generator e ISE.
4. Ambientação à cadeia de transmissão OFDM.
5. Implementação dos algoritmos de sincronização.
6. Teste de desempenho da sincronização num ambiente móvel.

7. Redação da dissertação.

1.3 Organização do documento

A dissertação está organizada em seis capítulos. O presente capítulo introduz o trabalho de investigação desenvolvido, apresentando o seu contexto e motivação assim como os objetivos.

O capítulo 2, Introdução de conceitos relevantes para o trabalho, apresenta uma breve revisão dos sistemas de comunicação sem fios, expondo as principais técnicas de multiplexação existentes. De seguida é feita uma revisão do *standard* LTE, uma vez que o trabalho desenvolvido tem como aplicação final a integração num sistema baseado nesta tecnologia. Por fim, é feita uma introdução à plataforma *Xilinx System Generator* e são revistos alguns algoritmos presentes na biblioteca disponibilizada pela Xilinx.

O capítulo 3, Sincronização em OFDM, faz uma revisão das características inerentes aos sistemas OFDM, mais concretamente referentes à precisão requerida na sincronização do recetor. De seguida é descrito o processo de compensação do CFO, sendo também revistos alguns dos principais métodos de estimação de início de trama e de CFO. A revisão dos algoritmos ML de van de Beek [1] e do algoritmo com base na posição dos pilotos no domínio do tempo de C. Ribeiro [2] é mais detalhada que os restantes, devido a serem estes os algoritmos utilizados na arquitetura de sincronização.

O capítulo 4, Sistema OFDM Assíncrono, apresenta o sistema disponibilizado para o desenvolvimento desta tese, o qual tem a particularidade de ter um princípio de funcionamento assíncrono. É feita uma comparação entre os sistemas síncronos e assíncronos, realçando as vantagens e desvantagens de ambos. Por fim, é descrita toda a implementação dos algoritmos de sincronização implementados nesta tese na plataforma de desenvolvimento utilizada.

O capítulo 5, Resultados, apresenta uma análise dos resultados do desempenho dos estimadores implementados perante os diferentes testes realizados. Estes testes consistiram na introdução de diferentes CFOs no sinal recebido pelo recetor, diferentes níveis de ruído AWGN e ainda a utilização de um canal rádio *outdoor*. Por fim, é feita uma análise do desempenho de ambos os estimadores em termos de precisão da estimação efetuada.

O capítulo 6, Conclusão e trabalho futuro, são apresentadas as conclusões deste trabalho e introduzidas algumas sugestões para trabalho futuro.

Capítulo 2

Introdução de conceitos relevantes para o trabalho

Neste capítulo é feita uma breve revisão dos conceitos relevantes para este trabalho. Nas primeiras secções são revistos os canais rádio, presentes nos sistemas sem fios, e as principais técnicas de multiplexação existentes, desde as mais simples às de maior complexidade. Como o tema central desta tese é a sincronização em OFDM, ser-lhe-á dada uma maior atenção em comparação com os restantes esquemas de multiplexação. De seguida é feita uma revisão do *standard* LTE e da plataforma *Xilinx System Generator*, revendo alguns dos principais algoritmos presentes na biblioteca disponibilizada pela Xilinx.

2.1 Canal rádio multi-percurso

A figura 2.1 ilustra um diagrama de blocos simplificado de um sistema de comunicações sem fios ponto a ponto, com um transmissor e um recetor. O transmissor tem como funções codificar a informação a ser transmitida, modular o sinal codificado para a banda de frequência alocada, e transmitir-lo na forma de ondas eletromagnéticas. As funções do recetor consistem na receção e no processamento dessas ondas eletromagnéticas, de forma a produzirem uma estimação fiável da informação transmitida. Um parâmetro importante nos sistemas de comunicações sem fios é o espectro alocado para a transmissão. Este determina a banda de frequência na qual as comunicações sem fios podem ser transmitidas e também a largura de banda do sistema de comunicação.

A transmissão normalmente ocorre em lugares com vários obstáculos entre o transmissor e o recetor, como por exemplo, edifícios ou árvores, tendo estes uma grande influência na performance do sistema. Esta influência deve-se ao facto destes obstáculos refletirem o sinal transmitido, originando assim L_p réplicas do sinal original. Estas réplicas são provenientes dos diferentes caminhos, onde cada caminho tem um atraso de propagação e

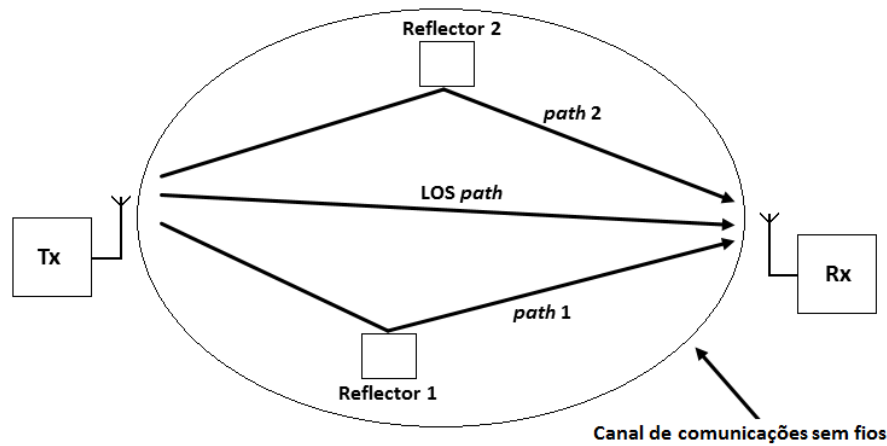


Figura 2.1: Diagrama de blocos que ilustra os efeitos da propagação em canais sem fios

uma atenuação específica. Este fenómeno é referido como propagação por multi-percurso, originando assim o canal rádio multi-percurso. O atraso máximo de propagação destes canais é definido como a diferença entre o tempo de chegada da primeira e última réplica do sinal recebido, correspondendo ao caminho mais curto e mais longo. Devido ao efeito de desvanecimento causado pelo multi-percurso, pequenas alterações na separação espacial entre o recetor e o transmissor provocam mudanças dramáticas no sinal recebido. Devido à rapidez destas alterações, este fenómeno é também chamado de desvanecimento rápido, podendo estas alterações verificar-se na amplitude, fase e ângulo de chegada do sinal recebido.

O sinal transmitido pode ser afetado por três tipos de propagação, sendo estes, reflexões, dispersões e difrações [6, 7]. No caso das reflexões, estas ocorrem quando a onda electromagnética transmitida é refletida numa superfície cujas dimensões são muito superiores ao comprimento de onda do sinal transmitido. As dispersões ocorrem nas condições opostas às anteriores, ou seja, quando o tamanho da superfície é inferior ao comprimento de onda do sinal transmitido, fazendo com que a energia do sinal seja refletida em várias direções. A difração ocorre quando os obstáculos obstruem o caminho LOS entre o transmissor e o recetor, chegando apenas ao recetor ondas provenientes de caminhos secundários. A sigla LOS (*Line of Sight*) significa que entre o transmissor e o recetor existe um caminho em linha de vista, chegando assim ao recetor um sinal com energia dominante.

As características dos canais rádio são revistas em vários livros, como por exemplo, [7, 6]. Quanto ao comportamento do canal, este é caracterizado por provocar desvanecimento no sinal recebido. Isto significa que as características do canal de propagação RF, entre dois pontos, variam continuamente no tempo. Portanto, o sinal recebido no recetor também varia no tempo apesar da potência transmitida pelo emissor ser constante. O desvanecimento é normalmente classificado de acordo com o seu comportamento em

função da frequência do sinal, podendo este ser classificado como seletivo ou não seletivo. No caso do desvanecimento seletivo, este afeta apenas algumas frequências que formam o sinal RF. Quando toda a banda do sinal RF é afetada de maneira semelhante, o desvanecimento é caracterizado como não seletivo na frequência. O canal pode ser classificado como tendo um desvanecimento *Rayleigh*, no caso do canal multi-percurso não possuir uma componente LOS. Podendo também ser classificado como tendo um desvanecimento *Ricean*, isto quando chega ao recetor uma componente com energia dominante proveniente do sinal LOS.

No caso da largura de banda do sistema ser pequena, a duração do símbolo é normalmente muito superior ao atraso máximo de propagação do canal. Neste caso, o desvanecimento do canal é caracterizado pela soma de diversas amostras do mesmo sinal no recetor com diferentes amplitudes, fases e atrasos, devido aos diferentes caminhos entre o recetor e o transmissor. No domínio da frequência, este tipo de canal tem uma resposta constante na banda utilizada, não introduzindo portanto distorções significativas na frequência.

Quando a largura de banda do sinal é grande, a duração do símbolo é menor que o atraso máximo de propagação do canal. Neste caso, o canal pode ser visto como um filtro FIR, sendo a resposta do canal diferente para as várias frequências dentro da banda de transmissão [8]. Portanto, este canal possui um desvanecimento seletivo na frequência. No domínio do tempo, este canal causa ISI no sinal recebido, o que pode degradar significativamente a performance do sistema. A sigla ISI significa *Inter-Symbol Interference*, sendo esta usada para indicar a ocorrência de interferência entre símbolos diferentes, ou seja, símbolos transmitidos em tempos distintos. Esta interferência ocorre devido à sobreposição das amostras destes símbolos no domínio do tempo, devido aos diferentes atrasos de propagação. É de realçar que as distorções introduzidas pelo canal dependem da relação entre os atrasos de propagação no canal e da largura de banda do sinal. Podendo o mesmo canal ser seletivo na frequência e constante, dependendo da largura de banda do sinal transmitido.

Para além do multi-percurso, o sinal transmitido também é afetado por ruído, usualmente modelado por ruído AWGN. As alterações causadas pelo canal AWGN são a adição de ruído branco com uma densidade espectral constante, em que a sua amplitude segue uma distribuição Gaussiana. Este tipo de ruído é proveniente de fontes naturais, como vibrações térmicas dos átomos nos condutores, da terra e outros objetos quentes, como é o caso do sol.

2.2 Esquemas de Multiplexação

2.2.1 Multiplexação FDD e TDD

Frequency Division Duplex e *Time Division Duplex* são os dois esquemas de multiplexação mais prevalentes usados em redes sem fios de banda larga. O esquema FDD ficou principalmente conhecido pelo seu uso em aplicações de voz. Usando canais distintos para *upstream* e *downstream*, permite assim transmissões simultâneas em ambas as direções. O facto de os canais terem uma largura de banda fixa faz com que este esquema seja ideal para aplicações com tráfego simétrico em ambas as direções, como é o caso de comunicações de voz. Contudo, pode também ser usado para aplicações com tráfego assimétrico, apesar de ser espectralmente ineficiente, uma vez que o tráfego recebido na frequência de *downlink* é muito superior ao tráfego transmitido na frequência de *uplink*.

No caso do esquema TDD, este usa apenas uma banda de frequência para transmitir os sinais em ambas as direções. A comunicação é baseada em *slots* temporais, ou seja, o recetor e o transmissor transmitem em tempos diferentes, sendo isto imperceptível ao utilizador. Este princípio de funcionamento permite tanto comunicações simétricas como assimétricas, através da atribuição de espaços temporais maiores para *downstream* do que para *upstream*. Na figura 2.2 estão ilustrados ambos os esquemas de multiplexação FDD e TDD.

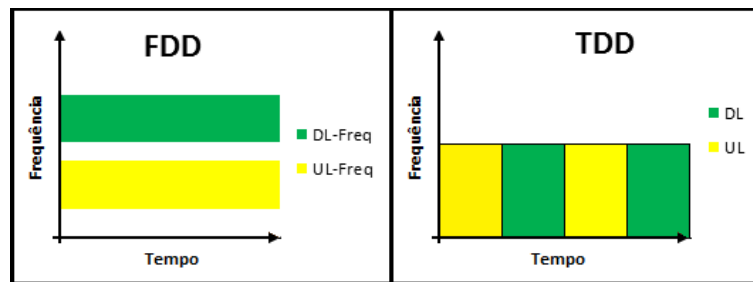


Figura 2.2: Esquemas de multiplexação FDD e TDD.

2.2.2 Acesso múltiplo TDMA, FDMA e CDMA

As siglas TDMA, FDMA e CDMA significam *Time Division Multiple Access*, *Frequency Division Multiple Access* e *Code Division Multiple Access*, respectivamente. Estas tecnologias têm como objetivo fazer um aproveitamento espectral, permitindo que múltiplos utilizadores utilizem o mesmo canal físico. No caso do TDMA, isto é conseguido dividindo o canal em espaços temporais, espaços estes atribuídos a cada utilizador tanto para transmitir como para receber. Quanto ao *FDMA*, os diferentes sinais são atribuídos a canais com frequências diferentes. No caso do *CDMA*, todos transmitem ao mesmo tempo,

portanto, o recetor tem de conseguir distinguir os sinais dos diferentes utilizadores. Isto é conseguido usando uma modulação denominada *Spread Spectrum*, tendo como princípio de funcionamento uma modulação pseudoaleatória. Esta modulação é aplicada e removida no domínio digital, permitindo recuperar os sinais uma vez que as sequências associadas aos utilizadores são ortogonais entre si, permitindo assim que múltiplos utilizadores ocupem a mesma banda de frequência. Na figura 2.3 pode ser observado o princípio das três tecnologias descritas acima.

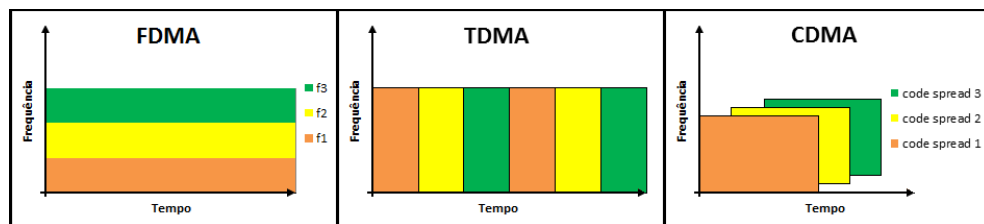


Figura 2.3: Esquemas de multiplexação TDMA, FDMA e CDMA.

2.2.3 OFDM

Nesta secção é feita uma revisão detalhada da técnica de multiplexação OFDM, demonstrando as potencialidades deste método. É realizada uma descrição da estrutura destes sistemas e das características que os tornam robustos e adequados a ambientes com baixos SNRs.

2.2.3.1 Múltiplas sub-portadoras e ortogonalidade

A sigla OFDM tem o significado *Orthogonal Frequency Division Multiplexing*, e é um método de multiplexação que divide a largura de banda disponível em várias sub-portadoras de banda estreita. Um símbolo OFDM resulta da combinação de todas as sub-portadoras, sendo o número de sub-portadoras definido por N_c . Como a transmissão é feita em vários *streams* em paralelo em vez de um simples *stream*, a duração do símbolo torna-se relativamente longa quando comparado com as características temporais do canal. Portanto, os sistemas OFDM apresentam uma elevada robustez ao desvanecimento presente nos canais com multi-percurso, e também à interferência inter-simbólica, ISI. A sigla ISI significa *Inter-Symbol Interference*, esta é classificada como uma forma de distorção do sinal transmitido, em que um dos símbolos interfere com um dos subsequentes. Modulando a informação com um dos esquemas de modulação M-PSK ou M-QAM, é possível manter um débito semelhante ao de um sistema de um único *stream*.

A característica principal do OFDM é aplicar o princípio da ortogonalidade entre as sub-portadoras, para que estas não interfiram entre si. Uma vez que um dos reflexos

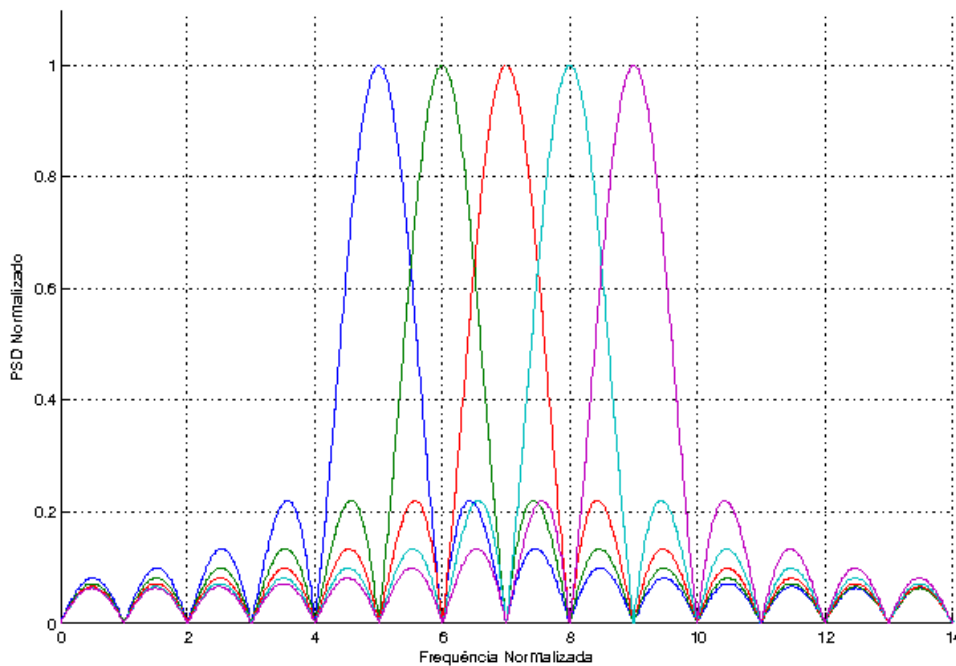


Figura 2.4: Espectro de 5 sub-portadoras de um sinal OFDM.

da ortogonalidade é que os máximos das sub-portadoras coincidem com os zeros das restantes, figura 2.4. Recorrendo ao uso da IFFT, os símbolos são convertidos do domínio da frequência para o domínio do tempo sendo ortogonais entre si devido às propriedades da IFFT. No lado do recetor é usada a FFT para converter novamente os símbolos para o domínio da frequência, os algoritmos IFFT/FFT serão explicados com maior detalhe na secção 2.4.4.

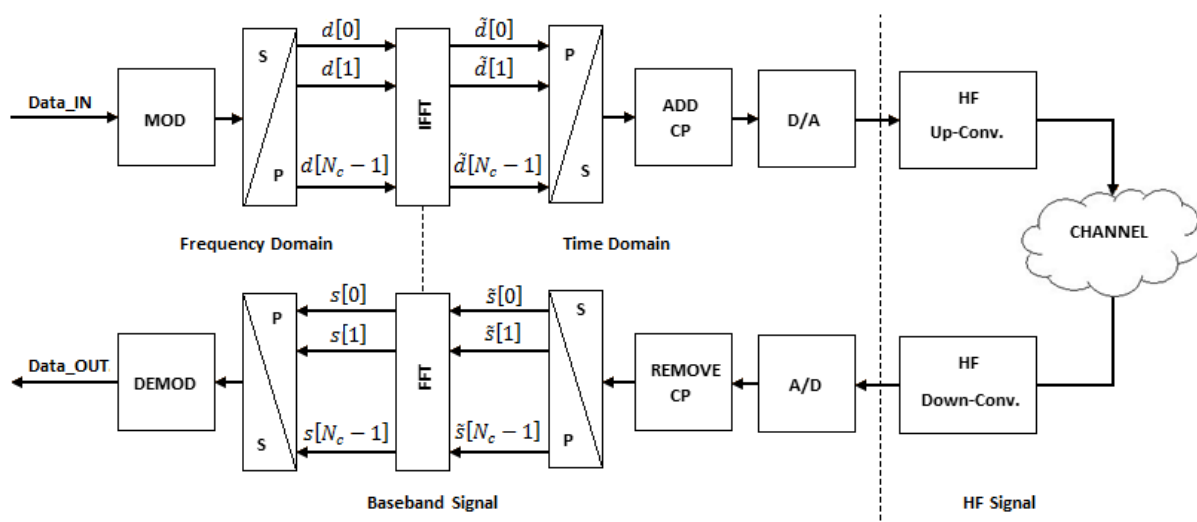


Figura 2.5: Estrutura de um sistema OFDM.

A estrutura de funcionamento do OFDM encontra-se ilustrada na figura 2.5. Onde o bloco *MOD* modula os bits da entrada *Data_in* em símbolos de 2^M níveis, onde M é o

número de bits transportado num símbolo. Assim, considerando que r_b representa o débito binário da entrada, o débito simbólico será $r_s = r_b/M$. Na transmissão por múltiplas sub-portadoras, cada sub-portadora é individualmente modulada por um modulador deste tipo, permitindo assim, que um símbolo OFDM transporte símbolos com diferentes débitos binários. Após a passagem pelo bloco *IFFT*, é adicionado o prefixo cíclico, este será explicado em detalhe na secção seguinte. De seguida o sinal é convertido do domínio digital para o analógico através de um DAC, sendo este posteriormente convertido da frequência em banda base para a frequência de transmissão. No recetor verifica-se o processo inverso, onde o sinal recebido à frequência de transmissão é convertido do domínio analógico para o digital através de um ADC. De seguida o bloco *REMOVE CP* faz a remoção do prefixo cíclico, enquanto o bloco seguinte, *FFT*, converte o sinal novamente para o domínio da frequência. Por fim, cada sub-portadora é desmodulada de acordo com a modulação pela qual foi modulado no transmissor.

2.2.3.2 Prefixo Cíclico

Tal como mencionado acima, o OFDM tem uma boa performance a lidar com canais com multi-percurso, sendo menos sensível às reflexões que causam ISI do que sistemas de uma única portadora. Para além de o tempo de símbolo ser superior, é também usado um tempo de guarda entre símbolos com o objetivo de limitar este efeito. Este tempo de guarda é preenchido com uma cópia, denominada de prefixo cíclico (CP), das últimas amostras do símbolo onde o número de amostras copiadas é definido por L , figura 2.6.

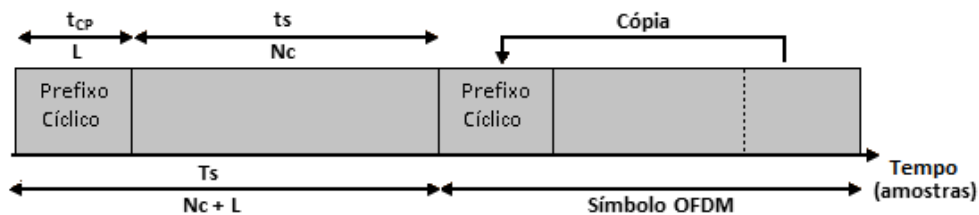


Figura 2.6: Estrutura do Prefixo Cíclico

Na figura 2.7 pode observa-se de que forma é que o CP contribui para evitar a interferência inter-simbólica. Os símbolos no topo representam os símbolos OFDM que estão a ser transmitidos, sendo estes recebidos com diferentes atrasos, dependendo dos caminhos entre o transmissor e o recetor. No caso da duração do CP ser superior ao maior atraso, não existirá ISI, uma vez que o recetor descarta o CP e as únicas amostras que se sobrepõem pertencem ao mesmo símbolo.

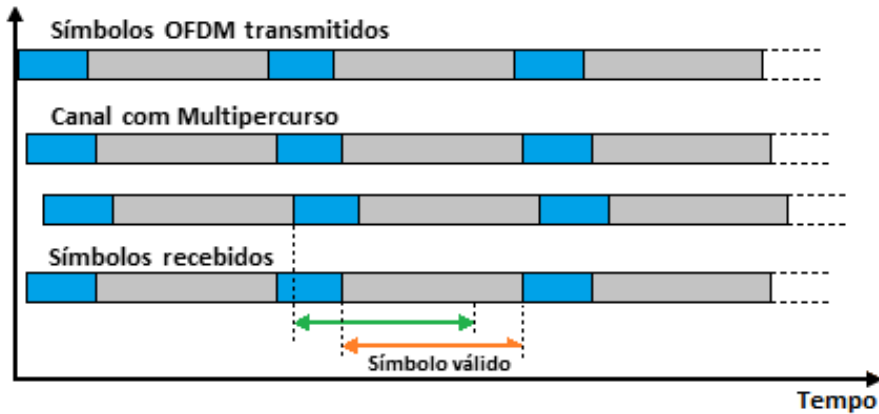


Figura 2.7: Aumento da robustez contra ISI devido ao uso do CP.

2.2.3.3 Sincronização, Equalização do canal e Modulação das sub-portadoras

Na maioria dos sistemas sem fios a sincronização é um dos papéis mais importantes do recetor. Pelo facto dos sistemas OFDM utilizarem múltiplas sub-portadoras, estes são bastante sensíveis a desvios de frequência, sendo portanto necessário encontrar este desvio e também o atraso temporal. De forma a estimar estes parâmetros é adicionada redundância ao sinal transmitido, normalmente denominada por pilotos. Os pilotos não são mais do que símbolos conhecidos transmitidos em algumas das sub-portadoras usadas na transmissão. As letras d e p serão usadas para denominar os vectores que contêm os valores dos dados e dos pilotos, respectivamente, no domínio da frequência. Podendo o sinal transmitido s , no domínio da frequência, ser definido como $s = d + p$.

As sequências de pilotos usadas nos sistema OFDM podem ser definidas como:

$$p[k] = \sum_{m=0}^{N_t-1} \delta[k - k_{ini} - mN_f], \quad (2.1)$$

onde $N_t = \frac{N_c}{N_f}$ e $k_{ini} \in \{0, \dots, N_f - 1\}$ é a primeira sub-portadora com piloto. A observação da equação 2.1, permite perceber que os símbolos pilotos no domínio da frequência estão regularmente espalhados a cada N_f sub-portadoras. A separação dos pilotos pode variar de 1 (caso particular em que todas as sub-portadoras do símbolo OFDM transportam pilotos) a N_c . Usando o algoritmo IFFT de N_c pontos, o vector de entrada é convertido do domínio da frequência para o domínio do tempo. Portanto, após esta operação, o vector \tilde{p} no domínio do tempo toma a seguinte forma:

$$\begin{aligned} \tilde{p}[n] &= N_c^{-1/2} \sum_{k=0}^{N_c-1} \sum_{m=0}^{N_t-1} \delta[k - k_{ini} - mN_f] \exp\left(j \frac{2\pi}{N_c} kn\right) \\ &= N_c^{-1/2} N_f^{-1} \exp\left(j \frac{2\pi}{N_c} k_{ini} n\right) \sum_{m=0}^{N_f-1} \delta[n - mN_t] \end{aligned} \quad (2.2)$$

Quanto aos elementos de dados, \tilde{d} , no domínio do tempo são definidos como:

$$\tilde{d}[n] = N_c^{-1/2} \sum_{k=0}^{N_c-1} d[k] \exp\left(j \frac{2\pi}{N_c} kn\right), \quad (2.3)$$

onde $d[k]$ representa os símbolos de dados no domínio da frequência. Os sinais piloto são conhecidos no recetor e são usados para efeitos de sincronização e/ou equalização. A densidade de pilotos no sinal transmitido é proporcional à qualidade do processo de sincronização, mas é inversamente proporcional à quantidade de dados transmitidos. Como referido anteriormente, cada sub-portadora pode usar uma modulação diferente para transmitir os seus dados. Esta possibilidade permite otimizar a transmissão, uma vez que possibilita adequar a modulação ao SNR de cada sub-portadora. Assim, as sub-portadoras menos afetadas pelo canal podem usar modulações com um número maior de bits por símbolo, enquanto as com SNRs inferiores utilizam modulações mais facilmente recebidas, como o BPSK.

2.2.4 OFDMA

Orthogonal Frequency Division Multiple Access é uma versão do esquema de modulação OFDM mas para múltiplos utilizadores, resultante da combinação do OFDM com o protocolo FDMA. O princípio deste esquema de multiplexação baseia-se na divisão das sub-portadoras disponíveis em grupos, sendo estes atribuídos a diferentes utilizadores para transmissão em simultâneo. Através desta estratégia o sistema ganha uma maior flexibilidade em termos de gestão dos recursos, comparando com um sistema OFDM. A figura 2.8 ilustra a alocação de sub-portadoras a diferentes utilizadores, isto no caso de sistemas OFDM e OFDMA. Para além da maior flexibilidade, o OFDMA mantém a capacidade de compensar as distorções causadas pelo canal no domínio da frequência, sem requerer o uso de equalizadores no domínio do tempo.

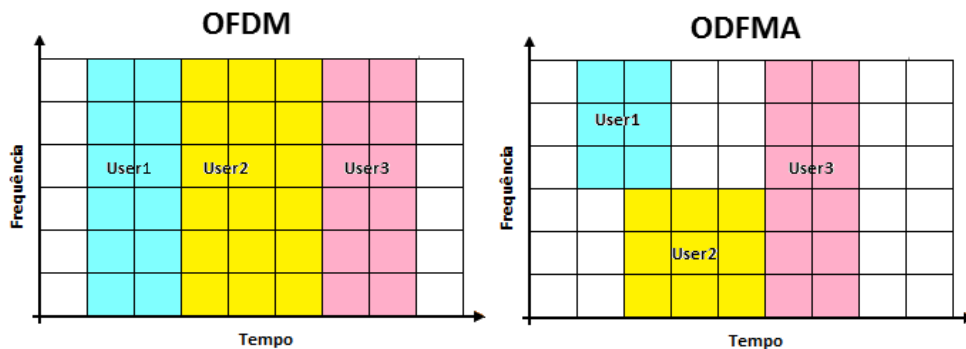


Figura 2.8: Alocação de sub-portadoras a diferentes utilizadores, no caso de sistemas OFDM e OFDMA.

Existem três métodos possíveis de distribuição das sub-portadoras pelos vários utilizadores, estando estes ilustrados na figura 2.9. No caso da divisão em sub-bandas CAS, figura 2.9 a), cada sub-canal é composto por um grupo de sub-portadoras adjacentes. Este método não tem em conta a diversidade espectral oferecida pelos canais com multi-percurso, uma vez que um desvanecimento profundo pode atingir um número significativo de sub-portadoras de um dado utilizador. Outro método é o CAS entrelaçado, figura 2.9 b), onde as sub-portadoras de cada utilizador estão uniformemente divididas pela largura de banda do sinal. Sendo o espaçamento entre sub-portadoras de um utilizador, o número de sub-canais. Uma vez que o OFDMA permite uma alocação flexível, podem ser os utilizadores a escolher as melhores sub-portadoras, por exemplo com maiores SNRs. Esta estratégia é denominada de CAS generalizado, estando ilustrada na figura 2.9 c). As sub-portadoras são associadas aos utilizadores de forma dinâmica, dando uma maior flexibilidade do que as estratégias anteriores.

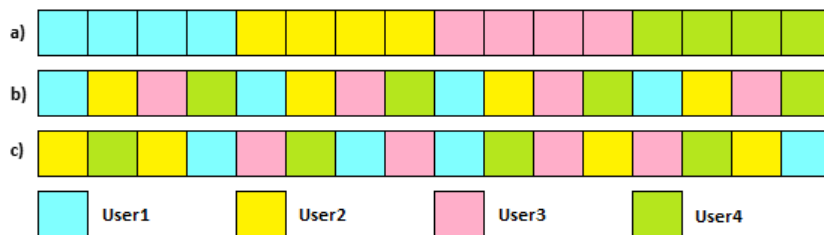


Figura 2.9: Métodos possíveis de distribuição das sub-portadoras pelos diferentes utilizadores, em sistemas OFDMA.

2.2.5 SC-FDMA

A sigla SC-FDMA significa *Single-Carrier* FDMA, tendo adquirido este nome devido ao sinal transmitido aparentar ser uma transmissão de portadora única. O SC-FDMA pode ser visto como um esquema OFDMA pré-codificado, uma vez que a única diferença é a inclusão de um processamento DFT antes do processamento convencional do OFDMA. Portanto, enquanto no OFDMA os símbolos modulam diretamente as múltiplas sub-portadoras, no SC-FDMA os símbolos são numa primeira fase codificados usando a DFT, figura 2.10. É de salientar que o SC-FDMA também adquire do OFDM a capacidade de eliminar a dispersão temporal causada pela propagação em multi-percurso.

O SC-FDMA é usado para reduzir o PAR, sendo este um ponto crítico nas transmissões *uplink* quando usando OFDMA. Quanto à estrutura do sinal, no OFDMA cada sub-portadora apenas contém informação relativa a um símbolo de dados, enquanto no SC-FDMA cada sub-portadora transporta informação de todos os símbolos. Ambos os símbolos são representados na figura 2.11.

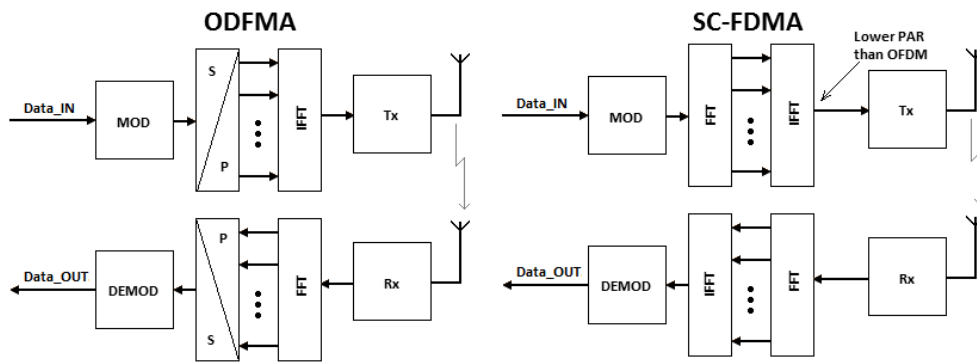


Figura 2.10: Estruturas dos sistemas OFDMA e SC_FDMA.

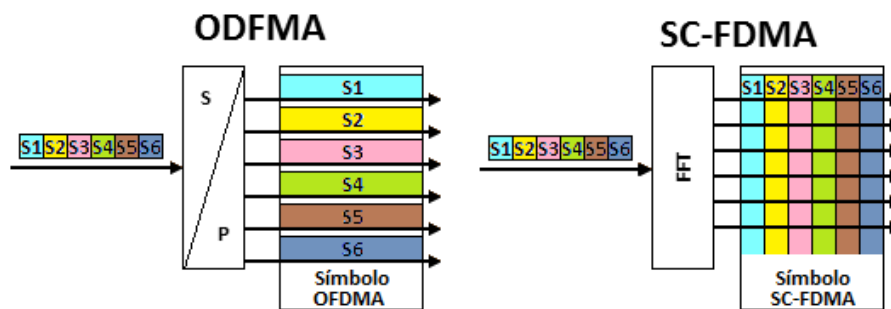


Figura 2.11: Aspecto dos símbolos OFDMA e SC-FDMA no domínio do tempo.

Esta multiplexação pode ser implementada de duas formas, denominadas FDMA localizado e FDMA distribuído. No caso do FDMA localizado, as saídas da FFT são ordenadamente inseridas nas entradas da IFFT, como ilustrado no esquema da esquerda da figura 2.12. Enquanto no caso do FDMA distribuído, as saídas da FFT são divididas pelas entradas da IFFT, existindo assim espaços entre as entradas utilizadas, ilustrado no lado direito da figura 2.12. A diferença entre estas formas de multiplexação reflete-se na saída. O primeiro método resulta numa interpolação no domínio do tempo. No caso do segundo método, o resultado é uma sequência repetida no domínio do tempo. Ambos os sinais, no domínio da frequência e do tempo, destes dois métodos estão ilustrados também na figura 2.12.

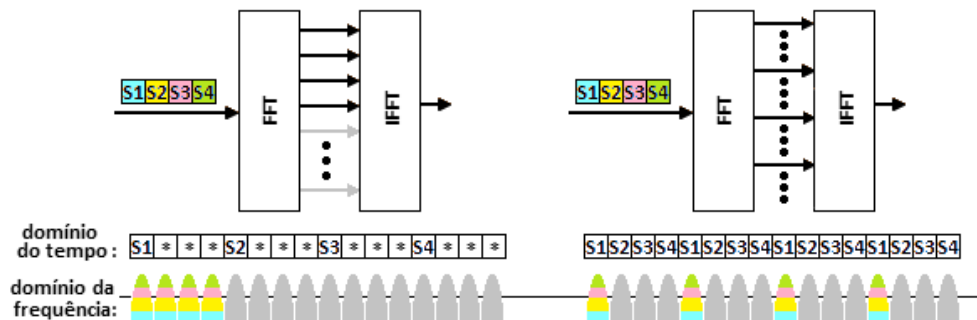


Figura 2.12: Aspecto dos símbolos SC-FDMA, no domínio do tempo e frequência, no caso da multiplexação ser implementada de forma localizada ou distribuída.

2.3 Long Term Evolution (LTE)

O LTE é um *standard* [9] de comunicações *wireless* 4G desenvolvido pelo 3rd Generation Partnership Project (3GPP). Este *standard* foi o passo seguinte na evolução da tecnologia UMTS, conseguindo transmissões com débitos de 100Mb/s no *downlink* e 50Mb/s no *uplink*, assim como redução da latência na transmissão dos pacotes. O LTE suporta larguras de banda entre 1.4 e 20 MHz, sendo estas escaláveis de forma a assegurar a compatibilidade com os sistemas de radio móveis existentes. Quanto ao *downlink* é empregado o método OFDMA, enquanto o *uplink* utiliza o esquema SC-FDMA, tendo sido ambos explicados nas secções 2.2.4 e 2.2.5, respectivamente. Relembrando que este método é semelhante ao OFDMA, mas com a vantagem que o sinal SC-FDMA exibe um PAR inferior. O LTE recorre a duas técnicas para separar as direcções da transmissão, sendo estas o FDD e o TDD. Tal como explicado na secção 2.2.1, no caso do FDD, o *downlink* e *uplink* são transmitidos usando frequências diferentes. Enquanto no modo TDD, o *uplink* e *downlink* são processados na mesma frequência e a separação ocorre portanto no domínio do tempo, recorrendo a *timeslots* específicos. De seguida será feita uma descrição da tecnologia TDD do LTE (TD-LTE), sendo também destacadas as principais diferenças para com a tecnologia FDD do LTE (FD-LTE).

2.3.1 LTE: Estrutura das *frames* e *sub-frames*

A *frame* usada no FD-LTE tem uma duração total de 10ms, sendo estes divididos em 20 *timeslots* individuais, figura 2.13. O conjunto de dois *timeslots* é denominado de *sub-frame*, existindo portanto 10 *sub-frames* em cada *frame*.

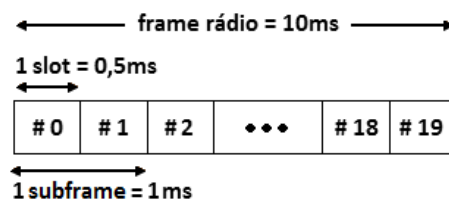


Figura 2.13: Estrutura da *frame* FD-LTE.

Quanto à estrutura da *frame* usada no TD-LTE, cada *frame* tem também uma duração de 10ms, figura 2.14. Esta é dividida em duas metades de comprimento idêntico, sendo cada uma destas dividida em cinco *sub-frames*, cada uma com a duração de 1ms. As *sub-frames* podem ser classificadas como *sub-frames standard* ou *sub-frames especiais*, estas últimas são constituídas por três campos DwPTS, GP e UpPTS. Estes campos significam, respectivamente, *Downlink Pilot Time Slot*, *Guard Period* e *Uplink Pilot Time Slot*. Estes podem ser individualmente configurados em termos de duração, contudo a duração total

dos três campos tem de ser 1ms.

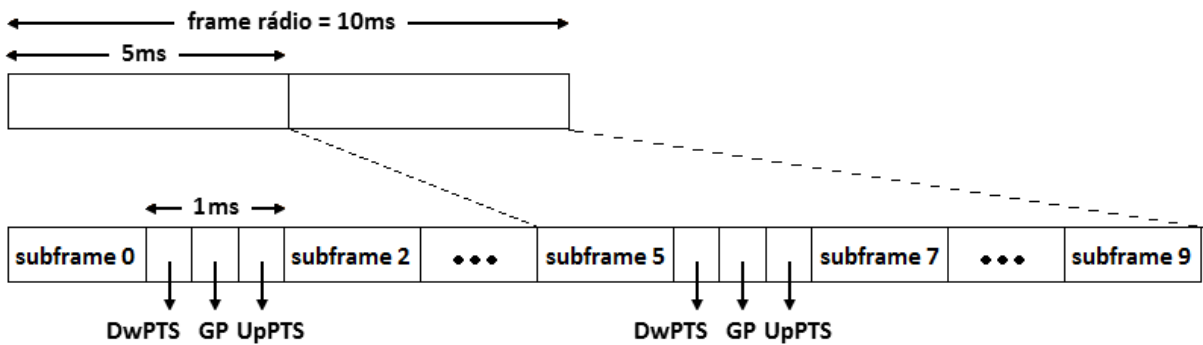


Figura 2.14: Estrutura da *frame* TD-LTE.

Como o TD-LTE usa as mesmas bandas de frequência para o *uplink* e *downlink*, as direções de transmissão são separadas usando diferentes *sub-frames*. A distribuição das *sub-frames* pode ser feita de forma simétrica, onde o número de *sub-frames* é igual para o *uplink* e *downlink*, como de forma assimétrica. A tabela 2.1 mostra as configurações de *uplink/downlink* definidas no TD-LTE. A letra "D" é usada para denominar que a *sub-frame* transporta dados de *downlink*, similarmente, "U" indica um transmissão de dados de *uplink* e "S" especifica os dados utilizados para sincronização. É também possível

Tabela 2.1: Alocação espectral do esquema TD-LTE.

Uplink-Downlink config.	Número da <i>sub-frame</i>									
	0	1	2	3	4	5	6	7	8	9
0	D	S	U	U	U	D	S	U	U	U
1	D	S	U	U	D	D	S	U	U	D
2	D	S	U	D	D	D	S	U	D	D
3	D	S	U	U	U	D	D	D	D	D
4	D	S	U	U	D	D	D	D	D	D
5	D	S	U	D	D	D	D	D	D	D
6	D	S	U	U	U	D	S	U	U	D

observar que as *sub-frames* 0 e 5 são sempre usadas para *downlink*. Para além disto, a *sub-frame* imediatamente a seguir às *sub-frames* usadas para transporte de dados especiais, é sempre usada para transmitir dados de *uplink*.

2.3.1.1 Estrutura dos recursos

A estrutura dos recursos é exatamente igual tanto para o TD-LTE como para o FD-LTE. A unidade de recursos mais pequena no domínio do tempo é um símbolo OFDM no *downlink* e um símbolo SC-FDMA no *uplink*. O número de símbolos OFDM/SC-FDMA

num *slot* depende do tamanho do prefixo cíclico que está a ser usado. Os parâmetros do *uplink* e *downlink* estão listados na tabela 2.2, onde é possível observar que no caso do espaçamento entre sub-portadoras ser de 15 kHz, são transmitidos 7 símbolos (CP normal) ou 6 símbolos (CP alargado) em cada *timeslot* (0.5ms). A menor unidade dimensional para atribuição dos recursos no domínio da frequência é denominada de *resource block* (RB), correspondendo a 12 sub-portadoras. Recorrendo à tabela 2.2, no caso do CP ter um tamanho normal e a transmissão ser no sentido de *downlink*, o espaçamento entre sub-portadoras é de 15 kHz, portanto, cada RB tem uma largura de banda de 180 kHz.

Tabela 2.2: Parametrização do *uplink/downlink* do LTE [4].

	Configuração	Espaçamento entre sub-portadoras (Δf)	N_c	N_{symb}
Downlink	CP normal	$\Delta f=15$ kHz	12	7
	CP estendido	$\Delta f=15$ kHz	12	6
		$\Delta f=7$ kHz	24	3
Uplink	CP normal	$\Delta f=15$ kHz	12	7
	CP estendido	$\Delta f=15$ kHz	12	6

Tal como já tinha sido referido, o LTE suporta várias larguras de banda diferentes, tornando-o compatível com as redes móveis de rádio existentes. A largura de banda do canal é definida pelo número de *resource block*, N_{RB} , sendo esta escalável. Esta escalabilidade permite que os recursos rádio sejam usados eficientemente. A tabela 2.3 lista as larguras de bandas suportadas pelo LTE e o número de blocos de recursos associados N_{RB} . Estes parâmetros estão definidos tanto para o TD-LTE como para o FD-LTE.

Tabela 2.3: Larguras de banda do LTE [4].

Largura de banda do canal (MHz)	1.4	3	5	10	15	20
N_{RB}	6	15	25	50	75	100

Durante a transmissão de uma *frame* rádio TD-LTE, existem várias alterações do sentido da transmissão (*downlink/uplink*). Durante o processo, os diferentes tempos de transição entre a estação base e as várias estações móveis devem ser tidos em consideração, de forma a prevenir conflitos com as *sub-frames* vizinhas. Portanto, cada estação móvel recebe informação fornecida pela estação base de quando deve começar a transmitir. Quanto maior for a distância entre estação base e o móvel, mais cedo a estação móvel deve começar a transmitir. Isto ajuda a assegurar que todos os sinais chegam à estação base de uma forma sincronizada. Na transição de *downlink* para *uplink*, é inserido um período de guarda (GP) entre os campos DwPTS e UpPTS. A duração do GP depende do tempo de propagação do sinal e também do tempo requerido para a estação móvel alterar do modo de receção para o modo de transmissão. Esta duração é configurada pela rede com base no tamanho da célula.

Em conclusão, uma *frame* tem a duração de 10ms e é constituída por 10 *sub-frames*, onde cada *sub-frame* tem a duração de 1ms, contendo 2 *timeslots*. Assim, cada *timeslot* tem a duração de 0.5ms e contém N_{RB} *resource blocks*, onde $6 < N_{RB} < 110$, dependendo da largura de banda alocada e dos recursos disponíveis.

2.4 FPGAs

Nesta secção é introduzida a plataforma utilizada para o desenvolvimento da cadeia OFDM implementada nesta tese. São também revistos alguns dos principais componentes, disponibilizados na biblioteca da Xilinx, utilizados na implementação do sistema desenvolvido.

2.4.1 Xilinx System Generator

No capítulo 1 foram apresentados os grandes avanços tecnológicos referentes a *hardware* reconfigurável, FPGAs. Devido à grande evolução nesta área, a concepção de *hardware* RTL tradicional obviamente não poderia seguir a tendência de desenvolvimento a curto prazo. Para completar a falha existente entre a modelação do algoritmo e a implementação final do *hardware* na FPGA foi criada a plataforma *Xilinx System Generator*, sendo esta explicada de seguida. O *System Generator* é uma nova ferramenta de *design* em FPGAs, tendo sido desenvolvido pela *Xilinx Inc.* em conjunto com a *Mathworks Inc.* A utilização deste *software* permite que os *chips* da *Xilinx* sejam programados utilizando apenas as ferramentas do *Simulink* do *Matlab*, tornando o sistema de simulação e verificação mais fácil. Para além disto, serve também como ligação entre o *design* do sistema DSP de alto nível e a implementação de *hardware* na FPGA, reduzindo o tempo de desenvolvimento do algoritmo [10]. Os utilizadores necessitam apenas de construir o modelo no *Simulink* usando os módulos presentes nas bibliotecas fornecidas pela *Xilinx*, denominadas *Xilinx Blockset*, *Xilinx Reference Blockset* e *Xilinx XtremeDSP Kit*. Estas bibliotecas incluem módulos básicos, como módulos de processamento digital de sinal (FIR e FFT), módulos matemáticos, módulos de memórias e módulos de interfaces. Posteriormente o *System Generator* faz a conversão do código de *Matlab* do *Simulink* destes módulos para código VHDL, código este reconhecido pelo ISE da *Xilinx*.

A solução baseada no *System Generator* pode então encurtar o processo de desenvolvimento, aumentar a eficiência, e aumentar a taxa de utilização dos recursos, tornando o desenvolvimento em FPGAs muito mais fácil. Há razões para acreditar que o desenvolvimento de *hardware* baseado no *System Generator* trará uma maior performance e incentivará ainda mais ao uso de FPGAs nos sistemas de comunicação multimédia. Muitos algoritmos de comunicação sem fios requerem a avaliação de funções elementares, como

operações trigonométricas, que não são suportadas pela biblioteca *ModelSim* das FPGA. O algoritmo CORDIC ganhou então muita atenção [11], sendo este normalmente usado como alternativa para a computação aritmética destas funções elementares.

2.4.2 Algoritmo CORDIC

O CORDIC foi em 1959 descrito por J.E. Volder [12] como uma maneira elegante de avaliar funções trigonométricas. Em 1971, J. Walther [13] estendeu o algoritmo CORDIC para funções hiperbólicas, sendo este algoritmo utilizado hoje em dia em vários campos científicos e de engenharia. Alguns dos fins para os quais o CORDIC é usado são computação matricial, processamento digital de sinal, processamento digital de imagens, comunicações, gráficos, entre outros [14]. O CORDIC ganhou muita atenção devido à demonstração feita por J. Walther. Este mostrou que com a modificação de apenas uns parâmetros, é possível formar um algoritmo para implementação unificada de um grande leque de funções trigonométricas elementares. Enquanto isso, D. Cochran [15] deu exemplos de vários algoritmos que poderiam ser substituídos pela técnica CORDIC. Tornando-se muito popular devido ao seu potencial para implementações eficientes e de baixo custo. De seguida serão explicados dois dos modos de operação do CORDIC, tendo estes sido utilizados para a estimação e compensação do CFO, Capítulo 4.

2.4.2.1 CORDIC: Modo rotação

O algoritmo CORDIC foi inicialmente concebido para fazer a rotação de um vector no plano xy . Como ilustrado na figura 2.15, o vector de entrada V_{in} , com um ângulo β , é rodado por ângulo α . O vector resultante é obtido de acordo com a seguinte relação [16]:

$$\begin{aligned} X_{out} &= |V_{in}| \cos(\beta + \alpha) = X_{in} \cos \alpha - Y_{in} \sin \alpha \\ Y_{out} &= |V_{in}| \sin(\beta + \alpha) = Y_{in} \cos \alpha + X_{in} \sin \alpha \end{aligned} \quad (2.4)$$

Tal como foi referido em cima, o CORDIC pode ser utilizado para vários fins. De forma a mostrar a versatilidade deste algoritmo, caso a entrada Y_{in} seja zero, nas saídas X_{out} e Y_{out} são colocados em simultâneo os valores do cosseno e seno do ângulo α , respectivamente.

2.4.2.2 CORDIC: Modo translação

O CORDIC quando usado neste modo faz a conversão de coordenadas cartesianas para polares. Na figura 2.16 encontra-se a representação de um número complexo em coorde-

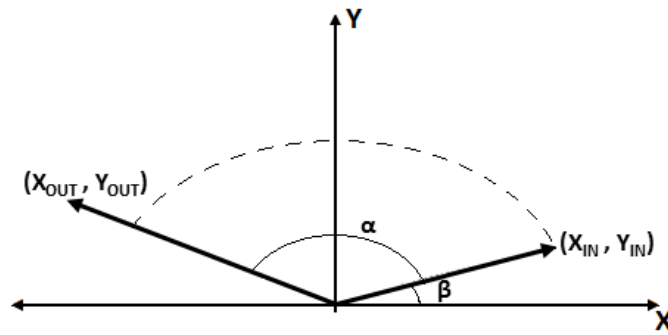


Figura 2.15: Rotação do vetor de entrada por um ângulo α .

nadas cartesianas. A representação em coordenadas polares é expressa na seguinte forma $Z = |Z|e^{j\alpha}$, em que $|Z|$ é o módulo e α é fase.

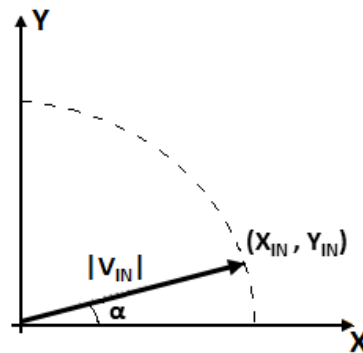


Figura 2.16: Translação de um vetor (coordenadas cartesianas para polares).

O CORDIC faz o cálculo do módulo e da fase dos números da entrada, X_{in} e Y_{in} , da seguinte forma:

$$\begin{aligned} |Z| &= \sqrt{X_{in}^2 + Y_{in}^2} \\ \alpha &= \arctan\left(\frac{Y_{in}}{X_{in}}\right) \end{aligned} \quad (2.5)$$

Este modo de funcionamento é particularmente útil para o cálculo do desvio de fase ocorrido nos dados, ou seja, na estimação do CFO. Enquanto o modo de rotação é utilizado para compensação do CFO, uma vez que faz a rotação dos dados de acordo com um dado ângulo de entrada. De forma a compensar o desvio de fase, o ângulo introduzido será o simétrico do ângulo estimado.

2.4.3 Direct Digital Synthesizer (DDS)

Os componentes *Direct Digital Synthesizers* geram sinais com grande resolução de frequência e elevadas velocidades de comutação de frequência. Estes são muito importantes nos sistemas de comunicações digitais, podendo ser usados na construção de conversores digitais *up* e *down*, desmoduladores, e na implementação de vários tipos de esquemas de modulação, inclusive PSK, FSK, e MSK. O DDS pode ser descrito como um sintetizador de frequência controlado de formas de onda sinusoidais. Este é muito utilizado para geração digital de sinusoides de valores complexos ou reais, baseando-se no princípio de que uma forma de onda analógica pode ser gerada usando as suas amostras. A ideia base do DDS consiste em empregar um esquema de endereçamento com LUTs, onde estes armazenam os valores de um ciclo único de uma forma de onda sinusoidal. Durante o processo de geração, é possível alterar as características da fase, frequência e amplitude do sinal de saída sintetizado [17, 18, 19, 20].

Comparando a estrutura DDS com um sintetizador PLL, o tipo mais usado durante os últimos anos, a grande vantagem do DDS é não necessitar de um processo de sintonização/ajuste. Após estarem montados e ligados, os circuitos DDS apresentam o sinal de saída desejado, com uma performance constante e precisão digital. Os circuitos PLL precisam de alguns ajustes após o passo de construção e inicialização, obtidos com um filtro de *loop*. Esta característica do DDS assegura uma construção com elevada repetibilidade, precisão e performance, sendo estes factores muitos importantes na produção de componentes, subsistemas e equipamentos.

2.4.4 Revisão de técnicas de simplificação da DFT

A DFT é sem dúvida uma das transformadas discretas mais importantes, tendo uma grande importância na área do processamento digital de sinal. Considere uma sequência discreta $x[n]$ de comprimento N , sendo a sua DFT, $X[k]$, definida da seguinte forma:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi nk/N}, k = 0, 1, \dots, N-1 \quad (2.6)$$

O sinal $x[n]$ é recuperado fazendo o processo inverso, sendo a IDFT definida como:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j2\pi nk/N}, k = 0, 1, \dots, N-1 \quad (2.7)$$

Matematicamente, a DFT converte uma sequência finita de amostras, $x[n]$, num conjunto de coeficientes resultantes da combinação finita de sinusoides complexas com a

sequência de entrada $x[n]$. Podendo assim ser dito que, no caso da DFT, é feita uma conversão das amostras do domínio do tempo para o domínio da frequência e o processo inverso no caso da IDFT. As frequências das sinusoides de saída são inteiros múltiplos de uma frequência fundamental, cujo período é o comprimento do intervalo entre amostras. Apesar da grande utilidade desta operação, o cálculo requerido é normalmente muito lento para ser praticável. Foi então que surgiu a FFT, que permite calcular o mesmo resultado de uma forma mais rápida.

As FFTs são grupos de algoritmos que aumentam significativamente a rapidez da computação da DFT. Este aumento da rapidez é conseguido através da redução do número de multiplicações e adições requeridas nos cálculos envolvidos. Alguns exemplos de algoritmos FFT são o Winograd, o Rader-Brenner, o Brunn, podendo ser consultados em [21, 22], sendo o mais popular o de Cooley-Tukey [23].

2.4.4.1 FFT *radix-2* e *radix-4*

Duas das FFTs mais conhecidas são a *radix-2* e a *radix-4*, a decimação destas pode ser feita no tempo ou na frequência. A operação mais elementar de simplificação da DFT é denominada *butterfly*, devido à sua forma distinta. A figura 2.17 ilustra as *butterflies* da FFT *radix-2* e *radix-4*. A *butterfly radix-2* é usada em sequências de comprimento múltiplo de 2, enquanto a *butterfly radix-4* é usada para sequências de comprimento múltiplo de 4. Cada *butterfly* processa o cálculo da DFT de tamanho n , onde n é o *radix*. Assim, no caso da *butterfly radix-2* é calculada uma DFT de 2 pontos, enquanto na *butterfly radix-4* é processada uma DFT de 4 pontos.

A *radix-2* de decimação no tempo (DIT - *Decimation In Time*) é o algoritmo FFT mais intuitivo, tendo sido o primeiro algoritmo apresentado por Cooley e Tukey em [23]. A FFT DIT é caracterizada pela divisão da DFT da sequência de dados no tempo em sequências DFT mais pequenas. No caso da decimação da *radix-2* ser DIT, a sequência de dados no tempo é recursivamente decimada em duas sequências menores de tamanho $N/2$.

Dada a sequência $x[n] = \{x[0], x[1], \dots, x[N-1]\}$, onde N é múltiplo de 2, a DFT de $x[n]$, $X[k]$, é obtida recorrendo a 2.6. Podendo o somatório da equação 2.6 ser separados em dois somatórios de comprimento $N/2$,

$$X[k] = \sum_{n=0}^{N/2-1} x[2n]e^{-j2\pi(2n)k/N} + \sum_{n=0}^{N/2-1} x[2n+1]e^{-j2\pi(2n+1)k/N} \quad (2.8)$$

Uma alternativa à DIT recursiva de entrada, é proceder à decimação da sequência de saída na frequência. Dando assim origem à *radix-2* com decimação na frequência (DIF - *Decimation In Frequency*), não sendo esta tão intuitiva como a DIT. Considere-

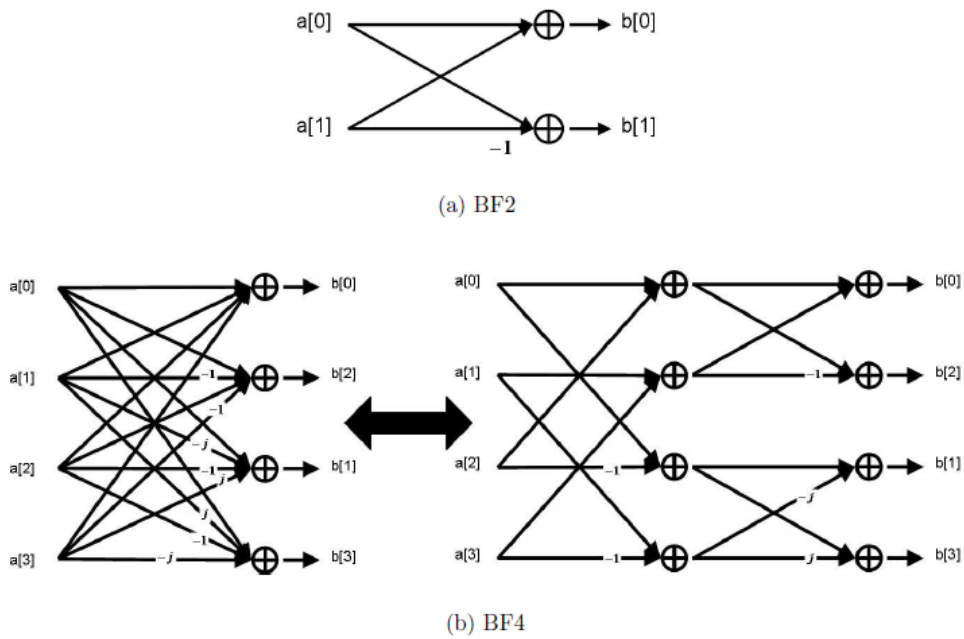


Figura 2.17: *Butterflies* radix-2 e radix-4 [3].

se novamente que o comprimento N da sequência $x[n]$ é múltiplo de 2. A sua DFT, $X[m]$, pode ser dividida em duas sequências de comprimento $N/2$, onde uma sequência contém todos os elementos pares, e a outra os elementos ímpares. Recorrendo novamente à equação 2.6, os cálculos dos elementos pares é o seguinte:

$$\begin{aligned}
 X[2k] &= \sum_{n=0}^{N-1} x[n] e^{-j2\pi n(2k)/N} \\
 &= \sum_{n=0}^{N/2-1} x[n] e^{-j2\pi n(2k)/N} + \sum_{n=N/2}^{N-1} x[n] e^{-j2\pi n(2k)/N} \\
 &= \sum_{n=0}^{N/2-1} x[n] e^{-j2\pi n(2k)/N} + \sum_{n=0}^{N/2-1} x[n + N/2] e^{-j2\pi (n+N/2)(2k)/N} \quad (2.9)
 \end{aligned}$$

Enquanto os elementos ímpares podem ser obtidos da seguinte forma:

$$\begin{aligned}
 X[2k+1] &= \sum_{n=0}^{N-1} x[n] e^{-j2\pi n(2k+1)/N} \\
 &= \sum_{n=0}^{N/2-1} x[n] e^{-j2\pi n(2k+1)/N} + \sum_{n=N/2}^{N-1} x[n] e^{-j2\pi n(2k+1)/N} \quad (2.10)
 \end{aligned}$$

A derivação do algoritmo *radix-4* é muito semelhante à do *radix-2*, pois em vez de se decimar a saída da DFT em dois somatórios, é feita uma decimação em quatro, passando a usar a *butterfly radix-4*, figura 2.17. Assim, o número total de *butterflies* requeridas é metade das necessárias na FFT *radix-2*. Contudo, tem a desvantagem de ser necessário calcular o dobro do número de adições complexas e de o tamanho dos dados de entrada ter de ser múltiplo de 4. A simplificação das equações 2.9 e 2.10, e as fórmulas das decimações da FFT *radix-4* podem ser consultadas em [3].

Devido a todas estas características, a FFT é um bloco fundamental nos sistemas DSP, com aplicações em *modems* digitais OFDM, ultrassons, RADAR e algoritmos de reconstrução de imagem. Apesar deste algoritmo ser facilmente entendido, as variações das arquiteturas de implementação e especificações são significantes. Devido à importância da operação FFT e das variações na sua implementação, foi incluído um bloco de processamento FFT na biblioteca *Xilinx Blockset* do *System Generator*. A *Xilinx* estima que a disponibilização deste bloco reduz o tempo de implementação da FFT de entre 3 a 6 meses, a um clique de um botão.

Capítulo 3

Sincronização em OFDM

Neste capítulo são revistas as características inerentes aos sistemas OFDM, mais concretamente referentes à precisão requerida na sincronização do recetor. Na secção 3.2 é feita uma descrição do processo de compensação do CFO. Na secção 3.3, é feita uma revisão de alguns dos principais métodos de estimação de início de trama e de CFO.

3.1 Limitações do OFDM

Na análise seguinte é assumido que o canal é não dispersivo e que o sinal transmitido, $s[n]$, é apenas afetado por ruído AWGN, $n[n]$. São consideradas duas incertezas no recetor, sendo estas a chegada de um símbolo OFDM e o desvio de frequência entre o oscilador do recetor e a frequência do sinal recebido, denominado CFO. Uma diferença na sincronização temporal provoca a rotação dos símbolos de dados, enquanto um desvio de frequência resulta num desvio de todas as sub-portadoras. Portanto, é necessário um bom mecanismo de sincronização, uma vez que as sub-portadoras só são ortogonais entre si, caso o recetor esteja a operar na mesma frequência que o sinal recebido. O sincronismo de receção OFDM engloba usualmente as seguintes fases: *Frame Synchronization*, *Symbol Timing Recovery* e *Frequency Offset Correction*.

Na 3.1 é possível observar-se as consequências provocadas pela presença de CFO no sistema. O CFO provoca um desvio nas frequências das sub-portadoras levando consequentemente a interferências consideráveis entre estas, sendo este efeito denominado por *Inter-Carrier Interference* (ICI). Portanto, ocorre um desvio do ponto máximo das sub-portadoras, ponto no qual as componentes das restantes sub-portadoras são nulas, para um ponto onde estas não são nulas. Assim, o símbolo resultante de uma sub-portadora para além de possuir informação dessa sub-portadora, contém também contribuições das restantes sub-portadoras. Esta perda da ortogonalidade entre sub-portadoras leva à degradação da performance BER [24]. O desvio de frequência pode ser causado tanto por

diferenças de oscilação entre os osciladores do recetor e transmissor, como pelo efeito de Doppler devido à existência de movimento entre o recetor e o transmissor. Os efeitos causados nos recetores OFDM devido a este desvio de fase foram investigados em vários trabalhos anteriores, como é o caso de [25, 26, 27, 28].

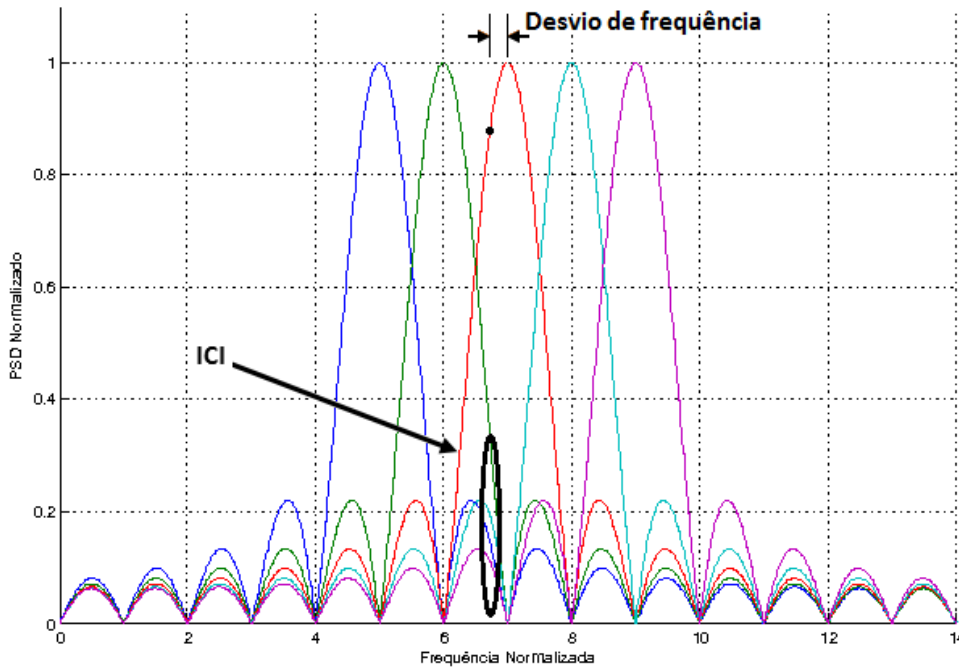


Figura 3.1: Consequências da presença de CFO.

O aparecimento de CFO deve-se ao facto de ser praticamente impossível criar cristais que oscilem a frequências precisamente iguais, para além da dependência destes em relação à temperatura, humidade, interferências eletromagnéticas e pressão. No caso do CFO causado pelo efeito de Doppler, este depende maioritariamente da direção em que o recetor se desloca em relação ao sinal recebido e da velocidade a que este se desloca. O desvio de frequência f_d introduzido pelo movimento do recetor é dado por:

$$f_d = \frac{v}{c} f_c \cos \varphi, \quad (3.1)$$

onde v é a velocidade do recetor, c é a velocidade da luz, f_c é a frequência da portadora e φ representa o ângulo de chegada. Por exemplo, no caso de um recetor móvel se deslocar na direção do transmissor a uma velocidade de 50km/h, este irá sofrer um desvio de 92.6 Hz no caso da frequência da portadora ser 2 GHz. Apesar do CFO aparentar ser pequeno quando comparado com a frequência da portadora, vimos em cima que pequenos CFOs podem causar degradações significativas. Uma vez que o CFO pode ser muito superior ao espaçamento entre sub-portadoras, é então dividido numa parte inteira e noutra fracionária. A parte inteira leva a um desvio das sub-portadoras tantas

posições quanto o valor deste número inteiro, não destruindo a ortogonalidade entre sub-portadoras. Enquanto a parte fracionária causa ICI, destruindo assim a ortogonalidade entre sub-portadoras. É de realçar que os sistemas OFDMA são ainda mais sensíveis a erros de frequência e tempo, realçando assim a importância de uma sincronização precisa. Sendo mais sensíveis no sentido em que para além de destruir a ortogonalidade e criar ICI, pode criar também interferência entre acessos, MAI.

3.2 Compensação do CFO

O processo de sincronização envolve então duas fases, sincronização temporal e na frequência. A primeira consiste na detecção da chegada de um símbolo OFDM, sendo a sensibilidade a desvios temporais mais crítica do que em sistemas de portadora única, como discutido em [29, 30]. Esta incerteza é tratada como um atraso na resposta a impulso do canal, $\delta[n - \theta_{ini}]$, onde θ_{ini} é um valor inteiro que representa o instante de chegada do símbolo. A fase seguinte é a sincronização na frequência, etapa na qual é estimado o CFO ocorrido, sendo este representado por f_o e o CFO normalizado ao espaçamento entre sub-portadoras por ε .

Assumindo que está presente no sistema um desvio de frequência f_o , este provoca um desvio f_o nas sub-portadoras dos símbolos OFDM. Este desvio leva a uma rotação da fase de $2\pi f_o t$ no domínio do tempo. Podendo ser escrito da seguinte forma:

$$r(t) = s(t)e^{j2\pi f_o t}, \quad (3.2)$$

onde $s(t)$ e $r(t)$ dizem respeito aos sinais no domínio do tempo transmitidos e recebidos, respectivamente. Tal como referido na figura 3.1, é possível observar-se o efeito do CFO, na qual há uma atenuação da amplitude do sinal desejado e a ortogonalidade entre as sub-portadoras é destruída. Assim, de forma a evitar ICI e a consequente degradação do SNR, o CFO residual tem de ser suficientemente pequeno. O CFO pode ser modulado como uma distorção complexa multiplicativa dos dados recebidos no domínio do tempo, $e^{\frac{j2\pi\varepsilon n}{N_c}}$, em que N_c representa o número de sub-portadoras e n é o índice do tempo discreto. Estas duas incertezas e o ruído AWGN resultam no seguinte sinal recebido:

$$r[n] = s[n - \theta_{ini}]e^{j2\pi\varepsilon n/N_c} + n[n] \quad (3.3)$$

Considerando que o recetor já fez a sincronização temporal, a equação 3.3 toma a seguinte forma:

$$r[n] = s[n]e^{j2\pi\epsilon n/N_c} + n[n] \quad (3.4)$$

Quanto à característica da ortogonalidade entre sub-portadoras, implica que quaisquer duas sub-portadoras estejam separadas exatamente por um número inteiro de ciclos. Como estas estão separadas por um múltiplo de $1/T_s$, em que T_s é a duração do símbolo OFDM, esta duração relaciona-se com o espaçamento entre sub-portadoras da seguinte forma: $\Delta f = \frac{1}{T_s}$. O espaçamento entre sub-portadoras pode ainda ser expressado das seguintes formas:

$$\Delta f = \frac{f_s}{N_c} = \frac{1}{N_c t_s}, \quad (3.5)$$

onde f_s e t_s representam a largura de banda e o intervalo temporal entre amostras, respectivamente. Posto isto, o deslocamento f_o que ocorre é igual a $\epsilon\Delta f$ Hz. Assim, um deslocamento normalizado de $\epsilon = 0.1$ representa 10% do deslocamento relativo à largura de banda de cada sub-portadora. Podendo então concluir, que quanto maior for o número de sub-portadoras maior será a sensibilidade ao desvio da frequência central da portadora.

Tal como foi mencionado o CFO pode ser modulado como $e^{\frac{j2\pi\epsilon n}{N_c}}$. A compensação do CFO é conseguida multiplicando as amostras no domínio do tempo por $e^{\frac{-j2\pi\epsilon n}{N_c}}$. Em que n varia entre 0 e $N_c - 1$ no caso de um símbolo OFDM. Após a compensação do CFO, o sinal recebido, equação 3.4, toma a seguinte forma:

$$\begin{aligned} r[n] &= (s[n]e^{j2\pi\epsilon n/N_c} + n[n]) e^{-j2\pi\epsilon n/N_c} \\ &= s[n] + n[n]e^{-j2\pi\epsilon n/N_c} \\ &= s[n] + n[n] \end{aligned} \quad (3.6)$$

É então necessário proceder a uma estimação do CFO normalizado ao espaçamento entre sub-portadoras, utilizando-o para compensar cada amostra pelo respectivo desvio de fase que esta sofreu. Entrando em maior detalhe, a primeira amostra não sofrerá qualquer efeito, pois é amostrada no instante zero, a amostra seguinte sofre um desvio de $2\pi\epsilon$, e a amostra três um desvio de $2\pi(2\epsilon)$ e assim sucessivamente. Na próxima secção serão apresentados alguns dos algoritmos de estimação tanto do CFO como do início da trama.

3.3 Algoritmos de Sincronização

Antes da estimação do CFO é necessário fazer uma estimação temporal, ou seja, detetar o início da trama. Esta sincronização tem como objetivo posicionar corretamente a janela da FFT com as amostras dos símbolos OFDM no domínio do tempo. Uma vez que o CFO é normalmente desconhecido nesta fase, é desejável que o esquema de recuperação temporal seja robusto contra grandes desvios de frequência. Um dos primeiros algoritmos de sincronização temporal para transmissões OFDM foi proposto por Nogami e Nagashima [31]. Este baseia-se na transmissão de um símbolo nulo, onde a queda de energia é usada para revelar a chegada de uma nova trama. Para além deste método fornecer estimações muito imprecisas, não é adequado para transmissões *burst*, uma vez que não se consegue distinguir entre a sequência nula e o período entre *bursts*.

Uma solução comum para superar este problema é o uso de sequências compostas por partes repetidas no domínio do tempo. Tirando partido da correlação existente entre as partes iguais, é possível fazer-se uma estimação com elevada robustez do início da trama. Este princípio foi inicialmente empregado por Schimdl e Cox (S&C) em [32], onde a sequência de referência é composta por dois símbolos OFDM idênticos. O método proposto por Van de Beek em [1], recorre à correlação existente entre as amostras do CP e as amostras das quais este foi copiado, para fazer a estimação temporal. É de realçar que todos os métodos de sincronização em OFDM têm limitações. Podendo estas ser ao nível da gama de estimação, da robustez na capacidade de detecção ou do *overhead* extra requerido.

3.3.1 Algoritmos baseados em símbolos *training* ou pilotos

Os algoritmos [32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42] baseiam-se na transmissão de sequências *training* no início da trama para estimar os parâmetros de sincronização. Tal como referido, estas sequências são normalmente compostas por partes repetidas, mantendo-se idênticas após passarem pelo canal excepto por um desvio de fase produzido pelo erro de frequência. A estimação é então feita medindo a diferença de fase ocorrida na sequência de referência. Devido aos bons resultados conseguidos, as sequências periódicas no domínio do tempo têm sido adotadas por vários *standards* [43, 44, 45], estas são usadas para estimação de início de trama e do CFO.

Dois dos algoritmos mais conhecidos de estimação de CFO recorrendo a símbolos *training* foram propostos por P. Moose [33] e por T.M. Schimdl & D.C. Cox (S&C)[32]. Moose propôs uma sequência de referência composta por dois símbolos OFDM consecutivos e idênticos. A diferença entre as amostras do primeiro e do segundo símbolo será um desvio de fase constante, sendo este estimado no domínio da frequência à saída da FFT.

É de realçar que este algoritmo requer o conhecimento do ponto de início do símbolo OFDM, ou seja, este método apenas pode ser utilizado para estimar o CFO. Quanto à gama de estimação deste estimador, esta está limitada a $[-\frac{\Delta f}{2}, \frac{\Delta f}{2}]$.

O algoritmo proposto por S&C [32] também usa uma sequência de referência composta por dois símbolos OFDM, figura 3.2. Este tem como vantagens sobre o método de Moose [33], o alargamento da gama de aquisição, para $[\Delta f, \Delta f]$, e o facto de permitir a estimação do início da trama. O primeiro símbolo OFDM, composto por duas partes idênticas no domínio do tempo, é usado para detecção do início de trama e estimação parcial do CFO. Enquanto o segundo símbolo, constituído por uma sequência PN1 nas sub-portadoras pares e uma PN2 nas ímpares, determina o CFO restante. Outro algoritmo semelhante ao S&C é o método proposto por Morelli e Mengali (M&M) em [35].



Figura 3.2: Constituição dos símbolos de referência propostos por Schimdl & Cox para estimação do início de trama e CFO.

O método proposto por C. Ribeiro em [2], permite a estimação do canal e do CFO no domínio do tempo, explorando a estrutura dos símbolos OFDM. Uma vez que o formato de transmissão é normalizado, o algoritmo tira partido do espaçamento entre símbolos piloto, no domínio da frequência, ser regular. A estimação do CFO baseia-se apenas nas componentes de dados presentes nos símbolos OFDM. O algoritmo consiste na utilização de um CFO candidato aplicado ao sinal de entrada, juntamente com o equivalente no domínio do tempo de um filtro multi-banda no domínio da frequência, que seleciona as sub-portadoras dos pilotos [46]. Este filtro é utilizado para fazer a soma dos desvios de fase das amostras na mesma relação de posição em todos os N_f segmentos de N_t amostras. Relembrando, que N_f é a distância entre pilotos no domínio da frequência e $N_t = \frac{N_c}{N_f}$. A função custo definida em [2], se dentro da gama de aquisição, revela o verdadeiro CFO quando o resultado desta for mínimo. A gama de aquisição deste estimador é $[-\frac{N_f \Delta f}{2}, \frac{N_f \Delta f}{2}]$. Uma vez que este foi um dos algoritmos implementados nesta dissertação para estimação do CFO, este será explicado com maior detalhe na secção seguinte.

3.3.1.1 Revisão do algoritmo proposto por C. Ribeiro em [2]

Este método, proposto por C. Ribeiro, permite a estimação dos vários canais e CFOs das respectivas antenas constituintes dos sistemas OFDM MIMO. A revisão feita concentrar-se-à apenas na componente de estimação do CFO para um sistema OFDM de uma só antena, uma vez que é o tema central desta tese. O artigo [47] faz a análise do método

para sistemas OFDM SISO. Como foi referido anteriormente, este algoritmo tira partido do facto dos símbolos piloto no domínio da frequência estarem regularmente espalhados a cada N_f subportadoras. Portanto, cada elemento dos símbolos recebidos no domínio do tempo, transportando dados e pilotos, e após a remoção do CP pode ser expressado da seguinte forma:

$$\tilde{r}[n] = \theta_{ini} \exp(jw_o n) \sum_{l=0}^{L-1} \tilde{d}[n-l] + \tilde{p}[n-l] + \tilde{n}[n], \quad (3.7)$$

onde w_o é o CFO angular, $\tilde{d}[n]$, $\tilde{p}[n]$ e $\tilde{n}[n]$ dizem respeito às componentes dos dados, pilotos e ruído no domínio do tempo, respectivamente. Recorrendo às Equações 2.2 e 2.3 e considerando o canal, a expressão anterior toma a seguinte forma:

$$\begin{aligned} \tilde{r}[n] &= \theta_{ini} N_c^{-1/2} \exp(jw_o n) \sum_{l=0}^{L-1} \sum_{k=0}^{N_c-1} \tilde{h}[l] d[k] \exp\left(j \frac{2\pi}{N_c} k(n-l)\right) + \tilde{n}[n] \\ &+ \theta_{ini} N_c^{-1/2} N_f^{-1} \exp(jw_o n) \sum_{l=0}^{L-1} \sum_{k=0}^{N_c-1} \tilde{h}[l] \exp\left(j \frac{2\pi}{N_c} k_{ini}(n-l)\right) \delta[n-l-mN_t] \\ &= \tilde{r}_d[n] + \tilde{r}_p[n] + \tilde{n}[n], \end{aligned} \quad (3.8)$$

onde \tilde{r}_d e \tilde{r}_p contêm as componentes dependentes dos dados e dos pilotos em \tilde{r} , respectivamente. Expandindo o vector \tilde{r}_p ,

$$\tilde{r}_p[n] = \theta_{ini} N_c^{-1/2} N_f^{-1} \exp(jw_o n) \sum_{m=0}^{N_f-1} \left(j \frac{2\pi}{N_c} k_{ini}(mN_t) \right) \tilde{h}[n-mN_t] \quad (3.9)$$

Tornando-se claro que é composto por N_f réplicas do CIR desviadas na frequência, com uma separação de N_t amostras entre elas.

O princípio de funcionamento do algoritmo consiste na procura, dentro da gama de aquisição, do valor mínimo da função custo. Uma das características principais deste algoritmo é a utilização de um CFO candidato, \hat{w} , aplicado ao sinal de entrada. Para além da utilização de um CFO candidato, é também feita a soma de todos os desvios de fase das amostras no TD das posições dos pilotos no FD [46]. Entrando em detalhe, são somadas as amostras na mesma relação de posição em todos os N_f segmentos de N_t amostras. Relembrando que N_f é a distância entre pilotos no domínio da frequência e $N_t = \frac{N_c}{N_f}$.

A *core* da função de custo é definida como:

$$\begin{aligned}\tilde{g} &= T \text{diag} \left(\left[1 \quad \exp \left(-j \frac{2\pi}{N_c} k_{ini} \right) \quad \dots \quad \exp \left(-j \frac{2\pi}{N_c} k_{ini} (N_c - 1) \right) \right] \right) C_{N_c}^H(\hat{w}) \tilde{r} \\ &= \tilde{g}_d + \tilde{g}_p + \tilde{v},\end{aligned}\quad (3.10)$$

onde a matriz T , de dimensões $N_t \times N_c$, é definida como $T = [I_{N_t} \dots I_{N_t}]$. A variável $\hat{w} = 2\pi\hat{\varepsilon}$ identifica o CFO angular normalizado candidato. Portanto, a componente $C_{N_c}(\hat{w})$ é definida da seguinte forma:

$$C_{N_c}(\hat{w}) = \text{diag} \left(\left[1 \quad \exp(j2\pi\hat{\varepsilon}/N_c) \quad \dots \quad \exp(j2\pi\hat{\varepsilon}(N_c - 1)/N_c) \right] \right) \quad (3.11)$$

Os vectores de N_t colunas \tilde{g}_d e \tilde{g}_p contêm as componentes dependentes dos dados e dos pilotos, respectivamente, e \tilde{v} o vector resultante do ruído. Os elementos de \tilde{g}_p podem ser expressados como:

$$\begin{aligned}\tilde{g}_p[n] &= \sum_{m=0}^{N_f-1} \exp \left(-j2\pi k_{ini} \frac{n + mN_t}{N_c} \right) \exp(-j\hat{w}(n + mN_t)) \tilde{r}_p[n + mN_t] \\ &= \theta_{ini} N_c^{-1/2} N_f^{-1} \exp(j(w_o - \hat{w})n) \sum_{m=0}^{N_f-1} \sum_{q=0}^{N_f-1} \exp \left(-j \frac{2\pi}{N_c} k_{ini} (n + (m - q)N_t) \right) \\ &\quad \exp(j(w_o - \hat{w})mN_t) \tilde{h}[n + (m - q)N_t]\end{aligned}\quad (3.12)$$

Podendo a equação ser simplificada...

$$\tilde{g}_p[n] = \theta_{ini} N_c^{-1/2} N_f^{-1} \exp(j(w_o - \hat{w})n) \exp \left(-j \frac{2\pi}{N_c} k_{ini} n \right) \tilde{h}[n] \sum_{m=0}^{N_f-1} \exp(j(w_o - \hat{w})mN_t) \quad (3.13)$$

Mostrando claramente que as amostras dependes dos pilotos são limitadas aos conjunto de L_p amostras, onde existe energia do CIR. As amostras restantes dependerão apenas dos dados transmitidos e do ruído.

Os elementos de g_d podem ser expressados como:

$$\begin{aligned}
\tilde{g}_d[n] &= \sum_{m=0}^{N_f-1} \exp\left(-j2\pi k_{ini} \frac{n+mN_t}{N_c}\right) \exp\left(-j\hat{w}(n+mN_t)\right) \tilde{r}_d[n+mN_t] \\
&= \theta_{ini} N_c^{-1/2} \sum_{m=0}^{N_f-1} \left(-j2\pi k_{ini} \frac{n+mN_t}{N_c}\right) \exp\left(j(w_o - \hat{w})(n+mN_t)\right) \\
&\quad \sum_{l=0}^{L-1} \sum_{k=0}^{N_c-1} \tilde{h}[l] d[k] \exp\left(j\frac{2\pi}{N_c} k(n-l+mN_t)\right) \\
&= \theta_{ini} N_c^{-1/2} \exp\left(j(w_o - \hat{w})n\right) \exp\left(-j\frac{2\pi}{N_c} k_{ini} \frac{n}{N_c}\right) \tilde{h}[l] \psi, \tag{3.14}
\end{aligned}$$

onde

$$\begin{aligned}
\psi &= \sum_{k=0}^{N_c-1} d[k] \exp\left(j\frac{2\pi}{N_c} k(n-l)\right) \sum_{m=0}^{N_f-1} \exp\left(j(w_o - \hat{w})mN_t\right) \exp\left(j\frac{2\pi}{N_c} (k - k_{ini})m\right) \\
&= \sum_{k=0}^{N_c-1} d[k] \exp\left(j\frac{2\pi}{N_c} k(n-l)\right) \sum_{m=0}^{N_f-1} \exp\left(j\frac{2\pi m}{N_c} (f_o - \hat{f})N_c \Delta t + (k - k_{ini})\right), \tag{3.15}
\end{aligned}$$

onde \hat{f} é o desvio de frequência inicial candidato. Em [2], após um tratamento cuidado da equação 3.15, foi concluído que o fator ψ é zero quando $(f_o - \hat{f}) = mN_f \Delta f$. Sendo de realçar, que a solução apresenta uma periodicidade $N_f \Delta f$ e inclui a condição em que o CFO é completamente eliminado, $(f_o - \hat{f} = 0)$. Podendo concluir que ψ tem valores mínimos espalhados $N_f \Delta f$ Hz, com $(N_f - 1)$ valores de magnitude máximos entre eles, separados de Δf Hz.

Se definirmos j como o vector de colunas $(N_t - L_p)$ que contém as amostras de \tilde{g} sem energia do CIR, ou seja, apenas contém a energia das componentes dos dados e o ruído. A função de custo, $J(\hat{w})$, é então definida como a energia de j , sendo definida da seguinte forma:

$$J(\hat{w}) = j^H j \tag{3.16}$$

Tal como tinha sido já referido na secção 2.2.3, o uso desta função de custo dentro da gama de aquisição, revela o verdadeiro CFO quando o resultado desta for mínimo. Os elementos em j podem ser obtidos a partir de um símbolo OFDM ou um conjunto de símbolos, no caso de ser necessária uma estimação com maior precisão. A gama de aquisição da função de custo é $\left[-\frac{N_f \Delta f}{2}, \frac{N_f \Delta f}{2}\right]$, sendo esta retirada a partir da análise do fator ψ . Portanto, a estimação do CFO pode ser definida como:

$$\hat{w}_o = \arg \{ \min J(\hat{w}) \}, \quad (3.17)$$

onde \hat{w}_o é o valor do CFO estimado. A secção 4.3 descreve toda a implementação deste algoritmo na plataforma *Xilinx System Generator*.

3.3.1.2 Sequências *training*: *Zadoff-Chu*

O processo de sincronização em canais HF requer que autocorrelação das sequências de sincronização seja perfeita. Nos anos recentes, as sequências CAZAC têm ganho grande destaque e são usadas em diversas áreas devido às suas propriedades de autocorrelação periódica perfeita [48]. As sequências *Zadoff-Chu* são classificadas como sequências CAZAC, tendo estas ganho uma grande atenção quando comparadas com outras sequências baseadas em pseudo-ruído.

A sequência CAZAC proposta foi descrita por Frank e Zadoff em [49], baseado neste trabalho Chu propôs um conjunto de sequências CAZAC sem restrições no seu comprimento, resultando na sequência *Zadoff-Chu* [50]. As sequências *Zadoff-Chu* têm sido propostas para sincronização em sistemas OFDM, tendo sido recentemente introduzidas nos standards LTE e LTE-Advanced, em ambos para sincronização no *uplink* e *downlink*. As sequências *Zadoff-Chu* são definidas como amostras de uma função complexa, como é mostrado na equação 3.18.

$$ZC_{N_{ZC},M}[k] = \begin{cases} e^{-j\frac{M\pi k^2}{N_{ZC}}} & \text{para } N_{ZC} \text{ um inteiro par} \\ e^{-j\frac{M\pi k(k+1)}{N_{ZC}}} & \text{para } N_{ZC} \text{ um inteiro impar,} \end{cases} \quad (3.18)$$

onde $ZC_{N_{ZC},M}[k]$ representa a sequência de comprimento N_{ZC} e M identifica a família, em que N_{ZC} tem de ser primo relativamente a M . Chu provou também, que caso a sequência sofra um certo desvio de fase na forma $e^{\frac{j2\pi qk}{N}}$, este não afeta a sua propriedade de correlação. Onde q , k e N são inteiros, sendo k primo relativamente a L .

Para além da propriedade de autocorrelação perfeita, as sequências *Zadoff-Chu* possuem outras características importantes, que atraem ainda mais o uso deste tipo de sequências. Tal como é observável na equação 3.18, esta possui uma amplitude constante, limitando assim o PAR. Esta é uma vantagem na implementação do amplificador de potência e na limitação da interferência a outros utilizadores. Outra característica importante, é o resultado da DFT de uma sequência *Zadoff-Chu* ser uma sequência *Zadoff-Chu*. Esta propriedade é importante na criação deste tipo de sequências, mas mais importante ainda, permite que a correlação possa ser feita tanto no domínio da frequência como no domínio do tempo.

Em conclusão, a complexidade requerida para estimação de CFO usando sequências de

sincronização é já bastante reduzida. Relembrando que o uso destas sequências acarreta um *overhead* extra na transmissão, reduzindo o débito dos dados efetivo. Na secção seguinte serão abordados os métodos *Non-Data Aided*, os quais não requerem *overhead* extra.

3.3.2 Algoritmos *Non-Data Aided*

Alguns dos algoritmos *Non-Data Aided* mais populares podem ser revistos em [51, 1, 52, 53], estes tiram partido da estrutura do símbolo OFDM. O algoritmo de van de Beek [1] é um dos mais conhecidos, baseando-se na redundância introduzida pelo CP para proceder à estimação. De seguida será feita uma revisão deste algoritmo, que foi utilizado para fazer a sincronização do recetor implementado na cadeia de transmissão OFDM. Este algoritmo foi utilizado para estimação do CFO e início de trama no recetor.

Sendo r_n a amostra recebida no instante n , a estimação ML do início de trama, θ_{ini} , e do CFO, ε , é encontrada maximizando a sua função de verosimilhança. Uma vez que esta função definida como um logaritmo da função densidade de probabilidade, a estimação ML conjunta é obtida da seguinte forma:

$$\hat{\theta}_{ML} = \arg \max \{ |\gamma(\theta_{ini})| - \rho \Phi(\theta_{ini}) \} \quad (3.19)$$

$$\hat{\varepsilon}_{ML} = -\frac{1}{2\pi} \angle \gamma(\hat{\theta}_{ML}), \quad (3.20)$$

onde \angle denota o argumento de um número complexo e

$$\gamma(m) = \sum_{n=m}^{m+L-1} r_n^* r_{n+N_c} \quad (3.21)$$

$$\Phi(m) = \frac{1}{2} \sum_{n=m}^{m+L-1} |r_n|^2 + |r_{n+N_c}|^2 \quad (3.22)$$

O primeiro termo da equação 3.19, $\gamma(m)$, que depende do CFO, é o somatório de L multiplicações de amostras com um distanciamento de N_c amostras entre si. O segundo termo, $\Phi(m)$, que não depende do CFO, é a energia presente nas L amostras do CP e nas amostras a partir das quais este foi copiado. A contribuição deste termo tem em conta o SNR introduzido pelo fator de ponderação ρ .

Na figura 3.3 é possível observar a estrutura do estimador ML de van de Beek, onde as estimações de θ_{ini} e ε são basicamente fornecidas pelo termo $\gamma(m)$. O módulo devolve a posição do início de trama, através do seu valor de pico, e a compensação deste pelo

termo da energia, $\Phi(m)$, resulta na estimação $\hat{\theta}_{ML}$. A equação 3.20 devolve o desvio de fase de amostras com um distanciamento de N_c amostras entre si, cujo resultado, nos instantes de tempo θ_{ini} , contém o valor da estimação do desvio de frequência, ε .

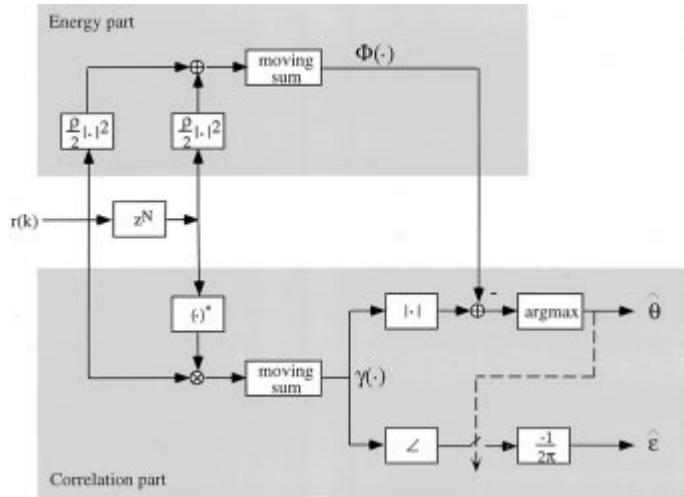


Figura 3.3: Estrutura do estimador ML de van de Beek [1].

Relembrando a equação 3.4, $r[n] = s[n]e^{j2\pi\varepsilon n/N_c} + n[n]$, e não considerando o ruído, a equação 3.21 resulta em:

$$\begin{aligned}
 \gamma(m) &= \sum_{n=m}^{m+L-1} s[n]e^{-j2\pi\varepsilon n/N_c} s[n+N_c]e^{j2\pi\varepsilon(n+N_c)/N_c} \\
 &= \sum_{n=m}^{m+L-1} s[n]s[n+N_c] \sum_{n=m}^{m+L-1} e^{-j2\pi\varepsilon n/N_c} e^{j2\pi\varepsilon(n+N_c)/N_c} \\
 &= \sum_{n=m}^{m+L-1} s[n]s[n+N_c] L e^{-j2\pi\varepsilon n/N_c + j2\pi\varepsilon(n+N_c)/N_c} \\
 &= L e^{j2\pi\varepsilon N_c/N_c} \sum_{n=m}^{m+L-1} s[n]s[n+N_c]
 \end{aligned} \tag{3.23}$$

Recorrendo agora à equação 3.20, o resultado da estimação do desvio de frequência normalizado ao espaçamento entre sub-portadoras é:

$$\hat{\varepsilon}_{ML} = -\frac{1}{2\pi} (2\pi\varepsilon N_c/N_c) = -\varepsilon \tag{3.24}$$

Na secção seguinte, serão então abordados os algoritmos *blind*, os quais não requerem o uso de seqüências de referência nem de pilotos. O facto de não introduzirem um *overhead* extra leva a que a complexidade computacional associada a este tipo de algoritmos seja

elevada.

3.3.3 Algoritmos de estimação blind

A maior parte das técnicas de estimação de CFO em OFDM recai sobre a transmissão periódica de símbolos de referência, causando inevitavelmente uma redução na eficiência da largura de banda. Em contrapartida, os algoritmos denominados *blind* são algoritmos que fazem a estimação do CFO apenas com base em propriedades estatísticas do sinal recebido, sem conhecimento explícito do sinal transmitido. Estas estimações são feitas tirando partido de certas propriedades presentes nos símbolos OFDM, como o CP no domínio do tempo e as sub-portadoras nulas no domínio da frequência. Algumas destas técnicas de estimação podem ser consultadas em [54, 55, 56, 57, 58, 59, 60, 61].

A técnica proposta por Liu & Tureli em [62] faz uma estimação de alta precisão do CFO, tirando partido da ortogonalidade existente entre as sub-portadoras, devido ao sinal OFDM manter uma estrutura algébrica dependente do desvio de fase. A estimação é feita no domínio da frequência e tem por base o facto das sub-portadoras no início e no fim do símbolo OFDM serem nulas.

Apesar dos bons resultados apresentados, este tipo de algoritmos não são muito usados devido à elevada complexidade associada. Portanto, os métodos apresentados anteriormente têm uma maior empregabilidade mesmo requerendo um *overhead* extra. Apesar disto, é de realçar a importância de encontrar métodos que reduzam a complexidade dos algoritmos *blind* sem causar grandes perdas de desempenho.

Capítulo 4

Sistema OFDM Assíncrono

Neste capítulo é feita uma comparação entre os sistemas que processam os seus dados de forma assíncrona, princípio de funcionamento do sistema disponibilizado para a realização desta dissertação, e os que os processam de forma síncrona. É também explicada a implementação prática, na plataforma *System Generator*, dos algoritmos de van de Beek [1] e C. Ribeiro [2] introduzidos na secção 3.3.

4.1 Processamento assíncrono

Num sistema síncrono, o fluxo de dados funciona com ritmo constante de bloco para bloco, necessitando por isso de um grupo de trabalho perfeitamente sincronizado, figura 4.1. No caso da adição de novos elementos é necessário proceder a ajustes, uma vez que o funcionamento geral do sistema é afetado. É portanto perceptível que, à medida que o projeto cresce torna-se rapidamente incontrolável, muitas vezes apenas controlável pela pessoa que o desenvolveu de raiz. Este tipo de funcionamento não tem qualquer tipo de controlo e em momento algum um bloco pode parar o seu processamento. É de realçar que o ritmo de funcionamento do sistema é imposto pelo ADC/DAC, sendo este o único elemento que necessita de funcionar a um ritmo constante.

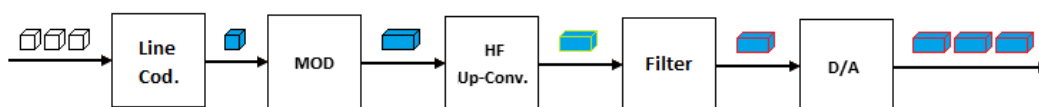


Figura 4.1: Sistemas síncronos: o fluxo de dados funciona com ritmo constante.

No caso de um sistema assíncrono, o fluxo de dados é processado em *bursts* de bloco para bloco, sendo o ADC/DAC o único que funciona com ritmo constante, figura 4.2. O assincronismo é conseguido recorrendo à utilização de *buffers* em cada bloco, utilizados

para armazenar os dados que aguardam processamento. Com esta arquitetura de funcionamento não é necessário sincronismo entre os vários blocos, bastando garantir que os *buffers* nunca encham nem fiquem vazios. Portanto, adicionar ou remover blocos passa a ser uma tarefa trivial não tendo impacto no funcionamento do sistema. Cada bloco faz o controlo do seu processamento, ao seu ritmo, tendo apenas que respeitar o enchimento do *buffer*.

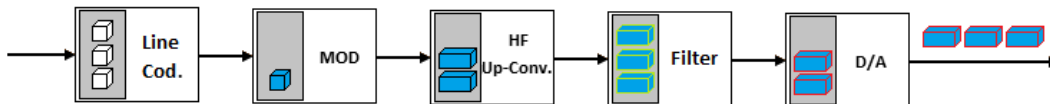


Figura 4.2: Sistemas assíncronos: o único bloco que funciona com um ritmo constante é o ADC/DAC.

Esta arquitetura de funcionamento assíncrono permite a concepção e teste individual de cada bloco, permitindo a reutilização do *hardware* implementado, sem a necessidade de fazer qualquer tipo de alteração no sistema ou no bloco. Contudo, há um acréscimo na complexidade de cada bloco, pela introdução das memórias FIFO, utilizadas como *buffers*, e de duas máquinas de estados: uma para controlo do nível de enchimento dos *buffers*, e outra para controlo do processamento de acordo com esse nível. Esta arquitetura tem também um atraso inicial associado ao enchimento dos *buffers* até aos níveis adequados, tomando de seguida um efeito de funcionamento cíclico.

4.1.1 Transmissor

A arquitetura dos blocos do transmissor é semelhante para todos os blocos, independentemente do seu processamento específico. A constituição geral dos módulos do transmissor é ilustrada na figura 4.3, onde é possível observar que os *buffers* isolam o processamento de cada bloco. A máquina de estados *Buffer_Ctrl* tem como função controlar o nível de enchimento das memórias, fornecendo esta informação à máquina de estados *Process_Ctrl* e ao bloco seguinte, através das saídas *En_Data* e *AE* (*Almost Empty*), respectivamente. Caso os *buffers* atinjam o limite superior, a saída *En_Data* é de-asserida, indicando ao *Process_Ctrl* que deve suspender o processamento deste bloco, para que não ocorram perdas de dados devido ao enchimento das memórias. No caso dos *buffers* atingirem os limites inferiores, a saída *AE* é asserida, indicando ao bloco seguinte que deve suspender o seu processamento de dados, para não esgotar os dados presentes no *buffer*. É espectável que a saída *AE* apenas seja asserida na inicialização do sistema, até que os *buffers* atinjam os seus níveis adequados. Caso isto ocorra noutros instantes, é um indício de que o bloco é demasiado lento, ou seja, não processa dados a um ritmo suficiente.

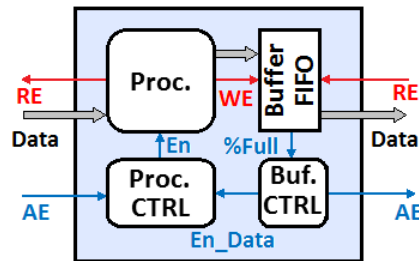


Figura 4.3: Sistemas assíncronos: Constituição geral dos módulos do transmissor.

A máquina de estados *Process_Ctrl* tem como função controlar o processamento específico efetuado pelo bloco. Este deve ser executado quando os níveis de enchimento dos *buffers* de saída se encontrarem dentro da gama de funcionamento, indicado pela entrada *En_Data*. Por outro lado, não deve ser executado quando os níveis de enchimento dos *buffers* do bloco anterior se encontrarem no limite inferior indicado pela entrada *AE*. Esta máquina de estados, para além destas condições obrigatórias, tem também condições dependentes do tipo de processamento.

Para além da entrada e saída *AE*, os blocos têm também entradas e saídas de dados de acordo com o tipo de processamento, podendo estas ser de bits ou símbolos. Possui também uma entrada *RE_{IN}*, usada pelo bloco seguinte para aceder aos dados armazenados nos *buffers*, e uma saída *RE_{OUT}* usada para aceder aos dados armazenados no bloco anterior, à medida que o processamento vai sendo efetuado.

4.1.2 Recetor

Quanto à arquitetura de funcionamento dos blocos do recetor, figura 4.4, esta é muito semelhante à do transmissor. Os blocos são igualmente constituídos pela parte de processamento, pelos *buffers* e pelas duas máquinas de estados. No caso dos blocos do recetor, os *buffers* são colocados à entrada de cada bloco, de forma a armazenar os dados à medida que estes chegam, sendo posteriormente processados. É o ADC que impõe o ritmo de escrita nas memórias de entrada do primeiro bloco do recetor. Enquanto na arquitetura do transmissor era usada a entrada/saída *AE*, no recetor é utilizada a entrada/saída *AF* (*Almost Full*). A saída *AF*, quando asserida, indica que os *buffers* estão perto dos limites superiores de enchimento, notificando o bloco anterior que deve suspender o seu processamento. Se o sistema funcionar corretamente, é espetável que as saídas *AF* nunca sejam asseridas. O asserimento de uma saída *AF* é um indício de que o bloco não está a processar os dados a uma velocidade suficientemente alta, o que fará com que os blocos anteriores parem de executar o seu processamento. Consequentemente, levará a perdas de dados uma vez que estes são escritos no primeiro bloco ao ritmo do ADC. A escrita nos *buffers* dos blocos do recetor é efetuada pelos blocos anteriores, utilizando a entrada/saída

WE_{IN}/WE_{OUT} . Quanto às máquinas de estados *Buffer_Ctrl* e *Process_Ctrl*, estas têm um princípio de funcionamento idêntico às usadas nos blocos do transmissor.

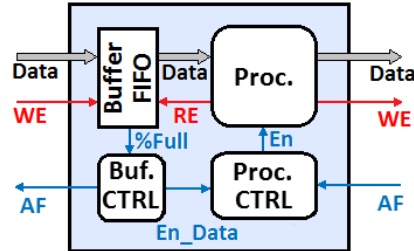


Figura 4.4: Sistemas assíncronos: Constituição geral dos módulos do recetor.

4.1.3 Constituição do Sistema

Existem já diversos sistemas OFDM baseados em SDR descritos na literatura. Por exemplo, em [63] é apresentado um modulador/desmodulador OFDM com duas opções de sincronização e duas técnicas de controlo de erros. O trabalho em [64] usa o GNU Radio para transferir os sinais OFDM com a modulação QPSK e BPSK para analisar a taxa de receção de pacotes, com o objetivo de avaliar a qualidade de serviço. Também baseado no GNU Radio, o sistema de comunicação em [63] é constituído por um modulador e desmodulador OFDM baseado nas camadas inferiores OSI do WiMAX. Implementações em FPGAs dos moduladores dos standards 802.11a e 802.16-2004 usando o *System Generator* podem ser encontrados em [65, 66]. Também baseado nesta plataforma, em [67] foram feitas as implementações dos moduladores dos standards 802.11a/g, suportando também OFDMA.

Na figura 4.5, encontra-se ilustrado o sistema OFDM disponibilizado para a realização desta tese. O transmissor é constituído por um modulador QPSK, com a função de converter os bits de dados em símbolos QPSK. O bloco *FD_Framing* tem a função de definir os símbolos OFDM no domínio da frequência, colocando os dados nas subportadoras centrais, deixando as restantes a zero. Estas subportadoras não são utilizadas para transmissão de dados, evitando assim *aliasing*. Para além disso, tem também a função de inserir os pilotos nas subportadoras e símbolos correspondentes, sendo os valores dos pilotos definidos pelo bloco *Pilot_Gen*. Após o símbolo estar composto no domínio da frequência, o bloco *IFFT_c/_CP* faz a conversão deste para o domínio do tempo, recorrendo ao módulo FFT disponibilizado na biblioteca *Xilinx Blockset*. A constituição das tramas temporais é definida pelo bloco *TD_Framing*, onde cada trama é constituída por um símbolo para efeitos de sincronização e quatro símbolos para transporte de dados. O símbolo de sincronismo, composto por uma sequência *Zadoff-Chu*, é definido no bloco *TD_Seq_Gen*. Por fim, o bloco *Slow_Sync_Output* emula o DAC, retirando amostras a

uma velocidade constante do *buffer* de saída do bloco anterior.

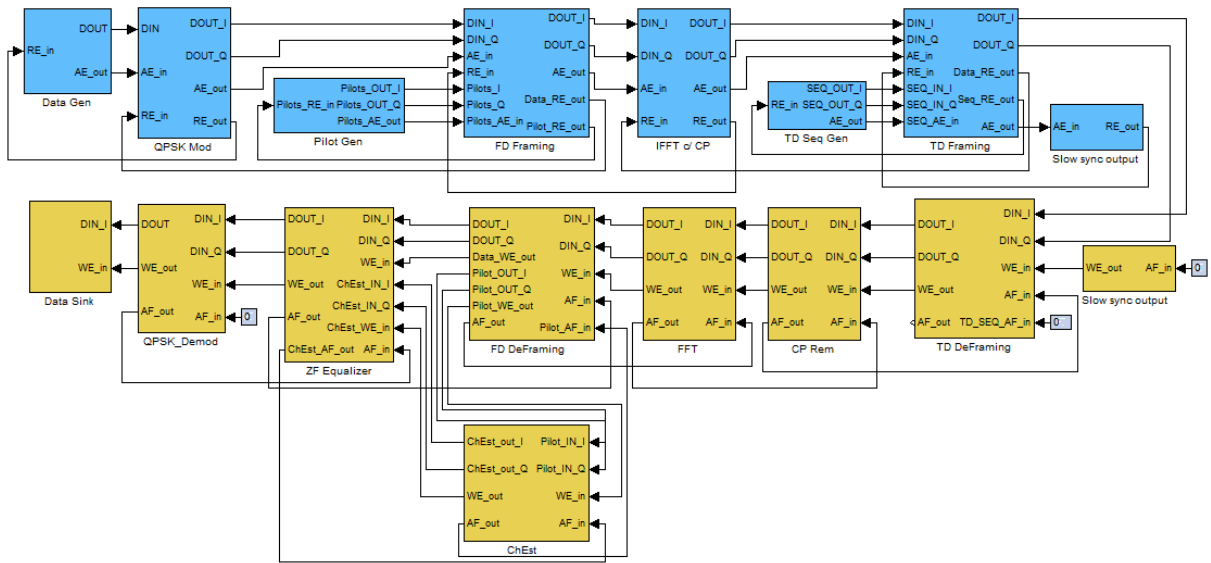


Figura 4.5: Implementação do sistema OFDM desenvolvido na plataforma *Xilinx System Generator*

No caso do recetor, este é constituído por um bloco *TD_Framing*, que tem a função de fazer a sincronização temporal. Para além de detectar o início da trama, tem também de estimar o CFO e compensar o efeito deste. Este bloco foi então dividido em dois, o primeiro denominado *CFO_Estimation*, tem como função a estimação do CFO e do início da trama, utilizando essa estimação para se sincronizar temporalmente. O segundo bloco, denominado *CFO_Correction*, tem como função compensar o efeito do CFO, recebendo para isso a estimação efetuada pelo *CFO_Estimation*. Uma vez que este é o tema central desta tese, o funcionamento destes blocos será abordado com maior detalhe nas Secções 4.2, 4.3 e 4.4. Após a sincronização temporal e o CFO ter sido compensado, procede-se à remoção do CP no bloco *CP_Rem*. Realçando que a remoção do CP deve ser efetuada após a compensação do CFO, isto para que não ocorram saltos na compensação causados pela eliminação das amostras do CP entre símbolos. A figura 4.6 ilustra o que ocorreria caso o CP fosse removido antes da compensação. Como o *CFO_Correction* compensa o desvio amostra a amostra, no caso do CP ser descartando antes da compensação, os ângulos estimados para as amostras do CP seriam associados às amostras seguintes ao CP, provocando assim saltos na constelação dos símbolos QPSK. Contudo este efeito poderia ser compensado pelo equalizador do canal.

De seguida, os símbolos OFDM são convertidos para o domínio da frequência, recorrendo novamente ao módulo FFT disponibilizado na biblioteca *Xilinx Blockset*. O bloco *FD_DeFraming* faz a separação dos dados e dos pilotos, sendo os pilotos utilizados pelo bloco *ChEst* para estimação do canal, sendo esta estimação posteriormente utilizada pelo bloco *ZF_Equalizer* para compensação do efeito do canal. Por fim, o desmodulador QPSK

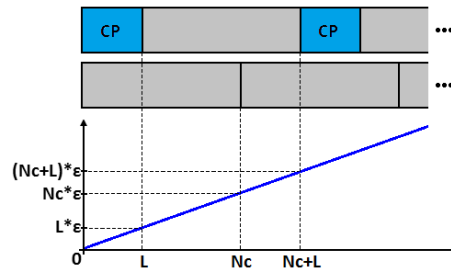


Figura 4.6: Consequências da remoção do CP antes da compensação do CFO.

volta a converter os símbolos QPSK em bits de dados. Na secção seguinte será explicado em detalhe todo o processo de estimação do início da trama e do CFO utilizando o algoritmo ML de van de Beek [1].

4.2 Implementação do algoritmo ML de van de Beek na plataforma *System Generator*

Para proceder à sincronização do recetor, foi implementado o algoritmo de van de Beek proposto em [1] utilizando também um símbolo *training*, composto por uma sequência *Zadoff-Chu*. A utilização conjunta do algoritmo de van de Beek e um símbolo de sincronismo já tinha sido implementado em [68], com uma estrutura síncrona de implementação. Tal como referido na secção 3.3, este algoritmo permite fazer a estimação do início da trama e do desvio de frequência entre o oscilador do recetor e a frequência do sinal recebido. A figura 4.7 ilustra a constituição do estimador implementada no *System Generator*, onde é possível observar que este mantém a arquitetura de funcionamento dos blocos do recetor, figura 4.4. Tal como em qualquer bloco do recetor, este é constituído por *buffers*, colocados à entrada do bloco, por duas máquinas de estados e pelo processamento específico. As máquinas de estados *RX_BUFFER_CTRL* e *CFO_Estimation_CTRL* têm como função controlar o nível de enchimento dos *buffers* e a execução do processamento, respectivamente. O processamento foi dividido em vários blocos: *Energy*, *Correlation* e *Frame_Detection*.

4.2.1 Energia e Correlação

Tal como referido na secção 3.3.2, o algoritmo de van de Beek tira partido da correlação existente entre o CP e as amostras das quais este é copiado. O algoritmo é então dividido em duas partes, numa é feito o cálculo da energia do CP e das amostras das quais este foi copiado e na outra, o cálculo da correlação existente entre estas amostras. Relembrando que o cálculo da energia e da correlação são obtidos de acordo com as equações 4.1 e

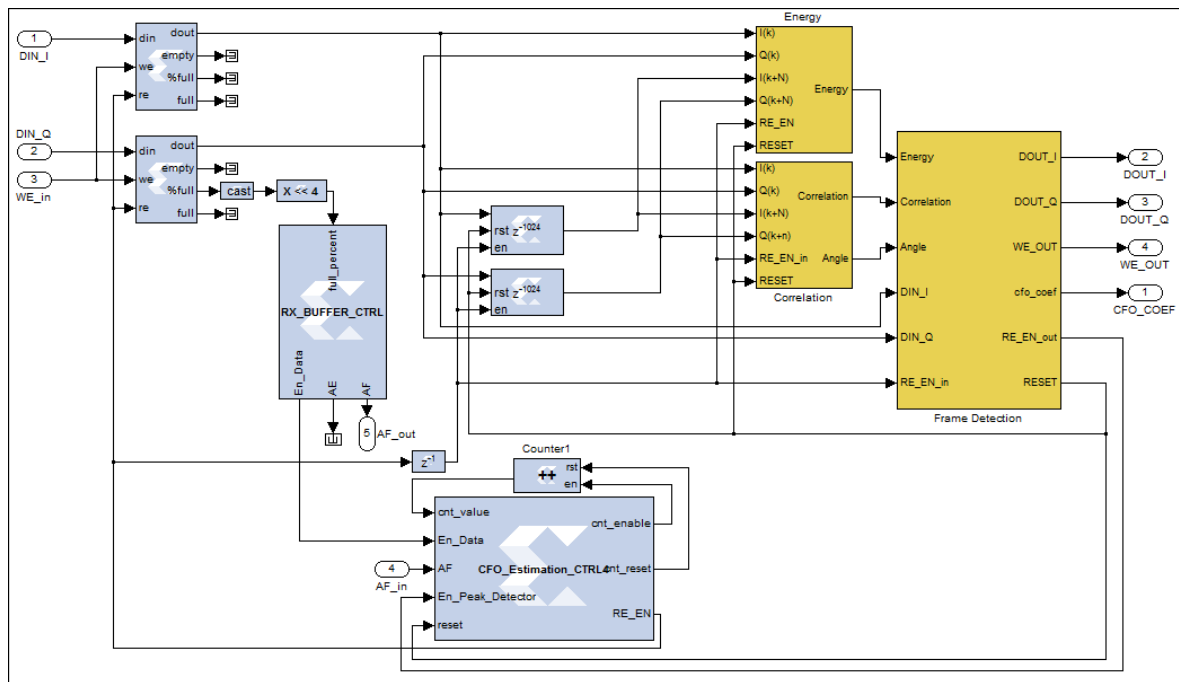


Figura 4.7: Implementação do algoritmo de estimação ML de van de Beek na plataforma *Xilinx System Generator*.

4.2, onde L é o tamanho do CP e N_c o número de subportadoras, sendo respectivamente 256 e 1024. Para efetuar o cálculo da energia são usados filtros IIR, como ilustrado no topo direito da figura 4.8. O filtro consiste na soma das últimas 256 amostras, uma vez que à medida que as amostras vão passando, as primeiras amostras vão sendo subtraídas. Obtém-se assim a energia de dois grupos de 256 amostras distanciados de 1024. A figura 4.9 apresenta resultado do cálculo da energia, onde é possível observar que o primeiro símbolo tem uma energia superior à dos símbolos de dados, correspondendo os picos máximos à última amostra da sequência de sincronismo. Isto ocorre devido à sequência de sincronismo ter sido transmitida com uma potência 6 dB acima da dos dados. Esta diferença teve como objetivo proporcionar a distinção da sequência de sincronismo, uma vez que esta é usada para sincronização do recetor.

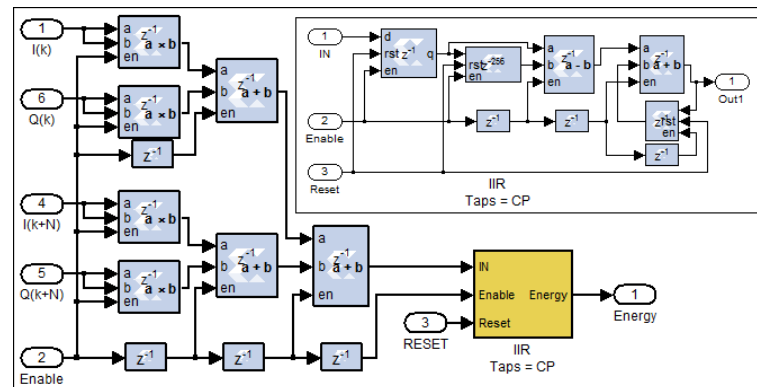


Figura 4.8: Implementação do cálculo da Energia na plataforma *Xilinx System Generator*.

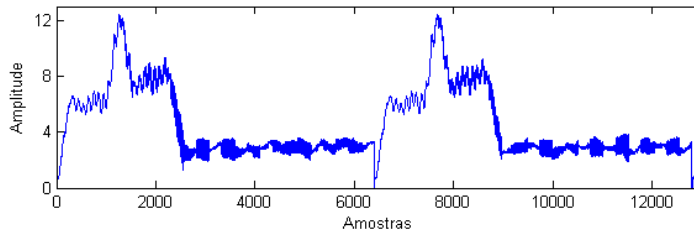


Figura 4.9: Resultado do cálculo da energia para duas tramas.

$$ms1 = \sum_{k=m}^{m+L-1} |r_k|^2 + |r_{k+N_c}|^2 \quad (4.1)$$

$$ms2 = \sum_{k=m}^{m+L-1} r_k^* r_{k+N_c} \quad (4.2)$$

Na implementação do cálculo da correlação é feita uma multiplicação complexa entre as amostras atuais e o complexo conjugado das amostras distanciadas 1024 posições destas, figura 4.10. De seguida são novamente utilizados dois filtros IIR, um para cada componente do resultado da multiplicação complexa, calculando assim a correlação para uma janela de 256 amostras. Um módulo CORDIC presente na biblioteca *Xilinx Blockset*, é usado para o cálculo do módulo e da fase à saída dos filtros. O resultado do cálculo do módulo apresenta um pico máximo quando a janela de procura se encontra sincronizada nas amostras do CP e nas amostras das quais este foi copiado. Na figura 4.11 (a) é possível observar este efeito, sendo perceptível que o pico de correlação da sequência *Zadoff-Chu* tem uma amplitude superior aos picos de correlação resultantes dos dados, novamente devido à sequência de sincronismo ser transmitida com uma potência superior. O CORDIC no modo de translação obtém o módulo da seguinte forma:

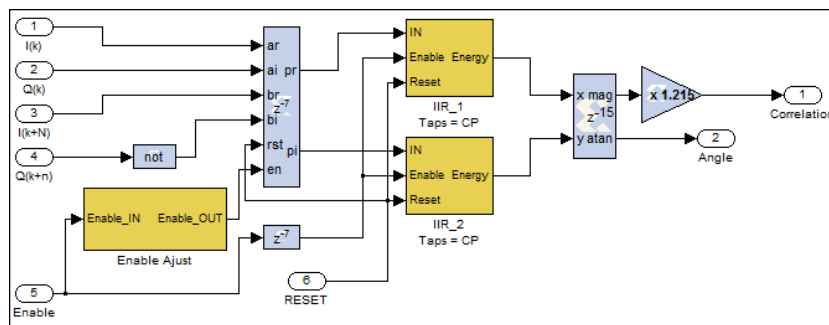


Figura 4.10: Implementação do cálculo da correlação e estimativa do desvio de fase na plataforma *Xilinx System Generator*.

$$|Z| = \sqrt{X_{in}^2 + Y_{in}^2} \quad (4.3)$$

onde X_{in} e Y_{in} representam as componentes real e imaginária colocadas nas entradas do CORDIC. Quanto ao ângulo, será uma estimativa do desvio de fase ocorrido entre dois conjuntos de 256 amostras distanciados de 1024 amostras, figura 4.11 (b). Nesta figura é possível observar a variação desta estimativa ao longo do tempo, estando esta síncrona com o resultado da correlação. Esta estimativa é importante quando o valor da correlação é máximo, permitindo assim saber o desvio de fase ocorrido nas amostras da sequência *Zadoff-Chu*. As imagens da figura 4.11 foram retiradas quando não existe CFO, portanto, os picos de correlação máximos estão associados a desvios de fase com valor zero. Esta estimativa será posteriormente usada para proceder à correção do CFO nos dados. A estimativa será calculada da seguinte forma pelo CORDIC:

$$\hat{\theta}_{ini} = \arctan\left(\frac{Y_{in}}{X_{in}}\right) \quad (4.4)$$

O valor resultante da correlação, saída do módulo do CORDIC, é posteriormente multiplicado por um fator de escala de 1.25. Este fator de escala é aplicado para que o resultado da estimativa, subtração da energia ao resultado da correlação, seja normalizado a 0. Foi também utilizado um filtro passa-baixo para suavizar este resultado.

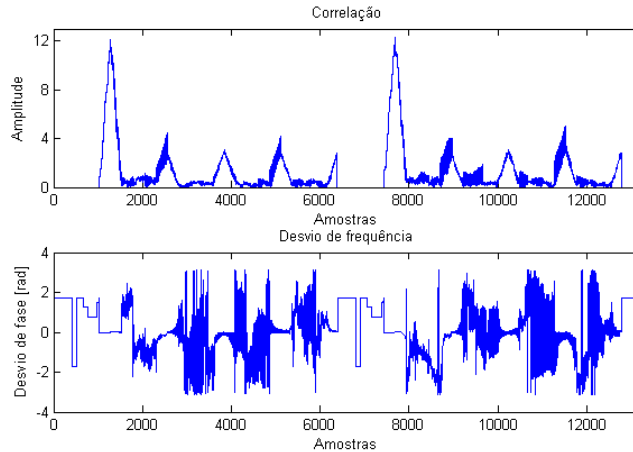


Figura 4.11: Resultados do algoritmo de estimação para duas tramas.

4.2.2 Decisor da detecção da sequência *Zadoff-Chu*

Após a implementação do algoritmo de van de Beek, é utilizada a estimação por este efetuada para proceder à sincronização do início da trama. Na figura 4.12 encontra-

se ilustra a implementação do bloco *Frame_Detection*, bloco cuja função é fazer esta sincronização e também a estimação do CFO. Foi então criada uma máquina de estados, denominada *Peak_Detector*, para este fim. Devido à sequência *Zadoff-Chu* ser transmitida com uma potência superior, a estimação para esta sequência resulta num pico distinto dos picos resultantes para os símbolos de dados, figura 4.13 (b). Estas características distintas foram utilizadas para proceder à deteção do início de trama. Realçando que os picos resultam da correlação existente entre as amostras do CP e das amostras das quais este foi copiado, estes ocorrem nas últimas amostras de cada símbolo OFDM.

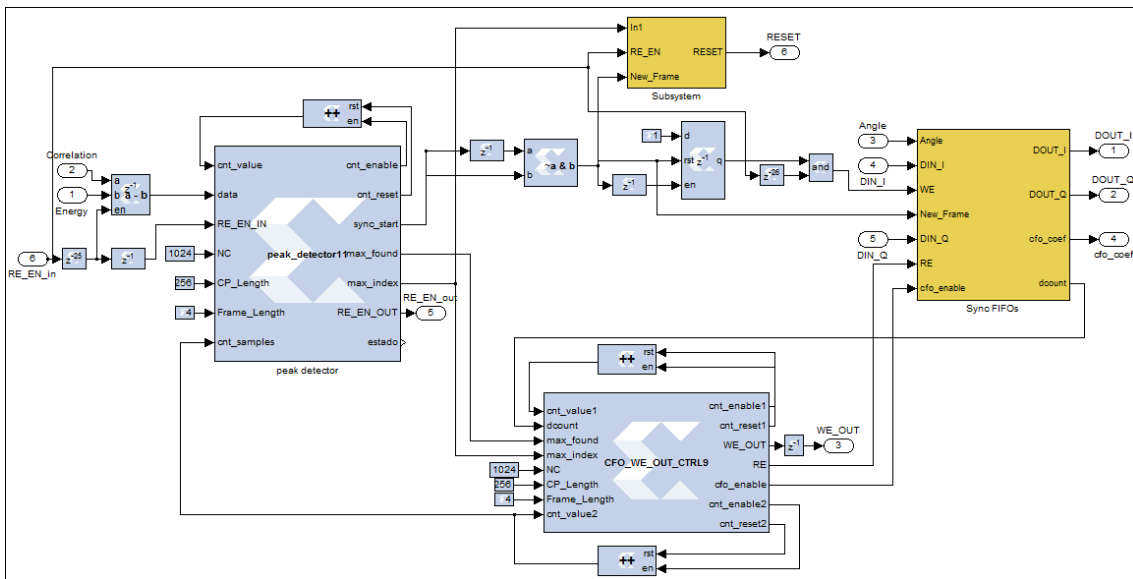


Figura 4.12: Implementação do bloco *Frame_Detection*.

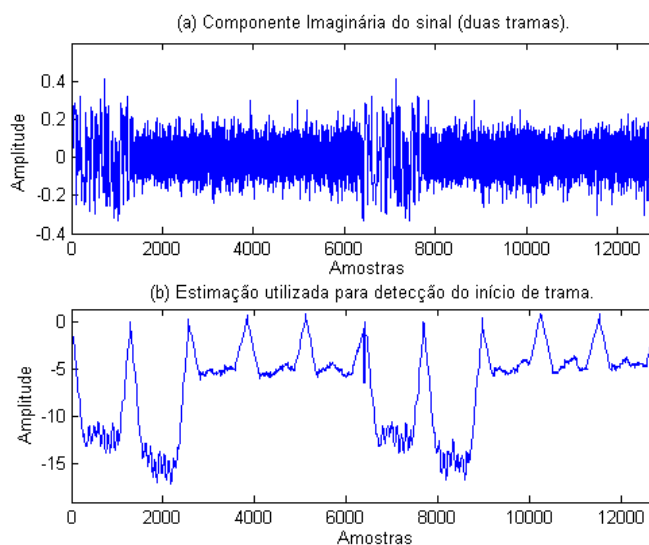


Figura 4.13: Resultado da estimador ML de van de Beek (b) em relação ao conteúdo contido nos símbolos OFDM. Encontrando-se apenas representada a componente imaginária dos símbolos OFDM (a).

O diagrama de estados do decisor é apresentado na figura 4.14. O estado inicial espera por uma queda no resultado da estimação, indício da chegada de uma sequência *Zadoff-Chu*, sendo o limiar de decisão -10. No próximo estado é verificado se o estimador já deixou sair todos os símbolos de dados, recorrendo a um contador referente ao número de amostras da trama que já saíram das memórias. Este estado é necessário, uma vez que o estimador se re-sincroniza no caso da detecção de uma nova sequência *Zadoff-Chu*, fazendo assim *reset* às memórias que armazenam os dados. Devido à arquitetura assíncrona do sistema, é necessário verificar se ainda resta informação do último símbolo de dados da trama, antes de proceder ao re-sincronismo, para que não ocorram perdas. Portanto, no caso das memórias possuírem dados respeitantes a parte do último símbolo de dados da trama, o estimador espera que estes saiam. Caso os dados sejam respeitantes a qualquer outro símbolo de dados da trama, é feito um re-sincronismo, garantindo assim que o estimador se re-sincroniza sempre que é detectada uma sequência *Zadoff-Chu*.



Figura 4.14: Diagrama de estados da máquina de estados *Peak_Detector*.

O estado seguinte consiste na procura do pico máximo, correspondente à última amostra da sequência *Zadoff-Chu*. Uma vez que esta amostra se encontra no fim do símbolo, amostra 1280 ($L + N_c$), é garantido que a janela de pesquisa apenas necessita de ter um comprimento de 1280 amostras. Isto é garantido porque, no caso ideal, o resultado da subtração da energia pela correlação baixa de zero na primeira amostra do símbolo e 1279 amostras depois, ocorre o máximo. Como o limiar de decisão da sequência *Zadoff-Chu* é

-10, é mais do que garantido que nas próximas 1280 amostras está o máximo resultante do pico de correlação, figura 4.15. Nesta figura é possível observar, a azul, o resultado da estimação do algoritmo ML de van de Beek para a sequência *Zadoff-Chu* e, a vermelho, a janela de pesquisa do máximo, indicada pela subida da linha vermelha. De forma a saber a posição do máximo, recorreu-se à utilização de um contador para referenciar a posição do máximo em relação à entrada no estado *Peak_Detection*.

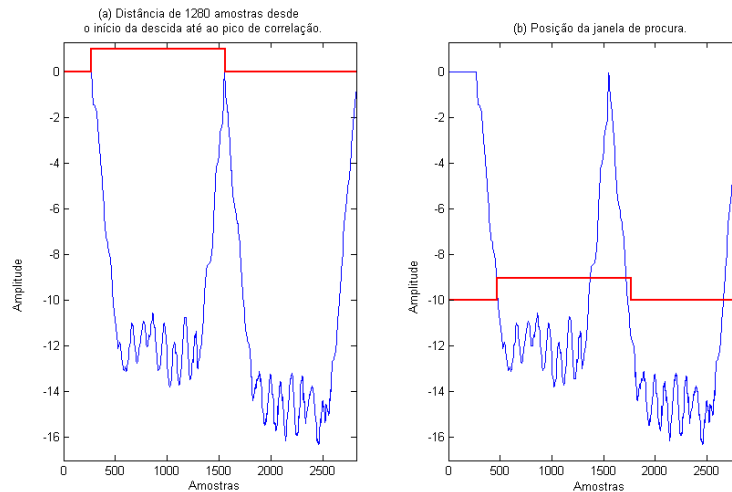


Figura 4.15: Janela de procura do pico de correlação resultante da sequência *Zadoff-Chu*.

Uma vez que há novamente uma quebra após o pico máximo da correlação, é necessário incluir um estado para esta descida e posterior subida, pois caso esta descida não fosse tida em conta, seria detetada com uma nova sequência *Zadoff-Chu*, procedendo ao re-sincronismo. Como tal, o estado seguinte consiste na detecção da descida, sendo o limiar de decisão o valor -3. Por fim, o último estado verifica a subida, com um limiar de -3. Após isto, a máquina de estados *Peak_Detector* volta ao estado inicial, aguardando a chegada de uma nova sequência *Zadoff-Chu*.

4.2.2.1 Estimação temporal recorrendo a memórias FIFO

A máquina de estados *Peak_Detector* procura encontrar o pico máximo da sequência *Zadoff-Chu*, sendo este usado para sincronização do início da trama. A amostra subsequente à amostra correspondente ao máximo de correlação, será a primeira amostra do primeiro símbolo OFDM de dados. Acima foi referido a utilização de um contador para encontrar a posição do máximo, este inicia a contagem no estado *Peak_Detection* da máquina de estados *Peak_Detector*. Apesar de existir uma referência da posição do máximo em relação à entrada no estado *Peak_Detection*, é também necessário referenciar os dados aquando da entrada neste estado. A sincronização temporal é então conseguida descartando tantas amostras, referentes à sequência *Zadoff-Chu*, quantas o valor indicado

pelo contador. Esta referência é conseguida recorrendo ao uso de memórias FIFO, uma memória para cada componente de dados e outra para a estimação do desvio de frequência. Garante-se assim um sincronismo entre o contador de detecção do máximo, as amostras das componentes real e imaginária, e a estimação do desvio de fase. Para que este sincronismo seja garantido, cada vez que o detetor de pico entra no estado *Peak_Detection*, é feito *reset* às memórias. Assim é garantido que as memórias se encontram vazias no início da escrita, uma vez que poderiam conter dados no caso de ter ocorrido um re-sincronismo.

Portanto, a escrita nas memórias FIFO, utilizadas para sincronismo com o contador, é iniciada após a detecção do máximo por parte do *Peak_Detector*. Tal como referido na secção 4.1.2, a arquitetura de funcionamento dos blocos do recetor consiste na escrita em memórias colocadas na entrada do bloco. Assim, as memórias utilizadas para sincronismo com o contador apenas devem ser escritas quando os dados são lidos das memórias de entrada, tendo em conta o atraso nestes, devido ao tempo de processamento da energia e da correlação.

4.2.3 Sincronização temporal e estimação do CFO

A máquina de estados *CFO_WE_Ctrl* tem como função fazer a sincronização temporal, estando o seu diagrama de estados ilustrado na figura 4.16. A sincronização é conseguida descartando as amostras referentes à sequência *Zadoff-Chu* e sincronizando o *WE_{OUT}* apenas com os símbolos de dados. Esta máquina de estados tem como entradas os valores de dois contadores, *cntvalue1* e *cntvalue2* e o número de amostras contidas nas memórias usadas para sincronização, *dcount*. Para além destas entradas, tem ainda informação referente ao número de subportadoras, N_c , ao tamanho do CP, L , e ao tamanho da trama excluindo a sequência *Zadoff-Chu*, *Frame_Length*. Para conseguir proceder à sincronização temporal, tem ainda como entradas as saídas do módulo *Peak_Detector*, *max_found* e *max_index*, possuindo assim informação da ocorrência da detecção de uma sequência *Zadoff-Chu* e do número de amostras até ao fim do símbolo de sincronismo.

Este módulo é iniciado no estado *Wait_Max_Found*, onde espera a detecção de uma sequência de sincronismo por parte do módulo *Peak_Detector*. A detecção da sequência *Zadoff-Chu* é indicada pela asserção da entrada *max_found*. No estado *Sync_State* procede-se então à sincronização temporal, ou seja, são descartadas as amostras referentes ao símbolo utilizado para sincronismo. Este estado consiste apenas na leitura das memórias FIFO, lendo tantas amostras quantas as indicadas pela entrada *max_index*. Na transição para o estado seguinte, a saída *cfo_enable* é asserida, com o objetivo de obter o valor da estimação do CFO neste (amostrado no instante correspondente à última amostra da sequência *Zadoff-Chu*). O valor desta estimação é posteriormente dividido pela distância entre os conjuntos de amostras da estimação. Esta distância é 1024, uma vez que a

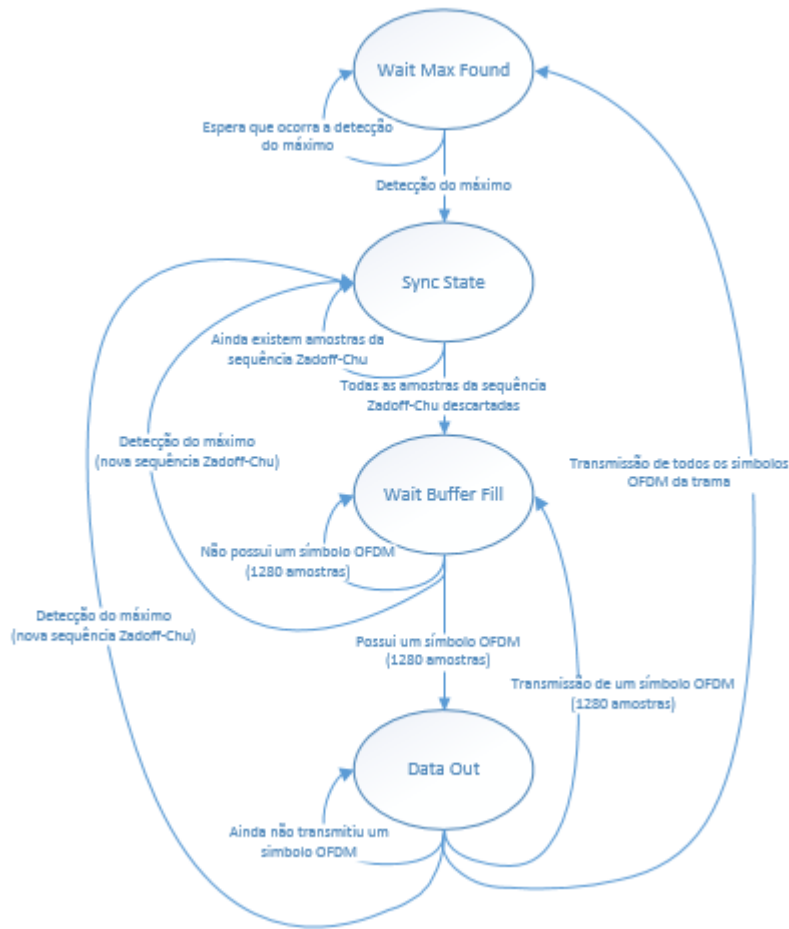


Figura 4.16: Diagrama de estados da máquina de estados *WE_Out_Ctrl*

estimação é feita utilizando o CP e as amostras das quais este foi copiado.

O estado *Wait_buffer_fill* é um estado intermédio de espera até que as memórias FIFO de sincronismo contenham um símbolo OFDM, portanto, 1280 amostras ($N_c + L$). Neste estado é também feito o controlo da variável auxiliar *aux_symbols*, variável utilizada para que a saída do módulo *CFO_Estimation* tenha um funcionamento em rajadas de símbolos OFDM. Esta variável é incrementada cada vez que as memórias FIFO contiverem 1280 amostras, ou seja, cada vez que possuírem um símbolo OFDM completo. O estado seguinte, *Data_out*, consiste no envio dos símbolos de dados. Como foi referido acima, consiste na transmissão dos símbolos de dados, símbolo a símbolo. Isto é conseguido transmitindo amostras até um valor múltiplo do tamanho do símbolo OFDM, mais precisamente $(N_c + L) * aux_symbols$. Após a transmissão de um símbolo, regressa ao estado anterior, *Wait_buffer_fill*, para garantir que as memórias contêm dados suficientes e onde é incrementada a variável *aux_symbols*. Este processo é repetido até enviar todos os símbolos de dados da trama, voltando ao estado inicial *Wait_Max_Found* após isto.

É de realçar que o módulo *CFO_WE_Ctrl* se re-sincroniza na eventualidade de ocorrer a

detecção de um símbolo de sincronismo, independentemente do estado em que se encontre. No diagrama de estados da figura 4.16 é possível observar que no caso de ocorrer a detecção de uma sequência *Zadoff-Chu*, o módulo regressa ao estado *Sync_State*. Regressando a este estado e não ao *Wait_Max_Found*, uma vez que a máquina de estados *Peak_Detector* já possui a posição do pico máximo, possibilitando assim a sincronização temporal.

Em conclusão, o módulo *CFO_Estimation*, faz uma estimação do início de trama e do CFO ocorrido, utilizando a sequência *Zadoff-Chu*. Esta estimação é utilizada para proceder à sincronização temporal, descartando o primeiro símbolo e transmitindo apenas os símbolos de dados. Para além de transmitir os símbolos de dados, transmite também a estimação do CFO, $\hat{w}_o = 2\pi\varepsilon$, para essa trama de dados. Na figura 4.17 é possível observar este princípio de funcionamento. No primeiro gráfico encontram-se ilustrados os símbolos OFDM, o primeiro é a sequência *Zadoff-Chu* e os quatro seguintes são símbolos de dados. O sinal a vermelho é o WE_{OUT} , que está apenas presente nos símbolos de dados, descartando assim a sequência utilizada para fins de sincronismo. No gráfico de baixo é possível observar o valor do CFO estimado, e tal como referido, este é constante para a trama de dados. O módulo *CFO_Correction* é quem tem a tarefa de corrigir o desvio de fase ocorrido, usando para isso esta estimação.

Na secção seguinte será feita uma descrição da implementação, na plataforma *Xilinx System Generator*, do algoritmo de estimação de CFO proposto por Carlos Ribeiro em [2].

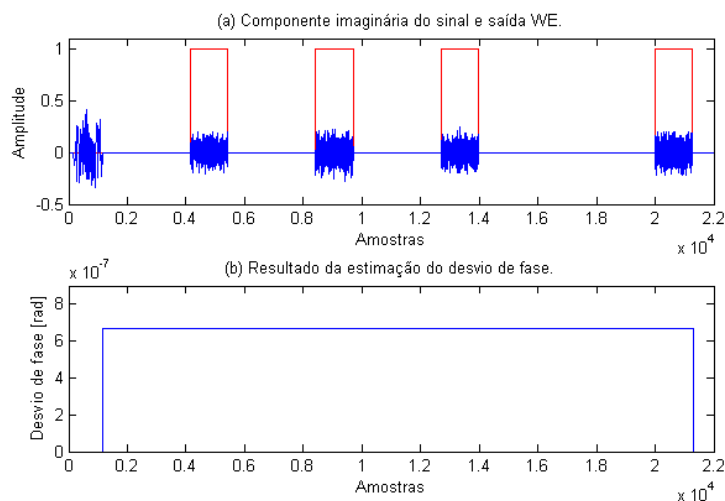


Figura 4.17: Comportamento das saídas do bloco *Frame_Detection*.

4.3 Implementação do algoritmo de C. Ribeiro na plataforma *System Generator*

Na secção 3.3.1.1 foi feita uma revisão teórica do método proposto por Carlos Ribeiro em [2], o qual permite a estimação do CFO presente nos símbolos recebidos recorrendo à estrutura dos pilotos. Tal como já foi referido, este algoritmo é usado apenas para estimação do CFO, portanto, a sincronização temporal é implementada pelo algoritmo de van de Beek [1]. Como o número de subportadoras é $N_c = 1024$ e a separação dos pilotos no FD é $N_f = 4$, logo $N_t = 256$. Na equação 3.10, a componente $diag \left(\left[1 \quad \exp \left(-j \frac{2\pi}{N_c} k_{ini} \right) \quad \dots \quad \exp \left(-j \frac{2\pi}{N_c} k_{ini} (N_c - 1) \right) \right] \right)$ é usada para desviar a frequência das subportadoras do símbolo, de acordo com a posição inicial do primeiro piloto no FD, k_{ini} . Uma vez que este algoritmo apenas utiliza as N_c subportadoras para estimação do CFO, é utilizado o bloco *CP_Rem* antes deste bloco de estimação.

Relembrando que este algoritmo tira partido dos símbolos OFDM compostos por dados e pilotos, as memórias FIFO que armazenam os dados que aguardam processamento apenas necessitam de armazenar os símbolos com estas características. Memórias FIFO adicionais são utilizadas para o armazenamento dos dados recebidos, colocando-os na saída à medida que o estimador obtém a estimação do CFO. Na figura 4.18 é possível observar a implementação deste algoritmo na plataforma *Xilinx System Generator*. A constituição deste é idêntica a qualquer bloco do recetor, contendo os *buffers* de entrada, as duas máquinas de estados e os blocos de processamento específico. É constituído ainda pelas memórias adicionadas, acima referidas, e por uma máquina de estados de controlo destas. Quanto aos blocos *Estimador* e *We_Pilot_Symbols*, o primeiro contém todo o processamento deste algoritmo, enquanto o segundo tem a função de seleccionar os símbolos OFDM compostos por dados e pilotos.

Este algoritmo, tal como referido na secção anterior, recorre à utilização de um CFO candidato para a estimação do CFO presente nos dados. De forma a aumentar a rapidez da estimação, são utilizados nove CFOs candidatos em paralelo. A gama destes CFOs inicialmente abrange toda a gama de estimação do algoritmo, sendo refinada à medida que o resultado da estimação se aproxima do verdadeiro CFO. O valor dos nove CFOs candidatos nesta fase inicial é $[-15, -7, -2.5, -1, 0, 1, 2.5, 7, 15]$ kHz, onde cada valor corresponde a um CFO candidato aplicado ao sinal de entrada. Relembrando que a gama de estimação está ligada ao espaçamento entre pilotos no FD, como $N_f = 4$, a gama é $\left[-\frac{N_f \Delta f}{2}, \frac{N_f \Delta f}{2} \right] = \left[-\frac{4 \cdot 15}{2}, \frac{4 \cdot 15}{2} \right] = [-30, 30]$ kHz. Apesar da gama inicial dos CFOs candidatos ser muito inferior ao limite da capacidade de estimação, devido à estrutura do estimador, este consegue estimar corretamente o valor do CFO mesmo quando este ultrapassa os 15 kHz. Quando a diferença entre as 2 estimações com menor valor da função

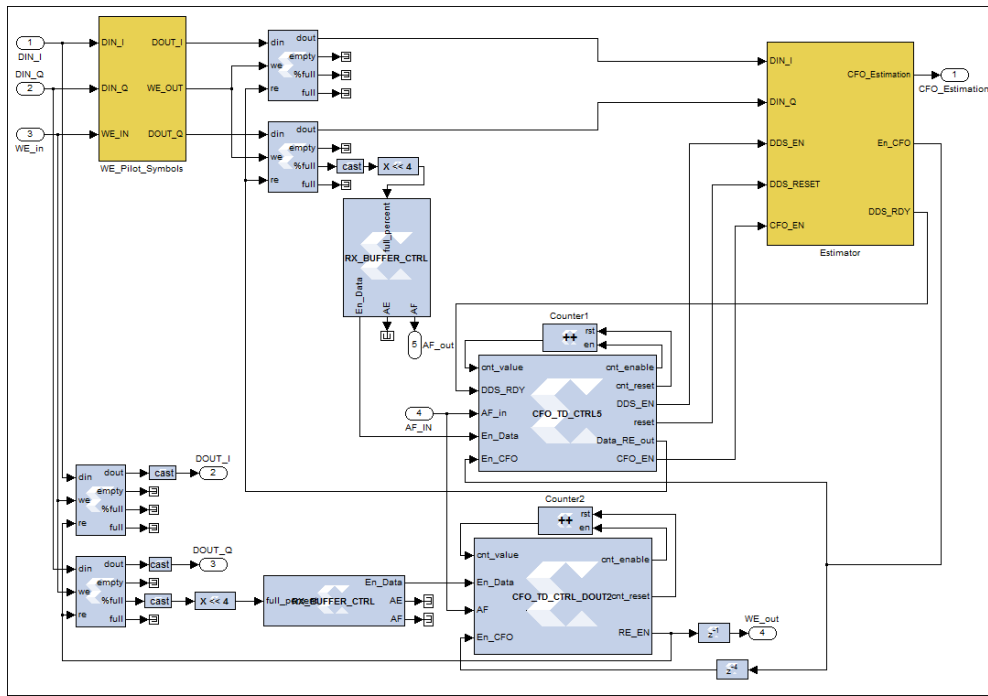


Figura 4.18: Implementação do estimador de CFO de C. Ribeiro, na plataforma *Xilinx System Generator*.

custo é inferior a 8 Hz, a janela de procura passa a ser em torno do CFO associado a essa estimação, variando entre $[CFO_{Lower} - 4 \text{ Hz}, CFO_{Lower} + 4 \text{ Hz}]$. CFO_{Lower} representa o CFO candidato, introduzido anteriormente, cujo resultado da função custo resultou no valor inferior, relembrando que o mínimo da função custo representa o CFO verdadeiro. Portanto, a gama da janela de procura passa a ser 8 Hz em torno do CFO estimado anteriormente. A gama pode ser bastante reduzida uma vez que não ocorrem variações bruscas de CFO de símbolo para símbolo, devido aos osciladores apresentarem variações lentas da sua frequência de saída.

A utilização de nove CFOs candidatos em paralelo leva, conseqüentemente, a que o restante processamento definido neste algoritmo também seja replicado nove vezes. Quanto à utilização da exponencial complexa dependente do k_{ini} , acima referida, é independente do CFO candidato aplicado e do consecutivo processamento. Esta apenas necessita de ser aplicada uma vez aos dados de entrada, sendo o resultado introduzido nos vários blocos replicados. Assim, o bloco *We_Pilot_Symbols* para além de fazer o controlo da escrita nas memórias FIFO, tem também como função aplicar a exponencial complexa dependente do k_{ini} . Conseqüentemente, as memórias FIFO armazenarão apenas os símbolos OFDM que contenham dados e pilotos, encontrando-se estes já desviados na frequência de acordo com a posição da primeira subportadora que contém um símbolo piloto.

4.3.1 Implementação do processamento do estimador

Na figura 4.19 é possível observar a constituição do bloco *Estimador* da figura 4.18, contendo todo o processamento envolvido na estimação do CFO. Este bloco tem como função aplicar os vários CFOs candidatos em paralelo e obter o resultado da função custo para os CFOs candidatos. Além disto, tem também como função a redefinição dos CFOs de acordo com os resultados das estimações anteriores. Na figura 4.19, os blocos denominados *CFO* introduzem nos dados recebidos o CFO candidato, sendo este definido pelo valor da entrada *CFO* destes blocos. Os vários blocos *Estimation* implementam o restante processamento necessário para a estimação do CFO. É feita uma divisão do símbolo OFDM em N_f partes de N_t amostras, sendo este conjunto de amostras posteriormente somado. De seguida, são descartadas as L_p amostras iniciais do resultado da soma anterior, eliminando assim as amostras que contêm energia do CIR. Por fim, é calculada a energia das amostras restantes, ou seja, das amostras que apenas dependem dos dados e ruído.

A máquina de estados *Estimation_ctrl*, presente no centro da figura 4.19, tem a função de encontrar as duas estimações com valores mínimos. Esta tem como entradas os nove resultados dos blocos *Estimation*, e como saídas a saída *Lower_Estimation* e *Second_Lower_Estimation*. Nestas saídas são colocados os índices, 0 a 8, do resultado inferior e do segundo resultado inferior, respectivamente.

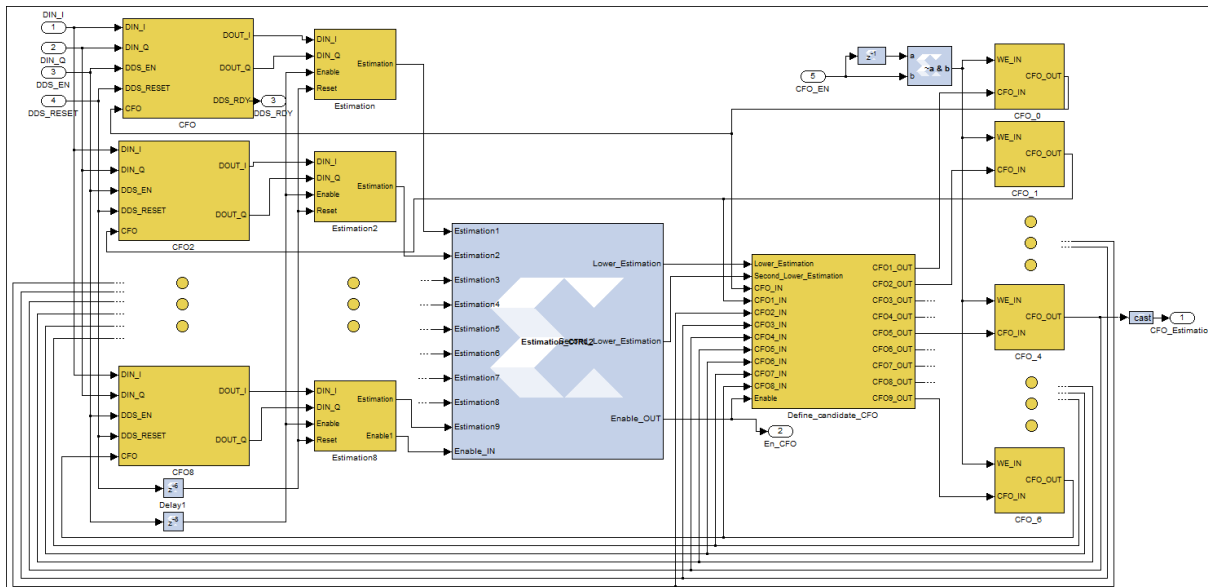


Figura 4.19: Implementação do bloco *Estimator*, na plataforma *Xilinx System Generator*

Na figura 4.20 é possível observar a constituição dos blocos *Estimation*, que têm a função de implementar o restante processamento da função custo. Uma vez que $N_f = 4$, são feitas 4 somas de 256 amostras ($N_t = 256$), implementadas provocando atrasos

múltiplos de N_t nos dados, figura 4.20. Assim, quando a primeira amostra chegar à saída do ramo com maior atraso ($3 * N_t = 768$), no ramo com um atraso $2 * N_t$ estará a amostra 257. No ramo com um atraso N_t , a primeira amostra presente será a 513 e no ramo sem atraso será a amostra 769. As amostras são somadas desde o instante em que a primeira amostra do símbolo OFDM chega à saída do ramo com o atraso superior, até perfazer a soma de todas as N_t amostras. Isto é facilmente conseguido lendo *bursts* de 1024 amostras, fazendo um *reset* aos atrasos e somadores no fim de cada *burst*. Assim, é garantido que os atrasos têm o funcionamento esperado, e que são apenas feitas 256 somas, voltando o processo a repetir-se a cada *burst*. De seguida, são descartadas 38 (L_p) amostras, facilmente conseguido de-asserindo o sinal *enable* para as primeiras 38 amostras. Por fim, é calculada a energia destas somas, obtendo o resultado final da função custo. Na próxima secção será explicado o processo de redefinição da gama da janela de procura do CFO.

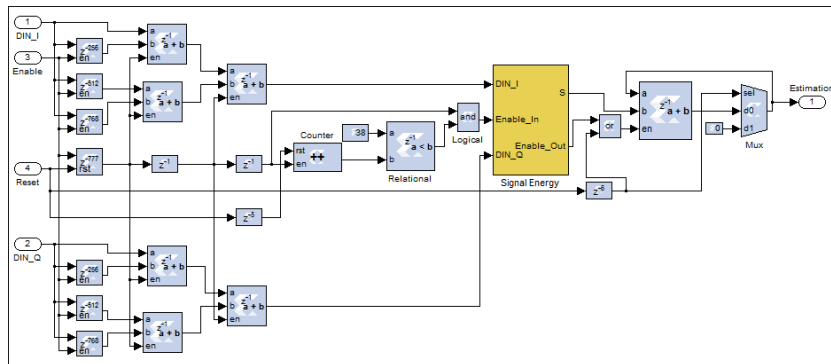


Figura 4.20: Implementação dos blocos *Estimation*, na plataforma *Xilinx System Generator*

4.3.2 Definição da gama da janela de estimação de CFO

Na secção anterior foi introduzido o facto de serem utilizados nove CFOs candidatos em paralelo, obtendo assim nove estimações por cada símbolo OFDM. Nesta secção será explicado como são usados os resultados dessas estimações para redefinição dos CFOs candidatos. Na secção anterior foi também referido o uso de uma máquina de estados, *Estimation_ctrl*, que indica qual a estimação com o resultado inferior e a segunda menor (esta informação é colocada nas saídas *Lower Estimation* e *Second Lower Estimation*, respectivamente). O bloco *Define_candidate_CFO* utiliza esta informação juntamente com a informação dos valores dos CFOs candidatos para proceder à redefinição destes.

O princípio de funcionamento do bloco *Define_candidate_CFO* baseia-se na utilização da diferença entre os dois CFOs candidatos referenciados pela máquina de estados *Estimation_ctrl*. Como esta fornece os índices das estimações inferiores, os valores dos CFOs

candidatos inferiores são facilmente obtidos recorrendo à utilização de *multiplexers*. Os nove CFOs candidatos são redefinidos de acordo com as expressões presentes na equação 4.5, onde $CFO_{Distance}$ representa o resultado da diferença acima referida. O CFO candidato que resultou na estimação inferior é definido como o CFO candidato central, ou seja, o CFO candidato número cinco. Enquanto os restantes CFOs candidatos são redefinidos recorrendo à diferença entre este e o segundo menor.

$$\begin{aligned}
CFO_{candidato1} &= CFO_{Lower} - CFO_{Distance} \\
CFO_{candidato2} &= CFO_{Lower} - 0.75 * CFO_{Distance} \\
CFO_{candidato3} &= CFO_{Lower} - 0.5 * CFO_{Distance} \\
CFO_{candidato4} &= CFO_{Lower} - 0.25 * CFO_{Distance} \\
CFO_{candidato5} &= CFO_{Lower} \\
CFO_{candidato6} &= CFO_{Lower} + 0.25 * CFO_{Distance} \\
CFO_{candidato7} &= CFO_{Lower} + 0.5 * CFO_{Distance} \\
CFO_{candidato8} &= CFO_{Lower} + 0.75 * CFO_{Distance} \\
CFO_{candidato9} &= CFO_{Lower} + CFO_{Distance}
\end{aligned} \tag{4.5}$$

Este princípio de redefinição mantém-se enquanto a diferença entre os CFOs candidatos, correspondentes às duas estimações com menor valor da função custo, for superior a 8 Hz. A partir do momento em que esta diferença passe a ser inferior a este valor, os CFOs candidatos passam a ser redefinidos de acordo com as expressões seguintes:

$$\begin{aligned}
CFO_{candidato1} &= CFO_{Lower} - 4 \\
CFO_{candidato2} &= CFO_{Lower} - 3 \\
CFO_{candidato3} &= CFO_{Lower} - 2 \\
CFO_{candidato4} &= CFO_{Lower} - 1 \\
CFO_{candidato5} &= CFO_{Lower} \\
CFO_{candidato6} &= CFO_{Lower} + 1 \\
CFO_{candidato7} &= CFO_{Lower} + 2 \\
CFO_{candidato8} &= CFO_{Lower} + 3 \\
CFO_{candidato9} &= CFO_{Lower} + 4
\end{aligned} \tag{4.6}$$

As expressões anteriores refletem o que já tinha sido referido na secção anterior, ou seja, que após o CFO mínimo ser encontrado, a janela de procura passa a ser em torno deste, variando apenas entre $[CFO_{Lower} - 4 \text{ Hz}, CFO_{Lower} + 4 \text{ Hz}]$, uma vez que não ocorrem variações bruscas de CFO de símbolo para símbolo.

Para armazenar os valores dos CFOs candidatos são utilizadas memórias RAM. Na

inicialização do sistema, os primeiros valores armazenados nestas memórias são obtidos a partir constantes, de acordo com a janela de procura. A partir do momento em que o bloco *Define_candidate_CFO* já possui os valores dos novos CFOs candidatos, as memórias RAM passam a armazenar os valores das saídas deste bloco.

Na figura 4.21 é possível observar a implementação do bloco *Define_candidate_CFO*, lembrando que este tem a função de redefinir os CFOs candidatos de acordo com a equação 4.5. Os CFOs de entrada estão ligados a *multiplexers*, possibilitando facilmente a escolha de um CFO específico, de acordo com os índices resultantes da estimação. Esta figura não representa toda a implementação do bloco *Define_candidate_CFO*, faltando apenas os blocos que implementam a definição dos CFOs candidatos de acordo com as expressões em 4.6. A implementação destas expressões não é mais do que uma cópia dos somadores e multiplicadores da figura 4.21. Esta implementação apenas difere da anterior na não utilização da diferença entre os CFOs candidatos que resultaram nas estimações com valores inferiores. Em vez disso, são somadas e subtraídas constantes com os valores indicados nas expressões em 4.6, sendo os resultados destas somas introduzidos nas portas *or* das respectivas saídas dos CFOs candidatos.

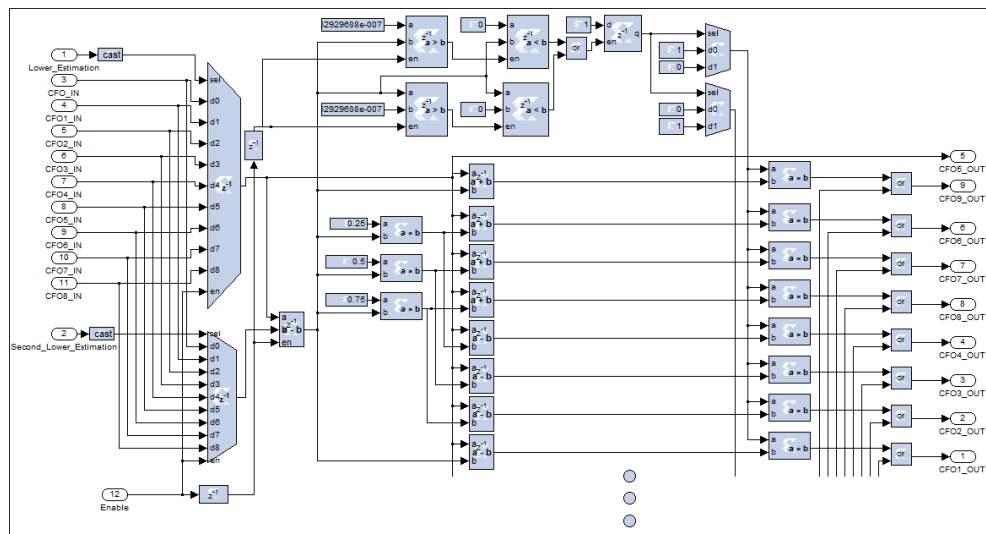


Figura 4.21: Implementação do bloco *Define_candidate_CFO*, na plataforma *Xilinx System Generator*.

Os comparadores no topo da figura 4.21 são utilizados para decidir qual o sinal que é colocado no CFO candidato de saída. Esta decisão é feita de acordo com o valor da diferença entre os CFOs candidatos. No caso deste valor ser superior a 8 Hz as saídas dos CFOs candidatos são definidos de acordo com as expressões em 4.5; no caso de ser inferior são definidas de acordo com as expressões em 4.6. Para tal, no primeiro caso as saídas dos somadores que implementam as expressões em 4.5 são multiplicados por 1 e as saídas dos somadores que implementam as expressões em 4.6 são multiplicadas por 0. No segundo caso, os papéis são invertidos, sendo as primeiras saídas multiplicadas por 0

e as segundas por 1. Por fim, os CFOs candidatos colocados nas saídas resultam da soma das componentes resultantes destas multiplicações.

4.4 Bloco *CFO Correction*

O bloco *CFO_Correction*, tal como o nome indica, tem como função compensar o desvio de frequência presente nos dados recebidos, figura 4.22. Este bloco utiliza a entrada *CFO_estimated*, alusiva ao valor do CFO estimado pelo bloco de estimação de CFO, para proceder a esta compensação. O bloco de estimação tanto pode ser o bloco com algoritmo de van de Beek, como o bloco de estimação com base nos pilotos de C. Ribeiro. Tal como foi referido no capítulo 3, o desvio de frequência segue a expressão $e^{\frac{j2\pi\epsilon n}{N_c}}$, sendo n é o índice temporal. A compensação do CFO passa por compensar cada amostra no domínio do tempo pelo respectivo CFO que esta sofreu. Isto é conseguido utilizando uma exponencial complexa com um desvio de frequência simétrico ao CFO estimado e normalizado, $e^{-\frac{j2\pi\epsilon n}{N_c}}$, compensando assim o efeito provocado por este.

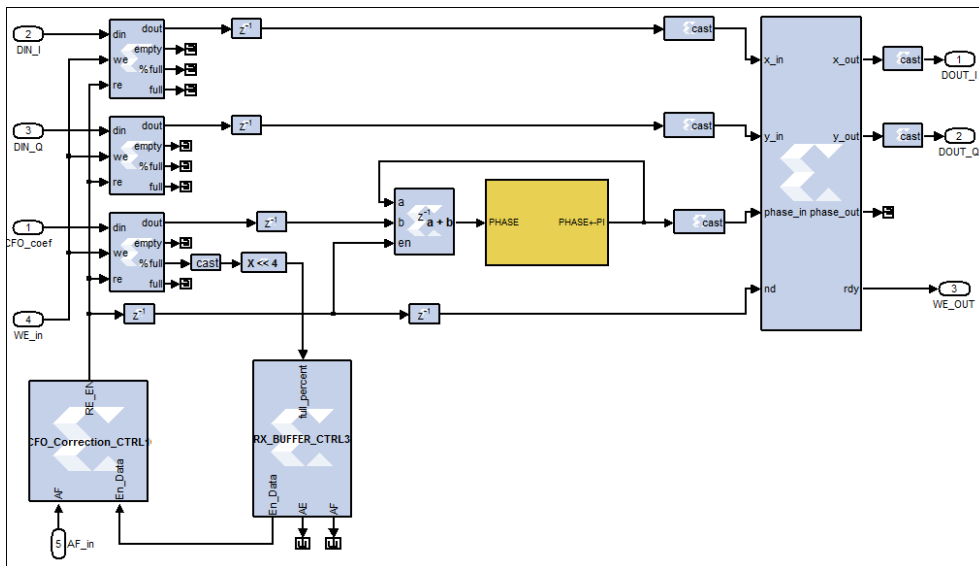


Figura 4.22: Implementação do bloco *CFO_Correction*, na plataforma *Xilinx System Generator*

O estimador de CFO já fornece uma estimativa de CFO normalizado ao espaçamento entre subportadoras, $2\pi\hat{\epsilon}$, mais corretamente o valor simétrico deste. Assim, o *CFO_Correction* apenas necessita de multiplicar este valor pelo índice temporal das amostras ao longo dos símbolos. Na prática, isto é facilmente conseguido incrementando o valor do CFO com o valor da saída do acumulador à medida que as amostras são lidas das memórias FIFO. Na figura 4.23, é possível observar as amostras da componente imaginária dos símbolos OFDM no domínio do tempo, juntamente com a respectiva fase de compensação pelo qual cada amostra será compensada.

Para proceder à compensação do CFO recorreu-se ao uso do CORDIC no modo de rotação. Tal como explicado na secção 2.4.2.1, as saídas do CORDIC são obtidas da seguinte forma:

$$I_{out} = I_{in} \cos \alpha - Q_{in} \sin \alpha \quad (4.7)$$

$$Q_{out} = Q_{in} \cos \alpha + I_{in} \sin \alpha, \quad (4.8)$$

onde I e Q representam as componentes real e imaginária dos símbolos OFDM no domínio do tempo, respectivamente. Uma vez que a entrada $phase_in$ do CORDIC é limitada entre $[-\pi, \pi]$, é necessário limitar o valor de α , uma vez que este vai aumentando/diminuindo à medida que o índice temporal aumenta. Assim, no caso do valor do incremento da fase de compensação ser inferior a $-\pi$ é-lhe somado 2π ; o processo inverso é feito no caso deste valor ser superior a π . Com isto é garantido que a fase de compensação varia dentro da gama de funcionamento do CORDIC. Na figura 4.23 é possível observar o incremento da fase de compensação ao longo do tempo, podendo observar-se que este varia dentro a gama correta.

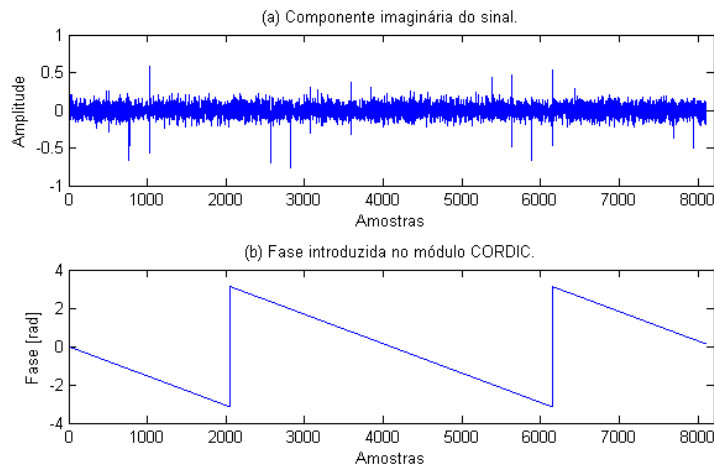


Figura 4.23: Compensação do CFO: Correspondência entre a fase e as amostras a ser compensadas por esta.

Quanto à operação feita pelo CORDIC para compensação do CFO, é fácil de demonstrar a origem das fórmulas usadas no cálculo das suas saídas. Tal como referido, a compensação passa por multiplicar os símbolos de dados, $I + jQ$, por um exponencial complexa, $e^{j\alpha}$. O cálculo feito pelo módulo CORDIC não é mais do que uma multiplicação das entradas por uma exponencial complexa com fase instantânea dada pela entrada $phase_in$, denominada por α na equação seguinte:

$$\begin{aligned}(I + jQ)e^{\alpha} &= (I + jQ)(\cos \alpha + j \sin \alpha) \\ &= I \cos \alpha + jQ \cos \alpha + jI \sin \alpha - Q \sin \alpha\end{aligned}\tag{4.9}$$

Se separarmos a parte real e imaginária do resultado, é facilmente perceptível que estas são iguais à equação 4.7 do cálculo das saídas do CORDIC.

No capítulo seguinte serão apresentados os resultados relativamente ao desempenho de ambos os estimadores implementados neste capítulo.

Capítulo 5

Resultados

Neste capítulo são analisados os resultados do desempenho do sistema perante os diferentes testes realizados. No cenário de simulação implementado, tal como já tinha sido referido anteriormente, os símbolos OFDM são constituídos por $N_c = 1024$ sub-portadoras e o CP tem um tamanho $L = 256$ amostras. Cada trama é constituída por 4 símbolos OFDM, dos quais o primeiro e terceiro símbolos transportam pilotos e dados (com uma separação entre pilotos na frequência, $N_f = 4$), enquanto os restantes apenas dados. Os testes realizados têm como objetivo analisar o desempenho dos estimadores implementados nas secções 4.2 e 4.3.

5.1 Estimador ML de van de Beek

Os testes realizados consistem em introduzir diferentes valores de CFO, com f_0 compreendido entre $[-12, 12]$ kHz. Uma vez que a gama do estimador ML de Beek é $[-\frac{\Delta f}{2}, \frac{\Delta f}{2}] = [-7.5, 7.5]$ kHz, o desvio de frequência introduzido é superior à gama do estimador. Foi também introduzido um canal rádio com um PDP com decaimento exponencial entre o transmissor e o recetor, usado para simular um ambiente *outdoor* (250ns rms delay spread). Além disso, foi também adicionado ruído AWGN, resultando em SNRs com valores entre 0 e 10 dB.

5.1.1 Desempenho em Ruído AWGN

Para as várias condições de simulação, o desempenho do estimador foi avaliado de acordo com o BER resultante e com a precisão da estimação de início de trama e do CFO. Na tabela 5.1 é possível observar o desvio médio da estimação do CFO efetuada por este estimador. Como esperado, o desempenho do estimador degrada-se à medida que a relação SNR diminui. No caso de não ser introduzido ruído AWGN, o resultado da estimação

revela bastante precisão, apresentando um desvio médio de cerca de 5 Hz, enquanto que para relações SNR desde 10 a 0 dB foram obtidos, em média, desvios de 26, 35, 43, 54, 90 e 105 Hz. Analisando a coluna referente ao CFO de 12 kHz, é possível constatar a limitação na gama de captura do estimador, uma vez que o erro de estimação é de 15 kHz, para qualquer que seja a relação SNR.

Tabela 5.1: Estimador ML de van de Beek: Desvio médio da estimação de CFO, em Hertz, para diferentes níveis SNR.

SNR (dB) \ CFO (Hz)	100	500	2000	4000	7000	12000
0	95.29	110.88	130.45	114.83	78.43	15056
2	76.89	78.17	74.67	115.71	100.48	14892
4	46.22	55.03	62.12	63.08	44.79	15034
6	40.23	45.92	33.52	51.14	43.26	15027
8	30.79	39.12	32.08	34.53	41.42	15027
10	25.98	21.97	21.86	25.48	34.85	15007
sem ruído	5.35	4.62	3.83	9.08	3.30	15001

A tabela 5.2 permite observar o desvio médio da estimação de início de trama efetuada por este estimador. É observável que o desempenho do estimador é perfeito para as relações SNR de 10 a 4 dB (quando o CFO introduzido não ultrapassa a gama de captura do estimador). Caso contrário, o resultado da estimação temporal degrada-se severamente. Para relações SNR de 2 dB ou 0 dB, o desempenho do estimador piora, chegando a falhar a posição ideal de início de trama em 148 amostras.

Tabela 5.2: Estimador ML de van de Beek: Desvio médio da estimação de início de trama efetuada, em amostras, para diferentes níveis SNR.

SNR (dB) \ CFO (Hz)	100	500	2000	4000	7000	12000
0	53	1	0	0	148	315
2	63	76	16	74	61	473
4	0	0	0	0	0	1024
6	0	0	0	0	0	1024
8	0	0	0	0	0	1024
10	0	0	0	0	0	1024
sem ruído	0	0	0	0	0	1024

A tabela 5.3 contém os valores do BER resultantes para os vários CFOs e níveis de relação SNR. Para relações SNR de 10 a 0 dB, a BER varia em média entre 1.32×10^{-3} e 3.64×10^{-1} , considerando que o CFO introduzido se encontra dentro da gama de captura do estimador. No caso das relações de 2 e 0 dB, as médias resultantes não deverão ser consideradas devido a existirem grandes variações no BER para os diferentes CFOs

introduzidos. Estas variações devem-se ao facto do *threshold* de decisão de início de trama ser atingido para os símbolos de dados, fazendo com que o estimador se re-sincronize em símbolos que não a sequência de sincronismo. Assim, uma vez que nem todos os símbolos são recebidos ou parte deles é perdida, a sua comparação com os símbolos transmitidos não é exequível.

Tabela 5.3: Estimador ML de van de Beek: BER resultante para os vários CFOs e níveis SNR.

CFO (Hz) \ SNR (dB)	100	500	2000	4000	7000	12000
0	4.28×10^{-1}	2.56×10^{-1}	2.58×10^{-1}	2.62×10^{-1}	4.93×10^{-1}	5.01×10^{-1}
2	4.08×10^{-1}	4.05×10^{-1}	1.82×10^{-1}	4.41×10^{-1}	3.29×10^{-1}	4.99×10^{-1}
4	1.02×10^{-1}	9.43×10^{-2}	9.17×10^{-2}	9.15×10^{-2}	8.76×10^{-2}	4.93×10^{-1}
6	2.88×10^{-2}	3.36×10^{-2}	3.30×10^{-2}	3.43×10^{-2}	3.35×10^{-2}	4.93×10^{-1}
8	6.37×10^{-3}	8.63×10^{-3}	1.62×10^{-2}	9.35×10^{-3}	9.17×10^{-3}	4.93×10^{-1}
10	1.49×10^{-3}	0.98×10^{-3}	1.22×10^{-3}	1.59×10^{-3}	1.32×10^{-3}	4.91×10^{-1}
sem ruído	0	0	0	0	0	4.92×10^{-1}

5.1.2 Desempenho em Ruído AWGN e canal rádio *outdoor*

Esta secção apresenta os resultados do desempenho do estimador quando introduzido um canal rádio *outdoor* (250ns rms delay spread) entre o transmissor e o receptor. Analisando as tabelas 5.4 e 5.5, é possível observar que a introdução do canal rádio *outdoor* tem uma elevada influência no desempenho do estimador. Quando na presença de apenas ruído AWGN, o estimador apresenta um erro máximo de CFO de 130.45 Hz, tabela 5.1, com a introdução do canal rádio *outdoor*, este erro chega a ser de 2121.98 Hz. Quanto à estimação do início de trama, esta também sofre uma degradação, chegando a falhar em média entre 50 a 264 amostras para os diferentes SNRs testados.

Tabela 5.4: Estimador ML de van de Beek: Desvio médio da estimação de CFO, em Hertz, para diferentes níveis SNR e um canal rádio *outdoor*(250ns rms delay spread).

CFO (Hz) \ SNR (dB)	100	500	2000	4000	7000
0	630.22	212.85	202.26	389.82	1769.19
2	675.77	953.31	1385.69	1256.58	785.29
4	402.06	416.60	773.42	1451.18	2121.98
6	447.19	240.59	783.66	857.77	1993.14
8	282.19	91.34	1024.40	1022.81	450.56
10	57.93	70.51	1088.88	1083.42	975.91
sem ruído	302.18	33.85	802.60	785.06	449.93

Como o canal rádio provoca desvanecimento, a energia do sinal recebido sofre grandes variações, fazendo com que o recetor perca de forma parcial ou total as tramas de dados

Tabela 5.5: Estimador ML de van de Beek: Desvio médio da estimação de início de trama, em amostras, para diferentes níveis SNR e um canal rádio *outdoor* (250ns rms delay spread).

SNR (dB) \ CFO (Hz)	100	500	2000	4000	7000
0	191	43	130	155	169
2	101	175	207	205	161
4	153	158	185	199	145
6	125	152	201	155	264
8	139	98	146	217	196
10	89	72	96	208	179
sem ruído	135	50	157	217	262

transmitidas. No caso da energia do sinal recebido ser muito baixa, o resultado do algoritmo de estimação temporal não atinge o *threshold* de decisão da detecção da sequência *Zadoff-Chu*, perdendo estas tramas por completo. Quando a energia é muito elevada, o *threshold* de decisão de início de trama é atingido durante os símbolos de dados. Os resultados apresentados na tabela 5.4 mostram que o desempenho do estimador piorou muito com a introdução do canal rádio. Contudo, após uma análise cuidada dos resultados, constatou-se que o resultado da estimação do CFO, por vezes, falhou completamente o CFO correto, fazendo com que o desvio médio das estimações aumente consideravelmente.

A figura 5.1 expõe o resultado do módulo do erro da estimação de CFO (em dB) ao longo do tempo, no caso de um CFO de 7 kHz com uma relação SNR de 10 dB. É possível observar que ocorreram estimações que falharam completamente o CFO correto, com erros perto dos 38.5 dB. Com vista a expor o desempenho do estimador

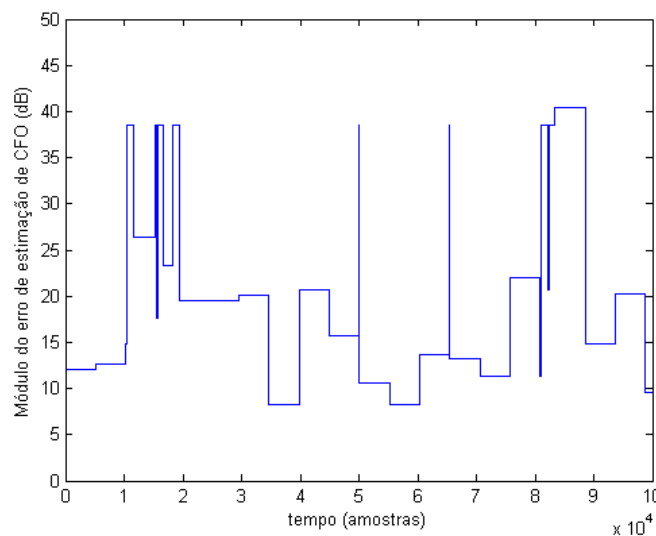


Figura 5.1: Módulo do erro de estimação de CFO (dB) ao longo do tempo.

Com vista a expor o desempenho do estimador de forma mais detalhada, foram elaboradas as tabelas 5.6 e 5.7. A primeira é semelhante à tabela 5.4, mas agora, para o cálculo do desvio médio foram ignoradas as estimações com uma diferença acima de 1500 Hz do CFO correto. A tabela 5.7 contém a percentagem de estimações de CFO com uma diferença superior a 1500 Hz, permitindo assim ter noção do impacto causado nas estimações efetuadas.

Tabela 5.6: Estimador ML de van de Beek: Desvio médio da estimação de CFO, em Hertz, ignorando diferenças superiores a 1500 Hz.

SNR (dB) \ CFO (Hz)	100	500	2000	4000	7000
0	248.37	212.85	182.59	165.74	107.01
2	188.24	156.05	113.04	111.96	90.75
4	94.96	94.07	146.74	86.63	99.31
6	98.98	102.36	137.25	56.64	71.25
8	61.37	78.55	89.37	61.32	79.75
10	57.93	70.51	46.98	60.40	62.43
sem ruído	32.30	33.85	66.10	43.24	51.16

Tabela 5.7: Estimador ML de van de Beek: Percentagem das estimações de CFO com uma diferença superior a 1500 Hz.

SNR (dB) \ CFO (Hz)	100	500	2000	4000	7000
0	9.52	0	0.98	3.79	16.01
2	10.45	16.39	35.59	20.25	9.12
4	6.73	6.92	20.13	23.48	22.73
6	9.19	8.85	19.85	15.53	23.43
8	5.18	0.23	23.11	16.22	4.55
10	0	0	25.81	16.52	10.08
sem ruído	6.33	0	24.96	12.57	5.70

Fazendo uma análise comparativa das tabelas 5.4 e 5.6, é possível observar o impacto significativo no desempenho do estimador, quando se ignoram as diferenças superiores a 1500 Hz, já que o maior erro médio passa de 2121.98 Hz para 248.37 Hz. Analisando a tabela 5.7, é possível observar que, em alguns casos, a percentagem de estimações de CFO com diferença superior a 1500 Hz é significativa, atingindo 35.59% das estimações efetuadas. Contudo, é de realçar que sensivelmente metade dos casos têm menos de 10% de estimações com diferenças superiores a 1500 Hz e as restantes têm em média 15% a 25%. A título de exemplo, mantendo as mesmas condições enunciadas para a figura 5.1, o desvio médio do estimador é de 975.91 Hz. Não tendo em conta as diferenças superiores a 1500 Hz e com uma percentagem de 10.08%, o desvio passa a ser 62.43 Hz. Estes valores podem ser consultados nas tabelas 5.6 e 5.7, respectivamente.

5.2 Estimador com base nos pilotos no TD de C. Ribeiro

Os testes realizados ao estimador proposto por C. Ribeiro em [2] centraram-se na precisão do CFO estimado. Como a gama dinâmica deste estimador é superior à do algoritmo de van de Beek, $\left[-\frac{N_f \Delta f}{2}, \frac{N_f \Delta f}{2}\right] = \left[-\frac{4*15}{2}, \frac{4*15}{2}\right] = [-30, 30]$ kHz, a gama de CFOs introduzidos foi superior à gama anteriormente referida. As restantes condições de simulação mantiveram-se inalteradas.

5.2.1 Desempenho em Ruído AWGN

Na tabela 5.8 é possível observar o desvio médio da estimação do CFO efetuada por este estimador. Tal como no caso anterior, é observável que o desempenho do estimador se degrada para relações SNR inferiores. No caso de não ser introduzido ruído AWGN, o resultado da estimação revela uma elevada precisão, apresentando um desvio médio inferior a 0.3 Hz.

O desempenho do estimador sofre uma degradação claramente visível para valores de SNR de 0 e 2 dB, cujo maior desvio médio surge quando a relação SNR e o CFO introduzido são 2 dB e 4 kHz, respectivamente, com um valor de 99.57 Hz. De forma a enaltecer a precisão deste estimador, quando o CFO introduzido é igual a 500 Hz, o resultado da estimação é bastante próximo do valor ideal, tendo um erro médio na ordem dos 13 Hz (este valor foi obtido através da média dos desvios médios para as diferentes relações SNR). Através dos resultados obtidos nas estimações, é possível verificar-se a veracidade do alargamento da gama dinâmica de estimação, tendo conseguido em alguns casos estimar CFOs até 29 kHz, relembrando que o limite teórico é de 30 kHz.

Tabela 5.8: Estimador de C. Ribeiro (Sincronização temporal ideal): Desvio médio da estimação de CFO, em Hertz, para diferentes níveis SNR.

SNR (dB) \ CFO (Hz)	100	500	2000	4000	7000	12000	15000	20000	25000
0	60.72	22.76	99.86	73.69	76.19	89.17	51.66	78.08	93.11
2	53.36	12.83	12.57	99.57	10.43	82.44	14.71	32.97	5.69
4	9.49	7.14	33.21	38.00	40.45	9.01	40.01	51.33	12.29
6	38.09	10.25	13.67	15.77	35.98	20.55	36.43	18.98	25.46
8	25.14	17.96	16.96	9.68	28.66	11.27	25.02	36.59	7.50
10	20.49	6.84	7.20	15.46	5.51	12.71	6.17	27.47	19.89
sem ruído	0.02	0.10	0.03	0.29	0.26	0.41	0.15	0.15	0.38

Tabela 5.9: Estimador de C. Ribeiro (Sincronização temporal de van de Beek): Desvio médio da estimação de CFO, em Hertz, para diferentes níveis SNR.

SNR (dB) \ CFO (Hz)	100	500	2000	4000	7000
0	274.26	107.13	255.24	46.60	87.72
2	18.4	68.49	61.02	22.85	290.27
4	31.3	98.37	71.06	27.75	64.78
6	51.81	36.31	30.58	16.28	70.68
8	15.0	20.03	36.57	27.71	61.87
10	14.8	0.12	18.13	31.21	17.40
sem ruído	0.25	0.12	0.03	0.31	0.35

5.2.2 Desempenho em Ruído AWGN e canal rádio *outdoor*

Nesta secção são apresentados os resultados do desempenho do estimador de C. Ribeiro quando exposto ao canal rádio *outdoor* (250ns rms delay spread). Devido à falta de tempo, as simulações foram realizadas para um número inferior de relações SNR, tendo sido testados para os casos: sem ruído e SNRs de 8, 4 e 0 dB. A tabela 5.10 apresenta os desvios médios da estimação do CFO quando a sincronização temporal é a ideal. É possível observar que a estimação do CFO, com a introdução do canal rádio, em média afasta-se mais do CFO correto do que quando apenas na presença de ruído AWGN, tabela 5.8. Quando na presença de apenas ruído AWGN, o estimador apresenta um erro máximo de CFO de 99.86 Hz, tabela 5.1, enquanto que com a introdução do canal rádio *outdoor*, este erro chega a ser de 116.10 Hz.

Tabela 5.10: Estimador de C. Ribeiro (Sincronização temporal ideal): Desvio médio da estimação de CFO, em Hertz, para diferentes níveis SNR e um canal rádio *outdoor* (250ns rms delay spread).

SNR (dB) \ CFO (Hz)	100	500	2000	4000	7000	15000
0	77.49	12.74	116.10	105.67	87.45	54.30
4	59.18	7.59	111.42	75.03	95.07	13.67
8	38.32	8.36	63.95	30.70	76.99	36.54
sem ruído	1.12	0.12	0.42	0.41	0.11	0

A tabela 5.11 expõe os resultados do estimador, para as mesmas condições anteriores, com a diferença que a sincronização temporal é realizada pelo estimador de van de Beek. Existe uma clara degradação do desempenho devido a vários fatores, como a sincronização temporal, a perda de tramas devido a estas terem uma baixa energia, e os re-sincronismos. Quanto aos dois últimos fatores, o primeiro faz com que o número de estimações do recetor diminua, uma vez que o resultado do algoritmo de estimação temporal não atinge o *threshold* de decisão da detecção da sequência *Zadoff-Chu*. Relembrando que quanto

maior for o número de estimações, melhor será o resultado da estimação, uma vez que este recorre aos resultados da função custo anteriores para redefinição do CFO candidato. No segundo caso, os dados que chegam ao estimador de CFO não são símbolos corretos, devido a parte destes ser perdida nos re-sincronismos, fazendo com que as posições analisadas pelo estimador possam não conter símbolos piloto.

Tabela 5.11: Estimador de C. Ribeiro (Sincronização temporal de van de Beek): Desvio médio da estimação de CFO, em Hertz, para diferentes níveis SNR e um canal rádio *outdoor*(250ns rms delay spread).

CFO (Hz) \ SNR (dB)	100	500	2000	4000	7000
0	115.04	474.85	1481	430.01	92.65
4	141.54	474.00	530.21	71.03	430.81
8	162.62	438.36	215.60	48.32	265.87
sem ruído	129.52	414.55	7.60	56.49	65.43

5.3 Comparação do desempenho dos estimadores

Nesta secção é feita uma comparação dos resultados expostos nas secções anteriores, tendo como objetivo analisar o desempenho, em termos de precisão, dos estimadores ML de van de Beek e do estimador com base nos pilotos de C. Ribeiro. A tabela 5.12 contém os valores do MSE, em dB, entre os resultados de ambos os estimadores para diferentes níveis SNR, com o estimador de C. Ribeiro a usar sincronização temporal ideal. Como o resultado do MSE é sempre positivo, por convenção, quando a estimação de van de Beek for favorável em relação à de C. Ribeiro, o resultado será mostrado como negativo. Para todos os casos, o estimador de C. Ribeiro tem um desempenho superior ao de van de Beek, para além de permitir o aumento da gama de estimação do CFO, obtendo ganhos entre 6.6 e 39 dB.

A tabela 5.13 também apresenta os valores do MSE, com a diferença de que agora o estimador de C. Ribeiro utiliza a sincronização temporal realizada pelo algoritmo de van de Beek. No caso de relações SNR mais elevadas, era expectável que os valores do MSE fossem idênticos aos obtidos com a sincronização temporal ideal, contudo, existem algumas diferenças consideráveis. Devido à falta de tempo não foi possível esclarecer a causa do erro, não sabendo se este foi proveniente de imprecisões temporais ou erros na estimação do CFO. Para SNR baixos, era expectável que as estimações do CFO piorassem devido à degradação da estimação temporal. Em alguns casos, esta não se degrada e, por vezes, chega a melhorar em relação à estimação que usa a sincronização temporal ideal. Novamente, estas questões não foram analisadas devido à falta de tempo.

Tabela 5.12: MSE, em dB, entre o desvio médio de CFO do estimador de C. Ribeiro (Sincronização temporal ideal) e do estimador de van de Beek, para diferentes níveis SNR.

SNR (dB) \ CFO (Hz)	100	500	2000	4000	7000
0	30.8	38.9	29.7	32.3	7
2	26.2	36.3	35.9	24.1	39.0
4	31.3	33.6	26.0	28.0	12.7
6	6.6	31.0	23.6	30.9	17.2
8	15.0	26.5	23.3	27.9	22.1
10	14.8	23.6	20.0	15.46	29.3
sem ruído	14.5	13.1	11.6	18.9	9.7

Tabela 5.13: MSE, em dB, entre o desvio médio de CFO do estimador de C. Ribeiro (Sincronização temporal ML de van de Beek) e do estimador de van de Beek, para diferentes níveis SNR.

SNR (dB) \ CFO (Hz)	100	500	2000	4000	7000
0	-45.0	11.5	-41.9	36.7	-19.4
2	35.3	19.7	22.7	39.3	-45.6
4	23.5	-32.7	-19.0	30.9	-26.0
6	-21.3	16.7	9.4	30.8	-28.8
8	23.9	25.6	-13.0	16.7	-26.2
10	21.0	26.8	11.5	-15.2	24.8
sem ruído	14.15	13.1	11.6	18.9	9.4

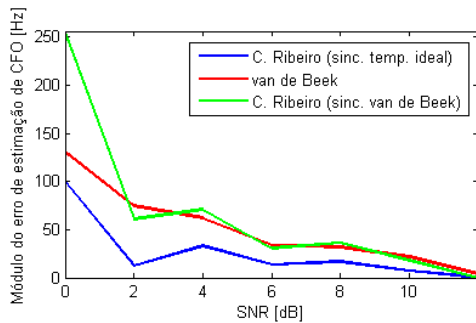
A tabela 5.14 é idêntica à 5.12, diferindo apenas nas condições às quais o sistema está exposto, agora um canal rádio *outdoor* (250ns rms delay spread). Os resultados apresentando surgem da comparação entre as tabelas 5.10 e 5.7, ou seja, dos resultados obtidos pelos estimador C. Ribeiro (sincronização ideal) e do estimador van de Beek (quando se ignoram as diferenças superiores a 1500 Hz). As estimações de estimador de C. Ribeiro revelaram uma maior precisão, resultando em MSEs com valores entre 8.8 e 46.0 dB, quando comparados com as estimações de CFO de van de Beek. Contudo, esta comparação é desigual uma vez que a estimacão de CFO de van de Beek usa a sua própria sincronização temporal.

A figura 5.2a mostra os desvios médios das estimacões de CFO, para 2 kHz sem canal rádio, obtidas pelo estimador de van de Beek e de C. Ribeiro, este último com sincronização temporal ideal e de ML van de Beek. A figura 5.2b mostra novamente estes desvios, agora com canal rádio. Em ambas as situaçoes, o estimador de C. Ribeiro (sincronização ideal) obteve estimacões mais próximas do CFO correto. No caso em que apenas é introduzido ruído AWGN, os resultados das estimacões de C. Ribeiro (sincronização van

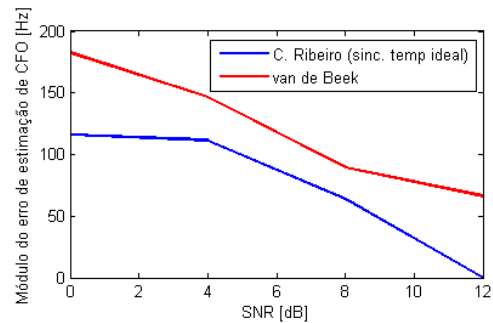
de Beek) e de van de Beek, no geral, tiveram desvios médios próximos.

Tabela 5.14: MSE, em dB, entre o desvio médio de CFO do estimador de C. Ribeiro (Sincronização temporal ideal) e do estimador de van de Beek, para diferentes níveis SNR e um canal rádio *outdoor*(250ns rms delay spread).

SNR (dB) \ CFO (Hz)	100	500	2000	4000	7000
0	44.6	46.0	36.5	35.6	25.8
4	31.1	38.7	21.8	21.3	12.5
8	27.3	36.9	23.3	29.7	8.8
sem ruído	29.9	30.6	36.3	32.6	34.2



(a) Sem canal rádio *outdoor*.



(b) Com canal rádio *outdoor*.

Figura 5.2: Desvios médios das estimações de CFO, para 2 kHz, para diferentes níveis SNR.

Capítulo 6

Conclusão e trabalho futuro

Neste capítulo serão apresentadas as conclusões e o trabalho futuro.

6.1 Conclusão

As conclusões do trabalho estão divididas em dois tópicos principais, o desempenho dos estimadores implementados e a arquitetura de funcionamento do sistema.

6.1.1 Conclusões sobre o desempenho dos estimadores

O último século ficou marcado pela proliferação de sistemas de transmissão *wireless* como LANs [69, 70] e MANs [71]. Os sistemas 4G basearam-se no sucesso obtido pelos recentes standards [9, 69, 70, 71, 72], tendo todos estes como base a tecnologia OFDM. Esta tecnologia é sem dúvida a técnica que ajudará os sistemas 4G a atingirem os requisitos impostos pela norma [73]. O uso de modulações de elevada ordem forçam os receptores a operarem em regimes de baixo SNR, condições inevitáveis nos sistemas 4G. De forma a tirar partido de todo o seu potencial, o sistema necessita de superar o impacto provocado por fatores como: multi-percurso, ruído, interferências e imperfeições do sistema. Os sistemas OFDM são muito sensíveis a erros de sincronismo, devido ao uso de múltiplas sub-portadoras, assim, os algoritmos de sincronização necessitam ter um desempenho perto da perfeição, de forma a não degradar o desempenho do sistema.

Ambos os algoritmos implementados nesta dissertação [1, 2] apresentam uma baixa complexidade na estimação do CFO. O algoritmo de van de Beek [1] recorre à informação redundante contida no CP para proceder à estimação temporal e do CFO. Quando na presença de apenas ruído AWGN, o desempenho do estimador de início de trama foi perfeito para relações SNR superiores a 4 dB, degradando-se para SNRs inferiores, chegando a falhar, no pior caso, até 148 amostras. Quanto à estimação do CFO, teve um

desempenho aceitável, registrando desvios médios de 5 a 105 Hz, para valores positivos de SNR. Com a adição do canal rádio, o desempenho do estimador de início de trama sofre uma degradação, chegando a falhar em média entre 50 a 264 amostras. Quanto à estimação do CFO, os resultados pioraram muito, em parte devido ao resultado de algumas dessas estimações se distanciarem bastante do CFO correto, fazendo com que o desvio médio suba significativamente. Estes desvios mais elevados atingiram em média 12% e, no limite, 35.59% das estimações efetuadas. As simulações permitiram também verificar a limitação na gama de estimação do CFO.

A inclusão de símbolos piloto nas *frames* OFDM tem uma elevada importância apesar do decréscimo inerente da eficiência do sistema. Os símbolos piloto são normalmente utilizados para estimação de canal, obtendo bons resultados quando a transmissão é efetuada em canais com multi-percurso. A tendência é o uso de algoritmos no domínio da frequência, contudo, no domínio do tempo existe a vantagem da energia do CIR estar limitada a um número de amostras reduzido. O algoritmo de C. Ribeiro [2] tira partido do espaçamento constante entre pilotos, N_f , e do facto de este estar relacionado com o tamanho da FFT, N_c . Explorando as propriedades da FFT para separar as componentes dependentes dos pilotos das dos dados, este algoritmo pode ser usado para estimação do canal e do CFO. Todos os símbolos que transportam pilotos são utilizados pelo estimador, não necessitando de *overheads* adicionais como símbolos de sincronismo ou sub-portadoras nulas. Os resultados são bastante melhores que os obtidos na estimação efetuada pelo algoritmo de van de Beek, tornando-o ainda mais adequado para eliminação do CFO nos sistemas OFDM. Para além do aumento de precisão, ficou também visível o aumento claro da gama dinâmica de estimação do CFO, chegando no limite a 29 kHz. A capacidade deste algoritmo de gerar estimações precisas com um número limitado de símbolos OFDM, torna-o apropriado para o uso em sistemas que tenham um funcionamento em modo *burst*. Estas características e o desempenho demonstrado nas simulações, tornam o algoritmo [2] adequado aos *standards* atuais e futuros de comunicações sem fios.

6.1.2 Conclusões sobre o assincronismo do sistema

Existem já diversos trabalhos que implementam cadeias de transmissão OFDM, como por exemplo, [67, 74, 75]. Todos estes são caracterizadas por ter um funcionamento síncrono, tendo já sido referido que à medida que o projeto cresce há um aumento enorme da complexidade. Nesta arquitetura de funcionamento, qualquer alteração realizada num bloco, levará conseqüentemente a que sejam realizadas alterações no sistema para que este continue a funcionar corretamente. Não esquecendo de realçar, que o único elemento que necessita de funcionar a um ritmo constante é ADC/DAC.

O sistema disponibilizado para esta dissertação tem como característica diferenciadora

o facto de ter um princípio funcionamento assíncrono. O fluxo de dados é processado em *bursts* de bloco para bloco, sendo o ADC/DAC o único que funciona com ritmo constante. Esta arquitetura traz sem dúvida inúmeras vantagens, principalmente na alteração ou adição de blocos constituintes do sistema, permitindo a concepção e teste individual de cada bloco. Existe assim a possibilidade de reutilização do *hardware* implementado, sem a necessidade de fazer qualquer tipo de alteração. Não podendo ter só vantagens associadas, este tipo de funcionamento obriga a um acréscimo da complexidade de cada bloco, pela introdução de *buffers* e de máquinas de estados para controlo do processamento.

Em conclusão, as vantagens associadas ao assincronismo são sem dúvida muito significativas. Considero como principal vantagem o aumento na rapidez de desenvolvimento dos blocos, devido a estes poderem ser testados individualmente, e como desvantagem, o facto de serem requeridos mais recursos para a implementação dos componentes extra. Não considerando como desvantagem a complexidade de desenvolvimento destes blocos, uma vez que o único componente alterado de bloco para bloco, para além do processamento específico, é a máquina de estados de controlo deste. Após a otimização dos recursos requeridos no sistema, como por exemplo, otimização do tamanho dos *buffers*, os componentes extra requeridos para implementação do assincronismo não são assim tão significativos. Assim, posso constatar firmemente que as desvantagens associadas ao assincronismo são insignificantes quando comparadas com as vantagens trazidas por este princípio de funcionamento.

6.2 Trabalho futuro

Quanto ao trabalho futuro, deverá proceder-se à implementação do sistema desenvolvido nesta dissertação na FPGA *Xilinx Virtex-VI*. Devendo realizar-se testes de desempenho do sistema no *hardware*, comparando-os com os resultados obtidos nas simulações. Deverá também proceder-se à implementação de outros esquemas de modulação M-QAM e M-QPSK, uma vez que o sistema apenas possui a modulação QPSK/QAM. Relembrando que o algoritmo proposto por C. Ribeiro em [2] permite também fazer a estimação do canal, e uma vez que o *core* do algoritmo já se encontra implementado, poderá no futuro ser alterado para este fim, podendo assim fazer-se uma comparação do desempenho deste com o algoritmo de estimação do canal *Zero Forcing* utilizado. Apesar dos bons resultados demonstrados por este estimador, ficou evidente a influência provocada pela precisão da estimação temporal. Esta influência é mais notória em baixos SNRs, devido à degradação da sincronização temporal por parte do algoritmo ML de van de Beek. De acordo com os resultados no *hardware*, se necessário, deverá proceder-se à implementação de outro algoritmo com maior precisão na estimação temporal. Poderá também alterar-se

a sequência de sincronismo usada, passando a utilizar-se conjuntos de sequências com tamanhos inferiores, sendo estas separadas no recetor recorrendo a filtros.

No recetor deverá ser incluído um *Automatic Gain Control* (AGC) devido à imprevisibilidade do nível de potência do sinal recebido, causada pelos desvanecimentos provocados pelo multi-percurso. Este fenómeno foi bastante visível nas simulações do canal rádio, onde o recetor perdeu várias tramas e se re-sincronizou inúmeras vezes. O AGC será usado para fazer um ajuste dinâmico da potência do sinal recebido, de forma a minimizar os erros [76], podendo este ser implementado de duas formas, *feed-forward* e *feed-backward*. No primeiro caso, a tensão de entrada controla a tensão de saída através do ajuste contínuo do ganho de saída, mantendo assim uma saída constante. Este tipo de AGC tem a vantagem de ser muito estável devido ao facto de não conter nenhum *feedback*. No segundo caso, a tensão de saída estimada é comparada com uma de referência, gerando assim um sinal de erro, sendo este usado para ajustar o sinal de entrada até que o erro seja cancelado. Também deverão ser incluídos códigos de correção de erros FEC, como por exemplo o esquema *reed Solomon* (RS), adotado já em vários *standards*. Estes são facilmente implementados devido os blocos RS e de entrelaçamento de bits se encontrarem na *toolbox* do *Simulink*.

Bibliografia

- [1] J.-J. van de Beek, M. Sandell, and P. Borjesson, “ML estimation of time and frequency offset in OFDM systems,” vol. 45, no. 7, pp. 1800–1805, 1997.
- [2] C. Ribeiro, “Channel and frequency offset estimation schemes for multicarrier systems,” in *PhD Thesis*, 2010.
- [3] J. Palmer, “THE HYBRID ARCHITECTURE PARALLEL FAST FOURIER TRANSFORM (HAPFFT),” Ph.D. dissertation, Aug. 2005.
- [4] “3GPP TS 36.201: Physical channels and modulation (release 8).”
- [5] U. Meyer-Baese, “Digital signal processing with field programmable gate arrays,” in *3rd ed. Springer Publishing Company, Incorporated*, 2007.
- [6] T. S. Rappaport, *Wireless communications: principles and practice*. Prentice Hall PTR, 1996.
- [7] J. D. Parsons, *The Mobile Radio Propagation Channel*. Wiley, Dec. 2000.
- [8] J. Fakatselis, “Criteria for 2.4 GHz PHY comparison of modulation methods,” in *Document IEEE*, 1997.
- [9] “LTE; evolved universal terrestrial radio access (e-UTRA); long term evolution (LTE) physical layer; general description,” in *ETSI Technical Specification TS 136 201 v8.3.0*, Apr. 2009.
- [10] T. Y. Xu Wenbo and H. Bin, “Xilinx ISE design suite 10.x FPGA design,” in *POSTS TELECOM PRESS, first edition*, 2008.
- [11] Y. Hu, “CORDIC-based VLSI architectures for digital signal processing,” vol. 9, no. 3, pp. 16–35, 1992.
- [12] J. Volder, “The CORDIC trigonometric computing technique,” in *IRE Trans. Electron. Compute*, vol. EC-8, 1959, pp. 330–334.

-
- [13] J. Walther, "A united algorithm for elementary functions," pp. 379–385, 1971.
- [14] P. Meher, J. Valls, T.-B. Juang, K. Sridharan, and K. Maharatna, "50 years of CORDIC: algorithms, architectures, and applications," vol. 56, no. 9, pp. 1893–1907, 2009.
- [15] D. Cochrab, "Algorithms and accuracy in the HP-35," in *HewlettPackard Journal*, Jun. 1972, pp. 1–11.
- [16] M. Rice, "Digital communications: A discrete-time approach," in *Earson Education, Inc.*, 2009.
- [17] V. F. K., "Direct digital frequency synthesizers," in *New York, Institute of Electrical and Electronics Engineers*, 1999.
- [18] V. J., "Direct digital synthesizers: Theory, design and applications," in *dissertation for the degree of Doctor of Science in Technology, Finland, Helsinki University of Technology, Espoo*, 2000.
- [19] B.-G. Goldberg, "Digital frequency synthesis demystified," in *New York, McGraw-Hill*, 1984.
- [20] V. J. and H. K., "Direct digital synthesizer: Theory, design and applications," Jul. 2001.
- [21] J. M. S. Winthrop W. Smith, "Handbook of real time FFT," in *IEEE Press*, 1995.
- [22] H. J. Nussbaumer, "Fast fourier transforms and convolution algorithms," in *Springer: Berlin*, 1981.
- [23] J. Cooley and J. Tukey, "An algorithm for the machine calculation of complex fourier series," in *Mathematics of Computation*, vol. 19, 1965, pp. 297–301.
- [24] T. Pollet and M. Moeneclaey, "The effect of carrier frequency offset on the performance of band limited single carrier and OFDM signals," in *Global Telecommunications Conference, 1996. GLOBECOM '96. 'Communications: The Key to Global Prosperity*, vol. 1, 1996, pp. 719–723 vol.1.
- [25] C. Muschallik, "INFLUENCE OF RF OSCILLATORS ON AN OFDM SIGNAL," in *, Proceedings of International Conference on Consumer Electronics, 1995*, 1995, pp. 148–.
- [26] T. Pollet, M. Van Bladel, and M. Moeneclaey, "BER sensitivity of OFDM systems to carrier frequency offset and wiener phase noise," vol. 43, no. 234, pp. 191–193, 1995.

- [27] L. Tomba, "On the effect of wiener phase noise in OFDM systems," vol. 46, no. 5, pp. 580–583, 1998.
- [28] L. Piazzo and P. Mandarini, "Analysis of phase noise effects in OFDM modems," vol. 50, no. 10, pp. 1696–1705, 2002.
- [29] L. Wei and C. Schlegel, "Synchronization requirements for multi-user OFDM on satellite mobile and two-path rayleigh fading channels," vol. 43, no. 234, pp. 887–895, 1995.
- [30] T. Pollet and M. Moeneclaey, "Synchronizability of OFDM signals," in , *IEEE Global Telecommunications Conference, 1995. GLOBECOM '95*, vol. 3, 1995, pp. 2054–2058 vol.3.
- [31] H. Nogami and T. Nagashima, "A frequency and timing period acquisition technique for OFDM systems," in , *Sixth IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 1995. PIMRC'95. Wireless: Merging onto the Information Superhighway*, vol. 3, 1995, pp. 1010–.
- [32] T. Schmidl and D. Cox, "Robust frequency and timing synchronization for OFDM," vol. 45, no. 12, pp. 1613–1621, 1997.
- [33] P. H. Moose, "A technique for orthogonal frequency division multiplexing frequency offset correction," vol. 42, no. 10, pp. 2908–2914, 1994.
- [34] U. Lambrette, M. Speth, and H. Meyr, "OFDM burst frequency synchronization by single carrier training data," vol. 1, no. 2, pp. 46–48, 1997.
- [35] M. Morelli and U. Mengali, "An improved frequency offset estimator for OFDM applications," in *Communication Theory Mini-Conference, 1999*, 1999, pp. 106–109.
- [36] R. S. C. B. Yang, K. B. Letaief and Z. Cao, "Timing recovery for OFDM transmission," vol. 18, no. 11, p. 2278–2291, Nov. 2000.
- [37] J. H. P. H. K. Song, Y. H. You and Y. S. Cho, "Frequency-offset synchronization and channel estimation for OFDM-based transmission," vol. 4, no. 3, p. 95–97, Mar. 2000.
- [38] A. Coulson, "Maximum likelihood synchronization for OFDM using a pilot symbol: analysis," vol. 19, no. 12, pp. 2495–2503, 2001.
- [39] —, "Maximum likelihood synchronization for OFDM using a pilot symbol: algorithms," vol. 19, no. 12, pp. 2486–2494, 2001.

-
- [40] P. M. T. Keller, L. Piazzo and L. Hanzo, “Orthogonal frequency division multiplex synchronization techniques for frequencyselective fading channels,” vol. 19, no. 6, pp. 999–1008,, Jun. 2001.
- [41] V. K. B. H.Minn and K. B. Letaief, “A robust timing and frequency synchronization for OFDM systems,” vol. 2, no. 4, p. 822–839, Jul. 2003.
- [42] K. Shi and E. Serpedin, “Coarse frame and carrier synchronization of OFDM systems: A new metric and comparison,” vol. 3, no. 4, p. 1271–1284, Jul. 2004.
- [43] “IEEE 802.11a: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: High-speed physical layer in the 5GHz band,” in *IEEE Std.*, Sep. 1999.
- [44] “IEEE 802.11g: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: Amendment 4: Further higher data rate extension in the 2.4 GHz band,” Jun. 2003.
- [45] “IEEE P802.11n/D1.10 wireless LAN medium access control (MAC) and physical layer (PHY) specifications: Enhancements for higher throughput,” in *IEEE Std.*, Feb. 2007.
- [46] A. G. C. Ribeiro, “Direct time-domain channel impulse response estimation for OFDM-based systems,” in *Proc. IEEE Vehicular Technology Conference*, Oct. 2007, p. 1082 – 1086.
- [47] —, “Mitigating CFO in OFDM systems by exploring the symbol properties.”
- [48] E. Zhou, X. Zhang, H. Zhao, and W. Wang, “Synchronization algorithms for MIMO OFDM systems,” in *2005 IEEE Wireless Communications and Networking Conference*, vol. 1, 2005, pp. 18–22 Vol. 1.
- [49] R. Frank and S. Zadoff., “Phase shift pulse codes with good correlation properties,” *IRE transactions on information theory*, vol. IT-8, Oct. 1962, pp. 381–382.
- [50] D. Chu, “Polyphase codes with good periodic correlation properties (corresp.),” vol. 18, no. 4, pp. 531–532, 1972.
- [51] M. Sandell, “Design and analysis of estimators for multicarrier modulation and ultrasonic imaging,” 1996.
- [52] N. Lashkarian and S. Kiaei, “Class of cyclic-based estimators for frequency-offset estimation of OFDM systems,” vol. 48, no. 12, p. 2139–2149, Dec. 2000.

- [53] U. M. M. Morelli, A. Andrea, “Feedback frequency synchronization for OFDM applications,” vol. 5, no. 1, p. 28–30, Jan. 2001.
- [54] F. Classen and H. Meyr, “Frequency synchronization algorithms for OFDM systems suitable for communication over frequency selective fading channels,” in *Proc. IEEE Vehicular Technology Conf.*, vol. 3, Aug. 1994, p. 1655–1659.
- [55] F. Daffara and O. Adami, “A novel carrier recovery technique for orthogonal multi-carrier systems,” p. 323–334, Aug. 1996.
- [56] J. Lei and T. Ng, “A consistent OFDM carrier frequency offset estimator based on distinctively spaced pilot tones,” vol. 3, no. 2, p. 588–599, Mar. 2004.
- [57] G. G. Y. Yao, “Blind carrier frequency offset estimation in SISO, MIMO and multiuser OFDM systems,” vol. 53, no. 1, p. 173 – 183, Jan. 2005.
- [58] M. Z. U. Tureli, H. Liu, “OFDM blind carrier offset estimation: ESPRIT,” vol. 48, no. 9, p. 1459 – 1461, Sep. 2000.
- [59] H. L. U. Tureli, “Blind carrier synchronization and channel identification for OFDM communications,” in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 6, May 1998, p. 3509 – 3512.
- [60] R. R. M. Luise, M. Marselli, “Low-complexity blind carrier frequency recovery for OFDM signals over frequency-selective radio channels,” vol. 50, no. 7, p. 1182 – 1188, Jul. 2002.
- [61] H. Bölcskei, “Blind high-resolution uplink synchronization of OFDM-based multiple access schemes,” in *Proc. IEEE Workshop Signal Processing Advances in Wireless Communications*, May 1999, p. 166 – 169.
- [62] H. Liu and U. Tureli, “A high-efficiency carrier estimator for OFDM communications,” vol. 2, no. 4, pp. 104–106, 1998.
- [63] M. Majó, “Design and implementation of an OFDM-based communication system for the GNU radio platform,” in *Master Thesis*, Dec. 2009.
- [64] A. Marwanto, M. A. Sarijari, N. Fisal, S. K. S. Yusof and R. and A. Rashid, “Experimental study of OFDM implementation utilizing GNU radio and USRP – SDR,” in *Proc. of the IEEE 9th Malaysia International Conference on Communications*, Dec. 2009, pp. 132–135.
- [65] J. Garcia and R. Cumplido, “On the design of an FPGA-based OFDM modulator for IEEE 802.11a,” in *2nd International Conference on Electrical and Electronics Engineering*, Sep. 2005, pp. 114–117.

-
- [66] E. J. Garcia and R. Cumplido, “On the design of an FPGA-based OFDM modulator for IEEE 802.16-2004,” in *2005 International Conference on Reconfigurable Computing and FPGAs*, 2005, pp. 22–25.
- [67] A. Dutta, “Reconfigurable OFDM receiver for next generation wireless mesh,” in *Master Thesis*, 2008.
- [68] A. G. C. R. Tiago Pereira, Manuel Violas and J. Lourenço, “Real time FPGA based testbed for OFDM development with ML synchronization,” in *ICSNC 2012 : The Seventh International Conference on Systems and Networks Communications*, 2012.
- [69] “Local and metropolitan area networks – specific requirements – part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications,” in *IEEE Std. 802.11, second edition*, Aug. 2005.
- [70] “Local and metropolitan area networks – specific requirements – part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications – amendment 4: Further higher data rate extension in the 2.4 GHz band,” in *IEEE Std. 802.11g-2003*, Jun. 2003.
- [71] “Local and metropolitan area networks – specific requirements – part 16: Air interface for broadband wireless access systems,” in *IEEE Std. 802.16-2009*, May 2009.
- [72] “Digital video broadcasting (DVB); framing structure, channel coding and modulation for digital terrestrial television,” in *ETSI Std. ETS 300 744 v1.6.1*, Jan. 2009.
- [73] “Framework and overall objectives of the future development of IMT-2000 and systems beyond IMT-2000,” in *ITU-R Recommendation M.1645*, Jun. 2003.
- [74] J. L. Buthler, T. Jessen, M. Buhl, and R. Simonsen, “Turbo codes and OFDM implementation,” 2011.
- [75] K. Sobaihi, “Design and verification of SC-FDE/OFDM communication schemes for millimetre-wave wireless system using FPGA technologies,” in *PhD Thesis*, 2012.
- [76] L. G. D. Qiang, J. Meilong and S. Ning, “ALL-digital AGC in CDMA base station receiver,” in *Communication Technology Proceedings, 2003. ICCT 2003. International Conference on*, vol. 2, 2003, pp. 1037–1041.