

**Spectrum Optimisation in Wireless
Communication Systems: Technology
Evaluation, System Design and Practical
Implementation**

A thesis submitted for the degree of Doctor of Engineering

(Eng.D)

by

Ryan Costadinos Grammenos

Communications and Information Systems Research Group

Department of Electronic and Electrical Engineering

University College London

March 2013

Statement of Originality

I, Ryan Costadinos Grammenos confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Signed:

Date:

In loving memory of my grandparents, Margaret, Raymond, Eleni and Costadinos.

It is difficult for one to complete a project without doing anything wrong, however, it is difficult even if one creates something foolproof for it not to fall in the hands of an unfair judge.

Socrates (469 BC - 399 BC)

Let the future tell the truth, and evaluate each one according to his work and accomplishments. The present is theirs; the future, for which I have really worked, is mine.

Nikola Tesla (1856 - 1943)

Patience is bitter, but its fruit is sweet.

Jean-Jacques Rousseau (1712 - 1778)

Abstract

This thesis explores novel communication techniques which promise to meet the capacity requirements in next generation networks. The study commences with an overview of research trends in current and future wireless and mobile systems. Two key technology enablers are then examined in more detail, namely Cognitive Radio (CR) and Spectrally Efficient Frequency Division Multiplexing (SEFDM).

The first part of this thesis proposes the use of traffic prediction in CR systems to improve the Quality of Service (QoS) for CR users. This work presents a generic selective opportunistic spectrum access framework allowing CR users to capture a frequency slot in an idle channel occupied by licensed users. This is achieved by using CR to sense and select target spectrum bands combined with traffic prediction to determine the optimum channel-sensing order.

The second part of this thesis considers the design, hardware implementation and performance evaluation of a novel multi-carrier modulation technique termed SEFDM, which improves spectrum efficiency at the expense of receiver complexity. The key challenge that arises in SEFDM systems is the self-created interference which complicates the design of receiver architectures. Previous work has focused on the mathematical modelling and optimisation of sophisticated detection algorithms, however, these suffer from an impractical computational complexity. Consequently, the aim of this work is two-fold; first, to reduce the complexity of existing algorithms to make them better-suited for application in the real world; second, to develop hardware prototypes to assess the feasibility of employing SEFDM in practical systems. Initially, a thorough analysis of the impact of fixed-point and sampling effects on the performance of SEFDM is carried out to identify the optimum design parameters and implementation trade-offs. This analysis is followed by the design and implementation of linear detection techniques, such as Zero Forcing (ZF) and Truncated Singular Value Decomposition (TSVD), using Field Programmable Gate Arrays (FPGAs). The performance of these FPGA based linear receivers is evaluated in terms of throughput and resource utilisation while their Bit Error Rate (BER) probability is compared to theoretical predictions. Finally, variants of the Sphere Decoding (SD) algorithm are investigated to ameliorate the error performance of SEFDM systems with targeted reduction in complexity. The Fixed Sphere Decoding (FSD) algorithm is implemented on a Digital Signal Processor (DSP) to measure its computational complexity. Modified sorting and decomposition strategies are then applied to this conventional FSD algorithm offering trade-offs between execution speed and BER.

Acknowledgements

It is always difficult to thank all the people who contribute to making such an odyssey a success. From one-off encounters to lifelong friends and family, every interaction has helped me overcome the highs and lows experienced throughout this journey. So first, here's to all these people.

I would like to thank my supervisor, Professor Izzat Darwazeh who willingly took me under his wing and guided me on this long voyage ensuring that I would eventually reach my destination. Despite the storms endured along the way, Izzat's experience and advice always served as a great source of inspiration.

I would also like to express my gratitude to all my colleagues and friends in the 804 and the once-upon-a-time 1105 research labs for their support and stimulating discussions over the years. In particular, Dr Ioannis Kanaras and Dr Safa Ahmed for explaining the SEFDM detection principles; Mr Marcus Perrett (now Dr Perrett) for sharing his expertise on FPGA design and for his collaboration on the joint ICC 2012 paper; Dr Guangxiang Yuan for his invitation to collaborate in the field of Cognitive Radio which led to the publication of two joint papers in PIMRC 2009 and the Transactions on Vehicular Technology 2010; Mr Tongyang Xu for his collaboration on the joint ICT 2013 paper and for continuing the hardware implementation of SEFDM; Mr George Smart, Mr Ali Anvari and Dr Manoj Thakur for comments on this thesis; and finally to Mr Spyridon Papadopoulos for being a brilliant listener and for his technical and moral support, especially during the last critical months of this journey. I extend these thanks to Mr Paul McKenna for his superb administration of the Engineering Doctorate (EngD) programme.

I am grateful to the UK Engineering and Physical Sciences Research Council (EPSRC) and British Telecom (BT) for funding this research work through the EngD programme. Xilinx's donation through the Xilinx University Program of the FPGA devices used in the designs of Chapters 6 and 7 is also gratefully acknowledged.

Last but not least, I am indebted to my parents for their endless support and extraordinary patience and without whom it would have been impossible to achieve my dreams and goals. In addition, I would like to thank my sister, relatives and friends located at different corners of the globe who have always been there for good times and difficult times. A special thanks goes to my partner, Demetria, for the love and happiness she brings to my life and for always standing by me and giving me hope when I need her the most.

May life bring them the health and happiness they deserve for the motivation, encouragement, support and friendship they have so kindly offered.

Contents

List of Figures	12
List of Tables	19
List of Abbreviations	21
List of Symbols and Operators	30
I Next Generation Communication Technologies	32
1 Introduction	33
1.1 Thesis Outline	39
1.2 Main Contributions	45
1.3 List of Publications	46
2 Multi-carrier Communication Techniques	49
2.1 Multiple Antenna Technologies	50
2.1.1 Multiple Input Multiple Output	50
2.1.2 Cyclic Delay Diversity	52
2.1.3 Other Technologies	54
2.2 OFDMA and SC-FDMA	55
2.3 OFDM	64
2.3.1 Basic Principles	65

2.3.2	Mitigating ICI and ISI	68
2.3.3	Parallelism, Orthogonality and the DFT	70
2.4	Conclusions	78
3	Cognitive Radio and Dynamic Spectrum Access	81
3.1	Cognitive Radio Basics	82
3.2	Cyclostationarity Based Detection	85
3.2.1	Principles of Cyclostationarity	86
3.2.2	Cyclostationary Signatures	87
3.3	Cognitive Radio Modulation Techniques	89
3.3.1	Cognitive Radio Testbeds	92
3.4	Improving Cognitive Radio Spectrum Access using Traffic Prediction . .	93
3.4.1	Selective Opportunistic Spectrum Access Framework	94
3.4.2	Numerical Results	100
3.5	Conclusions	105
II	Spectrally Efficient Multi-carrier Systems	107
4	SEFDM: Principles and Challenges	108
4.1	The Challenge - Spectrum Efficiency	109
4.1.1	Time-Frequency Packing	112
4.2	SEFDM System Description	117
4.2.1	SEFDM Transmission using the DFT	120
4.2.2	SEFDM Demodulation	125
4.3	Self-created Interference and Matrix Ill-conditioning	128
4.4	Sampling Issues and Spectral Leakage in DFT based SEFDM Systems .	134
4.5	Discussion on the Complexity and Performance of SEFDM Modems . .	141
4.6	Conclusions	144

5	SEFDM Detection and Practical Considerations	147
5.1	SEFDM Detection Techniques	148
5.1.1	Linear Detection	151
5.1.2	Iterative Detection	155
5.1.3	Complexity Order and Implementation Aspects of Linear and Iterative Detection	157
5.1.4	Maximum Likelihood and Sphere Decoding	160
5.2	SEFDM Performance Gains through Oversampling	162
5.2.1	Noise Model and Simulation Methodology	166
5.2.2	Numerical Results	168
5.3	Influence of Finite Word Lengths on the Performance of SEFDM Systems	178
5.3.1	Floating-point to Fixed-point Conversion	179
5.3.2	Quantisation Errors	182
5.3.3	FPGA Modelling and Simulation	184
5.3.4	Numerical Results	196
5.4	Conclusions	202
6	Implementation of Linear Detection Techniques	204
6.1	FPGA Preliminaries	205
6.2	Hardware Platform and Prototyping Methodology	209
6.3	FPGA Design and Implementation of ZF and TSVD	211
6.3.1	Design Considerations	213
6.3.2	Implementation and Real-time Verification	221
6.3.3	Numerical Results	224
6.4	Discussion of Design Architecture	233
6.4.1	Optimisation Techniques	234
6.5	Conclusions	237

7	Implementation of Tree-search Algorithms	240
7.1	Algorithm Principles	241
7.1.1	The Sphere Decoding Algorithm	242
7.1.2	The Fixed Sphere Decoding Algorithm	250
7.2	Joint FPGA-DSP Platform	255
7.3	DSP Evaluation	258
7.3.1	Computational Complexity	259
7.3.2	Execution Time	261
7.3.3	Error Performance	262
7.4	Results from the FPGA Implementation of the FSD Algorithm	267
7.4.1	Implementation Aspects	268
7.4.2	Numerical Results	269
7.5	Discussion on Algorithm and Circuit Optimisation	271
7.6	Conclusions	274
8	Conclusions	276
8.1	Thesis Key Achievements	279
8.2	Ongoing Related Work	280
8.3	Proposals for Future Work	281
	Appendix A SEFDM Transceiver MATLAB GUI Datasheet	285
	Appendix B Joint FPGA-DSP Platform Aspects and Operation	291
B.1	FPGA Design Cycle	291
B.2	System Description	295
B.3	Functional Simulation and Real-time Verification	296
B.3.1	Control and Synchronisation	298
B.3.2	Data Exchange	303

Appendix C Source Code	308
C.1 VHDL Code	308
C.2 C Code	309
List of References	311

List of Figures

1.1	Focus of Chapters 1 and 2 (OFDM, MIMO).	42
1.2	Focus of Chapter 3 (CR).	43
1.3	Focus of Chapters 4-7 (SEFDM).	44
2.1	MISO transmitter employing Delay Diversity.	52
2.2	MISO transmitter employing Cyclic Delay Diversity.	53
2.3	Delay Diversity vs. Cyclic Delay Diversity.	54
2.4	Sub-carrier mapping and grouping schemes.	56
2.5	Signal chains for OFDM, OFDMA and SC-FDMA.	57
2.6	OFDMA vs. SC-FDMA data symbol transmission (four sub-carriers). .	58
2.7	CCDF of PAPR for OFDMA for a different number of used sub-carriers ($N_{IFFT} = 512$, 4-QAM).	59
2.8	Distribution of PAPR for OFDMA for a different number of used sub- carriers ($N_{IFFT} = 512$, 16-QAM).	60
2.9	CCDF of PAPR for different SC-FDMA sub-carrier allocation schemes and constellation sizes ($N_{IFFT} = 256$).	60
2.10	Time domain signals for IFDMA, LFDMA and DFDMA ($N_{IFFT} = 256$). .	61
2.11	CCDF of PAPR for IFDMA, LFDMA and DFDMA with RC pulse shap- ing ($N_{IFFT} = 256$, 16-QAM).	62
2.12	CCDF of PAPR for IFDMA with different RC and RRC pulse shaping ($N_{IFFT} = 256$, 16-QAM).	62

2.13	CCDF comparison of SC-FDMA ($\beta_{RRC} = 0.5$) against OFDMA ($N_{IFFT} = 256$, 16-QAM).	63
2.14	Conventional FDM vs. OFDM.	65
2.15	Time-frequency representation of OFDM sub-carriers.	66
2.16	FFT bin locations in the absence and presence of ICI.	67
2.17	A cyclically extended OFDM symbol.	68
2.18	Architecture of an OFDM transceiver employing a bank of modems.	70
2.19	Spectrum of an OFDM signal and (a fraction of) its underlying sub-carriers in theory.	72
2.20	Power spectrum of a modulated OFDM signal and its underlying sub-carriers in practice.	73
2.21	Simplified diagram of a practical OFDM system.	75
2.22	Individual sub-carriers (top) and subsequent OFDM signal (bottom) for unmodulated (left) and modulated (right) scenarios ($N = 128$).	77
3.1	Selective spectrum sensing and access cycle.	94
3.2	PLR performance for increasing threshold values ($N = 20$).	102
3.3	PLR performance for an increasing number of primary channels ($S = 4$).	103
3.4	Throughput for an increasing number of primary channels ($\frac{t_s+t_h}{T_s} = 0.05$).	104
3.5	Throughput in each time slot for increasing sensing and handoff time ($N = 20$).	104
4.1	Timeline of spectrally efficient concepts and non-orthogonal communication techniques.	112
4.2	SEFDM vs. OFDM spectral efficiency.	114
4.3	SEFDM on the bandwidth efficiency plane.	115
4.4	Methods for profiting from SEFDM spectral efficiency gains.	118
4.5	Conceptual architecture for an SEFDM transceiver using a bank of modems.	119
4.6	Spectrum of an SEFDM signal for different values of α .	120

4.7	DFT based SEFDM transmitter types.	124
4.8	Example illustrating the concept of ill-conditioning.	129
4.9	Condition number of sub-carriers correlation matrix for different values of N and α	131
4.10	Amplitude of $\mathbf{C}_{m,n}$ in dB ($N = 32, \alpha \approx 0.67$).	132
4.11	Amplitude of pruned $\mathbf{C}_{m,n}$ in dB ($N = 32, \alpha \approx 0.67$).	132
4.12	Amplitude of $\mathbf{C}_{m,n}^{-1}$ in dB ($N = 32, \alpha \approx 0.67$).	133
4.13	Amplitude of $\mathbf{C}_{m,n}^{-1}$ for different values of N with $\alpha = 0.8$	133
4.14	Magnitude response of OFDM for different oversampling factors.	136
4.15	16-point IDFT and 20-point IDFT on the unit circle.	137
4.16	20-point IDFT versus 16-point SEFDM sub-carriers matrix.	138
4.17	OFDM and SEFDM signals in the time domain.	139
5.1	Practical SEFDM transceiver.	149
5.2	SEFDM demodulation and detection techniques.	150
5.3	SEFDM detection employing linear or iterative techniques.	151
5.4	Amplitude of $\mathbf{C}_{m,n}^{-1}$ (left) and $\mathbf{C}_{\xi_{m,n}}$ (right) ($N = 16, \alpha = 0.8$).	156
5.5	Condition number $\kappa(C)$ for increasing oversampling factors and different values of N and α	163
5.6	BER of different detection schemes for BPSK with $N = 16, \alpha = 0.8$ and $\rho = 2$	170
5.7	BER of different detection schemes for 4-QAM with $N = 16, \alpha = 0.8$ and $\rho = 2$	171
5.8	BER of matched filtering for BPSK with varying N ($\alpha = 0.5$ and $\rho = 1$).	172
5.9	BER of matched filtering for BPSK with increasing ρ ($\frac{E_b}{N_0} = 8dB$).	173
5.10	BER of different detection schemes for varying ρ ($N = 8, \alpha = 0.8$).	174
5.11	BER of different detection schemes for varying N ($\alpha = 0.8, \frac{E_b}{N_0} = 8dB$).	175
5.12	BER of different detection schemes for varying α with $N = 8$ ($\frac{E_b}{N_0} = 8dB$).	175
5.13	BER of different detection schemes for varying α with $N = 16$ ($\frac{E_b}{N_0} = 8dB$).	176

5.14	BER of hybrid and non-hybrid detection schemes for $\rho = 1$ (left sub-plot) and $\rho = 2$ (right sub-plot) ($N = 16$, $\alpha = 0.8$, FSD tree width $W = 16$).	178
5.15	Simulation based search to identify optimum bit widths.	186
5.16	Scaling and rounding options at different stages of a Xilinx FFT IP core.	188
5.17	Illustration of the unbounded dynamic range (DR) for \mathbf{C}^{-1} and \mathbf{C}_ξ .	191
5.18	Maximum values and minimum required bits for \mathbf{C}^{-1} .	192
5.19	Maximum values and minimum required bits for \mathbf{C}_ξ .	193
5.20	Distribution of the matrix elements pertaining to \mathbf{C}^{-1} and \mathbf{C}_ξ ($N = 16$, $\alpha = 0.8$).	193
5.21	Comparison of the scaled distribution of \mathbf{C}^{-1} and \mathbf{C}_ξ .	194
5.22	Comparison of the scaled distribution for different bits for \mathbf{C}_ξ .	195
5.23	Comparison of the scaled distribution for different bits for \mathbf{C}^{-1} .	195
5.24	Bit precision requirements to yield zero errors for $N = 8$.	198
5.25	Bit precision requirements to yield zero errors for BPSK.	198
5.26	Error performance due to quantisation noise for BPSK with $N = 16$.	199
5.27	Error performance due to quantisation noise for BPSK with $N = 8$ and $\rho = 1$.	200
5.28	Error performance due to quantisation noise for BPSK with $N = 8$ and $\rho = 2$.	201
5.29	Error performance due to quantisation noise for 4-QAM ($N = 8$, $\alpha = 0.5$, $\rho = 2$).	201
6.1	Outline of SEFDM testbed transceiver.	210
6.2	Top-level system diagram.	212
6.3	Hardware operation of the TSVD-SEFDM detector.	214
6.4	System block diagram for the FPGA based linear detector implementation.	222
6.5	BER performance of the TSVD detector for a BPSK-SEFDM system with different sizes and values of bandwidth compression α .	225

6.6	BER performance of the TSVD detector for a 4-QAM-SEFDM system with different sizes and a bandwidth compression of $\alpha=0.9$	226
6.7	BER performance of the floating-point TSVD vs. ZF detector for a BPSK-SEFDM system with 16 sub-carriers at different values of α	227
6.8	BER performance of the FPGA based TSVD detector employing BPSK for a varying number of sub-carriers N and bandwidth compression α . .	228
6.9	BER performance of the FPGA based TSVD detector employing 4-QAM for a varying number of sub-carriers at a fixed compression $\alpha = 0.81$. . .	229
6.10	Using multiple clock domains to allow block reuse in FPGA systems. . .	235
6.11	Unique elements of \mathbf{C} , \mathbf{C}^{-1} and \mathbf{C}_ξ versus the total number of their elements for different values of N	236
7.1	Sphere search constrained by the radius g	242
7.2	Search tree for a SD with $N = 3$ and BPSK modulation.	243
7.3	Schnorr-Euchner enumeration vs. Fincke-Pohst enumeration.	245
7.4	Conceptual SD architecture.	248
7.5	SEFDM receiver combining linear detection with fixed SD.	251
7.6	Flowchart of the FSD algorithm for the detection of SEFDM signals. . .	252
7.7	Spanning tree of Shortest-Path FSD (left) vs. Sort-Free FSD (right) ($N = 2$, $W = 2$).	254
7.8	System block diagram for the hybrid TSVD-FSD hardware detector implementation.	257
7.9	Computational complexity for increasing tree depth.	260
7.10	BER performance for the different FSD variants (4-QAM, $\alpha = 0.8$, $N = 16$ and $W = 16$).	262
7.11	Comparison of the FSD detection schemes for increasing complexity (4-QAM, $\alpha = 0.8$ and $N = 16$).	263
7.12	Comparison of the FSD detection schemes for different system sizes (4-QAM, $\alpha = 0.8$, $W = 16$).	264

7.13	Comparison of the FSD detection schemes at different values of α (4-QAM, $N = 16$, $W = 16$).	265
7.14	Error performance of analytical, modelled and experimental FSD detectors (4-QAM, $\alpha = 0.8$, $N = 8$, $W = 16$).	265
7.15	BER performance of the joint FPGA-DSP, TSVD-SF-FSD hardware detector for a varying number of sub-carriers N (4-QAM, $\alpha = 0.8$, $W = 16$).	266
7.16	BER performance of the joint FPGA-DSP, TSVD-SF-FSD hardware detector against E_b/N_o (4-QAM, $\alpha = 0.8$, $N = 8$, $W = 16$).	266
7.17	BER performance of the real-time FPGA based SF-FSD and SP-FSD detectors for a 4-QAM SEFDM system with $N = 16$, $\alpha = 0.8$ and $W = 16$	272
A.1	Area allowing key parameters, such as the number of sub-carriers N , the frequency spacing α and the modulation scheme to be configured.	286
A.2	Area allowing the type of signal and level of noise to be configured.	286
A.3	Message box illustrating a summary of the general choices made.	287
A.4	Message box illustrating a summary of the advanced options selected.	287
A.5	Pop-up window allowing a specific set of results to be examined.	287
A.6	Pop-up window allowing specific figures to be displayed.	288
A.7	Example of a time domain OFDM signal generated using this GUI.	289
A.8	Example of an OFDM signal's magnitude response generated using this GUI.	289
A.9	GUI startup screen.	290
B.1	Design flow.	292
B.2	Development platform by Sundance Multiprocessor Technology Ltd.	295
B.3	System block diagram (from vendor datasheet).	296
B.4	Low-level architecture of the developed FPGA based SEFDM receiver.	297
B.5	Code snapshot of the commands manually programmed in the FPGA for system control and data exchange with the DSP.	298

B.6	Commands issued by the DSP in real time to verify system synchronisation and initiate data processing.	299
B.7	DSP output during system synchronisation.	300
B.8	Verification of FPGA control and system synchronisation through functional simulation.	301
B.9	Verification of FPGA control and system synchronisation during real-time operation.	302
B.10	Values of the received statistics vector \mathbf{R} acquired from the functional simulation (bottom) and real-time operation (top) of an FPGA based SEFDM receiver employing TSVD.	303
B.11	Results acquired from the simulation of SEFDM employing TSVD in MATLAB.	304
B.12	Verification of the data transfer of \mathbf{R} from the FPGA to the DSP during functional simulation (bottom) and real-time operation (top).	305
B.13	Verification of the values and data transfer of the unconstrained $\tilde{\mathbf{s}}_{TSVD}$ and constrained $\hat{\mathbf{s}}_{TSVD}$ symbol estimates during functional simulation (bottom) and real-time operation (top).	306
B.14	Results captured from the FPGA data processing operations, subject to additional decoding operations executed within the DSP.	307

List of Tables

1.1	Comparison of Xilinx (Altera) device families.	38
2.1	Simulation parameters for OFDMA and SC-FDMA.	58
5.1	Ratio $\frac{\lambda_{max}(\mathbf{D})}{\lambda_{min}(\mathbf{D})}$ of the projections matrix \mathbf{D} for different N , α and ρ	164
5.2	Condition number $\kappa(\mathbf{C})$ of matrix \mathbf{C} for different N , α and ρ	165
6.1	DFT core versus FFT core.	216
6.2	Comparison of features for different Xilinx Virtex FPGA devices.	230
6.3	Resource utilisation for the TSVD-SEFDM hardware design using an XC4VFX12 chip.	230
6.4	Theoretical maximum data rate for different system sizes and modulation orders.	231
6.5	Resource utilisation for the complete FPGA based TSVD-SEFDM re- ceiver system.	232
6.6	FPGA resource requirements for each TSVD receiver block.	232
6.7	Latency for the different FPGA blocks.	233
7.1	Total number of operations per level for the Shortest-Path and Sort-Free FSD detectors.	259
7.2	Number of compare-select operations for the Shortest-Path and Sort- Free FSD detectors for increasing tree widths.	261

7.3	Comparison of the execution time for the two FSD variants for an increasing system size and tree width.	261
7.4	Comparison of an XC2V6000 to an XC6VLX240T	269
7.5	Estimated resource utilisation and clock performance for the SF-FSD detector after synthesis.	270
7.6	Estimated resource utilisation and clock performance for the SP-FSD detector after synthesis.	270
7.7	True resource utilisation and clock performance of SF-FSD and SP-FSD after implementation.	271
C.1	VHDL Code Hierarchy and File Description.	309
C.2	C Code Hierarchy and File Description.	310

List of Abbreviations

1D	one-dimensional
2D	two-dimensional
2G	2nd Generation
3G	3rd Generation
3GPP	3rd Generation Partnership Project
4G	4th Generation
A/D	Analogue-to-Digital
ACI	Adjacent Channel Interference
ADSL	Asymmetric Digital Subscriber Line
AED	Accumulated Euclidean Distance
AI	Artificial Intelligence
AMC	Adaptive Modulation and Coding
ASCII	American Standard Code for Information Interchange
ASIC	Application Specific Integrated Circuit
ASK	Amplitude Shift Keying
AWGN	Additive White Gaussian Noise
BCD	Binary-Coded Decimal

BER	Bit Error Rate
BPSK	Binary Phase Shift Keying
BRAM	Block Random Access Memory
BS	Base Station
CAF	Cyclic Autocorrelation Function
CCDF	Complementary Cumulative Distribution Function
CDD	Cyclic Delay Diversity
CDMA	Code Division Multiple Access
CDP	Cyclic Domain Profile
CDS	Channel-dependent Scheduling
CLB	Configurable Logic Block
CP	Cyclic Prefix
CPLD	Complex Programmable Logic Device
CPU	Central Processing Unit
CR	Cognitive Radio
CSTD	Cyclic Shift Transmit Diversity
D/A	Digital-to-Analogue
DAB	Digital Audio Broadcasting
DCM	Digital Clock Manager
DD	Delay Diversity
DFDMA	Distributed Frequency Division Multiple Access
DFE	Decision Feedback Equalisation
DFT	Discrete Fourier Transform

DIF	Decimation In Frequency
DIT	Decimation In Time
DMT	Discrete Multi-tone
DSA	Dynamic Spectrum Access
DSP	Digital Signal Processor
DSSS	Direct Sequence Spread Spectrum
DVB	Digital Video Broadcasting
EMIF	External Memory Interface
FCSD	Fixed Complex Sphere Decoder
FDE	Frequency Domain Equalisation
FDM	Frequency Division Multiplexing
FEC	Forward Error Correction
FF	Flip-Flop
FFR	Fractional Frequency Reuse
FFT	Fast Fourier Transform
FIFO	First In First Out
FOFDM	Fast Orthogonal Frequency Division Multiplexing
FOMS	Frequency Overlapped Multi-carrier System
FPGA	Field Programmable Gate Array
FSD	Fixed Sphere Decoding
FSM	Finite State Machine
FTN	Faster than Nyquist
FUSC	Fully Used Sub Channelisation

GB	Guard Band
GFDM	Generalized Frequency Division Multiplexing
GI	Guard Interval
GPU	Graphics Processing Unit
GS	Gram-Schmidt
GSD	Grid Steepest Descent
GUI	Graphical User Interface
HDL	Hardware Description Language
HDSL	High-bit-rate Digital Subscriber Line
HiperLAN	High Performance Radio Local Area Network
HMM	Hidden Markov Model
HPA	High-power Amplifier
I	In-Phase
i.i.d.	independent and identically distributed
I/O	Input/Output
IC	Iterative Cancellation
ICI	Inter-carrier Interference
IDFT	Inverse Discrete Fourier Transform
IDSD	Iterative Detection with Soft Decision
IEEE	Institute of Electrical and Electronics Engineers
IFDMA	Interleaved Frequency Division Multiple Access
IFFT	Inverse Fast Fourier Transform

IFrFT	Inverse Fractional Fourier Transform
IMD	Inter-modulation Distortion
IMGS	Iterative Modified Gram Schmidt
IMT	International Mobile Telecommunications
IOTA	Isotropic Orthogonal Transform Algorithm
IP	Intellectual Properties
ISI	Inter-symbol Interference
ITU	International Telecommunication Union
JTAG	Joint Test Action Group
LDPC	Low Density Parity Check
LFDMA	Localised Frequency Division Multiple Access
LNA	Low-noise Amplifier
LO	Local Oscillator
LPF	Low-pass Filter
LS	Least Squares
LSB	Least Significant Bit
LTE	Long Term Evolution
LUT	Look Up Table
MA	Multiple Access
MAC	Media Access Control
MAU	Multiplier-Accumulator Unit
MCM	Multi-carrier Modulation

MF	Matched Filtering
MIMO	Multiple Input Multiple Output
MISO	Multiple Input Single Output
MIT	Massachusetts Institute of Technology
ML	Maximum Likelihood
MMSE	Minimum Mean Squared Error
modem	modulator-demodulator
MRC	Maximal Ratio Combining
MRVD	Modified Real Valued Decomposition
MSB	Most Significant Bit
MUD	Multi-user Diversity
MUX	Multiplexer
NC-OFDM	Non-Contiguous OFDM
NGD	Native Generic Database
NGN	Next Generation Network
NP	Non-Polynomial
OFDM	Orthogonal Frequency Division Multiplexing
OFDMA	Orthogonal Frequency Division Multiple Access
OQAM	Offset Quadrature Amplitude Modulation
P/S	Parallel-to-Serial
PA	Power Amplifier
PAM	Pulse Amplitude Modulation

PAPR	Peak-to-Average Power Ratio
PCI	Peripheral Component Interconnect
PED	Partial Euclidean Distance
PHY	Physical
PLC	Power Line Communication
PLR	Packet Loss Ratio
PRBS	Pseudo Random Binary Sequence
PSD	Power Spectral Density
PSK	Phase Shift Keying
PUSC	Partially Used Sub Channelisation
Q	Quadrature
QAM	Quadrature Amplitude Modulation
QoS	Quality of Service
RAM	Random Access Memory
RC	Raised Cosine
ROM	Read Only Memory
RRC	Root Raised Cosine
RTL	Register Transfer Logic
RVD	Real-valued Decomposition
S/P	Serial-to-Parallel
SC-FDMA	Single Carrier Frequency Division Multiple Access
SCD	Spectral Correlation Density

SD	Sphere Decoding
SDP	Semi Definite Programming
SDR	Software Defined Radio
SE	Schnorr-Euchner
SEFDM	Spectrally Efficient Frequency Division Multiplexing
SelE	Selective Equalization
SF	Sort Free
SIC	Successive Interference Cancellation
SISO	Single Input Single Output
SNR	Signal-to-Noise Ratio
SOSA	Selective Opportunistic Spectrum Access
SP	Shortest Path
SQNR	Signal-to-Quantisation-Noise Ratio
STC	Space-Time Coding
SVD	Singular Value Decomposition
TSVD	Truncated Singular Value Decomposition
TVWS	Television White Spaces
USB	Universal Serial Bus
USRP	Universal Software Radio Peripheral
VHDL	Very-high-speed-integrated-circuit Hardware Description Language
VLSI	Very Large Scale Integration

LIST OF ABBREVIATIONS

Wi-Fi	Wireless Fidelity
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network
WMAN	Wireless Metropolitan Area Network
ZF	Zero Forcing

List of Symbols and Operators

$ \cdot $	Absolute value of a number
$\lceil\cdot\rceil$	Ceiling function, meaning rounding up to the closest integer
$chol\{\cdot\}$	Cholesky decomposition function
$\mathcal{O}(\cdot)$	The order of complexity
$\kappa(\cdot)$	Condition number of a matrix
\mathcal{M}	Set of constellation points pertaining to a specific modulation scheme, such as M-QAM or M-PSK
$diag(\cdot)$	Denotes a matrix having diagonal form
mod	$a \bmod b$ gives the modulus of the remainder after division of a by b
E_b/N_0	Energy per bit to noise power spectral density ratio
$\lambda(\cdot)$	Eigenvalue of a matrix
$\lambda_{max}(\cdot)$	Maximum eigenvalue of a matrix
$\lambda_{min}(\cdot)$	Minimum eigenvalue of a matrix
$\hat{\mathbf{s}}$	Estimate of the transmitted signal vector \mathbf{s} after slicing
$\tilde{\mathbf{s}}$	Unconstrained estimate of the transmitted signal vector \mathbf{s} before slicing
$\ \cdot\ $	ℓ^2 -norm also known as the Euclidean distance
$\lfloor\cdot\rfloor$	Floor function, meaning rounding down to the closest integer
$\Im m$	Imaginary part of a complex number
j	Imaginary unit defined as $j = \sqrt{-1}$

\in	$\mathcal{X} \in \mathcal{Y}$ means that \mathcal{X} takes values in the set \mathcal{Y}
∞	Denotes infinity
$\langle \cdot, \cdot \rangle$	Inner product on a vector space
A	Where a symbol appears in bold, this will denote a vector or a matrix of any dimension
\mathbf{A}^H	The Hermitian of a vector or square matrix denoting the conjugate transpose
\mathbf{A}^{-1}	The inverse of a square matrix
\mathbf{A}^T	The transpose of a square matrix
\Re	Real part of a complex number
$rect\{\cdot\}$	Rectangular pulse shaping function
$\sigma(\cdot)$	Singular value of a matrix
$\sigma_{max}(\cdot)$	Maximum singular value of a matrix
$\sigma_{min}(\cdot)$	Minimum singular value of a matrix
$\lfloor \cdot \rfloor$	Slicing operator, which rounds the value to the nearest constellation point
$E\{\cdot\}$	The expected value of a random variable, also referred to as statistical expectation
\subset	Denotes a sub-set
$[\cdot]'$	Vector or matrix transpose operation

Part I

Next Generation Communication Technologies

Chapter 1

Introduction

The Olympic Games that took place in London in the summer of 2012 presented a great challenge for the telecommunication providers putting mobile communications technology to the test. More than four million extra people were expected to arrive in London during the Games [1] accessing broadband services using a myriad of devices ranging from 2nd Generation (2G) mobile phones to the latest smartphones, tablets and laptops connecting to the Internet via mobile networks and Wireless Fidelity (Wi-Fi) hotspots. This year also marks the first 4th Generation (4G) trials taking place in the United Kingdom [2].

While modulation schemes for high-speed data transmission have been of continuing interest for well over half a century [3], the advancement of modern communication systems can be attributed to the use of Multi-carrier Modulation (MCM) techniques, which provide better immunity against multipath fading compared to single carrier schemes. The generic term MCM appeared in the 1990's [4], yet the concept of parallel data transmission dates back to 1958 with the Collins Kineplex data systems [5]. The theoretical principles of Orthogonal Frequency Division Multiplexing (OFDM) were established almost a decade later [6] leading to a subsequent patent [7] and practical performance evaluation [8]. Around the same period, the Discrete Fourier Transform (DFT) was considered for the design of OFDM systems [9] which favoured the imple-

mentation of a data communication system using digital circuitry [10].

Two decades later, one of the first commercial systems employing OFDM came to light, known as Discrete Multi-tone (DMT) [11]. DMT was designed for High-bit-rate Digital Subscriber Line (HDSL) access [12] leading to the standardised Asymmetric Digital Subscriber Line (ADSL) system. OFDM also formed part of the Institute of Electrical and Electronics Engineers (IEEE) 802.11 standards for Wireless Local Area Network (WLAN) access with 802.11n being the most recent one developed [13]. The technology based on the 802.11 standards is referred to as Wi-Fi with the European counterpart being High Performance Radio Local Area Network (HiperLAN) II [14]. OFDM has also found application in Digital Audio Broadcasting (DAB) [15] and especially Digital Video Broadcasting (DVB) [16] [17] [18].

While existing 3rd Generation (3G) systems utilise the Code Division Multiple Access (CDMA) air interface standard, OFDM is regarded as one of the key technology enablers in 4G mobile telecommunication networks [19]. The two primary competitors for providing mobile broadband access are Worldwide Interoperability for Microwave Access (WiMAX) [20] and 3rd Generation Partnership Project (3GPP)-Long Term Evolution (LTE) [21]. WiMAX is based on the IEEE 802.16 standard which defines the Wireless Metropolitan Area Network (WMAN) air interface for broadband wireless access [22]. The most recent version of the standard is IEEE 802.16m [23]. In the same fashion, LTE is a new air interface defined by 3GPP [24] with Release 10 currently under development, commonly referred to as LTE-Advanced.

Two additional technologies which promise significant performance gains in future, or what is termed Next Generation Networks (NGNs), are Multiple Input Multiple Output (MIMO) and Adaptive Modulation and Coding (AMC). MIMO increases spectrum efficiency by exploiting transmit diversity achieved through spatial multiplexing. AMC allows sub-carriers to be modulated with varying levels of Quadrature Amplitude Modulation (QAM), thus offering optimum throughput depending on the instantaneous channel conditions [20]. These technologies complement OFDM to enhance robustness

against wireless channel impairments while concurrently offering high data rates [25] [26].

Beyond wireless and mobile networks, the use of OFDM has been investigated in many other fields including optical systems [27] [28], Power Line Communication (PLC) [29] [30] and physical layer encryption [31] [32], the latter known as Masked-OFDM. More recently, OFDM has been combined with Offset Quadrature Amplitude Modulation (OQAM) with the aim of increasing capacity in PLC systems [33] [34].

Despite the success of the aforementioned systems during the 20th century, the beginning of the 21st century was marked by a pressing need for more spectrum [35]. Networks are confronted with an ever-growing number of users who run bandwidth-hungry applications using a wide range of devices. Yet, spectrum is a scarce and an expensive resource. This fact coupled with users' demand for higher data rates, better Quality of Service (QoS) and lower costs, has led telecommunication operators and engineers to recognise that the best means for increasing capacity in future communication systems is via improved spectrum efficiency.

So far, it has been established that OFDM is the preferred modulation method for 4G networks. Notwithstanding, OFDM has certain disadvantages which make it inapt for spectrum optimisation. First, the overlapping sub-carriers have to be perfectly orthogonal between them to allow a successful recovery of the transmitted symbols. This is achieved by setting the frequency spacing between the sub-carriers to equal exactly the reciprocal of the OFDM symbol period. Consequently, this poses a limit on the number of sub-carriers into which the signal frequency band may be divided. Second, local oscillator offsets and Doppler shifts which occur in practical scenarios cause frequency and timing errors resulting in loss of orthogonality and giving rise to Inter-carrier Interference (ICI) and Inter-symbol Interference (ISI) [36]. Tight synchronisation between the transmitter and receiver oscillators is therefore crucial to maintain orthogonality and counteract ICI. To combat ISI, a Guard Interval (GI) is added to the OFDM signal in the form of a Cyclic Prefix (CP). This CP, however, introduces

an overhead and thus reduces spectrum efficiency.

Hereby, the optimum use of radio spectrum seems to be a burning research topic for systems beyond 4G and calls for novel techniques to tackle the challenge of conveying more data within the available bandwidth. Research trends suggest that two indispensable technologies for optimising spectrum use in future cellular and wireless networks will be Cognitive Radio (CR) and spectrally efficient multi-carrier modulation schemes. CR has been the subject of research and development for over a decade. It is based on Software Defined Radio (SDR) [37] and allows the existing wireless spectrum to be exploited opportunistically via Dynamic Spectrum Access (DSA). With the recent switchover to digital television which released large areas of frequency, known as TV white spaces, CR is gradually moving from the laboratory to the commercial world [38]. Multi-carrier modulation techniques aimed at improving spectrum utilisation have only recently started to gain more attention by the academic community across the globe. These techniques propose time-frequency packing by:

- Deliberately and counter-intuitively violating the orthogonality principle defined for OFDM systems, hence reducing the frequency spacing between the sub-carriers to save bandwidth or increasing the data rate within the same bandwidth.
- Transmitting non-orthogonal pulses with special properties.
- Applying advanced detection algorithms at the receiver, such as soft decoding and Sphere Decoding (SD), which has been made feasible due to advances in silicon technology.

The idea of designing systems able to transmit data at speeds close to the Nyquist rate is not new [39]. In 1975, Mazo showed that for a reduction in transmission bandwidth by a factor $\rho = 0.8$ with reference to conventional Nyquist pulses, the signalling rate can be increased by $(\frac{1-\rho}{\rho}) \cdot 100 = 25\%$ without a penalty in error performance [40]. This concept, termed Faster than Nyquist (FTN) signalling, was validated only for one-dimensional (1D) modulation schemes, such as Pulse Amplitude Modulation (PAM),

Amplitude Shift Keying (ASK) and Binary Phase Shift Keying (BPSK). In 2005, Rusek and Anderson extended the concept to a two-dimensional (2D) space, for example 16-QAM [41]. FTN increases the data rate within the Nyquist signal bandwidth at the expense of introducing additional ISI [42].

One of the first systems designed to increase the data rate relative to an OFDM system was OFDM/OQAM [43]. This was achieved by discarding the guard interval. As a result, alternative prototype functions providing good localisation in the time domain had to be employed to mitigate the effects of ISI. The use of non-orthogonal functions offered new degrees of freedom favouring the design of ameliorated pulses which were better-suited for minimising the ICI and ISI effects present in frequency-selective and time-dispersive channels [44]. An example of a prototype function offering quasi-optimum localisation is the Isotropic Orthogonal Transform Algorithm (IOTA) function which is generated by applying an IOTA filter to a Gaussian function [45].

In retrospect, a number of systems have been explored with the aim of increasing spectrum efficiency through bandwidth savings and/or flexible frequency allocation. Examples of systems which reduce spectrum utilisation by half but constrained to 1D modulation schemes include Fast Orthogonal Frequency Division Multiplexing (FOFDM) [46] and M-ary ASK OFDM (MASK-OFDM) [47]. Systems designed to operate with 2D modulation methods include High Compaction MCM (HC-MCM) [48], Overlapped FDM (Ov-FDM) [49], FTN signalling [50], Frequency Overlapped Multi-carrier System (FOMS) [51], as well as Spectrally Efficient Frequency Division Multiplexing (SEFDM) [52].

The significance of the aforementioned spectrally efficient systems is further highlighted by the fact that these techniques are gradually being introduced in many fields beyond wireless and cellular networks. Notable examples include FOFDM, Optical Dense OFDM and FTN for optical communications [53] [54] [55], FTN for coding [56] and physical layer security [57], as well as Generalized Frequency Division Multiplexing (GFDM) for CR applications [58].

Table 1.1: Comparison of Xilinx (Altera) device families.

Device Family	Characteristics
Virtex (Stratix)	Highest density and bandwidth (high-end applications)
Spartan (Arria)	Balanced performance, price and cost (mid-range applications)
Artix (Cyclone)	Lowest power, footprint and cost (cost-sensitive applications)
Kintex (-)	Offers the best price/performance/watt

This thesis initially examines CR as one of the key technologies for improving spectrum utilisation in future communication systems. Subsequently, the attention is turned to SEFDM, a novel multi-carrier modulation technique, which tackles the issue of spectrum efficiency by breaking previously established theoretical barriers. A critical aspect, which will determine the success of both these technologies, is the validation of theoretical results through practical experimentation. Experimental work in turn is facilitated via testbeds and programmable platforms which increasingly employ Field Programmable Gate Arrays (FPGAs) [59]. In contrast to general-purpose processors, FPGAs offer greater flexibility and allow the designer to customise any aspect of a circuit, thus optimising its use [60]. Thanks to the implementation of designs in the spatial domain, which permits multiple levels of pipelining, dedicated FPGA and Very Large Scale Integration (VLSI) circuits are well-positioned to offer target throughput rates which cannot be met by conventional Digital Signal Processors (DSPs) [61]. Consequently, in this work, FPGAs were chosen as the preferred devices for the implementation of SEFDM transceiver algorithms.

The most popular vendors of FPGA chips include Xilinx, Altera, Lattice and Actel. However, the two primary competitors in this field are Xilinx and Altera [59]. The testbed employed in this work uses Xilinx devices. Two main reasons motivated this choice; first, the programmable platforms available in the laboratory at the time of development were equipped with Xilinx FPGAs; second, the predecessor to this work [59] incorporated Xilinx chips, hence using devices from the same vendor would favour compatibility, as well as smooth migration and/or scalability. For informative purposes,

a comparison between Xilinx and Altera device families is given in Table 1.1 [62] [63].

1.1 Thesis Outline

While there are a number of emerging techniques targeting increased spectrum efficiency, this thesis examines two key technologies, namely CR and SEFDM. To this end:

Chapter 2 gives an overview of technologies employed in NGNs for managing spectrum, enhancing capacity and exploiting multi-user diversity. These include MIMO, MCM, as well as Multiple Access (MA) techniques with a particular emphasis on Orthogonal Frequency Division Multiple Access (OFDMA) and Single Carrier Frequency Division Multiple Access (SC-FDMA). This chapter also delves into the fundamental principles surrounding MCM and gives a good qualitative and quantitative description of OFDM. The theoretical upper bounds for communication below orthogonality are also identified leading up to the treatment of SEFDM systems.

Chapter 3 explores spectrum management techniques for efficiently allocating and sharing the available radio resources. The first part discusses techniques exploiting the inherent cyclostationarity present in OFDM based systems to address the detection challenge in CR networks, such as cyclostationary signatures and automatic modulation classification. The second part, proposes a Selective Opportunistic Spectrum Access (SOSA) framework which aims to improve the QoS of a CR user by optimising the spectrum sensing cycle. This is achieved by applying traffic prediction techniques to a conventional CR system.

Chapter 4 provides a fresh look at the basic principles and properties surrounding SEFDM systems with an eye on implementation. The concept of time-frequency packing is explained in detail. It is shown that the ill-conditioning that arises in SEFDM may be attributed to deliberate aliasing and spectral leakage. An in depth review to determine the most suitable transceiver architecture for hardware realisation is also carried out.

Chapter 5 considers practical-level aspects involved in the application of SEFDM in the real world. This chapter introduces the available SEFDM detection algorithms, together with their corresponding complexity scaling behaviour, hence leading to an educated decision regarding the suitability of these choices for hardware implementation. The benefits and drawbacks of oversampling at the modulation, de-modulation and detection stages are quantified and discussed. The performance of SEFDM in a fixed-point environment which takes into account the limited precision available in hardware devices is also evaluated. To this end, this chapter forms the basis for the development and evaluation of transceiver algorithms and architectures employing FPGAs that appear in later chapters.

Chapter 6 focuses on linear detection techniques and describes the FPGA design and implementation of SEFDM receivers employing Zero Forcing (ZF) and Truncated Singular Value Decomposition (TSVD). Results show that these detectors are suitable for the recovery of signals employing real-valued modulation schemes, for example BPSK. Conversely, these techniques exhibit poor performance for complex-valued modulation schemes, for example M-QAM, in accordance with previous theoretical predictions and simulation results. The corresponding trade-offs in computational complexity and error performance are explained. The chapter closes with a discussion of optimisation techniques applied to FPGA architectures with an emphasis on the special properties of the matrices employed in SEFDM systems.

Chapter 7 is dedicated to the analysis and experimental evaluation of more sophisticated detection techniques. Based on previous work which demonstrates that Maximum Likelihood (ML) has an impractical computational complexity due to the exhaustive search nature of the algorithm, this chapter considers iterative tree-search algorithms. The chapter starts by presenting the concepts of SD and Fixed Sphere Decoding (FSD), the latter fixing the complexity of the conventional SD algorithm. This brings us to the next area of discussion which gives an account of the FSD variants employed in MIMO systems and their applicability to the SEFDM detection problem. A number

of algorithmic optimisations are described while the challenges encountered during an FPGA realisation are explained. The SD algorithm is initially employed as part of a hybrid hardware and software SEFDM pseudo-transceiver to verify the accuracy of experimental signals generated using an FPGA. The FSD algorithm is then evaluated in terms of suitability for hardware implementation. It is found that the original FSD exceeds the device's available resources due to the large system dimension in terms of the number of sub-carriers employed. Hence, modified sorting and Real-valued Decomposition (RVD) strategies are applied, which reduce the complexity of the original FSD targeting it for hardware implementation while offering trade-offs in execution speed and Bit Error Rate (BER) performance.

Finally, Chapter 8 presents conclusions and summarises the salient points discussed in previous chapters. This chapter also lists potential research areas open to further investigation.

Due to the nature of the Engineering Doctorate (EngD) programme, this thesis comprises a number of closely related topics. Fig. 1.1 depicts the background and motivation of this research work. Fig. 1.2 shows the links that exist between different aspects of CR leading on finally to the examination of the key topic in this thesis being that of SEFDM, as illustrated in Fig 1.3.

This thesis comprises three appendices. The first appendix describes a Graphical User Interface (GUI) developed in *MathWorks*® *MATLAB*® (henceforth referred to as MATLAB) for the simulation of OFDM and SEFDM systems. The second appendix presents schematic diagrams of the platform employed in Chapters 6 and 7. Waveforms and screenshots acquired during the real-time operation of the application under consideration are also illustrated. Finally, the third appendix details the code structure for the SEFDM receiver implemented using Very-high-speed-integrated-circuit Hardware Description Language (VHDL) and the C programming language. The DVD disc contains the MATLAB code, VHDL code and C code, developed during the EngD programme.

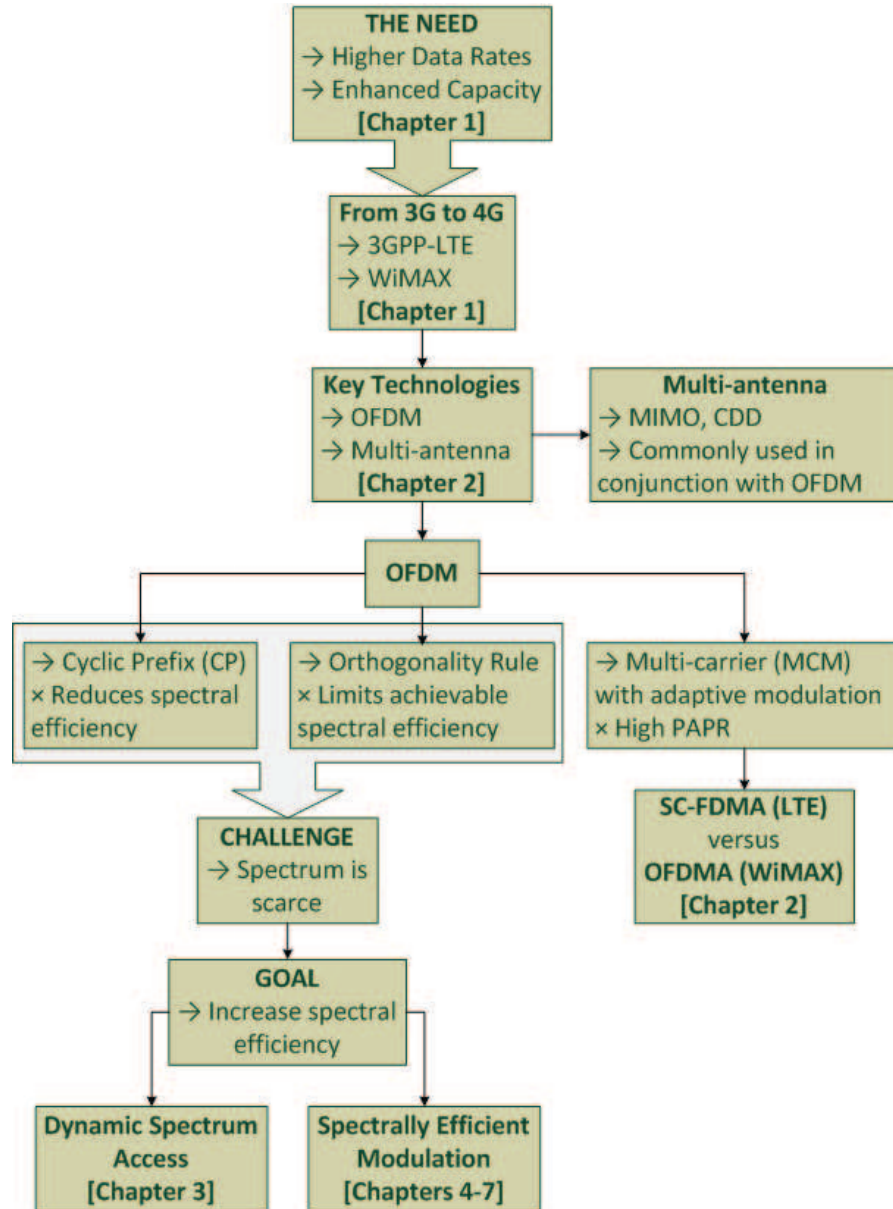


Figure 1.1: Focus of Chapters 1 and 2 (OFDM, MIMO).

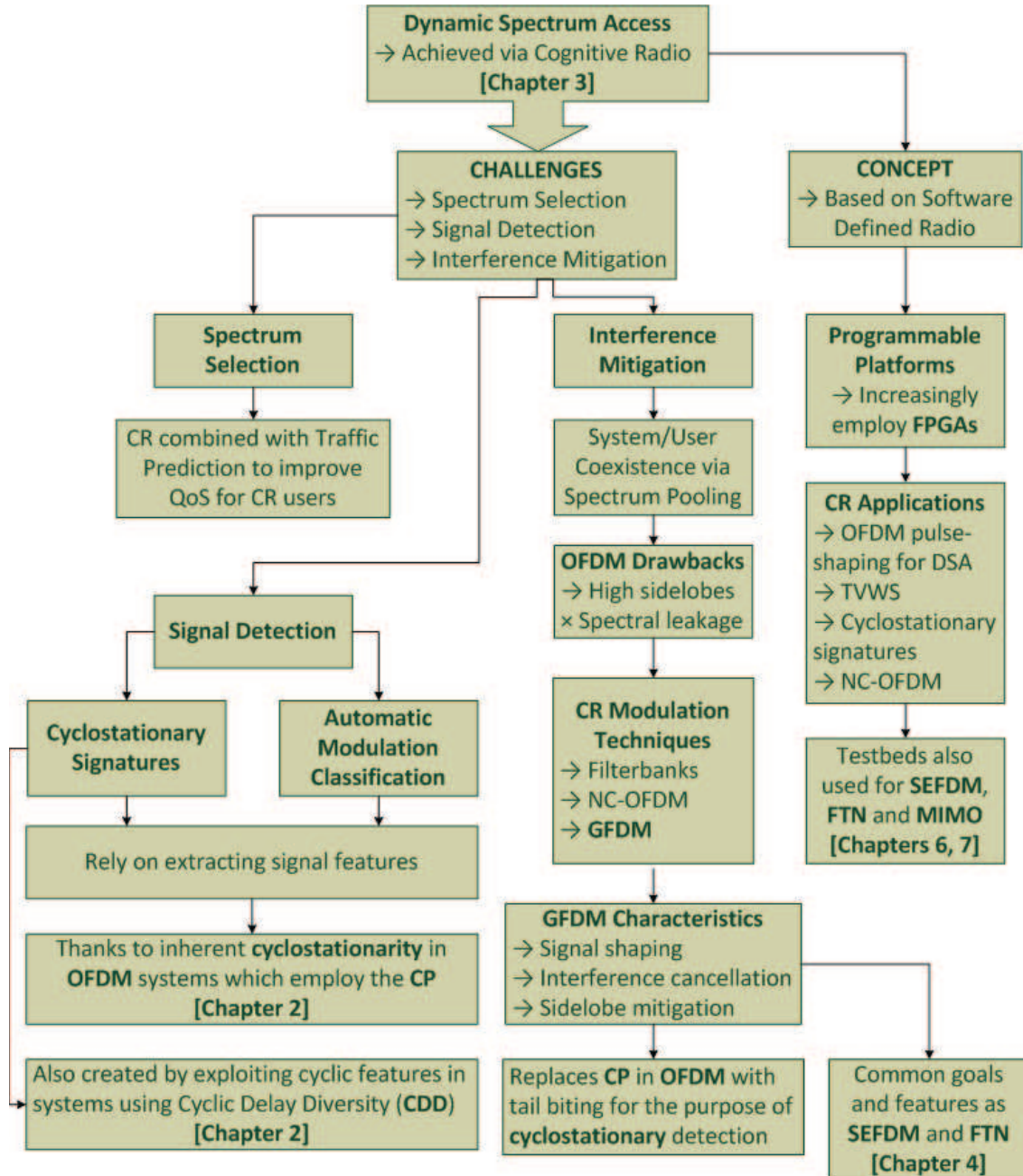


Figure 1.2: Focus of Chapter 3 (CR).

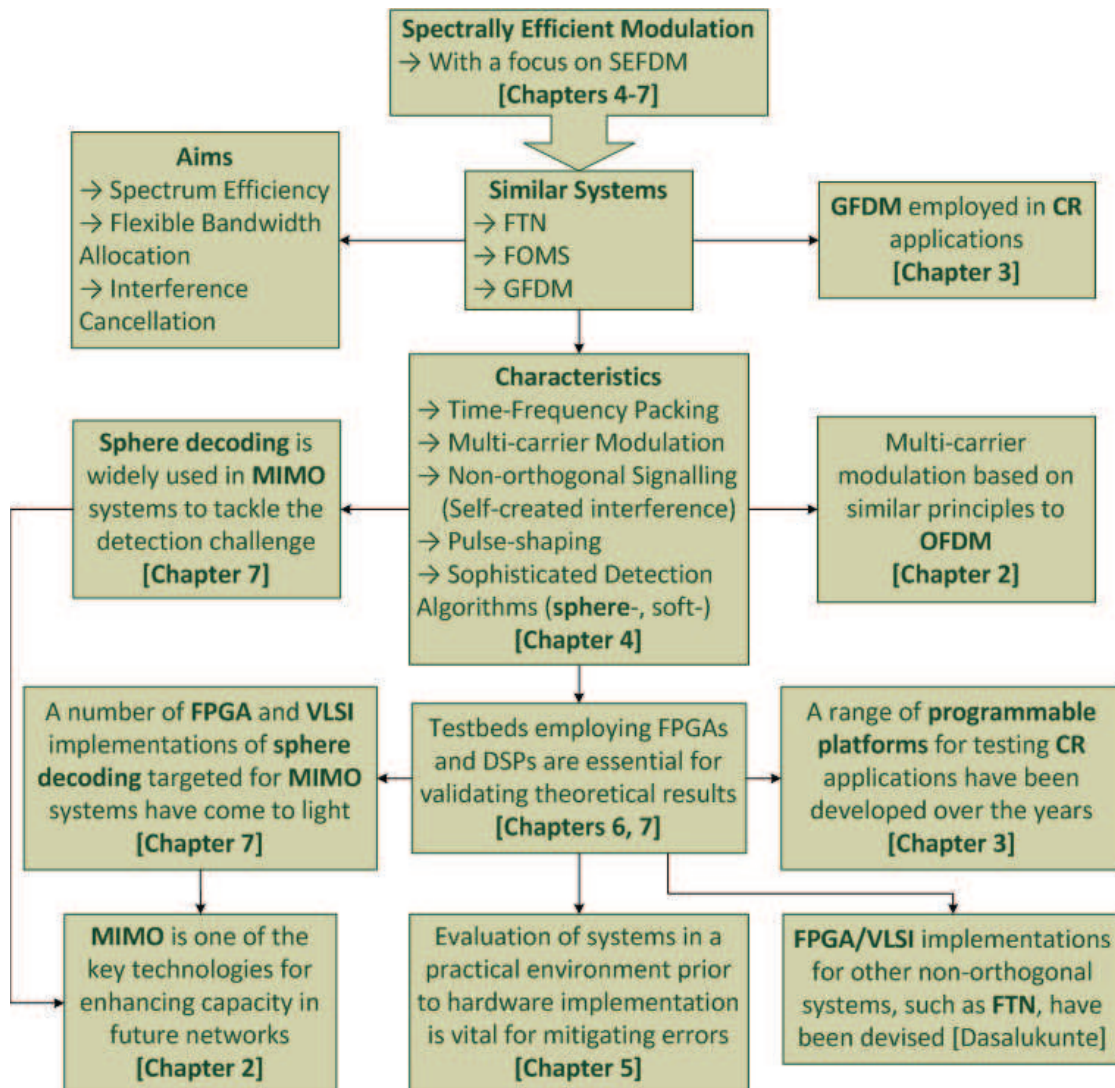


Figure 1.3: Focus of Chapters 4-7 (SEFDM).

1.2 Main Contributions

The goal of this thesis is to examine two key technology enablers for the purpose of spectrum optimisation in wireless communication systems, namely CR and SEFDM, with a focus on the latter. The detailed contributions of this work are as follows:

- Developed a SOSA framework to improve the QoS of a CR user. Contrary to other approaches, this model considers CR and traffic prediction in conjunction while taking into account practical issues, such as discontinuous target frequency bands and the limited dynamic range of the radio front-end used for spectrum sensing. It is shown that the proposed solution increases throughput and reduces packet losses with respect to a conventional system.
- Evaluated the impact of oversampling on the performance of SEFDM systems in an Additive White Gaussian Noise (AWGN) environment. It is shown that the conditioning of the SEFDM sub-carriers correlation matrix is ameliorated in proportion to the oversampling factor. As a result, it is demonstrated that the BER performance of different SEFDM detectors is improved accordingly, particularly for linear detection methods.
- Developed a bit-accurate, fixed-point, FPGA model of an SEFDM transceiver. Using this model, the impact of finite word length effects on the performance of SEFDM systems was evaluated. It is shown that bit scaling and dynamic range have a significant impact on resource requirements and quantisation noise.
- Implemented a reconfigurable SEFDM receiver with the aid of FPGAs. The receiver comprises a Fast Fourier Transform (FFT) based demodulator and a linear detector with the ability to switch between ZF and TSVD. The architecture is evaluated in terms of resource utilisation, latency and throughput.
- Introduced new variants of the FSD algorithm which reduce its complexity rendering it more suitable for application in the real world. It is shown that the

sort-free and modified decomposition approaches optimise the execution of FSD at both an algorithmic and architectural level.

- Implemented the original and modified FSD algorithms on a DSP to acquire more realistic estimates of the computational complexity associated with these detection techniques. Evaluated the experimental versions of these FSD variants in terms of execution speed, as well as the number of arithmetic and logic operations required by each algorithm.
- Examined the results obtained from the implementation of different FSD variants using a Xilinx Virtex 6 FPGA chip. Results demonstrate that the original FSD algorithm demands a high resource utilisation, which exceeds the chip’s available logic slices, in order to sustain the same throughput as the modified version of the algorithm adopting the sort-free strategy.

In summary, the majority of the work in this thesis evolves around a feasibility study to assess if the current technology is adequate to allow the application of SEFDM in the real world. Low complexity solutions including both algorithms and FPGA architectures have been devised for the reception of SEFDM signals while taking into account paramount system-level aspects at the transmitter side. The proposed designs are described in detail and provide a reference for the true hardware complexity of corresponding implementations. To the best of the author’s knowledge, these implementations are ranked among the world’s first with regard to the realisation of a spectrally efficient multi-carrier communication system.

1.3 List of Publications

The work presented in this thesis has resulted in the following conference and journal publications listed in chronological order:

1. G. Yuan, R. C. Grammenos, Y. Yang, and W. Wang, “Selective Spectrum Sensing

- and Access based on Traffic Prediction,” in *IEEE 20th Int. Symp. Personal, Indoor and Mobile Radio Commun.*, Tokyo, PIMRC, 2009, pp. 1078-1082.
2. R. C. Grammenos and Y. Yang, “Cyclostationary signatures for cognitive radio applications and novel multiple access systems,” in *London Commun. Symp.*, London, LCS, 2009.
3. G. Yuan, R. C. Grammenos, Y. Yang, and W. Wang, “Performance Analysis of Selective Opportunistic Spectrum Access With Traffic Prediction,” *IEEE Trans. Veh. Technol.*, vol. 59, no. 4, pp. 1949-1959, 2010.
4. R. Grammenos and I. Darwazeh, “SC-FDMA and OFDMA : The two competing technologies for LTE,” in *4th Int. Symp. Broadband Commun.*, Melaka, ISBC, 2010.
5. R. Grammenos and I. Darwazeh, “FPGA design considerations for non-orthogonal FDM signal detection,” in *London Commun. Symp.*, London, LCS, 2010.
6. R. C. Grammenos, S. Isam, and I. Darwazeh, “FPGA Design of a Truncated SVD Based Receiver for the detection of SEFDM Signals,” in *IEEE 22nd Int. Symp. Personal, Indoor and Mobile Radio Commun.*, Toronto, PIMRC, 2011.
7. R. C. Grammenos and I. Darwazeh, “FPGA design of low complexity SEFDM detection techniques,” in *London Commun. Symp.*, London, LCS, 2011.
8. M. R. Perrett, R. C. Grammenos, and I. Darwazeh, “A Verification Methodology for the Detection of Spectrally Efficient FDM Signals Generated using Reconfigurable Hardware,” in *IEEE Int. Conf. Commun.*, Ottawa, ICC, 2012.
9. R. C. Grammenos and I. Darwazeh, “Hardware Implementation of a Practical Complexity Spectrally Efficient FDM Reconfigurable Receiver,” in *IEEE 23rd Int. Symp. Personal, Indoor and Mobile Radio Commun.*, Sydney, PIMRC, 2012.

10. R. C. Grammenos and I. Darwazeh, “Performance Trade-Offs and DSP Evaluation of Spectrally Efficient FDM Detection Techniques,” to appear in *IEEE Int. Conf. Commun.*, Budapest, ICC, 2013.
11. T. Xu, R. C. Grammenos, and I. Darwazeh, “FPGA Implementations of Real-Time Detectors for a Spectrally Efficient FDM System,” to appear in *IEEE Int. Conf. Telecommun.*, Casablanca, ICT, 2013.
12. T. Xu, R. C. Grammenos, F. Marvasti, and I. Darwazeh, “An Improved Fixed Sphere Decoder Employing Soft Decision for the Detection of Non-Orthogonal Signals,” submitted to *IEEE Commun. Lett.*, 2013.

Chapter 2

Multi-carrier Communication Techniques

Over the past two decades, the International Telecommunication Union (ITU), which specifies the requirements for radio interfaces, has been working closely with the public and private sectors with the aim of developing a global, broadband multimedia, International Mobile Telecommunications (IMT) system. The IMT concept was introduced in the late 1990's (IMT-2000) and includes the family of 3G systems. The next standard called IMT-Advanced, also known as 'systems beyond IMT-2000', is designed to provide a unified platform to deliver a wide range of data rates and services in both low and high mobility¹ environments. A number of different access systems will exist but these will be compatible in order to allow interconnection and network sharing. The IMT-Advanced specification aims to achieve this by defining stringent requirements with regard to the Physical (PHY) and Media Access Control (MAC) layers. Hence, the key points outlined in IMT-Advanced [64] include:

- Co-existence and interoperability between diverse radio access technologies.
- Scalable bandwidth and variable transmission modes to deliver content according

¹In this context, mobility refers to the travelling speed of a user carrying a mobile device ranging from a stationary position to high-speed motion (for example, when travelling on a train).

to the particular application demand.

- Enhanced peak data rates of 1 Gbps and 100 Mbps in low and high mobility scenarios, respectively.

Currently deployed communication systems cannot achieve the aforementioned data rates [65]. Hereby, this chapter starts with an overview of state-of-the-art technologies which have already been or are in the process of being employed in today's networks to deliver the data rates specified by IMT-Advanced.

Section 2.1 looks at multiple antenna technologies, such as MIMO and Cyclic Delay Diversity (CDD). Section 2.2 is devoted to the comparison of OFDMA and SC-FDMA, since the latter addresses the Peak-to-Average Power Ratio (PAPR) problem associated with OFDM and has been included officially in the 3GPP-LTE standards. Simulation based studies of the effect of different sub-carrier mapping methods on the PAPR performance of OFDMA and SC-FDMA is detailed in Section 2.2 and a set of results is reported in the same section. The chapter closes with a detailed review of OFDM in Section 2.3 forming the foundation for the analysis of SEFDM in subsequent chapters. This final section describes the signal properties and transceiver architectures of OFDM based systems. Part of the work in this chapter was presented at the International Symposium on Broadband Communications (ISBC) in 2010 [66].

2.1 Multiple Antenna Technologies

2.1.1 Multiple Input Multiple Output

MIMO is without doubt one of the most cost-effective technologies for delivering Gbps data rates [67]. In conjunction with OFDM, field trials have demonstrated [68] that MIMO-OFDM systems can offer significant performance gains in terms of throughput, coverage and reliability.

MIMO increases data throughput by exploiting antenna diversity. This is achieved through spatial multiplexing, a technique which divides the data stream into multiple

sub-streams which are then transmitted in parallel. Data rates of up to 150 Mbps and 300 Mbps can be achieved in LTE systems using 2x2 and 4x4 MIMO antenna configurations, respectively. In practice, MIMO is used in conjunction with other smart antenna technologies, such as antenna beam-forming and Space-Time Coding (STC) to maximise performance gains [69]. Moreover, Maximal Ratio Combining (MRC) is used at the receiver to enhance link reliability in the presence of noise and fading [36].

MIMO research is important in the pursuit of SEFDM systems, as both technologies encounter similar detection challenges. For MIMO systems, these include channel matrix pre-coding and antenna stream decoding [70]. For the former, the requirement is for the matrix to be non-singular in order to be invertible. However, practical aspects, such as small antenna spacing, may degrade the conditioning of the channel matrix, thereby hindering the matrix's invertibility. Techniques like Singular Value Decomposition (SVD) are therefore employed to compute the pseudoinverse of a matrix. As for the decoding of the antenna streams, ML is the optimum detection scheme, yet the Non-Polynomial (NP) complexity linked to its execution renders it inapt for hardware implementation. Hence, sub-optimum techniques based on SD are explored offering a substantial reduction in complexity compared to the ML solution while maintaining a reasonable error performance.

As will be revealed in latter chapters of this thesis, the ill-conditioning of the sub-carriers correlation matrix in SEFDM, which has similar properties to the channel matrix in MIMO systems, complicates the design of receiver architectures. For this reason, methods previously employed in MIMO are adapted to address the detection challenge in SEFDM. Nonetheless, the key difference between MIMO and SEFDM is that the dimension of the problem for SEFDM is far greater compared to MIMO leading to higher computational complexity and thus a more involved implementation.

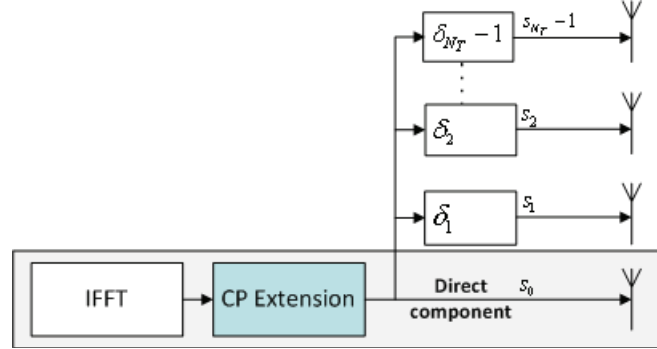


Figure 2.1: MISO transmitter employing Delay Diversity.

2.1.2 Cyclic Delay Diversity

Another technology which is being introduced in LTE-Advanced is CDD [71], which was originally investigated and implemented for use in DVB as part of the PLUTO project [72]. CDD is a low complexity and flexible spatial diversity scheme for OFDM based systems which intends to improve signal detection at the receiver by exploiting antenna transmit diversity. While the term was coined circa 2001 [73], the concept of using transmit diversity to improve bandwidth efficiency dates back to the early 1990's [74]. A patent describing the architecture and operation of the system was published in 2005 [75]. CDD is also being included in the WiMAX standard under the name Cyclic Shift Transmit Diversity (CSTD) [20].

CDD is based on the concept of Delay Diversity (DD) [73]. Contrary to MIMO systems, DD does not require the alteration of the receiver used in a conventional Single Input Single Output (SISO) OFDM system. Instead, it uses multiple antennas at the transmitting end to send delayed replicas of the original signal, as illustrated in Fig. 2.1. Hence, it should be evident that DD entails a Multiple Input Single Output (MISO) transceiver architecture. By this means, it translates spatial diversity into frequency diversity resulting in *uncorrelated frequency selectivity* [75].

With DD, each transmit antenna conveys a replica $s_i(k)$ of the original signal $s_0(k)$ differing by a delay δ_i . This delay δ_i has to be a multiple of the system sampling time

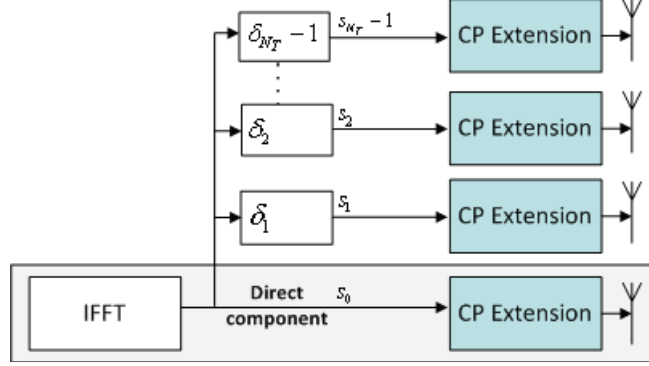


Figure 2.2: MISO transmitter employing Cyclic Delay Diversity.

[73] while the total number of delayed replicas is equal to $N_T - 1$ where N_T is the number of transmit antennas.

The drawback with DD is that the added delay results in increased channel delay spread τ_d , the latter signifying the longest delay of a reflected signal, thus making the system more prone to ISI. Conversely, this imposes a tight constraint on the maximum value of δ_i . CDD alleviates this problem by adding the cyclic prefix to each of the delayed replicas independently rather than at the output of the OFDM modulation process, as depicted in Fig. 2.2.

With CDD, the transmitted signal replicas differ in cyclic shifts rather than delays. In other words, the original signal $s_0(k)$ is rotated in phase prior to being cyclically extended, thus skipping redundant data which would be present in a signal with DD and which could potentially cause ISI to adjacent OFDM symbols, as shown in Fig. 2.3. From Fig. 2.3, it should be evident that DD is equivalent to sending the original signal with a delay in time while CDD is equivalent to sending the original signal with a delay in phase.

The advantage of using CDD over DD is in that the delay in CDD is independent of the CP, thereby relaxing the constraint imposed upon the length of the CP in standard OFDM systems leading to improved bandwidth efficiency [76]. Closely related diversity techniques include sub-carrier diversity, phase diversity and time-variant phase

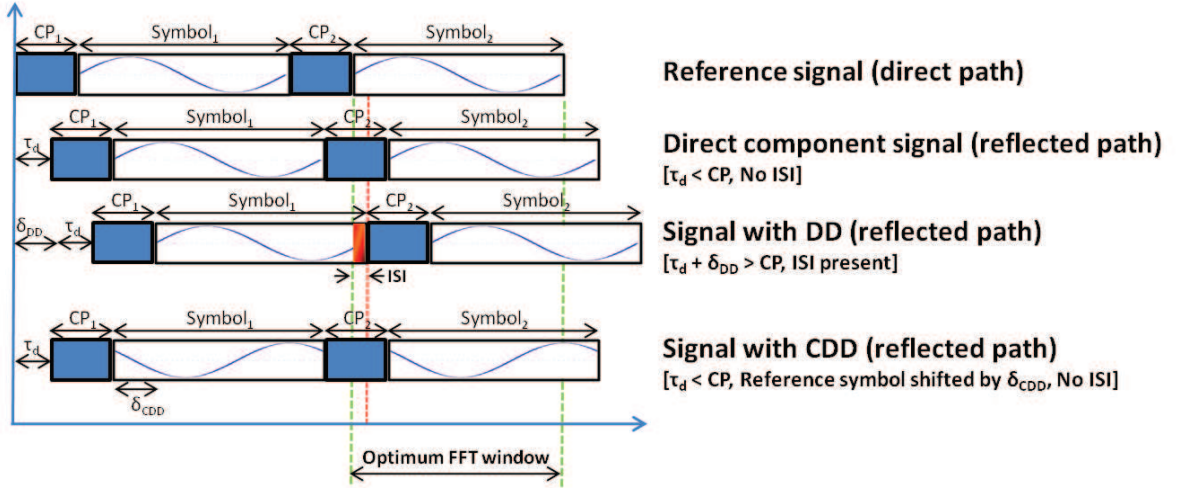


Figure 2.3: Delay Diversity vs. Cyclic Delay Diversity.

diversity, the latter offering additional time selectivity to yield further performance gains. All the aforementioned antenna diversity schemes share the same goal, which is to increase the frequency selectivity of the multipath channel by “uncorrelating the channel paths” [77].

2.1.3 Other Technologies

The aforementioned technologies are being complemented with additional enhancements, such as carrier aggregation, multi-hop relaying and Fractional Frequency Reuse (FFR). Carrier aggregation [78] is being accommodated in LTE-Advanced to provide scalable expansion of the effective bandwidth. This is achieved by combining the radio resources across multiple carriers located in different spectrum bands. Multi-hop relaying [79] trades off capacity for extended coverage by conveying user data between a base station and a mobile terminal via one or more relays. Combined with spatial diversity techniques, relays could provide large cellular coverage with high data rates. Finally, FFR [69] reduces interference by dividing the spectrum into frequency sub-bands, thus favouring flexible sub-channel reuse. FFR tends to increase the overall system capacity at the expense of slight performance degradation at the cell edges.

2.2 OFDMA and SC-FDMA

One of the pitfalls associated with OFDM signals is that they exhibit a high PAPR. This is due to the summation of multiple sub-carriers in parallel which causes high envelope fluctuations giving rise to the noise-like nature of OFDM signals. The ratio increases linearly with the number of sub-carriers N [80] while the maximum theoretical value can be as high as $10\log_{10}(N)$ [81].

PAPR is a crucial design factor in any communication system, since it determines the dynamic range of Analogue-to-Digital (A/D) and Digital-to-Analogue (D/A) converters. It also has a direct impact on the efficiency of Power Amplifiers (PAs), which yield maximum efficiency when they operate at their saturation point. When OFDM signals add up constructively, they may result in a high instantaneous peak power forcing the PA to operate with a large back-off, thus lowering power efficiency. Conversely, allowing the PA to operate in its non-linear region will lead to signal clipping causing Inter-modulation Distortion (IMD) [36]. This distortion increases out-of-band radiation which i) gives rise to unwanted signals interfering with the signal of interest, thereby degrading system performance, and ii) violates the standards set by telecommunications regulators.

To address the PAPR challenge, the SC-FDMA scheme, also known as DFT-spread OFDM, was introduced in LTE systems and is employed in the uplink channel. As will be presented in the following paragraphs, SC-FDMA can reduce the PAPR by at least 2 dB with reference to OFDM systems, thus extending the battery life of mobile devices [36].

SC-FDMA is a MA technique, which means that the time and frequency resources are allocated to multiple users simultaneously, contrary to standalone multiplexing, which allocates all resources to a single user. OFDMA is the competing technology against SC-FDMA. While OFDMA is being used in the downlink in both WiMAX and LTE systems, SC-FDMA is employed in the uplink only in LTE. A comparative review of these two MA schemes was carried out by the author and is provided in [66].

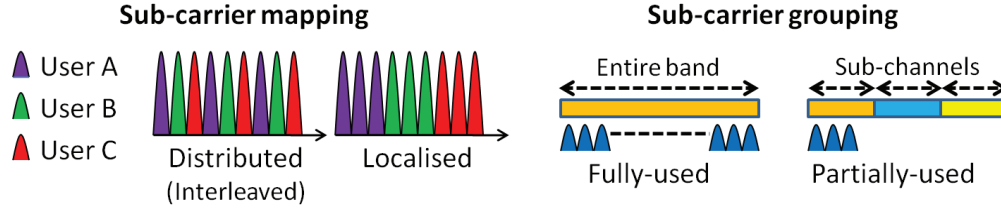


Figure 2.4: Sub-carrier mapping and grouping schemes.

OFDMA is an extension of OFDM, by means of which a fraction of the total number of sub-carriers per OFDM symbol is allocated to each user. This is achieved by dividing the total number of sub-carriers into groups, termed sub-channels in WiMAX or chunks in LTE. Sub-channelisation favours Multi-user Diversity (MUD) and allows resources to be allocated dynamically to users depending on their QoS needs [20].

Sub-carriers are allocated to sub-channels or chunks in either a localised or a distributed fashion. The first method uses a block of contiguous sub-carriers to form a sub-channel. In the latter method, sub-carriers are allocated to sub-channels pseudo-randomly using a permutation mechanism. Localised allocation is preferred for low mobility scenarios as it optimises throughput. The distributed method provides frequency diversity, thus is advantageous in high mobility scenarios. These allocation schemes can be configured to make use of all available sub-carriers or only a portion of them and are referred to respectively as Fully Used Sub Channelisation (FUSC) and Partially Used Sub Channelisation (PUSC). These concepts are illustrated in Fig. 2.4 with PUSC favouring inter-cell interference averaging [20]. Note that the sub-carrier configuration shown for the distributed method is actually a special case, known as Interleaved Frequency Division Multiple Access (IFDMA).

Fig. 2.5 presents the signal chains for OFDM, OFDMA and SC-FDMA highlighting their respective differences. The OFDM signal chain will be discussed in detail in Section 2.3. In OFDMA, a sub-carrier mapping block is added to the conventional OFDM signal chain which assigns N_{IFFT} data symbols to multiple users and then maps them to N_{IFFT} complex sub-carriers. OFDMA retains the advantages of OFDM in that

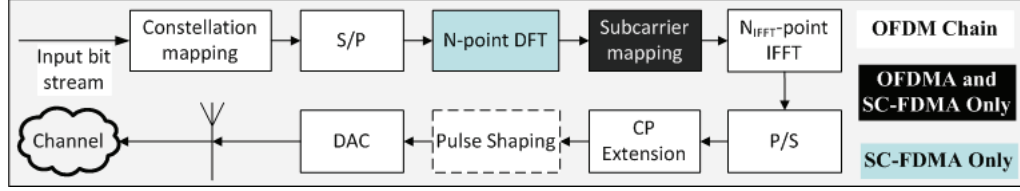


Figure 2.5: Signal chains for OFDM, OFDMA and SC-FDMA.

it provides immunity against multipath fading and narrowband interference with the additional benefits of flexible resource allocation and scalable bandwidths. The latter is achieved by maintaining the same frequency spacing between the sub-carriers and adjusting the Inverse Fast Fourier Transform (IFFT) size to produce scalable channel bandwidths from 1.25 MHz to 20 MHz [69].

In SC-FDMA, the data symbols are pre-coded via a DFT block prior to sending them to the sub-carrier mapping block. This process spreads the symbols over all the available sub-carriers and creates a ‘virtual’ single carrier modulation scheme. This is the reason SC-FDMA is commonly referred to as DFT-spread OFDM. As shown in Fig. 2.5, the N complex amplitudes at the output of the DFT block are transformed into an equivalent complex time domain signal. Ordinarily, the size of the IFFT is larger than the size of the DFT with the ratio $Q = \frac{N_{IFFT}}{N}$, termed the bandwidth expansion factor, signifying the number of simultaneous users the system can handle.

While the underlying waveform in SC-FDMA is single carrier, the scheme is still regarded as multi-carrier, thereby providing the same robustness against multipath fading as OFDMA. Unlike OFDMA though, where data symbols are transmitted in parallel, in SC-FDMA data symbols are transmitted sequentially over a symbol period, as illustrated in Fig. 2.6. This results in lower PAPR compared to OFDM transmission at the expense of higher sensitivity to ISI. Notwithstanding, SC-FDMA has only been introduced in the uplink, where it is presumed that the Base Station (BS) will have sufficient resources to carry out complex Frequency Domain Equalisation (FDE).

Since SC-FDMA has been recommended as a viable solution for reducing the high

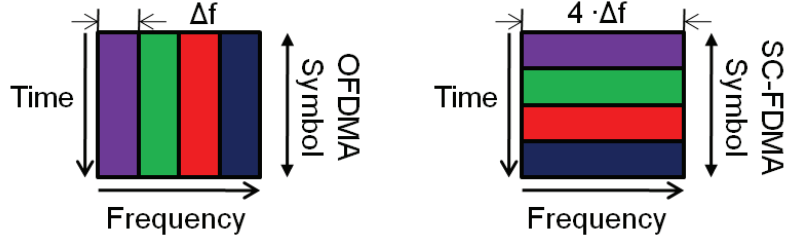


Figure 2.6: OFDMA vs. SC-FDMA data symbol transmission (four sub-carriers).

Table 2.1: Simulation parameters for OFDMA and SC-FDMA.

Parameter	Value
Total number of sub-carriers N_{IFFT}	256 / 512
Used sub-carriers N	16 / 64 / 128 / 256
Modulation scheme	4-QAM / 16-QAM
System bandwidth	5 MHz
Oversampling factor	4
Number of runs	10,000

PAPR present in OFDM systems, this section aims to evaluate its performance and compare it against OFDMA. Results are analysed in terms of the Complementary Cumulative Distribution Function (CCDF) indicating the probability of the PAPR being higher than a specified value $PAPR_0$. Table 2.1 summarises the system parameters employed in the simulations.

Fig. 2.7 compares the CCDF of the PAPR for a different number of transmitted data symbols when OFDMA is employed with the constellation scheme set to 4-QAM and the IFFT size fixed at $N_{IFFT} = 512$. The number of transmitted data symbols is commensurate with the number of used sub-carriers, hence the data sequence is padded with trailing zeros up to the length of the IFFT. From Fig. 2.7, it is clear that the probability of the PAPR being exceeded increases as the number of used sub-carriers is increased within one OFDM symbol. These results are complemented by those in Fig. 2.8 for 16-QAM which show that the PAPR is distributed from approximately 3.5 dB to a maximum value of 11.5 dB depending on the number of used sub-carriers. These latter results also indicate that even though the peak and average powers of 16-QAM

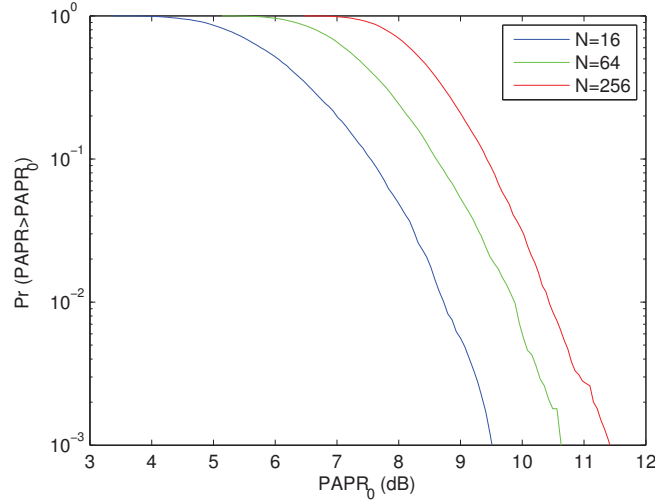


Figure 2.7: CCDF of PAPR for OFDMA for a different number of used sub-carriers ($N_{IFT} = 512$, 4-QAM).

will naturally be higher compared to 4-QAM, the corresponding PAPRs are in fact the same.

Fig. 2.9 compares the CCDF of diverse sub-carrier allocation schemes used in SC-FDMA, namely IFDMA, Localised Frequency Division Multiple Access (LFDMA) and Distributed Frequency Division Multiple Access (DFDMA), employing QAM with different constellation sizes and assuming no pulse shaping is applied. From Fig. 2.9, it is obvious that IFDMA outperforms the other allocation schemes. The reason IFDMA has such a low PAPR is attributed to the fact that the time domain signal for this case is nothing more than a repeating sequence of the original data symbols, hence the PAPR is identical to that for the single carrier case. This can be better comprehended by referring to Fig. 2.10.

In practice, Raised Cosine (RC) or Root Raised Cosine (RRC) filtering is employed to eliminate ISI and reduce out-of-band energy, thus controlling Adjacent Channel Interference (ACI). Fig. 2.11 compares the aforementioned SC-FDMA sub-carrier allocation schemes with different RC pulse shaping roll-off factors β for 16-QAM. From Fig. 2.11, it is evident that the PAPR performance of IFDMA is significantly degraded

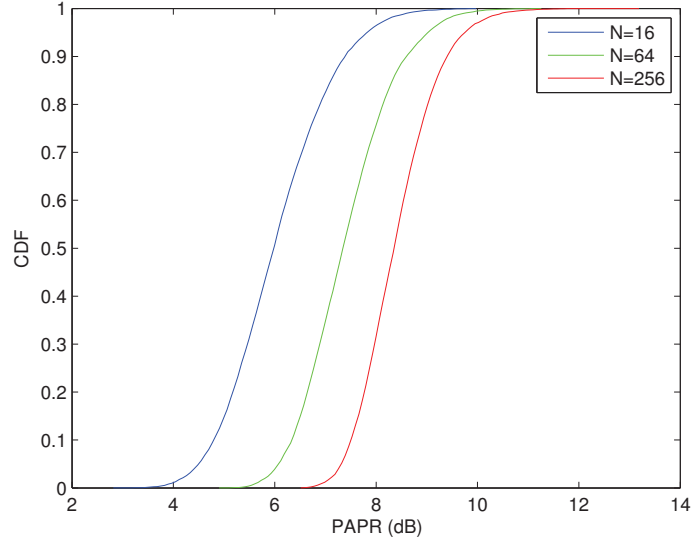


Figure 2.8: Distribution of PAPR for OFDMA for a different number of used sub-carriers ($N_{IFFT} = 512$, 16-QAM).

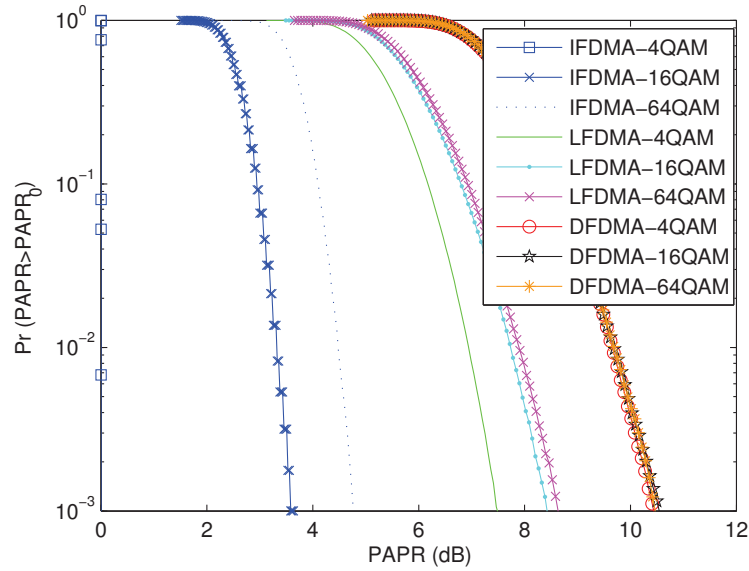


Figure 2.9: CCDF of PAPR for different SC-FDMA sub-carrier allocation schemes and constellation sizes ($N_{IFFT} = 256$).

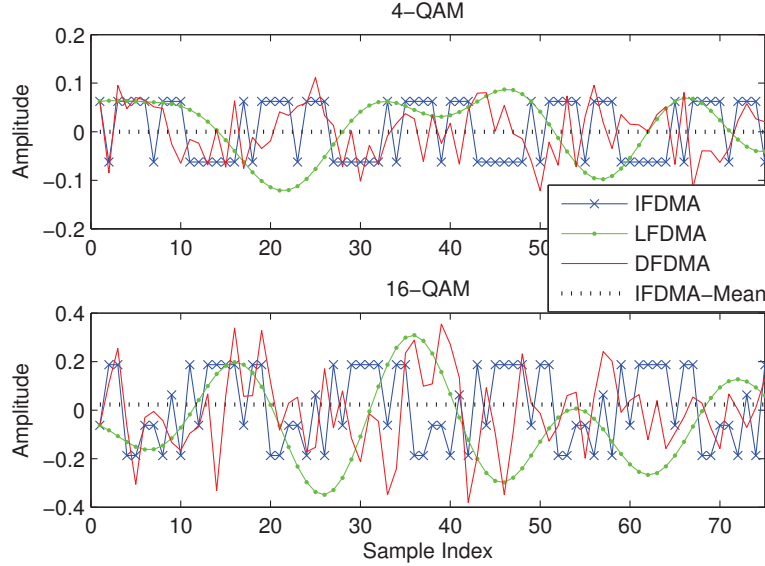


Figure 2.10: Time domain signals for IFDMA, LFDMA and DFDMA ($N_{IFFT} = 256$).

by over 4 dB when pulse shaping is applied compared to the other two schemes where the difference is less than 1.5 dB.

Fig. 2.12 shows the effect of RC and RRC pulse shaping on IFDMA. From Fig. 2.12, it can be inferred that RRC filtering with a roll-off factor of $\beta_{RRC} = 0.5$ provides an optimum trade-off between low PAPR and practical pulse shaping. Fig. 2.13 compares the performance of SC-FDMA-IFDMA and SC-FDMA-LFDMA against OFDMA for a different number of transmitted data symbols or equivalently used sub-carriers N . From Fig. 2.13, it is clear that OFDMA is outperformed in all cases. Moreover, an increase in the number of used sub-carriers has a significant impact on the performance of LFDMA and OFDMA but is marginal for IFDMA.

These observations show that IFDMA outperforms all other schemes in terms of low PAPR but rapidly loses its competitive advantage when pulse shaping is applied. These results are in accordance with the work presented in the literature [82]. DFDMA, which provides the best robustness against multipath fading by allocating sub-carriers in a random manner, exhibits the poorest PAPR performance, as expected and according to the work of the authors in [83]. Finally, LFDMA provides the highest throughput with

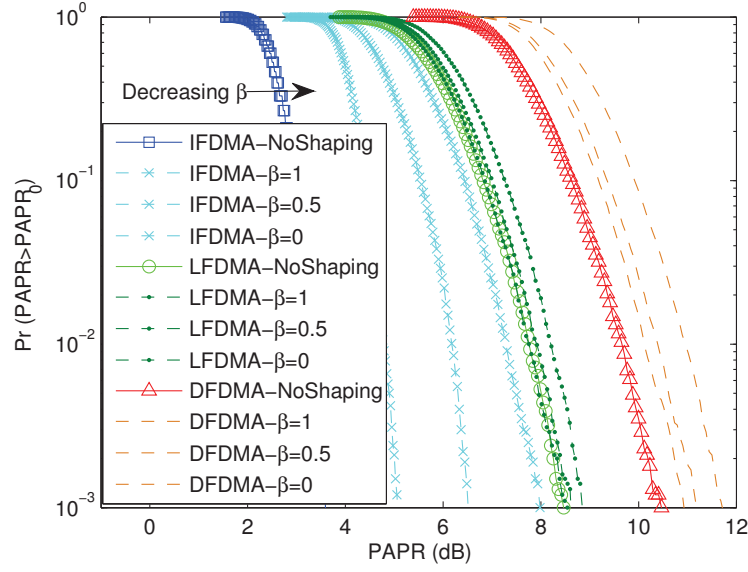


Figure 2.11: CCDF of PAPR for IFDMA, LFDMA and DFDMA with RC pulse shaping ($N_{IFFT} = 256$, 16-QAM).

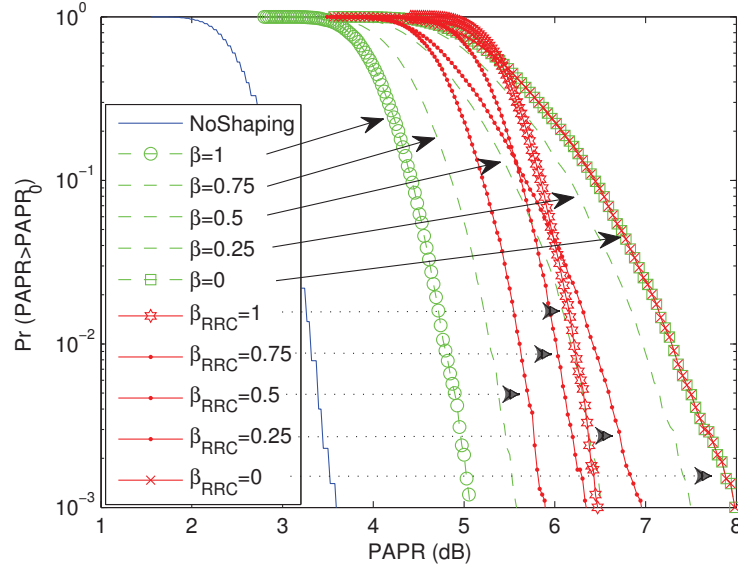


Figure 2.12: CCDF of PAPR for IFDMA with different RC and RRC pulse shaping ($N_{IFFT} = 256$, 16-QAM).

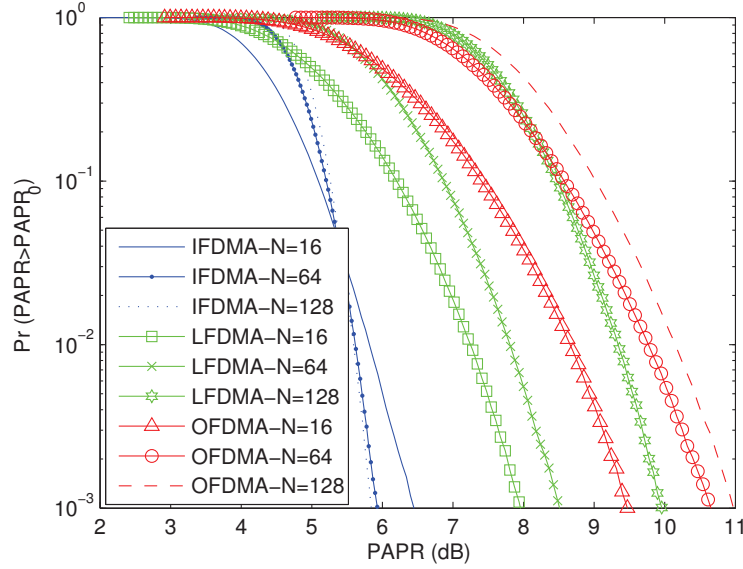


Figure 2.13: CCDF comparison of SC-FDMA ($\beta_{RRC} = 0.5$) against OFDMA ($N_{IFFT} = 256$, 16-QAM).

a negligible degradation in performance when pulse shaping is applied. In summary, the choice of the sub-carrier allocation scheme to be adopted strongly depends upon the application under consideration with trade-offs between low PAPR, high throughput and robustness against the impairments encountered in a multipath propagation channel.

Numerous techniques have been explored with the aim of improving the performance of SC-FDMA, however, these are beyond the scope of this thesis. To mention but a few, the authors in [82] suggest that LFDMA can offer significant throughput gains while improving frequency selectivity through Channel-dependent Scheduling (CDS). Other techniques that have been considered include the use of MIMO in the uplink [84], as well as PAPR reduction methods, such as adaptive digital pre-distortion which is a new feature in LTE [71], clipping, companding and the use of the wavelet transform [85].

2.3 OFDM

As mentioned in the introductory chapter, OFDM is the key enabling technology in 4G mobile networks. OFDM is both a Frequency Division Multiplexing (FDM) scheme and a multi-carrier technique with these two concepts in turn being closely related. Ordinarily, FDM refers to the splitting of a wideband radio into multiple narrowband channels with each channel allocated to a specific frequency range. These channels are non-overlapping, hence multiple users can occupy these channels concurrently without interfering with each other. Multi-carrier transmission refers to the splitting of each channel into several, narrower sub-carriers and in that sense multi-carrier transmission resembles FDM. However, it should be evident that multi-carrier modulation can in fact be considered a sub-set of FDM. For simplicity, henceforth it shall be assumed that the entire system bandwidth is commensurate with a single channel and that FDM refers to the multiplexing of the sub-carriers in that channel.

In a conventional multi-carrier system, the signal bandwidth is divided into parallel, non-overlapping sub-carriers with Guard Bands (GBs) placed in between the individual sub-carriers. Each sub-carrier is modulated separately with a data symbol followed by the summation of all modulated sub-carriers to generate the time-domain signal. While this configuration eliminates ICI, it makes inefficient use of the available spectrum [86]. This is the reason the introduction of OFDM in the 1960's [7] revolutionised multi-carrier systems, as it maximised spectral efficiency² through the use of overlapping but non-interfering sub-carriers. A comparison of conventional FDM and OFDM is illustrated in Fig. 2.14.

In a conventional FDM system, the frequency spacing between the sub-carriers is such that it obeys Nyquist's first criterion, also known as the Nyquist ISI criterion, which is the condition for zero ISI between transmitted symbols [87]. In OFDM, this spacing is equal to the reciprocal of the OFDM symbol period, which in turn is equal

²The terms *spectrum efficiency*, *spectral efficiency* and *bandwidth efficiency* will be used interchangeably throughout this thesis to signify the optimisation of radio resources.

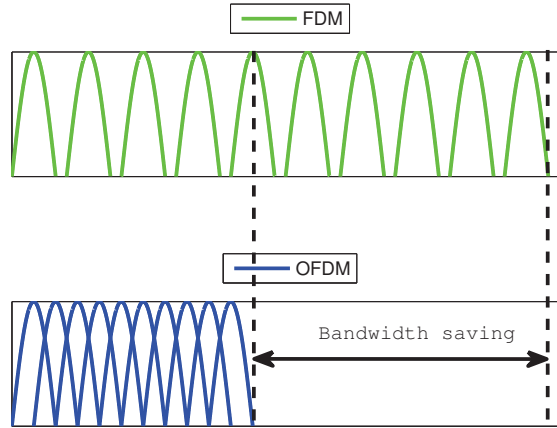


Figure 2.14: Conventional FDM vs. OFDM.

to the identical transmission speed of each sub-carrier. This relationship between the sub-carriers' frequency spacing and the OFDM symbol's period is at the core of OFDM systems, as it guarantees orthogonality between the sub-carriers. This orthogonality rule is paramount as it ensures that there is no cross-talk between the sub-carriers. Hence, this concept will be expounded throughout the remainder of this section.

2.3.1 Basic Principles

The compelling advantage of employing OFDM over single carrier systems is its increased robustness against the impairments that occur in frequency-selective and time-dispersive channels. Another acute aspect is that the classical method of generating and demodulating OFDM signals using a bank of modulators-demodulators (modems) can be replaced by a far more efficient implementation incorporating FFT operations. Consequently and not surprisingly, a number of hybrid systems have come to light over the past decade with the most notable examples being Flash OFDM (fast-hopped OFDM), Vector OFDM (OFDM with MIMO) and Wideband OFDM, the latter designed to introduce enough spacing between the channels to mitigate frequency errors [88], as well as multi-carrier (MC)-CDMA, which is Direct Sequence Spread Spectrum

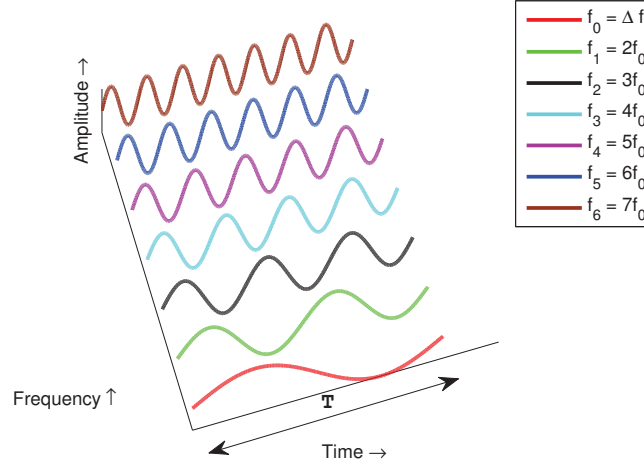


Figure 2.15: Time-frequency representation of OFDM sub-carriers.

(DSSS)-CDMA followed by OFDM [89]. Finally, Coded OFDM (COFDM) combines OFDM with Forward Error Correction (FEC) to correct sub-carriers which undergo deep fading [90].

OFDM operates by transmitting a high data rate input data stream over multiple lower data rate parallel streams. This configuration achieves two related objectives concurrently; first, the narrowband nature of the sub-carriers means that the fading experienced by each one can be considered to be approximately flat. This alleviates the need for complex equalisation rendering OFDM attractive for wireless and mobile applications; second, the duration of the OFDM symbol increases in proportion to the number of sub-carriers employed, thus providing a greater tolerance against the channel's delay spread.

OFDM can sustain the aforementioned advantages only if the orthogonality rule is obeyed at all times. This rule can be expressed in different, nonetheless equivalent ways:

- The frequency spacing Δf between the overlapping sub-carriers has to equal the reciprocal of the OFDM symbol period T .

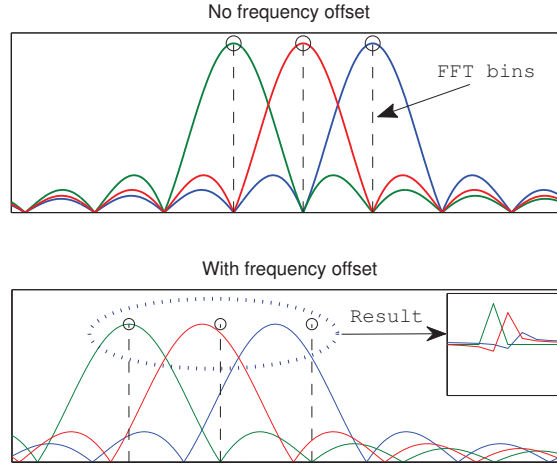


Figure 2.16: FFT bin locations in the absence and presence of ICI.

- The frequency of each sub-carrier has to equal an integer multiple of the frequency spacing Δf , as illustrated in Fig. 2.15. Equivalently, each sub-carrier must have an integer number of cycles within the OFDM symbol period T .
- The length of the FFT window applied during the demodulation stage and the length of the OFDM symbol period T must be commensurate. This ensures that the FFT bins are aligned perfectly to the peaks and the nulls of the corresponding sub-carriers.

Practical issues, however, render OFDM susceptible to many sources of interference which could destroy the orthogonality, thus hindering the successful recovery of the transmitted symbols. The Doppler effect, the phase noise associated with Local Oscillators (LOs), as well as mismatches between the transmitter and receiver oscillators may lead to frequency offsets. Such offsets alter the actual frequencies of the underlying sub-carriers and may cause a non-integer number of cycles to appear in the FFT window. This leads to loss of orthogonality giving rise to ICI, as a result of the FFT bins not lining up with the peaks and zero-crossings of the overlapping sub-carriers, as depicted in Fig. 2.16.

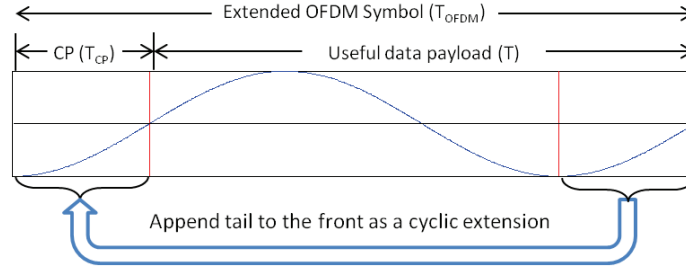


Figure 2.17: A cyclically extended OFDM symbol.

Another impairment which signals suffer during their propagation over a wireless channel is the spreading of their symbol duration. This elongation causes symbols to smear into adjacent ones giving rise to ISI. If this foreign content is present in the FFT window of the symbol to be reconstructed, the output of the demodulation process will not be correct.

2.3.2 Mitigating ICI and ISI

OFDM employs specific techniques to counteract the aforementioned impairments. First, it uses LOs with very low jitter and tight synchronisation to correct frequency errors, thus mitigating ICI. Second, it uses a GI between successive OFDM symbols to protect the content of each symbol. This GI is best induced by copying a section of the symbol's tail and appending it to the front of the symbol in a cyclic manner making the extended symbol appear periodic in time. This extension is known as a CP and the concept is illustrated in Fig. 2.17.

The CP has a multi-fold purpose; first, it protects the OFDM against ISI; second, it offers an extra degree of freedom in positioning the FFT window to avoid any overlap with the preceding or the subsequent OFDM symbol, given that the length of the FFT window is now smaller than the total symbol duration T_{OFDM} which includes the useful OFDM data, of length T , and the CP, of length T_{CP} ; third, it ensures the signal appears periodic in the FFT window having an integer number of cycles in accordance with the orthogonality rule. This would not have been the case if the GI had been

implemented as zero padding. This periodicity also converts the convolution with the channel response from linear to circular, thus allowing FFT operations to be employed in the frequency domain since multiplication in the frequency domain translates to circular convolution in the time domain [91]. Linear and circular convolution may be mathematically expressed as follows:

Linear convolution:

$$y[n] = h[n] * x[n] = \sum_{k=0}^{N_h-1} h[k]x[n-k], \quad n = 0, \dots, N_x + N_h - 1. \quad (2.1)$$

Circular convolution:

$$y[n] = h[n] \otimes x[n] = \sum_{k=0}^{N_h-1} h[k]x[n-k]_{N_L}, \quad n = 0, \dots, N_L, \quad (2.2)$$

where $x[n]$ is the channel input, $h[n]$ is the channel response and $y[n]$ is the output generated from the convolution of the input with the channel response. The lengths of the input and channel response are denoted by N_x and N_h , respectively, while N_L corresponds to the longest length, for example, $N_L = N_x$ if $N_x > N_h$. In linear convolution, both sequences are padded with zeros up to the length of the sum of their lengths, in other words $N_x + N_h$. In circular convolution, the output is as long as the longest sequence with the input wrapping round. For example, if $z = n - k$ and $z < 0$ then z is set to $z = n - k + N_L$.

Based on the above, it can be observed that there are conflicting trade-offs when selecting the number of sub-carriers to be employed in an OFDM system. On the one hand, increasing the number of sub-carriers provides greater immunity against frequency-selective fading and delay spread. On the other hand, assuming the same data rate is maintained, increasing the number of sub-carriers translates to a narrower frequency spacing rendering the system more sensitive to frequency offsets and thus ICI. For a small number of sub-carriers, the CP provides added tolerance against ISI, however, this comes at the expense of reduced bandwidth efficiency.

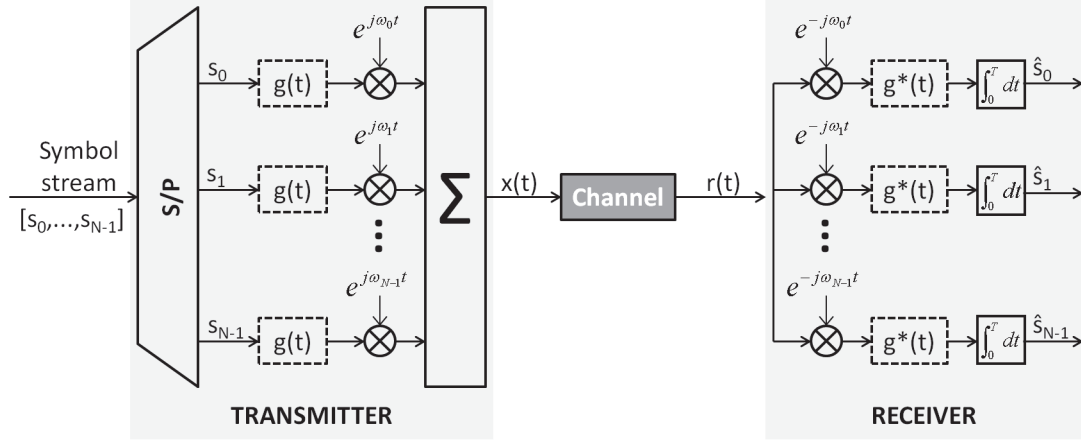


Figure 2.18: Architecture of an OFDM transceiver employing a bank of modems.

2.3.3 Parallelism, Orthogonality and the DFT

The previous subsections alluded to three key characteristics of OFDM systems which call for closer examination, namely parallelism, orthogonality and the use of DFT operations.

Parallelism refers to the basic principle of OFDM in dividing a serial input symbol stream into a number of equally spaced parallel sub-carriers each of which has a data rate R equal to the incoming data rate R_{sym} divided by the number of sub-carriers N . Reciprocally, the duration of the resultant multiplexed OFDM symbol is increased by the same factor N with respect to the period T_{sym} of the incoming data, in other words $T = NT_{sym}$. It is this increased duration of the OFDM symbol which provides greater tolerance to multipath delay spread. A typical architecture of an OFDM transceiver employing a bank of modems, similar to the one proposed in the late 1960's [8], is shown in Fig. 2.18.

It can be seen that a typical OFDM transmitter consists of three stages, namely a pulse shaping stage, a modulation stage and a summation stage. In OFDM, the pulse shaping function is typically a rectangular function $rect\{\cdot\}$ with width T and is defined

as

$$\text{rect}_T(t) \triangleq \begin{cases} 1 & \text{if } -\frac{T}{2} \leq t \leq \frac{T}{2}, \\ 0 & \text{otherwise.} \end{cases} \quad (2.3)$$

The serial symbol stream, which represents the information sequence is given by

$$s(t) = \sum_{k=-\infty}^{+\infty} s_k g(t - kT_{sym}), \quad (2.4)$$

where s_k represents the complex data symbols indexed by the integer variable k , $g(t)$ is the pulse shaping function, T_{sym} is the period of one data symbol equal to the reciprocal of the data symbol rate R_{sym} and ∞ denotes infinity. In OFDM, this sequence of data symbols is split into frames. During the modulation stage, the data symbols are multiplied by an exponential function $e^{j\omega_n t}$, where j is an imaginary unit defined as $j = \sqrt{-1}$ and $\omega_n = 2\pi n\Delta f$ represents the sub-carriers' angular frequencies. The modulated symbols of each frame are then multiplexed yielding an OFDM signal, which may be expressed as

$$x(t) = \sum_{l=-\infty}^{+\infty} \sum_{n=0}^{N-1} s_{l,n} g(t - lT) e^{j\omega_n t}, \quad (2.5)$$

where $l \in \mathbb{Z}$ denotes the frame or OFDM symbol index, $n \in \mathbb{Z}_{\geq 0}$ denotes the sub-carrier index and $s_{l,n}$ represents the information symbol conveyed by sub-carrier with index n during the time stamp with index l .

The next distinctive feature of OFDM signals is their inherent orthogonality. As noted earlier, the sub-carriers in OFDM overlap with each other, thus eliminating the need for GBs and favouring better utilisation of the available spectrum. In order to allow this overlapping of the sub-carriers without causing interference, the frequency spacing between them has to equal exactly the reciprocal of the OFDM symbol period, in other words $\Delta f = \frac{1}{T}$. A direct consequence of this rule is that the frequency of each sub-carrier is a multiple of the frequency spacing. Consequently, given that the lowest frequency f_0 (excluding the DC component) will equal the frequency spacing Δf , all subsequent higher frequencies will be harmonic to f_0 . This statement also equates

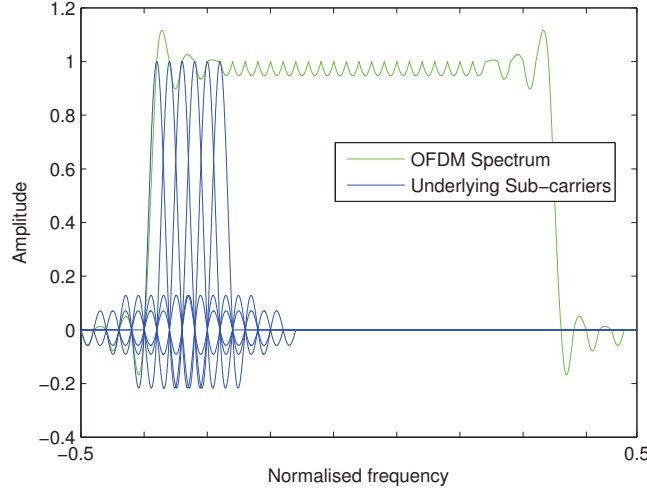


Figure 2.19: Spectrum of an OFDM signal and (a fraction of) its underlying sub-carriers in theory.

to each sub-carrier having an integer number of cycles within T , meaning that the area over one symbol period will equal zero, which translates to the sub-carriers being orthogonal between them [92]. Moreover, the exponential function used to modulate the incoming data symbols consists of sine and cosine components, which by default constitute a set of orthogonal basis functions.

Fig. 2.19 illustrates the theoretical spectrum of an unmodulated OFDM signal along with a portion of its underlying sub-carriers. Since a rectangular pulse is applied to each of the incoming data symbols, the resultant spectrum will be overlapping sinc spectra of the individual sub-carriers. Thanks to the orthogonality rule, the peak of each sub-carrier lands exactly at the zero-crossings of all the other sub-carriers, hence avoiding ICI.

In practice, the spectrum of a modulated OFDM signal has a more abrupt look, as depicted in Fig. 2.20. It should be evident that at the maximum amplitude of each sub-carrier the contribution from all other sub-carriers is zero, hence the reason an OFDM receiver can demodulate each sub-carrier free from any cross-talk [86]. As illustrated in Fig. 2.18, at the receiver a bank of demodulators downconvert the received signal

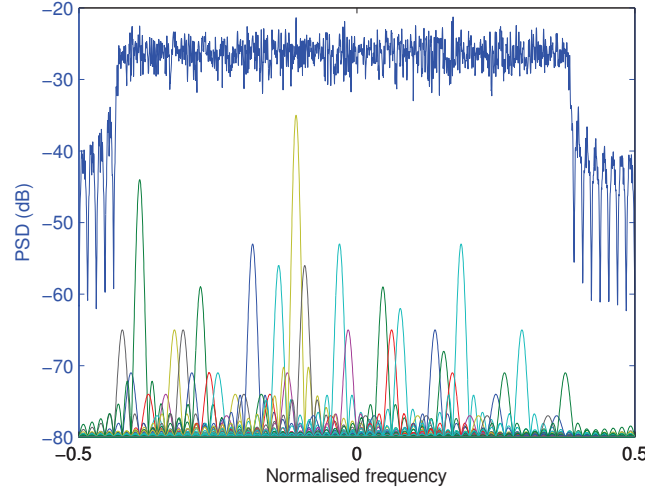


Figure 2.20: Power spectrum of a modulated OFDM signal and its underlying sub-carriers in practice.

to DC. Each sub-carrier is multiplied by the corresponding frequency and the product is integrated over one symbol period to recover the data from that sub-carrier. The integration process gives the estimated symbol value for that sub-carrier only since the result of the cross-correlation with all the other sub-carriers will equate to zero, in accordance with the orthogonality rule. In this sense, the modulation and demodulation processes in OFDM resemble the code spreading and despreading operations in CDMA.

The aforementioned discussion regarding the orthogonality in OFDM systems revealed one of its major advantages in that it alleviates the need for complex equalisation at the receiver. Nonetheless, a corollary of employing orthogonal basis functions is that it allows the use of DFT operations, as was initially observed in the 1970's [10]. To understand this relationship between OFDM and the DFT, the discrete-time equivalent of the continuous OFDM signal defined in Eq. 2.5 is generated. First, the bandwidth of an OFDM signal is approximately equal to $N\Delta f$ and as shown in Figs. 2.19 and 2.20, it is defined between $-0.5f_s$ and $0.5f_s$. Hence, assuming critical sampling is employed, the continuous signal $x(t)$ is sampled at a sampling frequency $f_s = N\Delta f$, which is equivalent to $\Delta f = \frac{f_s}{N} = \frac{1}{NT_s} = \frac{1}{T}$, where T_s is the sampling period with $T_s = \frac{1}{f_s}$.

Subsequently, the sampled version of $x(t)$ may be expressed as

$$x_s(t) = \sum_{m=-\infty}^{\infty} x(m)\delta(t - mT_s), \quad (2.6)$$

where $x(m)$ represents the discrete sample values indexed by $m \in \mathbb{Z}$. Considering a single discrete-time OFDM symbol, $x(m)$ is given by

$$x(m) = \sum_{n=0}^{N-1} s_n e^{j2\pi n \Delta f t} = \sum_{n=0}^{N-1} s_n e^{j2\pi n \frac{f_s}{N} m T_s} = \sum_{n=0}^{N-1} s_n e^{j2\pi \frac{nm}{N}}, \quad (2.7)$$

where $x(m) = x(0), \dots, x(N-1)$ represents the time samples making up one OFDM symbol. It is clear that Eq. 2.7 resembles the Inverse Discrete Fourier Transform (IDFT) synthesis equation given by

$$x(m) = \frac{1}{N} \sum_{n=1}^N X(n) e^{\frac{j2\pi(n-1)(m-1)}{N}}, \quad (2.8)$$

with the DFT counterpart given by

$$X(n) = \sum_{m=1}^N x(m) e^{-\frac{j2\pi(n-1)(m-1)}{N}}, \quad (2.9)$$

thereby demonstrating the way in which DFT operations may be used to replace their analogue equivalents. The DFT is what made the practical implementation of OFDM feasible since:

- It allows the entire modulation and demodulation stages to be implemented digitally alleviating the need for banks of modems which are expensive and require tight and accurate synchronisation, thus hindering scalability.
- It simplifies the generation and recovery of OFDM signals as the IDFT and DFT operations form a matched filter pair.
- The IDFT and DFT operations can be replaced by more efficient realisations

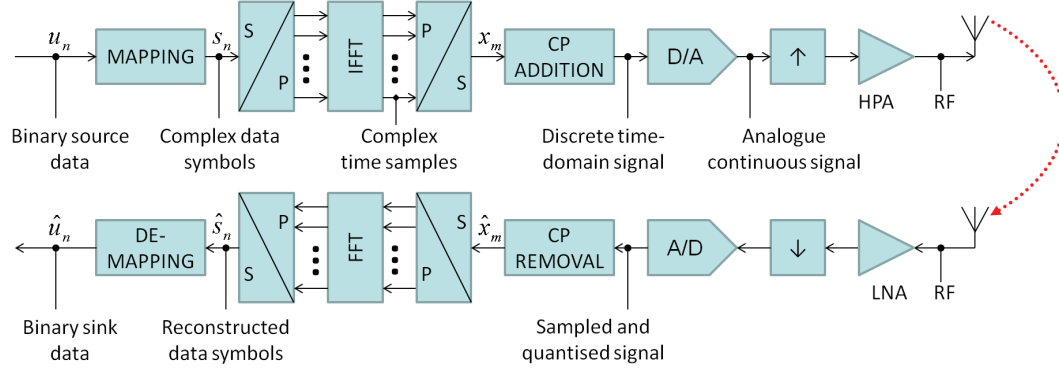


Figure 2.21: Simplified diagram of a practical OFDM system.

employing the FFT, hence yielding a considerable reduction in computational complexity.

- Processing is carried out at baseband obviating band-pass filtering while concurrently relaxing sampling rate constraints.

Fig. 2.21 shows an OFDM system comprising typical blocks that would be present in a practical scenario. For clarity, certain stages have been omitted, such as de-/coding, de-/interleaving, pilot insertion and channel estimation and correction. Fig. 2.21 illustrates that the multiplication and summation stages have been substituted by a single IFFT process simplifying and favouring a more efficient implementation of OFDM transmission compared to the conventional technique shown in Fig. 2.18.

Based on Fig. 2.21, the input bit stream is mapped initially to a constellation scheme to generate complex data symbols. First, the bit values u_{n_i} are grouped to generate the integer values d_n according to

$$d_n = \sum_{i=1}^{\log_2 M} u_n(i) 2^{\log_2 M - i}, \quad (2.10)$$

which are then mapped to a constellation diagram to yield the complex data symbols s_n . M denotes the constellation cardinality or equivalently the number of bits per symbol.

In the next stage, a Serial-to-Parallel (S/P) converter splits the data sequence into a number of parallel streams equal to the number of sub-carriers employed. The IFFT block is responsible for modulating the sub-carriers according to the incoming data symbols generating a discrete time domain signal at its output as per Eq. 2.7. The elements of this block are given by $e^{j2\pi \frac{nm}{N}}$ yielding an $N \times N$ matrix, where m denotes the row index and n denotes the column index. This matrix represents the unmodulated sub-carriers, where the rows and columns correspond to the complex time samples and frequencies, respectively. Since FFT operations are being employed, it is assumed that the size of the IFFT block is a power of two, hence this matrix gives rise to $\frac{N}{2} - 1$ unique frequencies plus the DC component with the remaining $\frac{N}{2}$ frequencies forming complex conjugate frequencies [91]. Consequently, the IFFT signal generation process obeys the Nyquist criterion since N time samples are produced with the highest frequency equal to $(\frac{N}{2} - 1)\Delta f$.

Ordinarily, the size of an IFFT matrix is commensurate with the number of incoming data symbols, the number of sub-carriers and the number of time samples of the resultant time domain OFDM signal. The sub-carriers are modulated in amplitude and/or phase by the incoming data symbols, which typically originate from a Phase Shift Keying (PSK) or QAM constellation. Having generated the discrete time domain samples, a Parallel-to-Serial (P/S) converter combines them into a serial stream which is then cyclically extended with a prefix. At this stage, the discrete-time baseband OFDM signal has been obtained. Fig. 2.22 gives an example of an unmodulated and modulated time domain OFDM signal along with the underlying individual unmodulated and modulated sub-carriers. Note that only the first five sub-carriers are shown in the top sub-plots for clarity of presentation.

A D/A converter is then used to translate the digital values into analogue voltages while a Low-pass Filter (LPF) is subsequently applied to smooth out sharp phase transitions. The signal is then upconverted and amplified using a High-power Amplifier (HPA). The receiver performs the reciprocal transmitter operations. A Low-noise

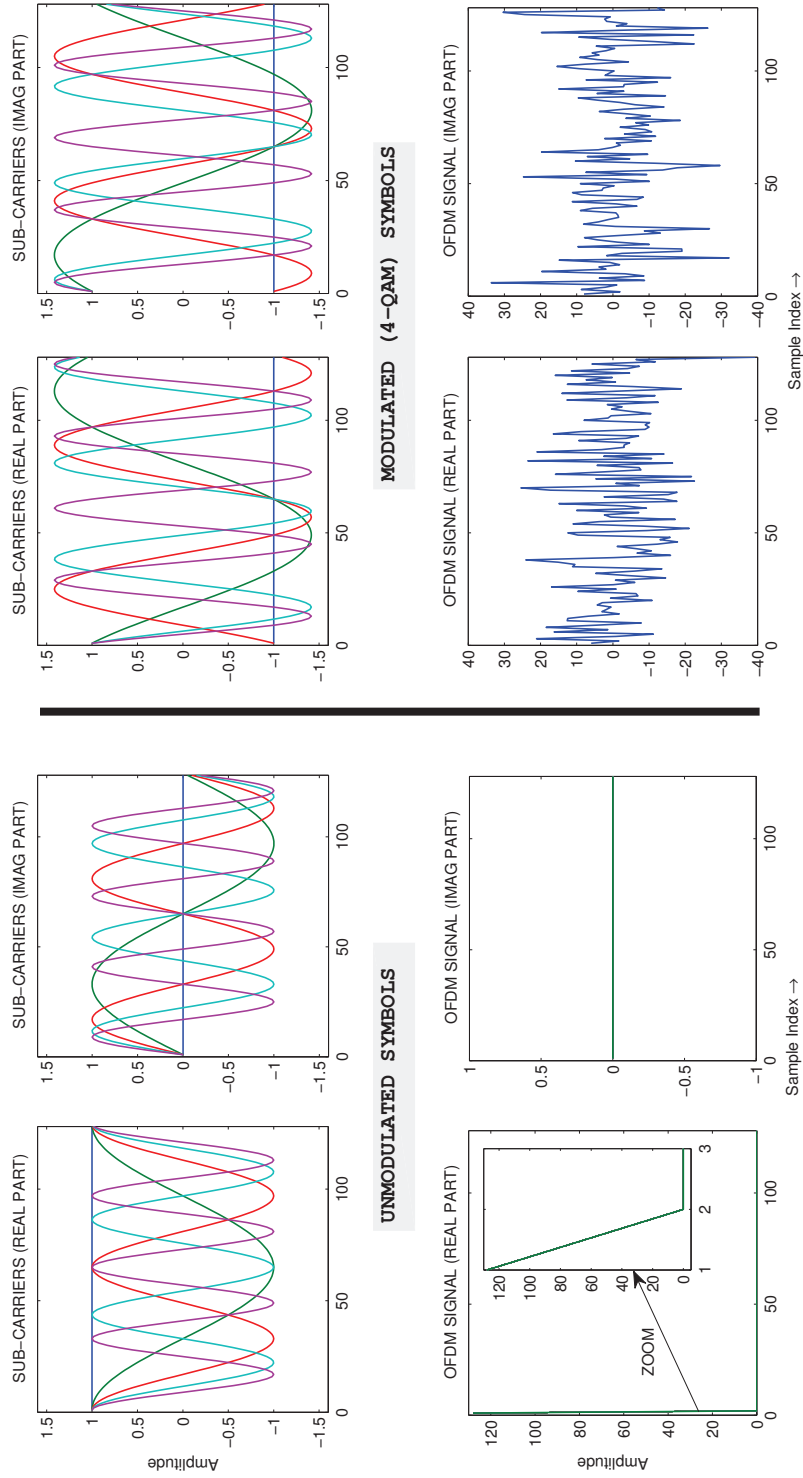


Figure 2.22: Individual sub-carriers (top) and subsequent OFDM signal (bottom) for unmodulated (left) and modulated (right) scenarios ($N = 128$).

Amplifier (LNA) is used instead to reduce the noise floor while an additional block responsible for time and frequency synchronisation (not shown in Fig. 2.21) is indispensable to ensure correct recovery of the transmitted symbols. Two important operations which have not been added to the signal chain in Fig. 2.21 for simplicity include pulse shaping and quantisation. Pulse shaping serves to reduce out-of-band radiation when a small number of sub-carriers is employed. With regard to quantisation, in a practical realisation the output of the IFFT would be automatically quantised, whereas this is not the case in a simulation environment. Quantisation and fixed-point effects will be explained in detail in subsequent chapters.

In this section, the issue of multipath fading and its effects on signal transmission was discussed. While an important topic, it is not feasible to deal with all its issues in this thesis. Furthermore, this work focuses on the practical implementation and performance of advanced detection techniques which could potentially make SEFDM feasible in the near future, in the same way the DFT was the catalyst for the success of OFDM in modern communication systems. Hereby, in this thesis only an AWGN channel will be considered, since this is the most common form of noise in communication systems [93]. AWGN has the same impact on an OFDM system as for a single carrier system. Consequently, the BER of an OFDM signal is identical to the BER of the underlying constellation scheme.

2.4 Conclusions

This chapter started with a description of the key technologies that will be employed in NGNs. These technologies, which are being introduced in the latest WiMAX and LTE products, are well-placed to deliver a wide range of data rates which will scale according to the spectrum allocated and the antenna configuration used. The main requirements are enhanced throughput, reduced power, flexible bandwidth allocation and increased spectrum efficiency.

The second part of this chapter addressed one of the key drawbacks associated

with OFDM signals, namely their PAPR. The simultaneous transmission of a large number of sub-carriers configured in parallel in OFDM systems leads to a high PAPR. Consequently, there is a complex trade-off between the use of highly linear PAs and IMD. SC-FDMA, as proposed by 3GPP for use in the uplink of LTE systems, was therefore examined as a solution to the PAPR problem in OFDM. The performance of SC-FDMA was compared against that of OFDMA taking into account that the latter will be employed in WiMAX systems in both the downlink and uplink directions.

The final section provided an account of OFDM. The concept and mathematical principles of this modulation scheme were thoroughly reviewed. This analysis emphasized the three tricks employed in OFDM which led to its successful integration in modern communication systems, namely parallelism, orthogonality and the use of FFT operations. The parallelism feature offers robustness against multipath fading and channel delay spread. The orthogonality principle allows for overlapping between sub-carriers, hence maximising the use of the available spectrum while guaranteeing an interference-free transmission. Finally, the replacement of conventional bank of modems by efficient FFT signal processing offers a significant reduction in complexity while allowing the entire baseband chain to be implemented digitally.

The ever-increasing demand for higher data rates and improved coverage has called for a change of regime with the ultimate goal of optimising spectrum utilisation. Hence, Chapter 3 provides an overview of CR techniques and hybrid methods combining CR with traffic prediction with the aim of improving signal detection and QoS for CR users. In subsequent chapters, a novel multi-carrier communication technique termed SEFDM is examined since it promises significant bandwidth savings at the expense of receiver complexity. SEFDM was developed to address the limits imposed upon spectral efficiency in OFDM systems. The use of a CP in OFDM has a direct impact on the maximum achievable spectral efficiency. Furthermore, the loss of orthogonality due to ICI leads to severe performance degradation rendering OFDM useless if not warranted for carefully. The performance, complexity and practical feasibility of sophisticated al-

gorithms developed for generating and recovering SEFDM signals is evaluated. SEFDM offers an additional degree of freedom while the algorithms are capable of mitigating the self-created ICI, hence offering trade-offs in complexity and performance for diverse bandwidth compression rates.

Chapter 3

Cognitive Radio and Dynamic Spectrum Access

As discussed in Chapter 1, the central theme of this thesis, alas a major concern in future wireless networks, is that of spectrum availability. Radio spectrum is sparse and is currently not being utilised adequately. This calls for spectrum management which considers the efficient allocation and sharing of the available radio resources. Consequently, one of the burning research topics in which many researchers are engaged is DSA, which exploits the existing wireless spectrum opportunistically [94].

One of the mainstream technologies enabling DSA is CR. CR allows a user to access dynamically discontinuous frequency bands and make use of idle channel resources, referred to as *spectrum holes* or *white spaces* [94], thereby increasing spectral efficiency and subsequently system capacity. Hence, this chapter commences by exploring the basics of CR technology.

Since CR is based on SDR, Section 3.1 explains the concepts of SDRs, the latter being totally programmable in terms of their components and functionalities including frequency ranges, channel access modes and channel modulation [95]. Section 3.2 examines different spectrum sensing techniques with an emphasis on cyclostationarity based detection. The discussion shows that exploiting the statistical properties

of signals with inherent cyclostationarity, constitutes a low complexity technique for accurate signal detection and system identification. Section 3.3 details novel CR modulation techniques aimed at mitigating interference through sidelobe suppression and pulse shaping. Finally, Section 3.4 describes the application of traffic prediction techniques to CR systems with the purpose of improving the QoS for CR users. It is shown that the proposed solution offers performance gains in terms of throughput and packet loss ratio.

A review of CR and the use of cyclostationarity based techniques in communications applications was carried out by the author and presented at the London Communications Symposium (LCS) in 2009 [96]. The research in Section 3.4 was presented at the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC) also in 2009 [97]. An extended version of this work was published in the IEEE Transactions on Vehicular Technology (TVT) in 2010 [98].

3.1 Cognitive Radio Basics

SDR can be viewed as the physical means for changing between different frequencies, whereas CR can be seen as the intelligent processing added to the radio front-ends to make the decisions. Hereby, CR devices may be regarded as *reconfigurable devices with frequency-agile front-ends capable of dynamically adapting their transmission parameters to obtain the “best available spectrum”* [94] [99] while satisfying certain criteria, such as link reliability and sustained transmission rate.

One of the first SDR initiatives was the Speakeasy Program [100] designed for military purposes. Key to this programme was the software reprogrammability of waveform processing achieved by using a modular and self-reconfigurable architecture. SDRs can be seen as an amalgamation of several different technologies including multi-band technology, multi-mode radios and programmable baseband processing [101]. Such devices should have agile front-ends capable of supporting multi-band antennas and wideband digital and analogue converters with the baseband functions implemented in software

using general-purpose programmable processors.

SDR devices are well-positioned to meet important requirements in modern communication systems like interoperability and flexibility. One of the aims is to seamlessly integrate multimedia services via increased programmability and joint control functions. The ultimate goal is to render SDRs fully autonomous with ‘plug-and-play’ capabilities, enabling them to enhance their features through regular software updates. A direct benefit of realising more radio components in software, is the reduced size of the corresponding hardware [37].

The notion of CR was originally introduced by Mitola [102] and was used to describe the intelligence of a radio device capable of sensing its environment, making decisions based on observations and learning from experience to improve future decision making. CRs may be identified through seven intrinsic characteristics, namely *awareness*, *intelligence*, *learning*, *adaptivity*, *reliability*, *efficiency* and *reconfigurability* [95]. Reconfigurability is taken care of by using SDR as the physical hardware for carrying out CR tasks. The remaining characteristics may be realised thanks to the significant advances in signal processing and machine learning.

The original reasons that triggered the development of CR was the need for better spectrum utilisation and the inclination to allow different systems with incompatible modulation schemes to coexist in the same frequency regions. During the last decade, this objective has been narrowed down to allowing CR users (also known as secondary users) to access radio channels which are temporally unused by licensed users (also known as primary or incumbent users). Additional spectrum has been released recently due to the switchover from analogue to digital broadcasting. This vacant spectrum has been designated as Television White Spaces (TVWS) and is intended to be made available to unlicensed broadband wireless devices [103]. The unique combination of bandwidth and coverage offered by TVWS renders them an attractive option for CR applications [38]. Evidently, CR users should be able to detect and make use of both incumbent and miscellaneous opportunistic signals.

To this end, the fundamental tasks pertaining to a CR device include *radio analysis*, *predictive modelling* and *dynamic spectrum management*, which in turn correspond to three crucial stages involved in a CR cycle, namely *sensing*, *feedback* and *action* [95]. Typical CR networks will comprise diverse wireless access systems spanning a wide range of discontinuous frequency bands. Hence, a CR user has to intelligently search for idle channel resources and exploit them through DSA without interfering with the communication of licensed users.

Thereby, CR devices are faced with conflicting objectives. On one hand, the CR device should be powerful enough to detect vacant frequency slots and fast enough to modify its transceiver parameters to take advantage of this opportunistic spectrum availability. On the other hand, this opportunistic spectrum access should not interfere with the transmission of the primary users. As a result, the radio front-end of the CR device should be sensitive enough to detect the weakest signal and concurrently minimise the probability of error. To jointly optimise interference control and dynamic reuse of the spectrum, a cross-layer approach entailing *spectrum sensing, analysis and decision* is necessitated [103].

Spectrum management at the physical layer is achieved through spectrum sensing. Spectrum analysis and decision are dealt with at higher layers. A major challenge for the former is the detection of weak signals over a wide spectrum range. This is particularly true in view of the *hidden node* problem that may arise in CR networks [38]. Novel techniques to tackle this challenge are discussed in Sections 3.2 and 3.3, whereas Section 3.4 proposes a solution to yield better decisions at higher layers. As with any communication system, trade-offs exist between i) the complexity of the solution, ii) the speed at which the CR device can sense its environment and modify its transmission configuration accordingly, and iii) the probability of error.

3.2 Cyclostationarity Based Detection

The objective in CR networks is the efficient and fair allocation of licensed spectrum while concurrently mitigating any interference caused to primary users [94]. The key obstacles that arise in meeting this objective are the detection of weak signals over a wide dynamic range and the ‘hidden node’ problem [38]. A number of solutions have been proposed to address these issues with spectrum sensing and geolocation databases widely accepted as the most promising candidates. Telecommunications regulators recognise that geolocation is the most important mechanism for protecting licensed users [38]. Nevertheless, to increase the probability of detecting a signal correctly, spectrum sensing has to be employed in conjunction with geolocation.

Spectrum sensing techniques may be categorised into transmitter detection, cooperative detection and interference based detection [94]. Cooperative transmission comes at the expense of a large overhead, subsequently increasing design costs and system complexity [38]. Interference based methods rely on measuring the *interference temperature*, a metric introduced by Haykin [95], yet questions arise as to the accuracy of the model employed to measure this temperature [94]. Hence, transmitter detection is currently the preferred method for spectrum sensing. Transmitter detection techniques can be further divided into three methods, namely *energy detection*, *matched filter detection* and *cyclostationary feature detection* [94].

Although energy detection is the simplest method, it is generally avoided due to its high sensitivity to noise uncertainty. Matched filter detection is the optimum detector in AWGN, however, it requires a priori knowledge of the primary user signal [94]. The best spectrum sensing performance is achieved by sophisticated, yet low complexity, algorithms, which exploit the cyclostationary properties of incumbent signals and are robust to the uncertainty in noise power. For this reason, cyclostationary feature detection is examined in more detail in subsequent sections.

3.2.1 Principles of Cyclostationarity

Cyclostationarity is a key technique for the detection of signals in CR and other systems which rely on statistical properties to extract highly corrupted signals [96]. These properties are determined by the physical layer scheme employed. In CR applications, the aim is to configure the transmitted signals in such a way, that their features are unique with respect to primary and secondary systems [104].

Landmark work on the theory of cyclostationarity and its applications has been carried out by Gardner [105]. According to Gardner, cyclostationarity arises in signals which exhibit an underlying periodicity. A signal exhibits cyclostationarity “if and only if the signal is correlated with certain frequency-shifted versions of itself” [106]. Exploiting these cyclostationary features allows the recovery of the signal when it is subjected to noise and interference.

From a mathematical perspective¹, a signal is cyclostationary if its non-linear transformation can generate spectral lines, in other words finite-strength additive sinusoidal components, at some non-zero frequencies [106]. For example, if the non-linear transformation of a signal $x(t)$ is denoted by $y(t)$, then $x(t)$ is cyclostationary if $\langle y(t)e^{j2\pi\alpha t} \rangle \neq 0$, for $\alpha = \frac{m}{T}$, where α denotes the non-zero cyclic frequencies at which spectral lines are obtained, T denotes the period of $x(t)$, m is an integer number, and $\langle \cdot \rangle$ denotes a time-averaging operation.

The most common form of non-linear transformation is the quadratic time-invariant transformation which induces second order cyclostationarity. This cyclostationarity gives rise to distinct correlation patterns in the spectrum of the signal [99]. In the previous example, the quadratic transformation of $x(t)$ is given by $y(t) = x(t)x(t - \tau)$, where τ denotes a non-zero delay. The cyclostationary properties of $x(t)$ are then evaluated using the Cyclic Autocorrelation Function (CAF), also referred to as the

¹For consistency with the work presented in related articles [97] [98], as well as that of other authors in the field of CR, the same notation has been adopted in this chapter. Consequently, certain symbols may appear to be the same as those employed in other chapters of this thesis. Nonetheless, their meaning is entirely different and should not be confused.

Fourier coefficient, denoted by $R_x^\alpha(\tau)$ and given by

$$R_x^\alpha(\tau) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x(t)x^*(t - \tau)e^{-j2\pi\alpha t} dt. \quad (3.1)$$

A signal exhibits second order cyclostationarity if its CAF is non-zero for a non-zero cyclic frequency [104]. Alternatively, the cyclostationarity may be characterised through the Spectral Correlation Density (SCD) function, which is obtained by taking the Fourier transform of the CAF. The most common applications exploiting cyclostationarity include:

- **Signal processing:** For generating better estimates of the original transmitted signal corrupted during its propagation over a hostile channel.
- **Radio-signal analysis:** For the detection and classification of signals, as well as for modulation recognition.
- **Communication systems:** For the design and analysis of systems where nonlinearities are present.

This discussion focuses on the use of cyclostationarity for signal processing tasks. The SCD function is employed to detect the presence of signals and classify them according to their modulation type. This is achieved via the generation of spectral lines at different cyclic frequencies followed by the detection of the presence of that spectral line. The SCD function may also be used for estimating parameters like carrier frequencies, phases and keying rate.

3.2.2 Cyclostationary Signatures

A *cyclostationary signature* is a feature deliberately embedded in the physical properties of a digital communications signal which can be exploited thanks to the underlying periodicity of that signal [99]. OFDM is the preferred modulation scheme due to the inherent cyclostationarity present in an OFDM signal employing the CP.

In CR systems, cyclostationary signatures are used for system identification and network coordination by embedding the signature in OFDM waveforms [99] [104]. This is achieved by configuring the signals at the transmitter in such a way, that the signal statistics between primary and secondary users are significantly different. By altering the length of the GI, distinct peaks can be induced at predefined cyclic frequencies where the signal exhibits the highest power levels. The amplitude of these peaks depends on the statistical nature of the signals dictated by the modulation format, the bit rate and the carrier frequency under consideration [99].

However, changing the GI length translates to a variable OFDM frame. For this reason, methods for inducing cyclostationary features in an OFDM waveform without altering the frame length have been proposed [104]. The first method, allocates a set of sub-carriers to be used as a preamble at the beginning of each OFDM frame. The second method, dedicates a set of sub-carriers for the entire duration of each OFDM frame. Both methods configure the time-domain or frequency-domain characteristics of the allocated sub-carriers in a specific way to induce unique CAF patterns allowing the receiver to distinguish between different systems.

An alternative technique for creating artificial cyclostationary signatures was proposed by Sutton using sub-carrier mapping [99]. In particular, one set of sub-carriers is carefully mapped to another set of sub-carriers in a predefined manner, resulting in data symbols being transmitted simultaneously on more than one sub-carriers. “A statistical dependence between certain sub-carriers of an OFDM signal” [99] is therefore created, generating distinct autocorrelation patterns while introducing a form of redundancy.

Contrary to conventional cyclostationary signal analysis, which requires long observation times to guarantee the reliability of a SCD function estimate, cyclostationary signatures can achieve the same performance with lower complexity. Since these signatures are unique, they can be used for network coordination in CR systems alleviating the need for a dedicated control channel. Taking into account that these signatures

are constantly present in a transmitted OFDM symbol, detection can be carried out using any segment of the received signal. Finally, they can be integrated in existing transceiver architectures without modifying the standards. The main drawback is the overhead incurred to map the sub-carriers appropriately, sacrificing transmission rate for detection performance [99].

The generation of cyclostationary signatures by exploiting the cyclic features present in CDD has also been examined [107]. In this case, the signatures are created by manipulating the cyclic delay, the latter introduced in Section 2.1.2. The authors demonstrate that altering the cyclic delay in OFDM based transmitters employing CDD, gives rise to CDD-induced signatures [107].

As a final note, the cyclostationary features present in an OFDM signal may also be used to improve automatic modulation classification, the latter defined as “the automatic recognition of the modulation format of a sensed signal” [108]. Ordinarily, automatic modulation classification takes place between the detection of a signal arriving at the receiver and its subsequent demodulation. The first objective is to identify the existence of a signal operating at a certain frequency or being present at a given location. The second objective is to determine the modulation type employed by the detected signal. These objectives are accomplished using the Cyclic Domain Profile (CDP) which corresponds to the maximum value of the normalised SCD function over all cyclic frequencies [108]. Having generated the CDP, the desired signal is extracted using pattern matching or Hidden Markov Models (HMMs).

In conclusion, it is evident that cyclostationary feature detection constitutes a low complexity, nevertheless, effective technique to address the detection, classification and coordination of signals and systems in CR networks.

3.3 Cognitive Radio Modulation Techniques

The technology which allows different wireless systems to coexist in the same spectrum space is known as *spectrum pooling* [109]. The idea is to merge continuous and dis-

continuous frequency bands into a common pool. Ordinarily, spectrum pooling allows the secondary utilisation of a licensed system in order to increase spectral efficiency. Spectrum pooling allows more users to access the same channels without calling for additional hardware nor changes to the existing incumbent system. The spectral efficiency gains and the maximum number of users allowed to transmit in various bands, when employing spectrum pooling, has been quantified by Haddad et al. [110].

Two methods employed to access the spectrum are *spectrum overlay* and *spectrum underlay* [94]. In spectrum overlay, systems exploit frequency bands which have not been used by primary users. In spectrum underlay, systems may transmit in channels occupied by licensed users, provided that the transmit power of the secondary users is low enough to be perceived as noise by the primary users. Spectrum underlay requires the use of spread spectrum technologies, thereby spectrum overlay is usually preferred, especially when cross-talk amongst users is high.

Evidently, spectrum pooling has to deal with a number of technical issues at the physical and upper layers. For this reason, novel CR modulation techniques are coming to light aimed at improved spectrum management and interference mitigation.

CR systems must be able to aggregate spectrum white spaces over discontinuous frequency bands without impeding the transmission of existing primary systems [111]. OFDM is therefore an attractive option for CR as it allows a set of sub-carriers to remain unused, essentially acting as an adaptive transmit filter [109]. One such example is Non-Contiguous OFDM (NC-OFDM) [112] which can achieve high data rates by pooling a number of (potentially fragmented) channels and reduce cross-talk by deactivating specific blocks of sub-carriers [38]. The key component of NC-OFDM is an FFT pruning algorithm which reduces the number of arithmetic operations required, hence offering the additional benefit of faster execution compared to a conventional FFT operation.

The high sidelobes associated with OFDM may give rise to spectral leakage causing secondary users to interfere with the communication of primary users. Consequently, alternative multi-carrier communication techniques have been proposed to suppress

the sidelobes. The most popular sidelobe suppression methods involve some form of signal shaping, for example, windowing, sidelobe cancellation via carrier insertion, sub-carrier weighting, multiple choice sequence, as well as adaptive symbol transition [113]. Through sidelobe suppression and pulse shaping, primary and secondary users may coexist in the same spectrum space without interfering with each other.

A system that has come to light recently, offering sidelobe mitigation, pulse shaping and interference cancellation, is GFDM [111]. GFDM reduces out-of-band radiation, thanks to RC pulse shaping, and improves signal detection with the aid of tail biting techniques. Tail biting allows a circular structure to be maintained in each transmitted block, thus inducing cyclostationarity. OFDM is in fact a special case of GFDM with the number of blocks set equal to one [103]. It has been demonstrated that GFDM is well-suited for allowing a secondary system employing this multi-carrier scheme to be overlaid on top of an OFDM primary system [114].

GFDM may be classified as a filterbank multi-carrier technique, the latter regarded as a promising method for allowing *agile waveforming* over fragmented spectrum [38]. Filterbank techniques have lower complexity compared to multi-taper methods, which are considered to be the optimum solution for spectrum sensing [95]. Filterbanks offer full control over spectral leakage allowing dynamic adjustment of the interference temperature, which is a desirable feature enforced by regulators [115].

Another drawback associated with OFDM is the use of the FFT, which limits the spectral dynamic range and may lead to CRs being unable to detect low-power users. For this reason, a more disruptive solution which suggests replacing the Fourier transform with the wavelet packet based transform, has been proposed [113].

In conclusion, spectrum pooling constitutes an innovative technology allowing diverse networks to coexist in the same frequency regions by overlaying one system on top of another. Despite the increasing number of novel modulation techniques entering the scene, it is still difficult to strike a balance between performance and complexity.

3.3.1 Cognitive Radio Testbeds

Although the literature is rich in CR concepts and techniques for DSA, there has been a limited number of prototypes. The known platforms that have been developed over the last decade have been surveyed by Pawelczak et al. [116]. The majority of these testbeds comprise both software and hardware modules employing reconfigurable devices, such as FPGAs. Three key successful prototypes include GNU Radio [117], which emerged from Massachusetts Institute of Technology (MIT), the IRIS (Implementing Radio In Software) system [118], which was developed by Trinity College, University of Dublin, and the WARP (Wireless open-Access Research Platform) [119] from Rice University. These systems are designed to carry out most signal processing operations in software, nevertheless, low-level and high-level tools, such as VHDL and MATLAB, may also be used. An open design front-end is usually employed, the most popular one being the Universal Software Radio Peripheral (USRP) provided by Ettus Research [120].

While the IRIS framework has been a popular system for testing different CR applications, trials and prototypes for experimenting with transmission and spectrum sensing in TVWS have also emerged. Key demonstrations include:

- Cyclostationary signature embedding and detection (Trinity College, University of Dublin) [121].
- An FPGA based autonomous adaptive radio (Trinity College, University of Dublin) [122].
- OFDM pulse-shaping for DSA (Trinity College, University of Dublin) [123].
- Transmission over TVWS (Institute for Infocomm Research, Singapore) [124].
- NC-OFDM with spectrum sensing (TU Delft, University of Twente, Netherlands) [125].

Unsolved challenges encountered during the practical realisation of CRs are the lack of receiver selectivity and the need for frequency-agile front-ends. The aforementioned

testbeds and demonstrations focus on DSA functionality and have been designed to work in licence-exempt bands. Hence, an important gap that needs to be addressed in the short-term future is the impact of secondary user access on the communication of primary users operating in a real, licensed system. Another interesting topic which would benefit from further research is the application of Artificial Intelligence (AI) techniques to the spectrum selection process [116]. In line with these observations, Section 3.4 proposes the incorporation of traffic prediction methods in the CR cycle, to improve spectrum access and thus QoS for secondary users.

3.4 Improving Cognitive Radio Spectrum Access using Traffic Prediction

The ability to capture a frequency slot for transmission in a vacant channel has a significant impact on the spectrum efficiency and QoS of a CR user. The radio front-ends of a CR user have limited bandwidth for spectrum sensing with the target frequency bands dispersed in a discontinuous manner. This forces the CR user to sense multiple target frequency bands in a small period of time prior to selecting an appropriate transmission opportunity. Consequently, the transmission path and QoS of the CR user could be severely degraded.

The ability to predict the variation of transceiver parameters reflecting network conditions is a key challenge in network management and control. The idea of predictive dynamic spectrum access is not new [126] [127]. Notwithstanding, existing techniques do not consider traffic prediction and spectrum sensing in conjunction.

To address this challenge, the work presented in this section brings together theory from the CR and machine learning domains with a two-fold aim; first, to improve the spectrum sensing process carried out by a CR user to identify vacant channels assigned to licensed users at a specific instantaneous point in time; second, to improve the QoS of a CR user while mitigating the interference induced to licensed users as a result of

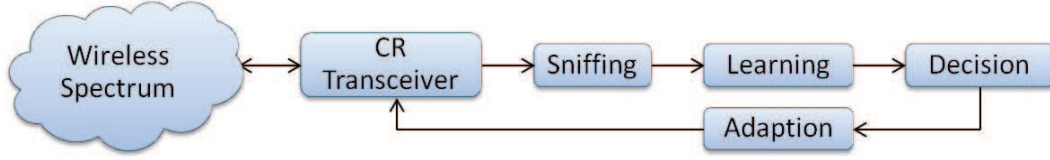


Figure 3.1: Selective spectrum sensing and access cycle.

this channel sharing.

To this end, a SOSA framework is proposed which applies traffic prediction techniques to conventional CR systems. Details of this framework have been reported in related work [97] [98]. In contrast to alternative approaches, the SOSA framework takes into account practical issues, such as discontinuous target frequency bands, as well as the limited spectrum sensing ability of CR users. The performance of the proposed solution is evaluated in terms of Packet Loss Ratio (PLR) and throughput.

3.4.1 Selective Opportunistic Spectrum Access Framework

CR users transmit information over temporally unused frequency bands. Two conflicting objectives arise from this configuration; the CR system must refrain from interfering with the transmission of primary users while simultaneously optimising the performance of secondary users. The proposed solution improves the ability of a CR user to transmit in a vacant channel during a specified time slot by selecting the target frequency band based on an optimum channel sensing order. This channel sensing order is dictated by the ability of the system to predict the traffic load and identify the channels that have the highest probability of appearing idle in the next timeslot. This probability coupled with the time required to sense the frequency bands under consideration have a direct impact on the performance of the system.

Spectrum Sensing and Access Strategy

The spectrum sensing cycle consists of four phases, namely *sniffing*, *learning*, *decision* and *adaption*. This process is shown diagrammatically in Fig. 3.1.

It is assumed that the CR is operating in a distributed and slotted wireless access environment using spectrum overlay. The purpose of each phase may be outlined as follows:

1. **Sniffing:** During this phase, information regarding the usage of the target frequency bands is collected and stored.
2. **Learning:** This stage is responsible for estimating the primary user's traffic pattern in order to predict the future traffic variation.
3. **Decision:** At this point, the system determines the optimum channel sensing order based on i) the secondary user's QoS requirements, ii) each channel's probability of appearing idle, and iii) each channel's transmission capability.
4. **Adaption:** The CR transceiver's parameters are adapted according to the outputs acquired from the preceding phases taking into account the instantaneous channel variations.

The spectrum sensing process is responsible for two crucial functions; first, it validates the state of the primary channels to be accessed to avoid potential interference to the primary users; second, it collects long-term usage data to improve the accuracy of the prediction mechanism.

Traffic Model and Nomenclature

The primary user's traffic pattern is modelled as an alternating renewal process consisting of busy (*on*) and idle (*off*) periods [94]. All primary user channels serve as target-sensing channels for secondary users. This is equivalent to a two-state birth-death process with death rate α and birth rate β . The following list outlines the notation² used in subsequent paragraphs:

²A reminder that the symbols used in this chapter refer to different parameters than those indicated in other chapters of this thesis, even though they adopt the same notation.

N : The total number of target frequency bands or primary channels that can be accessed by the secondary user.

K : Random variable representing the number of channels that are idle in a time slot, where $0 \leq K \leq N$.

M : Random variable representing the number of channels which are assumed to be busy in a time slot when in fact they are idle, where $0 \leq M \leq K$.

L : Random variable representing the number of channels which are assumed to be idle in a time slot when in fact they are busy, where $0 \leq L \leq N - K$.

i : The time slot index in the primary system.

T_s : The time slot length in the primary system.

t_s : The average time required to sense a primary channel.

t_h : The average time required for a secondary user to switch between two channels (handoff).

R : The target transmission rate of a secondary user.

C_n^i : The transmission capacity of the target secondary user on the n^{th} primary channel in the i^{th} time slot.

S : The spectrum sensing time threshold, defined as the maximum number of channels the secondary user is allowed to sense to find a vacant channel.

α_n : The mean value of the *off* period (idle state) for the n^{th} primary channel.

β_n : The mean value of the *on* period (busy state) for the n^{th} primary channel.

P_e : The average probability of error associated with the secondary user's prediction mechanism.

$P_{off}^{n,i}$: The predicted probability of the n^{th} primary channel appearing idle in the i^{th} time slot.

$P_{on}^{n,i}$: The predicted probability of the n^{th} primary channel appearing busy in the i^{th} time slot.

w_n^i : Channel state of the n^{th} channel at time slot i . $w_n(i) = 1$ means the channel is occupied by a primary user, in other words the channel is in a busy (*on*) state; $w_n(i) = 0$ means the channel is not being used by a primary user, in other words the channel is in an idle (*off*) state.

It is assumed that the overall time required to sense N primary channels does not exceed the length of a single time slot T_s , otherwise the secondary user will not be able to acquire the state of all channels.

Performance Metrics and Sensing Scenarios

The steps followed to carry out the SOSA cycle may be summarised as follows:

1. The secondary user retrieves the channel statistics during the sniffing stage.
2. Using the specified prediction model, the secondary user predicts the probability of each channel appearing ideal or busy in time slot $i + 1$.
3. The secondary user employs the predetermined ordering strategy to arrange the sensing sequence for time slot $i + 1$.
4. The secondary user starts to sense the primary channels according to the order identified during the previous step.
5. The secondary user locks on to the first vacant channel that becomes available.
6. The remaining primary channels are sensed until the sensing time reaches a pre-defined threshold.

7. Based on the channel statistics of the sensed channels, the secondary user may adjust the parameters of the prediction mechanism.
8. The process is repeated until all time slots have been exhausted.

The predicted probability of the n^{th} primary channel appearing idle in time slot $i + 1$ is denoted by p_n and is given by

$$p_n = \frac{\alpha_n}{\alpha_n + \beta_n}, \quad (3.2)$$

where $1 \leq n \leq N$ while α_n and β_n correspond to the mean values of the *off* and *on* periods for channel n , respectively. From Eq. 3.2, it can be deduced that the probability of the n^{th} appearing busy is $1 - p_n$. In this work, it is assumed that all primary channels have the same transmission capacity denoted by C_0 . It is further assumed that their channel-state variation is slow and remains constant over a long period of time, thereby avoiding the risk of distorting the results obtained from the prediction model.

Two key performance metrics are employed in the subsequent performance analysis, namely the PLR, which is dictated by the sensing time threshold, and the throughput, which is determined by the speed at which the secondary user can identify the first available idle channel. The number of occupied primary channels in a single time slot is given by $N - K$. Since all primary channels are independent and identically distributed (i.i.d.), it follows that each channel has the same probability of appearing idle. The secondary user will experience packet loss if no transmission opportunity is identified within the sensing time threshold S , expressed as

$$S = \left\lfloor \frac{T_s - R/C_0}{t_s + t_h} \right\rfloor, \quad (3.3)$$

where $1 \leq S < N$ and R is the constant rate at which the secondary user is required to transmit information. The throughput of a secondary user in time slot i is given by

$$I(i) = C_0 [T_s - \gamma_i^K (t_s + t_h)]. \quad (3.4)$$

where γ_i^K denotes the number of channels the secondary user has sensed prior to arriving at the first vacant channel. If a secondary user finds an idle channel in time slot i then the number of information bits in this time slot is $I(i)$.

These performance metrics are considered in three different spectrum sensing scenarios. Detailed derivations of the following formulae are provided in related work [98]. Here, they are presented and explained for the purpose of the performance evaluation discussed in Section 3.4.2.

- **Spectrum sensing without traffic prediction:** In this case, the secondary user senses the channels in a random or prefixed order. The probability that the secondary user cannot find an idle channel amongst K idle primary channels in time slot i is given by

$$P_{wo}^{i,K} = \prod_{j=1}^S \left(1 - \frac{K}{N - j + 1} \right), \quad (3.5)$$

where $0 \leq K \leq N$. Subsequently, the average PLR for this technique taking into account the sensing time threshold is given by

$$P_{wo}^{ave} = \prod_{n=1}^S (1 - p_n), \quad (3.6)$$

with $1 \leq S \leq N$. The average throughput may be expressed as

$$\bar{I}_{wo} = C_0 [T_s - \bar{\gamma}_{wo} (t_s + t_h)] - \prod_{i=1}^N (1 - p_i) \cdot C_0 [T_s - N (t_s + t_h)]. \quad (3.7)$$

where $\bar{\gamma}_{wo}$ represents the average sensing time and can be computed using the following formula

$$\bar{\gamma}_{wo} = \sum_{i=0}^N i \prod_{j=1}^{i-1} (1 - p_j) p_i. \quad (3.8)$$

- **Spectrum sensing with traffic prediction:** When traffic prediction is employed, the channels are ranked in order of decreasing probability of appearing

idle in the next time slot. Thanks to this optimum spectrum sensing order, the average PLR in this case is given by

$$P_w^{ave} = \prod_{j=1}^S (1 - p_n + P_e p_n), \quad (3.9)$$

while the average throughput is expressed as

$$\bar{I}_w = C_0 \left\{ T_s - \left[\sum_{k=0}^N \sum_{j=1}^{N-k+1} \binom{N}{k} P^k (1-P)^{N-k} j (1-P_e) \cdot P_e^{j-1} \right] \cdot (t_s + t_h) \right\}. \quad (3.10)$$

- **Spectrum sensing with precognition (upper bound):** In practice, perfect traffic prediction is not feasible. Nonetheless, this scenario is considered in this work in order to determine the system's performance upper bound. This translates to the secondary user having perfect knowledge of the exact state of each primary channel in future time slots with zero delay for spectrum sensing and handoff. Packet loss will occur if and only if there are no vacant channels in the time slot under consideration. Based on this description, the PLR in this scenario is given by

$$P_{pre} = \prod_{n=1}^N \frac{\alpha_n}{\alpha_n + \beta_n} = \prod_{n=1}^N (1 - p_n), \quad (3.11)$$

with the average throughput expressed as

$$I_{pre} = C_0 T_s \left[1 - \prod_{n=1}^N (1 - p_n) \right]. \quad (3.12)$$

3.4.2 Numerical Results

In this section, the performance of the proposed SOSA framework is evaluated in terms of PLR and throughput. The results are obtained over 10,000 time slots unless otherwise stated. It is assumed that all primary channels follow the same traffic model. It is further assumed that the lengths of the busy and idle periods follow an exponential

distribution. The birth and death rates for each primary channel are randomly generated while sufficient traffic statistics are recorded to estimate the average probability of each primary channel appearing idle. The magnitudes of these probabilities are then used to determine the optimum spectrum-sensing order.

Packet Loss Ratio Performance

In this section, the PLRs for the three different scenarios outlined in Section 3.4.1 are compared, namely the spectrum sensing cases with and without traffic prediction, as well as the system's performance theoretical upper bound. Packet loss occurs when the secondary user fails to locate a vacant channel during the sensing time threshold S .

Fig. 3.2 compares the different sensing schemes for a fixed number of primary channels ($N = 20$) with increasing threshold values. The curve with the lowest PLR (triangle markers) corresponds to the theoretical scenario (upper bound) in which the secondary user has precognition ability. Therefore, the secondary user is able to identify a vacant channel with the best transmission capability without the need for sensing, thus resulting in nil delay. This curve is used as a benchmark for evaluating alternative practical sensing methods. Fig. 3.2 illustrates that the method employing traffic prediction offers significant performance gains in contrast to the scheme without traffic prediction. PLR performance is improved with increasing threshold values attributed to the fact that the probability of an idle channel becoming available increases as S is increased. Conversely, the theoretical upper bound corresponding to the best-case scenario remains constant at all times since it is independent of S . Notwithstanding, an increase in the value of S translates to a reduction in the secondary user's effective transmission rate.

A relatively simple and low complexity prediction and channel-ordering technique was employed for the sensing method with traffic prediction. Consequently, it is expected that greater performance gains may be attained by substituting this technique with more sophisticated prediction methods. Fig. 3.2 shows that the results acquired

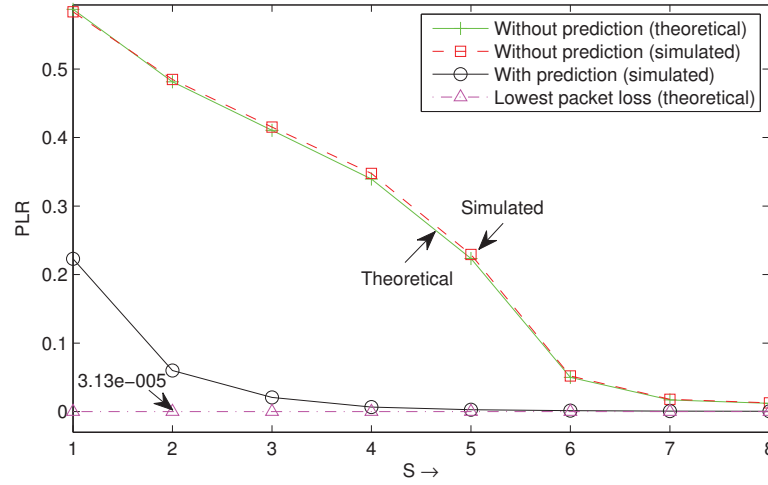


Figure 3.2: PLR performance for increasing threshold values ($N = 20$).

through simulation (square markers) match those obtained from theory (vertical line markers), thereby validating the proposed model.

Similar trends were observed when the number of primary channels was increased from 10 to 30 under a fixed threshold value ($S = 4$), as depicted in Fig. 3.3. In this situation, even though the number of primary channels available for opportunistic access is increased, the performance of the method without traffic prediction remains poor. This is caused by the lack of channel ordering which in turn is necessitated to allow the channels with the highest probability of appearing idle to be sensed first. Specifically, if the sensing threshold is fixed, the transmission opportunities offered by the first S primary channels in the case without traffic prediction will not change, thus the reason for the constant PLR shown in Fig. 3.3. As far as the theoretical upper bound (lowest PLR) and the sensing method with traffic prediction are concerned, increasing the number of primary channels translates in more transmission opportunities becoming available in any single time slot and therefore a higher probability of a primary channel being idle. Fig. 3.3 shows that simulation results are once again in accordance with theoretical values.

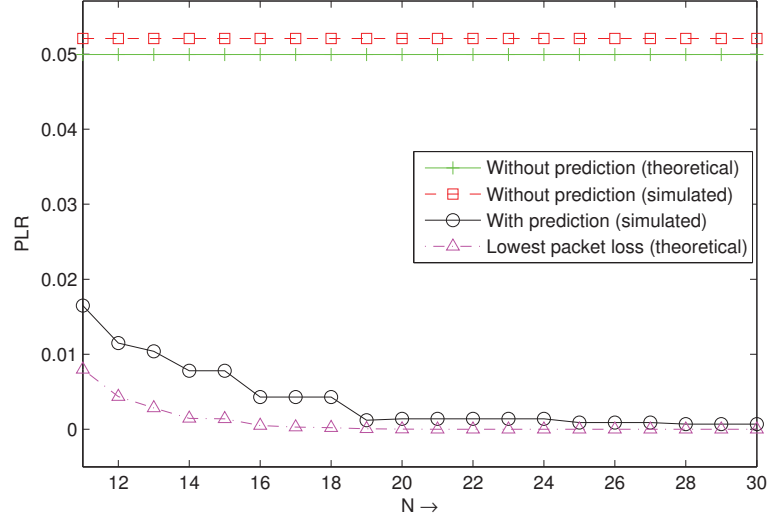


Figure 3.3: PLR performance for an increasing number of primary channels ($S = 4$).

Throughput Performance

This section evaluates the average throughput achieved in each time slot taking into account the number of primary channels available and the time consumed for spectrum sensing (t_s) and handoff (t_h). Fig. 3.4 shows that the maximum theoretical throughput can be achieved when the secondary user has precognition ability. The second best throughput performance is exhibited by the sensing method employing traffic prediction. Fig. 3.4 also demonstrates that beyond a certain number of channels, the gain in throughput is marginal.

Fig. 3.5 illustrates that an increase in the time assigned for spectrum sensing and handoff results in significant reduction in throughput. The gain of the sensing method with traffic prediction over the method without traffic prediction is more prominent for higher values of $(t_s + t_h)$. This is attributed to the fact that with traffic prediction a CR can identify and lock on to an idle channel much faster compared to a CR user who does not employ traffic prediction. Hence, in the former case, the effective transmission time is increased yielding a higher throughput, as depicted in Fig. 3.5.

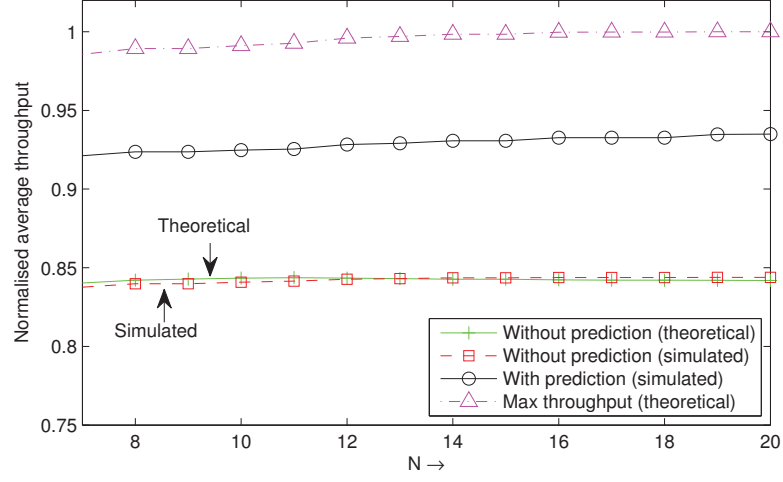


Figure 3.4: Throughput for an increasing number of primary channels ($\frac{t_s+t_h}{T_s} = 0.05$).

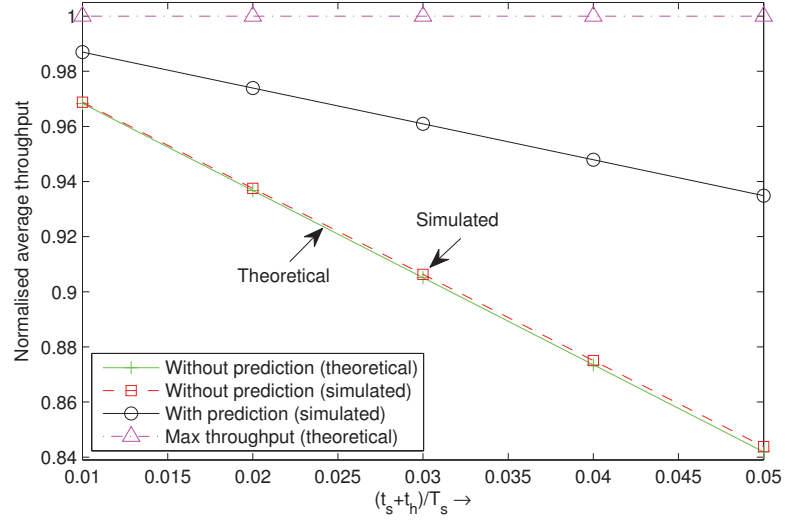


Figure 3.5: Throughput in each time slot for increasing sensing and handoff time ($N = 20$).

3.5 Conclusions

This chapter examined the concepts of CR and the novel techniques employed to address the coexistence of licensed and unlicensed users in the same spectrum space. An important aspect of CR devices is their ability to dynamically reconfigure their transceiver parameters to adapt to existing channel conditions. Interference avoidance and the detection of weak signals over large spectrum ranges, are the most important challenges in CR systems. Despite the wideband sensing capability associated with CR devices, spectrum sensing techniques and geolocation databases are required to guarantee communication reliability and protection of licensed users.

While energy detection is a simple spectrum sensing technique, its sensitivity to noise uncertainty calls for more sophisticated methods, such as cyclostationary feature analysis. Cyclostationarity is generally exploited to enhance the accuracy and reliability of information acquired from corrupted data sets. OFDM is deemed a suitable physical layer scheme for CR, thanks to the underlying periodicity inherent in an OFDM signal. By manipulating the cyclic features of such signals, unique signatures can be created allowing for accurate signal acquisition and system identification in CR networks. These artificial signatures are referred to as cyclostationary signatures and may be introduced by configuring the transmitter in a predefined manner, in terms of the signal's GI or sub-carrier mapping. In addition, a number of CR modulation techniques have come to light, aimed at mitigating interference through pulse shaping, filtering and/or windowing, sidelobe cancellation and sub-carrier deactivation. Notable examples include GFDM and filterbanks for multi-carrier communication.

The last part of this chapter explored a novel framework aimed at maximising the use of available channel resources in order to improve spectrum efficiency. The novelty of this work lies in complementing a conventional CR system with traffic prediction techniques. A selective opportunistic spectrum access framework was proposed to allow CR users to efficiently utilise the scarce radio spectrum which in turn consists of discontinuous target frequency bands. The sensing ability of CR users is improved

by determining the optimum channel sensing order with the aid of traffic prediction. The framework jointly considers the spectrum sensing efficiency and the transmission capacity of a secondary user to meet QoS guarantees. The performance of the sensing methods with and without traffic prediction was compared against the ideal case in which a secondary user has precognition ability. It was found that applying traffic prediction and channel ordering to the sensing mechanism yields significant performance gains in terms of throughput and PLR. The efficiency of spectrum utilisation is dictated by the accuracy of the traffic prediction incorporated in the CR system which in turn determines the appropriate reconfiguration of transceiver parameters.

Part II

Spectrally Efficient Multi-carrier Systems

Chapter 4

SEFDM: Principles and Challenges

The preceding chapters looked at different technologies aimed at improving spectral efficiency. The greater part of this thesis focuses on the practical design and implementation of a spectrally efficient multi-carrier transceiver as described in detail by Clegg et al. [128] based on the original idea by Rodrigues and Darwazeh [52]. New mathematical models leading to novel algorithms have been developed to generate and detect SEFDM signals with tolerable error penalties. Although previous work [129] [130] has given estimates of the complexity order associated with these algorithms, these estimates do not take into account all the design aspects and limitations encountered in a practical environment.

Hereby, the main purpose of this thesis is to address this gap through multiple objectives; first, to assess the impact of sampling and finite word length effects on the performance of SEFDM systems; second, to devise a reconfigurable real-time platform, entailing FPGAs and DSPs, for the implementation of SEFDM receivers; third, to reduce the complexity of existing algorithms rendering them suitable for hardware realisation; finally, to acquire realistic estimates of the *true silicon complexity* [61] associated with SEFDM.

To this end, this chapter commences with the theoretical principles underpinning SEFDM systems. The concept of time-frequency packing and the gain in spectral efficiency offered by SEFDM are described in Section 4.1. Section 4.2 reviews transmission techniques utilising the DFT, as well as demodulation methods employing orthonormalisation or matched filtering processes. Section 4.3 examines the nature of SEFDM systems, in terms of the self-created interference between the sub-carriers, which leads to the ill-conditioning of the sub-carriers correlation matrix. It is shown how this ill-conditioning can have a significant impact on the accuracy of the results acquired. Section 4.4 demonstrates through short mathematical proofs and simulation based studies, that the self-created interference present in SEFDM is due to spectral leakage and deliberate aliasing. This is an important observation taking into account that IDFT/DFT operations, which inherently have orthogonal properties, are employed for SEFDM signal synthesis and recovery. Finally, Section 4.5 compares and contrasts different DFT based SEFDM modems, according to their complexity and suitability for hardware realisation.

4.1 The Challenge - Spectrum Efficiency

As suggested by the title of this thesis, the research theme under consideration is spectrum optimisation. In 1948, Shannon determined the upper bounds on spectral efficiency for a given Signal-to-Noise Ratio (SNR) to yield an error-free transmission in his famous paper ‘A Mathematical Theory of Communication’ [131]. In line with this landmark work, the performance of all communication systems is evaluated in terms of achievable data rate and error performance, which in turn are proportional to the availability of two finite resources being power and transmission bandwidth. Consequently, systems are commonly classified as being bandwidth-limited, power-limited or at a point in between [132]. Boosting transmission power is not always an option, especially in light of continuous efforts by regulatory bodies pushing for ‘greener’, low power solutions. Increasing transmission bandwidth is no more feasible

given that spectrum is a scarce and an expensive resource.

Capacity, which is synonymous with spectral efficiency, is expressed as the ratio of the bit (information) rate to the available bandwidth and is thus measured in bits/second/Hertz (bits/s/Hz). The capacity formula is given by

$$C = B \log_2 \left(1 + \frac{S}{N} \right), \quad (4.1)$$

where B is the available transmission bandwidth, S is the average received signal power and N is the noise power.

In a practical system, the bit rate usually equals the sampling rate while the available bandwidth equals the transmitter's filter bandwidth. A number of issues may limit the effective bandwidth of a communication system. First, the filters used in the signal chain for shaping the pulse and/or for smoothing out quantisation effects are not ideal 'brick-wall' filters. Instead, they have a roll-off, which decreases spectrum efficiency. Second, the hostile wireless channel imposes strict limits on system capacity due to multipath propagation which leads to ISI and ICI.

The simplest way to reduce ISI is by reducing the symbol rate at the transmitter. For single carrier systems, this solution is in direct conflict with the requirements in modern communication systems which specify ever-increasing data rates and hence is unacceptable. Thanks to its orthogonality feature, OFDM can transmit the maximum possible number of independent data [129] and in this sense it yields the maximum feasible bandwidth efficiency with respect to a single carrier system.

The spectral efficiency of a conventional baseband digital scheme is given by

$$\eta = \frac{1}{BT_b} = \frac{R_b}{B} = \frac{R_{sym} \log_2 M}{B} \quad \text{bits/s/Hz}, \quad (4.2)$$

where T_b is the time duration of each bit, R_b is the bit rate, R_{sym} is the symbol rate and M is the size of the alphabet, known as the constellation cardinality. From Eq. 4.2, it should be evident that by increasing M or by decreasing the BT_{sym} product, the

resultant spectral efficiency can be increased. Assuming Nyquist filtering is applied and taking into account that the bandwidth of a system does not depend on the density of the constellation points [132], the symbol rate R_{sym} will equal the bandwidth B , therefore Eq. 4.2 becomes

$$\eta = \frac{R_b}{B} = \log_2 M \quad \text{bits/s/Hz.} \quad (4.3)$$

The bandwidth of an OFDM signal is given by

$$B_{OFDM} = \frac{(N-1) + 2}{T} = ((N-1) + 2)\Delta f \approx N\Delta f \quad \text{for large } N, \quad (4.4)$$

where N is the number of sub-carriers, B_{OFDM} is the bandwidth of an OFDM signal, T is the duration of one OFDM symbol and Δf is the spacing between the sub-carriers. Subsequently, the spectral efficiency of an OFDM signal is given by

$$\eta_{OFDM} = \frac{R_{sym} \log_2 M}{B_{OFDM}} \approx \frac{NR \log_2 M}{N\Delta f} = \log_2 M \quad \text{bits/s/Hz,} \quad (4.5)$$

where R is the data rate of each sub-carrier equal to Δf , which in turn equals the reciprocal of the OFDM symbol period T .

To eliminate completely the effects of ISI, a CP has to be added to an OFDM signal to protect the useful data payload. This CP, introduces an overhead and reduces bandwidth efficiency. In Chapter 3, CR was examined, which is aimed at improving spectral efficiency via better spectrum management. This is achieved by bridging re-configurable radio ideas, collectively known as SDR, with intelligent signal processing. Many institutions are approaching the challenge from a different angle by breaking the theoretical barriers of current technology with the objective of squeezing more data into the same bandwidth. This notion itself, however, presents a number of challenges, as discussed in the following sections.

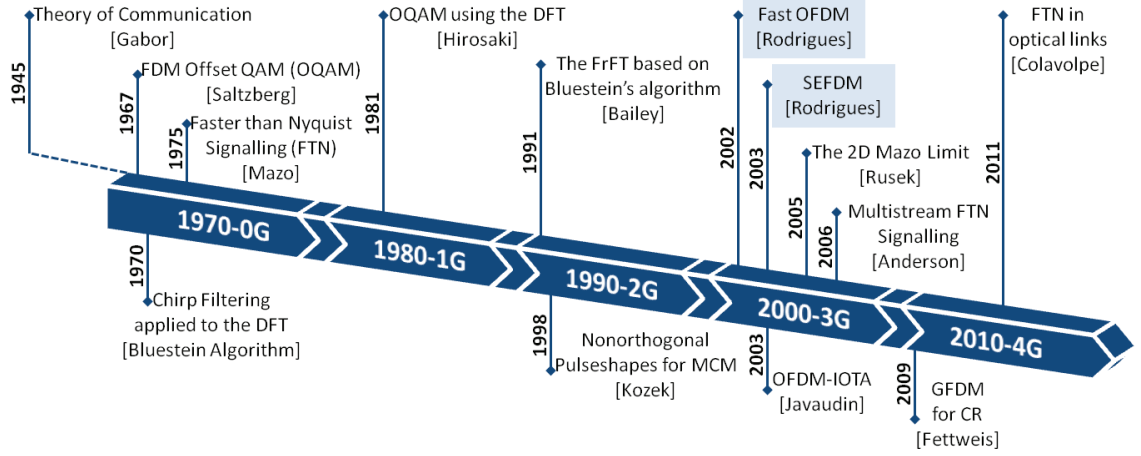


Figure 4.1: Timeline of spectrally efficient concepts and non-orthogonal communication techniques.

4.1.1 Time-Frequency Packing

Taking into account that the CP in OFDM systems reduces bandwidth efficiency by a factor of $\frac{N}{N+N_g}$, where N_g is the length of the CP in terms of discrete time samples, the first attempts to improve spectral efficiency focused on removing the CP and replacing rectangular pulses by pulses which were better localised in the time and frequency domains to avoid ISI. This alternative scheme termed OFDM/OQAM, which offsets the data by half a symbol, can guarantee orthogonality between sub-carriers only for real-valued symbols [45].

One of the first works examining the relationship between time, frequency, orthogonality and bandwidth efficiency was carried out by Gabor in 1945 [133]. This work along with the work by Mazo [40] triggered the pursuit of systems like Multi-stream FTN [50], GFDM [58] and SEFDM. While a detailed examination of all these systems is beyond the scope of this thesis, a relevant timeline of the fundamental concepts and techniques developed over the years is illustrated in Fig. 4.1.

The minimum required bandwidth for an ISI-free transmission is achieved through Nyquist filtering [132]. As the name suggests, FTN results in the transmission of more data within the minimum bandwidth specified by the Nyquist criterion. This causes

ISI which for a multi-carrier system also leads to ICI. Pulse shaping filters are therefore employed whose width, shape and roll-off determine how closely the sub-carriers can overlap with each other. Achievable information rates for FTN have been investigated recently [134]. While the focus in this thesis is on SEFDM, as proposed by Rodrigues and Darwazeh in 2003 [52], FTN, GFDM and SEFDM share common characteristics, such as:

- Aiming to increase spectral efficiency or favouring flexible bandwidth allocation.
- Using multi-carrier modulation similar to OFDM.
- Exploiting non-orthogonal signalling at the expense of self-created interference.
- Applying pulse shaping filters to mitigate ISI (FTN, GFDM).
- Employing iterative, tree-search and soft-decoding detection algorithms to recover the transmitted data successfully.

FOFDM [46] and MASK-OFDM [47] are special cases of spectrally efficient multi-carrier systems which discard the imaginary part of the signal allowing the spacing between the sub-carriers to be reduced by half relative to a conventional OFDM system. Nevertheless, such systems prevent the use of complex modulation schemes, such as M-QAM with $M > 2$. This motivated the invention of SEFDM [52], which offers an additional degree of freedom over OFDM by relaxing the stringent orthogonality constraint defined for OFDM. The beauty of SEFDM compared to other spectrally efficient techniques is the fact that it does not require special pulse shaping functions. Instead, it uses rectangular pulse shaping as in OFDM.

Recalling from Section 2.3.3, the orthogonality principle for OFDM is given by $\Delta f = \frac{1}{T} \Leftrightarrow \Delta f T = 1$. Earlier in this chapter, it was stated that spectral efficiency can be increased by decreasing the BT_{sym} product. In a similar fashion, SEFDM increases spectral efficiency with reference to OFDM by reducing the product $\Delta f T$. Specifically, $\Delta f T = \alpha$, where $\alpha \leq 1$ with $\alpha = 1$ corresponding to the OFDM case. The factor

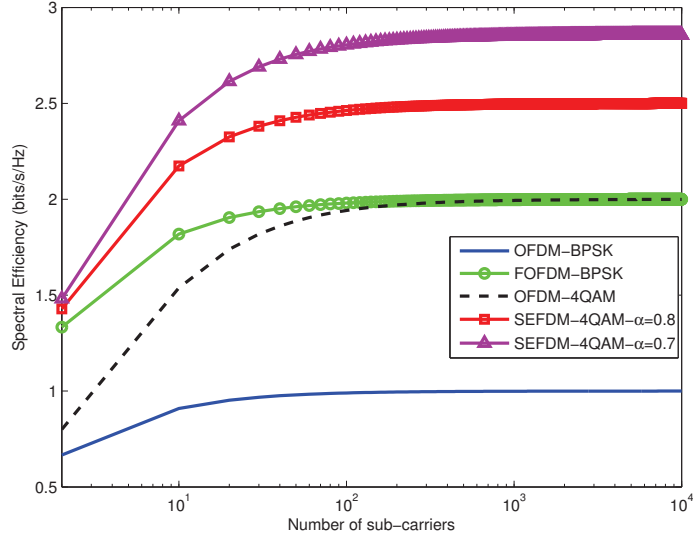


Figure 4.2: SEFDM vs. OFDM spectral efficiency.

α represents the bandwidth compression in SEFDM, thus the spectral efficiency of an SEFDM signal may be mathematically expressed as

$$\eta_{SEFDM} = \frac{R_{sym} \log_2 M}{B_{SEFDM}} \approx \frac{NR \log_2 M}{N \frac{\alpha}{T}} = \frac{\log_2 M}{\alpha} \quad \text{bits/s/Hz}, \quad (4.6)$$

whereby the gain with reference to an OFDM system is approximately $\frac{1}{\alpha}$ assuming that $B_{SEFDM} \approx N \Delta f \alpha$ based on the expression given for B_{OFDM} in 4.4.

Fig. 4.2 compares the spectral efficiency of SEFDM against OFDM for different bandwidth compression levels when BPSK and 4-QAM are employed. For this comparison, the exact formulae for the bandwidth efficiency have been used, in other words $\eta_{OFDM} = \frac{\log_2 MN}{2+(N-1)}$ and $\eta_{SEFDM} = \frac{\log_2 MN}{2+(N-1)\alpha}$. It is evident that for a large number of sub-carriers ($N > 100$), the spectral efficiency may be computed using the approximate formulae of Eq. 4.5 and 4.6, in accordance with the results presented in previous work by Kanaras [129].

Fig. 4.3 plots the theoretical spectral efficiency bound as defined by Shannon along with the potential gains that could be achieved using SEFDM. It is assumed that

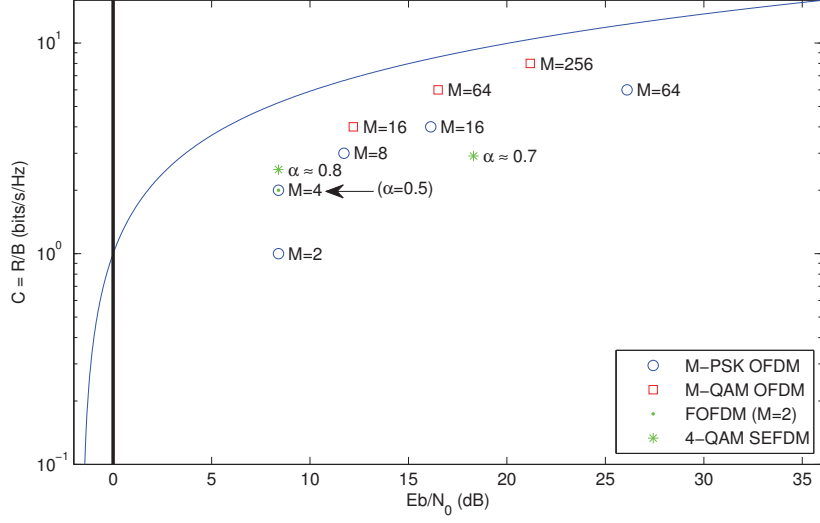


Figure 4.3: SEFDM on the bandwidth efficiency plane.

coherent modulation is employed, thus yielding a spectral efficiency of $\log_2 M$ bits for an OFDM system to achieve a BER of 10^{-4} . The upper bound denoted by the curve in Fig. 4.3 corresponds to the case where $R = C$ with

$$\frac{E_b}{N_0} \geq \frac{2^\eta - 1}{\eta}, \quad (4.7)$$

where E_b/N_0 denotes the energy per bit E_b to noise spectral density N_0 ratio.

Based on the definition of [132], SEFDM may be classified as a bandwidth-limited system, since it offers increased spectral efficiency at the expense of power. As illustrated in Fig. 4.3, SEFDM can achieve a higher spectral efficiency than OFDM without error penalties when $\alpha = 0.5$ (FOFDM) for BPSK and $\alpha \approx 0.8$ for 4-QAM modulation schemes. When $\alpha < 0.8$, a higher E_b/N_0 is required to achieve the same BER. These results are in accordance with the work presented by Kanaras [129], which demonstrated that for $\alpha < 0.8$, the self-created interference in SEFDM becomes more prominent. Similar work evaluating the maximum achievable spectral efficiency of FTN schemes in optical links has been carried out by Colavolpe [55].

The significance of the results presented in Fig. 4.3 is attributed to the fact that a practical detector was employed to achieve these spectral efficiency gains. In [129], it is assumed that an optimum detection is tangible for SEFDM systems to yield these performance gains. This statement is not entirely true since it is practically unfeasible to realise the optimum detector, being ML, due to its NP complexity¹. Even techniques which reduce the complexity of ML, such as SD and symbol-by-symbol detection, proposed in [129] and [55], respectively, are deemed inapt for hardware implementation.

The results presented in Fig. 4.3 were generated using a hybrid detector combining Iterative Detection with Soft Decision (IDSD) and FSD. Recent work [135] has demonstrated that this latter detector has tangible computational complexity favouring a practical detection². This is the reason the values of α shown in Fig. 4.3 are in fact equal to $\alpha = \frac{44}{64} \approx 0.7$ and $\alpha = \frac{51}{64} \approx 0.8$ due to practical design limitations, as will be discussed in Section 4.5.

Last but not least, from Fig. 4.3, it is evident that SEFDM can achieve nearly the same spectral efficiency as 8-PSK without error penalties at the expense of power. This observation may initially oppose the use of SEFDM for high bandwidth compression levels where $\alpha < 0.8$. Notwithstanding, the number of signalling points in the constellation dictates the number of quantisation levels required and thus the dynamic resolution of the D/A and A/D converters, which has to be increased in proportion to the constellation size to prevent performance degradation due to quantisation errors. Consequently, it may be the case that a 4-QAM SEFDM system with $\alpha \approx 0.7$ would be preferred over an 8-PSK OFDM system even at the expense of additional E_b/N_0 .

The spectral efficiency gains offered by SEFDM may be interpreted in different, nevertheless equivalent, ways:

- **Same information in less bandwidth:** This is achieved by reducing the frequency spacing and maintaining the same number of sub-carriers. In other words,

¹For example, for BPSK and 4-QAM modulation schemes with $N = 16$, the number of possible transmitted vector symbols amounts to 65,536 and 4,294,967,296, respectively.

²This is based on the continuation of this work by Tongyang Xu at UCL in late 2012, which is targeting the IDSD-FSD hybrid detector for hardware implementation.

$N_{SEFDM} = N_{OFDM}$ while $\Delta f_{SEFDM} < \Delta f_{OFDM}$ to yield $B_{SEFDM} < B_{OFDM}$.

- **More information in the same bandwidth:** This may be achieved using either of the following approaches
 1. Reducing the frequency spacing and transmitting data over a larger number of sub-carriers in the same bandwidth. This translates to $N_{SEFDM} > N_{OFDM}$ with $\Delta f_{SEFDM} < \Delta f_{OFDM}$ to yield $B_{SEFDM} = B_{OFDM}$.
 2. Maintaining the same frequency spacing and transmitting the same data over a smaller number of samples, thus increasing the effective data rate. This may be expressed as $\Delta f_{SEFDM} = \Delta f_{OFDM}$ and $N_{SEFDM} = N_{OFDM}$, however, $T_{SEFDM} < T_{OFDM}$ to yield $R_{SEFDM} > R_{OFDM}$. This latter approach could be regarded as being equivalent to FTN signalling.
- **A combination of reduced frequency spacing and increased data rate.**

The aforementioned methods may be subsumed under one term as *time-frequency packing* [136] and are illustrated in Fig. 4.4. Fig. 4.4 shows that $\frac{N}{\alpha} - N$ more sub-carriers can fit in the space 0 to $\frac{f_s}{2}$ compared to OFDM.

4.2 SEFDM System Description

SEFDM is based on similar key principles as OFDM in that a high data rate serial input stream is split into N parallel, overlapping sub-carriers. Consequently, the conventional transceiver for SEFDM, as originally proposed in [52], resembles that for OFDM, as illustrated in Fig. 4.5. It is assumed that rectangular pulse shaping is employed as in OFDM, hence this block has been omitted from Fig. 4.5.

As depicted in Fig. 4.5, N complex input symbols, denoted by $s_n = s_{n\Re} + js_{n\Im}$, modulate N sub-carriers generating a continuous time domain SEFDM signal expressed as

$$x(t) = \sum_{l=-\infty}^{+\infty} \sum_{n=0}^{N-1} s_{l,n} e^{j2\pi n \Delta f \alpha t}, \quad (4.8)$$

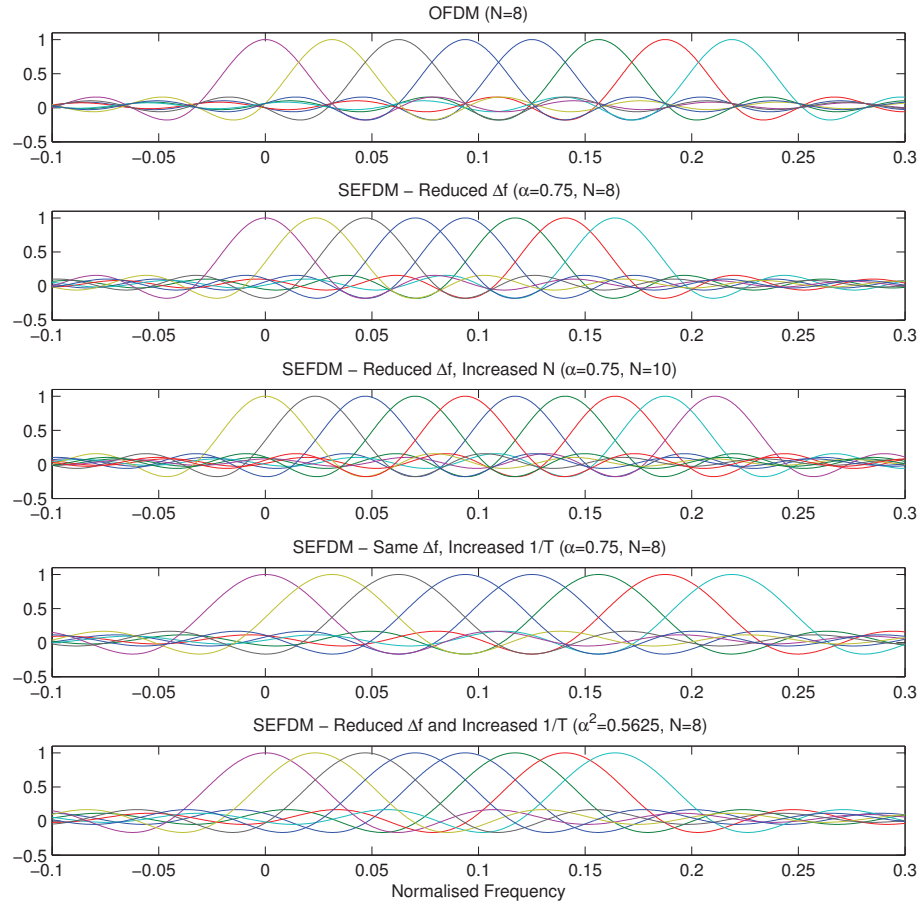


Figure 4.4: Methods for profiting from SEFDM spectral efficiency gains.

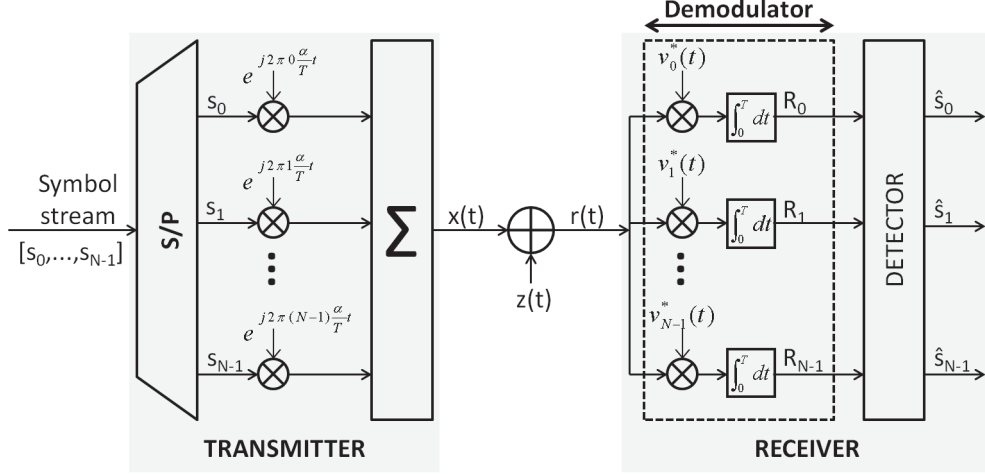


Figure 4.5: Conceptual architecture for an SEFDM transceiver using a bank of modems.

where \Re and \Im represent the real and imaginary parts of a complex number, respectively, $l \in \mathbb{Z}$ denotes the frame or SEFDM symbol index, $n \in \mathbb{Z}_{\geq 0}$ denotes the sub-carrier index and $s_{l,n}$ represents the information symbol conveyed by sub-carrier with index n during the time stamp with index l . Contrary to OFDM, SEFDM deliberately and counter-intuitively violates the orthogonality rule defined for OFDM by reducing the spacing between the sub-carriers. Hence, in SEFDM the sub-carrier spacing is equal to $\Delta f = \frac{\alpha}{T}$, where α dictates the level of bandwidth compression and T is the SEFDM symbol duration. As demonstrated in Section 4.1.1, SEFDM offers a spectral efficiency gain of approximately $\frac{1}{\alpha}$. Fig. 4.6 depicts the spectrum of an SEFDM signal for different values of α and in comparison to an OFDM signal. The spectrum was generated using the *pwelch* function provided by MATLAB for $N = 64$ with the last 12 sub-carriers deactivated, thus resembling the scenario for the IEEE 802.11a standard.

The SEFDM receiver is different compared to an OFDM receiver in that it incorporates two stages. The first stage involves a demodulation of the incoming received signal which generates a statistics vector \mathbf{R}^3 . This vector \mathbf{R} consists of N sufficient⁴

³Throughout the remainder of this thesis, where a symbol appears in bold, for example \mathbf{A} , this will denote a vector or matrix of any dimension.

⁴Sufficient statistics in relation to an unknown parameter, are statistics which cannot be compressed

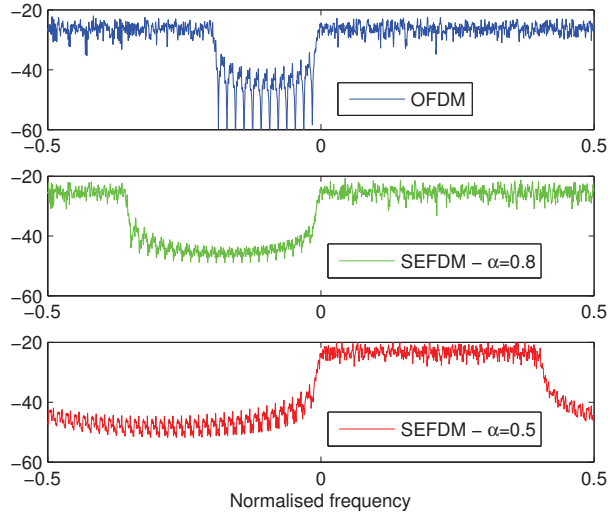


Figure 4.6: Spectrum of an SEFDM signal for different values of α .

statistics which are then fed to a detector to generate estimates of the transmitted data symbols. With reference to Fig. 4.5, the SEFDM signal $x(t)$ is transmitted through an AWGN channel and arrives at the receiver as

$$r(t) = x(t) + z(t), \quad (4.9)$$

where $z(t)$ represents the noise impairment introduced by the channel. The receiver model expressed mathematically by Eq. 4.9 is valid given two key assumptions; first, the demodulator is a linear filter to prevent noise colouring; second, the noise is Gaussian, in other words random and not burst, the latter being the case in a multipath propagation scenario due to fading and Doppler effects.

4.2.1 SEFDM Transmission using the DFT

This section considers the digital implementation of an SEFDM transmitter. The conventional method of generating SEFDM signals using a bank of analogue modulators does not scale well with the increase in the number of sub-carriers, taking into account any further without losing information about this parameter [137].

that the complexity of this method is of the order $\mathcal{O}(N^2)$. Hence, the potential of employing DFT operations to address this issue appears particularly attractive.

Since SEFDM is inherently a non-orthogonal system this means that standard DFT operations cannot be directly used. To this end, a first example of an algorithm employing the DFT in a modified configuration to generate non-orthogonal signals is the Inverse Fractional Fourier Transform (IFrFT) as proposed by Bailey [138], which considers fractional roots of unity. In analogy to an IDFT operation, the output of the IFrFT may be expressed as

$$x_{Fr}(m) = \frac{1}{N} \sum_{n=0}^{N-1} s_n e^{j2\pi m \alpha n / N}, \quad (4.10)$$

which converges to the IDFT formula for $\alpha = 1$. As noted in [138], the IFrFT can be computed in a fast and efficient manner using the algorithm proposed by Bluestein [139]. Using this latter algorithm the computation of the IFrFT requires $20N \log_2 N$ floating-point operations. While this results in the algorithm being 20 times more computationally intensive compared to the conventional IDFT algorithm, this technique still yields a significantly lower complexity compared to the analogue counterpart.

Notwithstanding, the ultimate goal is to generate non-orthogonal signals using standard DFT operations. The first attempts date back to the 1980's when Hirosaki used the DFT for an OQAM OFDM system [43]. A similar approach was adopted later on in FOFDM systems where the value of α is fixed at 0.5. More recently, however, standard DFT operations have been employed in non-orthogonal systems where the value of α is arbitrary. A first example appears in the HC-MCM system proposed by Hamamura [48]. Similar techniques were developed later on by Isam and Darwazeh [140] for use in SEFDM systems. These latter methods allow SEFDM signals to be generated using IDFT operations, thus offering the same advantages in terms of complexity as OFDM.

DFT based Implementations

The equation for the continuous time SEFDM signal was given in Eq. 4.8. Let T_s be the sampling period introduced to discretise the time and let ρ be an oversampling factor such that $Q = \rho N$ with $\rho \geq 1$. Then the discrete time axis will be given by mT_s while the symbol period will be given by $T = QT_s$. Consequently, the discrete time expression for a single SEFDM symbol may be expressed as

$$x(m) = \frac{1}{\sqrt{Q}} \sum_{n=0}^{N-1} s_n e^{j2\pi m \alpha n / Q}, \quad (4.11)$$

where $x(m)$ represents the time samples indexed by the integer m , Q represents the total number of time samples contained in a single SEFDM symbol, $\mathbf{s} = [s_0, \dots, s_{N-1}]'$ is a vector of complex data symbols modulated using PSK or QAM modulation and $[\cdot]'$ denotes a vector or matrix transpose operation. Specifically, \mathbf{s} represents the input data vector with $\mathbf{s} \in \mathfrak{M}^N$ where⁵ \mathfrak{M} represents a discrete alphabet, in other words the set of constellation points. The size of the alphabet, known as the constellation cardinality, is denoted by M . Where QAM is employed, it is assumed that the constellation is rectangular with M equal to even powers of two and that the data symbols are encoded using Gray encoding.

Three different, nevertheless equivalent, SEFDM transmitter types were proposed by Isam, which are illustrated in Fig. 4.7 with details provided in [140]. In these architectures, the bandwidth compression α is expressed as a ratio of integers, in other words $\alpha = \frac{b}{c}$ where $b, c \in \mathbb{Z}_{>0}$ with $b < c$. To simplify the forthcoming discussion and without loss of generality, let it be assumed that no oversampling is used ($\rho = 1$) yielding $Q = N$. The transmitters shown in Fig. 4.7 are then designed to offer diverse trade-offs. Type 1, known as the *proportional inputs* transmitter, is the simplest method in that it employs a standard IDFT block of length $\frac{N}{\alpha}$. This means that the size of this block compared to a conventional OFDM system is $\frac{1}{\alpha}$ times larger. The number

⁵If \mathcal{X} is a member of \mathcal{Y} then this is denoted by $\mathcal{X} \in \mathcal{Y}$.

of input data symbols remains the same, in other words N with the remaining $\frac{N}{\alpha} - N$ IDFT inputs padded with zeros. The output of the IDFT is truncated, hence only N samples are taken forward to construct the SEFDM signal while the remaining outputs are ignored. As will be shown in Section 4.4, this results in the SEFDM signal transmitting the same amount of information over the same number of sub-carriers as OFDM but with the benefit of reduced bandwidth utilisation. The key drawback of the Type 1 transmitter is that it requires the ratio $\frac{N}{\alpha}$ to be an integer number to prevent performance degradation. Consequently, this limits the flexibility in choosing the value of α .

For this reason, two alternative transmitter schemes were also proposed, namely Type 2 and Type 3, shown in Fig. 4.7. Type 2, known as the *rational α* transmitter, uses a cN length IDFT block. Only the inputs of the IDFT whose indices are integer multiples of b , in other words for $n = 0, b, 2b, \dots, (N-1)b$ are linked to the input data symbols while the remaining IDFT inputs are padded with zeros. This is achieved via the following condition

$$s_n = \begin{cases} s_{\frac{n}{b}} & \text{if } n \bmod b = 0 \\ 0 & \text{otherwise} \end{cases} \quad (4.12)$$

where *mod* is defined as the modulus of the remainder after division.

Complying with the condition of Eq. 4.12, the generation process for the Type 2 transmitter may be expressed as

$$x(m) = \frac{1}{\sqrt{N}} \sum_{n=0}^{cN-1} s_n e^{j2\pi \frac{mn}{cN}}. \quad (4.13)$$

The benefit of this transmitter technique is that b and c can be chosen independently to give any desired value of α for any number of sub-carriers N . The length of the IDFT block in this case is c times larger compared to an equivalent OFDM block and b times larger compared to the Type 1 SEFDM transmitter.

Thereby, the Type 3 transmitter, known as the *sum of multiple IDFTs*, was pro-

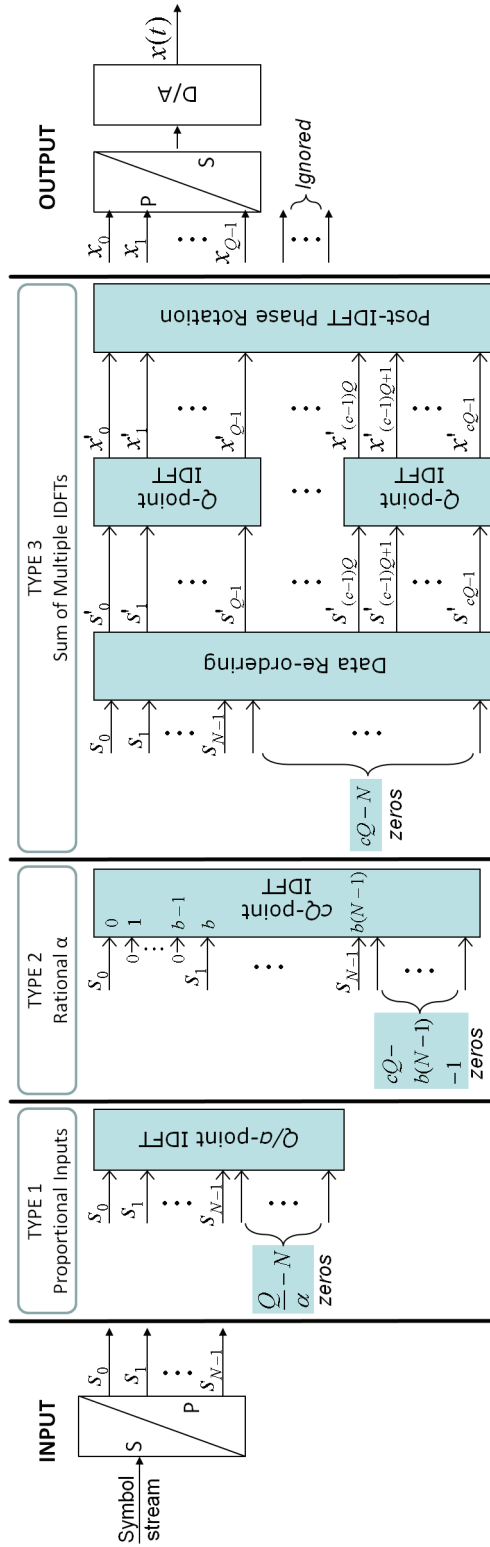


Figure 4.7: DFT based SEFDM transmitter types.

posed. This transmitter uses c separate IDFT blocks each of length N instead of one large cN IDFT block. The advantage of this transmitter over the Type 2 architecture is two-fold; first, it allows multiple IDFT blocks to be configured in parallel, a feature attractive for hardware implementation employing FPGAs; second, it reduces the computation time required for the IDFT processing stage provided that all c IDFT blocks can operate concurrently in parallel. It is important to note that the Type 3 transmitter requires two additional operations, namely a reordering stage before the IDFT block, as well as a post-IDFT phase rotation stage. The expression for the Type 3 transmitter is therefore given by

$$x(m) = \frac{1}{\sqrt{N}} \sum_{u=0}^{c-1} e^{j2\pi \frac{um}{cN}} \sum_{n=0}^{N-1} s_{u+nc} e^{j2\pi \frac{mn}{N}}. \quad (4.14)$$

Due to these additional processing stages required for the Type 3 transmitter, its reduced complexity and preference over a Type 1 transmitter becomes questionable, as will be revealed in Section 4.4, which discusses the complexity of the aforementioned transmitters. For the remainder of this thesis, all references to an SEFDM transmitter will assume that a Type 1 transmitter is employed with the values of N , b and c selected carefully to ensure that the ratio $\frac{N}{\alpha}$ is always an integer number.

4.2.2 SEFDM Demodulation

As illustrated in Fig. 4.5, the SEFDM receiver consists of a demodulation and a detection stage. This section looks at the demodulation stage. The first part of the demodulator involves the correlation of the received signal with a conjugate set of correlation functions $v_n(t)$. These functions must be orthonormal or orthogonal [129] to satisfy the linear filtering requirement pertaining to Eq. 4.9. The basis functions are orthonormal if i) they are mutually perpendicular, in other words $\langle v_i, v_j \rangle = 0$, when $i \neq j$ where $\langle \cdot, \cdot \rangle$ denotes an inner product, and ii) they have unit length, in other words $\langle v_i, v_i \rangle = 1$ [141]. If the basis functions do not have unit length then the set is orthogonal.

Two methods have been proposed to generate the basis functions $v_n(t)$. The first method uses *orthonormalisation procedures*, such as Gram-Schmidt (GS) or Löwdin processes, to generate an orthonormal base from a set of linearly independent but non-orthogonal sub-carriers [129] [142]. Denoting $b_n(t)$ the set of GS orthonormal basis functions, the output of the receiver's demodulator is given by

$$R_n = \int_0^T r(t)v_n^*(t)dt = \int_0^T r(t)b_n^*(t)dt, \quad n = 0, \dots, N-1. \quad (4.15)$$

Subsequently, the linear statistical model of Eq. 4.9 can be represented in matrix format as

$$\mathbf{R} = \mathbf{D}\mathbf{s} + \mathbf{Z}, \quad (4.16)$$

where $\mathbf{R} = [R_n]$ is the $N \times 1$ vector of observation statistics, $\mathbf{s} = [s_n]$ is the $N \times 1$ vector of complex transmitted data symbols and $\mathbf{D} = [D_{ij}]$ is the $N \times N$ covariance matrix of the SEFDM sub-carriers and the orthonormal base termed the projections matrix. The vector $\mathbf{Z} = [Z_n]$ represents N noise samples, which are complex i.i.d. Gaussian samples with mean $\mu = 0$ and variance $\sigma^2 = \frac{\mathcal{N}_0}{2}$, where \mathcal{N}_0 is the noise spectral density.

In [129], it is shown that the projections matrix \mathbf{D} is upper triangular since it is generated through a GS orthonormalisation process, which in turn employs a \mathbf{QR} decomposition method, where the \mathbf{Q} component is a unitary⁶ matrix while the \mathbf{R} component is an upper triangular matrix. If Φ is the matrix representation of the SEFDM sub-carriers matrix and \mathbf{B} is the matrix representation of the orthonormal basis functions, then the projections matrix \mathbf{D} is given by

$$\mathbf{D} = \mathbf{B}^H \Phi, \quad (4.17)$$

where \mathbf{B}^H denotes the Hermitian⁷ of matrix \mathbf{B} .

⁶A square matrix is unitary if its conjugate transpose is equal to its inverse, in other words $\mathbf{A}^H = \mathbf{A}^{-1}$ and hence $\mathbf{A}^H \mathbf{A} = \mathbf{I}$ is always valid [142].

⁷A Hermitian matrix is a square matrix whose complex entries are equal to their conjugate transpose denoted by \mathbf{A}^H [142].

The second method for generating the basis functions $v_n(t)$ uses *matched filtering*, as defined in [130]. In this case, the conjugate basis functions $v_n^*(t)$ equal the conjugate sub-carriers Φ^H , in other words

$$R_n = \int_0^T r(t)v_n^*(t)dt = \int_0^T r(t)e^{-j2\pi n\alpha\Delta f t}dt, \quad n = 0, \dots, N-1. \quad (4.18)$$

Hence, Eq. 4.16 becomes

$$\mathbf{R} = \mathbf{C}\mathbf{s} + \mathbf{Z}, \quad (4.19)$$

where \mathbf{C} is termed the sub-carriers correlation matrix and describes the self-created interference between adjacent sub-carriers. It should be evident that

$$\mathbf{C} = \Phi^H \Phi. \quad (4.20)$$

The two aforementioned methods are equivalent, as established in [130]. Here, a brief proof is presented. Taking into account that \mathbf{B} is orthonormal, this means that $\mathbf{B}^H \mathbf{B} = \mathbf{I}$ where \mathbf{I} is an $N \times N$ identity⁸ matrix. Consequently

$$\mathbf{D}^H \mathbf{D} = (\mathbf{B}^H \Phi)^H \mathbf{B}^H \Phi = (\Phi)^H \mathbf{B} \mathbf{B}^H \Phi = \Phi^H \Phi = \mathbf{C}, \quad (4.21)$$

in other words the Gramian matrix $\mathbf{D}^H \mathbf{D}$ [143] is equal to the sub-carriers correlation matrix \mathbf{C} . Conversely, \mathbf{D} may be obtained through the Cholesky decomposition of \mathbf{C} . A necessary condition for the latter is that \mathbf{C} is symmetric and positive definite [144].

Based on the above and assuming ideal sampling and ideal timing, the linear statistical model for the receiver may be expressed in general form as

$$\mathbf{R} = \mathbf{H}\mathbf{s} + \mathbf{Z}, \quad (4.22)$$

⁸Throughout this thesis the symbol \mathbf{I} will denote an identity matrix of arbitrary size, the size defined according to the mathematical expression under consideration.

where \mathbf{H} is termed the interference matrix and corresponds to either the projections matrix \mathbf{D} or the correlation matrix \mathbf{C} depending on the basis functions $v_n(t)$ employed at the receiver. For the remainder of this thesis, the model incorporating the correlation matrix \mathbf{C} shall be used primarily for two key reasons; first, this matrix can be generated using efficient FFT operations, thus alleviating the need for GS processes, as will be revealed in Section 4.4; second, GS procedures are vulnerable to numerical errors resulting in lower confidence levels. To conclude this section, it is worth mentioning that the projections matrix \mathbf{D} can be generated using Löwdin orthonormalisation which improves performance over the conventional GS methods as it generates vectors which resemble the most the initial set of SEFDM sub-carriers in the Least Squares (LS) sense⁹ [129]. Notwithstanding, the Löwdin algorithm cannot be implemented as efficiently as the matched filtering technique.

4.3 Self-created Interference and Matrix Ill-conditioning

The key challenge in SEFDM systems is the self-created ICI, as a result of the overlapping sub-carriers no longer being mutually independent. This correlation between the sub-carriers leads to the ill-conditioning of the matrix describing the self-created ICI. This ill-conditioning in turn causes the matrix to tend to singularity¹⁰, thus becoming non-invertible and complicating the design of receiver architectures.

Ill-conditioning means that small changes in the entries of the input matrix yield very large changes in the solution matrix. These small changes are typically caused by round-off errors during matrix computations. An illustrative example is shown in Fig. 4.8. This example shows that if a matrix is ill-conditioned, such as \mathbf{A} , then the solution cannot be trusted. An ill-conditioned matrix can become non-invertible even for a very small change in its entries. In SEFDM, the conditioning of a matrix depends on the system size and the level of bandwidth compression. The matrix becomes more

⁹The LS criterion is used to solve a linear set of equations in an approximate sense by minimising the sum of the squares of the residuals [145].

¹⁰A matrix is singular if its determinant is equal to zero [144].

If	$A = \begin{bmatrix} 2 & 1 \\ 4 & 1.999 \end{bmatrix}, \quad X_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad X_2 = \begin{bmatrix} 1.1 \\ 1.9 \end{bmatrix},$
then	$Y_1 = A^{-1}X_1 = \begin{bmatrix} 0.5 \\ 0 \end{bmatrix}, \quad Y_2 = A^{-1}X_2 = \begin{bmatrix} -149.45 \\ 300 \end{bmatrix},$
whereas if	$B = \begin{bmatrix} 5 & 3 \\ 3 & 4 \end{bmatrix},$
then	$Y_3 = B^{-1}X_1 = \begin{bmatrix} -0.1818 \\ 0.6364 \end{bmatrix}, \quad Y_4 = B^{-1}X_2 = \begin{bmatrix} -0.1182 \\ 0.5636 \end{bmatrix}.$

Figure 4.8: Example illustrating the concept of ill-conditioning.

ill-conditioned as the number of sub-carriers N is increased and/or the value of α is decreased leading to performance degradation.

Hence, the obvious question that arises is how is the conditioning of the sub-carriers correlation matrix assessed? This is achieved using two metrics, namely the condition number κ and the machine precision ϵ [146]. The condition number indicates the maximum amount by which an error may be amplified while the machine precision determines the precision of the solution. Let $q = \lfloor \log_{10}[\kappa(A)] \rfloor$ be the number of significant digits lost due to the ill-conditioning of the matrix, where $\lfloor \cdot \rfloor$ denotes the floor function, meaning rounding down to the closest integer. Let $p = \lfloor |\log_{10}(\epsilon)| \rfloor$ be the precision of the machine, where $|\cdot|$ denotes the absolute value of a number. Then the number of significant digits which are correct in the solution is given by $p - q$. In the example illustrated in Fig. 4.8, the condition number of A is equal to $\kappa(A) = 12498$ yielding $q = 4$. For a 64-bit machine with a 52-bit mantissa¹¹, the machine precision is equal to $\epsilon = 2^{-52} = 0.22204 \cdot 10^{-15}$, thus $p = 15$ giving $p - q = 11$. Therefore, in this

¹¹In accordance with the IEEE-754 format [147], the mantissa represents the fixed-point part of a binary number which dictates the precision used to represent the equivalent decimal value. The remaining bits are reserved for the sign bit and the exponent, the latter used to scale the number as required.

case the solution is precise to 11 digits. However, for a 15-bit mantissa, the precision is significantly reduced to $\epsilon = 0.30518 \cdot 10^{-4}$ resulting in $p - q = 0$. This latter scenario indicates that a low machine precision may not be sufficient to represent accurately ill-conditioned matrices.

The condition number may be defined in a number of ways [146]. The most common method is by computing the ratio of the largest ($\check{\sigma}_{max}$) to the smallest ($\check{\sigma}_{min}$) singular value $\check{\sigma}$ obtained via the SVD of the matrix [144] [146]. Mathematically, this is expressed as $\kappa(\mathbf{C}) = \frac{\check{\sigma}_{max}(\mathbf{C})}{\check{\sigma}_{min}(\mathbf{C})}$. Another method is by computing the product of the norms of the matrix and its inverse¹², in other words $\kappa(\mathbf{C}) = \|\mathbf{C}\| \cdot \|\mathbf{C}^{-1}\|$, where $\|\cdot\|$ denotes the ℓ^2 -norm commonly referred to as the Euclidean distance¹³. The method employed in [129] and [130] for computing the condition number is through the ratio of the largest (λ_{max}) to the smallest (λ_{min}) eigenvalue λ obtained through eigenvalue decomposition and expressed as $\kappa(\mathbf{C}) = \frac{\lambda_{max}(\mathbf{C})}{\lambda_{min}(\mathbf{C})}$. Nonetheless, the eigenvalue and singular value decomposition produce the same results only when the matrix is both symmetric (or Hermitian if its entries are complex) and positive definite. A matrix is symmetric (Hermitian) when it is equal to its transpose¹⁴ (conjugate transpose), in other words $\mathbf{C} = \mathbf{C}^T$ (or $\mathbf{C} = \mathbf{C}^H$). A matrix is termed positive semi-definite if its eigenvalues are non-negative and positive definite if its eigenvalues are greater than zero [144].

Fig. 4.9 shows the condition number of the sub-carriers correlation matrix \mathbf{C} for a different number of sub-carriers N and bandwidth compression α . For the right sub-plot, the upper bound for κ has been set to 1000 to improve clarity. Both sub-plot illustrate that the condition number increases exponentially as N is increased beyond 16 and α is decreased below 0.8. The left sub-plot of Fig. 4.9 also indicates that \mathbf{C} eventually reaches a saturation point where it becomes severely ill-conditioned with $\kappa(\mathbf{C}) > 10^{16}$, for example, when $N = 32$ and $\alpha = 0.7$. At this point, \mathbf{C} tends to singularity, thus becoming non-invertible. It also ceases to be positive definite, which

¹²The inverse of a matrix is denoted by \mathbf{A}^{-1} .

¹³The terms Euclidean distance, Euclidean norm and ℓ^2 -norm will be used interchangeably throughout this thesis.

¹⁴The transpose of a matrix is denoted by \mathbf{A}^T .

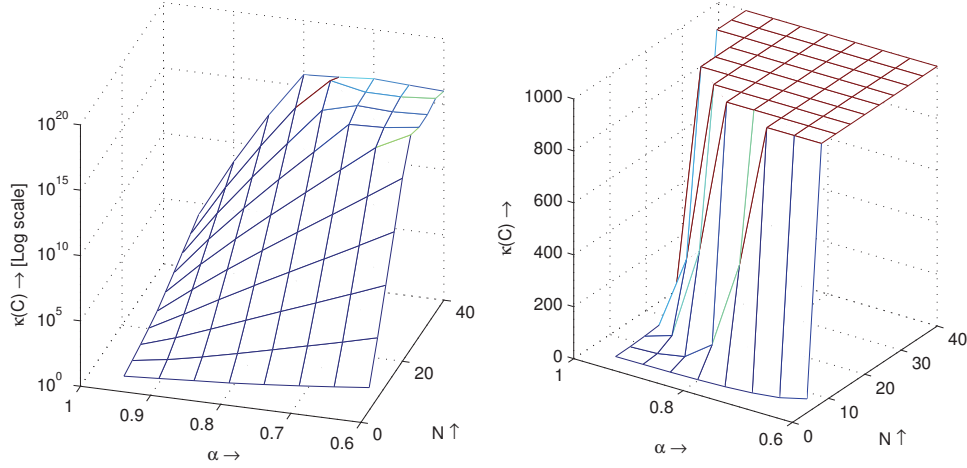


Figure 4.9: Condition number of sub-carriers correlation matrix for different values of N and α .

presents a significant challenge as the matrix has to be positive definite in order to apply Cholesky decomposition, the latter required during the execution of certain detection algorithms, such as SD.

Fig. 4.10 depicts the SEFDM crosstalk response for $N = 32$ and $\alpha = \frac{2}{3} \approx 0.67$ which is denoted by the amplitude of the sub-carriers correlation matrix. It is clear that the matrix has a Toeplitz format¹⁵. Although not shown here, the sub-carriers correlation matrix \mathbf{C} is also Hermitian, as determined in [130].

Fig. 4.11 illustrates the same matrix \mathbf{C} , however, in this case the matrix has been pruned by setting the very small¹⁶ values equal to zero. Hence, Fig. 4.11 gives a clearer view of the ICI caused to the desired symbols on the main diagonal by the neighbouring sub-carriers on the secondary diagonals. The number of small singular values in \mathbf{C} (similarly in the projections matrix \mathbf{D}) has been found to be approximately equal to $(1 - \alpha)N$ [129] [130].

Fig. 4.12 shows the amplitude of the inverse of the sub-carriers correlation matrix.

¹⁵A matrix which has constant diagonals is known as a Toeplitz matrix [144].

¹⁶defined as being below a user-defined threshold, in this case fixed at 10^{-10} .

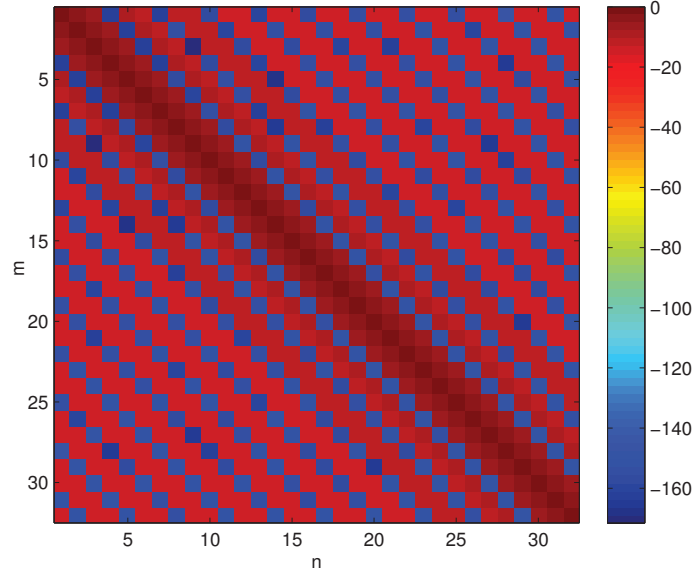


Figure 4.10: Amplitude of $C_{m,n}$ in dB ($N = 32$, $\alpha \approx 0.67$).

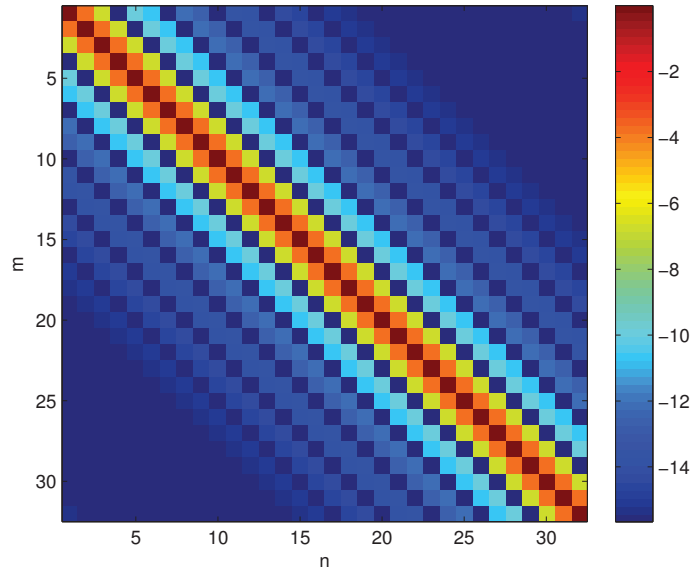


Figure 4.11: Amplitude of pruned $C_{m,n}$ in dB ($N = 32$, $\alpha \approx 0.67$).

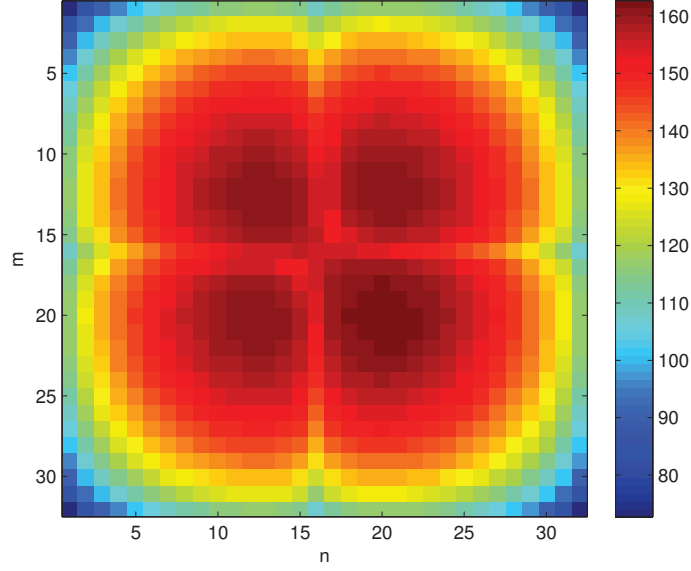


Figure 4.12: Amplitude of $\mathbf{C}_{m,n}^{-1}$ in dB ($N = 32$, $\alpha \approx 0.67$).

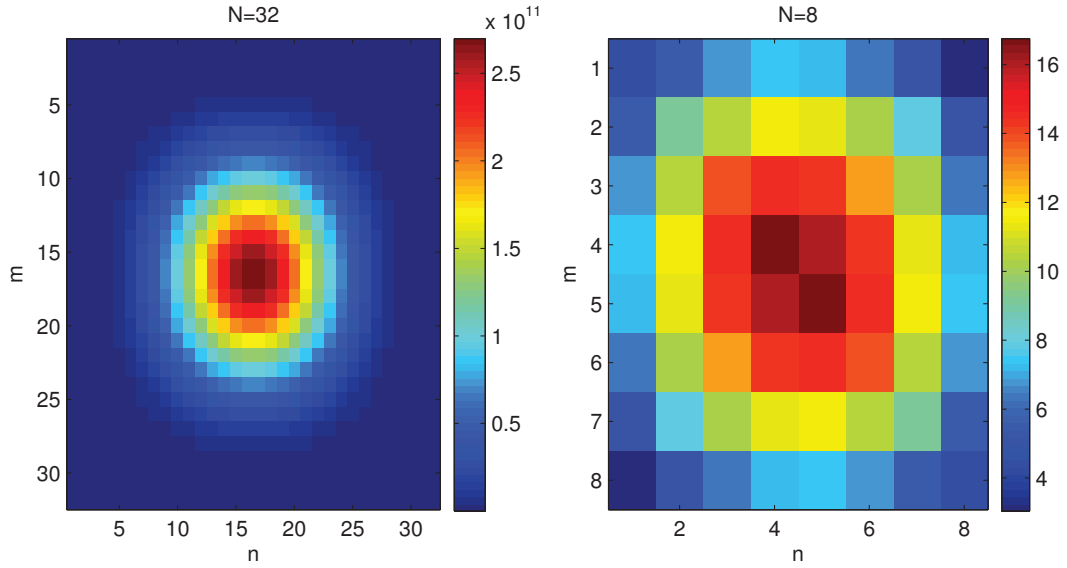


Figure 4.13: Amplitude of $\mathbf{C}_{m,n}^{-1}$ for different values of N with $\alpha = 0.8$.

It is clear that the format of this matrix is bisymmetric¹⁷. The significance of this result will become evident during the discussion of hardware design decisions in a latter part of this thesis. Finally, Fig. 4.13 illustrates the effect of ill-conditioning. The left and right sub-plots show the amplitude of \mathbf{C}^{-1} for $N = 32$ and $N = 8$, respectively, when $\alpha = 0.8$. It is evident that for $N = 32$ the vast majority of the values around the centre of the matrix tend to zero.

4.4 Sampling Issues and Spectral Leakage in DFT based SEFDM Systems

Hitherto, the model for an OFDM system has assumed a one-to-one relationship. In other words, the number of input data symbols, the length of the IDFT block, the number of sub-carriers and the number of time samples contained in a single OFDM symbol have identical values. Hence, for an OFDM system, the frequency spacing Δf is equal to the reciprocal of the OFDM symbol period T with the latter given by $T = NT_s$.

Since Section 4.2.1 identified that SEFDM signals can be generated using standard IDFT blocks with the aid of zero padding, this section aims to explain the concept and consequences of zero padding in practical scenarios. Two key assumptions are made; first the same sampling frequency f_s is adopted in all cases; second, the entire transmission takes place at baseband, in other words digital-over-digital. These assumptions allow the signal to be treated as simply a discrete sequence of amplitudes. In addition, operating at baseband permits the use of negative frequencies.

In OFDM and SEFDM, the IDFT inputs are considered to represent samples in the frequency domain. Zero padding in the frequency domain corresponds to interpolation in the time domain [91]. Thereby, by adding trailing zeros to the input data symbols, the resultant time domain signal will be interpolated, in other words upsampled. Recalling

¹⁷A square matrix is termed bisymmetric if it exhibits symmetry about both its main and northeast-southwest diagonals [142].

from Section 2.3.3, the generation of OFDM signals using the IDFT without zero padding obeys the Nyquist criterion by default, since the highest frequency is equal to $(\frac{N}{2} - 1)\Delta f$. Hence, the upsampling that occurs with zero padding is essentially equivalent to oversampling, meaning that more than one sample is used to represent each data symbol.

At this point, the question that arises is, what are the consequences of zero padding, or in this case, oversampling? Since the number of sub-carriers has not changed, the overall bandwidth of the signal will remain the same provided that the frequency spacing Δf between the sub-carriers is maintained. To account for the increase in the number of samples contained in the time domain signal while concurrently maintaining the strict orthogonality rule, this implies that a subsequent decrease in data rate will occur. To comprehend this, let $T_{US} = N_{IDFT}T_s$ with $N_{IDFT} = \rho N > N$, where T_{US} denotes the period of the upsampled signal and N_{IDFT} denotes the size of the IDFT block. The size of this block is equal to the total number of samples in the upsampled time domain signal. The orthogonality rule may then be expressed as

$$\Delta f = \frac{1}{T_{US}} = \frac{1}{N_{IDFT}T_s} = \frac{1}{\rho NT_s} = \frac{1}{\rho} \frac{1}{T}. \quad (4.23)$$

Eq. 4.23 signifies that by zero-padding the IDFT and considering all the IDFT outputs for the construction of the discrete OFDM signal, the frequency spacing and the orthogonality of the sub-carriers is maintained at the expense of reduction in the width of each sub-carrier commensurate with the reduction in data rate. These concepts are illustrated in Fig. 4.14.

For the relationship between Δf and T to hold ($\Delta f = \frac{1}{T}$), in order to guarantee orthogonality, the reciprocal of the above case should also hold, in other words increasing the data rate should result in a corresponding reduction in frequency spacing. Here, it is shown that this is actually true. Let it be assumed that only a fraction α of the zero padded IDFT output with $\alpha < 1$ is considered, in other words $T_Q = QT_s$ with $Q = \alpha N_{IDFT}$. Furthermore, let it be assumed that α is such that $Q = N$ (so essentially

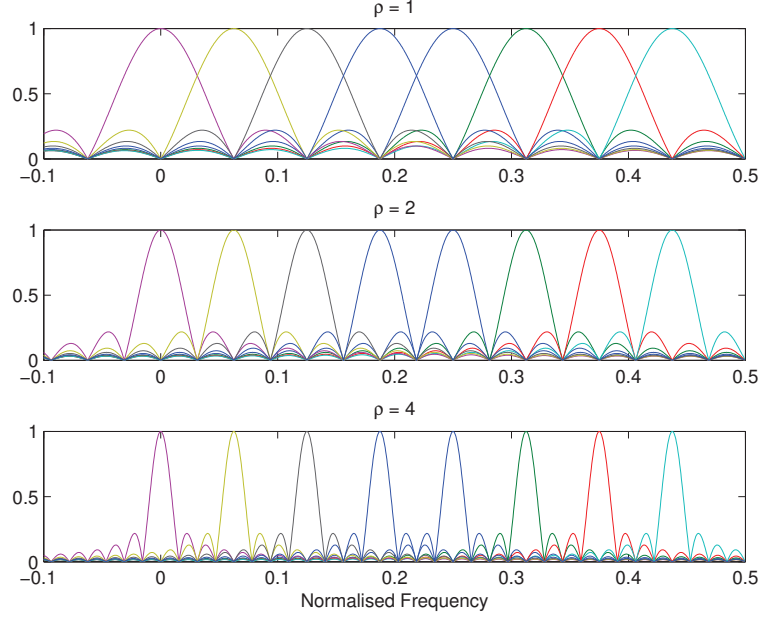


Figure 4.14: Magnitude response of OFDM for different oversampling factors.

$\alpha = \frac{1}{\rho}$). Then Eq. 4.23 becomes

$$\Delta f = \frac{1}{N_{IDFT}T_s} = \frac{1}{\alpha \frac{N_{IDFT}T_s}{\alpha}} = \frac{1}{\frac{QT_s}{\alpha}} \stackrel{Q=N}{\Leftrightarrow} \Delta f = \frac{\alpha}{NT_s} = \frac{\alpha}{T}. \quad (4.24)$$

Eq. 4.24 shows that a zero padded IDFT block can in fact achieve the same data rate as a conventional IDFT block without zero padding provided that the frequency spacing between the sub-carriers is reduced. This in turn is achieved by truncating the IDFT output which in turn constitutes the underlying principle in using the IDFT to generate SEFDM signals. The downside to this truncation is that it breaks the orthogonality between the sub-carriers.

The similarities and fundamental differences between OFDM and SEFDM systems may be better comprehended through an example. First, consider a 16-point IDFT and a 20-point IDFT on the unit circle, as shown in Fig. 4.15. The IDFT generates frequencies which are periodic in ω with period 2π given by $\omega_n = n2\pi \frac{B}{N_{IDFT}}$. Assuming that the same bandwidth B is available in both cases, it should be evident that the

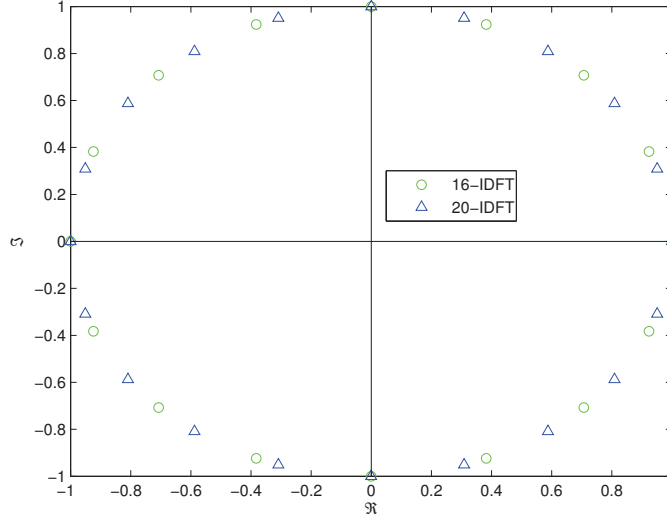


Figure 4.15: 16-point IDFT and 20-point IDFT on the unit circle.

frequency spacing Δf_{20p} for the 20-point IDFT will be smaller than the frequency spacing Δf_{16p} for the 16-point IDFT. More specifically, the points on the unit circle represent the IDFT matrix entries given by Ω^{mn} , where $\Omega = e^{\frac{j2\pi}{N_{IDFT}}}$ denotes the N^{th} root of unity and is commonly referred to as the *twiddle factor* [148].

Now let it be assumed that only 16 sub-carriers in the 20-point IDFT are activated while the remaining four are zero padded. In theory, the frequency spacing Δf_{20p} should remain the same since the 20-point IDFT is generated in exactly the same way. As explained earlier, via zero padding more samples have essentially been used to represent the same number of sub-carriers as in the 16-point IDFT. Thanks to the time-frequency duality, expansion of a signal in one domain corresponds to compression of the signal in the other domain and vice versa. Through oversampling, the time domain signal at the IDFT output has been expanded which results in the width of each sub-carrier being compressed. Consequently, for the orthogonality rule to remain valid, the actual frequency spacing between the sub-carriers increases, as illustrated in Fig. 4.14. Hence the reason the frequency spacing for a 20-point IDFT appears to be the same as that for a 16-point IDFT in practice when in theory it should be smaller.

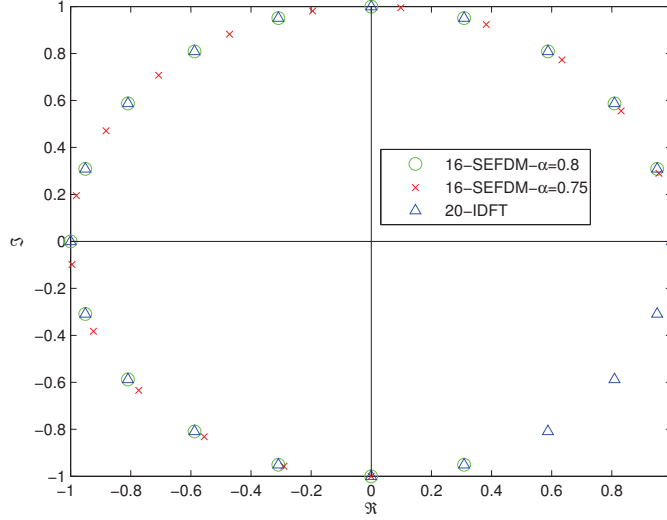


Figure 4.16: 20-point IDFT versus 16-point SEFDM sub-carriers matrix.

Following on from the aforementioned explanation, the 20-point IDFT output is truncated now to 16 samples to yield an SEFDM signal, in other words $\alpha = \frac{Q}{N_{IDFT}} = \frac{16}{20} = 0.8$. This results in the SEFDM signal having the same number of time samples as a 16-point IDFT OFDM signal, and therefore the same data rate. However, the frequency spacing actually corresponds to that for a 20-point IDFT. This can be better understood by comparing a 20-point IDFT to a 16-point SEFDM sub-carriers matrix (generated in an analogue fashion as $\Phi^H \Phi$) with $\alpha = 0.8$. From Fig. 4.16, it is clear that the 16-points of the SEFDM sub-carriers matrix coincide with the first 16 points of the 20-point IDFT matrix. Fig. 4.16 also shows the distribution of the points for $\alpha = \frac{3}{4} = 0.75$, which in practice cannot be implemented using a Type 1 transmitter since $\frac{16}{0.75} \approx 21.33$, which is not an integer number.

Fig. 4.17 compares the time domain signals for a 16-point OFDM signal, a 20-point OFDM signal and a 16-point SEFDM signal. For clarity, only the first sub-carrier is illustrated. It is obvious that the lowest frequency equal to $1 \cdot \Delta f$ (excluding the DC component) is smaller for the 20-point IDFT compared to the 16-point IDFT. Fig. 4.17 also shows that the SEFDM signal is essentially an abrupt cut-off of the would-have-

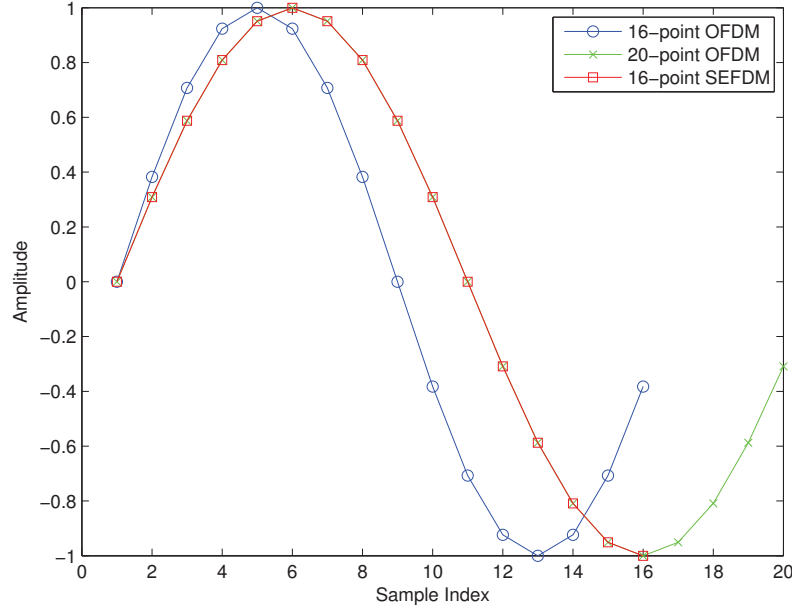


Figure 4.17: OFDM and SEFDM signals in the time domain.

been OFDM signal generated using the 20-point IDFT.

The use of the IDFT in OFDM systems guarantees an integer number of cycles in the IDFT interval. Two closely related effects that may appear in DFT based systems are aliasing and spectral leakage. Aliasing refers to the peaks of the desired signal appearing at the wrong frequencies due to them folding over from outside the $\frac{f_s}{2}$ band into the 0 to $\frac{f_s}{2}$ band. Spectral leakage refers to the ‘smearing’ of frequency amplitudes as a result of the received frequency not having an integer number of cycles in the DFT interval. For example, let $f_s = 48kHz$, $N_{DFT} = 16384$ and $n = 256$ be the number of cycles corresponding to the desired frequency $f_x = \frac{n f_s}{N_{DFT}} = 750Hz$. If this frequency arrives at the receiver as $\tilde{f}_x = 750.33Hz$, this will correspond to $n \approx 256.11$ cycles which is not an integer number. Consequently, the DFT bins will not line up with the peaks of the received signal’s frequency amplitudes causing inter-bin interference where the energy of one bin spills into adjacent bins [91]. These effects give rise to ICI in OFDM systems. The self-created ICI in SEFDM systems can thus be attributed to the

same effects by making two key observations based on Fig. 4.17:

1. The truncation of the 20-point IDFT signal's tail will give rise to discontinuities between successive segments, which will appear as high frequency harmonics in the frequency domain, thus causing (deliberate) aliasing.
2. As a result of this truncation, the SEFDM signal will not have an integer number of cycles in the DFT time record, thus leading to spectral leakage.

Similar to OFDM, as the number of sub-carriers is increased in SEFDM, the system becomes more prone to ICI. Since SEFDM systems have inherent interference, it should be evident that even small perturbations, for example due to AWGN, could significantly degrade system performance.

Finally, a direct consequence of using the IDFT and DFT operations to modulate and demodulate SEFDM signals is that the same operations can be used to generate the sub-carriers correlation matrix \mathbf{C} . Recalling from Eq. 4.20, $\mathbf{C} = \mathbf{\Phi}^H \mathbf{\Phi}$, where $\mathbf{\Phi}$ denotes the $N \times N$ SEFDM sub-carriers matrix assuming $\rho = 1$. Applying the IDFT operation to an input column vector gives a column vector. Consequently, if the IDFT operation \mathcal{F}^{-1} of size $\frac{N}{\alpha} \times N$ is applied to an identity matrix of size $N \times N$, the result will be the entries of the actual $\frac{N}{\alpha} \times N$ IDFT matrix denoted by \mathbf{F} . The SEFDM sub-carriers matrix is equal to the first N rows of the IDFT matrix. Mathematically, this is expressed as

$$\Phi_{m,n} = \mathbf{F}_{m,n}^{-1}, \quad (4.25)$$

where $m, n = 0, \dots, N - 1$.

Subsequently, a DFT operation \mathcal{F} of size $\frac{N}{\alpha} \times N$ is applied to $\mathbf{\Phi}$ yielding $\bar{\mathbf{C}} = \mathcal{F}\{\mathbf{\Phi}\}$. Once again, the last $\frac{N}{\alpha} - N$ entries are discarded resulting in the sub-carriers correlation matrix $\mathbf{C}_{m,n} = \bar{\mathbf{C}}_{m,n}$ with $m, n = 0, \dots, N - 1$. It is presumed that appropriate scaling is applied at each stage of the aforementioned process so that the interference values in the sub-carriers correlation matrix are normalised to unit energy. The concluding remark for this section is that the sub-carriers correlation matrix \mathbf{C} generated

with the aforementioned IDFT/DFT techniques matches the matrix generated with the conventional technique employing a bank of modems, if and only if the ratio $\frac{N}{\alpha}$ is an integer.

4.5 Discussion on the Complexity and Performance of SEFDM Modems

The complexity of a system can be defined using different metrics [61]. So far, the two most popular metrics that have been considered include the complexity order, denoted by $\mathcal{O}(\cdot)$, and the computational complexity. The former denotes the fastest growing term, for example when referring to a system employing DFT operations of order $\mathcal{O}(N^2)$, this means that the system's complexity is quadratic in N . Conversely, the computational complexity refers to the processing time required to run an algorithm, which in turn depends on the number of arithmetic operations that need to take place during the execution of the algorithm.

In this chapter, it was demonstrated that SEFDM signals can be generated and demodulated using standard DFT operations. This is a significant result, as it is well known that DFT operations can be executed far more efficiently using the FFT algorithm. Thanks to the FFT algorithm, the complexity order of a DFT operation is reduced from $\mathcal{O}(N^2)$ to $\mathcal{O}(N \log_2 N)$. This is due to the fact that the multiplications in the DFT are replaced by phase rotations in the FFT. Taking into account that the complexity order is usually quantified in terms of the number of multipliers required [61], it is evident that the complexity of the FFT grows slightly faster than linear compared to the conventional DFT whose complexity grows quadratically.

With regard to the three different SEFDM transmitter techniques explored in Section 4.2.1, a number of important remarks must be made since these have a direct impact on hardware design decisions:

- First, it was demonstrated that using a Type 1 SEFDM transmitter, the reduction

in frequency spacing can be achieved by zero padding the input and truncating the output of an IDFT block. However, the execution time for an N -point IDFT is lower compared to the execution time for an $\frac{N}{\alpha}$ -point IDFT. Moreover, simply discarding the unused outputs does not result in a reduction of complexity nor resource utilisation, as will be revealed in subsequent chapters. Instead, this truncation actually results in burst data which reduces the effective data rate. Hence, using this transmitter configuration, the benefit of using SEFDM over OFDM becomes questionable.

- The previous point assumed that no oversampling is present. In other words, the number of sub-carriers equals the number of time samples implying that no zero padding is used in OFDM. Ordinarily, in practical OFDM systems, zero padding is actually employed to prevent aliasing when the discrete time samples are filtered through a D/A converter and a LPF [86]. In WiMAX systems, zero padding is used for the purpose of nulling sub-carriers which serve as guard bands [20]. Oversampling has been used in multi-carrier systems for channel estimation [149], as well as for mitigating the effects of ICI [150]. Consequently, a more fair comparison of SEFDM against OFDM would be to test both systems in a scenario which calls for oversampling. Assuming the same DFT sizes were employed for both systems, this would infer that the oversampling factor for SEFDM would be lower compared to that for OFDM.
- As a converse to the previous point, another possible case would be to truncate the output of a standard OFDM system without oversampling. In other words, if an N -point IDFT is used to process N sub-carriers, the number of time samples for OFDM would be N . If now the output was truncated and only Q samples with $Q = \alpha N$ were taken forward, this would imply that the corresponding SEFDM system is undersampled since the number of time samples is actually smaller than the number of sub-carriers, in other words $\rho < 1$ leading to $Q < N$. Assuming that the issue of burst data could be tackled appropriately to yield

continuous data, this would mean that the effective data rate for SEFDM would be higher compared to that for OFDM, taking into account that the same size IDFT is employed, thus the execution time required in both cases is identical. Undersampling a signal though means that not all signal variations are caught which could potentially result in performance degradation. This latter topic is examined in further detail in Chapter 5.

The three aforementioned points assumed that a Type 1 SEFDM transmitter is employed. Hereafter, Type 2 and Type 3 transmitter schemes are considered and once again for simplicity, let $\rho = 1$ yielding $Q = N$ for the SEFDM system under consideration. As demonstrated in [130], the complexity order of both Type 2 and Type 3 transmitters is the same. Yet, the parallel configuration favoured by the Type 3 transmitter renders it suitable for FPGA implementation, as expounded in [59]. Considering the IDFT processing stage in isolation, the Type 3 transmitter results in the lowest possible latency by using multiple instances of an N -point IDFT block configured in parallel. Nonetheless, there are two key drawbacks with this transmitter technique:

1. The reordering stage prior to the IDFT stage has a latency of cN clock cycles assuming that the optimum *index implementation* method, as described in [59], is employed. Even if the latency of the post-IDFT block is negligible, the total latency for the Type 3 transmitter is $N + cN$, hence it is clear that the value of c has a direct impact on the overall latency of the system. The Type 3 transmitter also results in the highest resource utilisation. Conversely, the Type 1 transmitter only requires a single IDFT block whose processing latency and resource utilisation are directly determined by its size. The upper bound on this size is given by the highest number of sub-carriers that need to be supported and the lowest value of α , as defined by the ratio $\frac{N}{\alpha}$. Thence, it may be the case that the Type 3 transmitter does not yield the lowest overall latency and therefore its preference over a Type 1 transmitter is questionable.

2. It is true that the Type 2 and Type 3 transmitters offer greater flexibility over the Type 1 transmitter since the values of N and $\alpha = \frac{b}{c}$ can be chosen independently. As illustrated in [59], the maximum value of c to prevent the hardware design exceeding the device's available resources is approximately 12. Consequently, this imposes a strict upper bound on the granularity of α which cannot be any finer than $\frac{1}{12} \approx 0.083$. On the other hand, the granularity of the Type 1 transmitter is directly determined by the size of the IDFT block, in other words $\frac{1}{N_{IDFT}}$ considering that $\alpha = \frac{N}{N_{IDFT}}$. Based on the analysis so far and as will be revealed in subsequent chapters, it is expected that $N_{IDFT} > 12$.

These points serve to validate the choice of employing a Type 1 SEFDM transmitter and its reciprocal at the receiver for the purpose of this thesis. They also motivate the discussion of Chapter 5 which examines the subtle trade-offs between throughput and error performance. These system metrics are determined by the use of oversampling and the fixed-point effects present in a practical environment due to finite arithmetic precision. Finally, the range of values for these latter system parameters, in relation to oversampling and arithmetic precision, are limited by the actual architecture options available to a hardware designer.

4.6 Conclusions

This chapter presented the underlying principles of SEFDM systems and discussed practical aspects regarding the modulation and demodulation of such signals. The chapter commenced with a review of spectral efficiency as first described by Shannon in 1948. The spectral efficiency of a conventional single carrier system was compared to that of OFDM which, for a large number of sub-carriers, is identical to that of a single carrier system with the added benefit of robustness against multipath fading and channel delay spread. The notion of time-frequency packing was then explored which lies at the core of spectrally efficient multi-carrier systems sharing the same characteristics

as SEFDM. It was shown that SEFDM can achieve a greater spectral efficiency than OFDM without error penalties provided the level of bandwidth compression is within certain limits as dictated by $\alpha \geq 0.8$. For $\alpha < 0.8$, SEFDM can achieve an even higher spectral efficiency compared to OFDM, however, this comes at the expense of increased E_b/N_0 or degraded BER performance. The significance of these results compared to previous work, is that the claimed bandwidth efficiency gains can be achieved in the real world, for the first time, using a practical detector with tangible computational complexity suitable for hardware implementation.

The attention was then drawn to the application of the DFT in SEFDM systems. Due to the inherently non-orthogonal signals generated using SEFDM, the standard IDFT/DFT operations cannot be directly employed as in OFDM systems. While the IFRFT is based on the IDFT and may be used to generate SEFDM signals, its implementation still does not conform to the standard blocks deployed in existing OFDM systems. However, recent research efforts were cited which have achieved in arranging standard IDFT blocks in unique configurations for use in SEFDM systems.

The next area of interest examined the receiver model defined for an SEFDM system considering an AWGN channel. Two types of modulation/demodulation techniques were identified and described, namely matched filtering and orthonormalisation procedures. It was explained that matched filtering is preferred over orthonormalisation, since the latter technique cannot be implemented using efficient FFT operations and is also vulnerable to numerical errors.

Subsequently, the key challenge encountered in SEFDM systems, specifically the ill-conditioning of the projections or sub-carriers correlation matrix, was studied. The concept of ill-conditioning was explained followed by an account of the reasons it manifests itself in SEFDM systems, as a result of deliberately and counter-intuitively violating the strict orthogonality rule defined for OFDM systems. A method for quantifying the severity of ill-conditioning with the aid of the matrix's condition number and the machine's precision was demonstrated. This quantification is paramount, as it deter-

mines the sensitivity of the system to perturbations and thus the accuracy of the results acquired.

Finally, issues of sampling, deliberate aliasing and spectral leakage in DFT based SEFDM systems were addressed, since the potential of employing IDFT/DFT operations to modulate and demodulate non-orthogonal signals is invaluable. This is particularly true in view of the fact that new Fourier transform based algorithms are coming to light, which claim to run even faster than the state-of-the-art FFT and its variants, under certain conditions. One of the latest examples includes the Faster FFT developed by MIT [151]. The diverse trade-offs associated with previously reported SEFDM transmitters in terms of complexity and performance, especially when targeted for hardware implementation, were also discussed.

In conclusion, Chapter 4 provided the impetus for employing SEFDM in the real world. This chapter served as an important introduction to the nature of SEFDM and the research surrounding spectrally efficient systems. Chapter 5 examines the influence of practical design considerations, such as oversampling and fixed-point effects, on the performance of techniques employed to address the detection challenge in SEFDM.

Chapter 5

SEFDM Detection and Practical Considerations

Chapter 4 explained that SEFDM provides the same immunity against channel impairments as OFDM while making better use of the available spectrum. This chapter proceeds to the evaluation of the performance of SEFDM systems in practical scenarios taking oversampling and fixed-point effects into account.

The aim of this investigation is to identify the upper and lower bounds required for SEFDM to perform satisfactorily in the real world, with performance commensurate to that obtained in an ideal environment which assumes full, floating-point precision. A fine trade-off exists between the limited availability of hardware resources and the performance of SEFDM in terms of throughput and BER. As will be revealed, a higher degree of oversampling and a wider dynamic range yield better error performance at the expense of decreased throughput and/or increased resource utilisation, the latter incurring additional hardware costs. Hence, a thorough analysis to quantify these trade-offs is carried out.

To this end, this chapter commences with the preliminaries of SEFDM detection. Section 5.1 describes different linear and iterative detection techniques and compares them according to their complexity and ease of implementation. Section 5.2 shows the

impact of oversampling on the conditioning of the sub-carriers correlation matrix. The error performance of the detection techniques outlined in Section 5.1 is also compared for varying levels of oversampling. While iterative techniques offer the best performance as standalone detectors, it is explained that TSVD is the preferred method for hardware realisation, thanks to its more straightforward operation and performance enhancement when combined with sophisticated algorithms, such as FSD. Finally, Section 5.3 studies in detail the effects of finite arithmetic precision and quantisation errors on the performance of SEFDM systems. A bit-accurate FPGA model of a DFT based SEFDM transceiver was developed for validating the fixed-point representation of different blocks. This model also serves as a reference for making subsequent hardware design decisions. A number of detection techniques employed in SEFDM along with practical design considerations were surveyed by the author and presented at the London Communications Symposium (LCS) in 2010 [152].

5.1 SEFDM Detection Techniques

In Chapter 4, the nature, principles and modulation/demodulation techniques of SEFDM systems were considered. This chapter details the algorithms previously developed for the detection of SEFDM signals [129] [130].

Fig. 5.1 depicts a practical transceiver comprising three different stages for the demodulation and detection of SEFDM signals. The first stage is a demodulator, which may be implemented using orthonormalisation processes or matched filtering. Matched filtering is preferred since it is the optimum receiving filter and can be realised using DFT operations. The second involves the use of linear or iterative techniques while the third stage employs more sophisticated algorithms, such as ML or SD. A snapshot of the demodulation and detection methods employed in SEFDM is depicted in Fig. 5.2. It is not feasible to evaluate all these schemes from a practical perspective in this thesis. Hence, the focus shall be on the schemes indicated in the text boxes surrounded by a thick border line as per Fig. 5.2. The justification for these algorithmic choices

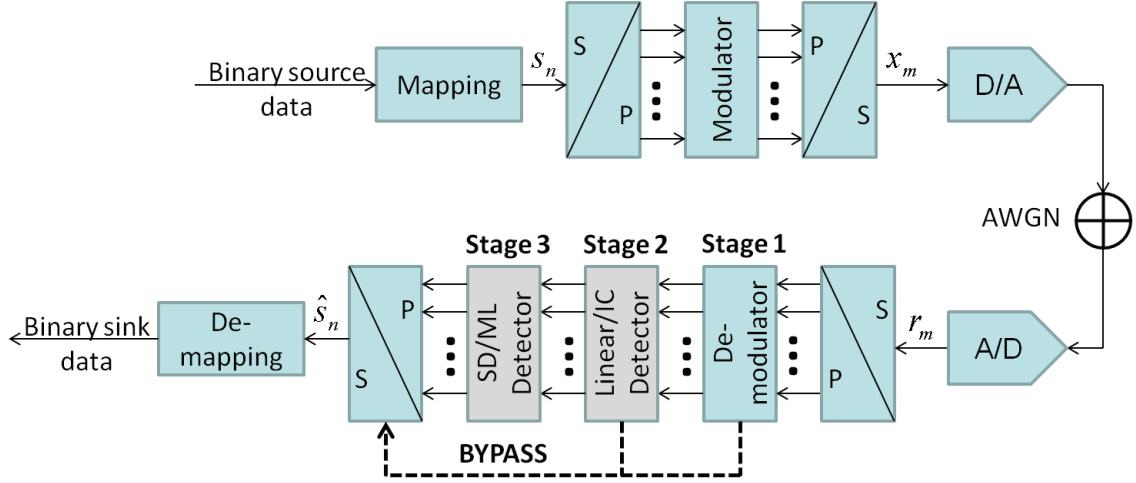


Figure 5.1: Practical SEFDM transceiver.

will be explained in the forthcoming analysis and discussion.

This chapter focuses on linear detection methods, in particular ZF and TSVD, while Chapter 7 considers more advanced techniques, such as FSD. Alternative detectors including Selective Equalization (SeE), Minimum Mean Squared Error (MMSE), ML and SD have been explored in previous work [129] [130]. In terms of linear detection, previous results have shown that TSVD outperforms ZF, SeE and MMSE, hence the reason for its subsequent hardware implementation.

The linear detection stage may be substituted by iterative detection, as illustrated by the dotted lines in Fig. 5.3. While the focus in this thesis is on linear and tree-search detection methods, it is worth touching upon iterative techniques considering that an iterative decoding algorithm employing Successive Interference Cancellation (SIC) has been applied in FOMS [51], FTN [153] and GFDM [111]. Furthermore, FOMS systems have considered the use of Decision Feedback Equalisation (DFE) [154], as well as soft cancellation followed by MMSE [155] for signal estimation. Recent work has also demonstrated that iterative detection with soft cancellation can improve the performance of SEFDM systems in terms of BER with a targeted computational complexity [135]. The following subsections examine iterative detection techniques in

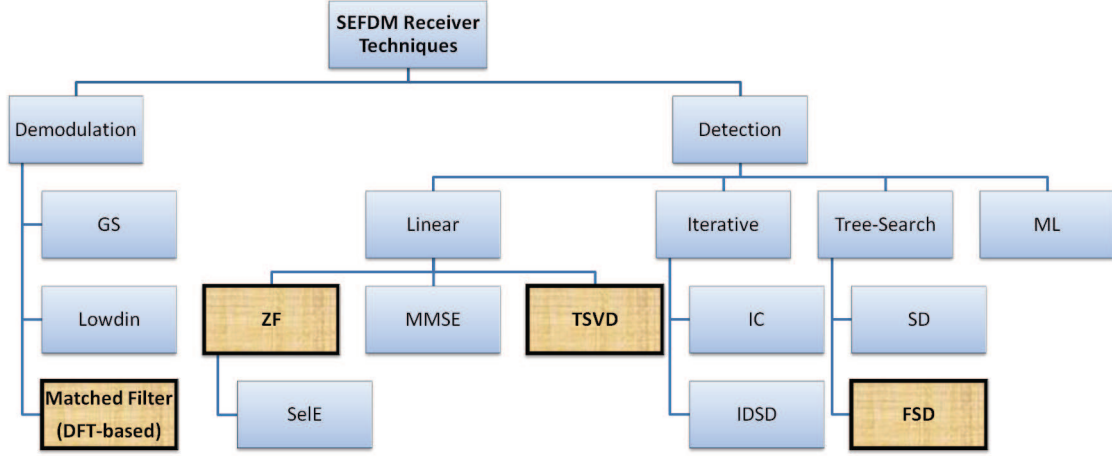


Figure 5.2: SEFDM demodulation and detection techniques.

addition to the aforementioned linear detection methods and compare their performance in terms of BER, complexity and their suitability for FPGA implementation.

Section 4.2 explained that the SEFDM demodulator is responsible for generating the statistics vector \mathbf{R} . Each element of this vector \mathbf{R} is merely a discrete sample which has a value proportional to the energy of the received symbol with the added noise. The task of the detector is to obtain the best possible estimate $\hat{\mathbf{s}}$ of each transmitted data symbol, where $\hat{\mathbf{s}}$ denotes the estimate of the transmitted signal vector \mathbf{s} after slicing, based on the values of \mathbf{R} and the properties of the sub-carriers correlation matrix \mathbf{C} .

Recalling from Eq. 4.22, the linear statistical model for the received SEFDM signal is given by $\mathbf{R} = \mathbf{H}\mathbf{s} + \mathbf{Z}$. Linear detection methods treat this relation as an unconstrained linear estimation problem and thus aim to recover the transmitted symbols by cancelling out the self-created interference. The most popular methods include ZF and MMSE with the former yielding the LS solution. If the estimator matrix used to generate the symbol estimates is \mathbf{G} , then the solution to the unconstrained problem

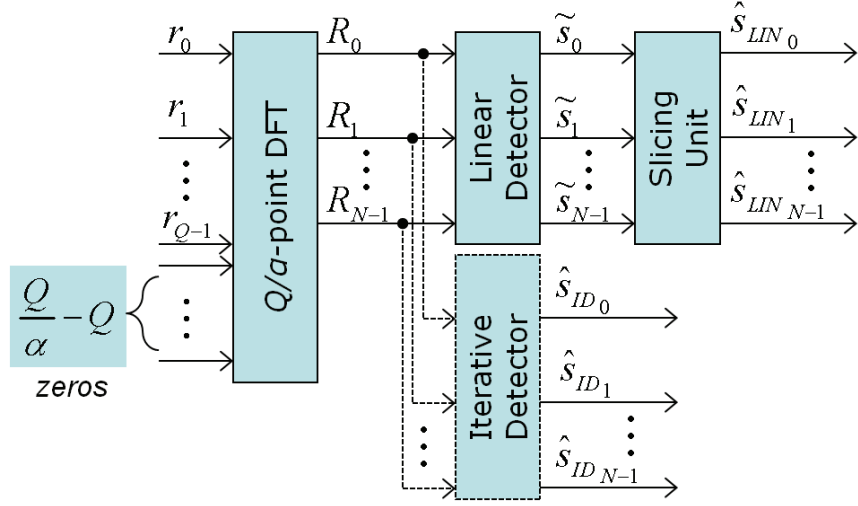


Figure 5.3: SEFDM detection employing linear or iterative techniques.

with respect to Eq. 4.22 is given by

$$\tilde{\mathbf{s}} = \mathbf{GR}, \quad (5.1)$$

with $\tilde{\mathbf{s}} \in \mathbb{C}^N$, where $\tilde{\mathbf{s}}$ denotes the unconstrained estimate of the transmitted signal vector \mathbf{s} before slicing and \mathbb{C}^N is the set of all complex N -tuples¹ assuming a complex constellation scheme is employed, for example 4-QAM. If the constellation scheme contains only real values, such as BPSK, the set \mathbb{C}^N may be reduced to the set \mathbb{R}^N .

5.1.1 Linear Detection

In this section, linear detection techniques are considered to address the detection challenge in a similar fashion to the work carried out by Kanaras [129] in SEFDM and by Burg [61] in MIMO systems. The solution of Eq. 5.1 is referred to as the unconstrained estimate since it does not take into account that the elements of \mathbf{s} can in fact take only the values pertaining to a limited set of constellation points defined in a discrete alphabet \mathfrak{M}^N . To this end, the problem is constrained by applying a slicing

¹An N -tuple is a sequence or ordered set of N elements.

operation to each of the entries of $\tilde{\mathbf{s}}$ yielding

$$\hat{\mathbf{s}} = \lfloor \tilde{\mathbf{s}} \rfloor, \quad (5.2)$$

with $\hat{\mathbf{s}} \in \mathbb{Z}^N$ where \mathbb{Z}^N denotes the set of integer N -tuples and $\lfloor \cdot \rfloor$ denotes a slicing operator, which rounds the value to the nearest constellation point.

The ZF detector generates symbol estimates by forcing the interference terms to zero. The estimator matrix is given by the Moore-Penrose pseudoinverse [142] of the matrix \mathbf{H} describing the self-created interference in SEFDM systems and expressed as

$$\mathbf{G} = \mathbf{H}^\dagger = (\mathbf{H}^H \mathbf{H})^{-1} \mathbf{H}^H, \quad (5.3)$$

recalling from Section 4.2.2 that \mathbf{H} may be set to equal the sub-carriers correlation matrix \mathbf{C} or the projections matrix \mathbf{D} .

Taking into account that \mathbf{H} is a square $N \times N$ matrix and upon the condition that \mathbf{H} is invertible, the Moore-Penrose pseudoinverse is the same as the straightforward inverse, hence Eq. 5.3 becomes

$$\mathbf{G} = \mathbf{H}^{-1}, \quad (5.4)$$

Based on Eq. 5.4, the ZF estimate is given by

$$\hat{\mathbf{s}}_{ZF} = \lfloor \mathbf{H}^{-1} \mathbf{R} \rfloor = \mathbf{H}^{-1} \mathbf{H} \mathbf{s} + \mathbf{H}^{-1} \mathbf{Z} = \mathbf{s} + \mathbf{H}^{-1} \mathbf{Z}, \quad (5.5)$$

which shows that the noise vector \mathbf{Z} is multiplied by the inverse of the interference matrix \mathbf{H}^{-1} . Even though ZF completely eliminates interference, it could potentially amplify the noise to a great extent, especially at points where spectral nulls occur [80], leading to severe performance degradation. This noise amplification in turn is directly determined by the properties of the interference matrix in terms of its ill-conditioning.

MMSE improves performance over ZF by taking the presence of noise into account with the aim of minimising the total expected error. In this case, the estimator matrix

is given by

$$\mathbf{G} = \mathbf{H} \left(\mathbf{H}\mathbf{H}^H + \frac{\sigma^2}{\sigma_s^2} \mathbf{I} \right)^{-1}, \quad (5.6)$$

where σ^2 and σ_s^2 denote the noise and signal power, respectively, while the ratio $\frac{\sigma^2}{\sigma_s^2}$ represents the inverse of the SNR. Consequently, the MMSE estimate is given by

$$\hat{\mathbf{s}}_{MMSE} = \left[\mathbf{H} \left(\mathbf{H}\mathbf{H}^H + \frac{\sigma^2}{\sigma_s^2} \mathbf{I} \right)^{-1} \mathbf{R} \right], \quad (5.7)$$

where the term $\frac{\sigma^2}{\sigma_s^2}$ adapts the estimate to the noise level. From Eq. 5.7, it should be evident that in a noiseless system, the MMSE estimate reduces to the ZF estimate since the term $\frac{\sigma^2}{\sigma_s^2}$ will equal zero and the remaining term $\mathbf{H}(\mathbf{H}\mathbf{H}^H)^{-1}$ will equal the Moore-Penrose pseudoinverse which as explained previously coincides with the straightforward inverse in this case.

Previous work has shown that the ill-conditioning of the interference matrix significantly degrades the estimates generated using the ZF or MMSE detectors [129] [130]. Moreover, when the interference matrix is singular it becomes non-invertible, thus the ZF technique cannot be employed. A technique termed TSVD was therefore proposed by Isam to address this challenge [156].

The TSVD method was considered in the late 1980's by Hansen to address ill-posed linear LS problems ² as an alternative to standard regularisation techniques, such as the well-known Tikhonov regularisation [157]. TSVD improves the accuracy of the solution through a process of truncation, in other words by neglecting the smallest singular values which contribute the most to the ill-conditioning of the matrix under consideration.

The steps of the TSVD algorithm, as applied to the SEFDM detection problem, may be summarised as follows:

1. The interference matrix \mathbf{H} is decomposed using the standard SVD algorithm [144]

²An ill-posed problem is a problem where the matrix under consideration is ill-conditioned. As demonstrated in [129], the ill-conditioning of the interference matrix in SEFDM systems is due to the singular values of the matrix decaying gradually to zero.

into three matrices \mathbf{U} , $\mathbf{\Sigma}$, \mathbf{V} as

$$\mathbf{H} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H, \quad (5.8)$$

where the left and right singular matrices \mathbf{U} and \mathbf{V} are unitary whose columns are the eigenvectors of $\mathbf{H}\mathbf{H}^H$ and $\mathbf{H}^H\mathbf{H}$, respectively. The matrix $\mathbf{\Sigma}$ is a diagonal matrix containing the singular values of \mathbf{H} expressed as

$$\mathbf{\Sigma} = \text{diag}(\check{\sigma}_1, \check{\sigma}_2, \dots, \check{\sigma}_N), \quad (5.9)$$

where $\text{diag}(\cdot)$ denotes a matrix having diagonal form.

The diagonal elements $\check{\sigma}_i$ of $\mathbf{\Sigma}$ appear in descending order as

$$\check{\sigma}_1 \geq \check{\sigma}_2 \geq \dots \geq \check{\sigma}_\xi \geq \check{\sigma}_N, \quad (5.10)$$

where ξ denotes the truncation index with $\xi \leq N$.

2. The truncation index ξ defines the number of singular values that are considered in the final solution. This is achieved by replacing the smallest $N - \xi$ singular values with zeros. The choice of ξ determines the conditioning of the new matrix \mathbf{H}_{TSVD} and thus the accuracy of the acquired solution. Recall from Section 4.3 that the number of ‘small’ singular values is approximately equal to $(1 - \alpha)N$. Based on this fact, the optimum truncation index is given by [156]

$$\xi = \lceil \alpha N \rceil + 1, \quad (5.11)$$

where $\lceil \cdot \rceil$ denotes the ceiling function, meaning rounding up to the closest integer. Hence, Eq. 5.9 becomes

$$\mathbf{\Sigma}_\xi = \text{diag}(\check{\sigma}_1, \check{\sigma}_2, \dots, \check{\sigma}_\xi, 0, \dots, 0), \quad (5.12)$$

3. Subsequently, the inverse of \mathbf{H}_{TSVD} is derived as follows

$$\mathbf{H}_\xi = (\mathbf{U}\mathbf{\Sigma}_\xi\mathbf{V}^H)^{-1} = \mathbf{V}\mathbf{\Sigma}_\xi^{-1}\mathbf{U}^H = \sum_{i=1}^{\xi} \frac{\mathbf{v}_i\mathbf{u}_i^H}{\check{\sigma}_i}, \quad (5.13)$$

where \mathbf{v}_i and \mathbf{u}_i are the column eigenvectors of matrices \mathbf{V} and \mathbf{U} , respectively, and $\mathbf{\Sigma}_\xi^{-1} = \text{diag}(1/\check{\sigma}_1, 1/\check{\sigma}_2, \dots, 1/\check{\sigma}_\xi, 0, \dots, 0)$.

Consequently, it should be evident that the TSVD approach generates an approximate pseudoinverse \mathbf{H}_ξ of the interference matrix \mathbf{H} . The truncation of the small singular values renders \mathbf{H}_ξ less sensitive to perturbations yielding a solution with higher confidence levels. The amelioration offered by TSVD over the conventional inverse is illustrated in Fig. 5.4, which shows the amplitudes of the straightforward inverse \mathbf{C}^{-1} and those of the TSVD approximate pseudoinverse \mathbf{C}_ξ with respect to the sub-carriers correlation matrix \mathbf{C} for $N = 16$ and $\alpha = 0.8$. From Fig. 5.4, it is clear that the distribution and dynamic range of the values in $\mathbf{C}_{\xi_{m,n}}$ is far more uniform and ‘smoother’ compared to the values in $\mathbf{C}_{m,n}^{-1}$. This yields two benefits; first, the solution employing \mathbf{C}_ξ is more accurate, thus improving BER performance; second, the smaller dynamic range associated with \mathbf{C}_ξ relaxes bit precision requirements, thus favouring a more cost-effective implementation.

5.1.2 Iterative Detection

The ZF and TSVD detection methods introduced in Section 5.1.1 are classified as matrix-multiplication based techniques since they involve a direct multiplication of the statistics vector \mathbf{R} with the inverse or pseudoinverse of the interference matrix \mathbf{H} . An alternative method to direct multiplication is back-substitution, which is the fundamental principle underpinning iterative detection techniques.

Kanaras in [129] applied an Iterative Cancellation (IC) method to the projections matrix \mathbf{D} to generate symbol estimates. The form of \mathbf{D} depends on whether GS procedures or Löwdin orthonormalisation has been employed. In the former case, \mathbf{D} is

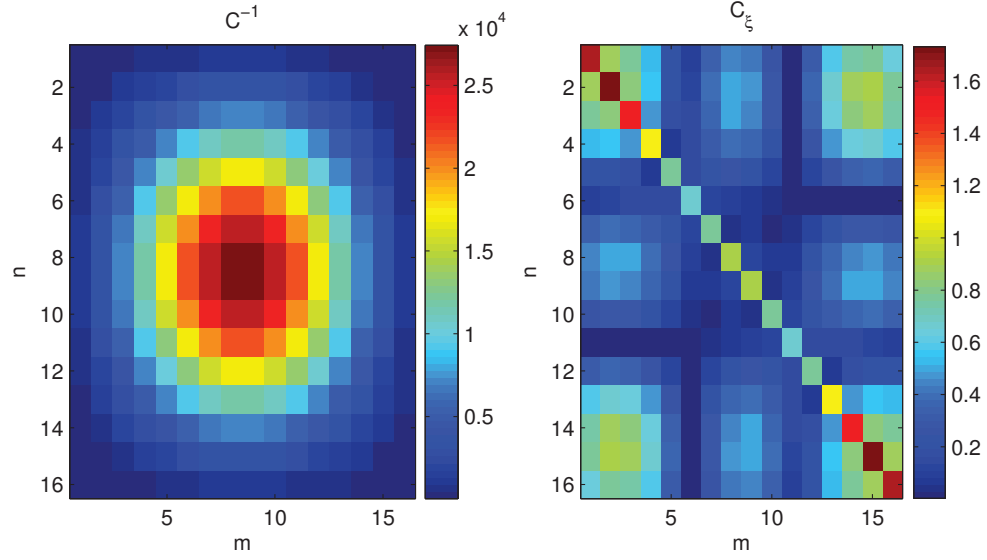


Figure 5.4: Amplitude of $\mathbf{C}_{m,n}^{-1}$ (left) and $\mathbf{C}_{\xi_{m,n}}$ (right) ($N = 16$, $\alpha = 0.8$).

upper triangular, hence IC can be directly applied. In the latter case, \mathbf{D} is Hermitian, therefore in order to apply the IC method, all elements lying below its main diagonal are forced to zero, in other words $d_{mn} = 0$ for $m > n$, where d_{mn} are the elements of \mathbf{D} with $m, n = 0, \dots, N - 1$.

Here, the IC technique³ described in [129] is applied to the sub-carriers correlation matrix \mathbf{C} . Recalling from Section 4.3, \mathbf{C} is Hermitian and Toeplitz, hence the upper triangle of \mathbf{C} has to be extracted with the lower triangle forced to zero. Having generated an upper triangular matrix, the symbol estimates can then be obtained via back-substitution. The steps of this process may be summarised as follows:

1. The form of \mathbf{C} is given by

$$\mathbf{C} = \begin{bmatrix} c_{\Re_{1,1}} + jc_{\Im_{1,1}} & c_{\Re_{1,2}} + jc_{\Im_{1,2}} & \cdots & c_{\Re_{1,N}} + jc_{\Im_{1,N}} \\ c_{\Re_{1,2}} - jc_{\Im_{1,2}} & c_{\Re_{1,1}} + jc_{\Im_{1,1}} & \cdots & c_{\Re_{2,N}} + jc_{\Im_{2,N}} \\ \vdots & \vdots & \ddots & \vdots \\ c_{\Re_{1,N}} - jc_{\Im_{1,N}} & c_{\Re_{1,N-1}} - jc_{\Im_{1,N-1}} & \cdots & c_{\Re_{1,1}} + jc_{\Im_{1,1}} \end{bmatrix}$$

³Note that this technique is different from the IDSD method proposed in recent work [135] which incorporates soft decision.

2. The highlighted elements of \mathbf{C} are then forced to zero yielding

$$\mathbf{C} = \begin{bmatrix} c_{\Re 1,1} + jc_{\Im 1,1} & c_{\Re 1,2} + jc_{\Im 1,2} & \cdots & c_{\Re 1,N} + jc_{\Im 1,N} \\ 0 & c_{\Re 1,1} + jc_{\Im 1,1} & \cdots & c_{\Re 2,N} + jc_{\Im 2,N} \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & 0 & c_{\Re 1,1} + jc_{\Im 1,1} \end{bmatrix}$$

3. The IC algorithm starts by initialising $\hat{\mathbf{s}}_N = \left\lfloor \frac{R_N}{c_{N,N}} \right\rfloor$ where $\hat{\mathbf{s}}_N$ is the estimate of the N^{th} transmit symbol, R_N is the N^{th} element of the statistics vector \mathbf{R} and $c_{N,N}$ is the N^{th} element of the sub-carriers correlation matrix \mathbf{C} . Hence, starting from $m = N$, the IC proceeds in a backwards fashion until $m = 1$ with the m^{th} symbol estimate given by

$$\hat{s}_m = \left\lfloor \frac{1}{c_{m,m}} \left(R_m - \sum_{n=m+1}^N c_{m,n} \hat{s}_n \right) \right\rfloor, \quad (5.14)$$

As will be illustrated by the simulation results presented in Section 5.2.2, the IC method results in an improved BER over ZF and TSVD. This is due to the fact that IC constrains the symbol estimate on each iteration, in other words it takes into account that each entry of \mathbf{s} pertains to a value from a finite set of constellation points \mathcal{M} . This is shown in Eq. 5.14 where each symbol estimate \hat{s}_m undergoes a slicing operation prior to being subtracted from the next estimate \hat{s}_{m-1} .

The choice of using linear or iterative detection is one which requires careful consideration as it is associated with complex trade-offs. The two key metrics are complexity and error performance, hence the aforementioned detection schemes are evaluated in terms of these metrics in the following sections.

5.1.3 Complexity Order and Implementation Aspects of Linear and Iterative Detection

SEFDM increases multiplexing gain with respect to an OFDM system by permitting the use of more sub-carriers in the default OFDM bandwidth. This gain, however,

comes at the expense of complex signal processing at the receiver which is required in order to separate the sub-carriers suffering from self-created ICI. A linear increase in the number of sub-carriers results in a more than linear increase in the complexity order of both the demodulation and detection stages.

A demodulator employing GS or Löwdin orthonormalisation has the same complexity order as a conventional analogue multi-carrier system equal to $\mathcal{O}(N^2)$. The matched filter technique addressed this issue by substituting these processes with DFT operations, thus allowing the use of the FFT algorithm reducing the complexity order to $\mathcal{O}(N \log_2 N)$.

The linear detection methods explored in Section 5.1.2 involve matrix inversion or decomposition. Analytical tools like MATLAB, generally apply Gaussian elimination to compute the inverse of a matrix, such as the computation required in the LS solution of the ZF estimate. The TSVD detector uses the standard SVD matrix decomposition method to generate an approximate inverse. In both cases, the order of complexity is assumed to be cubic $\mathcal{O}(N^3)$ [129] [61]. The inverted matrix is then multiplied by the statistics vector \mathbf{R} , which implies that the total number of multiplications amounts to N^2 . The slicing operator acts as a simple threshold detector, therefore its complexity is considered to be negligible compared to the complexity of the prerequisite matrix operations.

Conversely, IC techniques operate on upper triangular matrices, where the total number of multiplications equals $\frac{N}{2}(N+1) \approx \frac{N^2}{2}$, which is less than the number of multiplications required for the ZF or TSVD techniques. IC requires the inversion of only the diagonal elements of the matrix.

At first, it may appear that IC is the preferred method over linear detection. Both linear and iterative schemes present different challenges with varying trade-offs. The choice depends on the specific technology adopted to implement these algorithms in practice. FPGAs offer significant performance gains over conventional DSPs provided that their parallel processing capabilities are leveraged. The benefits and drawbacks of

using linear or iterative detection are as follows:

- **Linear detection:** The fundamental difficulty in this case lies in the inversion of the sub-carriers correlation matrix \mathbf{C} which has a large dimension equal to $N \times N$ and is also complex-valued which instantly doubles the dimension of the problem to $2N \times N$. The computational complexity associated with the inversion or decomposition of matrices is often prohibitive in wireless systems [80]. \mathbf{C} in SEFDM systems is deterministic and has constant values for a given configuration of N and α . Thereby, the elements of \mathbf{C}^{-1} only need to be generated once, a process which can take place offline, and subsequently stored in block memories. The computation of the symbol estimates is then a straightforward complex-valued matrix multiplication between \mathbf{R} and \mathbf{C}^{-1} . The default number of words required for the storage of the elements pertaining to \mathbf{C}^{-1} is N^2 . Matrix multiplication can be implemented efficiently through the concurrent execution of multiple multiply-accumulate operations using FPGAs. The main pitfall of this configuration is that it requires a high bit precision to represent the input values and intermediate results adequately without the risk of overflow.
- **Iterative detection:** IC alleviates the need for matrix inversion which means that the results suffer less from round-off errors. The performance of IC largely depends on the correctness of the first symbol decoded since any errors during the initial stages of the algorithm's execution will propagate through the iterative process and could potentially lead to severe BER degradation. While IC relaxes bit precision requirements, thanks to the slicing operation applied to each symbol estimate on every iteration, it does require a cascaded architecture which hinders parallel processing. One possible solution to this issue is to decompose the entire iteration process into a single step. In FPGA terms, this decomposition would necessitate combinatorial logic, which is generally avoided as it increases the load on the clock severely limiting the maximum achievable clock frequency.

Based on the above, in practice, linear detection algorithms can be implemented in a more straightforward manner compared to iterative techniques. Assuming that the preprocessing of the SEFDM interference matrix which requires inversion can be executed efficiently offline, the FPGA realisation of ZF and/or TSVD can be achieved easily using several Multiplier-Accumulator Units (MAUs) configured in parallel. Even though the performance of iterative techniques may be improved by introducing ordering schemes into the original (unordered) algorithm, this ordering will incur additional processing overhead complicating the design of receiver architectures.

The work and literature on SEFDM detection is abundant from a theoretical perspective, albeit less has been accomplished to date in terms of practical application. The final hurdle remains in rendering the developed algorithms suitable for hardware implementation. To this end, the first task is to optimise the algorithms at an algorithmic level to yield tangible computational complexity. This is particularly true for the more sophisticated algorithms, such as FSD. The second task involves optimisation at the circuit level, where the aim is to strike a balance between the available resources of the device in use, the maximum achievable clock frequency and the minimum pipelined latency. The ultimate goal is to minimise area and/or resource utilisation while concurrently meeting system requirements in terms of throughput and error performance.

5.1.4 Maximum Likelihood and Sphere Decoding

As a preface to Chapter 7, the optimum and quasi-optimum schemes considered to address the detection challenge in SEFDM are discussed briefly. It is well known that ML is the optimum detector for achieving the theoretical error performance. The detection problem in this case may be expressed as follows

$$\hat{\mathbf{s}}_{ML} = \arg \min_{\mathbf{s} \in \mathcal{M}^N} \|\mathbf{R} - \mathbf{H}\mathbf{s}\|^2, \quad (5.15)$$

which translates into finding the vector symbol $\hat{\mathbf{s}}$ from all possible transmitted vector symbols $\mathbf{s} \in \mathcal{M}^N$ that minimises the Euclidean distance, in other words the vector

symbol that has most likely been transmitted. Assuming that complex Gaussian noise has been added to each symbol and that all transmitted vector symbols have equal probability of occurring, the ML algorithm searches over the entire set $\mathbf{s} \in \mathcal{M}^N$ to find $\hat{\mathbf{s}}$. Hence, it is evident that this hinders a practical implementation of ML.

The ill-conditioning of the interference matrix in SEFDM systems is analogous to the ill-conditioning of the channel matrix in MIMO systems. Inspired by the techniques adopted in MIMO, SD was examined thoroughly [129] as a more viable solution for recovering SEFDM signals successfully with tolerable error penalties since it reduces the ML solution to a problem of polynomial complexity. The SD algorithm applied to the SEFDM detection challenge [129] follows similar principles to the one described by Hassibi and Vikalo [158], which in turn is based on the concept of *lattice decoding* as described by Viterbo in the early 1990's [159]. SD solves the same problem as ML, yet in this case, the search plane is limited to an N-dimensional hypersphere, whereby only the vector symbols that lie within this hypersphere are considered as possible solutions. Thereby, Eq. 5.15 becomes

$$\hat{\mathbf{s}}_{SD} = \arg \min_{\mathbf{s} \in \mathcal{M}^N} \|\mathbf{R} - \mathbf{H}\mathbf{s}\|^2 \leq g, \quad (5.16)$$

where g is a scalar denoting the radius of the sphere which is centred around the statistics vector \mathbf{R} . SD is classified as a tree-search algorithm because the constellation points refer to the nodes of the tree and the algorithm is executed by traversing the levels and branches of the tree until a complete path is found.

The conventional SD algorithm has a variable computational complexity which changes according to the noise in the channel and the properties of the interference matrix. Consequently, the complexity could potentially approach that of ML, thus rendering the SD algorithm inapt for hardware implementation. For this reason, variants of the SD algorithm which fix its complexity have been proposed with the most popular ones being the K-best decoder as described by Wong [160] and the Fixed Complex Sphere Decoder (FCSD) as described by Barbero [161] with application in MIMO sys-

tems. The FSD presented in Chapter 7 and described by Isam [130] is based on similar principles, however, in contrast to the FCSD, i) it uses RVD, and ii) the number of nodes searched is common for all tree levels, contrary to the FCSD where the number is variable.

In conclusion, this discussion showed that research in MIMO systems is highly relevant to SEFDM systems. The processing of the channel matrix in MIMO is of equal significance [61] to the processing of the interference matrix in SEFDM. The principal difference, alas one of great importance, is that the dimension of the system in MIMO, proportional to the number of input and output antennas, is typically much smaller compared to the dimension of the system in SEFDM, the latter expressed in terms of the number of sub-carriers employed. Recalling the fact that a linear increase in system size results in a higher than linear increase in receiver complexity, it should come as no surprise that the detection challenge in SEFDM is more difficult to harness.

5.2 SEFDM Performance Gains through Oversampling

In this section, the gains that may be obtained through oversampling are explored. Standard communication systems employ oversampling in order to acquire an accurate picture of the underlying waveform, in other words the sampling rate is ordinarily much higher than twice the highest frequency component in order to catch all signal variations [91]. Classic oversampling techniques are achieved via interpolation of the time domain signal, which usually entails two steps; first, the signal is upsampled, which involves adding zeros between each sample; second, the signal is filtered according to a specific interpolation technique, such as linear and *spline* methods. Here, oversampling is carried out by zero padding the sub-carriers matrix Φ and its conjugate Φ^H , or equivalently the IDFT and DFT blocks at the transmitting and receiving ends. It is therefore assumed that the interference matrix is oversampled by the same factor $\rho > 1$.

Table 5.1 shows the effect of oversampling on the projections matrix \mathbf{D} . As explained in Section 4.3, the eigenvalue ratio $\frac{\lambda_{max}(\mathbf{D})}{\lambda_{min}(\mathbf{D})}$ was used as a measure of the condi-

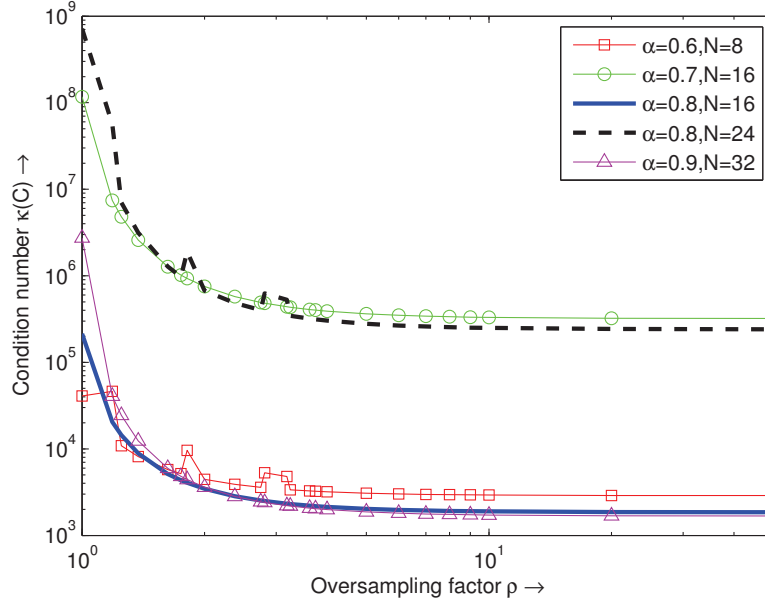


Figure 5.5: Condition number $\kappa(C)$ for increasing oversampling factors and different values of N and α .

tioning of \mathbf{D} in [129]. The values shown for $\rho = 1$ match the values presented⁴ in Table 5.1 in [129]. From Table 5.1 in this work, it is clear that oversampling can significantly improve the conditioning of \mathbf{D} . This improvement will be reflected and quantified in Section 5.2.2, which compares the error performance of SEFDM detection techniques for different oversampling values. Table 5.2 corroborates the results presented in Table 5.1 by comparing the condition numbers for the sub-carriers correlation matrix \mathbf{C} in the absence and presence of oversampling.

Fig. 5.5 depicts the value of the condition number for the sub-carriers correlation matrix \mathbf{C} according to the value of ρ for a varying number of sub-carriers N and bandwidth compression α . Without loss of generality, Fig. 5.5 shows that for $\rho > 2$ the performance gain is marginal. Consequently, from this point onwards, only oversampling factors with $\rho \leq 2$ shall be considered.

⁴ \mathbf{D} in this work corresponds to \mathbf{M} in [129].

Table 5.1: Ratio $\frac{\lambda_{\max}(\mathbf{D})}{\lambda_{\min}(\mathbf{D})}$ of the projections matrix \mathbf{D} for different N , α and ρ .

N, ρ	α		0.7	0.75	0.8	0.85	0.9	0.95	1
	$\rho = 1$	$\rho = 2$							
$N = 8$			4.25	2.93	2.12	1.60	1.27	1.07	1
			2.23	1.71	1.39	1.19	1.07	1.02	1
$N = 16$			33.90	14.71	6.97	3.60	2.03	1.28	1
			6.28	3.52	2.21	1.55	1.20	1.04	1
$N = 24$			290.11	78.88	24.46	8.59	3.42	1.58	1
			17.83	7.27	3.52	2	1.35	1.07	1
$N = 32$			2554.80	435.30	88.20	21.10	5.90	2	1
			50.67	15	5.62	2.62	1.52	1.10	1
$N = 40$			22860	2440	323	52	10	3	1
			144	31.12	8.98	3.42	1.72	1.14	1

Table 5.2: Condition number $\kappa(\mathbf{C})$ of matrix \mathbf{C} for different N , α and ρ .

N, ρ	α		0.7	0.75	0.8	0.85	0.9	0.95	1
	$\rho = 1$	$\rho = 2$							
$N = 8$	1918	281	439.26	79.48	102.16	25.91	7.29	2.75	1
					24.53	8.52	3.46	1.70	1
$N = 16$	$1.17 \cdot 10^8$		5090500		214020	8377	293.76	11.11	1
	753080		48794		3465.7	270.88	25.01	3.49	1
$N = 24$	$1.05 \cdot 10^{13}$		$8.93 \cdot 10^{10}$		$7.18 \cdot 10^8$	4895200	25329	84.17	1
	$2.52 \cdot 10^9$		38734000		671150	12858	277.75	8.75	1
$N = 32$	$1.18 \cdot 10^{16}$		$2.32 \cdot 10^{15}$		$2.8 \cdot 10^{12}$	$3.45 \cdot 10^9$	2733500	1033.4	1
	$9.09 \cdot 10^{12}$		$3.33 \cdot 10^{10}$		$1.42 \cdot 10^8$	677460	3615.3	25.67	1
$N = 40$	$1.78 \cdot 10^{16}$		$1.06 \cdot 10^{16}$		$7.34 \cdot 10^{15}$	$2.63 \cdot 10^{12}$	$3.27 \cdot 10^8$	12965	1
	$1.52 \cdot 10^{16}$		$2.95 \cdot 10^{13}$		$3.11 \cdot 10^{10}$	37377000	50068	83.69	1

5.2.1 Noise Model and Simulation Methodology

The simulation methodology adopted throughout this thesis has the following characteristics:

- The source data is generated using a uniform random number generator, whereby the probability of sending a zero is equal to the probability of sending a one.
- The bits are mapped to symbols using either BPSK or 4-QAM unless otherwise stated. In the case of 4-QAM, it is assumed that Gray coding is employed, thus ensuring that only one bit changes between adjacent symbols. The theoretical BER for 4-QAM is identical to the theoretical BER for BPSK since 4-QAM may be regarded as being equivalent to two independent BPSK signals transmitted on the In-Phase (I) and Quadrature (Q) channels.
- The number of bits processed is approximately equal to $\frac{10}{\text{Desired BER}}$.
- The channel is linear and discrete-time.
- The channel is time-invariant.
- The only source of error is due to AWGN, hence the errors are truly random with nil probability of burst errors occurring.
- The BER simulations compare and contrast the results in terms of $\frac{E_b}{N_0}$ penalty, in other words the additional power required to achieve the same BER.
- It is assumed that the receiver has perfect knowledge of the interference matrix \mathbf{H} at all times under all conditions and that decisions yielding symbol estimates are made strictly based on the received amplitude of the noisy signal.

For the cases where fixed-point effects are incorporated in the analysis, the following assumptions are made in addition:

- The arithmetic operations take into account practical sampling factors, limited dynamic range leading to quantisation noise, as well as round-off errors that occur during the computations.
- The error floor is taken to be the sum of the AWGN and the quantisation noise.

The theoretical BER probability for BPSK is given by [162]

$$P_e = \frac{1}{2} \text{erfc} \left(\sqrt{\frac{E_b}{N_0}} \right), \quad (5.17)$$

where P_e denotes the probability of error and erfc is the complementary error function. Taking into account that P_e is inverse proportional to the Euclidean distance between symbols, it should be evident that BPSK gives the lowest BER for any signal since the symbols are antipodal. For the special case where 4-QAM is employed with Gray coding, the probability of error is given by the same formula specified for BPSK in Eq. 5.17. For the general case, however, where higher order modulation schemes are considered, the higher density of the points in the constellation will render the system more susceptible to errors. In SEFDM, the self-created interference also means that the performance of BPSK-SEFDM is superior to 4-QAM-SEFDM in terms of BER.

Recalling from Section 4.2.1, the discrete-time SEFDM signal at the modulator's output when no oversampling is used is expressed as $x(m) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} s_n e^{j2\pi m n/N}$, therefore the corresponding average signal power is given by

$$\mathcal{S}_{avg} = \frac{\|\mathbf{x}\|^2}{N}, \quad (5.18)$$

For a conventional digital system, the average signal power is defined as $\mathcal{S}_{avg} = E_b R_b$, where R_b is the bit rate of the system. For a multi-carrier system, such as SEFDM, the transmission rate per sub-carrier is equal to $R = \frac{R_{ds}}{N} = \frac{R_b}{N \log_2 M}$. Solving for E_b yields

$$E_b = \frac{\mathcal{S}_{avg}}{R N \log_2 M} = \frac{\mathcal{S}_{avg} T}{N \log_2 M}. \quad (5.19)$$

Subsequently, the noise spectral density is given by

$$\mathcal{N}_0 = \frac{E_b}{10^{\frac{E_b}{\mathcal{N}_0}/10}} = \frac{\mathcal{S}_{avg}T}{N \log_2 M 10^{\frac{E_b}{\mathcal{N}_0}/10}}, \quad (5.20)$$

where the ratio $\frac{E_b}{\mathcal{N}_0}$ is expressed in decibels. \mathcal{N}_0 is the one-sided Power Spectral Density (PSD) of the noise. It has been shown [129] that the received statistics in SEFDM has a Gaussian distribution with zero mean and variance σ^2 . The noise variance is computed by multiplying the one-sided PSD by the noise bandwidth, the latter being equal to half the sampling frequency for a real signal [87]. For a complex signal, each of the I and Q components are bandlimited to half the sampling frequency using two separate transmit filters. Based on the above, the noise variance may be expressed as

$$\sigma^2 = \mathcal{N}_0 \frac{f_s}{2} = \frac{\mathcal{S}_{avg}T}{N \log_2 M 10^{\frac{E_b}{\mathcal{N}_0}/10}} \frac{f_s}{2} = \frac{\mathcal{S}_{avg}NT_s}{N \log_2 M 10^{\frac{E_b}{\mathcal{N}_0}/10}} \frac{1}{2T_s} = \frac{\mathcal{S}_{avg}}{2 \log_2 M 10^{\frac{E_b}{\mathcal{N}_0}/10}}. \quad (5.21)$$

5.2.2 Numerical Results

The numerical results presented in this section are empirical and aim to corroborate the trade-offs identified in earlier parts of this work with regard to the performance gains offered by different receiver schemes and system parameters. These results also serve as a baseline reference for the design decisions and choices made during the hardware implementation of selected detection algorithms. A theoretical explanation of the observations made is beyond the scope of this thesis. The reader is referred to [129] and [130] for an authoritative analytical account of the fundamental principles supporting the forthcoming arguments and conclusions. All the results are averaged over 10,000 randomly generated SEFDM symbols unless otherwise stated.

While the work in this thesis focuses on ZF and TSVD techniques, the remaining detection methods are provided for comparison purposes. Hence, the detection schemes considered in this section include the following:

- **Matched Filtering (MF):** The symbol estimates are given by $\hat{\mathbf{s}}_{MF} = \lfloor \mathbf{R} \rfloor =$

$[\Phi^H \mathbf{r}]$ where \mathbf{r} is a vector representing discrete time samples of the received signal having propagated through an AWGN channel.

- **Matched Filtering followed by Zero Forcing (MF-ZF):** The symbol estimates are given by $\hat{\mathbf{s}}_{ZF} = [\mathbf{C}^{-1} \mathbf{R}]$.
- **Demodulation using Iterative Modified Gram Schmidt (IMGS) processes [129] followed by Zero Forcing (IMGS-ZF):** The symbol estimates are given by $\hat{\mathbf{s}}_{ZF_{IMGS}} = [\mathbf{D}^{-1} \mathbf{R}]$.
- **Matched Filtering followed by Truncated Singular Value Decomposition (MF-TSVD):** The symbol estimates are given by $\hat{\mathbf{s}}_{TSVD} = [\mathbf{C}_\xi \mathbf{R}]$.
- **Matched Filtering followed by Iterative Cancellation (MF-IC):** The symbol estimates are given by $\hat{\mathbf{s}}_{IC} = \hat{s}_m$ with $m = 1, \dots, N$ and

$$\hat{s}_m = \left[\frac{1}{c_{m,m}} \left(R_m - \sum_{n=m+1}^N c_{m,n} \hat{s}_n \right) \right].$$
- **Demodulation using Iterative Modified Gram Schmidt processes followed by Iterative Cancellation (IMGS-IC):** The symbol estimates are given by $\hat{\mathbf{s}}_{IC_{IMGS}} = \hat{s}_m$ with $m = 1, \dots, N$ and $\hat{s}_m = \left[\frac{1}{d_{m,m}} \left(R_m - \sum_{n=m+1}^N d_{m,n} \hat{s}_n \right) \right]$.
- **Demodulation using Löwdin processes [129] followed by Iterative Cancellation (Lowdin-IC):** The symbol estimates are given by $\hat{\mathbf{s}}_{IC_{Lowdin}} = \hat{s}_m$ where $\hat{s}_m = \left[\frac{1}{l_{m,m}} \left(R_m - \sum_{n=m+1}^N l_{m,n} \hat{s}_n \right) \right]$, with $m = 1, \dots, N$ and where $l_{m,n}$ denotes the elements of the Hermitian matrix generated using the Löwdin orthonormalisation process.
- **Matched Filtering followed by Minimum Mean Squared Error (MF-MMSE):** The symbol estimates are given by $\hat{\mathbf{s}}_{MMSE} = \left[\mathbf{C} \left(\mathbf{C} \mathbf{C}^H + \frac{\sigma^2}{\sigma_s^2} \mathbf{I} \right)^{-1} \mathbf{R} \right]$.

Fig. 5.6 shows that for BPSK, the IC schemes using MF or Löwdin orthonormalisation offer the best and commensurate error performance. For SNR regimes where $\frac{E_b}{N_0} \geq 14\text{dB}$, the IC method employing IMGS approaches the performance of TSVD and

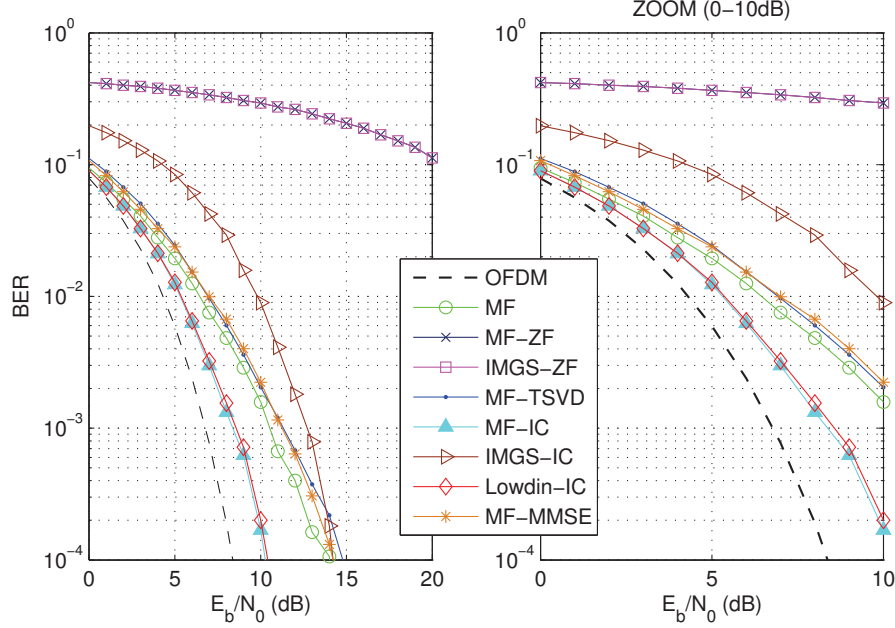


Figure 5.6: BER of different detection schemes for BPSK with $N = 16$, $\alpha = 0.8$ and $\rho = 2$.

MMSE. A significant observation from Fig. 5.6 is that MF on its own can provide the same if not better BER performance than both TSVD and MMSE. This is an important result as it allows the latter two detection stages illustrated in Fig. 5.1 to be completely bypassed, thus converging to the architecture and subsequent complexity of a conventional OFDM demodulator.

Fig. 5.7 illustrates the results when 4-QAM is employed instead of BPSK. Here, the varying performance for different SNR regimes is more prominent. For $\frac{E_b}{N_0} \leq 10dB$, the best performance is offered by MF-IC and Löwdin-IC followed by TSVD, MF and MMSE. For $\frac{E_b}{N_0} > 11dB$, IMGS-IC outperforms all other techniques.

Figs. 5.6 and 5.7 demonstrate that the choice of a suitable detection scheme is not trivial and is largely dependent upon the system requirements of the application under consideration. The following paragraphs will show that different system parameter settings, in terms of the number of sub-carriers, the bandwidth compression level and the oversampling factor, may give rise to diverse trade-offs in system performance, in

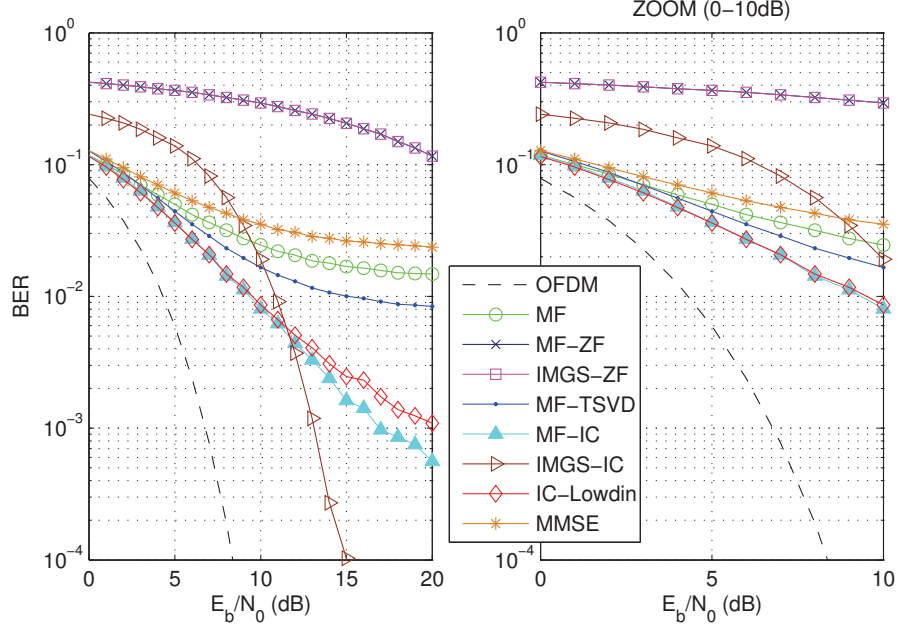


Figure 5.7: BER of different detection schemes for 4-QAM with $N = 16$, $\alpha = 0.8$ and $\rho = 2$.

terms of BER, bandwidth savings and throughput. For example, in Figs. 5.6 and 5.7, the oversampling factor was set to $\rho = 2$. This setting, however, translates to a 50% reduction in effective throughput, which is very undesirable as it constitutes a significant penalty in modern communication systems demanding ever-increasing data rates.

Fig. 5.8 illustrates the BER performance when only MF is employed. This configuration actually corresponds to FOFDM which is a special case of SEFDM with $\alpha = 0.5$. When no oversampling is used ($\rho = 1$), the performance improves as the number of sub-carriers is increased and reaches the theoretical OFDM performance when $N \geq 256$. This can be attributed to the fact that a larger number of sub-carriers acts as ‘virtual’ oversampling for the lower sub-carrier frequencies in the multiplexed time domain SEFDM signal. This concept resembles the increase in spectral efficiency proportional to the increase in the number of sub-carriers explored in Chapter 4.

To further support this argument, Fig. 5.9 depicts the BER of BPSK-SEFDM

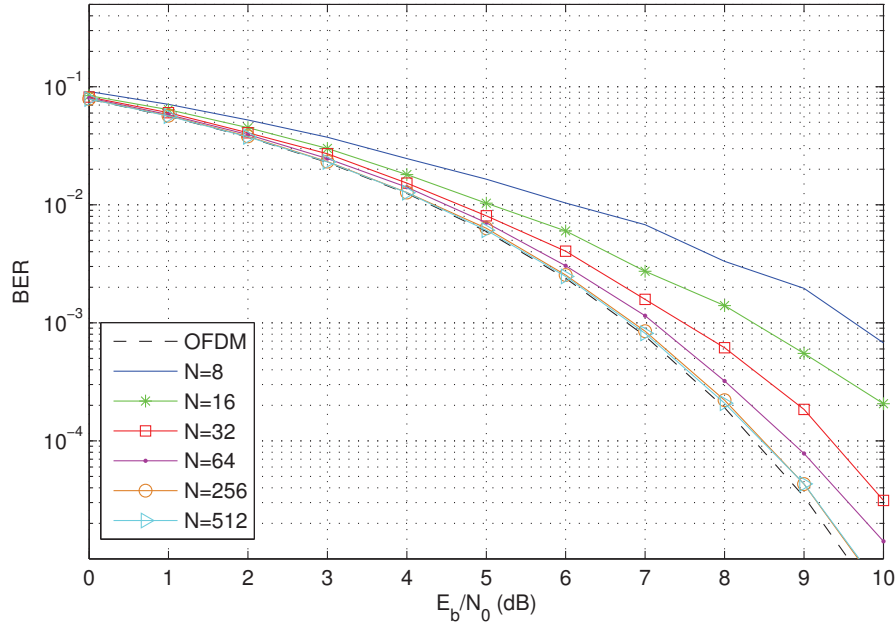


Figure 5.8: BER of matched filtering for BPSK with varying N ($\alpha = 0.5$ and $\rho = 1$).

in proportion to the oversampling factor for a different number of sub-carriers and bandwidth compression levels⁵. From Fig. 5.9, a number of important observations can be made. First, if employing BPSK then only FOFDM is capable of achieving the theoretical BER. In the case of FOFDM, if a small number of sub-carriers is employed ($N = 8$) then oversampling can significantly improve error performance. Conversely, oversampling does not offer any gain for a large number of sub-carriers ($N = 128$). If FOFDM is not employed, in other words $\alpha \neq 0.5$, then an oxymoron occurs since a small number of sub-carriers actually yields better error performance than a large number of sub-carriers. In this case ($\alpha \neq 0.5$), oversampling offers a marginal improvement in performance for $\rho > 2$, especially for decreasing values of α .

Having evaluated BPSK, the attention is now shifted to 4-QAM. Fig. 5.10 shows the BER performance of the three best detection schemes identified in Fig. 5.7 depending on the SNR regime under consideration, namely MF-TSVD, IMGS-IC and MF-IC. The

⁵For the results where E_b/N_0 has been set to a fixed value, the number was chosen to be $8dB$ given that this level of power is required to achieve an error rate of 10^{-4} as illustrated in Fig. 4.3.

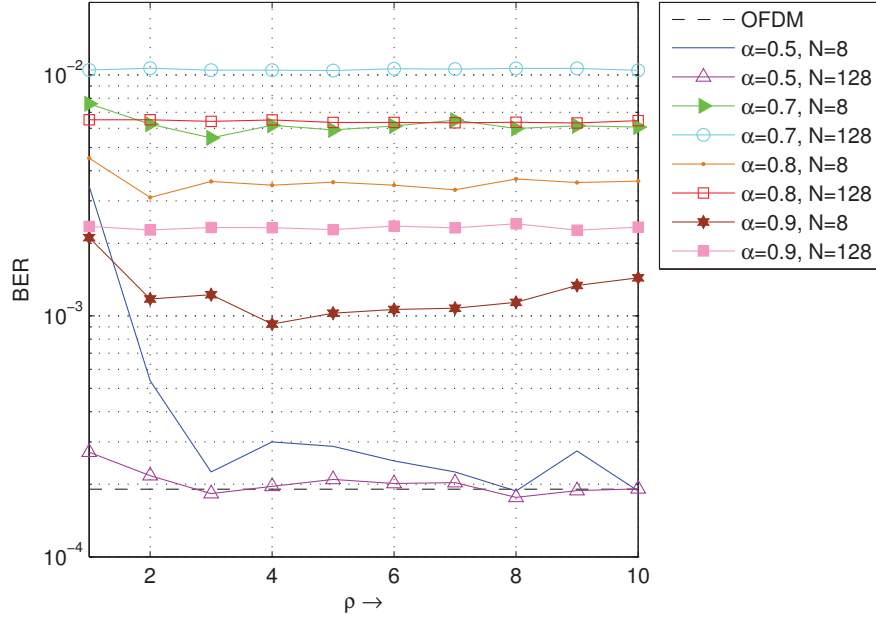


Figure 5.9: BER of matched filtering for BPSK with increasing ρ ($\frac{E_b}{N_0} = 8dB$).

difference in Fig. 5.10 is that the performance is evaluated for different oversampling factors. From Fig. 5.10, it is evident that when $\rho < 1$, the error performance is very poor for all three schemes. This would correspond to the case where the number of time samples Q is less than the number of sub-carriers N with the aim of increasing the effective data rate with respect to an OFDM system. Fig. 5.10 also shows that doubling the sampling rate, in other words $\rho = 2$, significantly improves the performance of MF-TSVD and IMGS-IC, especially for higher SNR regimes, whereas the gain is small for MF-IC.

Fig. 5.10 illustrates that IMGS-IC can outperform MF-TSVD, albeit the number of sub-carriers is small with $N = 8$. From Fig. 5.11, it is obvious that the performance of IMGS-IC degrades rapidly with an increase in the number of sub-carriers for low SNR regimes ($\frac{E_b}{N_0} = 8dB$), and for $N > 12$ it has inferior performance even to MF. Fig. 5.11 also shows that the oversampling gain is reduced as N is increased, which is in accordance with the results obtained for BPSK depicted in Fig. 5.9. Setting $\rho = 2$

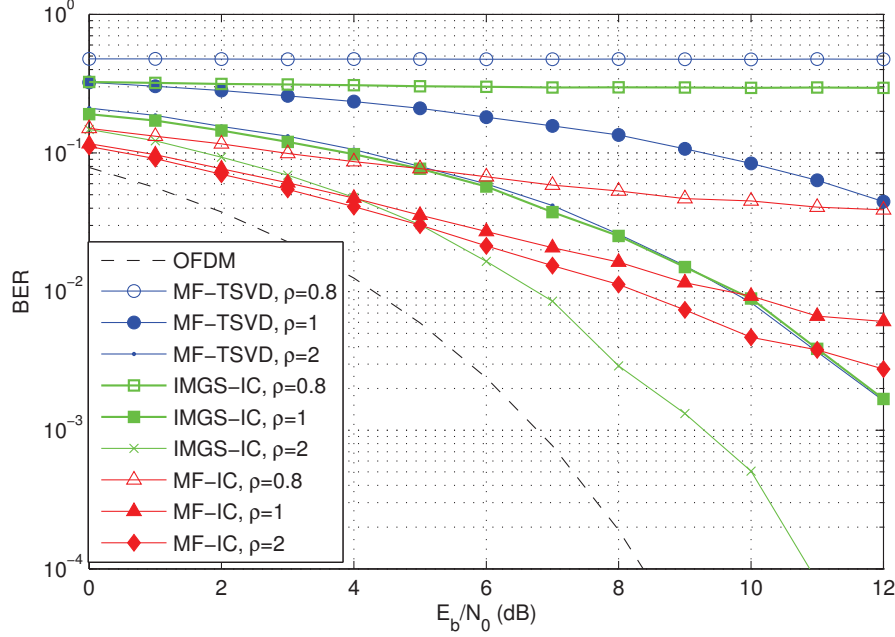


Figure 5.10: BER of different detection schemes for varying ρ ($N = 8$, $\alpha = 0.8$).

ameliorates the performance of MF-TSVD for $N \leq 12$ and overall results in a ‘less fluctuating’ performance for this particular detection scheme.

Fig. 5.12 shows the performance of the same aforementioned detectors for varying α when a small number of sub-carriers is employed ($N = 8$). From Fig. 5.12, it is evident that oversampling with $\rho = 2$ improves the BER performance of IMGS-IC in all cases, that of MF-TSVD for $\alpha \geq 0.65$, and that of MF and MF-IC for $\alpha \geq 0.75$. Furthermore, for $\alpha \geq 0.75$, IMGS-IC offers the best error performance while for $\alpha < 0.75$, MF-TSVD and MF-IC outperform the remaining detection methods.

The simulation was repeated for $N = 16$ with the results depicted in Fig. 5.13. From Fig. 5.13, it is evident that IMGS-IC rapidly loses its competitive advantage for $\alpha < 0.9$. Furthermore, for $\alpha \leq 0.85$, MF-TSVD approaches the performance of MF-IC, which in turn offers the best performance amongst all detectors for $\alpha \leq 0.85$.

Figs. 5.12 and 5.13 demonstrate that the performance of MF-TSVD is inferior to the alternative detection schemes for $\alpha \geq 0.8$ and $\alpha \geq 0.9$, respectively. This may be

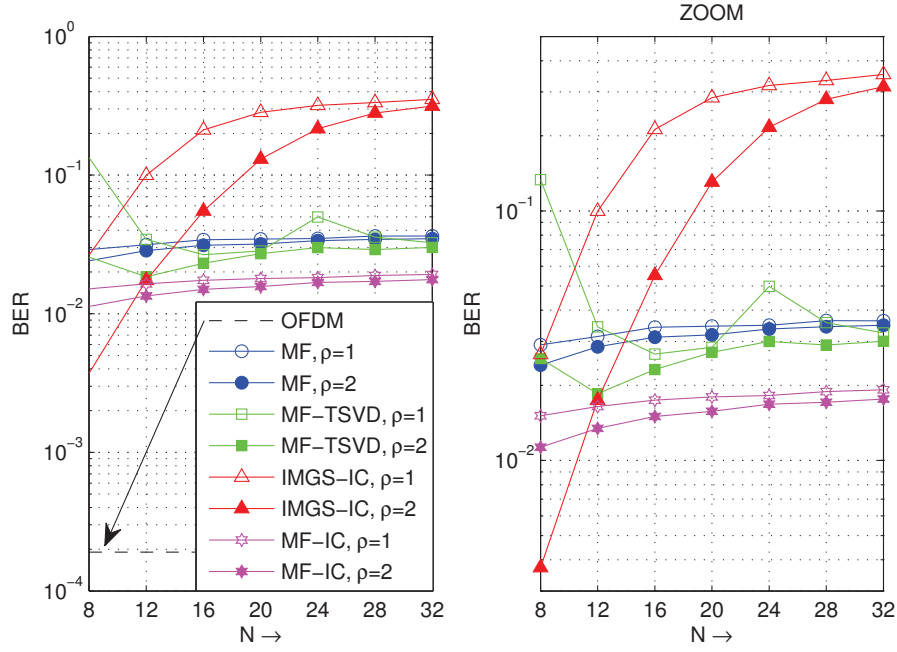


Figure 5.11: BER of different detection schemes for varying N ($\alpha = 0.8$, $\frac{E_b}{N_0} = 8dB$).

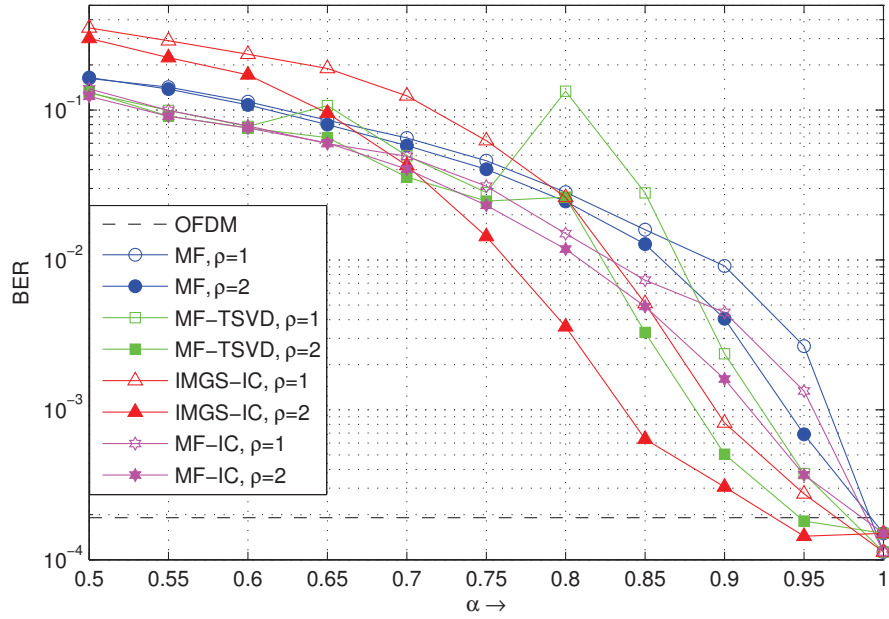


Figure 5.12: BER of different detection schemes for varying α with $N = 8$ ($\frac{E_b}{N_0} = 8dB$).

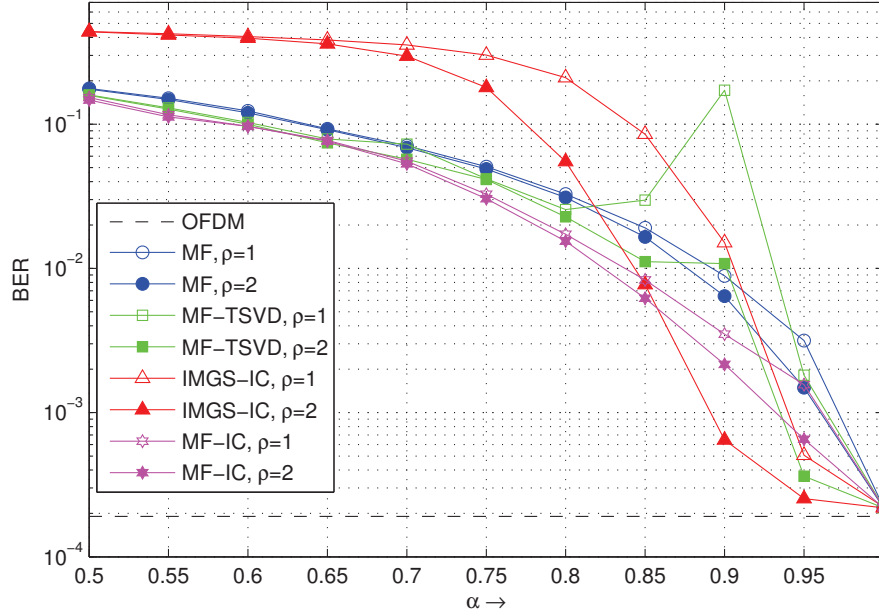


Figure 5.13: BER of different detection schemes for varying α with $N = 16$ ($\frac{E_b}{N_0} = 8dB$).

attributed to the fact that the truncation of the singular values in the TSVD matrix is dictated by both N and α which for certain cases means that the TSVD \mathbf{C}_ξ will be commensurate with the straightforward inverse of the sub-carriers correlation matrix \mathbf{C}^{-1} , the latter used during the computation of the ZF solution. For example, for $N = 8$ and $\alpha = 0.8$, $\xi = \lceil 8 \cdot 0.8 \rceil + 1 = 8$, and for $N = 16$ and $\alpha = 0.9$, $\xi = \lceil 16 \cdot 0.9 \rceil + 1 = 16$. Consequently, in such cases as these two examples, no truncation occurs and therefore the error performance of MF-TSVD is identical to the error performance of MF-ZF, which as shown in [129] is inferior to IC for $N = 8$ and $N = 16$.

The results in this section also demonstrated that the performance of MF-IC is similar to the performance of Lowdin-IC. Closer examination reveals that the properties of the projections matrix generated using Löwdin orthonormalisation are similar to the properties of the sub-carriers correlation matrix \mathbf{C} in that the amount of signal energy transferred to the interfering terms is less in Löwdin compared to IMGS. This point is exemplified by Kanaras (Fig. 3.10 in [129]) who shows that the size of the norm

of the diagonal elements decreases linearly in Löwdin and exponentially in IMGS with a reduction in the value of α , hence explaining the cause of the rapid performance degradation of IMGS-IC shown in Figs. 5.12 and 5.13.

To conclude this section, it is clear that oversampling can improve error performance, particularly for $N \leq 16$ and $\alpha \geq 0.8$, at the expense of reduction in effective throughput. The reasons for pursuing the hardware implementation of TSVD over alternative, in some cases better, detection schemes include the following:

- At the time of development, most of the theoretical work had focused on linear detection schemes, such as ZF, MMSE and TSVD, whereas studies on iterative techniques were minimal.
- After careful consideration, it was decided that the implementation of TSVD with the aid of block memories would be more straightforward compared to the architecture of iterative techniques. In addition, using block memories had the potential to leverage the parallel processing capabilities of FPGAs while freeing up valuable logic resources which would otherwise be required in IC to control the feedback loops.
- The intended hardware design of the TSVD detector would enable the receiver to switch instantly to using ZF, the latter being the optimum detector in a noiseless environment, without altering the FPGA architecture and with the aid of a simple and virtually costless Multiplexer (MUX).
- Finally, as illustrated in Fig. 5.14, the hybrid TSVD-FSD detector, which is examined in detail in Chapter 7, was found to outperform all other hybrid and non-hybrid detectors, thereby rendering TSVD the preferred choice for hardware realisation.

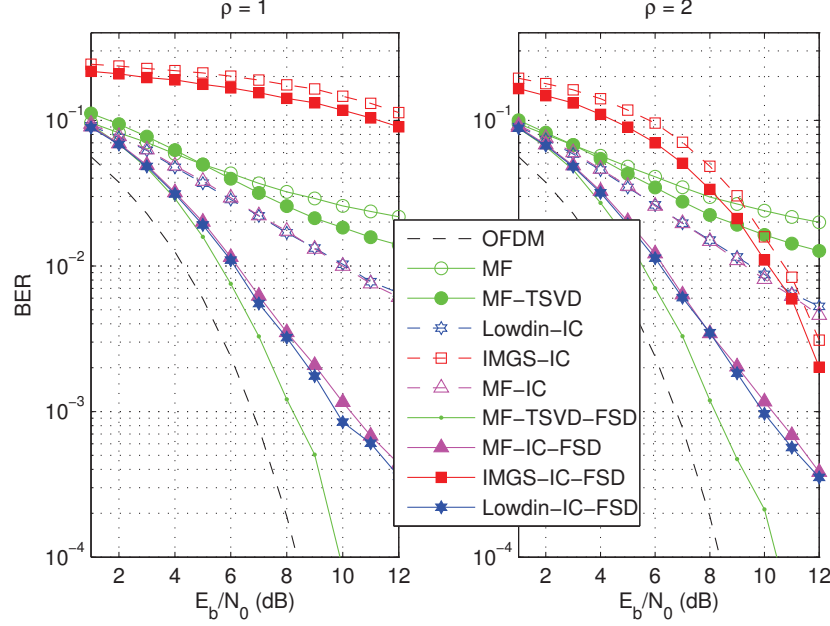


Figure 5.14: BER of hybrid and non-hybrid detection schemes for $\rho = 1$ (left sub-plot) and $\rho = 2$ (right sub-plot) ($N = 16$, $\alpha = 0.8$, FSD tree width $W = 16$).

5.3 Influence of Finite Word Lengths on the Performance of SEFDM Systems

Section 5.2 looked at the effects of oversampling on the performance of SEFDM systems. In this section, the impact of finite arithmetic precision on SEFDM performance is studied. Algorithm designers prefer floating-point representation over fixed-point since the former is capable of supporting a wider range of real numbers while obviating scaling operations which are normally required in fixed-point arithmetic. Many hardware details are often hidden from algorithm designers with the most crucial one being the limited arithmetic precision of real-time devices, such as DSPs and FPGAs, which is fixed at manufacturing (DSPs) or design (FPGAs) time. In addition, floating-point arithmetic is far more expensive in terms of area and power compared to fixed-point arithmetic [163].

To this end, a number of key issues are examined in this section including the upper

number precision limit offered by MATLAB, the conversion of floating-point values to fixed-point, as well as bit scaling. These issues in turn directly determine the choice of system parameters for hardware implementation, as expounded in Chapter 6. The aim is to establish a relationship between the design requirements and the system parameters to yield an error-free system assuming a noiseless environment.

In SEFDM systems, finite word length effects have to be considered at two stages, namely the modulation-demodulation stage and the detection stage. Here, a simulation based search is employed to identify the minimum bit precision required for different system sizes, bandwidth compression levels, oversampling factors and modulation schemes without compromising performance. While the modulation scheme does not affect the bandwidth requirement, for complex modulation schemes like 4-QAM, the amplitude of the signal will determine the required bit resolution to mitigate quantisation noise. Results show that the choice of the scaling factor for the conversion of a floating-point value to a fixed-point value, in particular for the modulation-demodulation stage, is paramount for the correct decoding of the original data.

5.3.1 Floating-point to Fixed-point Conversion

Up until now, all simulations were carried out on standard computer terminals which have floating-point processing units and plentiful resources to execute computations with a high precision, usually 64 bits in line with the IEEE-754 double precision format yielding results with a high accuracy. Yet, devices that use fixed-point representation, such as DSPs, FPGAs, D/A and A/D converters, have reduced precision capabilities. Hence, it is essential to identify the optimum word length which will allow the system to operate correctly without intolerable errors. First, the terminology commonly encountered in fixed-point applications is outlined:

- **Accuracy:** This metric refers to the correctness of the result, in terms of its difference to the theoretical prediction.
- **Precision:** Ordinarily, precision refers to the variance of the results, in other

words the distribution of the results for a number of repetitions. In this context, precision refers to the number of bits used to represent the data values.

- **Dynamic Range:** The dynamic range is defined as the ratio of the maximum to the minimum amplitude.

The most common format for representing decimal values by their binary equivalent is *two's complement*⁶. If the bit precision is denoted by β , then the range of the corresponding two's complement values is given by $-2^{\beta-1} \leq \chi \leq 2^{\beta-1} - 1$, where χ is an integer number $\chi \in \mathbb{Z}$. The equivalent binary representation is given by $\chi_B = b_{\beta-1}, \dots, b_1, b_0$, where $b_{\beta-1}$ and b_0 are referred to as the Most Significant Bit (MSB) and Least Significant Bit (LSB), respectively. For two's complement numbers, the MSB denotes the sign bit while the remaining bits correspond to the absolute value of the number. Since the decimal numbers encountered in this work are floating-point, the first step involves scaling all the numbers by the same factor in order to convert them to integers, which are inherently fixed-point, while making full use of the available bit precision. This scaling factor is determined by the device's resolution, as well as the largest magnitude in a set of Q data elements $[\chi_1, \dots, \chi_Q]$ with $\chi_{max} = \max(|\chi_1|, \dots, |\chi_Q|)$. Starting with the definition for the maximum positive value of a two's complement number $\chi_{max} = 2^{\beta-1} - 1$, the required bit precision to sufficiently represent all elements in the data set may be computed as follows

$$\begin{aligned} \chi_{max} = 2^{\beta-1} - 1 &\Leftrightarrow \chi_{max} + 1 = 2^{\beta-1} \Leftrightarrow \beta - 1 = \log_2(\chi_{max} + 1) \Leftrightarrow \\ \beta &= \log_2(\chi_{max} + 1) + 1. \end{aligned} \tag{5.22}$$

Eq. 5.22 assumes that the elements in the data set are integers. To include the scenario where χ is any real number, in other words $\chi \in \mathbb{R}$ and thus inherently floating-point,

⁶Two's complement representation allows the same hardware to be employed for addition and subtraction, in addition to having a single representation for zero, in contrast to the conventional *sign and magnitude* format [164].

Eq. 5.22 may be rewritten to compute the minimum required precision as

$$\beta_{min} = \lceil \log_2(\chi_{max} + 1) \rceil + 1. \quad (5.23)$$

In communication systems, the available precision is usually dictated by the resolution of the D/A and A/D converters located at the transmitter and receiver front-ends. Maximum precision is acquired when the full dynamic range of the signals is matched to the resolution of the converters. Conversely, if the range of a signal exceeds the range of the converter, an overflow occurs which may give rise to errors. If the available bit precision is denoted by β_{bx} with $\beta_{bx} \geq \beta_{min}$, then in order to leverage the device's resolution without the risk of overflow, all elements in the data set have to be scaled by a factor equal to 2^γ where $\gamma = \beta_{bx} - \beta_{min}$. This results in the worst-case scaling factor, in the sense that the maximum possible dynamic range of the data for an available precision β_{bx} is exploited while concurrently ensuring that no overflows occur.

The importance of selecting an appropriate scaling factor cannot be overemphasized. As demonstrated during the verification of an FPGA based SEFDM transmitter [165], the bit scaling value has a significant impact on the accuracy of the fixed-point results with respect to the analytical, floating-point results, the latter deemed to be ideal. An in-depth investigation revealed that the scaling used for the rotation matrix of the FPGA transmitter was too high causing the fixed-point data to diverge significantly from the floating-point results. These observations are corroborated in related work (Fig. 3 in [165]) showing that decreasing the bit scaling value from eight bits to six bits leads to a reduction in error.

Two key drawbacks with fixed-point representation is its limited dynamic range and the fact that the hardware designer has to keep track of the scaling applied at different stages of a design. An intermediate solution between fixed-point and floating-point representation is block-floating-point representation. Block-floating-point can support a wide dynamic range comparable to floating-point representation, however, the underlying operations are realised using fixed-point arithmetic, hence combining

the best of both worlds [166]. This is achieved by setting the mantissa equal to that of the input and tracking the scaling automatically using a block exponent. Block-floating-point is suitable for applications demanding a large dynamic range without incurring the high costs required for conventional floating-point arithmetic.

5.3.2 Quantisation Errors

Ordinarily, the process of quantisation refers to the conversion of an analogue signal with infinite precision to a digital signal with finite precision. This reduction in word length leads to round-off errors, known as quantisation errors or quantisation noise. The impact of quantisation noise depends on the resolution of the quantiser, which is measured as one part in $2^{BitResolution}$. To guarantee successful recovery of the transmitted data, the received sampled and quantised signal must match the original signal at the corresponding sampling instants.

This work does not deal with analogue signals, nevertheless, the precision used for the computer based simulations is considered to be high enough to resemble an analogue signal. In MATLAB, the precision depends on the length of the mantissa, which for double precision is equal to 52 bits. This can be confirmed by running the command *eps* which gives approximately $2 \cdot 10^{-16}$ and is equal to 2^{-52} . This resolution, known as the quantisation step and denoted by Δ , represents the smallest possible difference between numbers and determines the error in arithmetic operations. Thanks to its higher internal precision, MATLAB can support floating-point numbers between approximately $1.7977 \cdot 10^{308}$ down to $2.2251 \cdot 10^{-308}$ offering over 600 orders of magnitude dynamic range.

For a real-time hardware implementation, it is hard to achieve this level of precision. Consequently, finite word length effects need to be carefully accounted for which may otherwise degrade system performance. Fixed-point arithmetic is commonly associated with two types of errors; first, quantisation errors, which are caused by the rounding of the results due to the finite bit precision available to represent the data; second,

propagation errors, which pertain to the accumulation of rounding errors during intermediate stages of the computations carried out potentially giving useless results [167].

Bit width optimisation has to take both types of errors into account.

The quantisation error, proportional to the quantisation step size Δ , may be expressed as

$$e_q = (\chi_i)_q - \chi_i, \quad (5.24)$$

where χ_i represents the i^{th} unquantised data sample and $(\chi_i)_q$ the i^{th} quantised data sample. Quantisation errors are uniform and i.i.d. in nature [166].

The standard performance metric for evaluating quantisers is the Signal-to-Quantisation-Noise Ratio (SQNR), expressed in dB as $SQNR = 10 \log_{10} \frac{Signal}{QuantisationError}$. With reference to Eq. 5.24, the SQNR is given by

$$SQNR = \frac{E\{\chi^2\}}{E\{e_q^2\}}, \quad (5.25)$$

where $E\{\cdot\}$ denotes statistical expectation [168].

The two most popular quantisation methods are *convergent rounding* and *truncation* which are also included in Xilinx Intellectual Properties (IP) cores [169]. Convergent rounding assumes that the sample values fall in the centre of the quantisation step Δ , hence the maximum value of e_q will be given by $\frac{\Delta}{2}$ and is mathematically expressed as

$$E\{e_q\} = \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} e_q \frac{1}{\Delta} de = 0. \quad (5.26)$$

The corresponding variance of the quantisation error is given by

$$E\{e_q^2\} = \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} e_q^2 \frac{1}{\Delta} de = \frac{\Delta^2}{12}. \quad (5.27)$$

If truncation is employed instead of convergent rounding, the aforementioned assump-

tion no longer holds. When using truncation, the fractional bits are ignored, thus Eq. 5.26 and Eq. 5.27 become

$$Ee_q = \int_{-\Delta}^0 e_q \frac{1}{\Delta} de = -\frac{\Delta}{2}, \quad (5.28)$$

and

$$Ee_q^2 = \int_{-\Delta}^0 e_q^2 \frac{1}{\Delta} de = \frac{\Delta^2}{3}. \quad (5.29)$$

From the above results, it is evident that convergent rounding results in zero mean with a smaller variance compared to truncation. In principle, the former technique is preferred since it minimises noise variance [169]. In practice, rounding calls for additional hardware resources, while in many cases, techniques like dithering also have to be applied to randomise the error and avoid introducing a bias. Truncation on the other hand has zero cost in hardware. Conclusively, quantisation and rounding constitute paramount design decisions since they have a direct impact on resource utilisation and error performance. The ultimate goal is to represent the data with the highest accuracy required to meet design targets using the minimum number of bits possible. This is the subject of Section 5.3.3.

5.3.3 FPGA Modelling and Simulation

To evaluate the influence of finite word length effects on the performance of SEFDM systems, an FPGA model of a DFT based SEFDM transceiver was created. In particular, a bit-accurate FFT C model provided by Xilinx [170], which generates identical output data to the structural IP core entailed in the subsequent hardware implementation described in Chapter 6, is incorporated in the SEFDM simulation platform. Furthermore, the *Fixed-Point Toolbox*TM in MATLAB is used allowing the designer to configure a number of parameters including the input word lengths, the word lengths of the resultant products, sums and accumulations, the rounding mode and the overflow mode.

The aforementioned behavioural model may appear as restricting flexibility compared to generic, fixed-point models. Nonetheless, this model serves as the best possible approximation for the purpose of functional verification, as it partially emulates the FPGA device considering that the FFT C model behaves in the same manner as the actual hardware module. The overall model is well-suited for validating the fixed-point representations and yielding realistic estimates of the bit precision and dynamic range required for correct system operation. Where conventional floating-point operations are carried out, which are included for comparison purposes, it is assumed that double-precision arithmetic is employed.

Simulation based Search

The results presented in Section 5.3.4 assume that all data values are represented in fixed-point format with computations carried out using fixed-point arithmetic. A prerequisite is to determine the optimum bit widths required at different stages of an SEFDM transceiver to guarantee comparable accuracy provided by corresponding floating-point operations.

Identifying the optimum bit widths usually translates to finding the minimum bit widths yielding the desired level of accuracy. The method adopted here is simulation based search as depicted in Fig. 5.15. Even though crude, this method is the most straightforward process for establishing the best solution since it carries out an exhaustive search of all possible bit widths. The main pitfall of this method is that it can prove to be very time-consuming with the execution time increased in proportion to the number of parameters varied.

A key advantage of employing optimum bit widths in an FPGA design is the fact that a reduction in bit resolution can significantly reduce area and power consumption while simultaneously increasing the maximum achievable clock frequency. Although simulation based search is a laborious task, a designer can reap its benefits during the subsequent hardware implementation. Alternative techniques could be considered

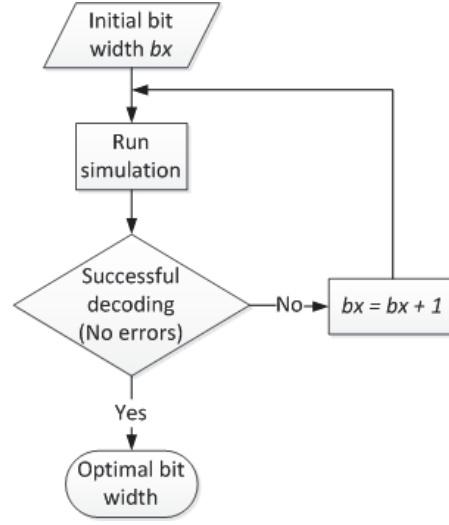


Figure 5.15: Simulation based search to identify optimum bit widths.

in future, such as model based optimisation, which significantly reduce computation time by creating adaptive models with the aid of behavioural profiling which predicts output errors and optimises bit widths accordingly. Examples include the Grid Steepest Descent (GSD) and Accelerated GSD algorithms [167].

Xilinx FFT bit-accurate C model

As presented in Chapter 4, techniques have been developed which allow the modulation and demodulation of SEFDM signals using DFT operations. The DFT in turn can be computed efficiently using the FFT algorithm as implemented in deployed OFDM systems. Here, a bit-accurate C model of the *LogiCORETM* IP FFT provided by Xilinx [171] is employed, which is based on the well-known Cooley-Tukey FFT algorithm [172].

The Xilinx FFT module supports both fixed-point and single-precision floating-point input data. Nonetheless, the latter is very expensive in terms of FPGA resources, as it essentially uses a much higher precision fixed-point FFT internally. For this reason, fixed-point format is used with the data samples represented as signed two's complement numbers while the bit precision supported ranges between eight and 34 bits. Recall

that two's complement fixed-point input numbers may be represented as

$$\Upsilon = -b_{\beta_{bx}-1} + \sum_{k=0}^{\beta_{bx}-2} b_k 2^{-(\beta_{bx}-1)+k}, \quad (5.30)$$

where Υ is the IFFT or FFT input data which takes values in the range $-1.0 \leq \Upsilon < 1.0$, β_{bx} is the number of bits used to represent Υ in two's complement number format and b_k is the k^{th} bit of Υ when expressed in binary format with $k = \beta_{bx}-1, \dots, 0$. The binary point is fixed to the right of the MSB irrespective of the bit width β_{bx} . As a sanity check, the decimal value of Υ having converted it to the required fixed-point format is given by $\chi \cdot 2^{-\beta_{bx}-1}$. For example, if $\chi = -54$ and $\beta_{bx} = 8$, the two's complement equivalent with the binary point at position zero is given by $\chi_{tc} = 11001010$. If the binary point is now shifted to the right of the MSB, the equivalent decimal value will be given by $\Upsilon = -0.421875$.

Scaling and round-off errors must be accounted for carefully in FFT arithmetic operations. The scaling of the input data and the size of the FFT have a direct impact on the output dynamic range and hence SNR [164]. To acquire an in-depth understanding of the effect of the input data width on the dynamic range, the reader is referred to the relevant datasheets provided by Xilinx [171]. If the input values are complex, the numbers processed during the FFT computation may potentially give rise to larger numbers. A scaling strategy must therefore be employed to accommodate the dynamic range expansion. The Xilinx FFT IP core offers three different options for managing scaling, namely *full-precision unscaled arithmetic*, *scaled fixed-point arithmetic* and *block-floating-point arithmetic*.

This work uses full-precision unscaled arithmetic, as it is the most straightforward implementation and it accounts for the worst-case scenario for bit growth [171]. In the general case, the maximum word growth is given by $1 + (r - 1)\sqrt{2}$ where r is the radix decomposition factor, which in turn implies a maximum bit growth of $\lceil \log_2(1 + (r - 1)\sqrt{2}) \rceil$. The factor $\sqrt{2}$ accounts for the potential bit growth in magni-

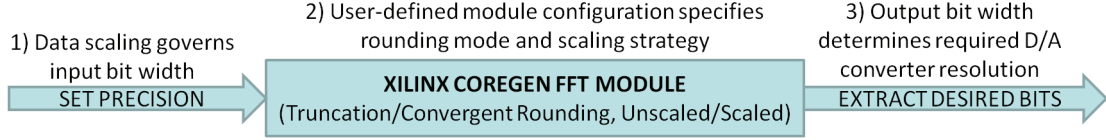


Figure 5.16: Scaling and rounding options at different stages of a Xilinx FFT IP core.

tude when the input is complex, such as $1 + j$, and the vector is rotated by 45 degrees yielding $\sqrt{1^2 + j^2} \approx 1.414$. In this work, it is assumed that all butterfly stages use Radix-2 decomposition, yielding an output bit precision equal to $\beta_{ba} + \log_2(N_{FFT}) + 1$. The +1 bit increase warrants the potential growth when complex inputs are employed. The $\log_2(N_{FFT})$ bit increase is attributed to the fact that an N_{FFT} -point FFT using Radix-2 decomposition has $\log_2(N_{FFT})$ butterfly stages, whereby each stage could potentially cause the result to grow by one bit if the adder was to add two full-scale values.

The round-off errors that occur in fixed-point FFT arithmetic operations depend on a number of factors. First, the transform's properties including its length, in terms of the number of points employed, its type, being either Decimation In Time (DIT) or Decimation In Frequency (DIF), as well as the algorithm employed to execute the transform, all contribute to different round-off errors. Very large block length FFTs can give problems because of the large number of butterfly operations that are cascaded [173]. Second, the resolution available at different stages of the transceiver determines the risk of overflows. Internally, this depends on the bit resolution used for the butterfly stages and the storage of intermediate results which together govern the magnitude of the errors propagated. Externally, the bit resolution is dictated by the specifications of the D/A and A/D converters employed. Last but not least, the numbering format and the rounding strategy also have an impact on round-off errors with rounding necessitated when the magnitude of a result requires a higher bit resolution than the available bit precision.

In summary, a hardware engineer has to consider fixed-point effects at various stages

of a design, as illustrated in Fig. 5.16, which illustrates the scaling and rounding decisions that need to be made when using the Xilinx FFT IP core. For this particular core, the following parameters have to be configured:

- **Input data scaling:** This is determined based on Eq. 5.23 and defines the required bit precision to accurately represent the input data.
- **Architectural choices:** The Xilinx tools offer the user several design options; first, the transform architecture with a choice of burst or pipelined design; second, the choice of unscaled or scaled output data, the latter requiring a scaling strategy; third, the option of applying truncation or convergent rounding to the results. These choices in turn dictate the trade-off between error performance, which is affected by round-off errors, throughput, which depends on the overall latency of the design, and resource requirements.
- **Output data bit resolution:** The number of bits taken forward usually depends on the resolution of the D/A converter, and in this case, on the configuration of the FFT module at the receiver. If the output bit width is too long, it can be truncated by either i) extracting directly the desired number of LSBs from the unscaled output, or ii) Shifting the result to the right, which is equivalent to dividing by a power of two, followed by extraction of the desired number of LSBs. The choice depends on the dynamic range of the output data, which in turn is dictated by the scaling of the input data.

An in-depth investigation of all the different scaling options and their trade-offs is beyond the objectives of this work. To simplify the forthcoming performance evaluation presented in Section 5.3.4, full-precision unscaled arithmetic is adopted for the simulation of the Xilinx FFT C model which guarantees zero overflows. Thereby, the only used-defined parameter is the scaling of the data at the input of the IFFT which has a direct impact on the dynamic range of the output data, the required resolution of the D/A and A/D converters and subsequently the bit width of the data samples at

the input of the FFT. The hardware designer has the option of extracting the LSBs or any custom bit range per se of the IFFT and FFT outputs provided that the number of extracted bits is sufficient to accurately represent the data.

Fixed-point Representation for Matrix Elements

Recalling from Section 4.4, since \mathbf{R} is equivalent to a fraction of the FFT output samples, the quantisation of \mathbf{R} is inherited from the quantisation properties of the Xilinx FFT block. The scaling used for the IFFT and FFT blocks at the transmitter and at the receiver, respectively, represents the scaling for the modulation and demodulation stages. Recall that linear detection involves the multiplication of \mathbf{R} with the inverse of the sub-carriers correlation matrix \mathbf{C} , the latter computed either directly yielding \mathbf{C}^{-1} or via TSVD giving \mathbf{C}_ξ . Therefore, for the detection stage, the fixed-point representation of \mathbf{R} and \mathbf{C}^{-1} or \mathbf{C}_ξ has to be taken into account.

While the fixed-point evaluation in Section 5.3.4 is carried out in a noiseless environment and thus uses ZF as the detection technique, the hardware implementation presented in Chapters 6 and 7 considers the operation of an SEFDM transceiver in an AWGN environment. For this reason, the current section examines and compares the dynamic range and scaling requirements of both \mathbf{C}^{-1} and \mathbf{C}_ξ matrices as they influence the design of the overall FPGA architecture.

One of the key challenges encountered during the generation of the ZF or TSVD estimate is the computation of the inverse or the pseudoinverse of \mathbf{C} attributed to its unbounded dynamic range [61]. This is illustrated in Fig. 5.17 which depicts the dynamic range, denoted by DR , for both \mathbf{C}^{-1} and \mathbf{C}_ξ . Here, the dynamic range is calculated as the ratio of the maximum to the minimum absolute value amongst all elements (of both real and imaginary parts) of \mathbf{C}^{-1} and \mathbf{C}_ξ .

It would be impractical to represent every element of these matrices in a dedicated hardware implementation. Consequently, the values need to be converted to fixed-point format using a finite number of bits, which inevitably results in certain values being

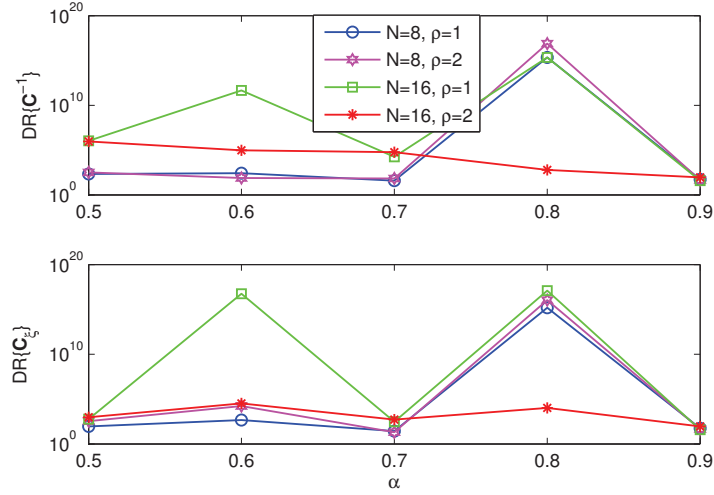
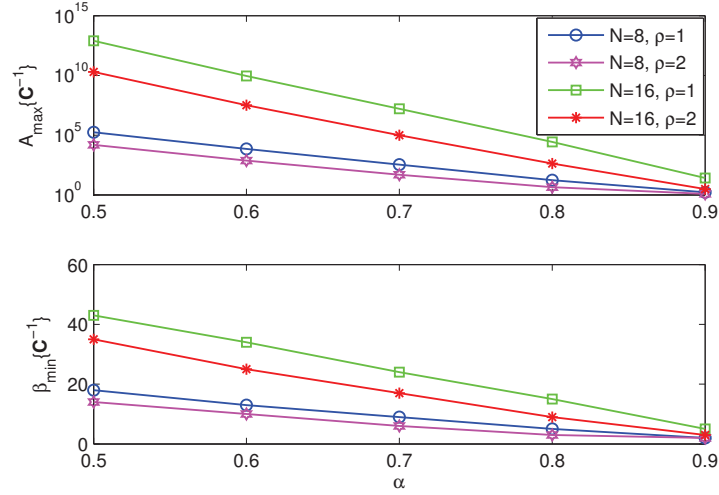


Figure 5.17: Illustration of the unbounded dynamic range (DR) for \mathbf{C}^{-1} and \mathbf{C}_ξ .

either truncated or saturated. Fig. 5.18 shows the maximum values of the real part of \mathbf{C}^{-1} for different values of N , ρ and α . Similar magnitudes are observed for the imaginary part of \mathbf{C}^{-1} . These maximum values A_{max} dictate the minimum number of bits β_{min} required to represent the highest value without saturation.

The top sub-plot in Fig. 5.18, indicates that the maximum value of \mathbf{C}^{-1} increases exponentially according to the reduction in spacing between the sub-carriers. The lower sub-plot in Fig. 5.18, shows that for a gradual reduction in α by 10%, approximately ten additional bits are required each time when $N = 16$. When $N = 8$, this bit requirement is reduced to approximately half. In Section 5.2, it was shown that oversampling ameliorates the condition number and improves the performance of SEFDM detection schemes in an AWGN environment. This is reflected in Fig. 5.18, which demonstrates that oversampling reduces the maximum values of \mathbf{C}^{-1} and thus relaxes the bit precision requirements.

Fig. 5.19 depicts the maximum values for the TSVD matrix \mathbf{C}_ξ . Thanks to the truncation of the small singular values, the highest value is very small compared to \mathbf{C}^{-1} and fairly constant across the values of α . The sudden peaks observed for $\alpha = 0.8$ and $\alpha = 0.9$ are due to the fact that the particular combinations of N and α , corresponding

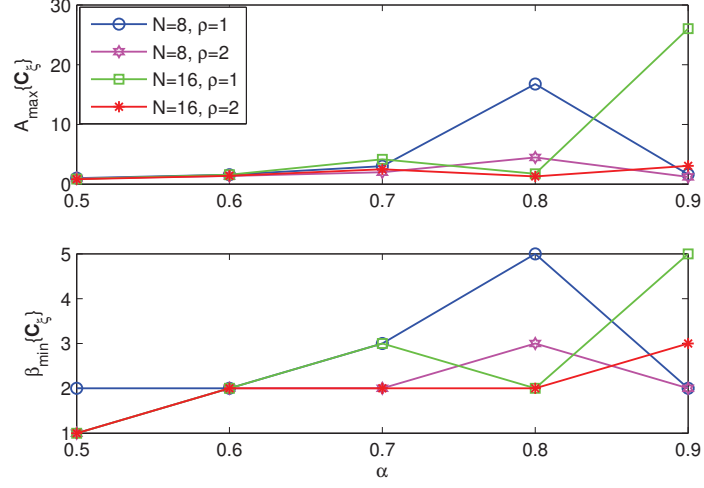
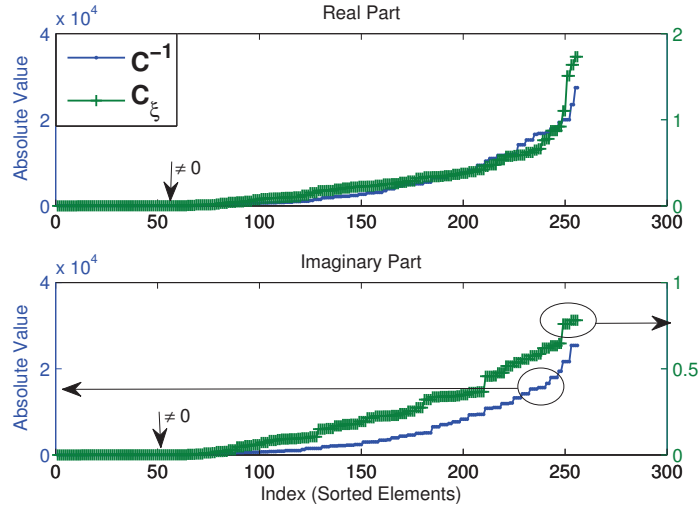

 Figure 5.18: Maximum values and minimum required bits for \mathbf{C}^{-1} .

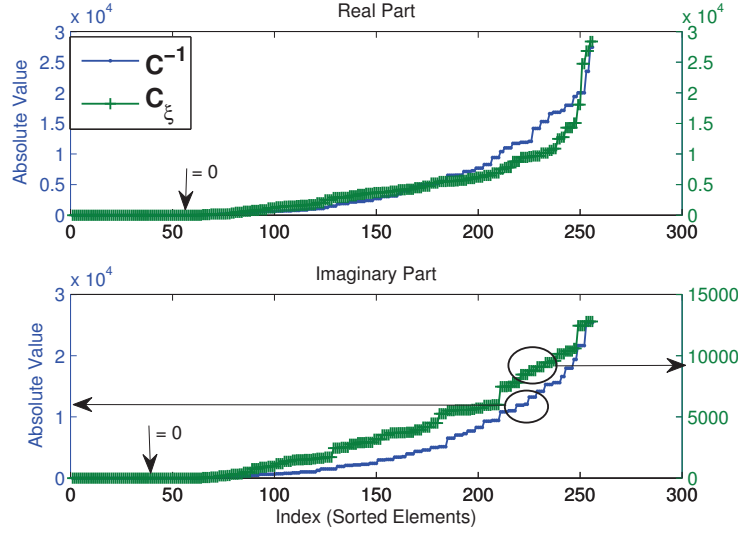
to these peaks, result in the ZF and TSVD matrices being commensurate.

Fig. 5.20 compares the distribution of the elements pertaining to the real and imaginary parts of \mathbf{C}^{-1} and \mathbf{C}_ξ for $N = 16$ and $\alpha = 0.8$ having sorted the elements in ascending order of absolute values. Fig. 5.20 illustrates that the two matrices exhibit similar trends with regard to the distribution of their elements. Notwithstanding, the maximum value for \mathbf{C}^{-1} is almost four orders of magnitude higher compared to the maximum value of \mathbf{C}_ξ .

Hence, the question arises as to what happens when a finite number of bits are used to represent these matrix elements? To answer this question, the minimum number of bits required to represent the highest value of \mathbf{C}^{-1} in Fig. 5.20 is considered first. In this case, this corresponds to $\beta_{\min}\{\mathbf{C}^{-1}\} = 15$ bits. Subsequently, the elements of \mathbf{C}^{-1} and \mathbf{C}_ξ are scaled up and rounded, based on this bit precision, yielding the results illustrated in Fig. 5.21. It is clear that the curves shown in Fig. 5.21 are in accordance with the curves illustrated in Fig. 5.20, nevertheless, the small values that are less than one are truncated in the scaled and rounded versions.

The key objective during any hardware implementation is to acquire the highest accuracy using the least number of bits possible. Fig. 5.22 demonstrates the impact a


 Figure 5.19: Maximum values and minimum required bits for \mathbf{C}_ξ .

 Figure 5.20: Distribution of the matrix elements pertaining to \mathbf{C}^{-1} and \mathbf{C}_ξ ($N = 16$, $\alpha = 0.8$).


 Figure 5.21: Comparison of the scaled distribution of \mathbf{C}^{-1} and \mathbf{C}_ξ .

reduction from 24 bits down to eight bits has on the distribution of the elements in \mathbf{C}_ξ (note that only the real part is shown in Fig. 5.22, however, the results are applicable to the imaginary part too). A reduction in bit precision causes the distribution of the elements to diverge from the original, floating-point version, however, the lower sub-plot in Fig. 5.22 shows that the trends are very similar.

Conversely, the impact of a reduction in bit precision is more severe for \mathbf{C}^{-1} . As shown in Fig. 5.23, reducing the number of bits from 24 down to eight causes the matrix values to saturate very rapidly compared to the values for \mathbf{C}_ξ in Fig. 5.22. This saturation in turn could give rise to significant numerical errors, thus degrading BER performance. Nonetheless, as discussed in Section 5.2, TSVD is the best performing sub-optimum linear detector in an AWGN environment, hence the results presented in this section are particularly relevant to the FPGA implementation of TSVD described in Chapter 6.

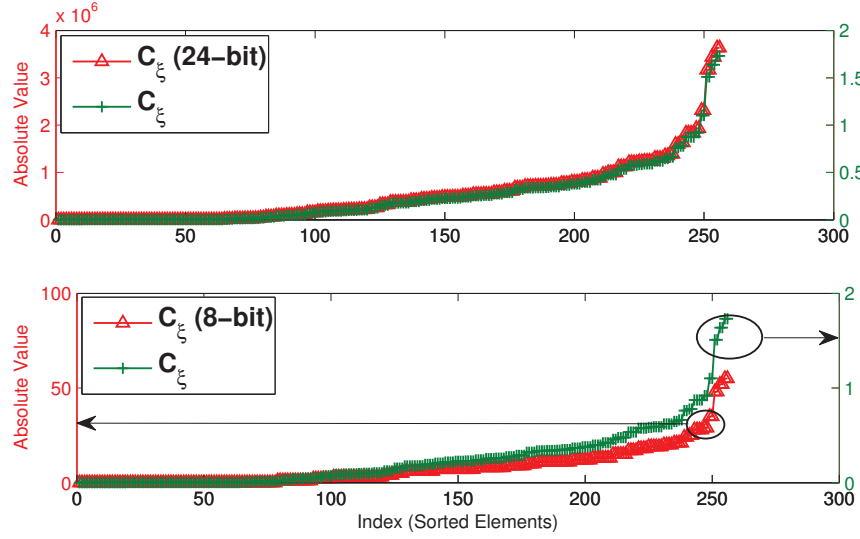


Figure 5.22: Comparison of the scaled distribution for different bits for \mathbf{C}_ξ .

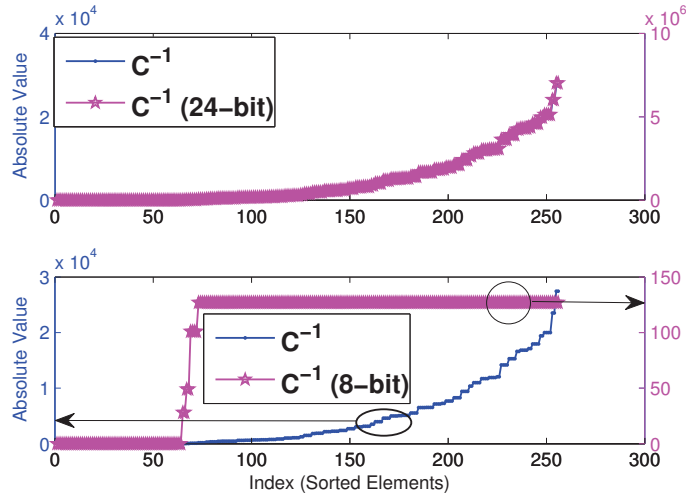


Figure 5.23: Comparison of the scaled distribution for different bits for \mathbf{C}^{-1} .

5.3.4 Numerical Results

The fixed-point evaluation presented in this section is achieved by simulating the FPGA behavioural model described in Section 5.3.3 in a noiseless environment, thus any errors that occur are strictly due to quantisation noise. The inner product of \mathbf{R} with \mathbf{C}^{-1} is computed using fixed-point operations. The results of ZF detection are used to determine the optimum bit widths recalling that ZF is the optimum detection technique in SEFDM when no noise is present. Since this evaluation aims to identify the worst-case scaling, all possible data symbol combinations are considered in a similar fashion to a maximum length Pseudo Random Binary Sequence (PRBS). The number of different parameter combinations is limited in this evaluation for the following reasons:

1. In addition to the default SEFDM system parameters, such as the number of sub-carriers N , the bandwidth compression level α , the oversampling factor ρ and the modulation scheme (BPSK or 4-QAM), additional parameters have to be taken into account like the IFFT module's input bit width, as well as the bit precision used to represent the elements of the straightforward or TSVD matrix inverse.
2. The time required to carry out fixed-point computations in MATLAB is significantly longer compared to conventional floating-point operations, especially when simulating the Xilinx FFT C model. In particular, the Xilinx FFT C model takes on average 0.05 seconds for a complete execution, which is 2,500 times slower compared to the execution of the native MATLAB *fft* function, that takes on average $2 \cdot 10^{-5}$ seconds to execute.
3. The previous point, in conjunction with the fact that all possible data symbol combinations are examined, results in an overly extended simulation time. For example, for $N = 16$ and BPSK, the number of different combinations is equal to $2^{16} = 65,536$, which is over a factor of six greater than the average number of random SEFDM symbols simulated in an AWGN environment which is typically 10,000. The tedium of fixed-point simulation is exemplified by the fact that a one-

off simulation for a single set of the aforementioned parameters for 65,536 SEFDM symbols takes approximately 2.5 hours to complete on a machine equipped with an Intel Core i7 processor running at 2.89 GHz and with 4 Giga Bytes (GB) of Random Access Memory (RAM).

In the following figures, γ_F denotes the amount of scaling necessitated at the input of the Xilinx IFFT module at the transmitter's end to yield zero errors, expressed in terms of bit precision. In a similar fashion, γ_C denotes the amount of scaling necessitated when converting the matrix inverse elements to fixed-point format. For example, the default values of BPSK symbols are $+1$ and -1 . Without scaling, these values would require one bit to represent the magnitude, plus an additional bit serving as the sign bit. If scaling is applied, these values are multiplied by a power of two, thereby increasing the IFFT module's output dynamic range rendering the signal more immune to quantisation errors. The subsequent discussion considers only the bit precision required to represent the magnitudes of the data values and assumes that in practice, an extra bit would be made available to differentiate between positive and negative numbers.

Fig. 5.24 compares the required bit precision for BPSK and 4-QAM when eight sub-carriers are employed. Overall, 4-QAM requires a higher γ_F than BPSK, especially for high bandwidth compression levels ($\alpha < 0.7$). In both cases, a reduction in frequency spacing leads to a quasi-exponential increase in γ_F . Conversely, γ_C is fairly constant for both BPSK and 4-QAM, though a slightly higher γ_C is required for 4-QAM. Fig. 5.24 also shows that oversampling results in marginal improvement, especially for higher values of α .

Fig. 5.25 compares the bit precision requirements when the number of sub-carriers is doubled. First, it is clear that doubling N from eight to 16 requires more than double the number of bits to yield zero errors. Fig. 5.25 also indicates that no value of γ_F was found for $\alpha \leq 0.6$ when $N = 16$. This is attributed to the limitations of the Xilinx IFFT and FFT modules. In this case, the size of the Xilinx IFFT and FFT modules has been set to $N_{FFT} = 64$ (the reason for this value is expounded in Chapter 6).

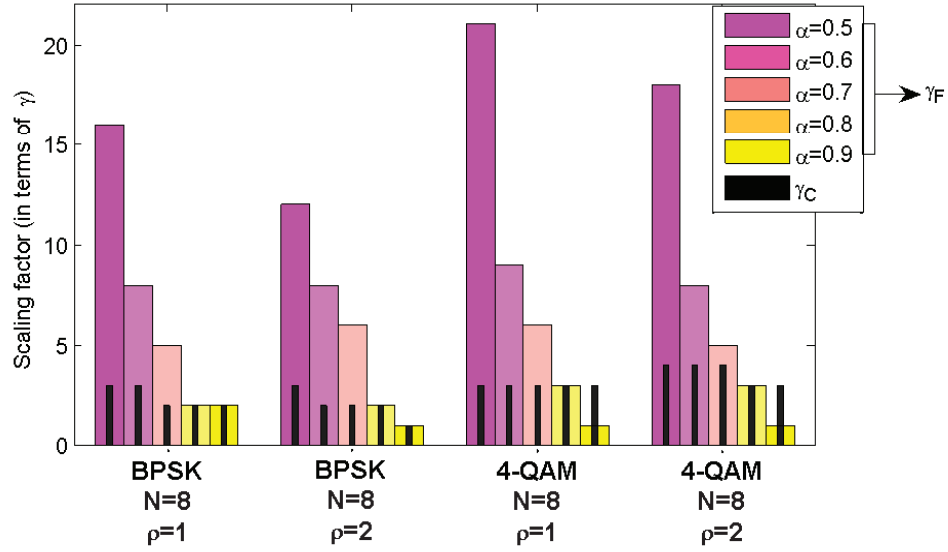
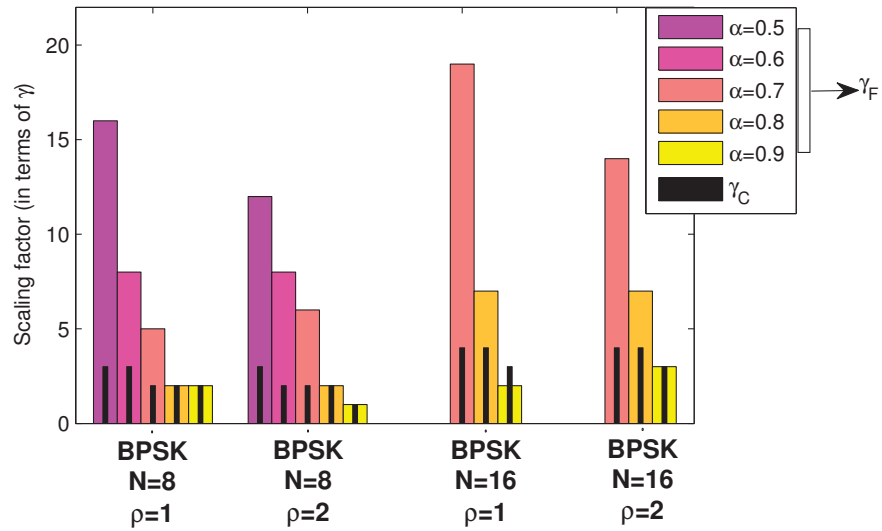

 Figure 5.24: Bit precision requirements to yield zero errors for $N = 8$.


Figure 5.25: Bit precision requirements to yield zero errors for BPSK.

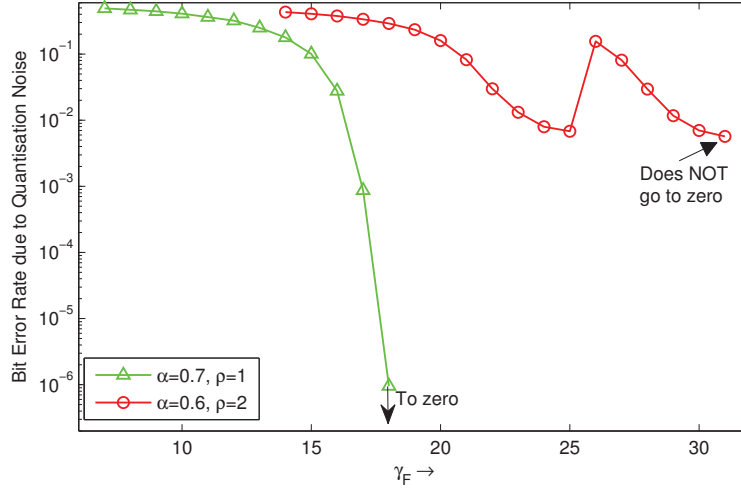


Figure 5.26: Error performance due to quantisation noise for BPSK with $N = 16$.

This implies a bit growth of $6 + 1 = 7$ bits, to account for complex multiplications too, recalling that full-precision unscaled arithmetic is adopted for this evaluation. In view of the fact that the input bit precision of both the IFFT and FFT modules is limited to a maximum of 34 bits [171] of which one bit is reserved for the sign bit and another for preventing overflow (for example, $\lceil \log_2(2^5 + 1) \rceil = 6$), it should be evident that the maximum value for γ_F at the input of the IFFT module cannot exceed 25 bits. This is illustrated in Fig. 5.26. An attempt to find a value for γ_F when $\alpha = 0.6$ and $N = 16$ was carried out by scaling down the IFFT outputs by N_{FFT} . In other words, the dynamic range of the IFFT module's output was reduced by six bits causing the abrupt jump seen in Fig. 5.26 when increasing γ_F from 25 to 26 bits (red curve with circle markers). Unfortunately, even this downscaling of the values did not yield a value for γ_F which would generate zero errors for the set $\alpha = 0.6, N = 16, \rho = 2$. Fig. 5.26 also corroborates the results presented in Fig. 5.25, as indicated by the γ_F values for the set $\alpha = 0.7, N = 16, \rho = 1$.

Figs. 5.27 and 5.28 depict the error performance due to quantisation noise when BPSK is employed for eight sub-carriers. It is evident, that the bit precision requirements increase significantly when $\alpha < 0.8$. In this latter case ($\alpha < 0.8$), oversampling

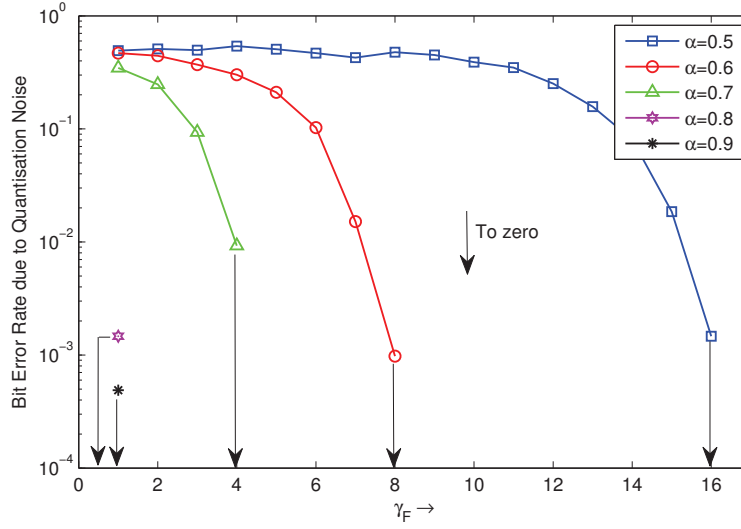


Figure 5.27: Error performance due to quantisation noise for BPSK with $N = 8$ and $\rho = 1$.

ameliorates the error performance. Similar error performance trends are demonstrated for 4-QAM in Fig. 5.29.

To conclude, it is worth pointing out that in certain cases, a lower value for γ_F can be traded with a higher value for γ_C . For example, in Fig. 5.24, the bit precision requirements for $\alpha = 0.7, \rho = 2$ are given by $\gamma_F = 6, \gamma_C = 2$. The results acquired via simulation demonstrated that for the same set of system parameters, no errors would occur for $\gamma_F = 4, \gamma_C = 3$. Hence, in this case, increasing γ_C relaxes the requirements for γ_F . In practice, this may prove to be an attractive scenario, since γ_F usually dictates the resolution of the D/A and A/D converters, whereas the additional bit required for the scaling of the matrix elements can be accommodated much more efficiently in FPGA fabric. Finally, the improvements noticed when oversampling is employed, may be attributed to the fact that oversampling translates to observing multiple samples of the same symbol, thereby reducing the uncertainty of a received symbol [174].

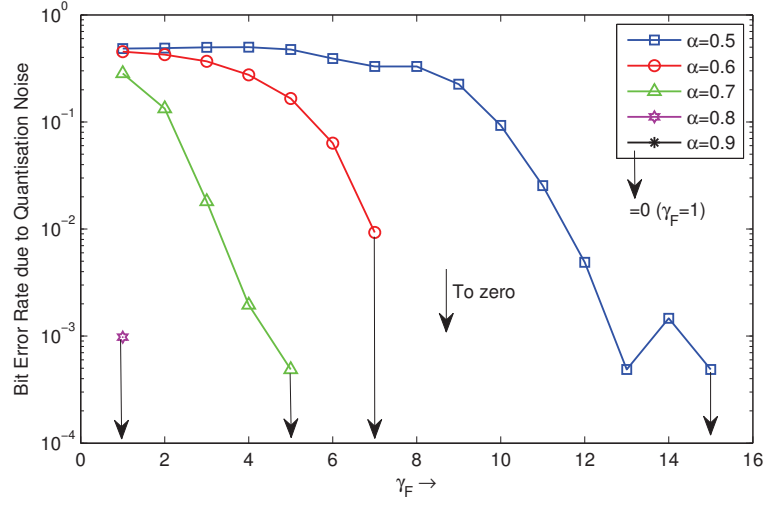


Figure 5.28: Error performance due to quantisation noise for BPSK with $N = 8$ and $\rho = 2$.

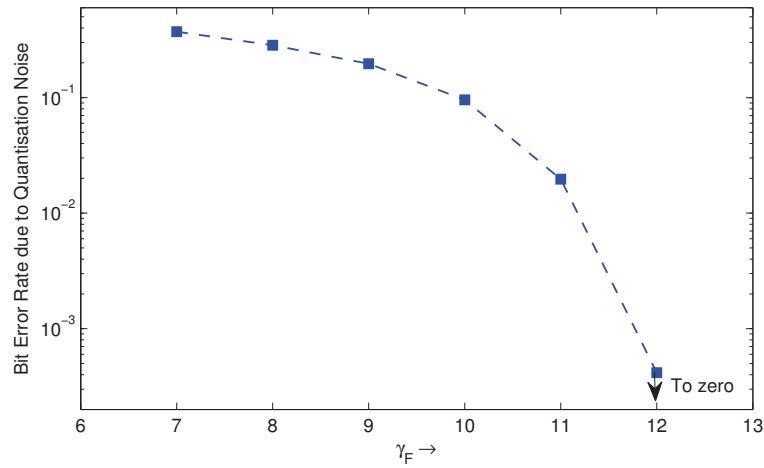


Figure 5.29: Error performance due to quantisation noise for 4-QAM ($N = 8$, $\alpha = 0.5$, $\rho = 2$).

5.4 Conclusions

This chapter examined the performance of SEFDM systems taking into account practical aspects, such as oversampling and finite word length effects. The chapter commenced with an overview of different detection techniques applied in SEFDM, with a particular emphasis on linear and iterative detectors.

Subsequently, the performance gains that could be acquired via oversampling were considered. It was found that oversampling can improve the conditioning of the sub-carriers correlation matrix, thus ameliorating BER performance in an AWGN environment. For the special case where $\alpha = 0.5$ (Fast OFDM) and the modulation scheme is BPSK, MF on its own can achieve the optimum solution provided that the number of sub-carriers is sufficiently high ($N > 100$) or an oversampling factor of at least $\rho \geq 2$ is used when N is small (for example, when $N = 8$). For all other cases, it was shown that MF followed by IC outperforms all remaining detection schemes, especially for low SNR regimes ($E_b/N_0 < 10dB$). With regard to linear detection methods, TSVD was found to outperform ZF and MMSE, rendering the technique the best performing method after MF-IC. Notwithstanding, subsequent chapters in this thesis examine hybrid detectors combining linear detection techniques with more sophisticated algorithms, such as FSD. Consequently, preliminary results presented in this chapter illustrated that TSVD followed by FSD outperforms the remaining hybrid detection methods. Therefore, TSVD was chosen as the preferred method for generating a first estimate of the original transmitted symbols.

The final part of this chapter looked at the effects of finite arithmetic precision on the performance of SEFDM systems. A number of issues including floating-point to fixed-point conversion, numerical accuracy, bit precision and dynamic range were discussed. An FPGA behavioural model was used to evaluate the trade-offs between bit precision and quantisation errors. The minimum bit precision required to yield an error-free system in a noiseless environment was identified and compared for different values of N , α and ρ . This evaluation further revealed the limitations of the FPGA IP

cores provided by Xilinx with respect to the maximum supported bit precision. The impact of the finite number of bits available in dedicated hardware on the representation of matrix elements was also considered. It was determined that a small number of bits, for example eight bits, has an important but non-detrimental effect on the distribution of values in the TSVD matrix. Conversely, the effect was found to be a lot more severe for the straightforward ZF matrix inverse.

In conclusion, oversampling ameliorates the performance of SEFDM systems as a result of observing more samples of the same received symbol. Oversampling also relaxes bit precision requirements since the quantisation noise is spread over a wider range of samples for the same number of data symbols. However, in all cases, proper scaling is required to avoid overflow while concurrently leveraging the available bit precision to optimise the output dynamic range.

Chapter 6

Implementation of Linear Detection Techniques

Chapters 4 and 5 were devoted to reviewing the design specifications of an SEFDM transceiver and the implications these have on hardware requirements. A common challenge encountered in real-world scenarios is that simulation results are not always in accordance with results acquired via practical experimentation [163]. There are two key reasons for this discrepancy; first, it is hard to predict the exact behaviour of the target device, whose results may vary due to hardware non-idealities, such as clock jitter and temperature variations; second, the normalisation employed may not be identical for both simulation and experimentation.

Hitherto, no particular methodology has been adopted to predict the true silicon complexity and real-time performance of the developed algorithms under consideration. This calls for a hierarchical analysis, design and implementation process. To this end, Section 6.1 presents the basics of FPGA devices followed by a description of the hardware platform and prototyping methodology in Section 6.2. The steps required to take an algorithm from concept through to its final FPGA realisation are set forth. Section 6.3 details the hardware design of ZF and TSVD techniques using FPGAs. The main purpose of this section is to evaluate the feasibility and performance of the TSVD de-

tector for use in practical SEFDM applications. This is the first time a hardware design for the TSVD algorithm has been devised using VHDL to run on FPGAs. Results show excellent fixed-point performance comparable to existing floating-point computer based simulations. The optimum parameters required to achieve this outcome combined with their effect on system performance are also identified.

Having presented the FPGA design considerations, Section 6.3.2 considers the FPGA implementation of the aforementioned SEFDM detection methods. This is particularly important given that FPGA and VLSI approaches for SEFDM transmitters have been the subject of related work [175] [176]. Furthermore, VLSI implementations of FTN transceivers have also come to light [177]. This final part demonstrates that the performance of an FPGA based TSVD hardware receiver is comparable to that obtained from the modelling carried out by Isam [156], thereby verifying, for the first time, the practicability of TSVD detection. The impact of finite FPGA resources against performance gain is also examined.

The research in this chapter was presented at multiple conferences including the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC) in 2011 [178] and 2012 [179], as well as the London Communications Symposium (LCS) in 2011 [180].

6.1 FPGA Preliminaries

Contrary to general-purpose processors, such as Central Processing Units (CPUs), Graphics Processing Units (GPUs) and DSPs, which are designed to execute a set of instructions in a sequential manner, FPGAs are programmable logic devices with dedicated hardware circuitry allowing multiple operations to be evaluated¹ concurrently thanks to their unique parallel execution capabilities. An FPGA consists of switching matrices which route the signals between the different programmable logic blocks on

¹An FPGA is programmed to evaluate logical functions instead of carrying out sequential computations as a general-purpose processor.

the device, hence programming an FPGA is essentially equivalent to “architecting a circuit” [181]. FPGAs are becoming increasingly popular because:

- They are capable of delivering throughputs comparable to multi-core CPUs and GPUs at a fraction of the cost and power [163].
- They favour reconfigurable designs, which in turn are often modular, thus allowing easier expansion and upgradability.
- They include dedicated clock control circuitry allowing the system to run at any rate [182].
- All bit widths are parameterisable, thus saving resources and allowing the hardware designer to select any custom bit precision depending on the application requirements. In contrast, processor based systems are bound to specific bit widths, for example 8-, 16-, 32- and more recently 64-bit precision, thereby severely limiting flexibility [163].

The key drawback of FPGAs is that they do not perform well for floating-point operations, however, the advent of the latest Xilinx Virtex 7 devices is likely to change the scene [163]. In current devices, carrying out floating-point arithmetic usually translates to employing fixed-point arithmetic with a much higher precision which can be very expensive in terms of resources and restricts the maximum achievable clock frequency. Chapter 5 demonstrated that the main pitfall associated with fixed-point arithmetic is its narrow dynamic range. FPGAs can overcome this challenge via dynamic bit growth at appropriate stages in the signal processing chain [181], a feature attributed to their programmable logic nature. In support of this argument, recall from Section 5.3 that fixed-point arithmetic allows the implementation of a design that provides sufficient numerical accuracy without the costly resources and power required for an equivalent floating-point design.

The upper bound on the performance of an FPGA based system is limited by the availability of finite hardware resources, referred to as the FPGA fabric, the maxi-

imum achievable clock frequency and the processing latency [183]. FPGA resources are broadly categorised into i) general-purpose slices, also known as Configurable Logic Blocks (CLBs), ii) dedicated digital signal processing slices, which are prebuilt blocks designed to execute fundamental arithmetic operations in a highly efficient manner and are thus known as embedded multipliers or DSP Slices, iii) Block Random Access Memories (BRAMs) used to store large data sets. CLBs comprise Flip-Flops (FFs) and Look Up Tables (LUTs). The former are simply binary shift registers, whereas LUTs act as truth tables stored in memory generating a specific output for a predefined set of inputs. FFs and LUTs constitute the fundamental building blocks of FPGA devices.

Embedded multipliers allow hardware designers to harness the computational power of FPGA devices. In this work, they are employed to carry out FFT operations and matrix operations involving complex multiplications. The remaining fabric is allocated to performing auxiliary system tasks, such as command and control, achieved with the aid of Finite State Machines (FSMs), which in turn are implemented using LUTs and MUXs. FFs serve as delay nets while BRAMs are commonly used to construct First In First Out (FIFO) memories, the latter acting as buffers given that the output data rates may be much higher than the available communication bandwidth. The choice of DSP Slices or CLBs is largely determined by the intended application or in this work by the complexity of the algorithm to be implemented in hardware. The use of embedded multipliers prevails when performance optimisation is paramount. Conversely, if a more cost-efficient design is sought, DSP Slices may be substituted by general-purpose blocks, such as ‘shift and add’ operations, at the expense of performance.

The ultimate goal of any FPGA design is to utilise all available resources as efficiently as possible to yield high throughput gains. The throughput depends on the running clock frequency of the design and the latency of its operation in terms of the number of clock cycles required for a complete execution. The two extreme solutions are a fully-combinatorial versus a fully-pipelined design. In theory, a design which is solely combinatorial offers the highest throughput since it results in a fully-parallel ar-

chitecture with negligible latency. Yet, this approach necessitates the highest resource utilisation and may hinder the maximum achievable clock frequency if the critical path delay² is not handled properly. In contrast, a fully-pipelined design can theoretically achieve the maximum clock frequency with the lowest resource utilisation at the expense of increased latency. In practice, a design combines parallel and pipelined stages to yield the best clock frequency performance. Certain tricks are applied by hardware designers to improve the target clock frequency, such as double-registering the input and output ports to reduce the clock dependence on Input/Output (I/O) placement [169], as well as ensuring that all blocks are active on every clock cycle. Evidently, applying these tricks incurs additional clock cycles to complete an operation, nonetheless, the end result is almost always more gainful. The fewer the number of logic levels and the lower the bit precision, the higher the clock frequency that may be achieved.

In the real world, the trade-offs are even more complicated due to non-ideal conditions. For example, the inevitable jitter associated with LOs may give rise to timing errors and frequency offsets, which in turn may lead to incorrect decoding. Moreover, the finite rise and fall times of digital data imposes an upper bound on the minimum clock period, thereby imposing an additional bound on the maximum achievable throughput. Vendor tool settings may also have a significant impact on the area use and the speed of any FPGA design [169] [171] [184]. Finally, as expounded in Chapter 5, finite arithmetic precision results in quantisation and rounding errors, which when accumulated could reduce system accuracy causing performance degradation.

The aforementioned factors show that the computational complexity of any design is bounded by the finite resources of an FPGA device. The key bottleneck in practice lies at the interface between the FPGA and external peripherals. This is especially true in systems which comprise both CPU and FPGA devices, where ordinarily the former act as the ‘brains’ of the system with the latter responsible for the actual

²The critical path delay is the delay of the longest I/O path in an FPGA design, hence constituting the most crucial timing constraint having a direct impact on the maximum frequency at which the FPGA clock can run.

‘number crunching’. Consequently, there is an increasing trend for deploying CPU-less systems relying entirely on the use of FPGAs, particularly where low latency is of vital importance³. To conclude this section, it is worth outlining the technical challenges that were encountered in this work:

- **Matrix computation:** VHDL has not been designed for the purpose of matrix operations as in MATLAB. Yet, matrix operations are crucial for the design and evaluation of SEFDM systems, hence methods for constructing and manipulating them in VHDL are essential.
- **Synchronisation:** The fundamental component in any synchronous design is the system clock. All processes must operate in accordance with this clock, whereby there is always a latency between the moment a signal takes on a new value and the moment this new value is read by another block whose input is this signal. Due to the inherent complexity of matrix computations, a small glitch in timing could cause an inconspicuous system malfunction.
- **Numerical precision:** Recalling from Section 5.3, the process of floating-point to fixed-point conversion is not trivial. Suitable bit widths and word lengths must be carefully selected to minimise errors and generate accurate and consistent results.

6.2 Hardware Platform and Prototyping Methodology

This section describes the testbed employed to implement the developed SEFDM detection algorithms in hardware. The prototyping methodology combines analytical tools and a standalone hardware platform incorporating FPGAs and DSPs. A simple layout of the SEFDM testbed transceiver is illustrated in Fig. 6.1. The computer terminal is

³For informative purposes, financial institutions are progressively migrating to FPGA solutions for high frequency trading and low latency market data. While conventional systems use CPUs as the ‘brains’ of the system and FPGAs for offloading functions and data processing, communication burden has led to the emergence of *FPGA Centric* systems which are CPU-less and employ a matrix of FPGA devices, thus making the overall architecture more scalable and less prone to bottlenecks [185].

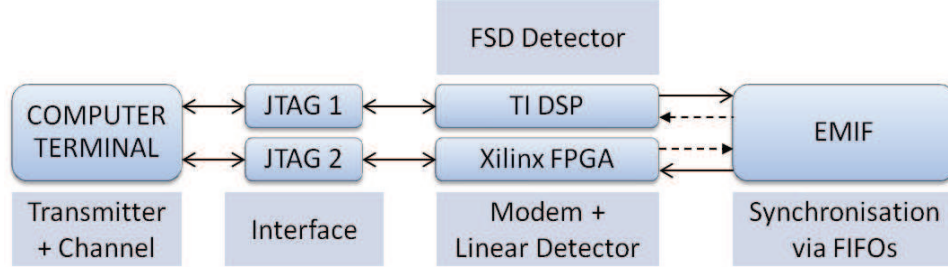


Figure 6.1: Outline of SEFDM testbed transceiver.

used to model the transmitter and the channel, as well as for the purpose of analysing the results acquired through simulation and experimentation. A software stack is used to exchange data between the dedicated devices and the computer terminal. A number of interfaces are available for this communication including Joint Test Action Group (JTAG), Universal Serial Bus (USB), Ethernet and Peripheral Component Interconnect (PCI) Express. The default method is JTAG which transfers data between the FPGA device and the computer terminal by accessing registers.

Chapter 7 will look at the execution of the FSD algorithm using DSPs and will provide further details regarding the interface between the FPGA and the DSP. This section focuses on the implementation of linear techniques, specifically ZF and TSVD, with the aid of FPGAs. Contrary to other work [186] [187] which use high-level synthesis tools, such as *Xilinx System Generator for DSPTM* (henceforth referred to as System Generator), the methodology adopted in this work requires detailed knowledge of hardware description languages. The disadvantage of this latter method is the fact it is time-consuming. Nevertheless, hand-coding gives the designer full control over the architecture and allows optimisation to be carried out at the component level. Moreover, it allows the designer to acquire an in depth understanding of the underlying operations, which is of vital importance when novel systems like SEFDM that utilise non-standard functions are targeted for hardware implementation.

A Xilinx Virtex 4 XC4VFX12 chip was employed for the initial feasibility study described in Section 6.3.1. The chip was then upgraded to a Virtex 5 XC5VFX100T for the

actual implementation and real-time verification of a reconfigurable SEFDM receiver as demonstrated in Section 6.3.2. The latest version of the system is currently being migrated⁴ to an even more advanced platform comprising a Virtex 6 XC6VLX240T chip, which has already been used for the FPGA implementation of a standalone FSD detector, as reported in [188]. At the time of writing, the Xilinx 7 series FPGAs are the state-of-the-art capable of delivering up to 1.33 teraflops⁵ on a single device [163].

6.3 FPGA Design and Implementation of ZF and TSVD

This section describes the design of hardware-efficient algorithms targeted for implementation on an FPGA device. Novel architectures are presented, which to the author's best knowledge, are a world-first. Recalling from Section 5.2.2 and in accordance with the work in [156], the TSVD detector yields significant performance gains in an AWGN environment in terms of BER when compared to ZF and MMSE without incurring a complexity penalty. For this reason, the forthcoming discussion will focus on the hardware realisation of TSVD.

Fig. 6.2 gives a top-level representation of the SEFDM transceiver model, which is partitioned between the analytical and the FPGA environments. A memory interface passes input and output values to and from the FPGA. The entire process may be summarised as follows:

1. At the transmitter's end, the TSVD algorithm is run offline to obtain the \mathbf{C}_ξ entries for a given set of N , α and ρ . A script is then used to generate the noisy SEFDM time samples $r(t)$ which arrive at the receiver's input.
2. At the receiver's end, the first step involves converting the \mathbf{C}_ξ entries and the SEFDM samples from floating-point to fixed-point format, as the FPGA architecture accepts input data only in the latter format.

⁴The continuation of this work aimed at devising a complete SEFDM transceiver on a single FPGA chip was commenced by Tongyang Xu at UCL in late 2012.

⁵flops, stands for floating-point operations per second.

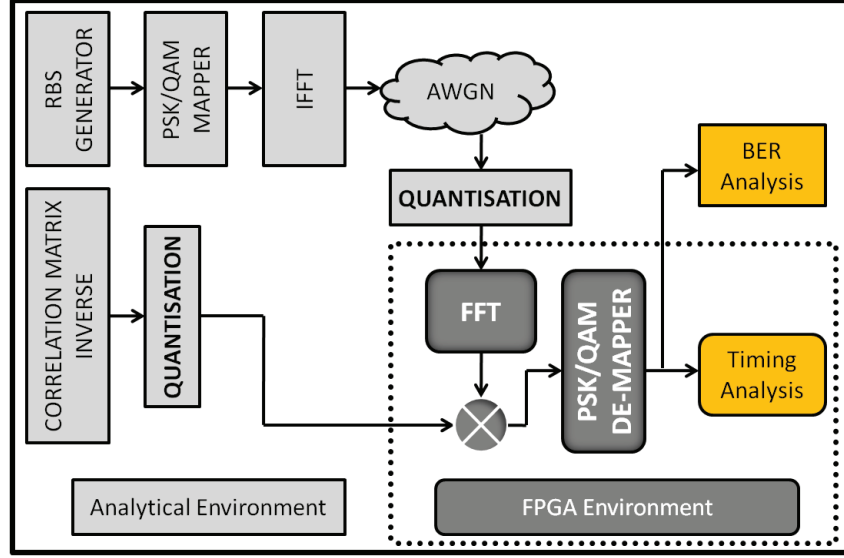


Figure 6.2: Top-level system diagram.

3. Having converted the data to two's complement numbers, as required by the Xilinx FFT IP core, the remaining operations take place in hardware to recover the original data symbols.

Fig. 6.3 illustrates the tasks carried out in hardware and the equivalent mathematical operation corresponding to each task. From Fig. 6.3, the function of the different FPGA blocks and thus the steps required for the detection of an SEFDM signal using the TSVD approach are identified as follows:

- **Internal memory:** Stores the received and quantised noisy SEFDM signal samples $r(t)$.
- **Read Only Memory (ROM) unit:** Stores the C_ξ entries.
- **FFT module:** Generates the statistics vector \mathbf{R} by correlating the incoming signal with the conjugate sub-carriers. Recalling from Section 4.2, this is equivalent to applying an FFT operation to the received samples $r(t)$.
- **TSVD unit:** The received statistics vector \mathbf{R} is fed to a detector unit to find estimates of the transmitted symbols. This is equivalent to carrying out a complex

multiplication between the FFT output values and the \mathbf{C}_ξ matrix entries. The matrix-vector product outputs are then accumulated over N cycles to obtain the symbol estimates $\hat{\mathbf{s}}_{TSVD}$.

- **PSK/QAM unit:** A decision element is used to demap the final outputs and recover the original bit stream.

The FFT block carries out the operations sequentially, followed by multiple instances of a complex multiplier and an accumulator configured in parallel. The \mathbf{C}_ξ matrix is transposed to facilitate the correct arithmetic operations on each clock cycle. This process may be better understood by referring to Fig. 6.3, where each FFT output value is multiplied by all values of the corresponding matrix row on every cycle (real and imaginary components separately) giving the equivalent result to a standard matrix arithmetic operation after N cycles. The system latency is directly proportional to the number of sub-carriers employed. Recalling that the number of sub-carriers also dictates the minimum FFT size, this infers that the value of N has an impact on the FPGA resource utilisation too.

6.3.1 Design Considerations

An FPGA design has to strike a balance between a number of important and often conflicting objectives. This work in particular, which has been reported in related articles [152] [178], takes into account the following parameters:

- **Granularity:** In this work, the granularity of the system refers to the granularity of the bandwidth compression factor α . Recalling from Section 4.5, the larger the DFT size, the higher the granularity of α . However, a larger DFT or FFT module results in an increased resource utilisation and latency.
- **Computational complexity:** This is defined by the total number of arithmetic operations that take place in the overall signal chain and dictates the number of multipliers and logic slices required to execute these operations.

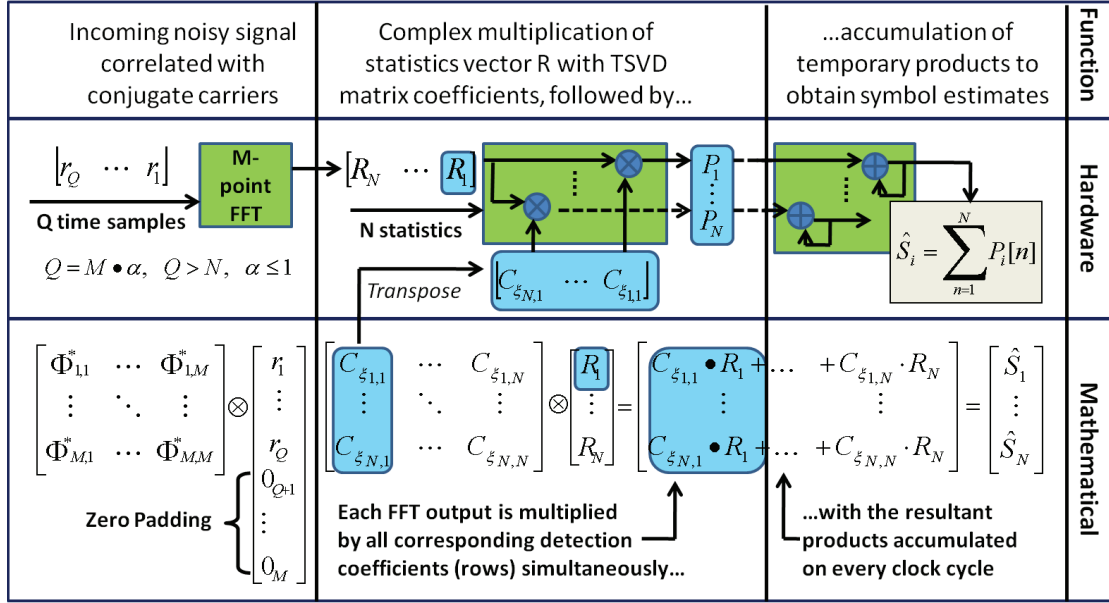


Figure 6.3: Hardware operation of the TSVD-SEFDM detector.

- Resource utilisation:** The design choices and the type of architecture (fully-parallel, fully-pipelined or mixed) affect the overall resource utilisation.
- Maximum clock frequency:** This depends on the critical path delay, which in turn is affected by the clock loading.
- Latency and throughput:** The latency of the system is measured in terms of the number of clock cycles required to process the data. This latency in conjunction with the achievable clock frequency determines the maximum theoretical throughput.
- Reconfigurability:** This refers to the ability of the architecture to modify its configuration dynamically on the fly without having to reprogram the device, thus adapting to user specifications, reflected in the system parameters, as well as environmental conditions, such as noise.
- Scalability:** This metric defines the flexibility in expanding the architecture, for example migrating from a single channel to a multiple channel architecture,

without requiring a complete overhaul of the initial design.

- **Ease of implementation:** This is purely a development issue and assesses the time and expertise necessitated to take a design from concept to fruition.

The rule of thumb for FPGA designs is small is best. This is attributed to the ‘fine-grain’ structure of FPGA devices, contrary to other programmable logic devices, such as Complex Programmable Logic Devices (CPLDs), which are ‘coarse-grain’ [189]. Small blocks are better distributed in the fabric, thus favouring more efficient routing, which results in smaller delays including the critical path delay.

Bit Precision

As explained in Section 6.1, the critical path delay is defined as the longest path between a set of registers amongst all FPGA blocks. Consequently, it should be evident that a smaller bit width will allow a higher clock frequency to be achieved [189]. Conversely, the clock frequency and latency specifications influence the bit precision choice [183]. In this work, 16 bits are initially chosen to represent the noisy data samples at the input of the receiver’s demodulator. This decision is primarily based on the fact that the FPGA transmitter implemented in [59] employs 16-bit D/A converters. This choice is further supported by referring to the work of other authors [186] [187] [190] all of whom select 16 bits to represent the data values. An additional yet equivocal point is that the default bit width value in the prebuilt blocks provided by the COREGEN tool is 16 bits, as it appears, for example, during the configuration of the complex multiplier block [169].

DFT Core versus FFT Core

As explained in Chapter 4, the DFT techniques developed in [130] for the modulation and demodulation of SEFDM signals alleviate the need for a bank of modems. From a practical perspective, if the conventional GS or Löwdin methods developed in [129]

Table 6.1: DFT core versus FFT core.

Transform Type	Max Frequency	Initial Latency	Pipelined Latency	DSP Slices	LUT-FF pairs
256-FFT	405 MHz	343 cycles	256 cycles	11%	67%
192-DFT	275 MHz	377 cycles	377 cycles	6%	55%

are employed, a new basis has to be generated each time the value of N or α is altered. In contrast, the DFT techniques can automatically accommodate the change in N or α by modifying the number of inputs and outputs considered. The DFT size may also be varied accordingly sacrificing granularity for increased throughput or vice versa.

The COREGEN tool offers both DFT and FFT IP cores. For the latter, it is well-known that the FFT sizes supported have to be a power of two. For the former, the specific core supports only DFT sizes which are multiples of 12 [184]. Recalling from Section 4.5, a conventional DFT requires N^2 multiplications and additions for N sub-carriers, whereas the complexity of an FFT is only slightly faster than linear. This complexity difference is reflected in Table 6.1 which clearly demonstrates that the Xilinx FFT IP core outperforms the Xilinx DFT IP core. In general, different digital signal processing algorithms offer diverse trade-offs when targeted for FPGA implementation [191].

The transforms in Table 6.1 have been synthesized on an XC5VFX100T device and in this case the latency is defined as the number of cycles between the first element of the input data and the last element of the output data. Here, the best FFT architecture option available, being the pipelined version which offers maximum throughput at the expense of highest resource utilisation, has been employed. As shown in Table 6.1, the FFT core requires almost double the amount of DSP Slices compared to the DFT core, nonetheless, this comes with the benefit of a higher achievable clock frequency and a lower pipelined latency. Furthermore, when OFDM is employed with no oversampling, the 256-point FFT can support a larger number of sub-carriers compared to the 192-point DFT. Conclusively, the FFT module offers significant throughput gains compared

to the DFT module with comparable slice utilisation (LUT-FF pairs) albeit a modest increase in the required number of DSP Slices. The practicality of the DFT core is therefore questionable. Two notable applications, however, come to mind; first, this core meets the LTE requirements in terms of the frame structure and timing specifications [184]; second, it may prove to be useful in this work if intermediate point sizes are sought. For example, if $N = 32$ and $\alpha = \frac{4}{5}$, then the minimum DFT size required is $\frac{N}{\alpha} = 40$. The smallest FFT size able to meet this requirement is $N_{FFT} = 64$. Conversely, the DFT core offers a number of intermediate options, such as $N_{DFT} = 48$ and $N_{DFT} = 60$. Thereby, it may be the case that in certain applications a DFT core is preferred over an FFT core.

FFT Core Architecture Options

The architecture of the Xilinx FFT IP core can be either pipelined or burst, trading resources with transform time. The pipelined architecture achieves continuous data processing, thus maximum throughput, by pipelining multiple Radix-2 butterfly engines. In this architecture, three tasks are carried out simultaneously, namely loading input samples for the next data frame, performing calculations on the current data samples and unloading the results of the previous samples computed. Consequently, this architecture results in the highest resource utilisation and highest initial latency with the benefit of performance optimisation. Burst architectures, which employ Radix-2 or Radix-4 decompositions, load and process data separately in an iterative manner. Hence, the resource utilisation is lower at the expense of a longer transform time, which is increased by approximately N_{FFT} with respect to a pipelined architecture. In general, “each architecture offers a factor of two difference in resource utilisation from the next architecture” [171].

The IP gives the option of reconfiguring the transform size in real-time on the fly on a frame-by-frame basis. Subsequently, the bandwidth compression level, the oversampling factor and even the number of sub-carriers considered, which is determined by

the amount of zero padding applied, may be adjusted on a frame-by-frame basis. With regard to data processing, there are two parts to the FFT core, namely the complex multipliers and the butterfly arithmetic. The 4-multiplier structure utilising DSP Slices offers the best performance while the 3-multiplier structure trades off performance for resource optimisation. If there is a lack of DSP Slices then standard CLB's may be employed instead to implement the core. Finally, the module offers two quantisation options, namely truncation and convergent rounding.

While this work utilises the FFT block provided by Xilinx, alternative DFT frameworks tailored for implementation on an FPGA device have been proposed. One example is the sliding DFT [183] which offers high throughput thanks to increased parallelism. A key benefit of the sliding DFT is that it does not require all input data to be present at once to compute the transform, thus significantly reducing latency at the cost of resource utilisation. Architectures for the implementation of the Discrete Fractional Fourier Transform using FPGAs have also come to light recently [192].

To conclude this section, the impact of real or complex data on the performance of the DFT is discussed. In theory, real-input DFTs require approximately half the computation time of complex-input DFTs [91]. For the hardware implementation presented here, this is not the case and the latency is identical irrespective of whether real or complex data is fed at the input. This infers that using complex-valued input data, such as 4-QAM, would be preferred over real-valued input data, such as BPSK, if the target is throughput optimisation.

Memory Architecture for the Storage of Matrix Elements

The structure of a ROM is arranged as a column vector of size $D_{ROM} \times W_{ROM}$ where D_{ROM} is the depth of the ROM denoting the maximum number of words that can be stored and W_{ROM} denotes the width of the ROM in terms of the number of bits used to represent the data values. Here, each data value corresponds to a single matrix element where the matrix is either the straightforward inverse \mathbf{C}^{-1} or the approximate

TSVD inverse \mathbf{C}_ξ . If the number of sub-carriers is N then the total number of matrix elements is $2N^2$ since the matrices consist of complex-valued data. Subsequently, the matrix elements may be stored in block memories on the FPGA device using one of the following methods:

- **Two $N \times N\beta$ ROMs:** The matrix is split into two sub-matrices both made up of real-valued data only and correspond to the real and imaginary elements of the original matrix. Each matrix element is represented in two's complement format using β bits precision. Considering that each sub-matrix is of size $N \times N$, it should be evident that $D_{ROM} = N$ and $W_{ROM} = N\beta$. Hence, each matrix element is accessed by selecting a specific ROM address and row bit range.
- **Multiple ($2N$) $N \times \beta$ ROMs:** In this case, the original matrix is split into multiple column vectors using separate ROMs to store each vector. Using this configuration, each matrix element is accessed by selecting a specific ROM block and ROM address.

In this work, the first method employing two $N \times N\beta$ ROMs is adopted. The main reason for this choice is one of development time and effort. In this case, the COREGEN tool only needs to run twice to create the two ROMs required for storing the matrix elements. The matrix elements of each row can then be accessed directly by tying the corresponding bits to a specific register and without having to worry about synchronisation. Conversely, for the second method, the COREGEN tool has to run $2N$ times to store all matrix elements in corresponding ROM blocks, hence protracting the development cycle. Furthermore, additional control logic is required to synchronise all ROM blocks and to ensure the correct memory address is accessed on the relevant clock cycles.

The second method fits better with general FPGA design rules which state that smaller blocks are preferable in FPGA devices due to their fine-grain structure. Using multiple ROMs also means that the blocks can be more flexibly spread out in the

fabric. Notwithstanding, the second method does result in a higher resource utilisation as $2(N-1)$ additional blocks are employed in the design with $D_{ROM} \times W_{ROM} = 18,432$ bits being the lowest memory primitive for a Xilinx Virtex 5 device [193].

Technical Challenges

Matrix computations are not a straightforward process in VHDL as in MATLAB. Managing arrays in VHDL is a non-trivial task and typing rules have to be strictly obeyed with signals, variables and numerical range carefully defined. Real numbers have to be converted to *standard logic vector* format, since the former are not synthesizable. A key drawback encountered in VHDL is the limited dynamic range of integer numbers compared to other mathematical tools, restricting the maximum achievable bit precision. To this end, a number of functions were created with the aid of MATLAB to simplify the hardware design of SEFDM receivers:

- **Function for converting decimal values to two's complement numbers:**

The MATLAB function *dec2bin* can only convert unsigned integer numbers to their binary equivalent. A custom function was therefore created to compute the two's complement of any integer number, defined as follows

$$\chi_B = \begin{cases} dec2bin(\chi, \beta) & \text{if } \chi \geq 0 \\ dec2bin(2^\beta - abs(\chi), \beta) & \text{if } \chi < 0 \end{cases} \quad (6.1)$$

where χ is the decimal number to be converted, χ_B is the two's complement equivalent and β is the word length in bits.

- **Function for converting integers to fixed-point fractions:** The Xilinx FFT C model requires the input values to be entered in fixed-point fractional format with the numbers being in the range $-1.0 \leq data < 1.0$ [170]. Having converted the decimal values to their two's complement equivalent, the corresponding frac-

tional values are computed according to the following formula

$$\chi_F = \frac{\chi}{abs(\chi)} \sum_{k=2}^{\beta} 2^{-(k-1)} \chi_B(k), \quad (6.2)$$

where χ_F is the fractional equivalent of the two's complement number χ_B and k denotes the bit index.

- **Function for reordering the bits in the required ROM format:** As outlined in Section 6.3.1, all matrix elements per row are grouped together to allow a single ROM to be employed rather than one memory block per column. In hardware terms, this is achieved by concatenating all the bits per matrix element per row into a single bit sequence which in turn constitutes a single ROM word.

Having reviewed the design considerations involved when devising FPGA architectures, Section 6.3.2 looks at the real-time FPGA based implementation of an SEFDM receiver comprising an FFT demodulator and a linear detector capable of performing both ZF and TSVD.

6.3.2 Implementation and Real-time Verification

The SEFDM hardware receiver, details of which are published in related work [179], is realised on an FPGA device with a flexible and reconfigurable design supporting different system sizes, oversampling factors, modulation orders and levels of bandwidth compression while providing a theoretical data rate of up to 136.8 Mbps. A high-level representation of the complete hardware system is illustrated in Fig. 6.4. Screenshots illustrating the real-time operation of this FPGA implementation are provided in Appendix B.

The received time samples $r(t)$ are transferred from the computer terminal to the FPGA via a JTAG interface. The matrix entries of \mathbf{C}^{-1} and \mathbf{C}_ξ are stored in ROM units at design time.

The demodulator stage consists of a single FFT block which generates the statistics

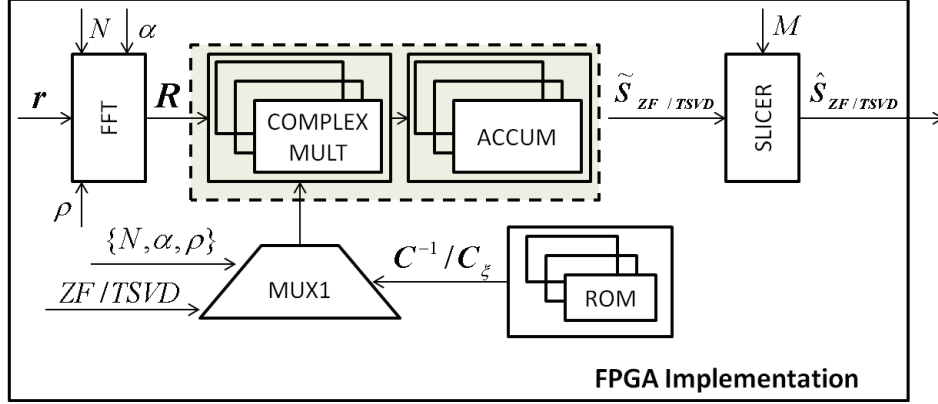


Figure 6.4: System block diagram for the FPGA based linear detector implementation.

vector \mathbf{R} by correlating the received time samples $r(t)$ with the conjugate sub-carriers, the latter operation being equivalent to an FFT function. The FFT block is run-time configurable and can change size on a frame-by-frame basis. The maximum FFT size N_{FFT} is determined by the highest number of sub-carriers N that needs to be supported by the system, the highest oversampling factor and the smallest value of α . It may be computed as follows

$$N_{FFT} = 2^{\lceil \log_2(\rho_{max} N_{max} / \alpha_{min}) \rceil}. \quad (6.3)$$

Recalling from Section 4.4, $\alpha = \frac{Q}{N_{FFT}}$. The level of bandwidth compression can therefore be configured by altering the number of output samples Q at the transmitter. Moreover, the size of the system may be changed by switching the desired number of sub-carriers *on* or *off* via zero-padding. The relationship between Q and N is determined by the oversampling factor ρ since $Q = \rho N$. Jointly, N , α and ρ serve as the control signals for the FFT block.

Subsequently, the received statistics vector \mathbf{R} is fed to a linear detector unit. This unit comprises N parallel instances of a complex multiplier followed by $2N$ parallel instances of an accumulator. The combination of these MAUs effects a complex matrix multiplication. The second input to the detector unit originates from a MUX which is connected to all the ROM blocks holding the \mathbf{C}^{-1} or \mathbf{C}_ξ matrix entries. Each

ROM block stores the matrix values corresponding to a specific combination of N , α and ρ , denoted as the set $\{N, \alpha, \rho\}$. This set is particular to the detection method employed being ZF or TSVD. Thus, both ZF and TSVD estimates can be generated for different system sizes and bandwidth compression levels by controlling the input signals to MUX1. This may be better comprehended by recalling that the TSVD estimated symbols are generated by multiplying the received statistics vector \mathbf{R} with the TSVD based pseudoinverse of \mathbf{C} . Hence, if \mathbf{R} is multiplied by the inverse of \mathbf{C} instead, a ZF detection has been performed on the incoming signal. This provides proof of the fact that the proposed hardware design is flexible and can accommodate multiple detection schemes with minimal hardware alterations.

The final stage is a slicing block which acts as a decision element recovering estimates of the original transmitted symbols. The size of the constellation cardinality M is the single control signal to the slicing unit and in this work it has been designed to support BPSK and 4-QAM modulation schemes.

All processing blocks have been configured for optimum latency with the aid of pipelining to improve timing, thus ensuring a high maximum clock frequency. DSP Slices have been used throughout the design since they favour high clock rates [194] and are deemed paramount for digital signal processing applications [195]. Alternatively, the design may be easily modified to utilise slice logic in the form of LUTs and FFs instead.

In conclusion, this section demonstrated that the FPGA based SEFDM receiver is highly flexible supporting different configurations of system sizes, oversampling factors, modulation orders and bandwidth compression levels, as well as a choice of linear detection techniques. The implementation is also run-time configurable enabling the user to make changes on the fly without the need for resynthesis. The system can be easily redesigned or upgraded with minimal hardware alterations and can operate with OFDM signals since the design employs standard blocks. The MAU unit highlighted in a grey dashed box would be bypassed in such a scenario with the FFT outputs being

directly fed to the slicer. This may be better comprehended by taking into account that the $N \times N$ sub-carriers correlation matrix \mathbf{C} for an OFDM system is equal to the $N \times N$ identity matrix.

6.3.3 Numerical Results

In this section, the performance of the FPGA based linear detector is evaluated in terms of error performance, throughput and resource utilisation. Recall that the SEFDM transmitter and AWGN channel in this work are modelled using mathematical tools. These tools are also responsible for generating the \mathbf{C}^{-1} and \mathbf{C}_ξ matrix entries, which are subsequently scaled, quantised and converted to two's complement numbers to store them in ROMs.

This section consists of two parts. The initial part presents preliminary results in terms of BER for the floating-point version of the TSVD algorithm versus the fixed-point model. This model is synthesized on a Xilinx Virtex 4 XC4VFX12 chip, however, the design is not targeted for real-time implementation. The aim of this preliminary work is to demonstrate that the TSVD algorithm can be realised on an FPGA, attributed to its low computational complexity. The error performance for a floating-point implementation of a ZF detector is also provided for comparison purposes.

The second part presents the results for a true, real-time implementation of the FPGA based SEFDM receiver. This phase involves real-world hardware data exchange to validate the receiver architecture and algorithm functionality. For this purpose, a development board equipped with a Xilinx Virtex 5 XC5VFX100T device was procured from Sundance Multiprocessor Technology Ltd., hence the design is targeted for this chip.

The performance of the SEFDM detector realisations incorporating TSVD was tested under two different modulation schemes, namely BPSK and 4-QAM. For the fixed-point model synthesized on the XC4VFX12 chip, numerical simulations were carried out in similar conditions as those adopted in [156] to assess the validity of the

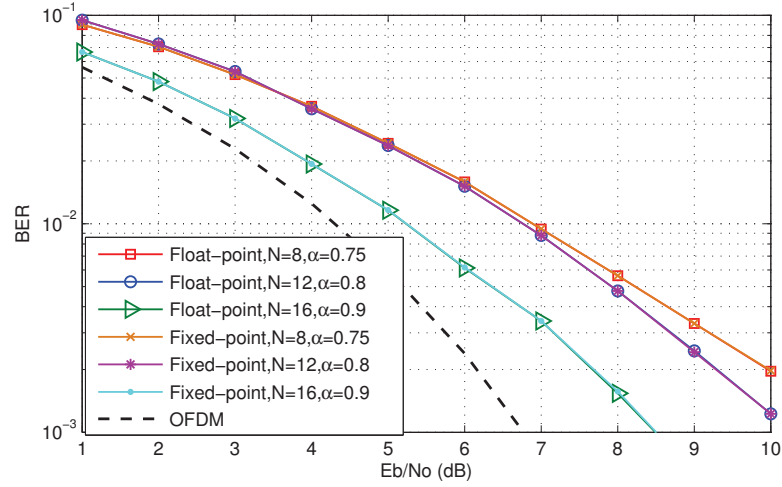


Figure 6.5: BER performance of the TSVD detector for a BPSK-SEFDM system with different sizes and values of bandwidth compression α .

fixed-point implementation, whereby the oversampling factor was set to $\rho = 4$. Each configuration employed a different set of parameters by varying the values of N and α whilst evaluating the BER for 10,000 transmitted SEFDM symbols in an AWGN environment. In line with the explanation given in Section 6.3.1, the received noisy data samples were quantised using 16 bits. A higher bit precision was used for the matrix entries, specifically 24 bits, to avoid potential saturation issues as discussed in Chapter 5.

Figs. 6.5-6.7 show the BER as a function of E_b/N_0 for the fixed-point, suboptimum TSVD detector. From Figs. 6.5 and 6.6, it is clear that the results for the fixed-point implementation match those for the floating-point version. These results are expected since the hardware design has not been optimised in terms of resources, meaning that the bit precision used to represent the data samples is much higher than that required to guarantee the same system functionality.

Fig. 6.5 illustrates that the detector can achieve a bandwidth saving of $\frac{1}{0.9} \Rightarrow 11\%$ when employing 16 sub-carriers to achieve an acceptable BER performance. Fig. 6.5 also indicates that TSVD performs well even when the system size is reduced to half the number of sub-carriers whilst the bandwidth saving is more than doubled at the

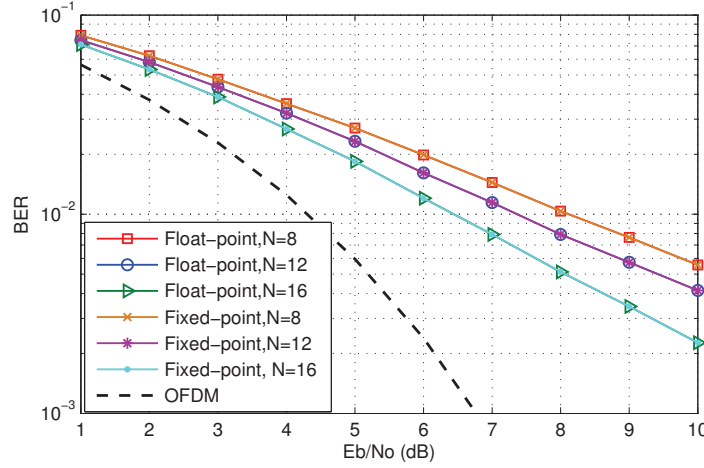


Figure 6.6: BER performance of the TSVD detector for a 4-QAM-SEFDM system with different sizes and a bandwidth compression of $\alpha=0.9$.

same time ($N=8$, $\alpha=0.75$).

Fig. 6.6 shows that the TSVD detector performs well even for higher modulation orders with performance gains proportional to the number of sub-carriers employed. Finally, Fig. 6.7 demonstrates that the TSVD detector outperforms the ZF detector, especially for bandwidth compression levels greater than 10%.

Error Performance

The error performance of the true hardware implementation using an XC5VFX100T chip is evaluated by comparing experimental results to existing computer based, floating-point simulations. All results were acquired assuming the same AWGN channel conditions for a range of E_b/N_0 values, however, in this case, only 1000 SEFDM symbols were considered due to the significantly extended time duration of the corresponding FPGA simulations in ModelSim. Furthermore, both the received SEFDM time samples and the matrix entries were quantised using eight bits for two reasons; first, preliminary experiments indicated that this bit precision provided an acceptable level of accuracy with a negligible impact on error performance; second, every effort was made to jointly optimise the design in terms of performance and resource utilisation. This mandated

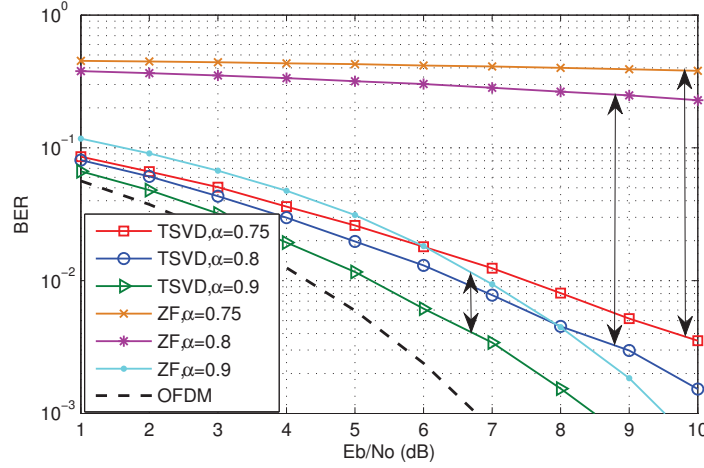


Figure 6.7: BER performance of the floating-point TSVD vs. ZF detector for a BPSK-SEFDM system with 16 sub-carriers at different values of α .

the use of DSP Slices for all arithmetic operations. Taking into account that certain cores, for example, the Xilinx Accumulator IP core, support the use of DSP Slices for a maximum word length of 48 bits, it is evident that the bit precision at the receiver's inputs has to be less to accommodate the gradual bit growth triggered by complex multiplications and accumulations. It must be noted that the levels of bandwidth compression presented in the following figures correspond to the actual values that can be obtained using the current system configuration according to the formula $\alpha = \frac{Q}{N_{FFT}}$. For example, the closest practical value to a bandwidth compression of $\alpha = 0.8$ would be $\alpha = \frac{52}{64} = 0.8125$.

Figs. 6.8 and 6.9 depict the BER as a function of E_b/N_0 for the real-time FPGA based TSVD detector under BPSK and 4-QAM modulation schemes. Both figures illustrate that the hardware implementation provides comparable performance to the analytical, floating-point results with a slight performance degradation for higher E_b/N_0 regimes ($E_b/N_0 > 5dB$). Fig. 6.8 in particular, demonstrates that the FPGA results are in accordance with the theoretical results for different configurations of N and α , thus verifying the correct functionality of the system. Further evidence is provided in Fig. 6.9, where all system sizes considered specifically in this work yield similar results

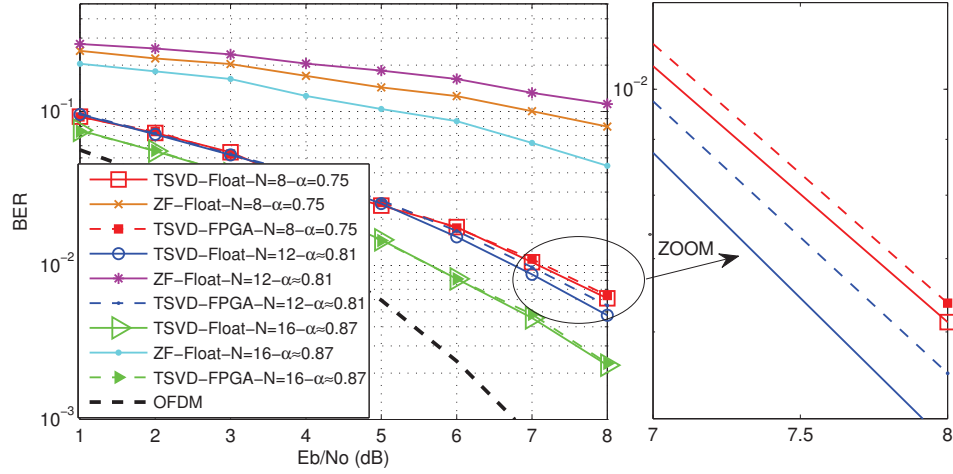


Figure 6.8: BER performance of the FPGA based TSVD detector employing BPSK for a varying number of sub-carriers N and bandwidth compression α .

for the FPGA implementation versus the theoretical simulation.

Figs. 6.8 and 6.9 illustrate that the error performance for the hardware implementation is slightly degraded for higher SNR regimes. This may be attributed to the fact that for low SNR regimes, the random noise, which has been added to the data samples as the signal propagates through the AWGN channel, essentially acts as dithering. Thereby, the effective quantisation of the received noisy samples is improved, as observed by Krone [174] who examined the capacity of communication receivers in AWGN channels. Conversely, for higher SNR regimes, quantisation noise is more prominent, hence deliberate noise would have to be added to yield similar performance gains.

Resource Utilisation and Throughput

Table 6.2 summarises the features for the Xilinx Virtex FPGA devices considered in this work [196]. It is evident that the Virtex 5 family offered a huge improvement over the previous Virtex 4 devices. The Virtex 6 family approximately doubled the available resources compared to the Virtex 5 family while the latest Virtex 7 series seem to be

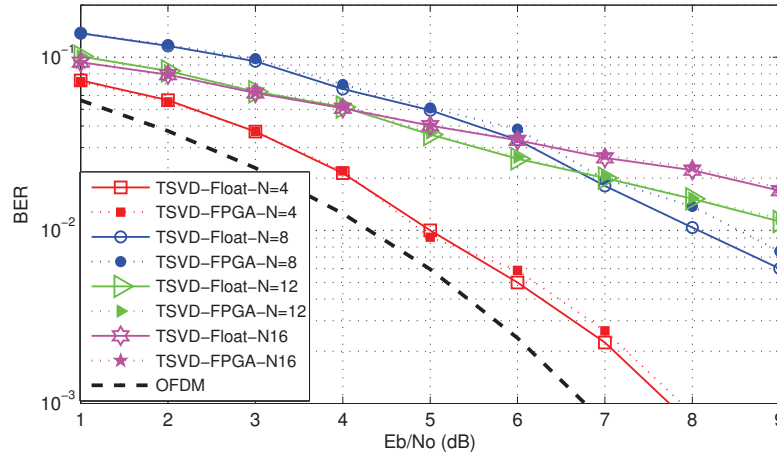


Figure 6.9: BER performance of the FPGA based TSVD detector employing 4-QAM for a varying number of sub-carriers at a fixed compression $\alpha = 0.81$.

well-positioned to revolutionise the capabilities of FPGA devices⁶. This observation is based on the availability of the considerably higher number of DSP Slices and memory blocks available on Virtex 7 chips.

Table 6.3 shows the resource utilisation of the initial FPGA design for the TSVD-SEFDM receiver targeted for the XC4VFX12 device. During these early design attempts, the system was configured to process a maximum of 16 sub-carriers with $\alpha_{min} = 0.75$ and $\rho \geq 2$, hence yielding a maximum FFT size $N_{FFT_{max}} = 64$. At first, it may appear that the resource utilisation is high with almost three quarters of the available slices and BRAMs utilised. This is due to a number of reasons. First, Table 6.3 reveals that only a third of the DSP Slices have been used. This is due to the fact that the particular design under consideration was configured to use primarily FFs and LUTs instead of embedded multipliers to prove that a cost-effective realisation of the TSVD receiver is feasible. Second, complex multipliers were implemented using a fully-combinatorial design to observe the impact of this configuration on resource

⁶As mentioned in Section 6.2, the continuation of this work is carried out by Tongyang Xu using the ML605 development board procured from Xilinx and equipped with a Virtex 6 XC6VLX240T chip. The choice for the Virtex 7 chip illustrated in Table 6.2 is based on the fact that the latest VC709 development board by Xilinx employs a Virtex 7 XC7VX690T device.

Table 6.2: Comparison of features for different Xilinx Virtex FPGA devices.

Device	Logic Cells	Slices	DSP Slices	18k BRAMs	36k BRAMs
XC4VFX12	12,312	5,472	32	36	-
XC5VFX100T	102,400	16,000	256	456	228
XC6VLX240T	241,152	37,680	768	832	416
XC7VX690T	693,120	108,300	3,600	2,940	1,470

Table 6.3: Resource utilisation for the TSVD-SEFDM hardware design using an XC4VFX12 chip.

Metric	Used	Available	Utilisation
Slices	4,068	5,472	74%
Flip-Flops	6,330	10,944	57%
4-input LUTs	5,051	10,944	46%
BRAMs	27	36	75%
DSP Slices	12	32	37%

utilisation. Third, as outlined in Section 6.3.3, a high bit precision was used to represent the matrix entries, hence necessitating the use of multiple 18k BRAMs. Finally, a number of strobe signals were integrated in the architecture to monitor the design’s functionality at various stages taking up valuable FPGA resources. In a practical implementation, these strobe signals would not be present, thus reducing the overall slice utilisation.

With this initial hardware design in mind, a new design optimised for performance and targeted for implementation on an XC5VFX100T was devised. The modules in the design were synchronised using the on-board 50 MHz oscillator. This frequency was doubled internally in the FPGA using a Digital Clock Manager (DCM) module. The synthesis results reported that the design was capable of running at over 250 MHz. Nonetheless, due to the oscillator available, the implementation was tested at 100 MHz.

Table 6.4 lists the theoretical maximum data rate R_b for different system configurations. These rates were computed based on an estimated maximum clock frequency

Table 6.4: Theoretical maximum data rate for different system sizes and modulation orders.

# Sub-carriers	FFT size	BPSK	4-QAM
$N = 4$	$N_{FFT} = 32$	34.2 Mbps	68.4 Mbps
$N = 8$	$N_{FFT} = 32$	68.4 Mbps	136.8 Mbps
$N = 12$	$N_{FFT} = 64$	51.3 Mbps	102.6 Mbps
$N = 16$	$N_{FFT} = 64$	68.4 Mbps	136.8 Mbps

of $f_{clk} = 273.673$ MHz, as reported by the synthesis results and according to

$$R_b = \frac{N \cdot \log_2 M}{N_{FFT}} \cdot f_{clk}. \quad (6.4)$$

Since the system was configured to support a maximum of $N_{max} = 16$ sub-carriers with $\rho_{max} = 2$ and $\alpha_{min} = \frac{40}{64} = 0.625$, the maximum FFT size required is found to be $N_{FFT} = 64$. Consequently, the maximum achievable data throughput at the running frequency of 100 MHz is 50 Mbps.

The resource utilisation of the entire FPGA implementation is shown in Table 6.5. With the exception of DSP Slices, approximately 50% of the used resources are occupied by the module required for the communication between the FPGA and the DSP, since the DSP is used to execute the FSD algorithm as described in Chapter 7. This is particularly true for the BRAMs, as expected, since all the outputs of the FPGA based linear detector are linked to multiple FIFOs which hold the input data required during the execution of the FSD algorithm. Table 6.5 also shows that the FPGA design under consideration is signal processing heavy with almost half the available embedded multipliers utilised.

Table 6.6 shows the FPGA resources occupied by each separate block of the TSVD-SEFDM receiver. Each complex multiplier requires four DSP Slices requiring a total of 64 for the current implementation employing $N_{max} = 16$ instances of this block configured in parallel. Similarly, each accumulator requires two DSP Slices, thus a total of 32, taking into account that real and imaginary parts are accumulated separately.

Table 6.5: Resource utilisation for the complete FPGA based TSVD-SEFDM receiver system.

Metric	Used	Available	Utilisation
Slices	4,190	64,000	7%
LUTs	4,222	64,000	7%
BRAMs	36	228	16%
DSP Slices	124	256	48%
Max. Freq.	273.673 MHz		
Run. Freq.	100 MHz		
For the FPGA-DSP Communications Module in Isolation			
Slices	2,373	64,000	4%
LUTs	2,244	64,000	4%
BRAMs	15	228	7%

Table 6.6: FPGA resource requirements for each TSVD receiver block.

Component	DSP Slices	Utilisation	18k BRAMs	Utilisation
FFT	28	23%	1	3%
Complex Multipliers	64	52%	-	0%
Accumulators	32	25%	-	0%
ROMs	-	0%	20	55%
I/O FIFOs	-	0%	15	42%

It was also found that $1 \times 18k$ BRAM is required for every four sub-carriers with reference to the ROM blocks holding the \mathbf{C}^{-1} or \mathbf{C}_ξ entries. Hence, if for example only the maximum case with 16 sub-carriers was employed, the availability of BRAMs in the FPGA would allow for 16 different levels of bandwidth compression to be stored. This quantity is inversely proportional to the system size.

Table 6.7 reports the initial and pipelined latency per block, the former corresponding to the number of cycles required to fill up the pipeline path. The pipelined latency to recover a single SEFDM symbol is equal to $N + ALU_{LATENCY}$, where $ALU_{LATENCY}$ is the sum of the pipelined latency for the MAUs and slicing units. For $N = 16$, the latency is 24 cycles. The remaining 40 cycles during the execution of the FFT transform remain inactive. This is the key pitfall of the design since it does not reflect the lowest latency implementation possible as achieved for the SEFDM transmitter implemented

Table 6.7: Latency for the different FPGA blocks.

Component	Initial Latency	Pipelined Latency
64-point FFT	271 cycles	64 cycles
Reconfigurable 64-point FFT	278 cycles	64 cycles
Complex Multipliers	0	4 cycles
Accumulator	0	2 cycles

in [175]. Instead, the current design offers a balanced trade-off between the resources utilised, the area occupied and the overall system latency. While the upper bound on the maximum achievable throughput is directly determined by the system size, the pipelined configuration reduces clock loading, thereby improving timing.

6.4 Discussion of Design Architecture

The design methodology for the implementation of an SEFDM receiver using FPGAs was discussed. The initial system design demonstrated that the TSVD detector can be realised on a Xilinx Virtex 4 FPGA device. Preliminary fixed-point performance results were in accordance with existing computer based simulations in terms of BER and showed that this sub-optimum detection scheme outperforms ZF. Even though sub-optimum, TSVD has complexity which allows it to be implemented using current silicon technology and is capable of providing reasonable bandwidth savings for a relatively small system size.

The architecture was then modified and targeted for implementation on a Xilinx Virtex 5 chip. Section 6.3.2 demonstrated that the new design is highly flexible and allows the system to be reconfigured in many aspects. For example:

- The FFT IP core is run-time reconfigurable and can change transform size on a frame-by-frame basis.
- Thanks to the FFT module's reconfigurability, the amount of oversampling, bandwidth compression, as well as the number of sub-carriers can be altered dy-

namically since the changes are reflected in the FFT size according to $N_{FFT} = 2^{\lceil \log_2(\rho_{max} N_{max} / \alpha_{min}) \rceil}$.

- The architecture has the ability to switch instantly between ZF or TSVD detection by configuring the MUX's input.
- The linear detector can be bypassed entirely allowing OFDM processing to be carried out without modifying the architecture.

Taking advantage of the unique parallel processing capabilities of FPGA devices, it was demonstrated that the linear detector can achieve a maximum theoretical throughput of up to 136.8 Mbps. The implemented design occupies under 50% of the available resources, thus there is still enough margin to increase system complexity and/or optimise performance sacrificing more resources to increase throughput without exceeding the device's resource bounds.

Errors due to integer overflow and round-off procedures are very common in hardware designs. In particular, due to finite word length effects, noise is introduced during the processing of the Xilinx FFT transform block [171]. Consequently, these issues have to be carefully accounted for since they have a direct impact on the performance of any system.

6.4.1 Optimisation Techniques

The beauty of FPGA devices is in their parallel processing capabilities. The throughput of the design realised on the XC5VFX100T may be improved by adopting a multi-channel architecture, which would allow multiple networks of DSP Slices and CLBs to execute concurrently. Alternatively, the reciprocal of the design detailed in [175] could be adopted, which employs multiple, smaller transform blocks trading higher resource utilisation for lower system latency.

Examples of optimisations at the component level include replacing multipliers by 'shift and add' operations and the reuse of modules, for example, a single FFT IP core

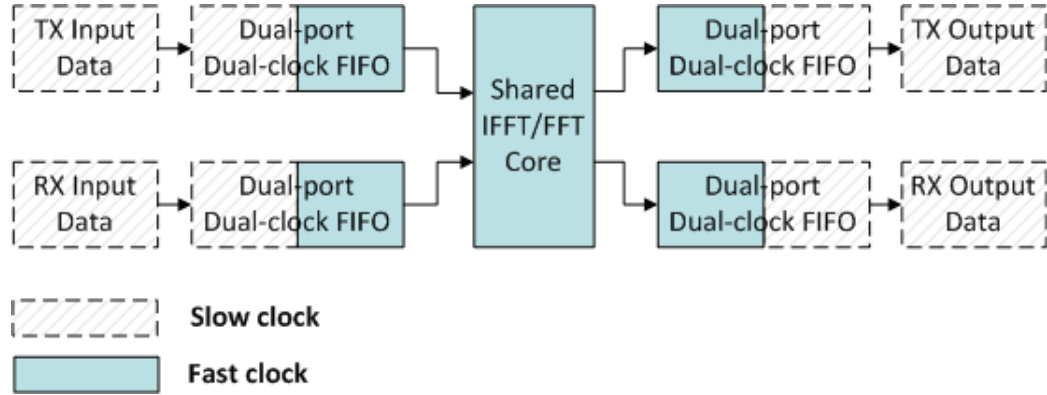


Figure 6.10: Using multiple clock domains to allow block reuse in FPGA systems.

may be configured to carry out a forward or an inverse transform on a frame-by-frame basis. Block reuse ordinarily demands the configuration of multiple clock domains, as illustrated in Fig. 6.10. The blocks which are reused must be clocked at a much faster rate compared to the symbol rate [197]. Furthermore, handshaking is required to synchronise cross-domain signals achieved with the aid of dual-clock dual-port FIFOs.

Specific to this work, optimisation of memory utilisation may be accomplished given the special properties of the sub-carriers correlation matrix and its inverse. Chapter 4 illustrated that the sub-carriers correlation matrix \mathbf{C} is Toeplitz. By definition, the maximum number of unique elements is equal to $2N - 1$ where N is the size of square matrix which in this work denotes the number of sub-carriers. The number of unique elements may be further reduced if the ratio $\frac{N}{\alpha}$ or in the more general case $\frac{Q}{\alpha}$ is an integer. Recalling from Section 4.4, \mathbf{C} may be generated by applying IDFT and DFT operations to a real-valued identity matrix. A direct consequence of applying the IDFT (or DFT) to real-valued inputs is that the output exhibits symmetry about the Nyquist frequency at $\frac{N_{IDFT}}{2}$ [91]. Consequently, the number of unique elements in \mathbf{C} , in terms of their absolute values, is equal to $\frac{\rho N}{2\alpha} + 1$. Chapter 4 also demonstrated that the inverse and approximate pseudoinverse of the sub-carriers correlation matrix, being \mathbf{C}^{-1} and \mathbf{C}_{ξ} , respectively, are bisymmetric. In this case, the maximum number of

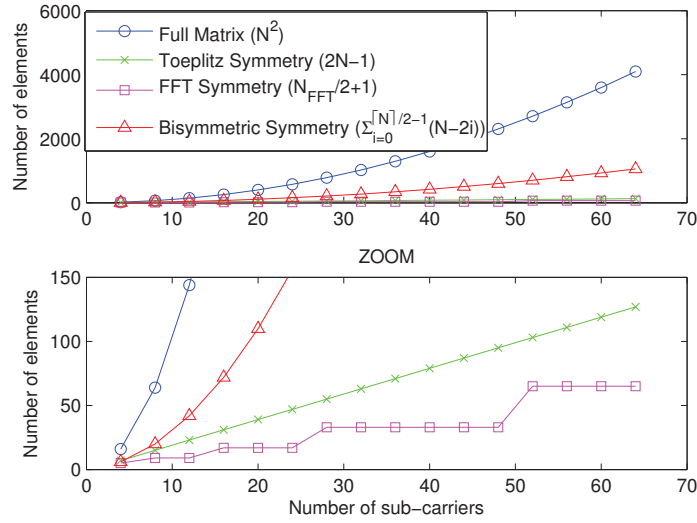


Figure 6.11: Unique elements of \mathbf{C} , \mathbf{C}^{-1} and \mathbf{C}_ξ versus the total number of their elements for different values of N .

unique elements, in terms of their absolute values, is given by

$$\sum_{i=0}^{\lceil \frac{N}{2} \rceil - 1} (N - 2i). \quad (6.5)$$

The aforementioned observations illustrate that the total number of elements that need to be stored in ROMs when targeted for hardware implementation may be significantly reduced by taking into account the special properties of these matrices, as illustrated in Fig. 6.11.

Taking advantage of these properties, a smaller number of ROMs in terms of size and/or quantity may be employed at the expense of additional control overhead. Two options are particularly feasible in this scenario:

- **Using a single β -bit ROM in conjunction with a shift register:** In this case, all the unique elements are stored in a single block memory and are queued in a shift register which in turn is connected to the inputs of the complex multiplier modules. Additional logic is required to read the relevant values from ROM

pertaining to the complex multiplication taking place on each clock cycle.

- **Storing the elements directly in registers:** This configuration obviates the need for ROMs, yet is costly in terms of logic slices since the elements are directly stored in registers. The key benefit of this method is that the elements are available instantly with zero latency and are acquired without buffering. This method still calls for additional control logic and could prove to be ill-suited for large system sizes.

To conclude, it is worth noting that in these methods, only the absolute values of the elements are stored since the sign of each element is easily manipulated by configuring the MSB which serves as a sign bit.

6.5 Conclusions

A hardware design and implementation for an SEFDM receiver employing the TSVD algorithm has been presented. While novel algorithms have been developed for the detection of SEFDM signals with promising results, these algorithms suffer from high computational complexity making them ill-suited for commercial applications. The ultimate goal of this work is to devise a platform that will facilitate the implementation of these algorithms in an end-to-end system.

The chapter commenced by explaining that the physical limiting factor of an FPGA device is the number of logic elements available. The complexity of the system directly dictates the resources required. In this work, it was established that the size of the FPGA design is directly proportional to the size of the SEFDM system. An increase in the number of sub-carriers and/or amount of oversampling, results in increased throughput and/or improved error performance, respectively, at the expense of increased area and latency.

Subsequently, the FPGA design cycle illustrated that the complexity of the system dictates the synthesis time, as well as the computation time required by the analytical

tools to run the algorithms and identify the optimum system parameters for a specific configuration. The exponential growth in development time in relation to system complexity accentuates the importance of simulation tools like ModelSim to expedite the successful realisation of any design. In terms of customisability, the COREGEN tool cuts down development effort to a great extent, however, it constrains the implementation options available to an FPGA designer, which in turn limits the potential for enhancing the performance of the final outcome.

Results showed that the latency of the system, defined as the total time delay that exists in a design, imposes an upper bound on the maximum achievable throughput. The area required by a design can be optimised by utilising fewer resources if a loss in performance is tolerable. To maintain the same or to optimise performance without taking up more area, DSP Slices have to be integrated in the system which come at a higher price. The aforementioned crucial design factors determine the amount of parallelism and/or pipelining that is incorporated in the design, which in turn are jointly considered to achieve the best clock performance and resource utilisation.

Based on the above, it was argued that a trade-off always exists between the complexity of a design and the corresponding performance. Fortunately, the programmable logic nature of FPGA devices offers the designer a great deal of freedom over the architecture of the implementation. On one end, a fully-pipelined architecture optimises resource utilisation at the expense of increased end-to-end latency. On the other hand, parallelisation is commonly used for performance optimisation at the expense of an increased design size, thus an increased cost.

Finally, it was demonstrated that the proposed FPGA architecture is highly re-configurable, attributed to the features of the Xilinx FFT IP core, thus supporting different bandwidth sizes. This is in line with the latest development in LTE systems which use carrier aggregation to support wider transmission bandwidths. The ultimate goal is the automatic and seamless integration of the entire SEFDM receiver in a single hardware wrapper similar to the platform presented in [182]. Such a platform makes

the verification process more efficient, easier to troubleshoot and less error prone while allowing algorithm designers to focus on the actual principles rather than the hardware subtleties.

The next chapter brings the work in this thesis to an end by considering more advanced detection techniques, in particular tree-search detection algorithms.

Chapter 7

Implementation of Tree-search Algorithms

Previous chapters demonstrated that the reduction in frequency distance between the sub-carriers in SEFDM systems gives rise to ICI complicating the receiver design. Linear detectors, such as ZF and TSVD are simple to implement but lead to significant BER performance degradation, particularly for complex modulation schemes like 4-QAM. Brute-force ML yields the optimum solution, yet its NP complexity prohibits its use in practical systems.

An alternative approach which avoids an exhaustive search is to use iterative tree-search algorithms, hence, this chapter commences by reviewing the basic principles of such algorithms. Section 7.1.1 discusses SD, since Kanaras showed [198] that SD provides quasi-optimum performance at significantly reduced complexity compared to ML. Subsequently, this algorithm is used for the verification of an FPGA based SEFDM transmitter.

Notwithstanding, SD has variable complexity which is dictated by the noise present in the channel, as well as the interference properties of the SEFDM system, thus rendering SD unsuitable for hardware realisation. For this reason, techniques employed in MIMO systems for fixing the complexity of SD [199], which were adapted by Isam

[156] to address the detection challenge in SEFDM, are explored in Section 7.1.2. Nevertheless, for high-dimensional systems, even these fixed complexity detectors present challenges for implementation. Hereby, the application of Sort Free (SF) and Modified Real Valued Decomposition (MRVD) techniques to SEFDM systems to yield performance gains, is explored. Section 7.2 details the joint FPGA-DSP platform while Section 7.3 describes the implementation of the proposed detector on a DSP chip. The experimental results acquired through the DSP implementation paves the way for an FPGA realisation while providing indispensable data to make informed design decisions. These outcomes are corroborated in Section 7.4, thanks to the results acquired from the recent FPGA realisation of the FSD algorithm in isolation [188].

The research in this chapter was presented at multiple conferences including the IEEE International Conference on Communications (ICC) [165] and the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC) [179], both in 2012. Part of this work has also been accepted for publication in the IEEE ICC 2013 [200] and the IEEE International Conference on Telecommunications (ICT) 2013 [188] conference proceedings and will be presented in due course. Finally, recent work detailing iterative techniques, which employ soft decision to improve the performance of SEFDM systems, has been submitted for consideration in the IEEE Communications Letters [135].

7.1 Algorithm Principles

Recall from Section 4.2.2 that the SEFDM system model is given by $\mathbf{R} = \mathbf{C}\mathbf{s} + \mathbf{Z}$ assuming that the correlation matrix \mathbf{C} is used to represent the interference between the sub-carriers. The ML detection problem is therefore formulated as

$$\hat{\mathbf{s}}_{ML} = \arg \min_{\mathbf{s} \in \mathcal{M}^N} \|\mathbf{R} - \mathbf{C}\mathbf{s}\|^2, \quad (7.1)$$



Figure 7.1: Sphere search constrained by the radius g .

indicating that ML is an NP hard combinatorial problem [129] with a complexity order $\mathcal{O}(M^N)$ which grows exponentially with the increase in the number of sub-carriers N and/or the constellation size M . Consequently, ML detection suffers from an impractical computational complexity even for moderate system sizes.

7.1.1 The Sphere Decoding Algorithm

The most popular tree-search algorithms encountered in MIMO systems and subsequently in SEFDM systems are SD and FSD along with their different variants. SD is commonly referred to as lattice decoding based on the description by Fincke and Pohst [201] who first proposed the algorithm as a method for identifying the minimum Euclidean distance in a given lattice, the latter term used to designate the representation of vectors in a Euclidean space [129].

The SD algorithm solves the LS problem subject to an additional constraint. Specifically, the value of the squared Euclidean norm $\|\mathbf{R} - \mathbf{C}\mathbf{s}\|^2$ has to be less than or equal to a predefined value g , expressed as

$$\|\mathbf{R} - \mathbf{C}\mathbf{s}\|^2 \leq g, \quad (7.2)$$

where $\mathbf{C}\mathbf{s}$ represents the lattice points. Thereby, SD constrains the search to only the candidate vector symbols \mathbf{s} for which the points $\mathbf{C}\mathbf{s}$ lie in an N -dimensional hypersphere centred around the received statistics vector \mathbf{R} with radius g , as illustrated in Fig. 7.1. Evidently, the radius dictates the search space and thus the complexity of the algorithm with the aim of converging to the optimum solution much faster compared to ML.

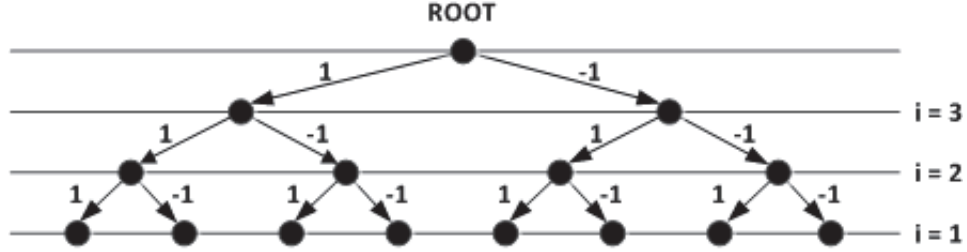


Figure 7.2: Search tree for a SD with $N = 3$ and BPSK modulation.

The SD algorithm is executed by spanning a tree of depth $N + 1$, as illustrated in Fig. 7.2. Each point in the tree is referred to as a node with the total number of nodes given by $\sum_{i=0}^N M^i$. The number of branches per node is equal to the constellation size, hence the total number of nodes present at each level is given by M^{N+1-i} . The nodes at level $i = 1$ are referred to as the leaves of the tree.

The SD algorithm will always converge to the ML solution provided that the sphere illustrated in Fig. 7.1 contains at least one candidate vector symbol. However, the fundamental point of using SD instead of ML is the increased computational efficiency associated with the former thanks to complexity reduction. This reduction is achieved by reducing the number of branches and leaves considered in the search for the ML solution, in other words by pruning the tree. The most common metric used to decide whether a node should be retained or discarded is the sphere radius g . Too large a radius will result in the SD execution approaching the complexity of ML. Too small a radius may result in no solution being found.

The nodes of the tree that have not been pruned and are considered in the search are referred to as admissible nodes. There are two approaches for traversing the tree:

- **Depth-first:** This method was employed in the original SD algorithm as applied to SEFDM by Kanaras [129]. In this scheme, the algorithm starts from the root of the tree ($i = N + 1$) and proceeds in both a downwards ($i = i - 1$) and upwards ($i = i + 1$) direction. All admissible child nodes of a parent node are initially examined (in a top-down approach) before visiting the admissible siblings of a

parent node (in a left-right approach).

- **Breadth-first:** This method differs from the depth-first approach in that it is not a recursive scheme. The algorithm proceeds in a downwards direction only and visits all admissible nodes at each level i in order to construct the new set of admissible nodes for the next level ($i = i - 1$). This scheme is adopted in the FSD algorithm described in Section 7.1.2.

In both cases, when a complete path is found from the root of the tree to one of its leaves, a point is enumerated in the sphere as per Fig. 7.1. The decoded vector symbol \hat{s}_{SD} corresponding to the ML solution is the path with the smallest Euclidean distance.

Schnorr-Euchner Enumeration

From the above, it should be evident that the radius determines the number of visits to the nodes of the tree and as noted by Kanaras, “the proper definition of the initial radius is decisive for the achievement of a practical detection” [129]. A highly efficient technique for accelerating the execution of the algorithm in a depth-first approach without compromising BER performance is to dynamically reduce the radius each time the algorithm reaches a leaf. This allows the initial sphere radius to be set high to eliminate the probability of no points being found.

When the constellation points are real-valued, the children of each parent node take on values from a specific set of points, referred to as the admissible interval [129] [61]. In this case, a reordering strategy, known as Schnorr-Euchner (SE) enumeration [202], enables the algorithm to converge rapidly to an appropriate radius. Contrary to the original Fincke-Pohst enumeration [201], which enumerates the points in numerical order from the lowest bound to the upper bound of the admissible interval, the SE strategy enumerates the points in ascending order of distance from the centre of the interval in a zig-zag manner, as illustrated in Fig. 7.3. While this ordering may incur additional complexity, this complexity is much smaller compared to the computational complexity associated with the overall search in a sphere having a very large radius.

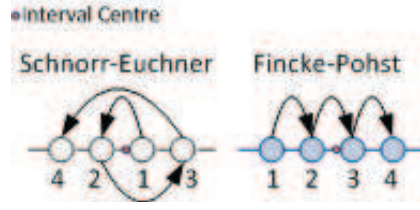


Figure 7.3: Schnorr-Euchner enumeration vs. Fincke-Pohst enumeration.

Real Valued Decomposition

The original algorithm proposed in [201] assumes that all elements are real-valued numbers. Furthermore, admissible intervals cannot be defined for complex-valued constellation points [61]. Thereby, a technique known as RVD is employed to transform the complex-valued problem to its equivalent real-valued representation. For the SEFDM system model in particular, the RVD of the system is given by

$$\begin{bmatrix} \text{Re}\{\mathbf{R}\} \\ \text{Im}\{\mathbf{R}\} \end{bmatrix} = \begin{bmatrix} \text{Re}\{\mathbf{C}\} & \text{Im}\{\mathbf{C}\} \\ -\text{Im}\{\mathbf{C}\} & \text{Re}\{\mathbf{C}\} \end{bmatrix} \begin{bmatrix} \text{Re}\{\mathbf{s}\} \\ \text{Im}\{\mathbf{s}\} \end{bmatrix} + \begin{bmatrix} \text{Re}\{\mathbf{Z}\} \\ \text{Im}\{\mathbf{Z}\} \end{bmatrix}$$

RVD can be applied only for rectangular constellation schemes, such as 4-QAM, 16-QAM, 64-QAM, with the admissible interval given by a sub-set of the original constellation set with size equal to $\mu = \sqrt{M}$. In contrast to the complex version of the system's model, the tree has $M - \mu$ fewer children per parent node, however, this comes at the expense of doubling the system dimension from N to $2N$. Consequently, the tree depth in this case equates to $2N + 1$.

Linear Detection Preprocessing

In this work, a linear detection stage precedes the SD stage. For this reason, the unconstrained ML solution given by $\tilde{\mathbf{s}} = \mathbf{C}^{-1}\mathbf{R}$ may be used as the centre of the sphere instead of the received statistics vector \mathbf{R} in order to reduce the search space of the SD algorithm [203] and thus the relevant complexity. Substituting $\mathbf{R} = \mathbf{S}\mathbf{s}$ into Eq. 7.2

yields

$$\|\mathbf{R} - \mathbf{C}\mathbf{s}\|^2 \leq g \Rightarrow (\mathbf{s} - \tilde{\mathbf{s}})^H \mathbf{C}^H \mathbf{C} (\mathbf{s} - \tilde{\mathbf{s}}) \leq g, \quad (7.3)$$

which represents a lattice centred at the unconstrained ML estimate and whose generator¹ matrix is the Grammian matrix $\mathbf{C}^H \mathbf{C}$. Here, it is worth pointing out that the unconstrained ML estimate may be obtained through either ZF or TSVD as $\tilde{\mathbf{s}}_{\mathbf{ZF}} = \mathbf{C}^{-1} \mathbf{R}$ or $\tilde{\mathbf{s}}_{\mathbf{TSVD}} = \mathbf{C}_\zeta \mathbf{R}$, respectively.

Subsequently, the search in an N -dimensional hypersphere is split into N one-dimensional consecutive searches formed as linear intervals and commonly referred to as Partial Euclidean Distances (PEDs)² [129] [61]. To allow this simplification, the generator matrix must be decomposed into an upper triangular matrix to enable the computation of the partial Euclidean norms. This decomposition may be achieved using Cholesky factorisation provided that the matrix is (conjugate) symmetric and positive-definite, the latter true when the matrix is non-singular. Hence, the generator matrix may be rewritten as $\mathbf{C}^H \mathbf{C} = \mathbf{L}^H \mathbf{L}$ where \mathbf{L} is the upper triangular matrix acquired through the decomposition $\mathbf{L} = \text{chol}(\mathbf{C}^H \mathbf{C})$ with $\text{chol}\{\cdot\}$ denoting the Cholesky decomposition function. Thereby, \mathbf{L} takes the role of the sub-carriers correlation matrix permitting Eq. 7.3 to be rewritten as

$$(\mathbf{s} - \tilde{\mathbf{s}})^H \mathbf{L}^H \mathbf{L} (\mathbf{s} - \tilde{\mathbf{s}}) \leq g \Rightarrow \|\mathbf{L}(\mathbf{s} - \tilde{\mathbf{s}})\|^2 \leq g, \quad (7.4)$$

which may equivalently be given by

$$\sum_{i=1}^N l_{ii}^2 \left| (s_i - \tilde{s}_i) + \sum_{j=i+1}^N \frac{l_{ij}}{l_{ii}} (s_j - \tilde{s}_j) \right|^2 \leq g, \quad (7.5)$$

where l_{ij} are the elements of \mathbf{L} . Eq. 7.5 represents the sum of all PEDs which form a complete path from the root of the tree $i = N + 1$ to one or more of its leaves at level

¹A generator matrix \mathbf{G} of a matrix \mathbf{A} is a matrix whose rows and columns can be transformed to generate all the elements of \mathbf{A} .

²While in this context, the linear intervals actually correspond to squared Euclidean distances, the term PED will be adopted throughout this work for clarity of presentation.

$i = 1$. In particular, the PED for each child node, denoted by d_i , is given by the sum of the PED of each parent node, denoted by d_{i+1} , and the distance increment, denoted by e_i , expressed as

$$d_i = d_{i+1} + e_i. \quad (7.6)$$

From Eq. 7.6, it should be evident that d_{i+1} represents the Accumulated Euclidean Distance (AED) from the root of the tree down to level $i + 1$. With reference to Eq. 7.5, the distance increment per level may be given by

$$\begin{aligned} e_i &= l_{ii}^2 \left| (s_i - \tilde{s}_i) + \sum_{j=i+1}^N \frac{l_{ij}}{l_{ii}} (s_j - \tilde{s}_j) \right|^2 = \left| l_{ii}(s_i - \tilde{s}_i) + \sum_{j=i+1}^N l_{ij}(s_j - \tilde{s}_j) \right|^2 \Rightarrow \\ e_i &= \left| l_{ii}s_i - \sum_{j=i}^N l_{ij}\tilde{s}_j + \sum_{j=i+1}^N l_{ij}s_j \right|^2, \end{aligned} \quad (7.7)$$

the last line being in accordance with the formulae presented in [61]. Note that each PED is subject to the same constraint as the original ML detection problem, in other words the value of each PED must be less than or equal to the sphere radius, otherwise all of its children and leaves are discarded.

Verification of an FPGA based SEFDM Transmitter using the SD Algorithm

In [165], the analytical version of the SD proposed by Kanaras [198] employing RVD was used to decode the real signals generated by the FPGA hardware transmitter. A top-level diagram of the detection process is depicted in Fig. 7.4.

The received signal is initially demodulated using a DFT generating the statistics vector \mathbf{R} . The initial unconstrained estimate is then obtained using ZF. The inputs to the SD block include the unconstrained ZF estimate $\tilde{\mathbf{s}}$, the sub-carriers correlation matrix \mathbf{C} and the initial sphere radius g , which in this case is set to an arbitrary large number. This work in particular incorporates a binary search algorithm, which initiates the process with a very large radius and rapidly determines the minimum radius required to yield a successful solution.

the sphere and therefore the complexity of the solution.

A detailed examination of the complexity and variants of the SD algorithm is beyond the scope of this work. A number of SD variants were explored by Kanaras including a Complex SD, a Regularised SD and a Pruned SD, the latter entailing Semi Definite Programming (SDP) [129]. There is also rich literature discussing these issues in MIMO systems, a notable example being the work presented by Viterbo and Boutros [204]. An authoritative account of the complexity associated with SD is given by Hassibi and Vikalo [205], as well as Jalden and Ottersten [206].

With regard to the real and complex versions of SD, though the SD employing RVD is widely used, its application is only valid for rectangular constellation schemes. Conversely, the complex version of the SD can be applied for any constellation size, however, its execution is less straightforward. The reader is referred to the work by Hochwald and Brink [207] for details on this method.

Finally, it must be underlined that Cholesky decomposition requires the matrix to be both symmetric and positive definite. Yet, for severely ill-conditioned matrices, the matrix becomes positive semi-definite and so the Cholesky method cannot be applied. Consequently, future work should consider the use of techniques like pivoting to convert semi-definite matrices into definite matrices. Alternatively, other decomposition methods, such as QR decomposition, should be explored (see for example the work by Damen and Gamal [208]), especially in view of the fact that QR decomposition offers better numerical stability compared to Cholesky decomposition [203].

In conclusion, while SD provides a notable reduction in complexity, its high sensitivity to noise and the interference properties of SEFDM systems render it inapt for hardware realisation. In addition, the sequential nature of the algorithm does not fit well with the parallel processing capabilities of FPGAs. For this reason, a variant of the SD algorithm which fixes the complexity and has an architecture which is better tailored for FPGA implementation is examined in the next section.

7.1.2 The Fixed Sphere Decoding Algorithm

Although SD is able to achieve ML performance, its variable complexity results in variable throughput. Hereby, this section considers the FSD algorithm as proposed in [156] which guarantees a constant throughput by fixing the complexity of the conventional SD algorithm employed in [198]. Furthermore, this complexity is independent of the noise present in the system, as well as the ill-conditioning of the interference matrix. There are two key differences between the SD and the FSD techniques; first, the FSD traverses the tree using a breadth-first approach; second, the number of nodes examined at each level is fixed at design time.

The FSD method described in [156] is based on similar principles to the K-Best algorithm proposed by Wong [160] in that the number of admissible nodes per level is constrained by the parameter K. Nevertheless, K-Best decoding does not take into account the SEFDM system model. The FSD detector under consideration also differs from the FCSD proposed by Barbero [161] in two aspects; first, the FCSD is based on the complex SD version; second, the number of nodes examined at each level is not fixed as in FSD. Instead, the number of points searched at each level is determined by the MIMO channel matrix ordering and tends to be higher for the first levels of the tree and lower for the last levels. Fig. 7.5 depicts the architecture of the hybrid SEFDM receiver.

The initial unconstrained estimate $\tilde{\mathbf{s}}$ is generated via a linear detector, in this work being either ZF or TSVD. This unconstrained estimate may then be fed to a slicing unit to obtain the actual symbol estimates $\hat{\mathbf{s}}_{LIN}$ as

$$\hat{\mathbf{s}}_{ZF} = \lceil \mathbf{C}^{-1} \mathbf{R} \rceil, \quad (7.8)$$

or

$$\hat{\mathbf{s}}_{TSVD} = \lceil \mathbf{C}_\xi \mathbf{R} \rceil, \quad (7.9)$$

with the slicing unit performing QAM de-mapping. Alternatively, $\tilde{\mathbf{s}}$ may be fed to a

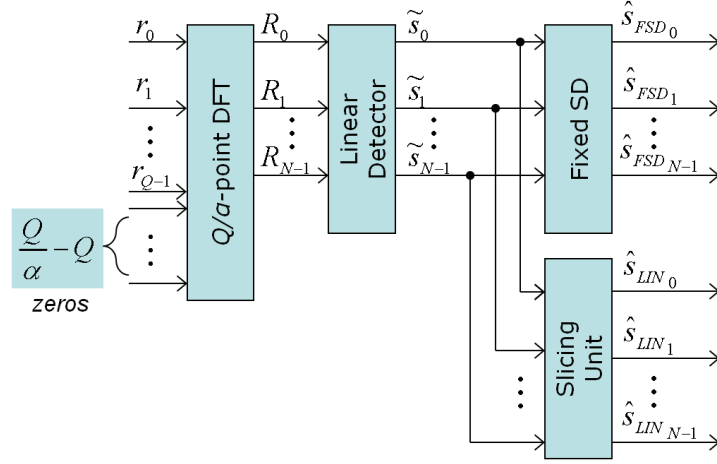


Figure 7.5: SEFDM receiver combining linear detection with fixed SD.

FSD block to yield

$$\hat{\mathbf{s}}_{FSD} = \arg \min_{\mathbf{s} \in \mathcal{H}} \|\mathbf{R} - \mathbf{C}\mathbf{s}\|^2 \leq g. \quad (7.10)$$

From Eq. 7.10, it is evident that the FSD imposes an additional constraint upon the original SD algorithm by restricting the search to a limited sub-space of the problem [156]. Rather than searching over the entire N -dimensional complex constellation set of M^N points, the FSD only searches in a sub-set $\mathcal{H} \subset M^N$ around the centre of the sphere, where the symbol \subset denotes a sub-set. The radius g is set to the sliced linear detection estimate according to

$$g = \|\mathbf{R} - \mathbf{C}\hat{\mathbf{s}}\|^2. \quad (7.11)$$

As demonstrated in [156], TSVD outperforms ZF in an AWGN environment. Taking into account that the key factors governing a successful FSD decoding are the generator matrix, the initial radius and the initial unconstrained estimate, it should come as no surprise that TSVD provides a better performance compared to ZF both as a standalone detector, as well as when combined with FSD. The hybrid TSVD-FSD receiver improves performance by using the TSVD estimate instead of the ZF estimate for the

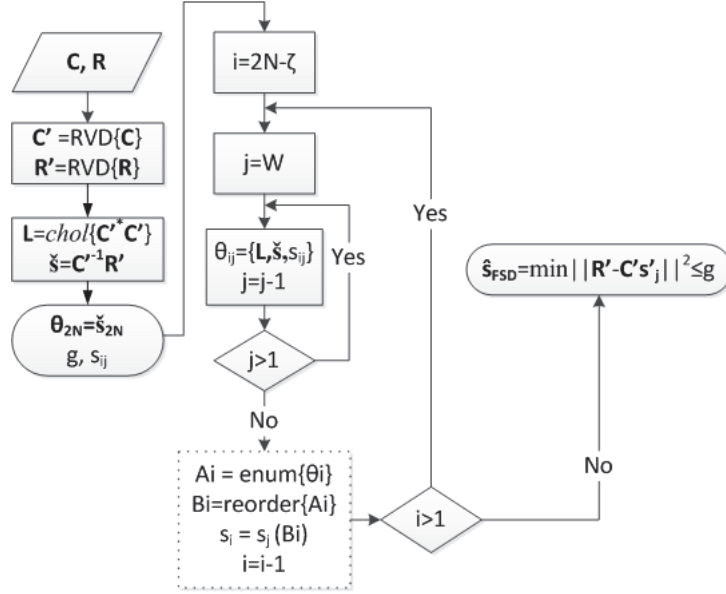


Figure 7.6: Flowchart of the FSD algorithm for the detection of SEFDM signals.

computation of the initial radius according to

$$g = \|\mathbf{R} - \mathbf{C}\hat{\mathbf{s}}_{TSVD}\|^2. \quad (7.12)$$

If no solution is found when the algorithm has completed execution, the FSD reverts to the linear TSVD estimate $\hat{\mathbf{s}}_{TSVD}$.

The algorithm for the FSD is outlined in Fig. 7.6. It is assumed that the constellation is complex-valued rectangular M-QAM, hence an RVD process is carried out during the initialisation stage. Fig. 7.6 shows that the operation of the FSD resembles that for SD. Instead of updating the radius and the interval centre on each iteration, as in the conventional SD algorithm, the FSD only computes the PEDs based on θ and the partial symbol estimates s_{ij} acquired as the algorithm moves down the tree.

The complexity of the algorithm is user-defined by setting the parameter ζ , which defines the complexity order of the algorithm, with $W = (\sqrt{M}^\zeta)$ where W denotes the tree width. During the initialisation phase, the first ζ levels are fully expanded. In the subsequent levels, only the best W candidate nodes are expanded. The tree width

determines the complexity of the FSD, as it defines the number of node visits per level. The larger the tree width, the better the performance at the expense of increased complexity. From Fig. 7.6, it should be evident that the update interval centres θ_{ij} at each level can be computed independently in parallel rendering the FSD algorithm an attractive choice for hardware implementation using FPGAs.

The sorting stage which performs comparisons amongst all candidate nodes B_i at each level constitutes one of the main bottlenecks. The larger the tree width, the higher the number of comparisons, leading to an increase in the overall latency of the algorithm and subsequently reduction in throughput. On the other hand, reducing the tree width or removing the sorting stage affects the order in which a parent node branches out to a child node, potentially degrading BER performance. To strike a balance between these conflicting objectives, the application of SF detection is proposed, which in conjunction with MRVD, promise to deliver performance gains both in terms of latency and BER performance as established in MIMO systems [195].

FSD Algorithm Enhancements

While the hybrid TSVD-FSD detector described in [156] significantly lowered the algorithm's execution time, the focus of this section is the enhancement of this receiver to make it better-suited for application in the real world. The first part considers the simplification of the sorting stage. Fig. 7.7 compares the spanning tree for two variants of the FSD with different sorting strategies, namely the Shortest Path (SP), entailed in the original FSD algorithm, and the novel SF technique. For simplicity of presentation, the number of sub-carriers has been set to two, resulting in a four-level tree, assuming RVD is employed, with the tree width also fixed at a value of two.

Fig. 7.7 shows that both algorithms share the same iterative tree traversal process, which is carried out in a breadth-first manner. From Fig. 7.7, it is evident that at each level the SP technique retains the best W candidate nodes having the minimum distance from the interval centre. This is achieved thanks to the SE reordering that

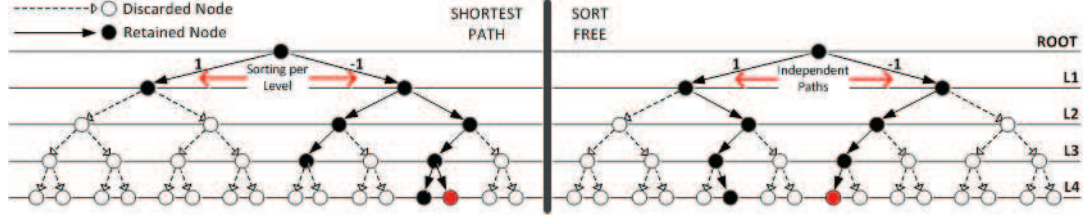


Figure 7.7: Spanning tree of Shortest-Path FSD (left) vs. Sort-Free FSD (right) ($N = 2$, $W = 2$).

takes place which may lead to discarding certain candidate nodes at the level under consideration albeit being selected from the immediately preceding level. Conversely, in the SF version, every candidate node selected at the current level branches out to one of the (two in this case) child nodes being the one closest to θ .

Consequently, it should be evident that the SP method naturally converges to the best solution, whereas for the SF method, the path with the smallest distance from the centre point is chosen at the end of the tree traversal. This is the reason SF leads to an inferior error performance with the benefit of substantial reduction in computational complexity, which is inverse proportional to the execution speed of the algorithm and thus the resultant throughput.

To visualise the difference between the SP and SF methods, the blocks in Fig. 7.6 with a dotted border, would be made redundant in a SF implementation. The removal of the sorting stage in the SF method allows the W independent paths to be executed concurrently rendering the algorithm almost fully parallel and favouring the use of FPGAs for its practical realisation.

Another modification aimed at improving the BER performance is MRVD as described in [209]. This modified version of RVD involves the permutation of the system vectors and matrices according to

$$\begin{bmatrix} \text{Re}\{R_0\} \\ \text{Im}\{R_0\} \\ \vdots \\ \text{Re}\{R_{N-1}\} \\ \text{Im}\{R_{N-1}\} \end{bmatrix} = \mathbf{C}_{\text{MRVD}} \begin{bmatrix} \text{Re}\{s_0\} \\ \text{Im}\{s_0\} \\ \vdots \\ \text{Re}\{s_{N-1}\} \\ \text{Im}\{s_{N-1}\} \end{bmatrix} + \begin{bmatrix} \text{Re}\{Z_0\} \\ \text{Im}\{Z_0\} \\ \vdots \\ \text{Re}\{Z_{N-1}\} \\ \text{Im}\{Z_{N-1}\} \end{bmatrix}$$

MRVD does not incur additional computational complexity and hence is ideal for performance trade-offs between the size of the system and the complexity of the FSD algorithm, as discussed in Section 7.3. The reason for the improved performance of MRVD lies in the fact that the first ζ levels that are fully expanded do not consist of solely the imaginary part of the initial SEFDM symbol estimates. Instead, at each level, the algorithm flips between the real and imaginary parts of the next symbol in line. Therefore, MRVD can be regarded as providing ‘more complete’ solutions compared to the conventional RVD.

7.2 Joint FPGA-DSP Platform

While FPGA implementations of FSD have appeared in MIMO systems, the difference in the application of FSD to SEFDM as opposed to MIMO, lies in the much larger system dimension associated with the former resulting in significant increase in computational complexity. Preliminary studies showed that a standalone FPGA realisation is hindered by the device’s limited resources, in this case being a Xilinx Virtex 5 chip. To this end, this work considers the joint hardware implementation of the hybrid TSVD-FSD receiver with the aid of FPGAs and DSPs. The target FPGA device is the XC5VFX100T employed in Chapter 6, whereas the DSP is a *Texas Instruments* (henceforth referred to as TI) TMS320C6455 processor running at 1 GHz. Screenshots illustrating the data exchange between the FPGA and the DSP are provided in Appendix B.

A high-level representation of the complete hardware system is depicted in Fig. 7.8.

The FPGA outputs are transferred to the DSP over an asynchronous, bi-directional, communication port with the aid of an External Memory Interface (EMIF) [210].

The receiver is fully reconfigurable allowing the user to change the number of sub-carriers and/or the level of bandwidth compression on the fly. Recalling from Section 6.3.1, the first key component is an FFT block whose length can be reconfigured at runtime. This enables the user to achieve optimum latencies depending on the number of sub-carriers selected. The second important block is the bank of ROMs each of which is responsible for storing the inverse or pseudoinverse of the sub-carriers correlation coefficient matrix values defined for a specific set of N , α and ρ . Furthermore, the user can choose whether ZF or TSVD is employed for the initial estimation of the transmitted symbols. The final block of the FPGA partition is a slicing unit which is currently configured to decode BPSK or 4-QAM signals on a frame-by-frame basis.

On the DSP side, a commercially available DSP development environment was used to implement the SP and SF FSD detectors on the device and to handle the data transfer between the DSP and the computer terminal. This partition of the overall hardware system is also reconfigurable in that the user can select the desired degree of complexity by setting the tree width W during the execution of the algorithm.

The DSP acts as the brains of the system activating and deactivating the multiple control signals and polling the FPGA for status updates. The use of MUXs is paramount in this setup, since they control the exchange of data between the two devices. Once the FPGA has completed processing of the received data, providing an initial ZF or TSVD estimate, the initial symbol estimates are stored in FIFOs. The data is then transmitted over the port to the DSP where the original (SP) or modified (SF) FSD algorithm is executed to provide a better estimate of the original, transmitted symbols. Custom control modules were developed for both the FPGA and the DSP to monitor the bi-directional communication. These modules are responsible for ensuring the FPGA and the DSP are synchronised at all times while reporting errors for invalid commands, invalid data, as well as system malfunction.

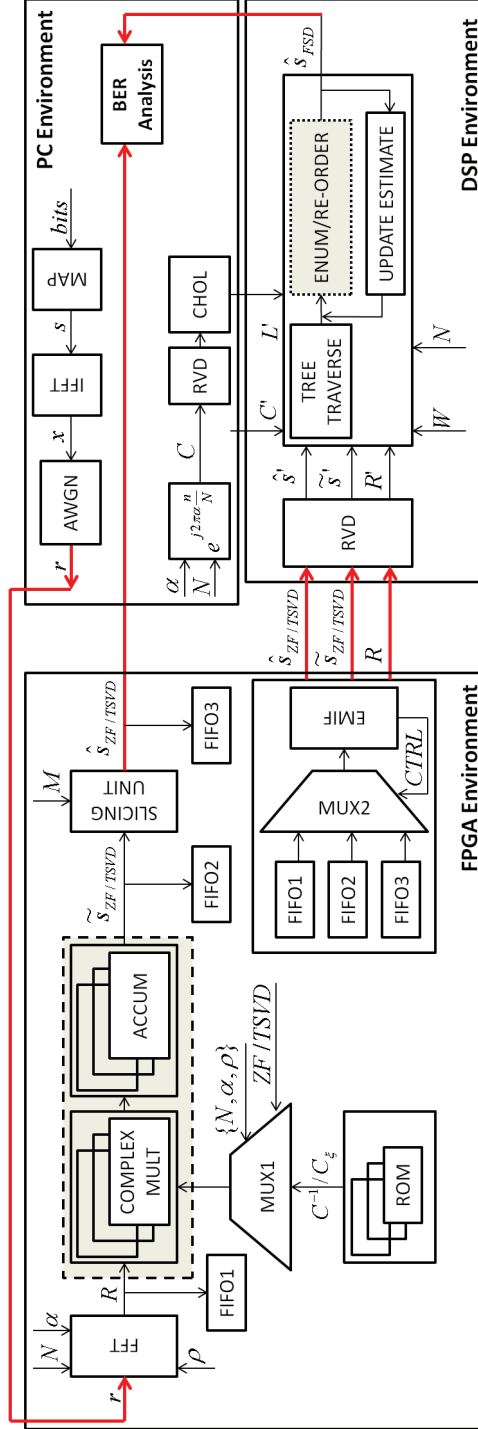


Figure 7.8: System block diagram for the hybrid TSVD-FSD hardware detector implementation.

One of the key challenges encountered during the programming of the interface between the FPGA and the DSP was number conversion. Two number formats are recognised by the DSP, namely Binary-Coded Decimal (BCD), which is essentially American Standard Code for Information Interchange (ASCII) format and IEEE-754 single-precision format. The FPGA on the other hand only recognises standard formats, such as binary, decimal or hexadecimal. Therefore, functions had to be created to convert the values from one format to another to guarantee that the data values were being translated correctly.

7.3 DSP Evaluation

In this section, the performance of the proposed detector is analysed in terms of complexity and error performance through simulation and experimentation. For the latter, the practical measurements were acquired using the TMS320C6455 chip introduced in Section 7.2. The proposed detector is compared against the original TSVD-FSD detector described in [156] taking into account the aforementioned metrics. The modulation scheme adopted throughout the analysis was 4-QAM, however, higher modulation orders could also be accommodated, since they may be expressed as linear combinations of 4-QAM symbols [198].

As explained in Section 7.1.2, the sorting stage in the SP method adds a significant burden to the computational complexity of the algorithm. The SF technique alleviates the need for this stage and can be visualised in Fig. 7.8, where the grey dotted box in the DSP partition would be made redundant in such a scenario. For the purpose of this work, it is assumed that matrix operations, such as RVD and Cholesky decomposition are carried out in preprocessing units in a computer environment prior to the execution of the FSD algorithm on the DSP.

Table 7.1: Total number of operations per level for the Shortest-Path and Sort-Free FSD detectors.

Metric	Shortest-Path (SP)	Sort-Free (SF)
Multiplications	$W(D - k)$	$W(D - k)$
Additions	$W(2(D - k) + 1 + \sqrt{M})$	$W(2(D - k) + 1)$
Comparisons	$\frac{1}{2}(MW^2 + \sqrt{M}W + 4W)$	W

7.3.1 Computational Complexity

The algorithmic complexity is determined by two aspects; first, the number of arithmetic operations required to execute the algorithm, such as multiplications, additions³ and subtractions; second, the number of compare-select operations that take place during the sorting stage.

Table 7.1 compares the total number of operations required at each tree level for the SP and SF detection schemes. The depth of the tree, commensurate with the system dimension, is denoted by D which is equal to $2N$ as a result of using RVD. From this table, it is evident that the number of comparisons at each level is fixed and independent of the system dimension. For the SF method, a single comparator is required for each node to compute the minimum distances to the centre point. The number of comparisons is equal to the tree width. On the contrary, the SP technique requires multiple cascaded minimum finders to perform the reordering of the candidate nodes with respect to the minimum distance from the centre point. In both cases, the number of comparisons is dictated by the tree width.

As far as arithmetic operations are concerned, the lower sub-plot in Fig. 7.9 shows that the number of multiplications and additions increases linearly at each level of the tree. This is due to the fact that the computation of the new symbol estimate at each level takes into account all points from the root of the tree up to and including the level under consideration. Consequently, the growth in the total number of operations required for the complete execution of the algorithm is of quadratic complexity, as

³In this context, it is assumed that the number of additions encapsulates relevant operations, such as accumulations and subtractions.

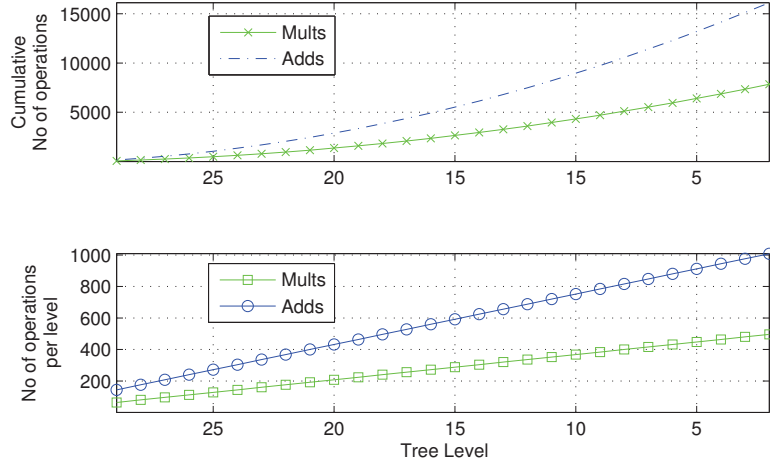


Figure 7.9: Computational complexity for increasing tree depth.

depicted in the upper sub-plot of Fig. 7.9. Note that the values on the x-axis have been listed in descending order, since the root of the tree corresponds to the level with the highest value being $2N + 1$.

The results from Fig. 7.9 imply that the implementation complexity of FSD for an SEFDM system is at least one order of magnitude greater compared to existing MIMO implementations. Recalling from the introduction of this chapter, the number of antennas for a typical MIMO system employing FSD is 4×4 , resulting in eight tree levels, which as can be seen from Fig. 7.9 requires a substantially lower number of arithmetic operations compared to a 16 sub-carrier SEFDM system with 32 tree levels.

Table 7.2 compares the number of operations required at the sorting stage for the SP and SF versions of the TSVD-FSD algorithm. The size of the system is eight sub-carriers and the complexity is shown for increasing tree widths. Note that the number of multiplications and additions is virtually the same for both methods, as can be deduced from Table 7.1. From Table 7.2, it is clear that the SF technique requires almost 20 times fewer compare-select operations as opposed to the SP technique for a small tree width ($W = 8$). The performance gain increases in proportion to the tree width. For example, a SF detection with an algorithm complexity ($W = 256$) 32 times greater than

Table 7.2: Number of compare-select operations for the Shortest-Path and Sort-Free FSD detectors for increasing tree widths.

FSD Variant	$W = 8$	$W = 16$	$W = 64$	$W = 256$
Shortest Path [156]	1976	6720	83840	1054720
Sort Free	104	192	640	2048

Table 7.3: Comparison of the execution time for the two FSD variants for an increasing system size and tree width.

FSD Variant	$W = 8$			
	$N = 4$	$N = 8$	$N = 12$	$N = 16$
Shortest-Path	3 ms	11 ms	21 ms	40 ms
Sort-Free	1 ms	4 ms	8 ms	15 ms
	$W = 64$			
	$N = 4$	$N = 8$	$N = 12$	$N = 16$
Shortest-Path	29 ms	174 ms	339 ms	586 ms
Sort-Free	6 ms	30 ms	64 ms	118 ms

a SP detection ($W = 8$) requires almost the same number of compare-select operations.

7.3.2 Execution Time

Table 7.3 compares the execution times for the SP and SF variants when implemented on the DSP. For both schemes, a linear increase in the number of sub-carriers leads to a quadratic increase in the execution time. This is due to the fact that the number of multiplications and additions increases linearly as the algorithm traverses the tree according to the formulae presented in Table 7.1. Similarly, an exponential increase in tree width leads to an exponential increase in execution time. Table 7.3 also shows that the SF method can provide greater speed efficiency compared to the SP method of at least two times, for example when $W = 8$ and $N = 16$, and up to almost six times, for example when $W = 64$ and $N = 8$. These execution times are provided as an indication of the performance trade-offs one might expect for varying algorithmic complexity.

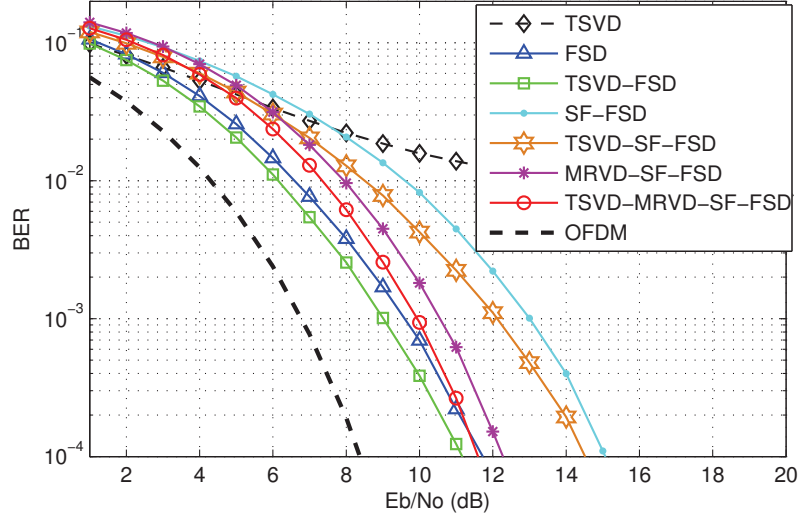


Figure 7.10: BER performance for the different FSD variants (4-QAM, $\alpha = 0.8$, $N = 16$ and $W = 16$).

7.3.3 Error Performance

This final sub-section, provides numerical results evaluating the error performance of the FSD algorithm and its variants. The computer based simulations were carried out in an AWGN channel for a range of SNR values and for 1000 SEFDM symbols unless otherwise stated. The experimental results acquired using the DSP are compared against the analytical floating-point and modelled fixed-point data.

Fig. 7.10 compares the BER of the original FSD detector against its variants. 10,000 SEFDM symbols were simulated in this sole case to obtain higher confidence levels. It is clear that TSVD and MRVD when separately applied to the standalone SF yield significant performance gains, especially in the higher SNR regimes. When both techniques are combined and applied to SF, the resultant TSVD-MRVD-SF FSD detector approaches the performance of the more optimum, SP detector, yet with the benefit of reduced computational complexity.

Fig. 7.11 shows that the combined TSVD-MRVD-SF FSD detector has comparable performance to the SP FSD detector for different tree widths. From Fig. 7.11, it is

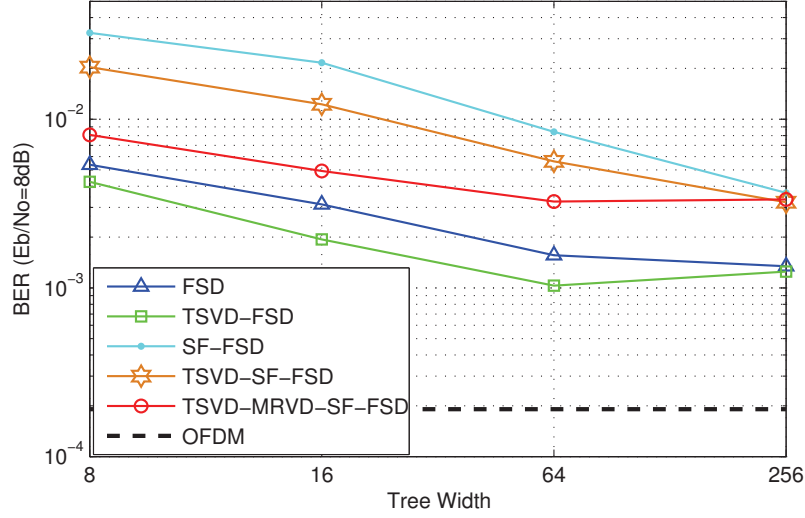


Figure 7.11: Comparison of the FSD detection schemes for increasing complexity (4-QAM, $\alpha = 0.8$ and $N = 16$).

also evident that the combined detector provides greater performance gains for smaller tree widths, which in turn imply a lower computational complexity, the latter being favourable in any hardware design.

Fig. 7.12 illustrates that the performance gains acquired with the combined TSVD-MRVD-SF FSD detector are more prominent for large system sizes ($N > 8$), thanks to the better quality initial estimate provided by TSVD. For small system sizes ($N = 4$), the combined detector leads to a slight performance degradation, which appears only for the FSD variant employing MRVD. Further investigation is required to evaluate the performance of MRVD for small system sizes.

Fig. 7.13 shows the error performance for varying degrees of bandwidth compression. It is evident that the combined TSVD-MRVD-SF FSD provides the best trade-off between BER and complexity for $\alpha \geq 0.8$, taking into account the substantial reduction in the number of compare-select operations. Moreover, the SF FSD architecture is well-suited for a fully parallel and pipelined FPGA implementation. These BER results coupled with the complexity discussion of Section 7.3.1, demonstrate that the

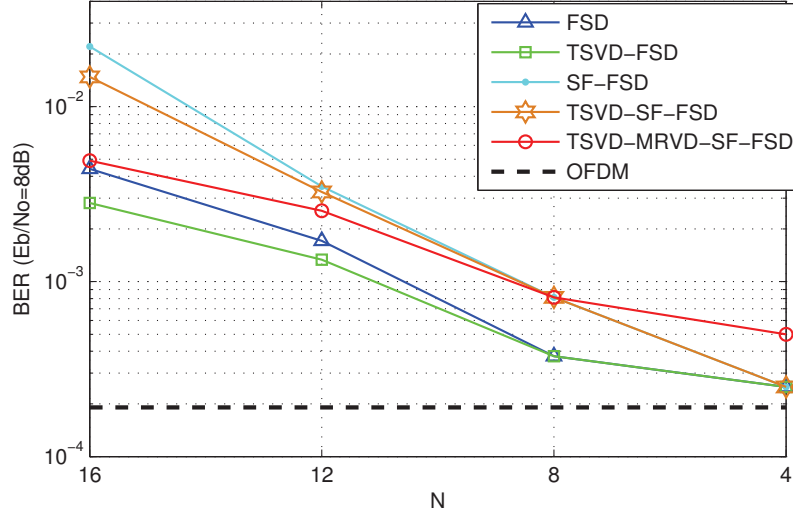


Figure 7.12: Comparison of the FSD detection schemes for different system sizes (4-QAM, $\alpha = 0.8$, $W = 16$).

combined TSVD-MRVD-SF FSD provides the best trade-off between BER and complexity for a bandwidth compression up to 20%, making it the preferred choice for a feasible hardware realisation.

Fig. 7.14 compares the error performance at E_b/N_0 values of 5, 6 and 7 dB, of the analytical floating-point, modelled fixed-point and experimental DSP results. The bandwidth compression is equal to 20% while results were acquired for two different system sizes, namely $N = 8$ and $N = 12$. The figure provides evidence that the experimental results are in accordance with the theoretical results and match the results obtained via the simulation of the fixed-point model.

Finally, Figs. 7.15 and 7.16 report the BER performance for the hybrid TSVD-FSD detector implemented on the joint FPGA-DSP platform. Fig. 7.15 shows that all FSD variants outperform the standalone TSVD detector. The proposed TSVD-SF-FSD detector provides comparable performance to the original TSVD-FSD detector for medium system sizes while a degradation in performance is noticed for $N \geq 12$. Fig. 7.15 depicts that the hybrid FPGA-DSP hardware receiver approaches the theoretical

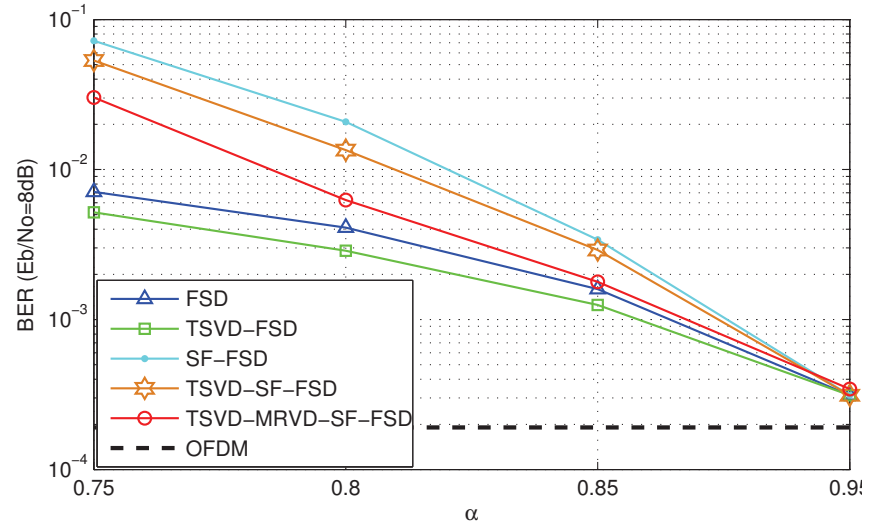


Figure 7.13: Comparison of the FSD detection schemes at different values of α (4-QAM, $N = 16$, $W = 16$).

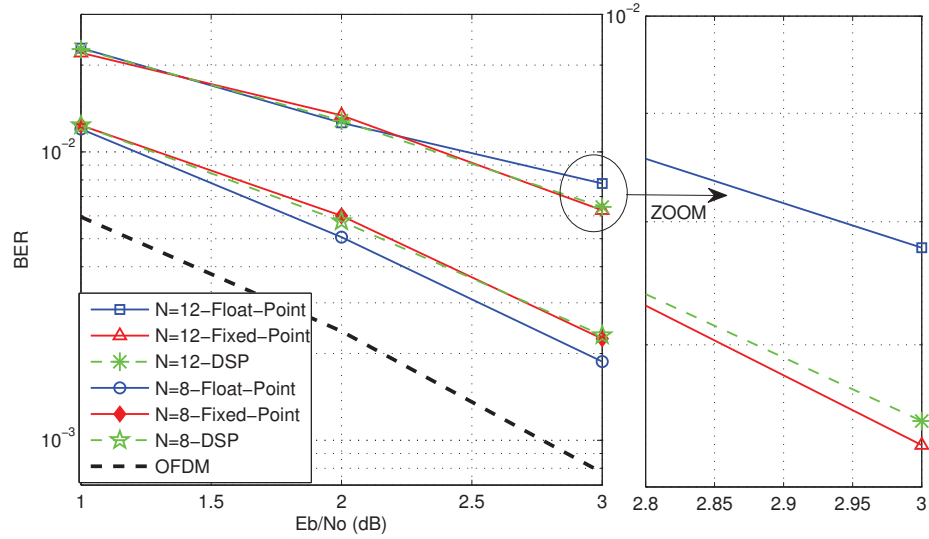


Figure 7.14: Error performance of analytical, modelled and experimental FSD detectors (4-QAM, $\alpha = 0.8$, $N = 8$, $W = 16$).

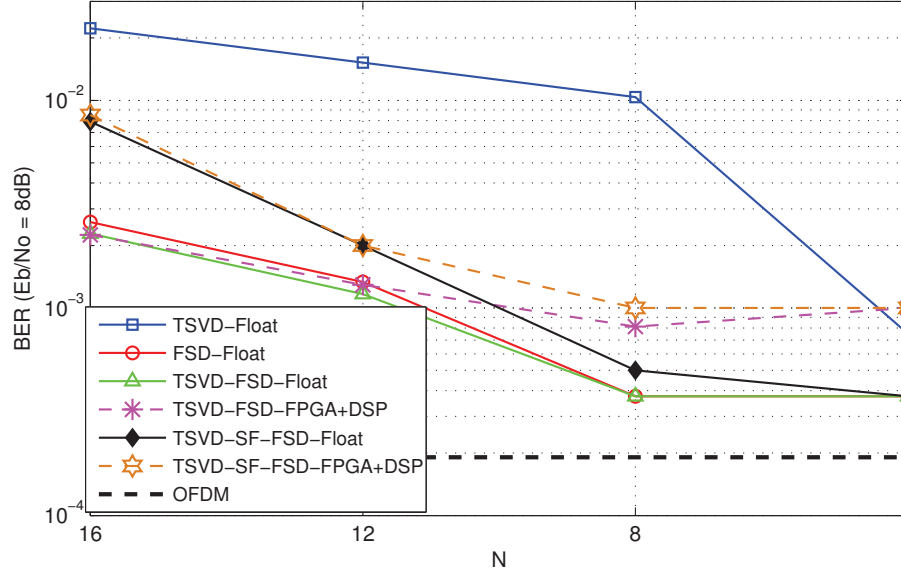


Figure 7.15: BER performance of the joint FPGA-DSP, TSVD-SF-FSD hardware detector for a varying number of sub-carriers N (4-QAM, $\alpha = 0.8$, $W = 16$).

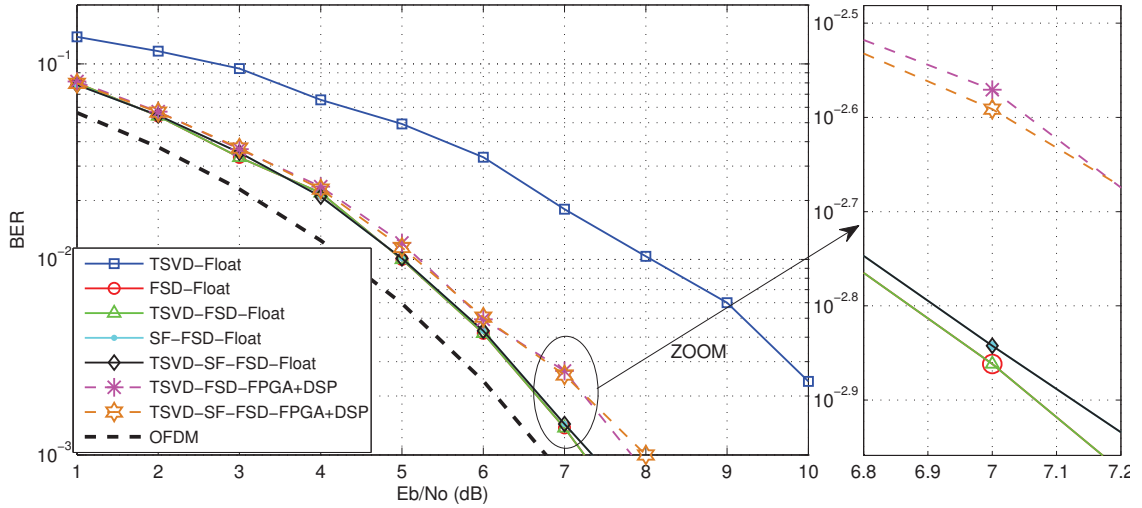


Figure 7.16: BER performance of the joint FPGA-DSP, TSVD-SF-FSD hardware detector against E_b/N_0 (4-QAM, $\alpha = 0.8$, $N = 8$, $W = 16$).

simulation results with increasing system sizes. Fig. 7.16 illustrates that the experimental results are in accordance with the results obtained from theory, thus verifying the correct functionality of the hardware implementation.

7.4 Results from the FPGA Implementation of the FSD Algorithm

In modern systems, FPGAs have superseded generic processors thanks to their unique parallel processing and reconfigurability features [194]. Independent FPGA realisations of a reconfigurable SEFDM transmitter and ZF-TSVD receiver have appeared in [175] and [179], respectively. In [60], the authors show that an FPGA chip can provide 225 times greater efficiency compared to general-processor platforms for an N-body algorithm. This particular algorithm requires similar feedback and update operations as the FSD algorithm explored in this work. The throughput gain is achieved by leveraging the FPGA's capabilities, such as parallelism, pipelining and loop unrolling [60].

The FPGA implementation of the FSD algorithm as applied to SEFDM systems is challenging due to the large dimension of the problem. While hardware implementations of different SD and FSD variants have appeared in MIMO systems using FPGAs [211] and VLSI chips [212], a linear increase in the number of tree levels or algorithm complexity has a high impact on area and clock loading. Furthermore, FPGAs are slower than Application Specific Integrated Circuits (ASICs) and would require a prohibitive amount of resources for parallel processing to achieve the same result [213].

Despite these challenges, recent work [188] has achieved a standalone FSD implementation on an FPGA device thanks to the state-of-the-art Virtex 6 chip employed.

7.4.1 Implementation Aspects

Details of the FPGA architecture devised for the FSD algorithm considering both SP and SF variants are presented in [188]. This section discusses key challenges encountered during such a hardware realisation and the impact system parameters have on the overall performance.

First, the high throughput rates reported in FPGA implementations of FSD variants in MIMO systems is attributed to the fact that multiple PED blocks are configured in parallel and are executed concurrently resulting in a very low latency. Nevertheless, a small change in one parameter may adversely affect the entire design.

For example, the VLSI prototype of the K-Best algorithm presented in [61] can achieve a throughput of 376 Mbps for a 4×4 MIMO system with 16-QAM modulation when $K = 5$. Increasing the value of K ameliorates the BER performance, however, this costs dearly in terms of hardware resources. For the same example, doubling the value of K from five to $K = 10$ leads to reduction in throughput by almost 80% yielding 80 Mbps. There are two key causes effecting this drop in throughput; first, the number of clock cycles required for a complete execution of the algorithm is increased, thereby increasing the pipelined latency; second, since a parallel configuration has been adopted in this design, increasing the value of K translates directly to increasing the number of blocks executed in parallel. In addition to the increased resource utilisation, the additional blocks result in a longer critical path delay which in turn reduces the maximum achievable clock frequency.

Table 7.4 compares the resource availability of the FPGA devices used for the implementation of tree-search algorithms in MIMO systems [61] and SEFDM systems [188], respectively. While the XC6VLX240T has no doubt more powerful digital signal processing capabilities compared to the XC2V6000, the previous example exemplifies the effect algorithmic complexity can have on silicon area requirements.

Table 7.4: Comparison of an XC2V6000 [61] to an XC6VLX240T [188].

Device	Logic Cells	Slices	DSP Slices	18k BRAMs	36k BRAMs
XC2V6000	51,840	33,792	144	144	-
XC6VLX240T	241,152	37,680	768	832	416

7.4.2 Numerical Results

The results presented in this section correspond to the FPGA implementation of an SEFDM system [188] which considers 16 sub-carriers with $\alpha = 0.8$ and $W = 16$. Four architectures were devised, two for the SF variant and two for the SP variant of the FSD algorithm. The throughput depends on the clock frequency f_{clk} of the design, as well as the number of pipelined cycles $P_{LATENCY}$ required per detection, according to

$$R_b = \frac{N \cdot \log_2 M}{P_{LATENCY}} \cdot f_{clk}, \quad (7.13)$$

in a similar fashion to the FPGA implementation presented in Chapter 6. Furthermore, BRAMs are used for buffering, synchronisation and the storage of matrix elements, the latter accomplished with the aid of ROMs. The DSP Slices handle the signal processing complexity of the algorithm, whereas the LUTs comprise the adders and control logic.

Table 7.5 compares the estimated resource utilisation reported by the synthesis process for the two FPGA architectures adhering to the SF method. A fully-parallel configuration translates to computing 16 PEDs concurrently, thus theoretically achieving the highest throughput. Evidently, from Table 7.5, such a configuration exceeds the device's resource bounds, therefore an alternative semi-parallel approach where eight PED blocks operate at a time in a pipelined fashion is adopted.

Table 7.5 shows that by doubling the pipelined latency of the original SF architecture, the resource utilisation is reduced by more than half. However, this resource optimisation does result in a throughput penalty of the same magnitude. Table 7.6 repeats the results for the SP method. These results corroborate the predictions pre-

Table 7.5: Estimated resource utilisation and clock performance for the SF-FSD detector after synthesis.

Metric	Available	Fully-Parallel	Hybrid Parallel-Pipelined
Slices	301,440	257,507 (85%)	58,295 (19%)
LUTs	150,720	356,251 (237%)	149,222 (99%)
DSP Slices	768	742 (96%)	371 (48%)
Max. Freq.	-	173.913 MHz	171.145 MHz
Initial Latency	-	198 cycles	284 cycles
Pipelined Latency	-	1 cycle	2 cycles
Throughput	-	5.565 Gbps	2.738 Gbps

Table 7.6: Estimated resource utilisation and clock performance for the SP-FSD detector after synthesis.

Metric	Available	Hybrid Parallel-Pipelined	Fully-Pipelined
Slices	301,440	312,684 (103%)	117,291 (38%)
LUTs	150,720	801,107 (531%)	146,772 (97%)
DSP Slices	768	370 (48%)	301 (39%)
Max. Freq.	-	171.145 MHz	173.913 MHz
Initial Latency	-	786 cycles	1,274 cycles
Pipelined Latency	-	2 cycles	16 cycles
Throughput	-	2.738 Gbps	348 Mbps

sented in Section 7.3.2, since it is clear that in order to attain the same throughput performance, the resource requirements for the SP technique exceed those mandated for the SF technique by over five times. Comparing Tables 7.5 and 7.6 it is evident that even a considerable reduction in throughput for the fully-pipelined SP technique still occupies almost the same amount of area as the semi-parallel SF technique.

The aforementioned results indicate that pipelining plays a dominant role in generating a design that can actually be implemented in the real world. Nonetheless, introducing additional pipeline stages has a serious impact on initial latency, as well as pipelined latency.

Table 7.7 presents the results acquired from having completed the full FPGA implementation cycle which creates a bitstream ready to be downloaded to the target device. As expected, the resource utilisation is almost the same as that reported by

Table 7.7: True resource utilisation and clock performance of SF-FSD and SP-FSD after implementation.

Metric	Available	SF (Hybrid Parallel-Pipelined)	SP (Fully-Pipelined)
Slices	301,440	58,295 (19%)	117,291 (38%)
LUTs	150,720	134,846 (89%)	115,942 (76%)
DSP Slices	768	371 (48%)	301 (39%)
Max. Freq.	-	66.344 MHz	67.618 MHz
Initial Latency	-	284 cycles	1,274 cycles
Pipelined Latency	-	2 cycles	16 cycles
Throughput	-	1.06 Gbps	135 Mbps

the synthesis cycle, if not better. However, the clock frequency is almost always considerably lower due to the inability of the tools to reduce the critical path delay during the placing and routing of the design on the target FPGA device. As a final note, it is important to emphasize that the data rates presented in Table 7.7 are unrealistic since they correspond to the raw output of the FPGA based FSD block operating in isolation. It is expected that the proper data rates, when the FSD block is integrated with the remaining modules entailed in an SEFDM receiver, will be significantly lower.

Finally, Fig. 7.17 compares the error performance of the analytical and FPGA based SF and SP variants of the FSD algorithm when TSVD is used as the initial estimate. Clearly, the benefits of the SF technique in terms of throughput and resource utilisation are traded off with a degradation in BER performance. This calls for the need to strike a balance between design requirements and target outcomes based on the application in mind and the specified criteria.

7.5 Discussion on Algorithm and Circuit Optimisation

Two key techniques which have been employed in MIMO systems to improve system performance from an algorithmic point of view are channel ordering⁴ and soft decod-

⁴In this context, channel ordering refers to the order in which each of the parallel data streams in multi-antenna and/or multi-carrier systems is decoded according to the per stream SNR and should not be confused with the enumeration strategies, such as SE ordering, presented in previous sections.

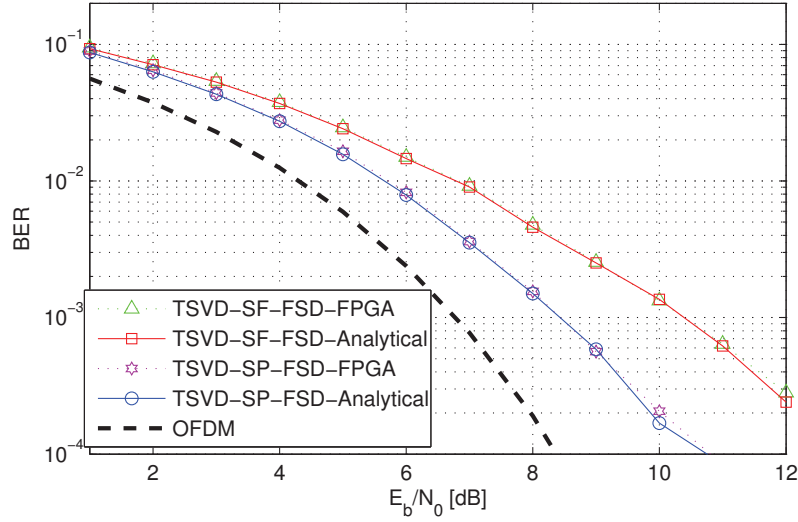


Figure 7.17: BER performance of the real-time FPGA based SF-FSD and SP-FSD detectors for a 4-QAM SEFDM system with $N = 16$, $\alpha = 0.8$ and $W = 16$.

ing. For the former, the most popular channel ordering strategies include column-norm ordering, the V-BLAST scheme and sorted QR decomposition [61]. In column-norm ordering, as the name suggests, the columns of the channel matrix are reordered according to descending values of the norm of the rows. The V-BLAST and sorted QR decomposition are classified as preprocessing methods for SIC, however, sorted QR has a significantly lower computational complexity compared to V-BLAST and is therefore a more attractive choice for hardware implementation. These channel ordering methods yield different performance gains for the SD and K-Best algorithms. For the SD technique, ordering leads to reduction in complexity since the algorithm is likely to converge more rapidly to the ML solution. For the FSD method, assuming a fixed tree width W , ordering does not alter the complexity, though it does improve BER performance as the paths included in the search are more likely to contain the ML solution.

Soft decoding is the second technique used to improve performance since it essentially results in a coded SD or FSD algorithm. For example, a soft decoding process

was adopted to complement the FCSD algorithm proposed by Barbero [214] to generate candidate points which are of a better quality compared to the uncoded version.

In summary, it should be evident that the above techniques could be adapted to the SEFDM system model with the aim of yielding performance gains. In terms of circuit optimisation, Section 7.4.2 demonstrated that the current implementation challenges lie in the finite resources available in an FPGA device and the inherent high complexity of the detection algorithms applied in SEFDM. To mitigate this, some simple methods for optimising resource utilisation at the expense of performance degradation (in terms of throughput and/or BER performance) include:

- Computing the ℓ^1 norm, known as the Manhattan distance, according to $|a| + |b|$, instead of the current ℓ^2 norm or Euclidean distance given by $\sqrt{a^2 + b^2}$ alleviating the need for square and root square operations.
- Using a 3-multiplier structure instead of the current 4-multiplier configuration to carry out the multiplication between complex numbers.
- Replacing multipliers by equivalent shift and add operations.

To conclude, it is worth mentioning that the key challenge associated with both the SD and FSD algorithms is the inability to determine the optimum set of design parameters. For example, an increase in tree width improves the error performance of the FSD algorithm, yet no mathematical model exists which dictates the optimum tree width for a given set of N , α and ρ . This is similar to the case for K-Best decoding where no analytical formulae are available to dictate the relationship between the design parameter K and the required data rate [61]. In both scenarios, the optimum set of parameters has to be identified through simulation and/or experimentation. Finally, while SD is able to achieve ML performance provided the decoding time is not constrained, the same does not apply for the FSD algorithm which can only approach quasi-ML performance, though with the benefit of a fixed and theoretically much higher throughput.

7.6 Conclusions

This final piece of work completes the puzzle having examined performance trade-offs associated with reduced complexity detection techniques to deduce the optimum method for devising a practical SEFDM hardware receiver. The chapter commenced with an overview of the SD algorithm, since Kanaras showed that applying SD to SEFDM systems results in bandwidth gains at significantly reduced computational complexity compared to brute force ML [198]. Notwithstanding, it was quickly discovered that this technique is not suitable for hardware implementation due to its variable complexity.

Subsequently, the FSD algorithm proposed in [156] was considered, since this algorithm fixes the complexity of the original SD algorithm by examining a fixed number of nodes at each level of the tree. It was found that the sorting stage in this algorithm constitutes a primary bottleneck for practical realisation. For this reason, techniques previously employed in MIMO systems were adapted to address the challenge presented in the detection of non-orthogonal FDM signals with a two-fold aim; first, to reduce the computational complexity of the FSD algorithm making it better-suited for application in the real world; second, to improve the error performance of the previously proposed TSVD-FSD detector by employing MRVD instead of conventional RVD without incurring additional computational costs.

These findings were validated via simulation and practical experimentation, the latter carried out with the aid of a DSP chip. More precisely, the DSP was used to evaluate the fixed-point performance of the FSD algorithm and to quantify the gains in speed obtained by employing the SF strategy. The results provided evidence that the proposed algorithm performs well under fixed-point conditions with comparable performance to existing floating-point results. It was also shown that the modified FSD adopting a SF approach is capable of providing up to six times greater speed compared to the conventional FSD at the expense of increased BER. MRVD was then employed which improved the performance of the SF approach, especially for high SNR

regimes.

Numerical results also demonstrated that for $\alpha \geq 0.8$, the enhanced TSVD-MRVD-SF FSD detector is the preferred choice for providing the best trade-off between desirable error performance and tangible computational complexity. The investigation revealed that the high dimensionality associated with the SEFDM system sizes under consideration in contrast to existing MIMO implementations, presents paramount technical challenges in terms of system resources and latency. However, the ever-increasing speed and resource availability of current state-of-the-art devices, in particular FPGAs, favour the feasibility of a real-time SEFDM receiver.

To this end, the final part of this work presented for the first time the hardware realisation of a hybrid TSVD-FSD detector using a joint FPGA and DSP platform. A more powerful FPGA chip was then employed to implement the FSD algorithm in isolation with the aim of assessing the impact of its complexity on silicon area and throughput.

Future work will involve the transition to an advanced FPGA platform comprising high-level software tools to accelerate development and simplify hardware implementation. The two primary objectives are; first, to realise the entire SEFDM receiver on a single FPGA chip, thus avoiding the bottlenecks associated with the communication between different devices, in this work between the FPGA and the DSP; second, to reach target throughputs which cannot be achieved using generic processors or earlier generation FPGA chips. Research will also encompass the study of more sophisticated iterative detection techniques, which alleviate the need for intractable arithmetic operations, such as matrix inversion.

Chapter 8

Conclusions

Part I of this thesis explored key technologies aimed at enhancing capacity and increasing data rates in next generation networks, in particular, OFDM and CR. Chapter 2 highlighted the merits and drawbacks of OFDM with an emphasis on the limitations imposed upon the maximum achievable spectrum efficiency. The use of SC-FDMA to address the PAPR problem in OFDM based systems was discussed and followed by a simulation based study to evaluate and compare different sub-carrier mapping and allocation schemes. It was found that these mapping schemes offered diverse trade-offs in terms of throughput, PAPR performance and robustness against multipath fading.

Chapter 3 described techniques for ameliorating system performance in CR networks. The first part considered techniques for improving signal detection and system identification. The discussion established that cyclostationary detection offers the best performance in terms of low complexity and response accuracy. The second part proposed a spectrum access framework to improve the QoS of secondary users. It was shown that by incorporating traffic prediction techniques in the CR cycle, the throughput and packet loss ratio of secondary users is optimised.

Notwithstanding, this initial examination revealed that the lack of spectrum availability constitutes the primary challenge in future wireless and mobile communication systems. Hence, the focus for the latter part of this thesis was shifted to novel multi-

carrier modulation techniques which share the common goal of increasing spectral efficiency. More precisely, Part II of this thesis, addressed the gap of rendering SEFDM detection algorithms suitable for hardware implementation. This comprised multiple objectives; first, the performance of SEFDM was evaluated in a practical environment in the presence of quantisation errors and limited sampling rates; second, previously reported algorithms were optimised to acquire a targeted reduction in complexity; third, selected SEFDM detectors were implemented using FPGAs and DSPs, providing, for the first time, a reference of the true hardware complexity associated with an SEFDM receiver.

Chapter 4 presented the principles of SEFDM systems and the challenges encountered due to the ill-conditioning of the sub-carriers correlation matrix. This work demonstrated that the self-created ICI in SEFDM actually constitutes a form of spectral leakage and deliberate aliasing. It was also shown that the ill-conditioning of a matrix in conjunction with machine precision determine the accuracy of the results acquired from matrix computations. The practical implications associated with SEFDM transmitter architectures were analysed and indicated that a Type 1 SEFDM transmitter and its reciprocal, offered a balanced trade-off between resource requirements and frequency spacing granularity.

Chapter 5 considered the effect of oversampling on the conditioning of the sub-carriers correlation matrix. It was found that a factor of at least $\rho = 2$ significantly ameliorates the conditioning of this matrix, as well as the BER performance of linear detectors, at the expense of throughput. It was shown that MF alone can achieve optimum error performance when $\alpha = 0.5$ for real-valued modulation schemes, otherwise IC techniques offer the best performance. Notwithstanding, TSVD was selected as the preferred method for hardware implementation thanks to its straightforward operation and better BER performance when combined with FSD. The latter part of this chapter assessed the impact of limited bit precision and round-off errors on the performance of SEFDM systems. It was found that SEFDM performs well in a fixed-point environment

when $\alpha \geq 0.8$. However, when $\alpha < 0.8$, the bit precision mandated grows beyond the standard resolution offered by most D/A and A/D converters.

Chapter 6 detailed the process for taking a design from concept to fruition. It was shown that the proposed architecture is scalable and reconfigurable and occupies less than 50% of a Xilinx Virtex 5 FPGA chip. Results verified the functionality of the FPGA implementation and demonstrated that the error performance matched that obtained from theory. The reconfigurable nature of the FFT block offers the additional benefit of adapting dynamically the number of sub-carriers and oversampling ratio in accordance with the instantaneous channel conditions, sacrificing BER performance for throughput or vice versa. However, the key limitation of the implemented design is that the demodulated data is acquired in burst periods. This is attributed to the fact that the architecture has not been optimised from a circuit point of view to achieve maximum throughput. Instead, the focus was on leveraging the available FPGA resources in order to yield a high performance design with a balanced impact on resource utilisation and latency.

The poor performance of linear detectors for complex modulation schemes motivated the examination of more sophisticated algorithms in Chapter 7. It was found that the sorting stage in the FSD algorithm constitutes the main bottleneck hindering its execution speed. A sort-free strategy was therefore adopted which was shown to provide up to six times greater speed efficiency compared to the original FSD technique. This gain, however, came at the expense of increased error penalties. A modified real-valued decomposition was therefore applied which ameliorated error performance. Experimental results indicated that the dimension of the SEFDM problem presents significant challenges compared to the application of similar techniques in MIMO systems. Hence, a joint FPGA-DSP arrangement, based on a commercially available platform, was designed and implemented to serve as a testbed. Using this arrangement, the performance of a hybrid TSVD-FSD detector was evaluated, for the first time, in a practical environment. Yet, the capabilities of the platform were limited by the out-

dated communication interface between the FPGA and the DSP, thereby restricting the range of results which could be acquired.

To conclude, it is worth noting that the SEFDM work described in this thesis assumed an ideal channel affected only by random noise and hence random errors. This is, of course, far from realistic. In a real-world environment, channel imperfections, external interference and non-Gaussian noise effects result in further signal degradation and may lead to complete channel loss or burst errors. As this thesis aimed to address the fundamentals of SEFDM and its ‘proof of concept’ implementations, such RF effects were not taken into account. Notwithstanding, it is important that these issues are dealt with in the near future, as proposed in Section 8.3, to allow SEFDM to be considered for practical communication system applications.

8.1 Thesis Key Achievements

This thesis is an amalgamation of multiple projects carried out during the EngD programme, thus the scope is considerably wider compared to a conventional doctorate thesis. Consequently, the work presents a breadth of knowledge rather than depth. While it would be intriguing to examine each topic in greater detail, this thesis has instead brought together the latest technologies aimed at improving spectrum utilisation. The challenges were considered from diverse angles including mathematical properties, algorithm evaluation and optimisation, practical considerations, hardware implementation, as well as hybrid verification and performance assessment. Thence, the key achievements may be summarised as follows:

- Detailed study examining the feasibility of employing SEFDM in the real world.
- Optimisation of SEFDM detection methods to yield tangible complexity.
- Design and implementation of a reconfigurable FPGA based SEFDM receiver employing linear detection.

- Hardware realisation of the best performing SEFDM detector to date using a hybrid FPGA-DSP platform.
- Demonstration of realistic estimates of the true hardware complexity and execution speed of different SEFDM detection algorithms.

To the best of the author's knowledge, the aforementioned hardware prototypes developed for the reception of SEFDM signals, constitute a world first.

8.2 Ongoing Related Work

The continuation of the hardware implementation of advanced SEFDM detection techniques is currently underway at UCL. Following on from the FPGA implementation of the original FSD and its reduced complexity variants [188], the next step entails two phases; first, the integration of this design with the FPGA design of the TSVD detector presented in Chapter 6; second, the addition of an A/D converter to the signal chain, thus yielding a complete SEFDM receiver as illustrated in Fig. 5.1. The ultimate goal is to devise an end-to-end SEFDM transceiver using a state-of-the-art FPGA platform. One of the crucial aspects of this platform which must be addressed early on, is the connectivity between the FPGA devices and the computer terminal, the latter used for programming the devices and capturing data for further analysis.

Research is also being carried out in developing novel iterative techniques which employ soft decision to address the detection challenge in SEFDM. Preliminary results [135] demonstrate that these iterative detectors have the potential to offer better BER performance compared to existing schemes without incurring complexity penalties.

Recently, the use of SEFDM in optical communications has also been proposed. This idea was motivated by similar research considering the application of FTN techniques to long-haul optical links [55]. Preliminary work though has indicated that fundamental issues of timing synchronisation and frequency offsets have to be addressed prior to applying the sophisticated detection techniques studied in this thesis.

8.3 Proposals for Future Work

As is the case with most projects, this work has generated a number of questions which would benefit from future research and experimentation. Interesting avenues include:

Mathematical Modelling:

- An analytical study to better comprehend the properties of the subcarriers correlation matrix \mathbf{C} is required. Specifically, the upper bounds of N , α and ρ which determine the conditioning of \mathbf{C} have to be considered in conjunction. This is important since the ill-conditioning of \mathbf{C} eventually transforms the matrix from positive definite to positive semi-definite. As a result, in the latter case, decomposition methods, such as Cholesky factorisation, cannot be applied. Subsequently, detection techniques like FSD, which mandate this factorisation, cannot be used. It would be beneficial if the conditioning of \mathbf{C} was considered in isolation from the SEFDM system as a general ill-posed problem. Hence, techniques aimed at improving the conditioning of the matrix, for example pivoting and preconditioners [144], may be applied.

Signal Processing:

This thesis has shown that selecting an appropriate SEFDM detector is coupled with complex trade-offs. The main pitfall of techniques like FSD, is the exponential increase in complexity in proportion to the dimension of the system. Thereby, alternative techniques may be considered, such as:

- **Stream ordering:** In MIMO systems, it has been shown that channel ordering can significantly improve error performance with tangible computational complexity. An example is the popular V-BLAST algorithm [61]. In line with this observation, it would be interesting to assess if similar ordering techniques could be applied in SEFDM by exploiting certain properties of the sub-carriers correlation matrix, for example, the norm of its rows.

- **Iterative detection:** Chapter 5 demonstrated that iterative cancellation techniques operating as standalone detectors perform better than their linear counterparts. Recent work [135] has also shown that iterative detection with soft decision can achieve quasi-optimum performance, both as a standalone detector, as well as in hybrid format when combined with other methods. Detection techniques which employ decision feedback equalisation or interference cancellation, while alleviating the need for matrix inversion, have been considered in similar non-orthogonal multi-carrier systems [111] [154]. Consequently, iterative techniques incorporating soft decoding may prove to be a promising solution.
- **Error correction:** Error correcting codes, such as turbo codes and Low Density Parity Check (LDPC) codes, could also be considered in conjunction with iterative detection techniques, since it has been shown [153] that the use of these codes improves the performance of FTN signalling systems.

Practical Experimentation:

Hitherto, most of the work on SEFDM has assumed an AWGN channel. If SEFDM is to compete with alternative technologies in wireless communication systems, additional scenarios have to be considered, such as:

- **Multipath fading:** Although related work has considered the joint channel equalisation and detection of SEFDM signals in multipath channels [215] [216], this still constitutes an important research gap, hence studies in this area should be intensified. In addition, the evaluation of SEFDM for higher order modulation schemes beyond 4-QAM and for a larger number of sub-carriers ($N > 100$), is instrumental for its success. This is especially true in view of the fact that alternative non-orthogonal systems, such as FTN [177] and GFDM [111], have already acquired results for system sizes exceeding 100 sub-carriers.
- **RF effects:** An interesting study would be the comparison of SEFDM to OFDM in a real environment where a number of practical non-idealities are present. Such

non-linearities include; DC coupling and filtering, signal clipping, timing jitter, frequency offsets, as well as loading from measurement equipment. This study should also consider channel impairments and the possibility of burst errors, as well as the impact these effects have on the behaviour and performance of the SEFDM detectors examined in this thesis.

Hardware Implementation:

As outlined in Section 8.2, the hardware implementation of end-to-end SEFDM transceivers is being pursued. As such, recommendations for future development include:

- Implementing an iterative detector using FPGAs and comparing its true hardware complexity to existing realisations of linear detection methods.
- Developing a multi-channel architecture allowing multiple SEFDM signal streams to be decoded simultaneously, thus yielding significant throughput gains.
- Migrating to higher level tools, such as Simulink, to alleviate the need for expert knowledge required at the hardware design level.
- Enriching the platform with hardware-in-the-loop capabilities to expedite the performance evaluation of diverse SEFDM transceiver configurations.

Other Applications:

- The principles of SEFDM have found application in other domains, such as physical layer security [31]. Furthermore, non-orthogonal systems like GFDM have been shown to exhibit excellent cyclostationary properties [103] for use in CR applications. Consequently, it would be worthwhile extending the application of SEFDM to timely research areas including but not limited to CR and multiple access systems.

In conclusion, this thesis confirmed the practicability of SEFDM for moderate system sizes. The reasons for different design choices were explained. It was shown that there is

still room for optimisation at the circuit level, thereby favouring reduction in detection latency. The designed and implemented testbed has opened up new opportunities for further research, development and experimentation. Notwithstanding, if SEFDM is to compete with OFDM and similar systems sharing common objectives, its performance in fading channels for a larger system dimension has to be evaluated. In all cases, there will always be a trade-off between multiple performance metrics, including throughput, BER performance and silicon area requirements.

Appendix A

SEFDM Transceiver MATLAB GUI Datasheet

This appendix presents a GUI developed in MATLAB for the purpose of simulating OFDM and SEFDM systems. A screenshot of the GUI startup screen is illustrated at the end of this appendix in Fig. A.9. While most of the settings required to configure this GUI are self-explanatory, this appendix gives an overview of the different choices available with the aid of examples.

First, Fig. A.1 depicts the key parameters which dictate the performance of the system. These include the number of sub-carriers N , the bandwidth compression or reduction in frequency spacing α , as well as the type and size of the modulation scheme employed being either QAM or PSK. The user may also select the number of data elements to be generated in the form of symbols or bits.

Fig. A.2 illustrates the options available to the user with regard to the type of signal generated and the level of noise, if any. The signal may be generated using either a bank of modulators (referred to as the analogue method) or an IFFT process. An optional CP may also be added to the signal followed by windowing if desired.

The *Advanced Options* area illustrated in Fig. A.9 allow the user to calibrate the time axis and/or modify the amount of zero padding applied at the input of the mod-

Input Parameters

Input Data Type: Bits

Number of elements: 1600

Number of subcarriers: 16

Mapping: QAM

Alphabet Size: 2^4

Signal Frequency (Hz): 1000000

Frequency Distance (alpha): 1

Figure A.1: Area allowing key parameters, such as the number of sub-carriers N , the frequency spacing α and the modulation scheme to be configured.

Processing Choices

Signal Type: Analogue&Digital

Cyclic Prefix: Analogue

CP Value (between 0 and 1): Analogue&Digital

Noise Addition: Enabled

SNR: 20

Windowing: Disabled

Figure A.2: Area allowing the type of signal and level of noise to be configured.

ulator and demodulator blocks. The default options ensure that a sufficient resolution is employed for up to 64 sub-carriers. Specifically, the *IFFT Size* determines the total number of time samples that will be present in the time domain signal with the amount of oversampling equal to $\rho = \frac{\alpha \cdot IFFTSize}{N}$.

Having selected the desired settings, the user may verify these options by clicking the corresponding push buttons. For example, clicking *View Summary of Choices* will display the message box illustrated in Fig. A.3, whereas clicking *View Summary of Advanced Options* will display the message box shown in Fig. A.4. Subsequently, the simulation is initiated by clicking the push button *RUN THE SIMULATION*.

Once the simulation process has ended, the pop-up window depicted in Fig. A.5 is

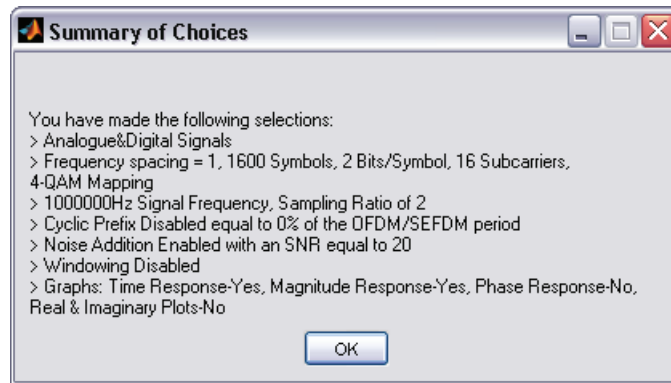


Figure A.3: Message box illustrating a summary of the general choices made.

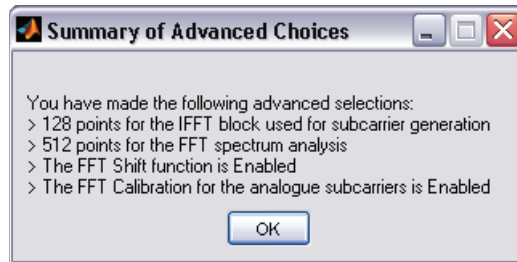


Figure A.4: Message box illustrating a summary of the advanced options selected.

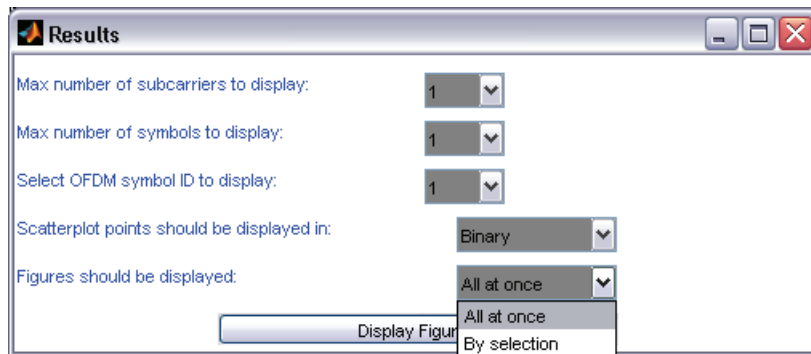


Figure A.5: Pop-up window allowing a specific set of results to be examined.

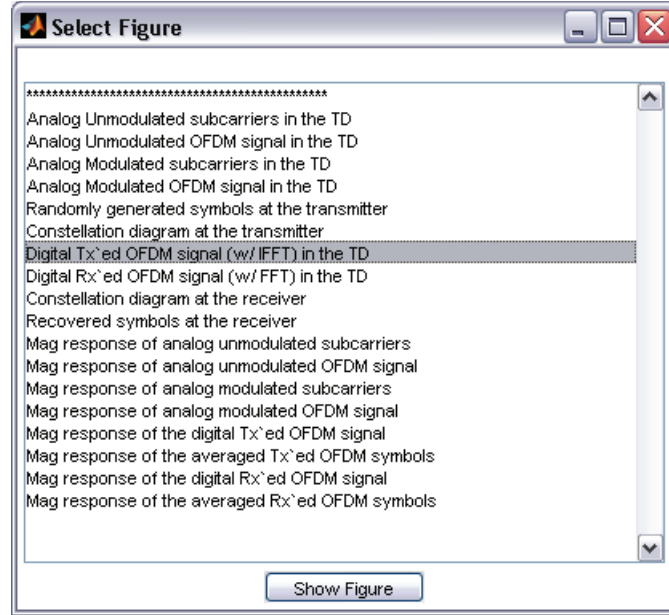


Figure A.6: Pop-up window allowing specific figures to be displayed.

presented to the user. This particular screen allows the user to control the number of figures generated, as well as the data that appears within these figures. For example, if the individual sub-carriers comprising an OFDM signal were to be displayed in the time domain, it should be evident that any number higher than eight sub-carriers would degrade the clarity of presentation. If the figures are to be displayed *By selection*, the window illustrated in Fig. A.6 pops up. This screen allows the user to view a list of the figures available for display and select manually the figure of interest. For example, the user may wish to examine the time domain signal generated at the transmitter and the magnitude response of the same signal at the receiver following propagation through a noisy channel. These examples are depicted in Figs. A.7 and A.8, respectively.

To conclude, this GUI makes it easy for the user to modify the underlying interface or transceiver code, as shown in Fig. A.9. This interface would benefit from additional functionality which would enable the user to select different channel models and generate BER estimates.

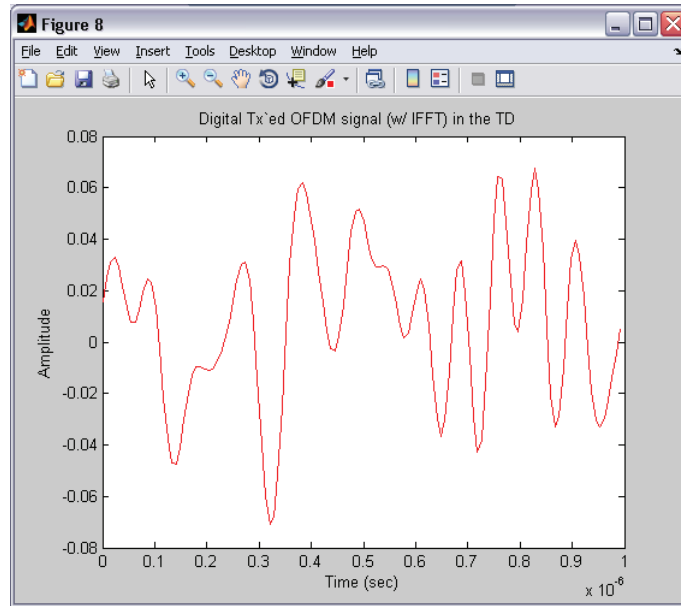


Figure A.7: Example of a time domain OFDM signal generated using this GUI.

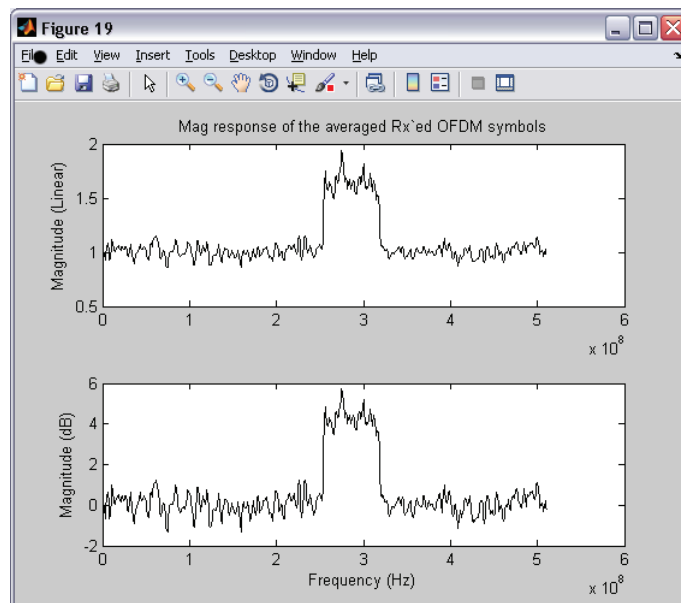
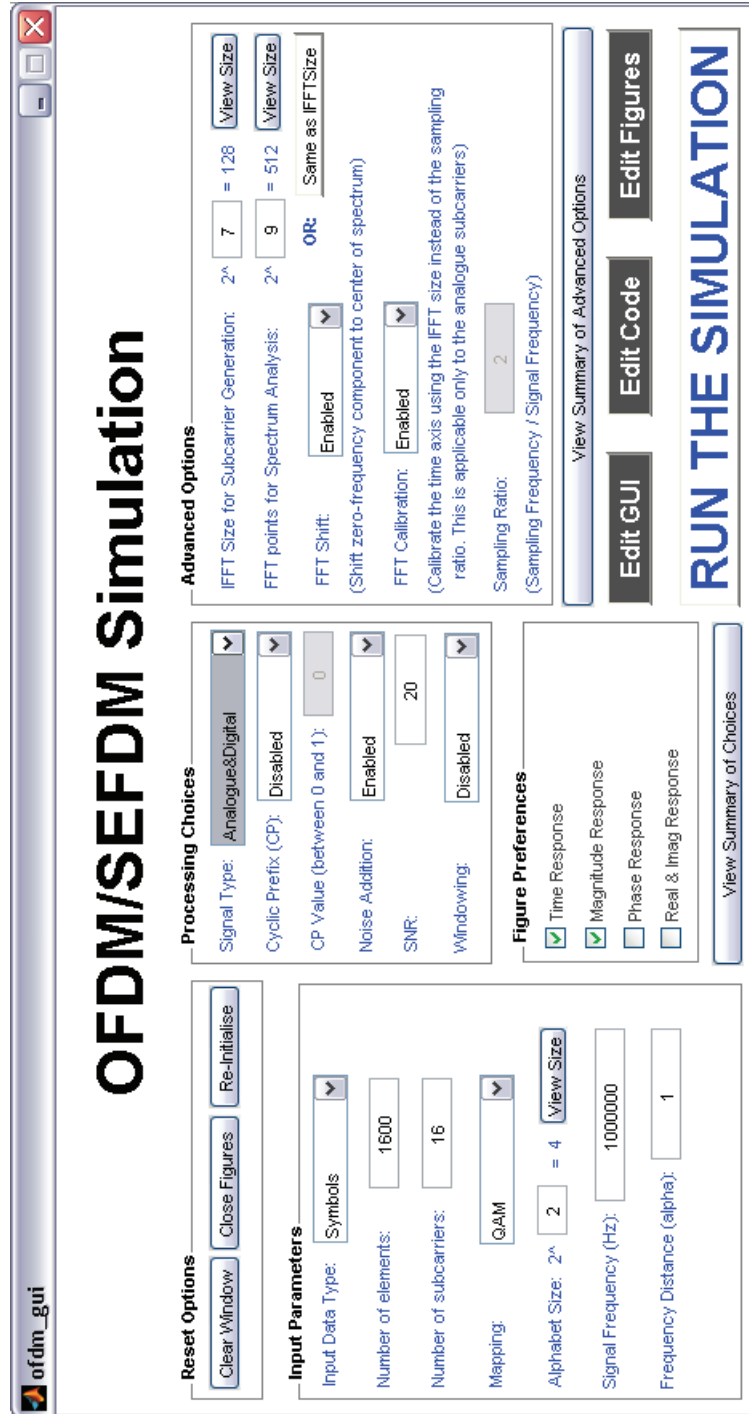


Figure A.8: Example of an OFDM signal's magnitude response generated using this GUI.



ofdm_gui

OFDM/SEFDM Simulation

Reset Options

Clear Window Close Figures Re-Initialise

Input Parameters

Input Data Type: Symbols

Number of elements: 1600

Number of subcarriers: 16

Mapping: QAM

Alphabet Size: $2^4 = 4$ View Size

Signal Frequency (Hz): 1000000

Frequency Distance (alpha): 1

Processing Choices

Signal Type: Analogue&Digital

Cyclic Prefix (CP): Disabled

CP Value (between 0 and 1): 0

Noise Addition: Enabled

SNR: 20

Windowing: Disabled

Advanced Options

FFT Size for Subcarrier Generation: $2^7 = 128$ View Size

FFT points for Spectrum Analysis: $2^9 = 512$ View Size

FFT Shift: Enabled OR: Same as FFT Size

(Shift zero-frequency component to center of spectrum)

FFT Calibration: Enabled

(Calibrate the time axis using the IFFT size instead of the sampling ratio. This is applicable only to the analogue subcarriers)

Sampling Ratio: 2

(Sampling Frequency / Signal Frequency)

Figure Preferences

☒ Time Response

☒ Magnitude Response

☐ Phase Response

☐ Real & Imag Response

View Summary of Choices

Edit GUI Edit Code Edit Figures

RUN THE SIMULATION

Figure A.9: GUI startup screen.

Appendix B

Joint FPGA-DSP Platform Aspects and Operation

This appendix details the architecture and real-time operation of the development platform employed for the realisation of different SEFDM receivers described in Chapters 6 and 7. Section B.1 describes the FPGA implementation cycle from the Register Transfer Logic (RTL) design, through to synthesis, translation, placing and routing. Section B.2 provides an overview of the platform and the receiver's underlying logic. Section B.3 shows that the system operates correctly, as demonstrated through functional simulation and real-time verification.

B.1 FPGA Design Cycle

A proper design flow is indispensable for cutting back development time and minimising the risk of errors. While a design can be tested against diverse criteria, this work considers bandwidth savings, throughput and BER performance. Design parameters include bandwidth compression levels, the number of sub-carriers employed, oversampling factors, bit precision and transform lengths. The design parameters have to be configured appropriately to meet the desired performance criteria while concurrently

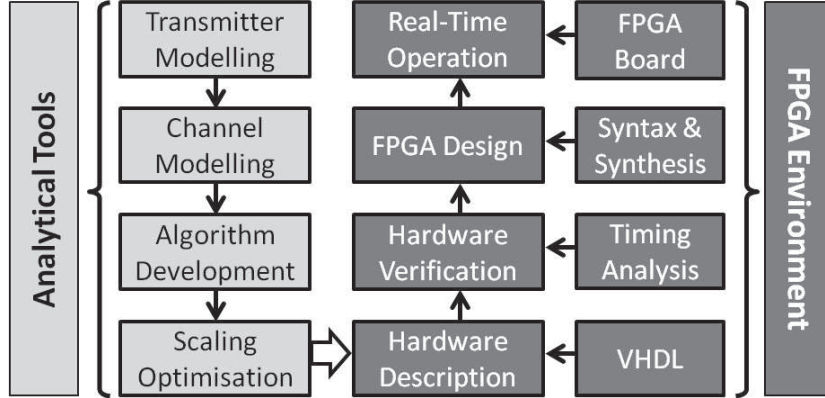


Figure B.1: Design flow.

ensuring that the design can fit on the available FPGA device. Different aspects may affect the performance in different ways, for example, pipelining can affect the maximum clock frequency, oversampling can affect the effective data rate, quantisation noise can affect error performance, whereas dynamic range, bit precision and transform lengths can have a significant impact on resource utilisation [181].

The design flow for creating and validating FPGA architectures is illustrated in Fig. B.1. A number of hardware and software tools were utilised to build and verify these FPGA designs. The following paragraphs explain the phases involved during the implementation of an FPGA design which entail behavioural modelling, structural architecting, functional simulation, design synthesis and real-time verification.

The first phase involves the creation of a model which emulates the behaviour of the actual hardware IP core analogous to the one examined in Chapter 5. This model is built with the aid of mathematical tools whose use is extended to:

- Implementing the functionality of the transmitter and the channel before the receiver.
- Developing and testing the different SEFDM detection algorithms.
- Automating the search to identify optimum parameters which meet specified criteria.

- Validating the correctness of the system and evaluating its performance.

The next phase entails writing the VHDL code which describes the receiver architecture. The industry standard *Xilinx ISE[®] Design Suite* (henceforth referred to as ISE) is used as the coding environment and is responsible for compiling the VHDL code and synthesizing it into a hardware circuit realisation. This suite comprises the *CORE GeneratorTM System* (henceforth referred to as COREGEN) tool which generates highly efficient fixed-point hardware for standard communication blocks and mathematical functions, such as FFT transforms and complex multiplications [163]. During synthesis and implementation, the ISE tools apply timing constraints and attempt to yield an efficient resource utilisation. The actual steps may be summarised as follows:

1. **Design entry:** During this initial stage, the architecture of the design is constructed using VHDL. The target FPGA pinouts are also assigned to the design signals.
2. **Synthesis:** This stage is responsible for converting the Hardware Description Language (HDL) design into a gate-level netlist. This netlist essentially defines the connectivity between the different FPGA blocks. Prior to moving to the implementation stage, a behavioural verification of the RTL design is carried out which yields three important bits of information:
 - It allows the hardware designer to compare the FPGA design outputs against the theoretical values.
 - It identifies timing and area constraints.
 - It gives a first estimate of the maximum achievable clock frequency.
3. **Implementation:** This final stage consists of three steps all of which contribute to translating the netlist into a *placed and routed* FPGA layout.

- (a) **Translate:** The gate-level netlist is converted to a Native Generic Database (NGD) netlist which contains approximate information regarding switching delays.
- (b) **Map:** The NGD netlist is mapped onto the actual FPGA fabric, such as LUTs, FFs, BRAMs and DSP Slices. It also contains exact information regarding switching delays.
- (c) **Place and route (PAR):** This step is the most important and time-consuming part of the implementation process since it determines where the different modules are positioned and the manner in which they are connected inside the FPGA.

The last two steps of the implementation stage are responsible for ensuring the design meets timing constraints.

The final phase involves testing the design using industry standard timing analysis and debugging software, such as *ModelSim*[®] and *Xilinx ChipScope*TM *Pro* (henceforth referred to as ChipScope). The former performs a non real-time simulation of the design to verify the functionality of the HDL code. VHDL testbenches are utilised in this simulation allowing probe signals to be inserted at different stages in the design to monitor all internal and external signals. Each FPGA block is initially assessed in isolation prior to interconnecting all modules in cascade to test the overall system. The benefit of functional simulation is that it reduces the number of unsuccessful synthesis attempts, which can be very time-consuming given that a behaviourally-incorrect system would never work in a physical implementation. Subsequently, ChipScope is used for real-world testing. ChipScope is an invaluable tool at this stage as it allows signals to be captured and recorded in real-time alleviating the need for costly logic analyser equipment. ChipScope records cycle-accurate data and offers multiple triggering points to test different aspects of the FPGA implementation. It is capable of monitoring the device's inputs and outputs, as well as its internal states to help troubleshoot any er-

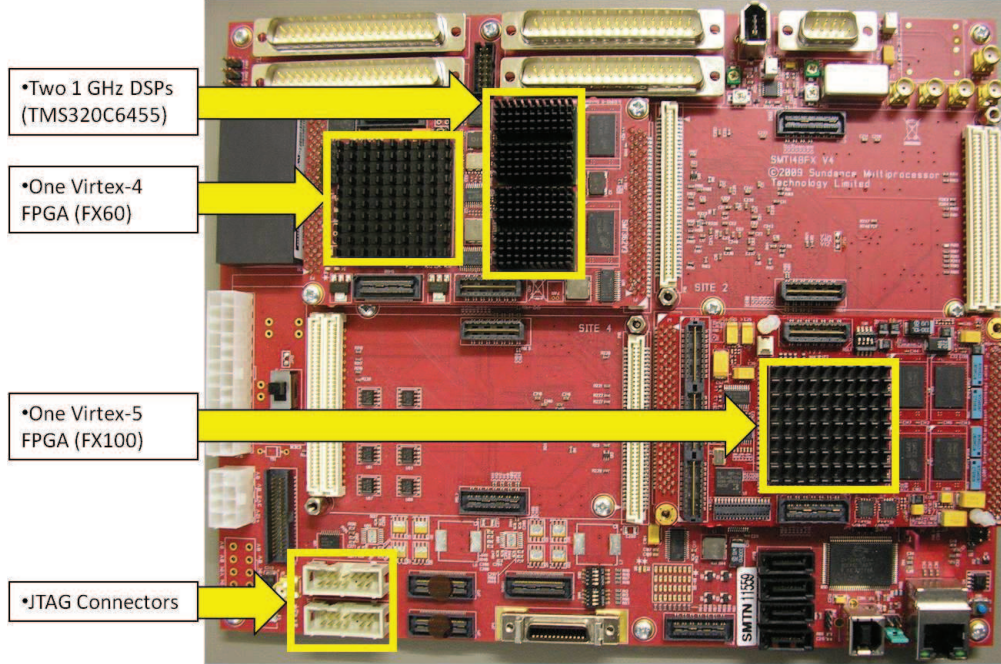


Figure B.2: Development platform by Sundance Multiprocessor Technology Ltd.

rors. Having generated a fully-working FPGA implementation, the outputs acquired are fed to a computer terminal for results analysis and system performance evaluation.

B.2 System Description

The prototyping board used in this work was procured from Sundance Multiprocessor Technology Ltd. A photographic image of this board is illustrated in Fig. B.2. Although not clearly visible due to the chips being concealed by heatsinks, this board is equipped with two FPGA and two DSP devices. The Virtex 5 chip is used to implement the FPGA based SEFDM receiver employing linear detection. The Virtex 4 chip is used to handle connectivity between the Virtex 5 device and the DSPs. Of the two DSP chips, one (DSP B) is used for acquiring data from the Virtex 5 device and for running the FSD algorithm, whereas the other (DSP A) contains code for configuring the Virtex 4 chip. This setup may be better comprehended by referring to the system's schematic diagram, depicted in Fig. B.3. The thick green arrow shows the path

Figure B.3: System block diagram (from vendor datasheet).

followed from the acquisition of data using an A/D converter to the point the digital samples arrive at the DSP, subject to any intermediate processing carried out by the FPGAs.

Fig. B.4 presents the structure of the FPGA based SEFDM receiver employing linear detection at the block level. The design entails multiple instances of different components configured in parallel, in order to process many data symbols concurrently. A number of control signals are employed to enable and synchronise the different blocks.

B.3 Functional Simulation and Real-time Verification

A number of tools were employed to verify the system's operation. The HDL simulator integrated within ISE (henceforth referred to as ISim) was used to carry out a functional verification of the design. Subsequently, ChipScope was used to test the system's operation in real time. Finally, *Code Composer StudioTM* (henceforth referred to as

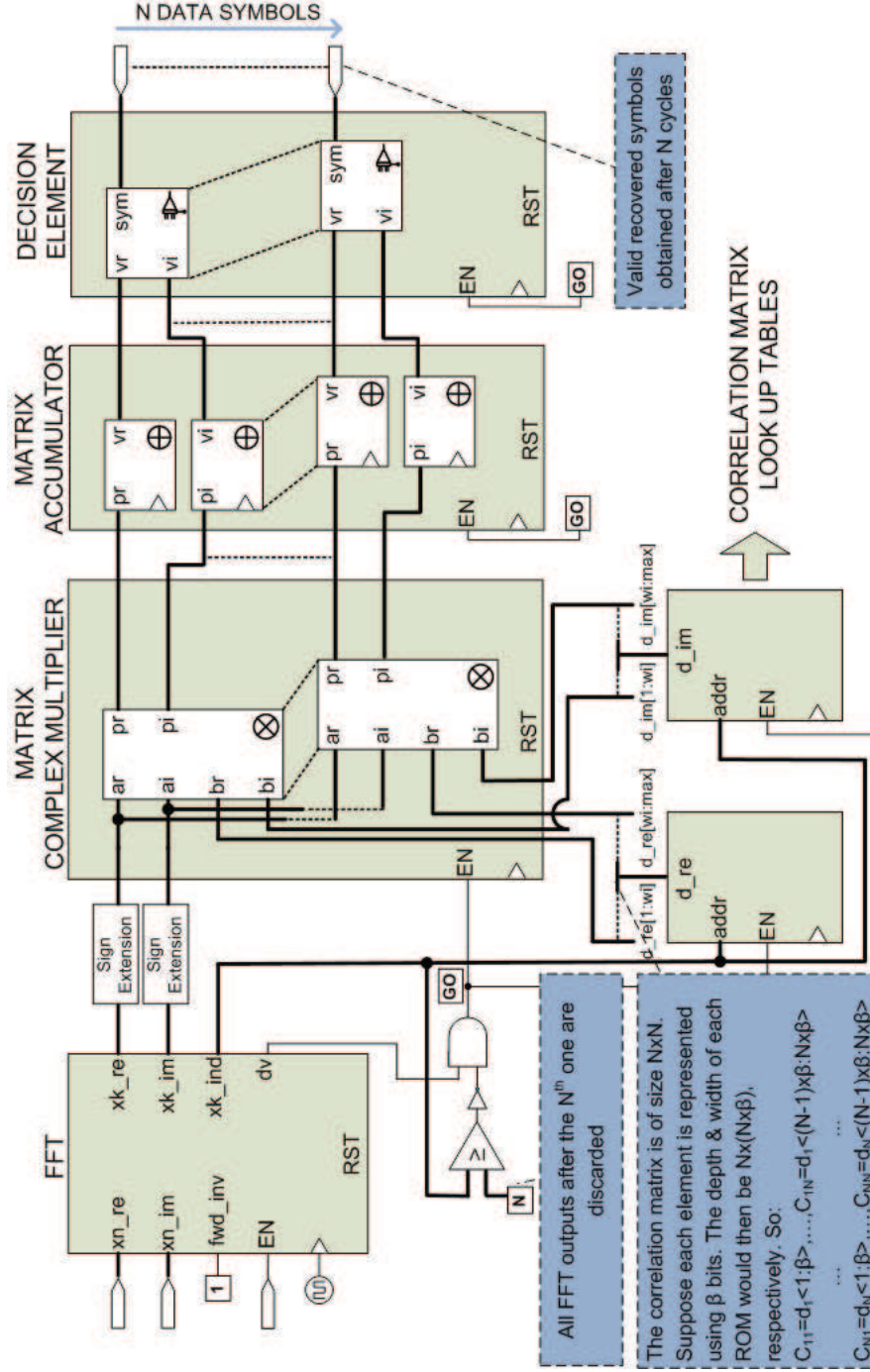


Figure B.4: Low-level architecture of the developed FPGA based SEFDM receiver.

```

43 -- CONSTANTS
44 -- Commands from DSP
45 constant CMD_EN          : std_logic_vector(3 downto 0) := "1100";
46 constant CMD_START       : std_logic_vector(3 downto 0) := "1101";
47 constant CMD_STOP        : std_logic_vector(3 downto 0) := "1110";
48 constant CMD_LOOP        : std_logic_vector(3 downto 0) := "1111";
49
50 constant GET_RXVEC_REAL   : std_logic_vector(3 downto 0) := "0001";
51 constant GET_RXVEC_IMAG   : std_logic_vector(3 downto 0) := "0010";
52 constant GET_SYMEST_REAL  : std_logic_vector(3 downto 0) := "0011";
53 constant GET_SYMEST_IMAG  : std_logic_vector(3 downto 0) := "0100";
54 constant GET_CYCLES_RXVEC : std_logic_vector(3 downto 0) := "0101";
55 constant GET_CYCLES_SYMEST : std_logic_vector(3 downto 0) := "0110";
56
57 -- Confirm commands to DSP
58 constant BUSY            : std_logic_vector(31 downto 0) := X"42555359";
59 constant ERROR           : std_logic_vector(31 downto 0) := X"45524f52";
60 constant ENOK            : std_logic_vector(31 downto 0) := X"454e4f4b";
61 constant GOOK            : std_logic_vector(31 downto 0) := X"474f4f4b";
62 constant STOP            : std_logic_vector(31 downto 0) := X"53544f50";
63 constant NFIN            : std_logic_vector(31 downto 0) := X"4e46494e";
64 constant NCMD            : std_logic_vector(31 downto 0) := X"4e434d44";
65 constant NVAL            : std_logic_vector(31 downto 0) := X"4e56414c";
66 constant DONE            : std_logic_vector(31 downto 0) := X"444f4e45";
67
68 -- SIGNALS
69 -- For the DSP CP Decode
70 signal execcmd, writeout_s, get_data : std_logic;
71 signal datafromDSP_32bit            : std_logic_vector(31 downto 0);
72 signal cmdfromDSP                    : std_logic_vector(3 downto 0);
73
74 -- For the SEFDM Rxer
75 --Inputs
76 signal rxer_rst, rxer_en            : std_logic;
77 signal rxer_commence, start_success : std_logic;
78 signal rxer_sel                      : std_logic_vector(3 downto 0);

```

Figure B.5: Code snapshot of the commands manually programmed in the FPGA for system control and data exchange with the DSP.

CCStudio) was used to test the functionality of the programme running on the DSP.

B.3.1 Control and Synchronisation

This section is aimed at verifying the correct synchronisation between the Virtex 5 FPGA device and DSP B. A snapshot of the different commands issued by the DSP to control the FPGA and handle the exchange of data, is presented in Fig. B.5. The FPGA responses transmitted to the DSP, for example *BUSY*, represent the ASCII equivalent of the corresponding hexadecimal value, for example X"42555359", thereby

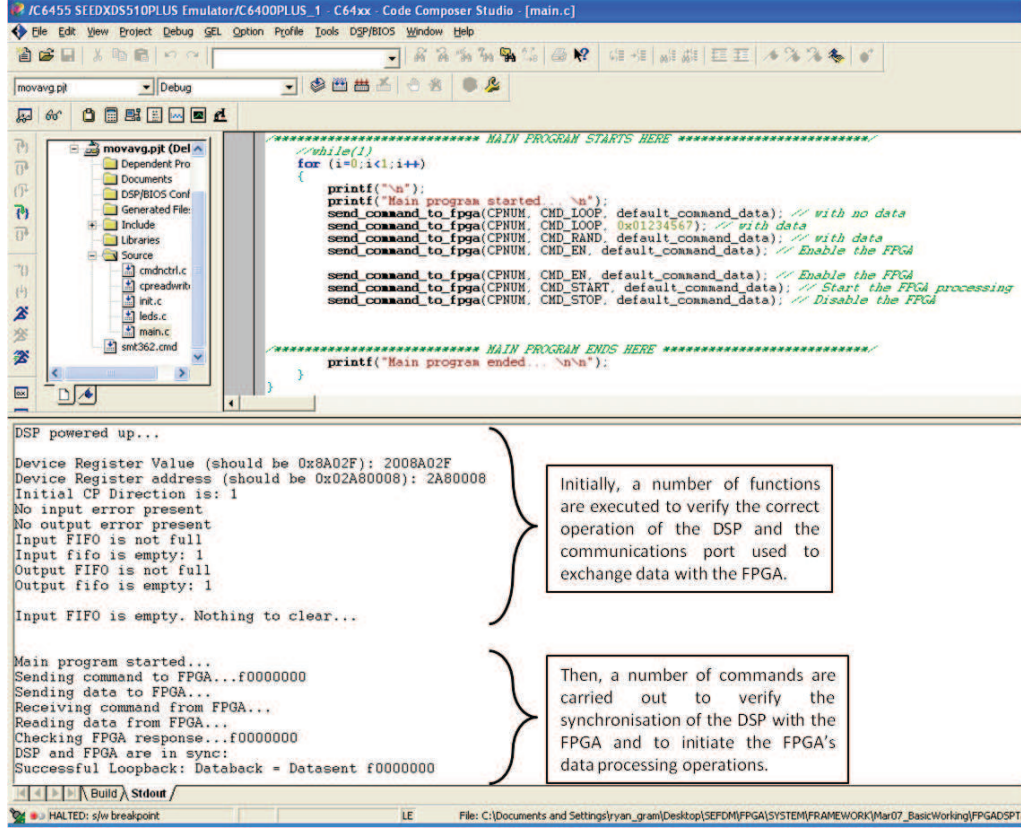


Figure B.6: Commands issued by the DSP in real time to verify system synchronisation and initiate data processing.

simplifying the troubleshooting process.

Fig. B.6 shows the DSP sending commands to the FPGA in real time. A detailed view of all the commands sent by the DSP and the responses received by the FPGA, for this particular case, is depicted in Fig. B.7. Fig. B.8 illustrates the outputs acquired in ISim by emulating the DSP commands in order to carry out a functional simulation of the FPGA design under consideration. Fig. B.8 also shows that each value arriving from the DSP, denoted by *datain*, is a 32-bit number of which the first four bits are used to construct the commands presented in Fig. B.5. Finally, Fig. B.9 demonstrates that the responses transmitted by the FPGA to the DSP, which were captured in real time using ChipScope, match those expected in accordance with Fig. B.7.

DSP Output		
DSP powered up...	Sending command to FPGA...f1234567	Sending command to FPGA...c0000000
Device Register Value (should be 0x8A02F): 2008A02F	Sending data to FPGA...	Sending data to FPGA...
Device Register address (should be 0x02A80008): 2A800008	Receiving command from FPGA...	Receiving command from FPGA...
Initial CP Direction is: 1	Reading data from FPGA...	Reading data from FPGA...
No input error present	Checking FPGA response...f1234567	Checking FPGA response...444f4e45
No output error present	DSP and FPGA are in sync:	That command has already been sent.
Input FIFO is not full	Successful Loopback: Databack = Datasent f1234567	Cannot repeat! 444f4e45
Input fifo is empty: 1	Command sent and received	Command sent and received
Output FIFO is not full	Sending command to FPGA...b0000000	Sending command to FPGA...d0000000
Output fifo is empty: 1	Sending data to FPGA...	Sending data to FPGA...
Input FIFO is empty. Nothing to clear...	Receiving command from FPGA...	Receiving command from FPGA...
Main program started...	Reading data from FPGA...	Reading data from FPGA...
Sending command to FPGA...f0000000	Checking FPGA response...4e434d44	Checking FPGA response...474f4f4b
Sending data to FPGA...	The command you requested has not been programmed!	DSP and FPGA are in sync:
Receiving command from FPGA...	DSP and FPGA are in sync:	FPGA processing successfully started:
Reading data from FPGA...	Not a valid command: 4e434d44	474f4f4b
Checking FPGA response...f0000000	Command sent and received	Command sent and received
DSP and FPGA are in sync:	Sending command to FPGA...c0000000	Sending command to FPGA...e0000000
Successful Loopback: Databack = Datasent f0000000	Sending data to FPGA...	Sending data to FPGA...
Command sent and received	Receiving command from FPGA...	Receiving command from FPGA...
	Reading data from FPGA...	Reading data from FPGA...
	Checking FPGA response...454e4f4b	Checking FPGA response...53544f50
	DSP and FPGA are in sync:	DSP and FPGA are in sync:
	FPGA successfully enabled: 454e4f4b	FPGA successfully disabled: 53544f50
	Command sent and received	Command sent and received
	Command sent and received	Main program ended...

Figure B.7: DSP output during system synchronisation (with reference to Fig. B.6).

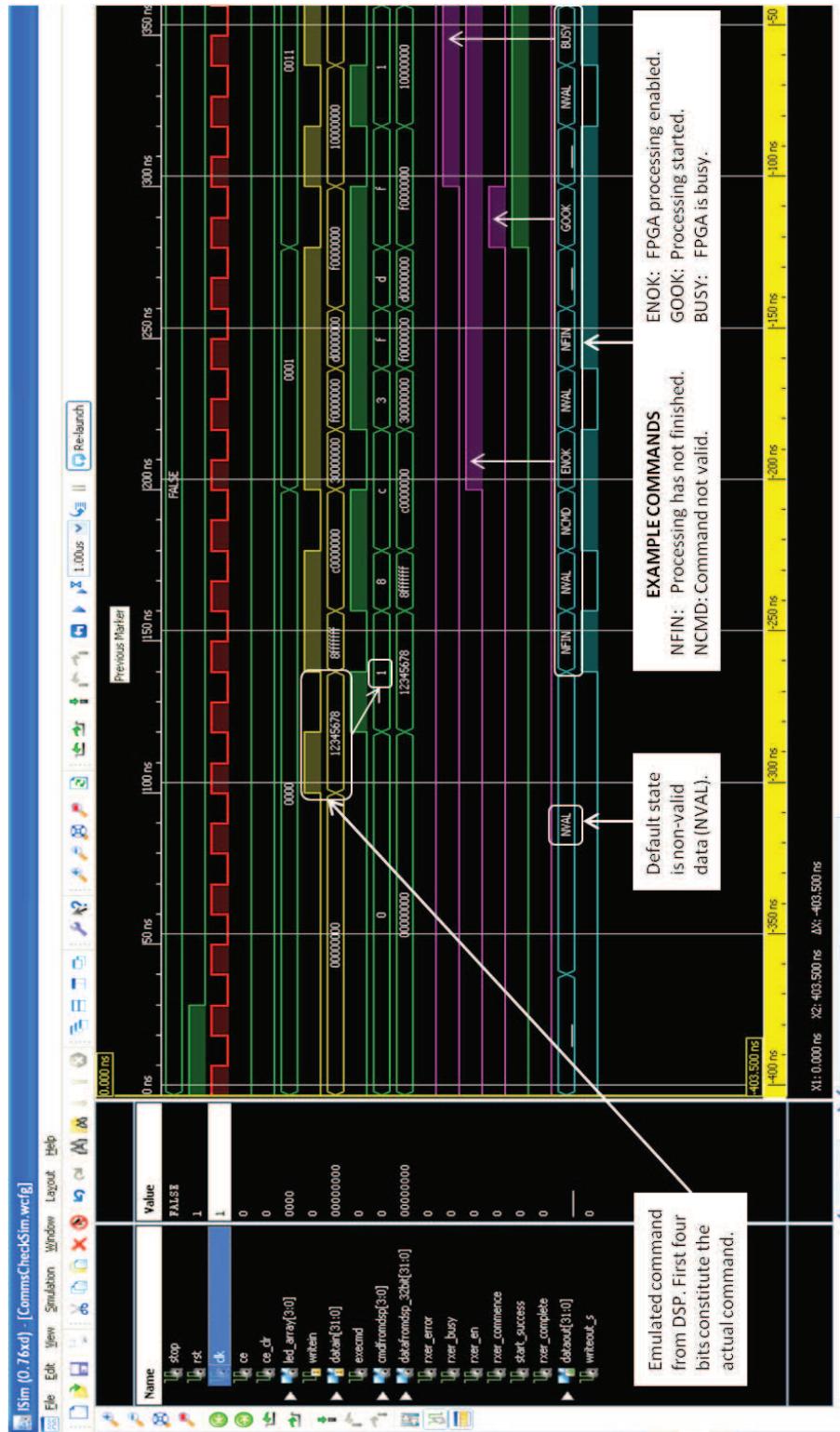


Figure B.8: Verification of FPGA control and system synchronisation through functional simulation.

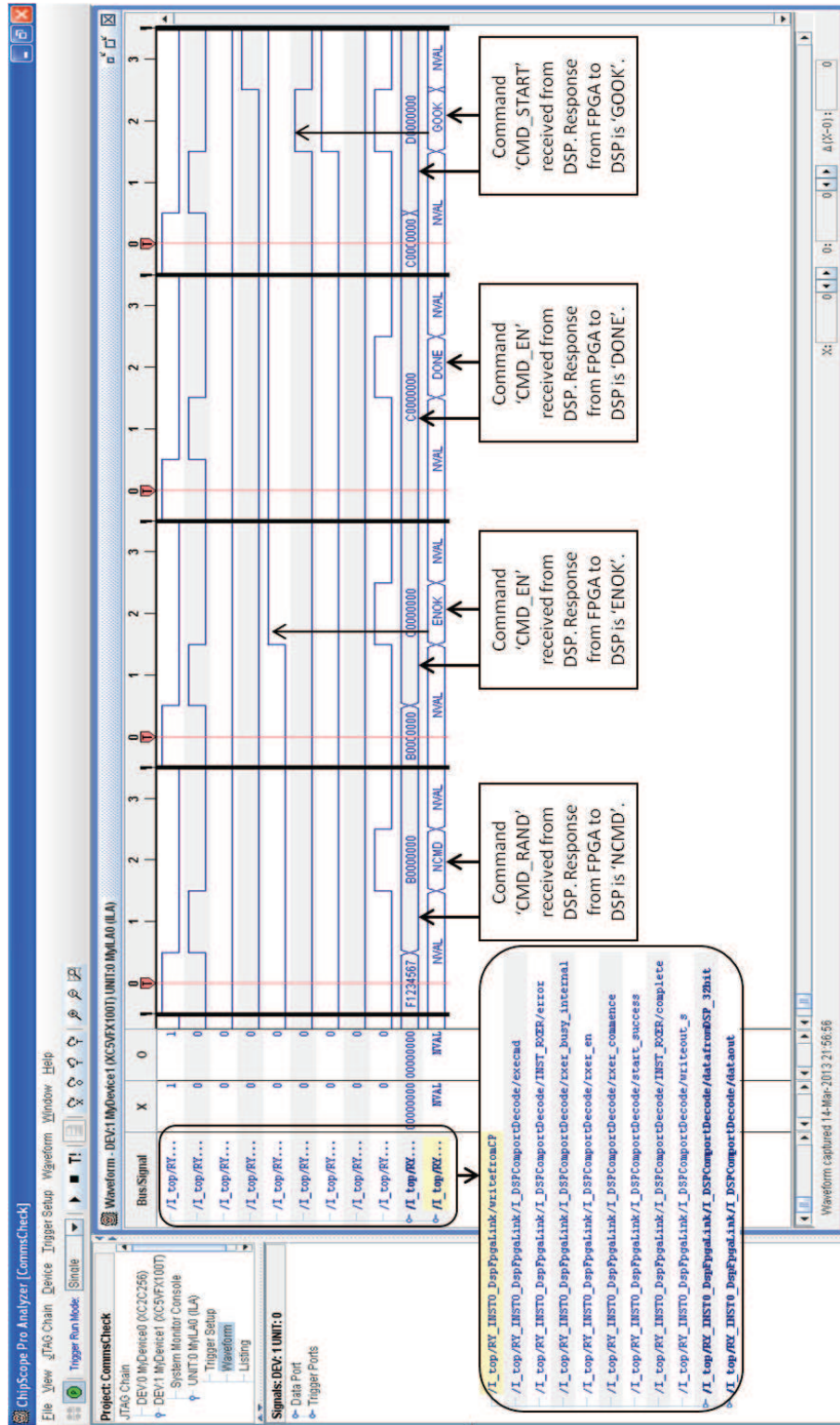


Figure B.9: Verification of FPGA control and system synchronisation during real-time operation.

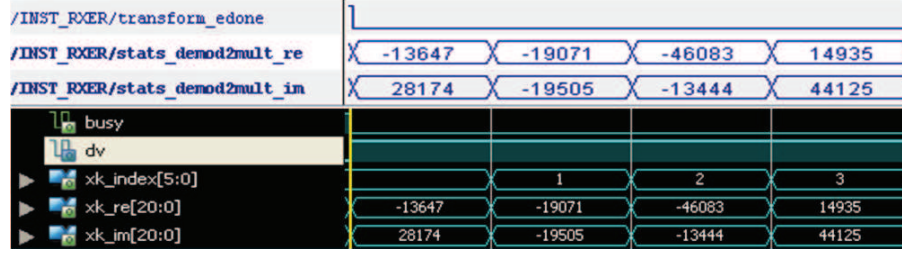


Figure B.10: Values of the received statistics vector \mathbf{R} acquired from the functional simulation (bottom) and real-time operation (top) of an FPGA based SEFDM receiver employing TSVD.

B.3.2 Data Exchange

This section compares the results acquired from the FPGA and DSP data processing stages to those generated in MATLAB¹. Fig. B.11 shows the data values generated at different stages of an SEFDM transceiver employing TSVD, simulated in MATLAB for 16 sub-carriers modulated using 4-QAM. Fig. B.10 depicts the first four FFT outputs obtained during the functional simulation of the FPGA based SEFDM receiver in ISim followed by its real-time verification in ChipScope. A fraction of the total number of these FFT outputs constitutes the received statistics vector \mathbf{R} . Fig. B.10 also demonstrates that these results match the corresponding theoretical values shown in the fourth column of Fig. B.11.

Having generated \mathbf{R} , the values of this vector have to be transferred to the DSP for further processing and data analysis. Due to the construction of the development platform itself, these values can only be transmitted one after the other in a sequential fashion, as illustrated in Fig. B.12. Similar results are depicted in Fig. B.13 for the TSVD estimated symbols. Fig. B.13 demonstrates that the unconstrained TSVD symbol estimates match those in the fifth column of Fig. B.11. Due to limitations of the ISim software, negative values with a very large magnitude cannot be displayed.

A slicing unit is then responsible for generating the constrained TSVD symbol

¹Note that for the figures combining screenshots of ISim and ChipScope outputs, the partition with a black background corresponds to the waveforms generated in ISim.

Generated Symbols	Mapped Data Symbols	Unscaled FFT Output (Statistics Vector)	Scaled FFT Output (Statistics Vector)	TSVD Estimate (Unconstrained)
0	-1.0000 + 1.0000i	-3.3319	-13648	-6.9022e+012
1	-1.0000 - 1.0000i	-4.6555	-19069	-8.2606e+012
1	-1.0000 - 1.0000i	-11.251	-46085	-1.027e+013
2	1.0000 + 1.0000i	3.6461	14934	3.3192e+012
0	-1.0000 + 1.0000i	-13.527	-55405	-1.2769e+013
2	1.0000 + 1.0000i	6.2071	25424	2.8902e+012
2	1.0000 + 1.0000i	6.4705	26503	5.0622e+012
2	1.0000 + 1.0000i	9.1446	37456	7.9441e+012
1	-1.0000 - 1.0000i	-5.6379	-23093	-5.9208e+012
0	1.0000 + 1.0000i	-5.7367	-23497	-6.3991e+012
2	1.0000 + 1.0000i	10.698	43818	9.0025e+012
1	-1.0000 - 1.0000i	-5.4861	-22471	-3.8656e+012
0	-1.0000 + 1.0000i	-0.5465	-2238.5	-1.6258e+012
1	-1.0000 - 1.0000i	-1.4575	-5970	-2.6576e+012
1	-1.0000 - 1.0000i	-8.905	-36475	-6.2331e+012
2	1.0000 + 1.0000i	4.9077	20102	6.5023e+012

Figure B.11: Results acquired from the simulation of SEFDM employing TSVD in MATLAB.

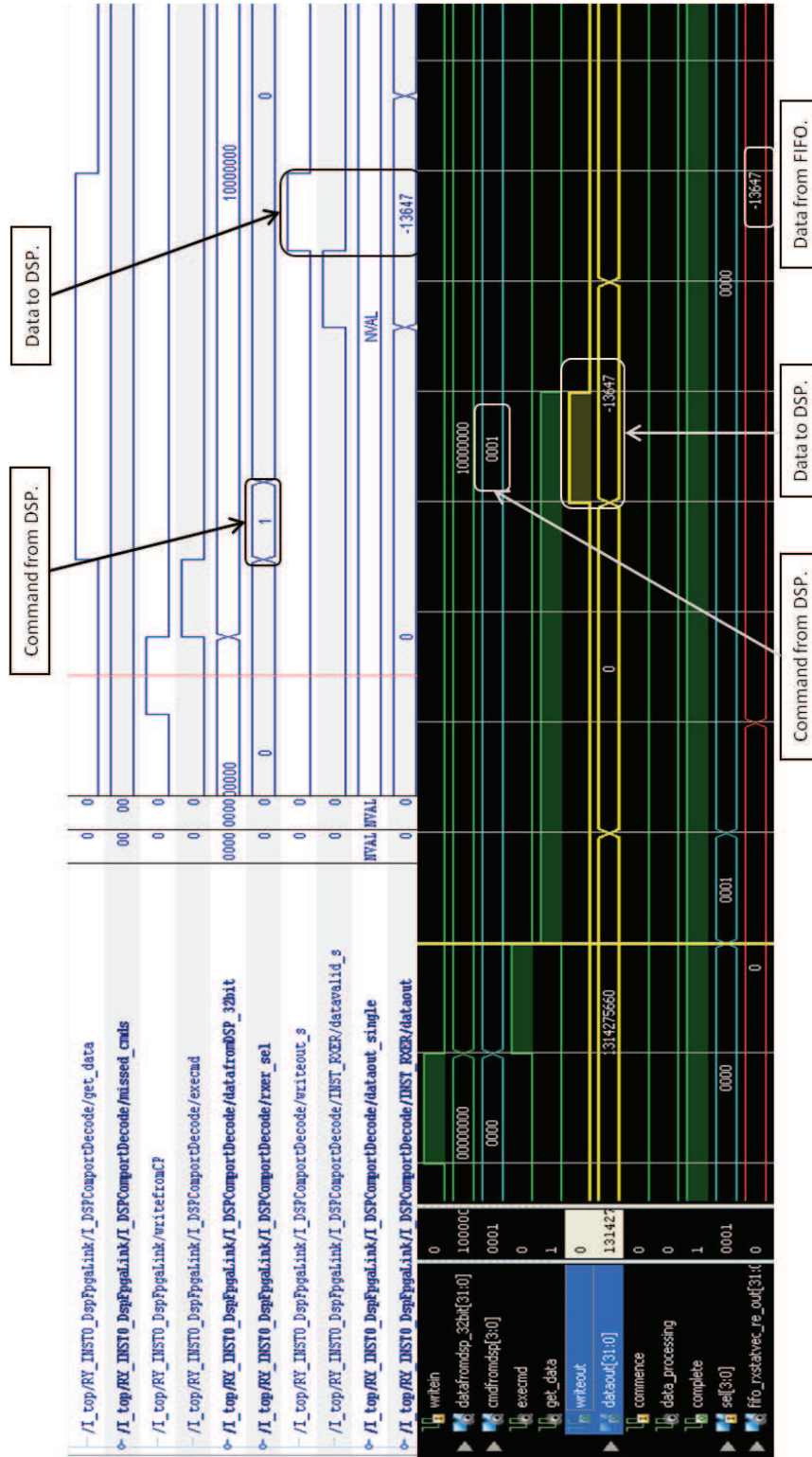


Figure B.12: Verification of the data transfer of **R** from the FPGA to the DSP during functional simulation (bottom) and real-time operation (top).

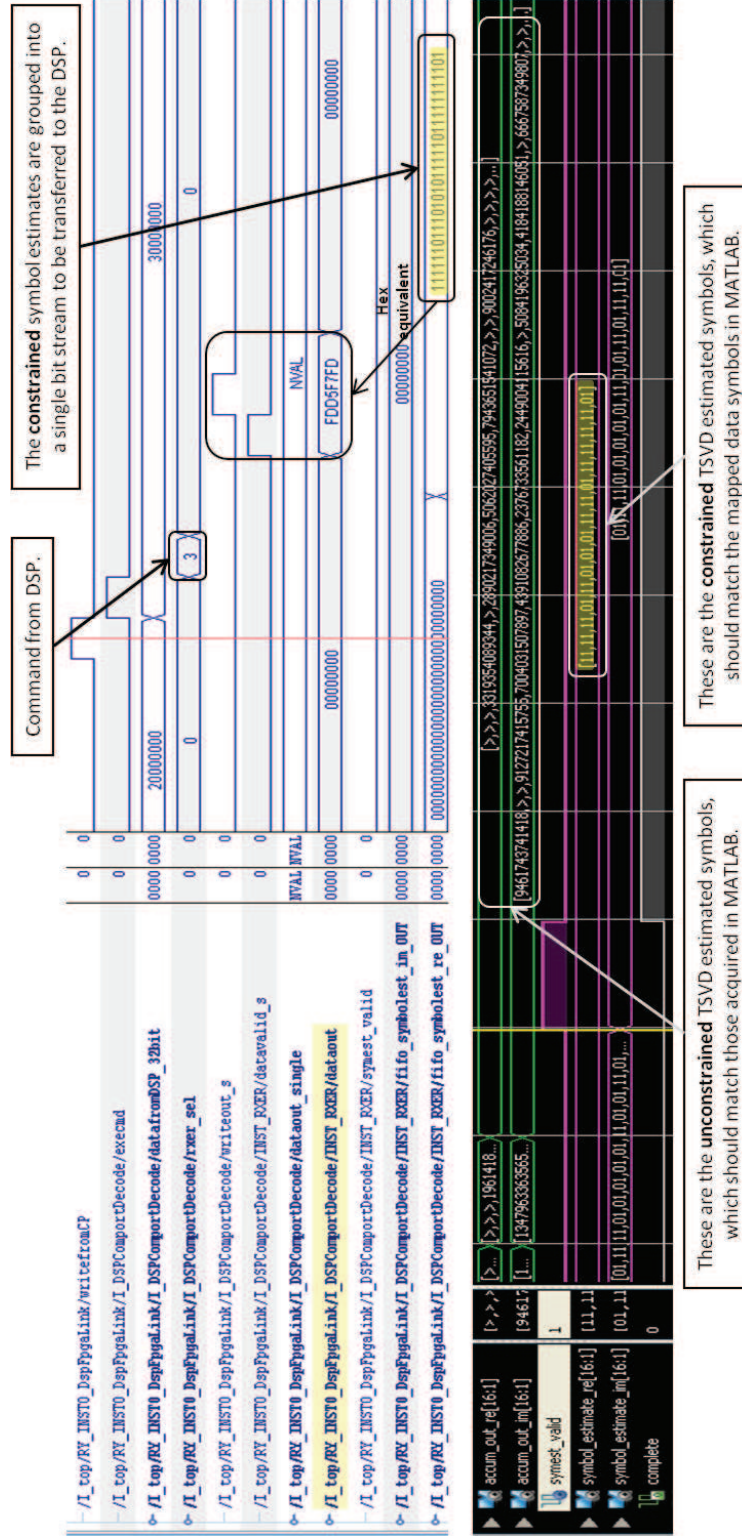


Figure B.13: Verification of the values and data transfer of the unconstrained \hat{s}_{TSVD} and constrained \hat{s}_{TSVD} symbol estimates during functional simulation (bottom) and real-time operation (top).

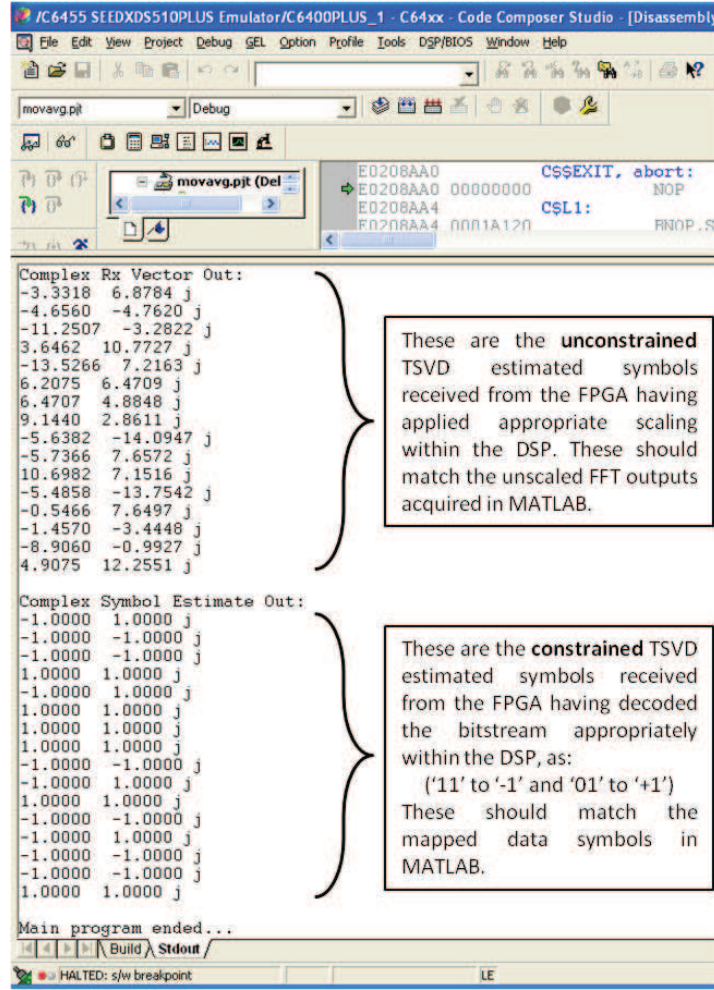


Figure B.14: Results captured from the FPGA data processing operations, subject to additional decoding operations executed within the DSP.

estimates. Subsequently, these constrained estimates are grouped into a single bit stream to simplify storage and transfer, as per Fig. B.13. Finally, Fig. B.14 shows the outputs generated by the DSP having applied appropriate scaling and decoding operations to the data values received from the FPGA. It is evident that these outputs match the values illustrated in the second and third columns of Fig. B.11.

Appendix C

Source Code

This appendix details the structure of the code used to implement the SEFDM receiver hardware designs, as well as the joint FPGA-DSP arrangement, described in Chapters 6 and 7, respectively. The files listed in the following sections may be found on the companion CD-ROM.

C.1 VHDL Code

This section lists the VHDL code for the FPGA based partition of the SEFDM receiver testbed. A hierarchical design approach was adopted, as illustrated in Table C.1. The file *FrameworkFpga.vhd* lies at the root of this hierarchy with subsequent levels responsible for instantiating lower level entities of the overall design. Each entity constitutes a component of the system and defines the ports of the lower level functions. Certain components are instantiated multiple times, for example, the number of instances of the complex multiplier is equal to the number of sub-carriers employed. Furthermore, the module *fifo_sym_est.xco* is mapped multiple times to offer multiple buffers for storing the real and imaginary parts of the received statistics vector and the constrained symbol estimates, separately. A special type of file, referred to as a testbench and denoted by *framework_tb.vhd* in this design (not shown in Table C.1), is used for the purpose of functional simulation.

Table C.1: VHDL Code Hierarchy and File Description.

Filename	Level	Description
FrameworkFpga.vhd	0	Clock and reset managers
top.vhd	1	Design component instantiation
FrameworkFpga.ucf	1	Pin assignments and timing constraints
general_pkg.vhd	1	System parameters and signal types
DspFpgaLink.vhd	2	Interface for communications port
DSPComportDecode.vhd	3	Commands for DSP read/write operations
sefdm_rxer_top.vhd	4	SEFDM receiver component instances
sefdm_fft_demod.xco	5	Fast Fourier Transform module
sefdm_complex_mult.xco	5	Complex multiplier module
sefdm_accum.xco	5	Accumulator module
tsvd_corr_re.xco	5	Storage for TSVD matrix real elements
tsvd_corr_im.xco	5	Storage for TSVD matrix imag elements
zf_corr_re.xco	5	Storage for ZF matrix real elements
zf_corr_im.xco	5	Storage for ZF matrix imag elements
sefdm_slicer.xco	5	Slicing unit
fifo_sym_est.xco	5	Buffer for storing results

C.2 C Code

This section lists the C code used to program the TMS320C6455 DSP chip. All implementation files with the extension **.c* have an associated header file which comprises the function prototypes. These header files allow each implementation file to access the corresponding functions. Table C.2 depicts the code structure and provides a brief description of each file. The file *main.c* is the master file responsible for starting the programme's execution. Finally, two additional files, namely *fsd_treeinit.c* and *fun_fsd.c*, are responsible for running the FSD algorithm.

Table C.2: C Code Hierarchy and File Description.

Filename	Level	Description
main.c	1	Top-level programme functionality
DspGpio.h	1	Memory map of DSP I/O
rtwtypes.h	1	Type definitions (auto-generated)
cmdnctrl.c	2	Control functions
init.c	2	DSP initialisation
outformat.c	2	Data output formatting
cpreadwrite.c	3	Communications port read/write functions
leds.c	3	Functions for toggling LEDs
fsd_treeinit.c	2	FSD algorithm initialisation
fun_fsd.c	2	FSD main programme execution

List of References

- [1] BBC News Technology, “Olympics put mobile communications technology to the test,” (2012, Jul. 27). [Online]. Available: <http://www.bbc.co.uk/news/technology-19004151>
- [2] M. J. Thomas, “Everything Everywhere 4G Trial,” (2012, Jul. 23). [Online]. Available: <http://www.4g.co.uk/PR/July2012/everything-everywhere-4g-trial.html>
- [3] E. Hopner, “An Experimental Modulation-Demodulation Scheme for High-Speed Data Transmission,” *IBM J.*, 1959.
- [4] J. Bingham, “Multicarrier Modulation for Data Transmission: An idea whose time has come,” *IEEE Commun. Mag.*, vol. 28, no. 5, pp. 5–14, 1990.
- [5] R. R. Mosier and R. G. Clabaugh, “Kineplex, a bandwidth efficient binary transmission system,” *AIEE Trans. Part I, Commun. Electron.*, vol. 76, pp. 723–728, 1958.
- [6] R. Chang, “Synthesis of band-limited orthogonal signals for multichannel data transmission,” *Bell Syst. Tech. J.*, vol. 45, no. 10, pp. 1775–1796, 1966.
- [7] R. W. Chang, “US Patent US3488445A - Orthogonal Frequency Division Multiplex Data Transmission System,” 1966.

- [8] R. Chang and R. Gibby, "A Theoretical Study of Performance of an Orthogonal Multiplexing Data Transmission Scheme," *IEEE Trans. Commun.*, vol. 16, no. 4, pp. 529–540, Aug. 1968.
- [9] J. Salz and S. B. Weinstein, "Fourier transform communication system," *Proc. First ACM Symp. Problems Optimization Data Commun. Syst.*, pp. 99–128, 1969.
- [10] S. Weinstein and P. Ebert, "Data Transmission by Frequency-Division Multiplexing Using the Discrete Fourier Transform," *IEEE Trans. Commun. Technol.*, vol. 19, no. 5, pp. 628–634, 1971.
- [11] J. Cioffi, "A Multicarrier Primer," *ANSI T1E1*, pp. 91–157, 1991.
- [12] J. S. Chow, J. C. Tu, and J. M. Cioffi, "A Discrete Multitone Transceiver System for HDSL Applications," *IEEE J. Sel. Areas Commun.*, vol. 9, no. 6, pp. 895–908, 1991.
- [13] E. Perahia, "IEEE 802.11n development: History, process, and technology," *IEEE Commun. Mag.*, vol. 46, no. 7, pp. 48–55, 2008.
- [14] ETSI TS 101 475 (V1.3.1), "Broadband Radio Access Networks: HIPERLAN Type 2, Physical (PHY) Layer," Tech. Rep., 2001.
- [15] European Telecommunication Standard Institute, "Digital Audio Broadcasting (DAB); DAB to Mobile, Portable and Fixed Receivers," ETS 300 401, Tech. Rep., 1995.
- [16] H. Sari, G. Karam, and I. Jeanclaude, "Transmission Techniques for Digital Terrestrial TV Broadcasting," *IEEE Commun. Mag.*, vol. 33, no. 2, pp. 100–109, 1995.
- [17] European Telecommunication Standard Institute, "Digital Video Broadcasting (DVB); Framing Structure, Channel Coding and Modulation for Digital Terrestrial Television (DVB-T)," ETS 300 744, Tech. Rep., 1997.

- [18] U. H. Reimers, "DVB - the family of international standards for digital video broadcasting," *Proc. of the IEEE*, vol. 94, no. 1, pp. 173–182, 2006.
- [19] R. Westwick, "Mobile WiMAX versus LTE: A comparison of next generation mobile broadband technologies," *J. Telecommun. Manage.*, vol. 1, no. 1, pp. 79–85, 2008.
- [20] WiMAX Forum Member Organizations, "Mobile WiMAX Part I : A Technical Overview and Performance Evaluation," WiMAX Forum, Tech. Rep., 2006.
- [21] E. Dahlman, H. Ekstrom, A. Furuskar, Y. Jading, J. Karlsson, M. Lundevall, and S. Parkvall, "The 3G long-term evolution - radio interface concepts and performance evaluation," in *IEEE 63rd Veh. Technol. Conf.*, Melbourne, VTC, 2006.
- [22] C. Eklund, R. Marks, K. Stanwood, and S. Wang, "IEEE Standard 802.16: A Technical Overview of the WirelessMAN Air Interface for Broadband Wireless Access," *IEEE Commun. Mag.*, pp. 98–107, 2002.
- [23] M. Cudak, "IEEE 802.16m System Requirements," Tech. Rep., 2008.
- [24] 3GPP TS 36.211 V8.7.0 (2009-05), "3rd Generation Partnership Project-Technical Specification Group Radio Access Network-Evolved Universal Terrestrial Radio Access (E-UTRA): Physical Channels and Modulation (Release 8)," Tech. Rep., 2009.
- [25] M. Jiang and L. Hanzo, "Multiuser MIMO-OFDM for next-generation wireless systems," *Proc. of the IEEE*, vol. 95, no. 7, pp. 1430–1469, 2007.
- [26] T. Keller and L. Hanzo, "Adaptive Multicarrier Modulation: A Convenient Framework for Time-Frequency Processing in Wireless Communications," *Proc. of the IEEE*, vol. 88, no. 5, pp. 609–610, May 2000.

- [27] W. Shieh and C. Athaudage, "Coherent optical orthogonal frequency division multiplexing," *IEEE Electron. Lett.*, vol. 42, no. 10, pp. 587–589, 2006.
- [28] J. Armstrong, "OFDM for optical communications," *J. Lightw. Technol.*, vol. 27, no. 3, pp. 189–204, 2009.
- [29] S. Ramseler, "MV and LV powerline communications: new proposed IEC standards," *IEEE Transmission and Distribution Conf.*, vol. 1, pp. 235–239, 1999.
- [30] HomePlug, "AV white paper," (2012, Oct. 12). [Online]. Available: <https://www.homeplug.org/home/>
- [31] A. Chorti and I. Kanaras, "Masked M-QAM OFDM : A Simple Approach for Enhancing the Security of OFDM Systems," in *IEEE 20th Int. Symp. Personal, Indoor and Mobile Radio Commun.*, Tokyo, PIMRC, 2009, pp. 1682–1686.
- [32] A. Chorti, "Masked-OFDM : A Physical Layer Encryption for Future OFDM Applications," in *IEEE GLOBECOM Workshops*, Miami, GC Wkshps, 2010, pp. 1254–1258.
- [33] R. Srinivasan, "IEEE 802.16m Evaluation Methodology Document (EMD)," IEEE 802.16 Broadband Wireless Access Working Group, Tech. Rep., 2008.
- [34] P. Achaichia, M. Le Bot, and P. Siohan, "OFDM/OQAM: A Solution to Efficiently Increase the Capacity of Future PLC Networks," *IEEE Trans. Power Delivery*, vol. 26, no. 4, pp. 2443–2455, 2011.
- [35] J. Burns, "Measuring spectrum efficiency - The art of spectrum utilisation metrics," Aegis Systems Ltd, Tech. Rep., 2002.
- [36] J. Zyren, "Overview of the 3GPP Long Term Evolution Physical Layer," Freescale Semiconductor Inc., Tech. Rep., 2007.
- [37] J. Mitola, "The Software Radio Architecture," *IEEE Commun. Mag.*, vol. 33, no. 5, pp. 26–38, 1995.

- [38] M. Nekovee, “Cognitive Radio Access to TV White Spaces: Spectrum Opportunities, Commercial Applications and Remaining Technology Challenges,” in *IEEE Int. Symp. New Frontiers Dynamic Spectrum Access Networks*, Singapore, DySPAN, Apr. 2010.
- [39] B. R. Saltzberg, “Performance of an Efficient Parallel Data Transmission System,” *IEEE Trans. Commun. Technol.*, vol. 15, no. 6, pp. 805–811, 1967.
- [40] J. Mazo, “Faster-than-Nyquist signaling,” *Bell Syst. Tech. J.*, vol. 54, no. 8, pp. 1451–1462, 1975.
- [41] F. Rusek and J. B. Anderson, “The Two Dimensional Mazo limit,” in *Proc. Int. Symp. Inform. Theory*, Adelaide, ISIT, 2005, pp. 970–974.
- [42] A. Liveris and C. Georgiades, “Exploiting Faster-Than-Nyquist Signaling,” *IEEE Trans. Commun.*, vol. 51, no. 9, pp. 1502–1511, Sep. 2003.
- [43] B. Hirosaki, “An Orthogonally Multiplexed QAM System using the Discrete Fourier Transform,” *IEEE Trans. Commun.*, vol. 29, no. 7, pp. 982–989, 1981.
- [44] W. Kozek and A. F. Molisch, “Nonorthogonal Pulseshapes for Multicarrier Communications in Doubly Dispersive Channels,” *IEEE J. Sel. Areas Commun.*, vol. 16, no. 8, pp. 1579–1589, 1998.
- [45] J.-P. Javaudin and D. Lacroix, “Technical description of the OFDM/IOTA modulation,” France Telecom R&D, Tech. Rep. R1-03-168, Tech. Rep., 2003.
- [46] M. Rodrigues and I. Darwazeh, “Fast OFDM: A Proposal for Doubling the Data Rate of OFDM Schemes,” in *IEEE Int. Conf. Telecommun.*, Beijing, ICT, 2002, pp. 484–487.
- [47] F. Xiong, “M-ary Amplitude Shift Keying OFDM System,” *IEEE Trans. Commun.*, vol. 51, no. 10, pp. 1638–1642, 2003.

- [48] M. Hamamura and S. Tachikawa, "Bandwidth efficiency improvement for multi-carrier systems," in *IEEE 15th Int. Symp. Personal, Indoor and Mobile Radio Commun.*, Barcelona, PIMRC, 2004, pp. 48–52.
- [49] W. Jian, Y. Xun, Z. Xi-lin, and D. Li, "The Prefix Design and Performance Analysis of DFT-based Overlapped Frequency Division Multiplexing (OvFDM-DFT) System," in *3rd Int. Workshop on Signal Design and Its Applicat. in Commun.*, Chengdu, IWSDA, Sep. 2007, pp. 361–364.
- [50] J. B. Anderson and F. Rusek, "Improving OFDM : Multistream Faster-than-Nyquist Signaling," in *4th Int. Symp. Turbo Codes and Related Topics*, Munich, ISTC, 2006.
- [51] T. Matsumoto and T. Asai, "Frequency-OverLapped Multicarrier Signaling with Inter-Carrier Interference Cancellation (FOLMSIC)," *IEICE .SAT*, vol. 97, no. 486, pp. 69–76, 1998.
- [52] M. Rodrigues and I. Darwazeh, "A spectrally efficient frequency division multiplexing based communications system," in *Proc. 8th Int. OFDM Workshop*, Hamburg, 2003, pp. 48 – 49.
- [53] J. Zhao and A. D. Ellis, "A Novel Optical Fast OFDM with Reduced Channel Spacing Equal to Half of the Symbol Rate Per Carrier," in *Optical Fiber Commun. collocated Nat. Fiber Optic Engineers Conf.*, Cork, OFC/NFOEC, 2010.
- [54] S. Yamamoto, K. Yonenaga, A. Sahara, F. Inuzuka, and A. Takada, "Achievement of Subchannel Frequency Spacing Less Than Symbol Rate and Improvement of Dispersion Tolerance in Optical OFDM Transmission," *J. Lightw. Technol.*, vol. 28, no. 1, pp. 157–163, Jan. 2010.
- [55] G. Colavolpe, T. Foggi, A. Modenini, and A. Piemontese, "Faster-than-Nyquist and beyond: how to improve spectral efficiency by accepting interference," in

- 37th European Conf. and Exhibition on Optical Commun.*, Geneva, ECOC, Dec. 2011.
- [56] U. Erez, G. Wornell, and M. Trott, "Faster-than-Nyquist Coding: The Merits of a Regime Change," in *Proc. 42nd Annu. Allerton Conf. Commun.*, 2004.
- [57] A. Chorti and H. V. Poor, "Faster than Nyquist Interference Assisted Secret Communication for OFDM Systems," in *Forty Fifth Asilomar Conf. Signals, Syst. Comput.*, Pacific Grove, ASILOMAR, Nov. 2011, pp. 183–187.
- [58] G. Fettweis, M. Krondorf, and S. Bittner, "GFDM - Generalized Frequency Division Multiplexing," in *IEEE 69th Veh. Technol. Conf.*, Barcelona, VTC, 2009.
- [59] M. R. Perrett, "Wireless multi-carrier communication system design and implementation using a custom hardware and software FPGA platform," Eng.D. thesis, University College London (UCL), 2012.
- [60] S. Sirowy and A. Forin, "Where's the Beef? Why FPGAs Are So Fast," Microsoft Research, Tech. Rep. TR-2008-130, Tech. Rep., 2008.
- [61] A. Burg, "VLSI Circuits for MIMO Communication Systems," Ph.D. dissertation, Swiss Federal Institute of Technology (ETH) Zurich, 2006.
- [62] Xilinx, "FPGA Families," (2013, Mar. 20). [Online]. Available: <http://www.xilinx.com/products/silicon-devices/fpga/index.htm>
- [63] Altera, "Altera FPGAs," (2013, Mar. 20). [Online]. Available: <http://www.altera.co.uk/devices/fpga/fpga-index.html>
- [64] ITU Radio Communication Study Groups, "Requirements related to technical performance for IMT-Advanced Radio Interface(s)," Working Party 5D, Tech. Rep., 2008.
- [65] J. P. Conti, "LTE vs WiMAX: the battle continues," (2010, Sep. 20). [Online]. Available: <http://eandt.theiet.org/magazine/2010/14/lte-vs-wimax.cfm>

- [66] R. Grammenos and I. Darwazeh, "SC-FDMA and OFDMA : The two competing technologies for LTE," in *4th Int. Symp. Broadband Commun.*, Melaka, ISBC, 2010.
- [67] A. J. Paulraj, D. A. Gore, R. U. Nabar, and H. Bölcskei, "An overview of MIMO communications - a key to gigabit wireless," *Proc. of the IEEE*, vol. 92, no. 2, pp. 198–218, 2004.
- [68] H. Sampath, S. Talwar, J. Tellado, V. Erceg, and A. Paulraj, "A Fourth-Generation MIMO-OFDM Broadband Wireless System: Design, Performance, and Field Trial Results," *IEEE Commun. Mag.*, vol. 40, no. 9, pp. 143–149, 2002.
- [69] ATDI White Papers & Case Studies, "Mobile WiMax: From OFDM-256 to S-OFDMA," ATDI Solutions, Tech. Rep., 2007.
- [70] H. Zarrinkoub, "Modeling a 4G LTE System in MATLAB," (2012, Jun. 19). [Online]. Available: http://www.mathworks.co.uk/company/events/webinars/wbnr66171.html?s_v1=42431177_1-D5YAEA
- [71] A. Roessler, "Carrier aggregation in LTE-Advanced: From simulation to practical implementation, testing and verification," (2012, Nov. 10). [Online]. Available: http://www.rslteforum.com/Carrier_Aggregation.php
- [72] J. Cosmas, "Physical Layer DVB Transmission Optimisation (PLUTO)," 2006.
- [73] A. Dammann and S. Kaiser, "Performance of Low Complex Antenna Diversity Techniques for Mobile OFDM Systems," in *3rd Int. Workshop on Multi-carrier Spread Spectrum and Related Topics*, Oberpfafenhofen, MC-SS, 2001.
- [74] A. Wittneben, "A New Bandwidth Efficient Transmit Antenna Modulation Diversity Scheme for Linear Digital Modulation," in *IEEE Int. Conf. Commun.*, Geneva, ICC, 1993, pp. 1630–1634.

- [75] P. Larsson, “Patent 09/667504 - Cyclic delay diversity for mitigating intersymbol interference in OFDM systems,” 2005.
- [76] A. Dammann, S. Plass, and S. Sand, “Cyclic Delay Diversity-A Simple, Flexible and Effective Multi-Antenna Technology for OFDM,” in *IEEE 10th Int. Symp. Spread Spectrum Tech. Applicat.*, no. 3, Bologna, ISSSTA, 2008, pp. 550–554.
- [77] S. Kaiser, “Spatial transmit diversity techniques for broadband OFDM systems,” in *IEEE Global Telecommun. Conf.*, vol. 3, San Francisco, GLOBECOM, 2000, pp. 1824–1828.
- [78] M. Iwamura, K. Etemad, M. Fong, R. Nory, and R. Love, “Carrier aggregation framework in 3GPP LTE-advanced [WiMAX/LTE Update],” *IEEE Commun. Mag.*, vol. 48, no. 8, pp. 60–67, 2010.
- [79] R. Pabst, B. Walke, D. Schultz, P. Herhold, H. Yanikomeroglu, S. Mukherjee, H. Viswanathan, M. Lott, W. Zirwas, M. Dohler, H. Aghvami, D. Falconer, and G. Fettweis, “Relay-Based Deployment Concepts for Wireless and Mobile Broad-band Radio,” *IEEE Commun. Mag.*, vol. 42, no. 9, pp. 80–89, 2004.
- [80] A. Goldsmith, *Wireless Communications*. Cambridge University Press, 2005.
- [81] L. Hanzo and T. Keller, *OFDM and MC-CDMA: A Primer*. Wiley-IEEE Press, 2006.
- [82] H. Myung, J. Lim, and D. Goodman, “Peak-To-Average Power Ratio of Single Carrier FDMA Signals with Pulse Shaping,” in *IEEE 17th Int. Symp. Personal, Indoor and Mobile Radio Commun.*, Helsinki, PIMRC, Sep. 2006.
- [83] G. Berardinelli, “OFDMA vs. SC-FDMA: Performance Comparison in Local Area IMT-A Scenarios,” *IEEE Wireless Commun.*, vol. 15, no. 5, pp. 64–72, 2008.
- [84] L. A. M. R. de Termino, G. Berardinelli, S. Frattasi, K. Pajukoski, and P. Mogensen, “Single-User MIMO for LTE-A Uplink: Performance Evaluation of

- OFDMA vs. SC-FDMA,” in *IEEE Radio and Wireless Symp.*, San Diego, RWS, 2009, pp. 305–308.
- [85] F. Al-kamali, M. Dessouky, B. Sallam, F. Shawki, and F. A. El-Samie, “Transceiver scheme for single-carrier frequency division multiple access implementing the wavelet transform and peak-to-average-power ratio reduction methods,” *IET Commun.*, vol. 4, no. 1, p. 69, 2010.
- [86] R. Van Nee and R. Prasad, *OFDM for Wireless Multimedia Communications*. Artech House, Inc. Norwood, MA, USA, 2000.
- [87] J. Proakis, *Digital Communications*, 4th ed. New York: McGraw-Hill, 2000.
- [88] I. Poole, “OFDM Basics Tutorial,” (2010, Apr. 12). [Online]. Available: <http://www.radio-electronics.com/info/rf-technology-design/ofdm/ofdm-basics-tutorial.php>
- [89] J.-P. Linnartz, “Multi-Carrier Transmission over Mobile Radio Channels,” (2010, Aug. 10). [Online]. Available: <http://4g-portal.com/multi-carriertransmission-over-mobile-radio-channels>
- [90] W. Zou, “COFDM: An Overview,” *IEEE Trans. Broadcast.*, vol. 41, no. 1, pp. 1–8, Mar. 1995.
- [91] S. W. Smith, *The Scientist and Engineer’s Guide to Digital Signal Processing*. California Technical Publishing, 1998.
- [92] D. M. Dobkin, “Orthogonal Frequency-Division Multiplexing,” in *RF Engineering for Wireless Networks: Hardware, Antennas, and Propagation*. Newnes, 2004, ch. 3, pp. 30–40.
- [93] B. L. Litwin and M. Pugel, “The principles of OFDM,” *RF signal processing*, 2001.

- [94] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, “NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey,” *Computer Networks*, vol. 50, no. 13, pp. 2127–2159, Sep. 2006.
- [95] S. Haykin, “Cognitive Radio: Brain-Empowered Wireless Communications,” *IEEE J. Sel. Areas Commun.*, vol. 23, no. 2, pp. 201–220, Feb. 2005.
- [96] R. Grammenos and Y. Yang, “Cyclostationary signatures for cognitive radio applications and novel multiple access systems,” in *London Commun. Symp.*, London, LCS, 2009.
- [97] G. Yuan, R. C. Grammenos, Y. Yang, and W. Wang, “Selective Spectrum Sensing and Access based on Traffic Prediction,” in *IEEE 20th Int. Symp. Personal, Indoor and Mobile Radio Commun.*, Tokyo, PIMRC, Sep. 2009, pp. 1078–1082.
- [98] G. Yuan, R. Grammenos, Y. Yang, and W. Wang, “Performance Analysis of Selective Opportunistic Spectrum Access With Traffic Prediction,” *IEEE Trans. Veh. Technol.*, vol. 59, no. 4, pp. 1949–1959, 2010.
- [99] P. Sutton, K. Nolan, and L. Doyle, “Cyclostationary Signatures in Practical Cognitive Radio Applications,” *IEEE J. Sel. Areas Commun.*, vol. 26, no. 1, pp. 13–24, Jan. 2008.
- [100] R. J. Lackey and D. W. Upmal, “Speakeasy: The Military Software Radio,” *IEEE Commun. Mag.*, vol. 33, no. 5, pp. 56–61, 1995.
- [101] J. Mitola, “Software Radio Architecture: A Mathematical Perspective,” *IEEE J. Sel. Areas Commun.*, vol. 17, no. 4, pp. 514–538, Apr. 1999.
- [102] J. Mitola and G. Maguire, “Cognitive Radio: Making Software Radios More Personal,” *IEEE Personal Commun.*, vol. 6, no. 4, pp. 13–18, 1999.

- [103] D. Panaitopol, R. Datta, and G. Fettweis, "Cyclostationary Detection of Cognitive Radio Systems using GFDM Modulation," in *IEEE Wireless Commun. and Networking Conf.*, Paris, WCNC, Apr. 2012, pp. 930–934.
- [104] K. Maeda, A. Benjebbour, T. Asai, T. Furuno, and T. Ohya, "Cyclostationarity-Inducing Transmission Methods for Recognition among OFDM-Based Systems," *EURASIP J. Wireless Commun. Networking*, pp. 1–14, 2008.
- [105] W. A. Gardner, "Signal interception: a unifying theoretical framework for feature detection," *IEEE Trans. Commun.*, vol. 36, no. 8, pp. 897–906, 1988.
- [106] —, *Cyclostationarity in Communications and Signal Processing*. New York: I.E.E.E. Press, 1994.
- [107] H. Guo, H. Hu, and Y. Yang, "Cyclostationary Signatures in OFDM-Based Cognitive Radios with Cyclic Delay Diversity," in *IEEE Int. Conf. Commun.*, Dresden, ICC, Jun. 2009, pp. 1–6.
- [108] B. Ramkumar, "Automatic Modulation Classification for Cognitive Radios Using Cyclic Feature Detection," *IEEE Circuits Syst. Mag.*, vol. 9, no. 2, pp. 27–45, 2009.
- [109] T. A. Weiss and F. K. Jondral, "Spectrum Pooling: An Innovative Strategy for the Enhancement of Spectrum Efficiency," *IEEE Commun. Mag.*, vol. 42, no. 3, pp. 8–14, 2004.
- [110] M. Haddad, A. Hayar, and M. Debbah, "Spectral efficiency of spectrum-pooling systems," *IET Commun.*, vol. 2, no. 6, pp. 733–741, 2008.
- [111] N. Michailow, R. Datta, and S. Krone, "Generalized Frequency Division Multiplexing: A Flexible Multi-Carrier Modulation Scheme for 5th Generation Cellular Networks," in *Proc. German Microw. Conf.*, Ilmenau, GeMiC, 2012.

- [112] R. Rajbanshi, A. M. Wyglinski, and G. J. Minden, "An Efficient Implementation of NC-OFDM Transceivers for Cognitive Radios," in *1st Int. Conf. Cognitive Radio Oriented Wireless Netw. Commun.*, 2006, pp. 1–5.
- [113] I. Budiarto, H. Nikookar, and L. P. Ligthart, "Cognitive Radio Modulation Techniques," *IEEE Signal Processing Mag.*, vol. 5, no. 6, pp. 24–34, 2008.
- [114] N. Michailow and M. Lentmaier, "Integration of a GFDM Secondary System in an OFDM Primary System," in *Future Network and Mobile Summit*, Warsaw, 2011.
- [115] B. Farhang-Boroujeny and R. Kemper, "Multicarrier Communication Techniques for Spectrum Sensing and Communication in Cognitive Radios," *IEEE Commun. Mag.*, no. 4, pp. 80–85, 2008.
- [116] P. Pawelczak, K. Nolan, L. Doyle, S. W. Oh, and D. Cabric, "Cognitive Radio : Ten Years of Experimentation and Development," *IEEE Commun. Mag.*, vol. 49, no. 03, pp. 90–100, 2011.
- [117] GNU Radio, "Introduction to GNU Radio," (2011, Mar. 18). [Online]. Available: <http://gnuradio.org/redmine/projects/gnuradio/wiki>
- [118] CREW Project, "IRIS Documentation," (2011, Mar. 18). [Online]. Available: <http://www.crew-project.eu/portal/IRISdoc>
- [119] WARP Platform, "Project Overview," (2011, Mar. 18). [Online]. Available: http://warp.rice.edu/project_overview.php
- [120] Ettus Research, "USRP Platform," (2011, Mar. 18). [Online]. Available: <http://www.ettus.com/site/about>
- [121] P. D. Sutton, K. E. Nolan, and L. E. Doyle, "Cyclostationary Signatures for Rendezvous in OFDM-Based Dynamic Spectrum Access Networks," in *IEEE*

- Int. Symp. New Frontiers Dynamic Spectrum Access Networks*, Dublin, DySPAN, Apr. 2007.
- [122] J. Lotze, S. A. Fahmy, J. Noguera, and L. E. Doyle, “An FPGA-based Autonomous Adaptive Radio,” in *Annu. Conf. ACM Special Interest Group Data Commun.*, Barcelona, ACM SIGCOMM, 2009.
- [123] P. Sutton, B. Ozgul, I. Macaluso, and L. Doyle, “OFDM Pulse-Shaped Waveforms for Dynamic Spectrum Access Networks,” in *IEEE Int. Symp. New Frontiers Dynamic Spectrum Access Networks*, Singapore, DySPAN, Apr. 2010.
- [124] S. W. Oh, W. Zhang, T. P. C. Le, Y. Zeng, A. A. Phyu, and S. N. A. A, “TV White-Space Device Prototype Using Covariance-Based Signal Detection,” in *IEEE Int. Symp. New Frontiers Dynamic Spectrum Access Networks*, Chicago, DySPAN, 2008.
- [125] S. B. Raghunathan, M. V. D. Oever, R. Doost-mohammady, P. Paweczak, and I. Budiarjo, “Dynamic Spectrum Access AAF Platform,” in *IEEE Int. Symp. New Frontiers Dynamic Spectrum Access Networks*, Chicago, DySPAN, 2008.
- [126] L. Yang, L. Cao, and H. Zheng, “Proactive Channel Access in Dynamic Spectrum Networks,” *Physical Communication*, vol. 1, no. 2, pp. 103–111, Jun. 2008.
- [127] T. Clancy and B. Walker, “Predictive Dynamic Spectrum Access,” in *Proc. SDR Forum Tech. Conf.*, Orlando, Florida, 2006.
- [128] R. Clegg, S. Isam, I. Kanaras, and I. Darwazeh, “A practical system for improved efficiency in frequency division multiplexed wireless networks,” *IET Commun.*, vol. 6, no. 4, p. 449, 2012.
- [129] I. Kanaras, “Spectrally Efficient Multicarrier Communication Systems : Signal Detection , Mathematical Modelling and Optimisation,” Ph.D. dissertation, University College London (UCL), 2010.

- [130] S. I. Ahmed, “Spectrally Efficient FDM Communication Signals and Transceivers : Design, Mathematical Modeling and System Optimization,” Ph.D. dissertation, University College London (UCL), 2011.
- [131] C. E. Shannon, “A Mathematical Theory of Communication,” *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, 1948.
- [132] B. Sklar, “Designing Digital Communication Systems,” Tech. Rep.
- [133] D. Gabor, “Theory of Communication,” *J. of the IEE - Part III: Radio and Commun. Eng.*, vol. 93, no. 26, pp. 429–441, 1946.
- [134] Y. G. Yoo and J. H. Cho, “Asymptotic Optimality of Binary Faster-than-Nyquist Signaling,” *IEEE Commun. Lett.*, vol. 14, no. 9, pp. 788–790, 2010.
- [135] T. Xu, R. C. Grammenos, F. Marvasti, and I. Darwazeh, “An Improved Fixed Sphere Decoder Employing Soft Decision for the Detection of Non-Orthogonal Signals,” *submitted to IEEE Commun. Lett.*, 2013.
- [136] A. Barbieri, D. Fertonani, and G. Colavolpe, “Improving the Spectral Efficiency of Linear Modulations Through Time-Frequency Packing,” in *IEEE Int. Symp. Inf. Theory*, no. 2, Toronto, ISIT, 2008, pp. 2742–2746.
- [137] L. Scharf, *Statistical Signal Processing*. Addison-Wesley, 1991.
- [138] D. Bailey, “The Fractional Fourier Transform and Applications,” *SIAM review*, vol. 33, no. 3, pp. 389–404, 1991.
- [139] L. Bluestein, “A Linear Filtering Approach to the Computation of Discrete Fourier Transform,” *IEEE Trans. Audio Electroacoust.*, vol. 18, no. 4, pp. 451–455, 1970.
- [140] S. Isam and I. Darwazeh, “Simple DSP-IDFT Techniques for Generating Spectrally Efficient FDM System Signals,” in *7th IEEE, IET Int. Symp. Commun. Syst. Networks and Digital Signal Process.*, Newcastle, CSNDSP, 2010.

- [141] T. Rowland, “Orthonormal Basis,” (2012, Nov. 22). [Online]. Available: <http://mathworld.wolfram.com/OrthonormalBasis.html>
- [142] G. H. Golub and C. F. v. V. Loan, *Matrix Computations*, 3rd ed. The Johns Hopkins University Press, 1996.
- [143] E. W. Weisstein, “Gram Matrix,” (2012, Nov. 22). [Online]. Available: <http://mathworld.wolfram.com/GramMatrix.html>
- [144] G. Strang, *Introduction to Linear Algebra*, 4th ed. Wellesley Cambridge Press, 2009.
- [145] C. Moler, “Least Squares,” in *Numerical Computing with MATLAB*. Society for Industrial and Applied Mathematics, 2008, ch. 5.
- [146] —, “Eigenvalues and Singular Values,” in *Numerical Computing with MATLAB*. Society for Industrial and Applied Mathematics, 2008, ch. 10.
- [147] IEEE Std 754-1985, “IEEE Standard for Binary Floating-Point Arithmetic,” American National Standard (ANSI), Tech. Rep., 1985.
- [148] J. G. Proakis and D. K. Manolakis, *Digital Signal Processing*, 4th ed. Prentice Hall, 2006.
- [149] B. Chen, H. Wang, and S. Member, “Blind Estimation of OFDM Carrier Frequency Offset via Oversampling,” *IEEE Trans. Signal Process.*, vol. 52, no. 7, pp. 2047–2057, 2004.
- [150] A. B. Kokkeler and G. J. Smit, “An Oversampled Filter Bank Multicarrier System for Cognitive Radio,” in *IEEE 19th Int. Symp. Personal, Indoor and Mobile Radio Commun.*, Cannes, PIMRC, Sep. 2008, pp. 1–5.
- [151] D. Schneider, “A Faster Fast Fourier Transform,” *IEEE Spectrum*, Mar. 2012.

- [152] R. Grammenos and I. Darwazeh, "FPGA design considerations for non-orthogonal FDM signal detection," in *London Commun. Symp.*, London, LCS, 2010.
- [153] D. Dasalukunte, F. Rusek, and V. Owall, "An Iterative Decoder for Multicarrier Faster-Than-Nyquist Signaling Systems," in *IEEE Int. Conf. Commun.*, Cape Town, ICC, 2010, pp. 1–5.
- [154] T. Matsumoto, "Zero Forcing and Decision Feedback Detectors in MIMO Communication Channels and Their Applications to Frequency-Overlapped Multi-Carrier Signaling," *IEICE Trans. Commun.*, vol. 83, no. 10, pp. 2386–2393, 2000.
- [155] H. Omori, T. Asai, and T. Matsumoto, "A Matched Filter Approximation for SC/MMSE Iterative Equalizers," in *IEEE Veh. Technol. Conf. Fall*, vol. 3, Atlantic City, New Jersey, VTC, 2001, pp. 1917–1921.
- [156] S. Isam, I. Kanaras, and I. Darwazeh, "A Truncated SVD Approach for Fixed Complexity Spectrally Efficient FDM Receivers," in *IEEE Wireless Commun. and Networking Conf.*, Cancun, WCNC, 2011.
- [157] P. C. Hansen, "The truncated SVD as a method for regularization," Numerical Analysis Project, Stanford University, Tech. Rep., 1986.
- [158] B. Hassibi and H. Vikalo, "On the expected complexity of integer least-squares problems," in *IEEE Int. Conf. Acoust., Speech, Signal Process.*, Orlando, Florida, ICASSP, 2002.
- [159] E. Viterbo and E. Biglieri, "A universal decoding algorithm for lattice codes," in *Quatorzieme Colloque Grets*, Juan-les-Pins, 1993, pp. 611–614.
- [160] K.-W. Wong, C.-Y. Tsui, R. S.-K. Cheng, and W.-H. Mow, "A VLSI architecture of a K-best lattice decoding algorithm for MIMO channels," in *IEEE Int. Symp. Circuits and Syst.*, vol. 3, Scottsdale, Arizona, ISCAS, 2002, pp. 273–276.

- [161] L. Barbero and J. Thompson, “A fixed-complexity MIMO detector based on the complex sphere decoder,” in *IEEE Workshop on Signal Process. Advances in Wireless Commun.*, Cannes, SPAWC, 2006.
- [162] B. Sklar, *Digital Communications: Fundamentals and Applications*, 2nd ed. Prentice Hall, 2002.
- [163] T. Vanevenhoven, “High-Level Implementation of Bit- and Cycle-Accurate Floating-Point DSP Algorithms with Xilinx FPGAs,” *White Paper WP409: 7 Series FPGAs*, 2011.
- [164] D. G. Manolakis and V. K. Ingle, “Finite wordlength effects,” in *Applied Digital Signal Processing*. New York: Cambridge University Press, 2011, ch. 15.
- [165] M. R. Perrett, R. C. Grammenos, and I. Darwazeh, “A Verification Methodology for the Detection of Spectrally Efficient FDM Signals Generated using Reconfigurable Hardware,” in *IEEE Int. Conf. Commun.*, Ottawa, ICC, 2012.
- [166] K. Kalliojarvi and J. Astola, “Roundoff Errors in Block-Floating-Point Systems,” *IEEE Trans. Signal Process.*, vol. 44, no. 4, pp. 783–790, 1996.
- [167] Z. Zhao and M. Leeser, “Precision Modeling and Bit-width Optimization of Floating-Point Applications,” *High Performance Embedded Computing*, 2003.
- [168] W. H. Tranter, K. S. Shanmugan, T. S. Rappaport, and K. L. Kosbar, *Principles of Communication Systems Simulation with Wireless Applications*. Prentice Hall, 2003.
- [169] Xilinx, “LogiCORE IP Complex Multiplier v3.1,” *Product Specification DS291*, 2011.
- [170] —, “LogiCORE IP Fast Fourier Transform v7.1 Bit Accurate C Model,” *User Guide UG459*, 2010.

- [171] —, “LogiCORE IP Fast Fourier Transform v7.1,” *Product Specification DS260*, 2011.
- [172] J. W. Cooley and J. W. Tukey, “An Algorithm for the Machine Calculation of Complex Fourier Series,” *Mathematics of Computation*, vol. 19, pp. 297–301, 1965.
- [173] Q. Nguyen and I. Kollar, “Limited dynamic range of spectrum analysis due to roundoff errors of the FFT,” in *IEEE Instrum. Meas. Technol. Conf.*, Budapest, IMTC, 1993, pp. 47–50.
- [174] S. Krone and G. Fettweis, “Capacity of Communications Channels with 1-bit Quantization and Oversampling at the Receiver,” in *IEEE Sarnoff Symposium*, New Jersey, SARNOFF, May 2012, pp. 1–7.
- [175] M. R. Perrett and I. Darwazeh, “Flexible Hardware Architecture of SEFDM Transmitters with Real-Time Non-Orthogonal Adjustment,” in *IEEE Int. Conf. Telecommun.*, Ayia Napa, ICT, 2011.
- [176] P. Whatmough, M. Perrett, S. Isam, and I. Darwazeh, “VLSI Architecture for a Reconfigurable Spectrally Efficient FDM Baseband Transmitter,” *IEEE Trans. Circuits Syst. I: Reg. Papers*, vol. 59, no. 5, pp. 1–12, 2012.
- [177] D. Dasalukunte, F. Rusek, and V. Öwall, “Multicarrier Faster-Than-Nyquist Transceivers Hardware Architecture and Performance Analysis,” *IEEE Trans. Circuits Syst. I, Reg. Papers*, pp. 1–12, 2010.
- [178] R. C. Grammenos, S. Isam, and I. Darwazeh, “FPGA Design of a Truncated SVD Based Receiver for the detection of SEFDM Signals,” in *IEEE 22nd Int. Symp. Personal, Indoor and Mobile Radio Commun.*, Toronto, PIMRC, 2011.
- [179] R. C. Grammenos and I. Darwazeh, “Hardware Implementation of a Practical Complexity Spectrally Efficient FDM Reconfigurable Receiver,” in *IEEE 23rd Int. Symp. Personal, Indoor and Mobile Radio Commun.*, Sydney, PIMRC, 2012.

- [180] —, “FPGA design of low complexity SEFDM detection techniques,” in *London Commun. Symp.*, London, LCS, 2011.
- [181] National Instruments, “An Introduction to High-Throughput DSP in LabVIEW FPGA,” *White Paper*, 2011.
- [182] C. Paiz, C. Pohl, R. Radkowski, J. Hagemeyer, M. Porrmann, and U. Ruckert, “FPGA-in-the-Loop-Simulations for Dynamically Reconfigurable Applications,” in *Int. Conf. Field-Programmable Technol.*, Sydney, FPT, 2009, pp. 372–375.
- [183] A. V. D. Byl, M. Inggs, and R. H. Wilkinson, “A Many Processing Element Framework for the Discrete Fourier Transform,” in *Int. Conf. Field-Programmable Technol.*, Beijing, FPT, 2010.
- [184] Xilinx, “IP LogiCORE Discrete Fourier Transform v3.1,” *Product Specification DS615*, 2011.
- [185] Y. Charles, “Nanosecond Market Data Feeds FPGA Centric vs. FPGA Accelerated Designs,” (2012, Feb. 22). [Online]. Available: <http://low-latency.com/blog/nanosecond-market-data-feeds--fpga-centric-vs-fpga-accelerated-designs>
- [186] C. Dick and F. Harris, “FPGA QAM demodulator design,” *Lecture Notes in Computer Science*, vol. 2438, pp. 112–121, 2002.
- [187] L. Barbero and J. Thompson, “Rapid prototyping of the sphere decoder for MIMO systems,” in *2nd IEE/EURASIP Conf. on DSPenabledRadio*, 2005.
- [188] T. Xu, R. C. Grammenos, and I. Darwazeh, “FPGA Implementations of Real-Time Detectors for a Spectrally Efficient FDM System,” in *IEEE Int. Conf. Telecommun. (to appear)*, Casablanca, ICT, 2013.
- [189] S. Kilts, *Advanced FPGA Design: Architecture, Implementation, and Optimization*. Wiley-Blackwell, 2007.

- [190] K. Amiri, M. Duarte, J. R. Cavallaro, C. Dick, and A. Sabharwal, “FPGA in Wireless Communications Applications,” in *FPGA Based Designs and Applications*. Springer, 2011.
- [191] R. Woods, J. McAllister, G. Lightbody, and Y. Yi, *FPGA-based Implementation of Signal Processing Systems*. Wiley-Blackwell, 2008.
- [192] M. Prasad, K. Ray, and A. Dhar, “FPGA implementation of Discrete Fractional Fourier Transform,” in *Int. Conf. Signal Process. Commun.*, Bangalore, SPCOM, 2010.
- [193] Xilinx, “LogiCORE IP Block Memory Generator v4.3,” *Product Specification DS512*, 2010.
- [194] J. Primeau, R. Demko, and R. Moquin, “A Primer on FPGA-based DSP Applications,” Acromag White Paper, Tech. Rep., 2008.
- [195] K. Amiri, C. Dick, R. Rao, and J. Cavallaro, “Flex-Sphere: An FPGA configurable sort-free sphere detector for multi-user MIMO wireless system,” in *Proc. Software Defined Radio Tech. Conf.*, Washington D.C, SDR, 2008.
- [196] Xilinx, “Product Support & Documentation,” (2012). [Online]. Available: www.xilinx.com/support/
- [197] Altera Corporation, “Implementing OFDM Modulation for Wireless Communications,” *Application Note 503*, no. 1, pp. 1–18, 2008.
- [198] I. Kanaras, A. Chorti, M. Rodrigues, and I. Darwazeh, “Spectrally Efficient FDM Signals: Bandwidth Gain at the Expense of Receiver Complexity,” in *IEEE Int. Conf. Commun.*, Dresden, ICC, Jun. 2009.
- [199] L. Barbero and J. Thompson, “Fixing the Complexity of the Sphere Decoder for MIMO Detection,” *IEEE Tran. Wireless Commun.*, vol. 7, no. 6, pp. 2131–2142, Jun. 2008.

- [200] R. C. Grammenos and I. Darwazeh, “Performance Trade-Offs and DSP Evaluation of Spectrally Efficient FDM Detection Techniques,” in *IEEE Int. Conf. Commun. (to appear)*, Budapest, ICC, 2013.
- [201] U. Fincke and M. Pohst, “Improved Methods for Calculating Vectors of Short Length in a Lattice, Including a Complexity Analysis,” *Mathematics of Computation*, vol. 44, pp. 463–471, 1985.
- [202] C. P. Schnorr and M. Euchner, “Lattice Basis Reduction : Improved Practical Algorithms and Solving Subset Sum Problems,” *Mathematical Programming*, vol. 66, no. 2, pp. 181–199, 1994.
- [203] L. M. Davis, “Scaled and Decoupled Cholesky and QR Decompositions with Application to Spherical MIMO Detection,” in *IEEE Wireless Commun. and Networking Conf.*, New Orleans, Louisiana, WCNC, 2003, pp. 326–331.
- [204] E. Viterbo and J. Boutros, “A Universal Lattice Code Decoder for Fading Channels,” *IEEE Tran. Inf. Theory*, vol. 45, no. 5, pp. 1639–1642, 1999.
- [205] B. Hassibi and H. Vikalo, “On the Sphere-Decoding Algorithm I. Expected Complexity,” *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 2806–2818, 2005.
- [206] J. Jalden and B. Ottersten, “On the Complexity of Sphere Decoding in Digital Communications,” *IEEE Trans. Signal Process.*, vol. 53, no. 4, pp. 1474–1484, Apr. 2005.
- [207] B. M. Hochwald and S. ten Brink, “Achieving Near-Capacity on a Multiple-Antenna Channel,” *IEEE Trans. Commun.*, vol. 51, no. 3, pp. 389–399, Mar. 2003.
- [208] M. Damen, H. El Gamal, and G. Caire, “On Maximum-Likelihood Detection and the Search for the Closest Lattice Point,” *IEEE Trans. Inf. Theory*, vol. 49, no. 10, pp. 2389–2402, Oct. 2003.

- [209] K. Amiri, C. Dick, R. Rao, and J. Cavallaro, "Novel sort-free detector with modified real-valued decomposition (M-RVD) ordering in MIMO systems," in *IEEE Proc. Global Telecommun.*, New Orleans, GLOBECOM, 2008, pp. 1–5.
- [210] Texas Instruments, "Communications & Telecom," (2012). [Online]. Available: <http://www.ti.com>
- [211] L. Barbero and J. Thompson, "Rapid Prototyping of a Fixed-Throughput Sphere Decoder for MIMO Systems," in *IEEE Int. Conf. Commun.*, Istanbul, ICC, 2006, pp. 3082–3087.
- [212] A. Burg, M. Borgmann, M. Wenk, M. Zellweger, W. Fichtner, and H. Bolcskei, "VLSI Implementation of MIMO Detection using the Sphere Decoding Algorithm," *IEEE J. Solid-State Circuits*, vol. 40, no. 7, pp. 1566–1577, Jul. 2005.
- [213] S. Haene, D. Perels, and A. Burg, "A Real-Time 4-Stream MIMO-OFDM Transceiver: System Design, FPGA Implementation, and Characterization," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 6, pp. 877–889, Aug. 2008.
- [214] L. Barbero and J. Thompson, "Extending a Fixed-Complexity Sphere Decoder to Obtain Likelihood Information for Turbo-MIMO Systems," *IEEE Trans. Veh. Technol.*, vol. 57, no. 5, pp. 2804–2814, Sep. 2008.
- [215] A. Chorti, I. Kanaras, M. Rodrigues, and I. Darwazeh, "Joint Channel Equalization and Detection of Spectrally Efficient FDM Signals," in *IEEE 21st Int. Symp. Personal Indoor and Mobile Radio Commun.*, Istanbul, PIMRC, 2010.
- [216] S. Isam and I. Darwazeh, "Robust Channel Estimation for Spectrally Efficient FDM System," in *IEEE Int. Conf. Telecommun.*, Jounieh, ICT, Apr. 2012.