University of A Coruña

Department of Electronics and Systems

PhD Thesis

Rapid Prototyping for Evaluating Vehicular Communications

Tiago Manuel Fernández Caramés

PhD Advisor: Miguel González López

2011

D. Miguel González López

CERTIFICA:

Que la memoria titulada "Rapid Prototyping for Evaluating Vehicular Communications", ha sido realizada por D. Tiago Manuel Fernández Caramés bajo mi dirección en el Departamento de Electrónica y Sistemas de la Universidade da Coruña y concluye la Tesis que presenta para optar al grado de Doctor.

A Coruña, 20 de julio de 2011

Fdo: Dr. Miguel González López
Director de la Tesis Doctoral
Profesor Contratado Doctor
Dpto. de Electrónica y Sistemas
Universidade da Coruña

Tesis doctoral: *Rapid Prototyping for Evaluating Vehicular Communications*
Autor: Tiago Manuel Fernández Caramés
Director: Miguel González López
Fecha:

Tribunal

Presidente:

Vocal 1:

Vocal 2:

Vocal 3:

Secretario:

Summary

This Thesis details the different elements of a rapid prototyping system able to implement and evaluate vehicular communications fast, according to the continuously evolving requirements of the industry. The system is basically composed of a testbed and a channel emulator, which allow evaluating communication transceivers in realistic vehicular scenarios.

Two different testbeds are introduced: a generic 2×2 system and a vehicular platform. The former is used to compare and study space-time block coding (STBC) transmissions at 2.4 GHz over different indoor channels. The latter makes use of software transceivers whose performance is evaluated when they work under artificial high-speed Rayleigh-fading scenarios.

To show the capabilities of both platforms, three software transceivers have been developed following the specifications for the physical layers of the standards IEEE 802.11p, IEEE 802.11a and IEEE 802.16e (Mobile WiMAX). The present work details the different elements that make up each transceiver and indicates how to connect them to the rest of the system to perform evaluation measurements.

Finally, single-antenna and multi-antenna performances are measured thanks to the design and implementation of three FPGA-based channel emulators that are able to recreate up to seven different vehicular scenarios that include urban canyons, suburban areas and highways.

Resumo

A presente Tese detalla os elementos necesarios para constituir un sistema basado en prototipado rápido capaz de levar a cabo e avaliar comunicacións vehiculares. O hardware do sistema está composto básicamente por unha plataforma de probas (*testbed*) e un emulador de canal, os cales permiten avaliar o rendemento de transceptores inarámicos recreando diferentes escenarios vehiculares.

Inicialmente, este traballo céntrase na descripción do hardware do sistema, detallando a construcción e proba dunha plataforma multi-antena e un testebed vehicular. Estos sistemas permitiron, respectivamente, estudar o comportamento de códigos STBC (space-time block codes) en interiores e medir o rendemento de tranceptores software ao traballar a distintas velocidades vehiculares en canais con desvaecemento Rayleigh.

Tres transceptores software foron creados seguindo as especificacións das capas físicas dos estándares IEEE 802.11p, IEEE 802.11a e IEEE 802.16e (Mobile WiMAX). Este traballo detalla os diferentes componentes de cada transceptor, indicando cómo conectalos ao resto do sistema para realizar a avaliación do seu rendemento. Dita avaliación realizouse coa axuda de tres emuladores de canal basados en tecnoloxía FPGA (Field Programmable Gate Array), os cales son capaces de recrear ata sete escenarios vehiculares distintos, incluindo cañóns urbanos, zonas suburbanas e autopistas.

Resumen

La presente Tesis detalla los elementos necesarios para constituir un sistema basado en prototipado rápido capaz de llevar a cabo y evaluar comunicaciones vehiculares. El hardware del sistema está compuesto por una plataforma de pruebas (*testbed*) y un emulador de canal, los cuales permiten evaluar el rendimiento de transceptores inalámbricos recreando diferentes escenarios vehiculares.

Inicialmente, este trabajo se centra en la descripción del hardware del sistema, detallando la construcción y prueba de una plataforma multi-antena y un testebed vehicular. Estos sistemas han permitido, respectivamente, estudiar el comportamiento de códigos STBC (space-time block codes) en interiores y medir el rendimiento en canal con desvanecimiento Rayleigh de tranceptores software a distintas velocidades vehiculares.

Tres transceptores software han sido creados siguiendo las especificaciones de las capas físicas de los estándares IEEE 802.11p, IEEE 802.11a e IEEE 802.16e (Mobile WiMAX). Este trabajo detalla los diferentes componentes de cada transceptor, indicando cómo conectarlos al resto del sistema para realizar la evaluación de su rendimiento. Dicha evaluación se realizó con la ayuda de tres emuladores de canal basados en FPGAs (Field Programmable Gate Array), los cuales son capaces de recrear comunicaciones multi-antena en hasta siete escenarios vehiculares distintos, incluyendo cañones urbanos, zonas suburbanas y autopistas.

Á memoria de Gala e aos meus pais,

Acknowledgements

This Thesis has been possible thanks to the support of many people and entities. Please, receive my apologies in advance if I have forgotten to mention anyone of you.

First, I must thank my Ph. D. advisor, Dr. Miguel González López for his guidance and support during the last five years. This Thesis would not exist without his recommendations and meticulous reviews. I also have to express my gratitude to Dr. Luis Castedo Ribas who gave me the oppotunity of working for the GTEC and whose experience and tips have been invaluable to my work.

My sincere thanks to professors Carlos Escudero, Adriana Dapena, Paula Castro and Daniel Iglesia who have supported my work in different research projects and teaching activities. Moreover, I must thank professors José Juan Lamas Seco and José María Domínguez, whose lectures and tips have played a key role in the election of my career path.

Of course, I cannot forget about other researchers that have helped me (in some cases unconsiously) during these last five years: Analía, Ángel, Belén, Fran, Héctor, Javi, Jose, Jose Antonio, Josmary, Manuel, Miguel Rubinos, Néstor, Óscar, Paula Fraga, Pedro, Santi, Sonia and Valentín.

Furthermore, I must express my gratitude and appreciation to Cristina Ribao, whose help and advice have been inestimable during the last years.

Finally, I would like to thank the GTAS group (specially David Ramírez, Ignacio Santamaría, Jesús Pérez, Jesús Ibañez and Javier Vía) for their collaboration and their excellent treat during my visits to Santander.

Contents

List of Figures **xxi**

List of Tables **xxvii**

1 Introduction **1**

1.1 Thesis Overview . . . 2

1.2 Related Publications and Projects . . . 3

2 Rapid prototyping for hardware testbeds **7**

2.1 Introduction . . . 7

2.2 General View of the MIMO 2 × 2 Testbed . . . 9

2.3 Baseband Modules . . . 11

2.3.1 Transmitter . . . 11

2.3.2 Receiver . . . 14

2.4 RF Modules . . . 17

2.5 MIMO 2 × 2 Platform Versus Other Platforms . . . 18

2.6 Multi-layered Software Architecture for Testbed Automation . . . 19

2.6.1 General View of the Architecture . . . 20

2.6.2 Protocol Operation . . . 22

2.6.3 Middleware Layer . . . 24

2.6.4 Use Cases . . . 25

2.7 Experiments: Space-Time Block Coding . . . 26

2.7.1 Characteristics of the Experiments . . . 28

2.7.2 Results . . . 29

2.8 Conclusions . . . 35

3 Rapid prototyping for vehicular demonstrators **37**

3.1 FlexVehd: a Flexible Vehicular Demonstrator . . . 37

3.2 FlexVehd Architecture . . . 38

3.3 Middleware Layer . . . 40

3.4 FlexVehd Hardware Description . . . 40

3.5 IEEE 802.11p PHY Layer Implementation . . . 41

3.6 Experimental Performance Results of IEEE 802.11p PHY Under High Mobility Conditions . . . 42

3.7 Conclusions 44

4 Rapid prototyping for vehicular wireless transceivers 45

4.1 Introduction 45

4.2 IEEE 802.11p and IEEE 802.11a 48

4.2.1 Background: IEEE 802.11p Evaluations 48

4.2.2 IEEE 802.11p/a Transceivers 49

4.2.2.1 SISO Transceiver 50

4.2.2.2 MIMO Transceiver 52

4.3 Mobile WiMAX for Vehicular Scenarios 55

4.3.1 Background 55

4.3.2 Mobile WiMAX Transceiver 60

4.3.2.1 Mobile WiMAX MATLAB/Simulink Transmitter . . . 61

4.3.2.2 Mobile WiMAX MATLAB/Simulink Receiver 63

4.4 Conclusions 64

5 Rapid prototyping for SISO vehicular channel emulators 65

5.1 Introduction 65

5.2 Background: Commercial and Ad-Hoc Channel Emulators 66

5.3 Implemented Vehicular Channel Models 69

5.4 Theoretical Model 73

5.5 Hardware and Software 75

5.6 Hardware Set-up to Evaluate Physical Transceivers 76

5.7 FPGA Design Overview 77

5.7.1 Acquisition of the Channel Parameters 77

5.7.2 White Gaussian Noise Generation 78

5.7.3 Doppler Filtering 79

5.7.4 Interpolation 82

5.7.5 FIR Filtering 83

5.8 Emulator Basic Operation 84

5.9 Emulator input/output interface 85

5.10 Experiments: Evaluating Vehicular SISO Transceiver Performance . . . 87

5.10.1 Evaluation of the IEEE 802.11p Physical layer over Vehicular Channels 87

5.10.2 Observed Inter-Carrier Interference (ICI) in the Implemented Vehicular Channels 89
5.10.3 Importance of Coding over Vehicular Channels 91
5.10.4 IEEE 802.11p Vs IEEE 802.11a in Vehicular Channels 91
5.10.5 IEEE 802.11p Vs IEEE 802.11a Vs IEEE 802.16e 93
5.10.5.1 Performance over AWGN and Rayleigh Fading Channels 94
5.10.5.2 Performance over Vehicular Channels 95
5.10.5.3 Discussion: IEEE 802.16e or IEEE 802.11p? 98
5.11 Conclusions 101

6 Rapid Prototyping for MIMO Vehicular Channel Emulators 103
6.1 Introduction 103
6.2 Background 106
6.2.1 Mobile and Vehicular Channel Models 106
6.2.2 MIMO Channel Emulators 107
6.3 MIMO Performance Evaluation Systems 110
6.3.1 FPGA-based MIMO Channel Emulator Built by Upgrading the SISO Vehicular Channel Emulator 110
6.3.1.1 First Optimization 111
6.3.1.2 Second Optimization 111
6.3.1.3 Third Optimization 112
6.3.1.4 MIMO Time-Multiplexed Approach 113
6.3.2 FPGA-based MIMO Channel Emulator for the New MIMO 2 × 2 Acosta's Channel Model 114
6.3.2.1 MIMO System Design 115
6.3.2.2 Coefficient Generation Process 116
6.4 Experimental Results 117
6.4.1 Co-Simulation Vs. Software Simulation Lags 117
6.4.2 Performance Measurements 118
6.4.2.1 Characteristics of the Measurements 118
6.4.2.2 Performance in Frequency-flat Rayleigh Channel 118
6.4.2.3 Channel Estimation Error in Rayleigh Channel 120

6.4.2.4 Performance in the Vehicular Channels Built by Upgrading Acosta's SISO models 121

6.4.2.5 Performance in the New MIMO 2×2 Acosta's Channel Model 129

6.5 Conclusions 130

7 Conclusions and Future Work 135

7.1 Conclusions 135

7.2 Future Work 137

7.2.1 Implementation of Other SISO and MIMO Channels 137

7.2.2 Evaluation of Transceivers in Vehicular Environments 138

7.2.3 BER/FER Input for Vehicular Simulators 138

7.2.4 Evaluation of Infotainment Services Performance 139

7.2.5 Web-Controlled Vehicular Platform 139

References 141

Appendices 147

A Data Model and Characteristics of the MIMO-STBC System 149

A.1 Data Model for MIMO STBC systems 149

A.2 Orthogonal STBCs 152

A.3 Channel Estimation in MIMO-OSTBC systems 153

A.3.1 Pilot-Aided Channel Estimation. 153

A.3.2 SOS-based Blind Channel Estimation 154

A.4 Differential STBCs 156

A.4.1 Encoding Algorithm 156

A.4.2 Decoding Algorithm 157

B List of Acronyms 159

List of Figures

1.1 Abstract view of the elements of the rapid prototyping system described in this Thesis. 3

2.1 In color: components of the rapid prototyping system that are described in Chapter 2. 9

2.2 Schematic diagram of the MIMO 2 × 2 platform. 10

2.3 A picture of the MIMO 2 × 2 platform. 10

2.4 Schematic diagram of the components of the MIMO 2 × 2 transmitter. . 12

2.5 Picture and components of the MIMO 2 × 2 transmitter. 12

2.6 Hardware trigger for the MIMO 2 × 2 testbed. 16

2.7 RX board schematic of the MIMO 2 × 2 testbed. 17

2.8 Generic architecture for a MIMO testbed. 20

2.9 Generic MIMO testbed architecture. 21

2.10 Example of a frame transmission through the MIMO 2 × 2 testbed. . . . 22

2.11 Sequence diagram of the protocol working in a multi-user environment. 26

2.12 Antenna locations for the MIMO 2 × 2 STBC experiments at the University of Cantabria. 29

2.13 Antenna locations for the MIMO 2 × 2 STBC experiments at the University of A Coruña. 30

2.14 Frame structure chosen for the MIMO 2 × 2 STBC experiments. 30

2.15 Symbol constellations at the receiver during the MIMO 2 × 2 STBC experiments. 31

2.16 BER for the LOS scenario during the MIMO 2 × 2 STBC experiments. . 32

2.17 BER for the NLOS scenario during the MIMO 2 × 2 STBC experiments. 32

2.18 BER for the Rician simulation during the MIMO 2 × 2 STBC experiments. 33

2.19 BER for the Rayleigh simulation during the MIMO 2 × 2 STBC experiments. 34

2.20 BER for the 5 Mbaud tests during the MIMO 2 × 2 STBC experiments. 35

3.1 In color: components of the rapid prototyping system that are described in Chapter 3. 38

3.2 FlexVehd platform diagram. 39

3.3 FlexVehd architecture scheme. 39

3.4 Picture of FlexVehd showing the Tx and Rx PCs. 40
3.5 BER and PER Vs SNR of FlexVehd for different vehicle speeds. BPSK, code rate $R_c = 3/4$, 64-byte packets. 42
3.6 BER and PER Vs SNR of FlexVehd for different vehicle speeds. 4-QAM, code rate $R_c = 1/2$, 64-byte packets. 43
3.7 BER and PER Vs SNR of FlexVehd for different vehicle speeds. 16-QAM, rate $R_c = 1/2$, 64-byte packets. 43

4.1 In color: components of the rapid prototyping system that are described in Chapter 4. 47
4.2 IEEE 802.11p transceiver connected to the SISO vehicular channel emulator. 50
4.3 Placement of the DC, data and pilots into the IEEE 802.11p subcarriers. 51
4.4 Time-multiplexed evaluation system for multiple-antenna IEEE 802.11p transceivers. 53
4.5 Time-multiplexed MIMO 2×2 emulator model without buffering. . . . 54
4.6 Evaluation system of a MATLAB/Simulink IEEE 802.16e transceiver using a vehicular channel emulator. 62

5.1 In color: components of the rapid prototyping system that are described in Chapter 5. 66
5.2 General view of the System Generator model optimized for the vehicular SISO channel *VTV-Expressway Oncoming*. 74
5.3 Hardware set-up for testing an IEEE 802.11p physical transceiver with IF/RF outputs on the vehicular channel emulator. 77
5.4 Resource-efficient 24-output Gaussian noise generator. 80
5.5 Optimized blocks for applying the Doppler Spectrum in the SISO vehicular channel emulator. 81
5.6 Generation and addition of the LOS and NLOS components of each path in the SISO emulator. 82
5.7 One path's linear interpolator of the SISO emulator. 82
5.8 General view of the complex FIR filter of the SISO channel emulator. . 84
5.9 BER of the IEEE 802.11p transceiver for the different SISO vehicular channels. 88

5.10 PER of the IEEE 802.11p transceiver for the different SISO vehicular channels. 89
5.11 BER/PER when transmitting with and without coding through *RTV-Expressway*. 92
5.12 BER/PER when transmitting with and without coding through *VTV-Expressway Same Direction with Wall*. 92
5.13 BER/PER comparison between IEEE 802.11p and IEEE 802.11a when transmitting over the channel emulator. 93
5.14 Performance comparison of IEEE 802.11p, IEEE 802.11a and IEEE 802.16e transceivers when transmitting over an AWGN channel. 94
5.15 Performance comparison of IEEE 802.11p, IEEE 802.11a and IEEE 802.16e transceivers when transmitting over a block-fading Rayleigh channel. 95
5.16 Channel estimation error of the IEEE 802.11p and IEEE 802.16e transceivers when transmitting over an AWGN channel. 96
5.17 Performance comparison of IEEE 802.11p, IEEE 802.11a and IEEE 802.16e when transmitting over *VTV-Urban Canyon Oncoming*. 96
5.18 Performance comparison of IEEE 802.11p, IEEE 802.11a and IEEE 802.16e when transmitting over *RTV-Urban Canyon*. 97
5.19 Performance comparison of IEEE 802.11p, IEEE 802.11a and IEEE 802.16e when transmitting over *RTV-Suburban Street*. 98
5.20 Performance comparison of IEEE 802.11p, IEEE 802.11a and IEEE 802.16e when transmitting over *VTV-Expr. Oncoming*. 98
5.21 Performance comparison of IEEE 802.11p, IEEE 802.11a and IEEE 802.16e when transmitting over *RTV-Expressway*. 99
5.22 Performance comparison of IEEE 802.11p, IEEE 802.11a and IEEE 802.16e when transmitting over *VTV-Expressway Same Direction with Wall*. 100

6.1 In color: components of the rapid prototyping system that are described in Chapter 6. 105
6.2 New Doppler filtering stage for the SIMO 1 × 2 channel emulator. . . . 112
6.3 New Gaussian noise generator block for the SIMO 1 × 2 channel emulator.113

6.4 BER performance of SISO, SIMO and MIMO IEEE 802.11p transceivers for frequency-flat block fading Rayleigh channel. 120

6.5 FER performance of SISO, SIMO and MIMO IEEE 802.11p transceivers for frequency-flat block fading Rayleigh channel. 121

6.6 Channel estimation error for frequency-flat block fading Rayleigh channel for different Doppler frequencies for SIMO and MIMO IEEE 802.11p transceivers. 122

6.7 Channel estimation error for the MIMO 4×4 IEEE 802.11p transceivers around SNR=10 dB for frequency-flat block fading Rayleigh channel. . . 123

6.8 BER performance for SISO, SIMO and MIMO *VTV-Expressway Oncoming*. 123

6.9 FER performance for SISO, SIMO and MIMO *VTV-Expressway Oncoming*. 124

6.10 BER performance for SISO, SIMO and MIMO *RTV-Urban Canyon*. . . 124

6.11 FER performance for SISO, SIMO and MIMO *RTV-Urban Canyon*. . . 125

6.12 BER performance for SISO, SIMO and MIMO *RTV-Expressway*. 125

6.13 FER performance for SISO, SIMO and MIMO *RTV-Expressway*. 126

6.14 BER performance for SISO, SIMO and MIMO *VTV-Urban Canyon Oncoming*. 126

6.15 FER performance for SISO, SIMO and MIMO *VTV-Urban Canyon Oncoming*. 127

6.16 BER performance for SISO, SIMO and MIMO *RTV-Suburban Street*. . . 128

6.17 FER performance for SISO, SIMO and MIMO *RTV-Suburban Street*. . . 129

6.18 BER performance for SISO, SIMO and MIMO *VTV-Exp. Same Direction With Wall*. 130

6.19 FER performance for SISO, SIMO and MIMO *VTV-Exp. Same Direction With Wall*. 131

6.20 BER performance comparison among the different MIMO 4×4 vehicular channels. 132

6.21 FER performance comparison among the different MIMO 4×4 vehicular channels. 133

6.22 Channel estimation error in MIMO 4×4 vehicular scenarios. 133

6.23 BER performance comparison among the different MIMO 2×2 vehicular channels. 134

6.24 FER performance comparison among the different MIMO 2×2 vehicular channels. 134

List of Tables

2.1 Rate for the different channel estimation methods used during the MIMO 2×2 STBC experiments. 31

3.1 IEEE 802.11p PHY key parameters 41

4.1 Feature comparison of the IEEE 802.11p, IEEE 802.11a and IEEE 802.16e standards. 49

5.1 Main characteristics of the SISO vehicular models. 72

5.2 General parameters and resources occupied by the SISO vehicular channel emulator. 73

5.3 Estimated savings obtained thanks to the optimization of the Gaussian generator for the SISO channel emulator. 79

5.4 Savings due to the optimization of the Doppler filter block in the SISO vehicular channel emulator. 81

5.5 Minimum E_b/N_0 to obtain a PER of 10% in the SISO vehicular channels. 89

5.6 Energy ratios for the SISO vehicular channels. 90

6.1 Resource utilization of different versions of the SIMO/MIMO vehicular channel emulator. 111

6.2 Main parameters of the new Acosta's MIMO 2×2 empirical model. . . 115

6.3 General parameters and resources occupied by the new MIMO 2×2 vehicular channel emulator 116

6.4 Co-Simulation and Simulink lags for different channels and multiple-antena systems. 119

6.5 SNR (dB) required to obtain a FER of 10% in each SIMO/MIMO vehicular channel. 128

1
Introduction

Wireless local area network (WLAN) communications have experienced a huge growth in the last decade. Since the beginning of this Thesis (2005), the IEEE group responsible for defining new WLAN standards (IEEE 802.11) has released 10 standards and amendments (1). As of writing, the same group is working on another 9 standards and amendments that are expected to be published before 2014.

This fast succession of events has lead researchers to a situation that requires to be continuously updated, both in terms of scientific knowledge and hardware. In the past, the companies and institutions with large research budgets would be the only ones able to keep up with the pace of the evolution of technology: most of the hardware and software had to be developed specifically for a transceiver, what requires an important amount of human and technical resources. Nowadays the situation has changed. The proliferation of new affordable rapid prototyping tools and techniques has allowed small research groups to compete in terms of scientific results with wealthier and larger entities. One of areas where such competition has taken place is Intelligent Transport Systems (ITS), especially in the field of Vehicular Communications, due to the demand for applications to improve safety and travel comfort.

Currently, IEEE 802.11p seems to be the best positioned standard for providing vehicular services, although other wireless communication standards have to be still assessed. To determine the suitability of each wireless standard, performance measurements have to be carried out. Such tests require hardware and software that have to be flexible enough to be able to follow different transceiver specifications, which should be evaluated in realistic situations.

A testbed is probably the most useful tool to test new data processing algorithms and techniques. Testbeds should be ideally composed of modular hardware and need two different kinds of software: applications to control the inner workings of the hardware and software dedicated to implement the transceivers to be evaluated.

Two different approaches can be followed when testing a transceiver in a realistic vehicular scenario. The most straightforward one consists in carrying the hardware platform inside a vehicle and driving through different environments, but this is a time-consuming task and the experiments can be affected by unintended side effects that may be uncontrollable. It is more convenient to use a hardware channel emulator and measure the performance inside a testing lab. Since vehicular channel models are still being investigated by researchers, the channel emulator used have to be flexible enough to be reconfigured easily in order to adapt its parameters to future specifications.

1.1 Thesis Overview

This Thesis describes a rapid prototyping system able to perform and evaluate vehicular communications fast, according to the continuously evolving requirements of the industry. The system is composed of four different elements: a hardware platform (two different platforms are described in Chapters 2 and 3), the software to control such platform (detailed in Chapter 2), software transceivers (in Chapter 4) and a channel emulator (Chapters 5 and 6).

The Thesis begins in Chapter 2 setting the basics of the hardware platform: it is described how to build a Multiple-Input Multiple-Output (MIMO) testbed with two antennas at transmission and two antennas at reception (2×2) that it is used to compare and study space-time block coding (STBC) transmissions at 2.4 GHz over different indoor channels. In Chapter 2 it is also detailed an intermediate software layer that let the wireless transceiver designer forget about the low-level details of the hardware platform and just focus on the transceiver implementation details. Next, Chapter 3 shows how to extrapolate the know-how obtained during the creation of the MIMO 2×2 testbed in order to build a vehicular testbed.

Chapter 4 describes the implementation of different wireless transceivers that show the capabilities of the evaluation systems developed. Such transceivers were implemented using MATLAB® and SIMULINK® following the specifications for the physical layers of the standards IEEE 802.11p, IEEE 802.11a and IEEE 802.16e (Mobile WiMAX). The performance of the transceivers has been analyzed when transmitting in different vehicular scenarios: in Chapter 5 single-antenna performances are compared when using an FPGA-based channel emulator whereas in Chapter 6 two different MIMO channel emulators are connected to multi-antenna IEEE 802.11p transceivers to observe how they behave in vehicular scenarios.

The different hardware and software elements that make up the platform are represented in Fig. 1.1. There it can be seen that the hardware used is composed of a transmitter and a receiver (of any of the two testbeds developed), which connect with one of the three available vehicular channel emulators. In the figure mentioned two different kinds of software applications are shown: the ones aimed at controlling the hardware and the ones that allow implementing the specifications given by a wireless standard. In order to facilitate the reading of this Thesis, at the beginning of each chapter it is highlighted in colors which elements of Fig. 1.1 the Chapter deals with.

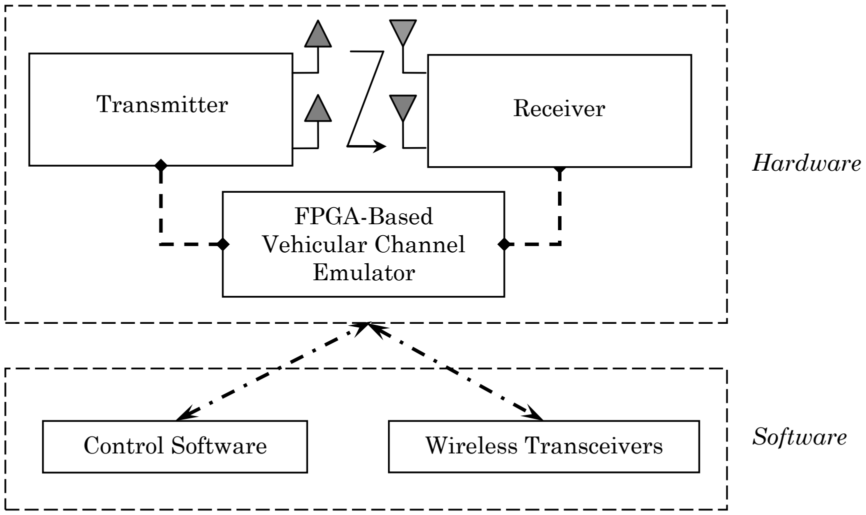

Figure 1.1: Abstract view of the elements of the rapid prototyping system described in this Thesis.

1.2 Related Publications and Projects

Most of the content of this Thesis has already been published in different journal and conference papers:

- The hardware platform presented in Chapter 2 is described in (2). The different elements that make up the testbed have been detailed extensively in (3), (4), (5) and (6). A deep and comprehensive description of the control software of the testbed is given in (7), (8) and (9). It must be also said that the novelty of the platform has been recognized in the form of an industrial patent (patent number: ES 2340560 B1).

- The description of the vehicular platform covered by Chapter 3 is given in (10).
- The content of Chapters 4 and 5 has been exposed in the book chapter in (11).
- Chapter 5 is based on two articles published in the EURASIP Journal on Wireless Communications and Networking (12) and Computer Networks (13). A short version of the former article was presented in Lille at the International Conference on Intelligent Transport System Telecommunications (ITST) (14), whereas the foundations of the vehicular channel emulator were initially settled in (15).
- Chapter 6 is a refined version of the content exposed in (16), which, at the time of writing, has been submitted to an special issue of the EURASIP Journal on Wireless Communications and Networking.

The research performed for this Thesis has contributed to the following projects:

- *MIMESIS*: funded by the Spanish Ministerio de Educación y Ciencia (reference: TEC2004-06451-C05-01).
- *Incentivo al proyecto MIMESIS*: funded by the Galician Consellería de Innovación, Industria e Comercio of the Xunta de Galicia (reference: PGIDT05PXIC 10502PN).
- *Receptores iterativos para WLAN de última generación (IEEE 802.11n)*: funded by the Galician Consellería de Innovación, Industria e Comercio of the Xunta de Galicia (reference: PGIDT06TIC10501PR).
- *MULTIMIMO*: funded by the Spanish Ministerio de Educación y Ciencia (reference: TEC2007-68020-C04-01).
- *TECOMVEH*: funded by the Spanish Ministerio de Industria, Turismo y Comercio (reference: TSI-020302-2008-4).

- *COMONSENS*: funded by the Spanish Ministerio de Ciencia e Innovación (reference: CSD2008-00010).
- *COSIMA*: funded by the Spanish Ministerio de Ciencia e Innovación (reference: TEC2010-19545-C04-01).
- *Estudio de nuevas técnicas de procesado de señal y codificación de canal aplicables al standard IEEE802.16e (WiMax Mobile) mediante plataforma de simulación software*: funded by Wireless Galicia.
- *COOPER@*: funded by Nomasystems S. L. .
- *MOWI*: funded by Indra Sistemas S. A. .

2

Rapid prototyping for hardware testbeds

2.1 Introduction

Vehicular radio communications have recently received a lot of interest due to the importance of the applications they will support. Several standardization proposals have arisen to define the architecture of communication systems suitable for vehicular scenarios. For instance, the IEEE 802.11p wireless standard (17) defines a convenient Physical (PHY) layer aimed at operating in vehicular environments. However, radio communications between moving vehicles (V2V) and from vehicles to the infrastructure (V2I) still need further research to reach the best possible performance. Radio testbeds like the ones presented in this Chapter and in Chapter 3 can be very useful for such research since they are able to perform realistic tests of new transmission and reception techniques.

It is important to remark first that testbeds differ from prototypes in their main goal. Prototypes try to provide full real-time operation of the techniques they integrate at the expense of limiting their flexibility to incorporate new techniques. On the other hand, testbeds try to provide the maximum flexibility to incorporate new techniques at the expense of limiting their real-time operation. Obviously, testbeds are able to perform real radio signal transmissions to preserve realism in experimental testing, buffering the acquired signals at reception for further software processing.

The fundamental role that testbeds play in bridging the gap between theory and real operation has been made clear in recent communication paradigms, such as Multiple-Input Multiple-Output (MIMO) systems. Many MIMO testbeds have been constructed (e.g. (18, 19, 20)), and the experimental tests they have allowed to perform have been valuable for researchers. In the same way, research in vehicular communications can take advantage from testbeds that are constructed specifically to enable experimental testing over the particular scenarios they address.

It must be also noted that not so long ago testbeds had to be created ad-hoc for supporting the development of a specific wireless transceiver. Then, the building process required large budgets for paying the combined work of electronic designers, radio engineers and software developers. Nowadays the situation has changed drastically: generic testbeds can be made up faster and cheaper than ever thanks to the use of plug-and-play modules that can be mixed to create systems that comply with almost any current wireless transceiver specification. Furthermore, the software to drive such modules, which formerly required a deep knowledge of the inner workings of the hardware, can be also developed really fast with the help of rapid prototyping software tools.

The testbed presented in this Chapter and the one described in Chapter 3 were built taking advantage of the latest rapid prototyping technologies. Both platforms are made up of a combination of different hardware modules manufactured by the company Sundance Multiprocessor Technology Ltd. and their higher software layers can be accessed for developing software transceivers with the aid of rapid prototyping software tools.

Before showing how to build a specific hardware platform for vehicular environments, in Section 2.2 it is described how to construct a generic MIMO testbed. The know-how obtained from the process of building such platform cannot be only applied to the Single-Input Single-Output (SISO) vehicular testbed described in Chapter 3, but to many other generic or specific hardware demonstrators.

To obtain a fully-functional testbed, it is required the implementation of transmission signaling methods and signal processing algorithms. Such methods and algorithms can be really difficult to implement, even using a testbed instead of a prototype, since low-level programming is involved to access the hardware, thus making it also difficult to easily test new methods (21). This reason motivated the design of a distributed multi-layer architecture that acts as the access interface to a radio testbed (such architecture will be detailed later in Section 2.6). The distributed muti-layered paradigm designed enables the user to operate the platforms described in this Thesis remotely and at a high abstraction level (i.e. using MATLAB/Simulink).

To show the performance of the MIMO testbed developed with two antennas at transmission and two antennas at reception (2×2), this Chapter exposes the results obtained in three real indoor scenarios where the performance of different space-time block coding schemes was evaluated.

The rest of this Chapter is structured as follows. Sections 2.2 to 2.4 describe the hardware used by the MIMO 2×2 testbed. Section 2.5 compares the features of the platform implemented with the ones from other MIMO platforms. Section 2.6 details the software that controls the platform, whilst Section 2.7 exposes the experimental results. Finally, Section 2.8 is devoted to the conclusions of the Chapter.

As it can be concluded from this structure, the present Chapter will deal with the two different elements of the rapid prototyping system highlighted in Fig. 2.1: the hardware testbed and the control software.

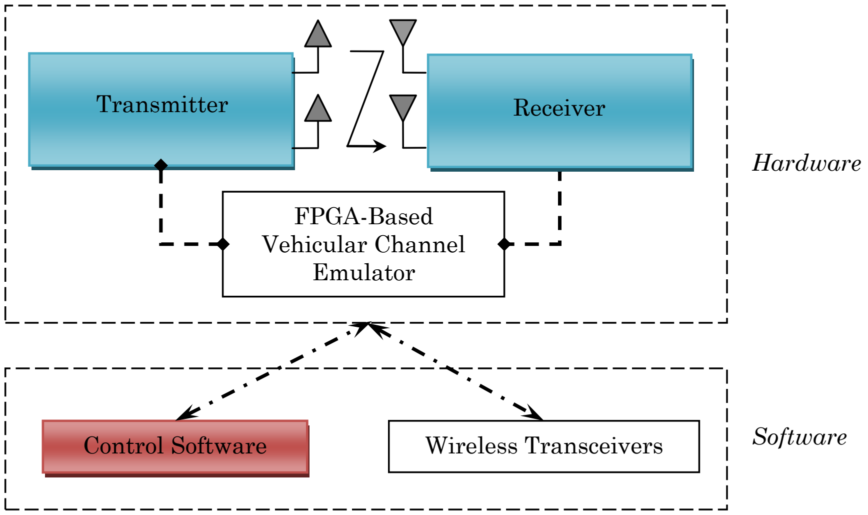

Figure 2.1: In color: components of the rapid prototyping system that are described in Chapter 2.

2.2 General View of the MIMO 2×2 Testbed

Before building an optimized vehicular hardware platform, a MIMO 2×2 testbed was constructed using rapid prototyping techniques. Such MIMO testbed was developed jointly by the universities of A Coruña and Cantabria with the aim of testing MIMO baseband modules. A schematic diagram of the platform is shown in Fig. 2.2 and a picture of the real system is shown in Fig. 2.3.

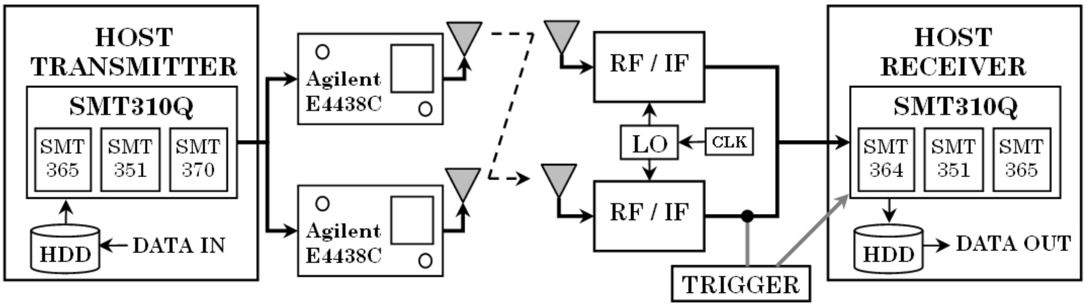

Figure 2.2: Schematic diagram of the MIMO 2 × 2 platform.

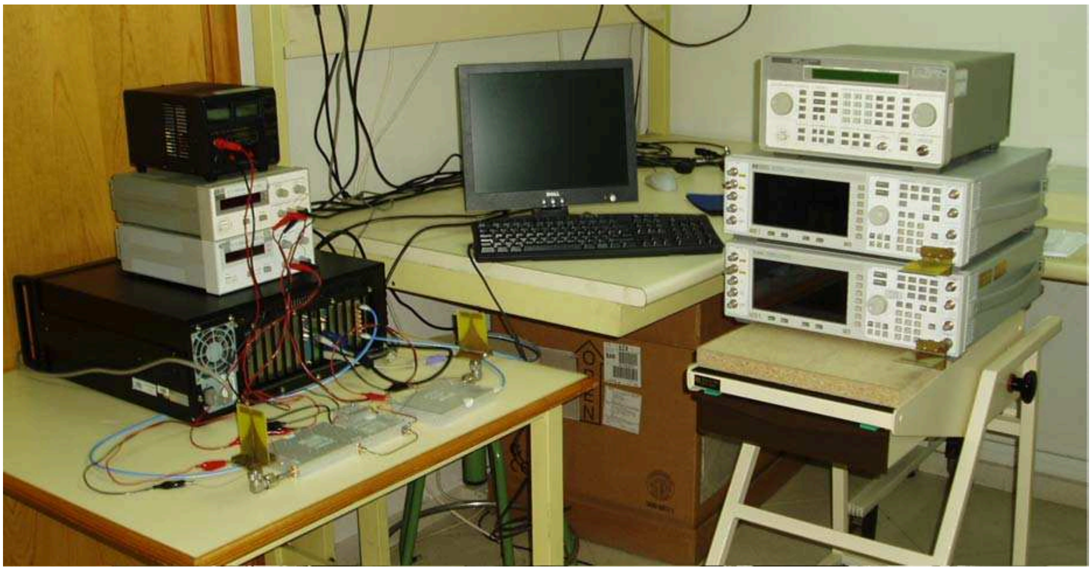

Figure 2.3: A picture of the MIMO 2 × 2 platform.

The elements that constitute the platform are described in detail in the next Sections, but first, it is important to see a general view of the way it works, which is basically as follows:

1. Signal generation, modulation and space-time coding at transmission are carried out offline using MATLAB.

2. The transmitting PC contains a board that generates the analog signals at an intermediate frequency (IF) of 15 MHz. Since this board is equipped with a large (1 GB) fast memory, the versatility of the platform is extremely high.

3. The upconversion from IF to the carrier radio frequency (RF) of 2.385 GHz is performed by two Agilent ESG E4438C signal generators and the signals are then transmitted through two printed dipole antennas.

4. At the receiver side, two downconverters specifically designed for this platform move the RF signal to IF.

5. The IF signals are acquired by the receive host PC using another board with two analog-to-digital converters (ADCs) with a maximum sampling frequency of 105 MHz. Another fast and high capacity (1 GB) memory module is used to store the acquired signals.

6. The memory content can be subsequently downloaded into the hard disk of the receiver host PC where synchronization, channel estimation, demodulation and decoding are performed offline using MATLAB.

2.3 Baseband Modules

2.3.1 Transmitter

The transmitting host (represented in Figs. 2.4 and 2.5) uses a Sundance SMT310Q PCI carrier board containing the following modules, all compliant with the Texas Instruments Module (TIM) standard :

- An SMT365 digital signal processor (DSP) module equipped with the DSP TMS320 C6416 from Texas Instruments, 16 MB of ZBTRAM (Zero Bus Turnaround RAM) and a XILINX® Virtex–II XC2V1000 FPGA.

- An SMT351 module with 1 GB of DDR SDRAM (Double Data Rate Synchronous Dynamic Random Access Memory) First-In First-Out (FIFO) memory.

- An SMT370 DAQ module equipped with one AD9777 dual digital-to-analog converter (DAC) from Analog Devices.

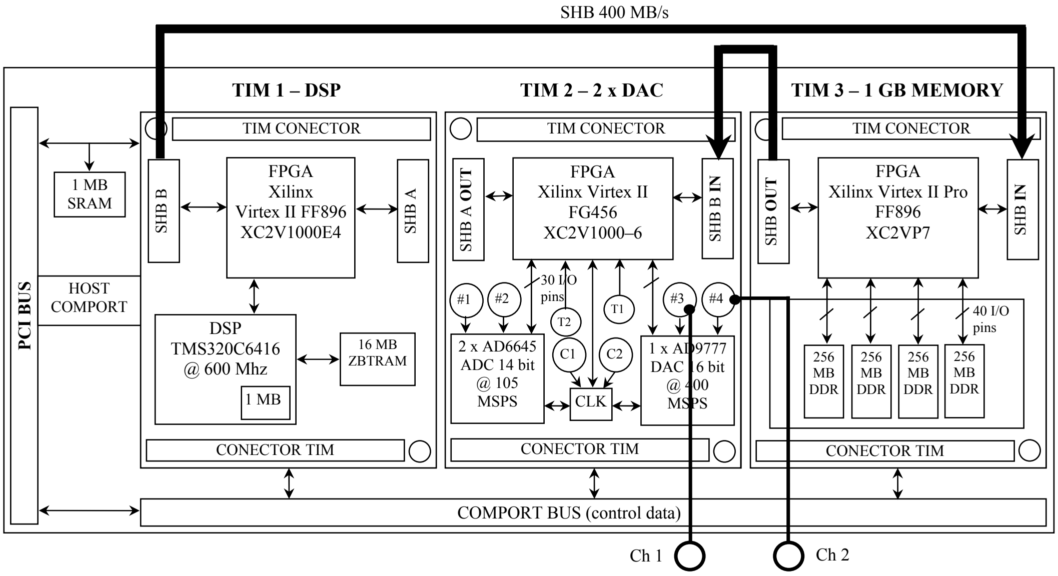
SHB 400 MB/s
TIM 1 – DSP
TIM 2 – 2 x DAC
TIM 3 – 1 GB MEMORY
TIM CONECTOR
PCI BUS
1 MB SRAM
HOST COMPORT
SHB B
SHB A
FPGA Xilinx Virtex II FF896 XC2V1000E4
DSP TMS320C6416 @ 600 Mhz
1 MB
16 MB ZBTRAM
SHB A OUT
FPGA Xilinx Virtex II FG456 XC2V1000–6
SHB B IN
30 I/O pins
#1
#2
#3
#4
T1
T2
C1
C2
2 x AD6645 ADC 14 bit @ 105 MSPS
CLK
1 x AD9777 DAC 16 bit @ 400 MSPS
SHB OUT
FPGA Xilinx Virtex II Pro FF896 XC2VP7
SHB IN
40 I/O pins
256 MB DDR
CONECTOR TIM
COMPORT BUS (control data)
Ch 1
Ch 2

Figure 2.4: Schematic diagram of the components of the MIMO 2 × 2 transmitter.

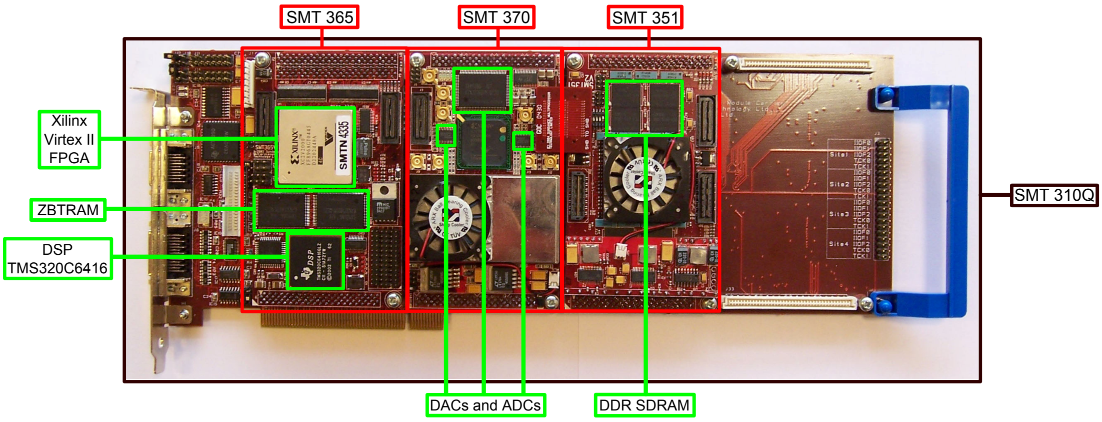

Figure 2.5: Picture and components of the MIMO 2 × 2 transmitter.

The data transmission between these modules is carried out through Sundance High Speed Buses (SHB). Such buses work at a clock frequency of 100 MHz with a total bandwidth of 32 bits, what results in a maximum transfer data rate of 400 MB/s. The SHB is divided into two 16-bit Sundance Digital Buses (SDB) that are connected to the two transmitting channels. Thus, data are transferred from the memory to the DAC in a multiplexed mode at a maximum speed of 200 MB/s per channel. The transmitting process is controlled by the SMT365 module that contains a DSP which has 1 MB of internal memory and runs at 600 MHz. Apart form the DSP, this module also contains:

- A Xilinx Virtex-II XC2V1000 FPGA (Field-Programmable Gate Array) that is used to control the SHB FIFO queue and implements the protocol to control the SHB and the ComPort buses.
- A 16 MB ZBTRAM that can be accessed using the EMIF (External Memory Interface) address space at a maximum clock frequency of 133 MHz. This memory is used both as the address space for the DSP application and as a buffer for data transmission. The data set to be transferred through the DAC is read directly from files and stored in the SMT351 memory. During this process, the 16 MB of ZBTRAM are used as a temporary buffer in order to increase the speed of the transfer.
- A Flash ROM (Read-Only Memory) of 8 MB that is used to store the FPGA bit stream and the boot code for the DSP.

The SMT365, together with the SMT310Q carrier board, is able to send and receive data from/to the host using the PCI bus. This feature allows the platform to transfer large amounts of data at high speeds between the host and the SMT351 module.

The ComPort bus is used to send the control signals from the SMT365 module to the other modules of the SMT310Q carrier board at a transfer rate of up to 20 MB/s. The SMT351 and the SMT370 are configured by default to read control data from the ComPort bus.

The SMT351 module is used as a very large FIFO queue for the SHB bus. It is needed because the SMT370 DAC may operate at a clock frequency of 100 MHz, which is too high for the DSP. Therefore, the data is previously sent to the SMT351 in a cyclic process that consists in reading the data from the host and storing them in the SMT351 memory. After reading the data, the transmission starts and the data are transferred from the SMT351 to the SMT370 at the desired speed. Thus, the system allows transferring up to 1 GB of data, which is enough for most scenarios.

The SMT351 module is equipped with a Xilinx Virtex-II Pro FPGA XC2VP7 and 1 GB of double data rate SDRAM at 133 MHz that is divided into four memory banks of 256 MB which can be accessed in parallel.

Finally, the SMT370 module consists of a Xilinx Virtex-II FPGA and a dual AD9777 DAC from Analog Devices with 16 bits of resolution and a maximum sampling frequency of 400 MHz (with interpolation, 160 MHz without interpolation). Also, the SMT370 module contains two AD6645 ADCs that are not used by the transmitting host.

The SMT370 has MMBX terminals for the input and the output signal connectors, for the clock input and for the trigger signals. Two MMBX-to-BNC connectors are used to transport the signal from/to the SMT370.

The complete configuration is shown in Fig. 2.4, where the SMT365 module is located in TIM 1, the SMT370 in TIM 2 and the SMT351 in TIM 3. The system requires two ComPort connections: one to connect the SMT365 to the SMT370 and another to connect the SMT365 with the SMT351. Also, there is an SHB link between the SMT365 and the SMT351 and another one between the SMT351 and the SMT370.

At the transmitter the operations of encoding, modulation and generation of the data files are performed offline in MATLAB. The data files generated in MATLAB (one per each of the two channels) are read and processed by the DSP application. The DSP program has been programmed in C using the 3L Diamond software which is specifically designed to work with the modules and buses previously described. The DSP program also performs the following tasks:

- Configuration of the transmitter: adjustment of the DAC, initialization of the communication ports, etc.
- Reading of the data files from the host.
- Data transferences from the DSP to the 1 GB memory module through the SHB. These data will be later sent from the memory module to the DAC.

2.3.2 Receiver

The receiving host consists of two devices: an external trigger and a Sundance SMT310Q PCI board which carries modules with ADCs, DSPs, FPGAs and a large amount of memory.

The external trigger helps with a crucial aspect of the acquisition process: the detection of the presence of the transmitted signal at the receiver. Although such task can be done by performing software synchronization on the received signal, this technique has several drawbacks when it is carried out on the testbed described in this Chapter:

- There exists a large delay between the signal detection and the beginning of the acquisition what leads to lose most of the data sent.
- Very poor computational efficiency: the involved algorithms are quite heavy and slow.
- It is required to adjust the transmission/capture times to compensate the monitoring delays.

To avoid these limitations, a custom triggering hardware device was designed (its picture is shown in Fig. 2.6). This device monitors the received signal power, waking up the ADCs of the receiving host when the detected power goes beyond a prefixed threshold. This external trigger presents the following advantages:

- Because of being external, it is independent from the rest of the MIMO platform.
- The delay between the signal detection and the beginning of the acquisition is limited to a few microseconds (the sampling time of the trigger's ADC plus the time to switch on the outputs).
- The power threshold can be easily configured by means of the onboard switch.

The external trigger has two main components: a power detector and a microcontroller. The power detector chosen is the LTC5507 from Linear Technology. This integrated circuit accepts input signals between 1 KHz and 1 GHz, quantifies the power of the signal and sends a voltage level to the ADC of the microcontroller. Such microcontroller is a Philips LPC936 that was chosen due to its fast ADC and the availability of its evaluation board, which allows embedding easily the microcontroller. The LPC936 monitors the voltage coming from the power detector, activating the ADCs of the receiving host when it exceeds the prefixed threshold. The microcontroller software was developed in C using the LPC935 and LPC936 libraries.

The hardware of the receiving host is quite similar to the hardware used in the transmitter but instead of DACs, it has ADCs (see Fig. 2.7). It consists of a Sundance SMT310Q PCI board which contains the modules SMT365 (with a DSP, an FPGA

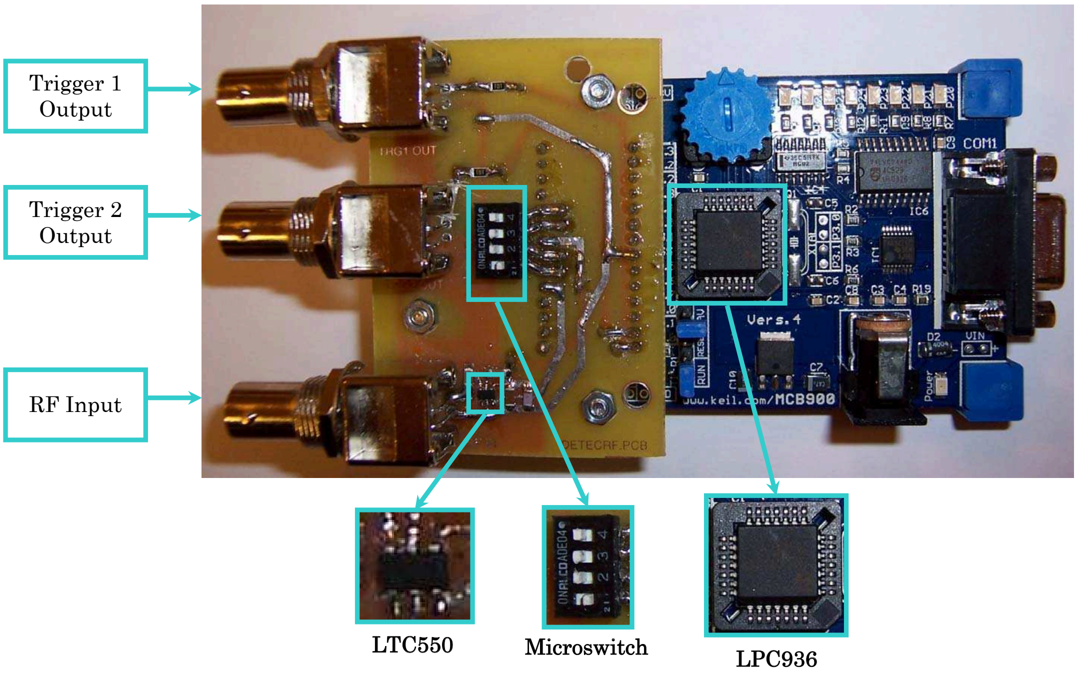

Figure 2.6: Hardware trigger for the MIMO 2 × 2 testbed.

and 16 MB of ZBTRAM), SMT351 (1 GB of memory that acts as a FIFO memory) and SMT364 (with four ADCs). Each ADC is an AD6645 from Analog Devices, with a resolution of 14 bits and a maximum sampling frequency of 105 MHz.

When the trigger activates the receiving host, the capturing process starts. The signals go through the ADCs and the digital samples are stored in the memory module. When the memory is full, the DSP writes the data (or only a part of all the data) into the computer's hard disk. The stored data files can then be processed in MATLAB.

The software of the receiving host was programmed in C using the 3L Diamond API to access Sundance hardware. This software performs the following tasks:

- Configuration of the receiving host (initialization of the ADCs, communication ports, the FIFO memory module, etc.).
- Writing to the hard disk: the sampled data are read from the memory module and then written into the receiver's hard disk.

Thanks to this flexible scheme, it is really easy to use the MIMO platform with a

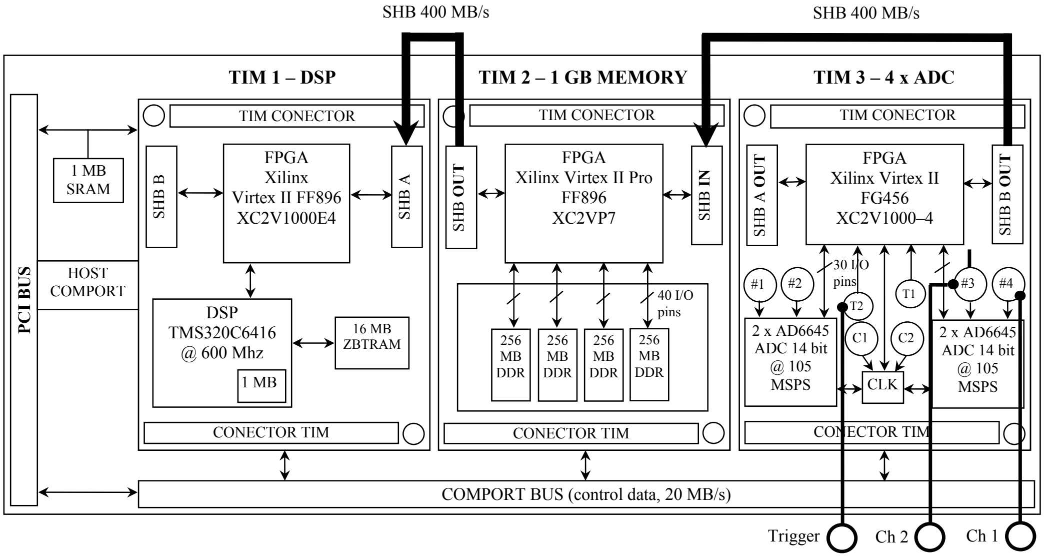

Figure 2.7: RX board schematic of the MIMO 2 × 2 testbed.

transmitter and a receiver implemented in MATLAB (or in other programming languages).

2.4 RF Modules

The upconversion from an IF of 15 MHz to the carrier RF of 2.385 GHz is performed by two Agilent ESG E4438C signal generators. RF signals are then radiated by two printed dipole antennas. Note that the signal generators are used merely as upconverters.

At reception, two downconverters translate the RF signal to the IF. The downconverters have been specifically designed for this platform and their main characteristics are the following: bandwidth of 20 MHz (2.385 ±10 MHz), gain of 50 dB, noise figure less than 2 dB and sensitivity of -88 dBm.

As of writing this Thesis, the use of the mentioned up/downconverters makes no sense in terms of economic cost. Such devices work great and are very flexible, but their functionalities are excessive for an ad-hoc testbed. It is cheaper to add to each SMT 310Q board a Sundance SMT 349 module (as we did in (5)) that works as follows:

- The SMT 349 module of the transmitter upconverts the IF signals into an RF carrier frequency of 2.45 GHz. The IF signals have to be created and sampled with the objective of generating a replica at 70 MHz (the SMT 349 module takes its input at such frequency).

- The receiver board have to contain another SMT 349 module. In this case the module captures signals at 2.45 GHz and downconverts them to 70 MHz. The ADCs have to sample the signal at a specific frequency in order to obtain a replica at the desired IF.

2.5 MIMO 2×2 Platform Versus Other Platforms

In recent years many researchers have been working on their own multi-antenna hardware platforms to test new techniques and algorithms. Such platforms can be classified into two main groups. The first group is composed of testbeds aimed at implementing certain standard or specification, while the second group is formed by the ones that are generic. Examples of testbeds of the first type are the narrowband MIMO prototype in (22), the MIMO WCDMA (Wideband Code Division Multiple Access) for 3G telephone systems in (23), a MIMO 3G prototype for high-speed downlink packet access reception (24), or a MIMO OFDM testbed for 4G telephone systems developed by ETRI (25). This kind of platforms usually exhibits good technical characteristics but they are extremely expensive and barely flexible.

The MIMO testbed exposed in this Chapter belongs to the second group of platforms, i.e., general purposed testbeds. Other examples of this type of platforms can be found in (19, 20, 26, 27, 28, 29, 30, 31, 32, 33, 34). If the features of the implemented MIMO platform are compared with the others found in the testbeds cited, it can be concluded that:

- The implemented platform is very flexible at transmission because it makes use of commercial signal generators. However, at reception the testbed is limited to a carrier frequency of 2.4 GHz and a bandwidth of 20 MHz. Regarding this aspect, the platform implemented outperforms or equals the single-band platforms at 2.4 GHz (20, 26, 27, 29, 30, 33, 34) but is not capable of receiving signals at higher frequencies and/or higher bandwidths like (19, 28, 31, 32).

- Memory storage capacity and speed are one of the most remarkable features of the platform implemented since it has a storage capacity of 512 MB per channel (both at reception and transmission) accessible at the maximum speed of the DACs and ADCs. This storage capacity is far superior to those of the other platforms.

- The implemented platform has two DSPs with higher computational power than most of the other platforms. Only (27) and (34) make use of DSPs that have similar or superior characteristics. In terms of FPGA computational capacity, the testbed implemented is only worse than (28) but outperforms (19, 20, 30, 33, 34).
- The transfer speed between the processing, storage, generation and acquisition units inside the platform (which is crucial when processing real-time wideband signals) reaches, as well as (20), the highest transfer speed between modules among all the testbeds compared. This fact is due to the use of the proprietary SHB bus, which is able to transfer data at a maximum speed of 400 MB/s.

2.6 Multi-layered Software Architecture for Testbed Automation

The control of a testbed usually involves cumbersome low-level programming tasks to access the hardware, making it difficult to easily test new methods. For this reason, it is desirable to incorporate an end-user interface to the MIMO testbed that would allow researchers to focus exclusively on the development of new MIMO techniques, releasing them from low-level programming. This subsection describes a software architecture that provides a testbed with an abstraction layer according to the access level needed by signal processing researchers (i.e. MATLAB or similar software). The separation of the signal processing implementation from the hardware access control and configuration leads to exploit the hardware testbed more efficiently. Moreover, to facilitate the task of extensive experimental trials the software developed allows the testbed to be accessed from any remote personal computer (PC) connected to the Internet.

The multilayer software architecture is composed of three layers: a middleware that interacts with the testbed hardware, a signal processing chain that converts discrete symbol sequences into IF signals at the transmitter (and vice versa at the receiver), and an end-user layer that presents the testbed to the user at a high abstraction level. To support a wide range of communication scenarios and end-user needs, the software architecture is designed under the premise of minimizing dependencies between its layers. Thus, each layer can be extended or customized to satisfy different requirements only affecting the others slightly. Moreover, since each layer should be able to run on a different PC, the layer intercommunication is performed through sockets over standard network links.

2.6.1 General View of the Architecture

As it can be seen in Figs. 2.8 and 2.9, the testbed hardware is composed of two ordinary PCs, one for the transmitter (Transmit PC in Fig. 2.8, but called Tx PC in the following subsections) and another one for the receiver (Receive PC in Fig. 2.8, but called Rx PC in the next subsections) that run the processes TxHost, TxDSP, RxHost and RxDSP, which constitute the middleware. The hardware of the transmitters/receivers implemented is described in Sections 2.2 and 3.4. Regarding the middleware, it works at the lowest software level and is required to be installed in the same computers as the testbed hardware. More details on the middleware can be found in Section 2.6.3.

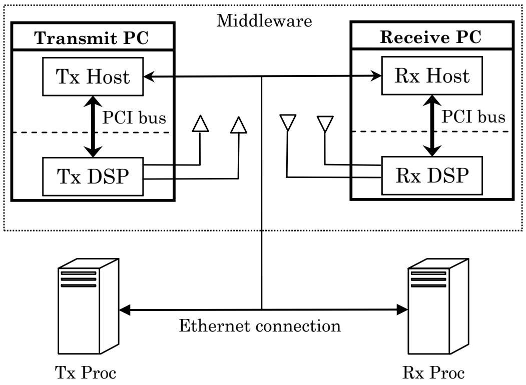

Figure 2.8: Generic architecture for a MIMO testbed.

In order to minimize the time the Tx-Rx PCs are busy, the rest of the architecture layers are detached from them, so different signal processing chain layer instances can run simultaneously in different PCs. Finally, note that, as it can be observed at the top of Fig. 2.9, there exists an end-user layer, which has to be installed in the same computer as the end user programs.

Data frames are transmitted as follows:

- After the discrete-time signals to be transmitted have been generated by the PHY layer, they are transferred to the multilayer software architecture through the user

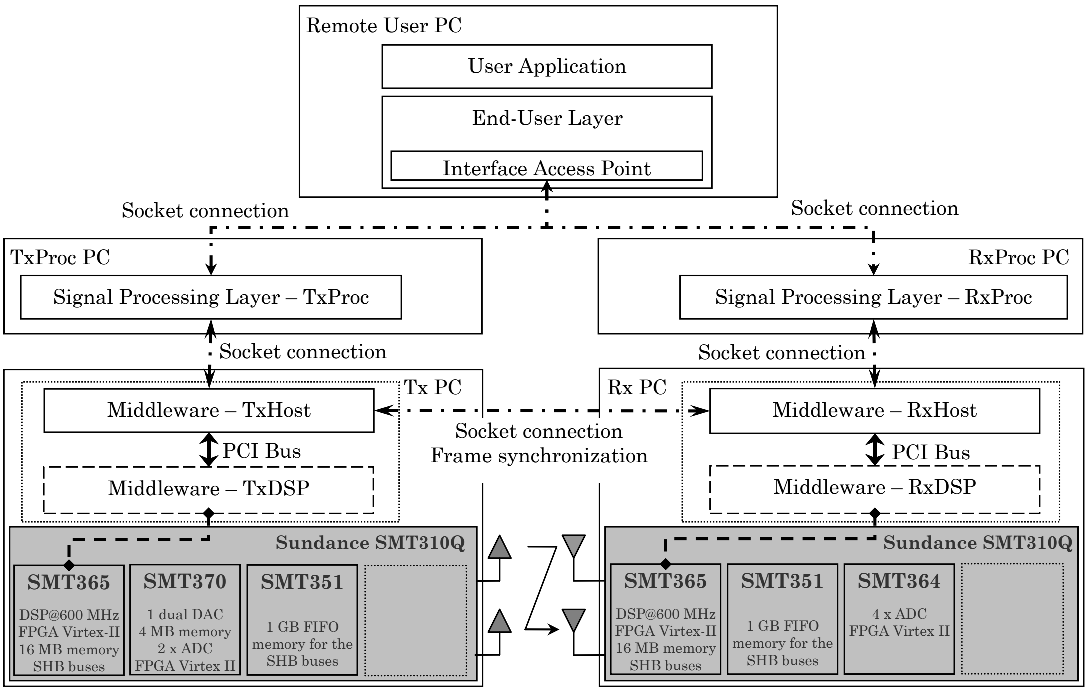

Figure 2.9: Generic MIMO testbed architecture.

layer entry point (currently it is implemented as a MATLAB function).

- The samples are then forwarded from layer to layer where additional processing may be performed if desired. Each layer at transmission is linked with its counterpart at reception by a protocol based on standard TCP/IP connections.

- When the middleware layer of both PCs is ready to perform a transmission, the signals are passed to the testbed hardware, where digital-to-analog conversion and RF upconversion are performed.

- At the receiver side, the middleware stores the acquired signals into the hardware buffers and then forwards them to the signal processing layer.

- At this moment, the Tx-Rx PCs are ready to process another frame while the acquired signals are processed by the receiver signal processing layer. The signal processing layer performs specific tasks (e.g. frequency offset compensation and time synchronization) before sending the data to the user layer.

- Finally, samples are processed by the user application (e.g. the IEEE 802.11p receiving chain).

2.6.2 Protocol Operation

Fig. 2.10 shows a basic sequence diagram of the protocol. In addition to TxHost, RxHost, TxDSP and RxDSP, two elements are added (TxProc and RxProc) that are, respectively, the signal source and the signal sink. The designed protocol consists of the seven stages described below. Other stages can be added in order to achieve a more complex behavior.

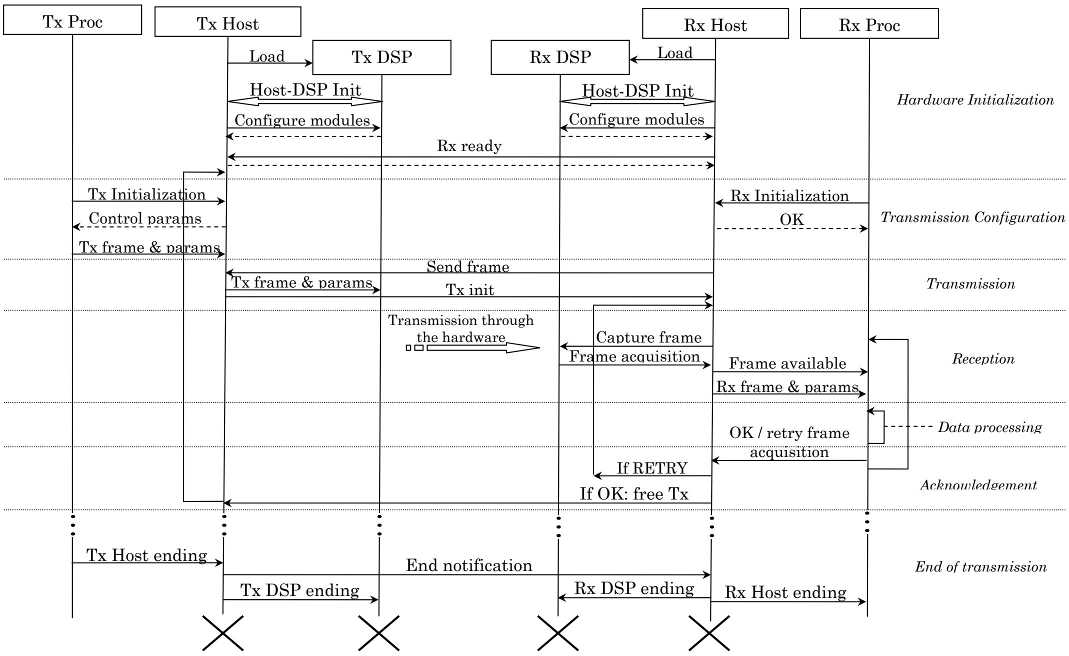

Figure 2.10: Example of a frame transmission through the MIMO 2 × 2 testbed.

- Hardware initialization. Both transmitter and receiver hardware must be set up before beginning a transmission. TxHost and RxHost load the binaries of the TxDSP and RxDSP into the DSP. Once loaded, these processes configure the ADCs/DACs, initialize the RF modules and reserve memory for the communications through the PCI bus.

- Transmission configuration. After connecting to the signal source (TxProc) and before transmitting any data, TxHost negotiates and sets the transmission parameters. Control parameters such as the frame size or the number of frames to be transmitted are configured. Moreover, other internal parameters, such as the signal power, can be sent to TxHost during this stage.

- Transmission. Once TxHost has received all the parameters, the transmission process can begin. First, RxHost must notify that it is able to receive frames. Then, TxHost configures the transmit DSP with the proper parameters (received during the Transmission Configuration stage) and the transmission starts: the DSP receives the signals from TxHost and sends them to the DACs. After that, the signals go to the RF module where the upconversion is performed.

- Reception. To acquire a signal the receiver has to ask the DSP to capture a new frame. Then, the RF hardware begins to downconvert the signal received, which is first sent to the ADCs and then to the 1 GB module. When this memory module is full, the data are sent through the PCI bus to the receiver PC.

- Data processing. In the platform described in this Chapter and in Chapter 3 data processing is carried out offline, but it could be done using real-time processing by means of, for instance, an FPGA. Basically, during this stage the received data are processed, performing operations such as demodulation, channel estimation, detection, error correction, etc.

- Acknowledgement. After processing the data, the receiver must tell the transmitter what to do next. The receiver can ask for a retransmission of the last frame or for a new frame. In the former case, the receiver moves to the Transmission stage. In the latter, the receiver needs to prepare a new transmission, so it must go back to the Transmission configuration stage.

- End of transmission. TxProc sends a message to TxHost when it has transmitted the number of frames desired. Then, TxHost sends a signal to the receiver, starting the closing of all the connections. The DSP processes are also closed because they are no longer needed.

2.6.3 Middleware Layer

The middleware layer links the platform hardware and the PHY layer, allowing discrete-time signals to be transferred through the PCI bus and making possible the synchronization between the Tx PC and the Rx PC using a TCP/IP network connection.

The middleware architecture is split into two different sublayers. Fig. 2.9 shows the middleware with its two sublayers plus the connections with the adjacent layers. The sublayer at the top (Tx/RxProc) is responsible for establishing the network connections between the transmitter and the receiver, and with higher layers (like the signal processing layer). The middleware sublayer at the bottom corresponds to the testbed hardware configuration and control software.

The middleware is constituted by four different processes. The first two (TxHost and RxHost) run, respectively, on the Tx PC and the Rx PC. They are implemented in standard C++ and use sockets to establish the necessary network connections. The other two processes are the transmitter and the receiver processes that run on the DSPs available in the testbed hardware. Thus, the DSP process of the transmitter (TxDSP) performs data transfers through the PCI bus jointly with the TxHost process and configures and controls the hardware components of the Tx PC. In the same way, the RxHost process and the DSP process of the receiver (RxDSP) are responsible for transferring the data through the PCI bus and for controlling and configuring the testbed hardware components.

The middleware serves each request while the data for the next frame is still being generated by any computer in the network and, when the first frame is passed to the signal processing layer, the middleware is ready to accept a new frame. Thanks to this architecture scheme, the middleware is able to serve simultaneous requests from other layers that may be running in different PCs.

In order to free the middleware implementation from dealing with the lowest level hardware details, the Sundance SMT6025 software development kit and the Sundance SMT6300 operating system driver are used in the TxHost and the RxHost implementations, while the Texas Instruments Code Composer compiler, as well as the 3L Diamond, are employed for the TxDSP and RxDSP implementations.

The middleware concept constitutes a great leap forward in MIMO testbed technology, permitting to access the hardware platform using ordinary network connections and synchronizing the transmitter and the receiver. It also allows creating a wired feedback channel, multi-user environments or many other extensions, since it has been designed to support their addition. Moreover, although the DSP processes are hardware-dependent, the control logic between the transmitter and the receiver (as well as the communications with the higher layers) is valid for different hardware platforms. Finally, since the hardware interfaces from the host side and the software related to DSPs are developed as separated software modules, they can be replaced by implementations suitable for other testbeds.

2.6.4 Use Cases

To show how relevant experimental setups can be easily implemented thanks to the middleware layer proposed, four different common use cases are described below. Such cases illustrate the flexibility of the middleware developed:

- Burst transmission. This case involves the transmission of a number of data frames with the objective of testing a specific transmission system. Fig. 2.10 can be used to show how the protocol works: the transmitter does not send new frames until the receiver notifies (during the acknowledgement stage) that the current frame has been received correctly.

- Creation of performance measurements. Another typical case of use of a testbed consists in generating performance measurements for evaluating a particular algorithm or a specific transmission system. In this case the protocol works in a quite similar way as in a burst transmission: during the data processing stage the measurements are performed and the receiver decides whether it has to keep on doing measures or it has already collected enough data. For instance, this functioning mode was used when doing performance measures in Sections 2.7.1 and 3.6.

- Using a feedback channel. The performance of a wireless communication system can be improved if the receiver feeds back the channel state information (CSI) to the transmitter, so it can adapt itself to the channel characteristics. It is straightforward to implement a feedback channel within the protocol: the CSI packet must be sent during the acknowledgement stage to notify the transmitter whether it has to reconfigure the transmission parameters or not. To do so, in Fig. 2.10 there should be added two messages 'Send CSI': one message from RxHost to TxHost and another one from TxHost to TxProc.

- Working in a multiuser environment. This final example describes the functioning of the protocol designed when working in an environment where several terminals can receive information from a central node (in a multicast-like communication). The protocol just needs to be generalized to support the synchronization of several terminals. An example of use is represented in Fig. 2.11 where two users receive frames from a central node. Although the protocol behavior could be more complex, for the sake of clarity, the example shown assumes that the transmitter only sends a new frame when all the users have received correctly the previous one.

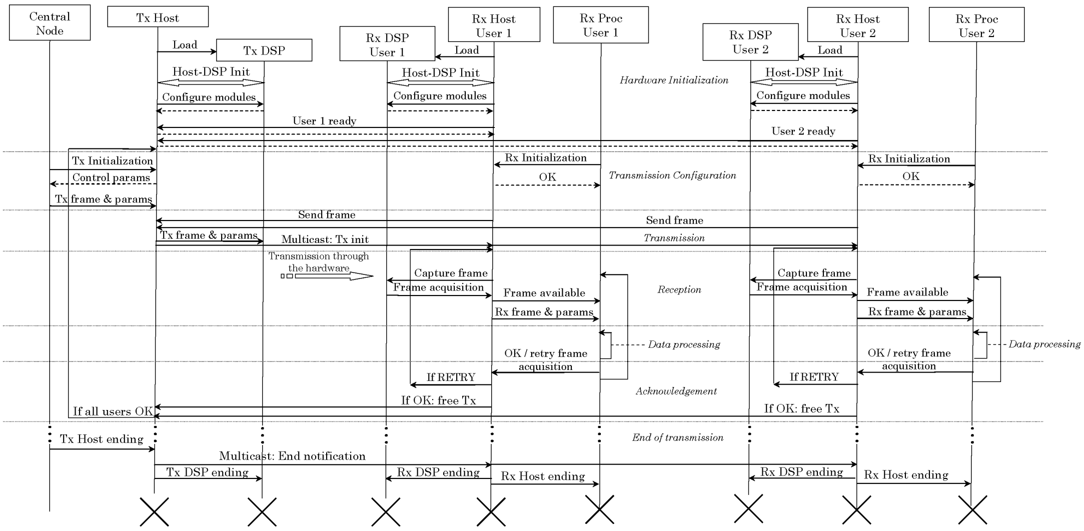

Figure 2.11: Sequence diagram of the protocol working in a multi-user environment.

2.7 Experiments: Space-Time Block Coding

Since the pioneering work of Foschini and Telatar (35, 36), multiple transmit and receive antennas have been used to improve drastically the performance of wireless communication systems (37, 38, 39, 40). Specifically, since the work of Alamouti (41), and the later generalization by Tarokh *et al.* (42), space-time block coding (STBC) has emerged as one of the most promising techniques to exploit spatial diversity in MIMO systems.

Among space-time coding schemes, orthogonal space-time block coding (OSTBC) is one of the most attractive because it is able to provide full diversity gain without any CSI knowledge at transmission and with very simple encoding and decoding procedures. The special structure of OSTBCs implies that the optimal Maximum Likelihood (ML) decoder is a simple linear receiver, which can be seen as a Matched Filter (MF) followed by a symbol-by-symbol detector. This linear receiver also maximizes the Signal-to-Noise Ratio (SNR) for each data symbol (43) using the knowledge of the channel matrix.

The CSI required for coherent detection of OSTBCs is typically acquired by sending a training sequence that is known at the receiver side (44). However, the price to be paid is reduced bandwidth efficiency and energy loss because training sequences do not carry information. Popular approaches to avoid these limitations include the so-called Differential STBC (DSTBC) schemes (45, 46, 47) and Unitary Space-Time Modulation (48, 49), which do not require channel knowledge at the receiver. However, these approaches incur a penalty in performance of 3 dB (differential coding) or 2–4 dB (unitary modulation) when compared to the coherent ML receiver (48). Moreover, the receiver complexity for the unitary scheme increases exponentially with the number of points in the unitary space-time constellation.

In order to overcome the limitations of differential codes whilst, at the same time, avoiding the bandwidth reduction of pilot-aided techniques, several methods for blind channel estimation have recently been proposed (50, 51). These methods can be divided into two groups depending on whether they exploit the Higher-Order Statistics (HOS) or the Second-Order Statistics (SOS) of the signals. HOS-based methods exhibit two major drawbacks: they present, in general, a higher computational cost and may require long streams of data to obtain accurate estimates. For these reasons, SOS-based methods are preferable in practice. Recently, a reduced-complexity SOS-based method for blind channel estimation under OSTBC transmissions has been proposed in (52). Its performance has been evaluated by means of numerical examples, finding that in most cases it renders accurate channel estimates, provided that $n_R > 1$ receive antennas are available. However, for some OSTBCs (including Alamouti) some ambiguities appear and have to be avoided using, for instance, linear precoding at the transmitter or resorting to HOS.

This subsection is focused on the evaluation of several of the above STBC transmission techniques over realistic indoor scenarios using the MIMO 2×2 testbed described in this Chapter. Due to the limitations in the number of transmit antennas, the tests were constrained to the Alamouti code (41) and the differential STBC for two transmit antennas (45). For Alamouti coherent decoding it was employed a pilot-aided CSI estimation technique (44) and the blind technique proposed in (53), which avoids the indeterminacy problems of (52) by reducing in a few bits per second the transmission rate.

A detailed description of the data model of MIMO STBC system and of the OSTBC/DSTBC scheme is given in Appendix A.

2.7.1 Characteristics of the Experiments

Three different experiments were performed to compare the STBC schemes described in Appendix A for two transmit and two receive antennas. More specifically, this subsection exposes the results obtained for the Alamouti scheme (with pilot aided channel estimation and with blind channel estimation) and for the Differential STBC. Experiments 1 and 2 took place in the laboratory of the Signal Processing Group (GTAS) at the University of Cantabria, whilst the third experiment was carried out in the laboratory of the Electronic Technology and Communications Group (GTEC) at the University of A Coruña. In the first experiment the transmitters and the receivers were approximately two meters away from each other, with a clear line-of-sight (LOS) between them. In the second experiment the transmitters were located farther away from the transmitters (≈ 10 meters) and the transmitting antennas were also moved to avoid a clear line-of-sight (see Fig. 2.12). Finally, in the third experiment, the transmitter and the receivers were approximately five meters away from each other (see Fig. 2.13). To simplify the symbol and frame synchronization tasks, it was designed a frame structure composed of 63 preamble symbols for frame synchronization, up to 64 pilot symbols for channel estimation (for pilot-aided techniques) and 1000 data symbols (see Fig. 2.14). In the preamble a pseudorandom sequence (PN) was used to facilitate frame synchronization and coarse symbol timing acquisition. Notice that this frame was selected to simplify the synchronization and estimation algorithms and not to maximize throughput.

Regarding the modulation parameters, QPSK modulation was employed. The pulse shaping filter was a square-root raised cosine filter with a roll-off factor of 0.4. The symbol rate was 1 Mbaud for the first and second experiments and 5 Mbaud for the third experiment, so the RF bandwidth was 1.4 and 7 MHz, respectively. The sampling frequency was 80 Msamples/sec. at both the transmitter and the receiver. These low symbol rates were used only for illustration purposes, in order to simplify the synchronization and channel estimation algorithms, and to obtain a flat fading channel. There is no problem for the baseband hardware to achieve symbol rates of up to 20 Mbaud. At the receiver, carrier offset estimation was performed to eliminate the carrier modulation. Afterwards, frame and symbol synchronization were carried out by exploiting the PN preamble. The final baseband observations were obtained through matched filtering and sampling at the symbol rate.

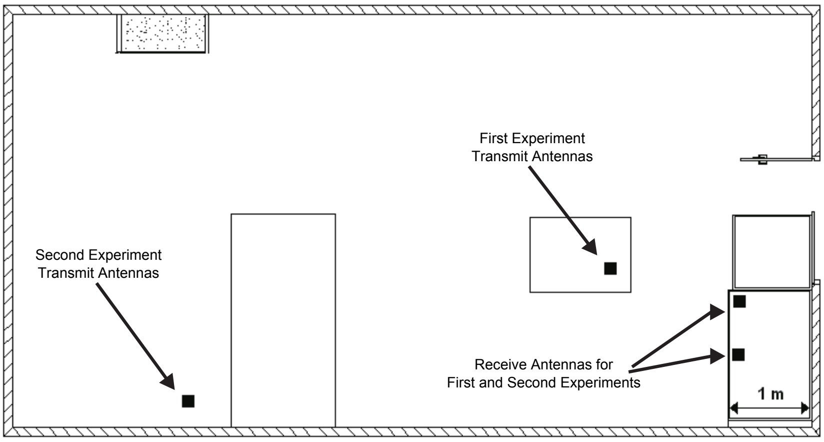

Figure 2.12: Antenna locations for the MIMO 2 × 2 STBC experiments at the University of Cantabria.

2.7.2 Results

An example of the signals received in the first two scenarios (LOS and Non-LOS (NLOS) at R_s=1 Mbaud.) is shown in Fig. 2.15, where it is represented the signal received at one antenna (upper left), the signal after time synchronization (upper right), after synchronizing in time and frequency (lower left) and after Alamouti decoding (lower right). Note that in such example the signals were transmitted in a LOS scenario with a transmitted power per antenna of -10 dBm.

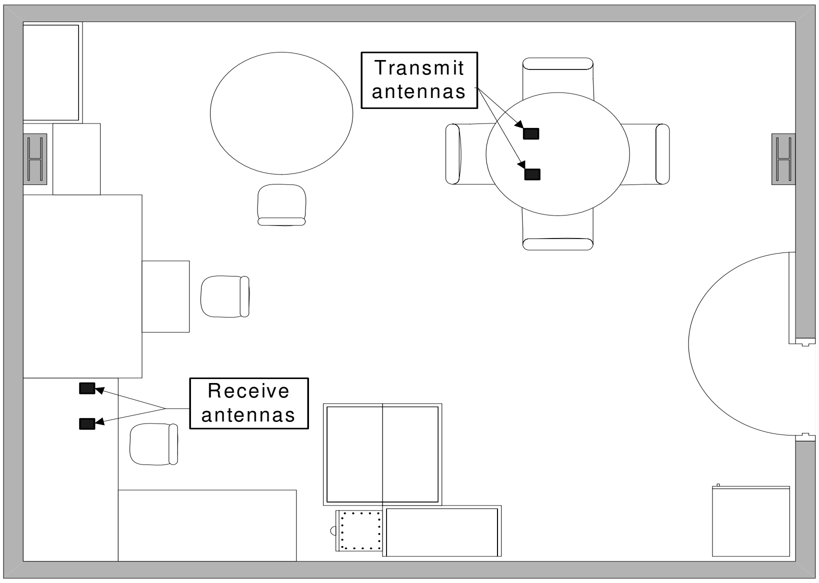

Figure 2.13: Antenna locations for the MIMO 2×2 STBC experiments at the University of A Coruña.

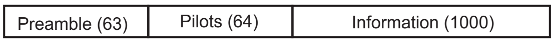

Figure 2.14: Frame structure chosen for the MIMO 2×2 STBC experiments.

For both scenarios (LOS and NLOS) the experiment was repeated varying the transmitting power per antenna and averaging the obtained results. Since the generation and coding at the transmitter side, and the demodulation, channel estimation and decoding at the receiver side are carried out offline, the time between two consecutive trials is much larger than the coherence time of the channel, so it can be assumed a block-fading channel. With this set-up it was obtained the bit error rate (BER) curves versus transmitting power that is shown in Figs. 2.16 and 2.17 (for LOS and NLOS, respectively). Specifically, in these figures it is compared:

- The Alamouti scheme with pilot-aided channel estimation (labeled as K pilots).
- The Alamouti scheme with blind channel estimation, labeled as Blind (BX-NY), where X is the number of Alamouti blocks in which we eliminate one real symbol (to avoid the ambiguity) and Y is the number of blocks used to estimate the correlation matrix.

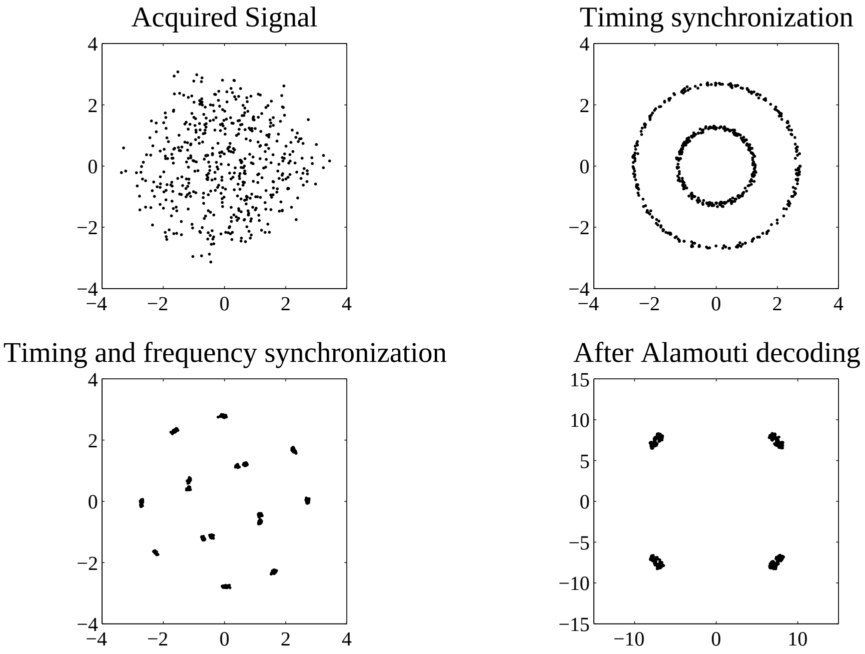

Figure 2.15: Symbol constellations at the receiver during the MIMO 2 × 2 STBC experiments.

- The DSTBC.

It is important to notice that both channel estimation methods (pilot-aided and blind), which are used for coherent detection, incur in a rate penalty (see Table 2.1).

Method	Rate
64 Pilots	0.9398
16 Pilots	0.9843
Blind (B10-N250)	0.9750
Blind (B10-N500)	0.9750

Table 2.1: Rate for the different channel estimation methods used during the MIMO 2×2 STBC experiments.

As it can be seen in Figs. 2.16 and 2.17, the blind technique with $N = 500$ blocks achieves almost the same performance as the pilot-aided method with 64 pilots. This

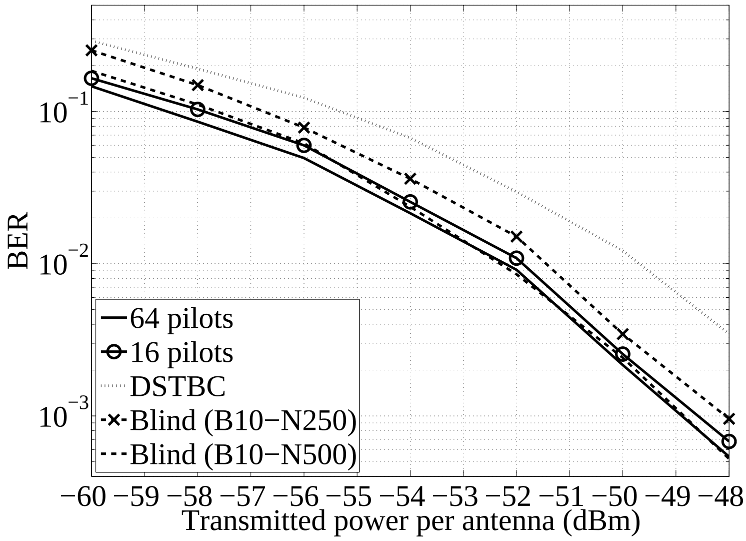

Figure 2.16: BER for the LOS scenario during the MIMO 2 × 2 STBC experiments.

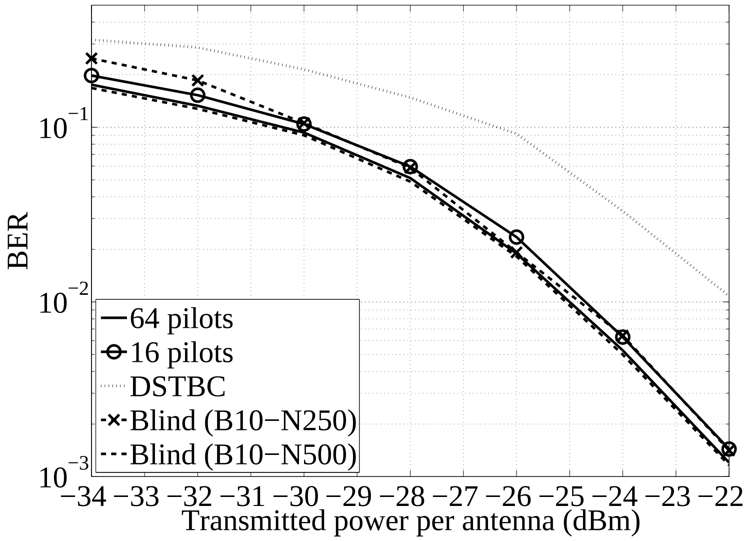

Figure 2.17: BER for the NLOS scenario during the MIMO 2 × 2 STBC experiments.

improvement is achieved at the expense of a moderate increase of the computational cost, since the blind technique has to obtain the main eigenvector of a 8×8 correlation matrix (for this particular setup: Alamouti coding and complex modulation). On the other hand, it can be also observed the expected 3 dB loss for the DSTBC and that the pilot-aided method with 16 pilots losses about 0.4 dB with respect to the same technique with 64 pilots. No more than 64 pilots are used since the channel estimate obtained with more pilots does not improve the performance of the system.

Finally, if fewer blocks are used for channel estimation with the blind technique, the estimate of the correlation matrix is worst and this causes a loss in BER. Specifically, if we use $N = 250$ instead of $N = 500$ blocks, the loss is about 0.9 dB. However, the use of a reduced number of blocks for blind channel estimation permits the use of shorter frames, which is especially important when the channel coherence time is smaller. These conclusions are valid for both the LOS and NLOS scenarios. The main difference between the two examples is that the NLOS requires an increase of almost 27 dBm of transmitting power to attain the same BER.

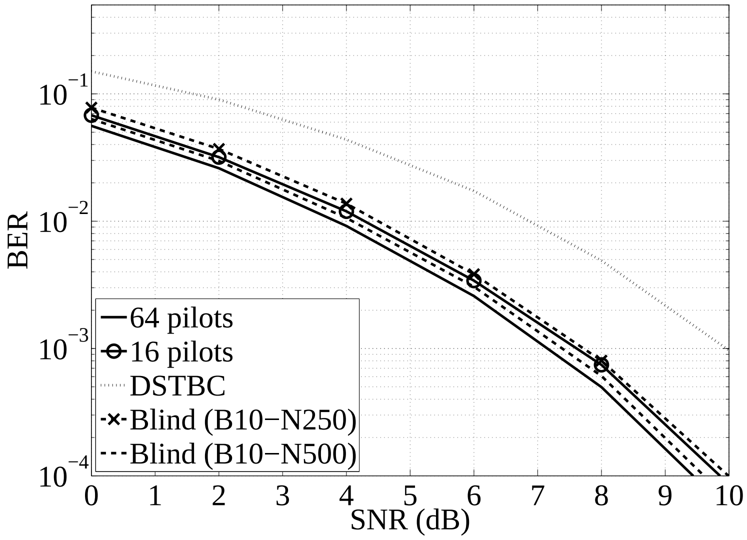

Figure 2.18: BER for the Rician simulation during the MIMO 2 × 2 STBC experiments.

To gain more insight about the experiments, Figs. 2.18 and 2.19 include theoretical references obtained through simulation. Although a perfect characterization of the channel was not obtained, it was found that a Rayleigh fading and a Rice fading channel model with a Rice factor of 3 dB are reasonable approximations for the behavior observed in NLOS and LOS scenarios, respectively. Regarding the behavior of the different STBC transmission techniques, the conclusions reached for both the

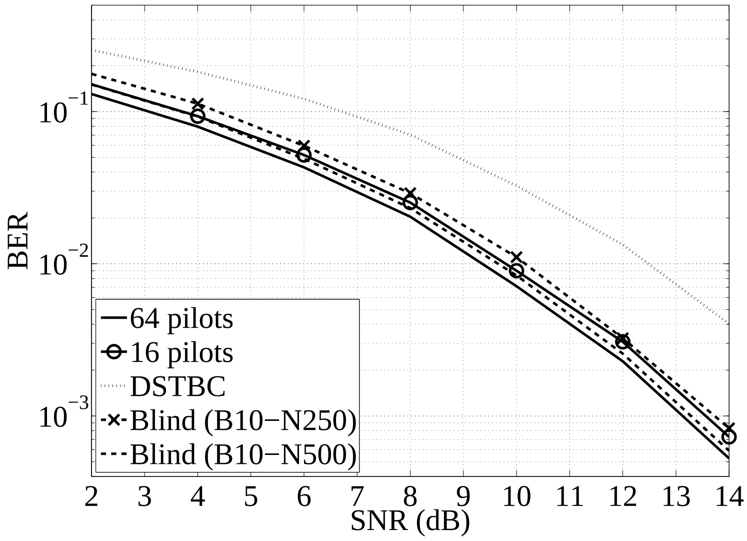

Figure 2.19: BER for the Rayleigh simulation during the MIMO 2 × 2 STBC experiments.

experiments and the simulations are consistent. Despite the similarities, there are also some differences between the experimental and the theoretical curves. For instance, the diversity gain estimated as the slope of the BER curve at high SNRs is not the same. This can be attributed to the fact that the actual fading channel is not exactly neither Rayleigh (for NLOS) nor Rice (for LOS). Furthermore, the simulations assume uncorrelated channels, but this is not true in practice. Finally, frequency and time synchronization errors also appear in the experimental curve. A final detail is that for the experimental curves it is plotted the BER Vs. the transmitted power, whereas for the simulation results it is plotted the BER Vs. the SNR. The reason for doing this is the fact that during the experimental setup is difficult to have a precise estimate of the SNR at the input of the receiver, since any SNR estimate would also include other impairments such us the channel estimation error, the timing jitter, etc.

Fig. 2.20 shows the results for the third experiment, which used a higher transmission rate R_s=5 Mbaud. These results are very similar to those obtained in experiment 2, as expected since there is not a clear line-of-sight. The results for the third experiment confirm the conclusions obtained for the first and second experiments.

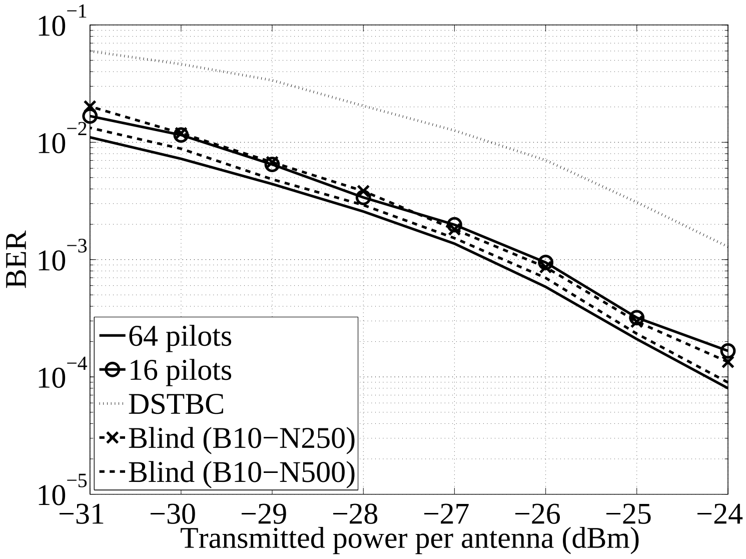

Figure 2.20: BER for the 5 Mbaud tests during the MIMO 2 × 2 STBC experiments.

2.8 Conclusions

In this Chapter it was described how to build a generic MIMO testbed that is able to perform the evaluation of new algorithms and processing techniques. Both its hardware and software have been described, emphasizing the novelty of the software used to control the hardware platform.

As an example of the results that can be obtained using the platform, performance curves of several MIMO-STBC systems were shown. Such curves evaluate the performance in real indoor scenarios where the testbed was working at 2.4 GHz. In particular, it was compared the Alamouti orthogonal scheme for two receive antennas with coherent and non-coherent demodulation. Two different channel estimation methods were considered: a conventional pilot-aided technique and a recently proposed blind algorithm based on second-order statistics. The results presented were obtained in three different scenarios. In all those cases the blind channel estimation technique provides similar BER performance than the pilot-aided method, with a slight increase in the effective data rate and a moderate increase in the computational complexity of the detector. On the other hand, the differential STBC presents, as expected, a 3 dB penalty with respect to coherent schemes.

3

Rapid prototyping for vehicular demonstrators

3.1 FlexVehd: a Flexible Vehicular Demonstrator

With the know-how obtained during the construction of the Multiple-Input Multiple-Output (MIMO) platform described in Chapter 2, it was possible to build a similar platform aimed at evaluating transceiver performance in vehicular scenarios. Such platform, called FlexVehd, has been built following the distributed multi-layer paradigm exposed in Chapter 2, enabling the user to operate it remotely and at a high abstraction level (i.e. MATLAB/Simulink).

FlexVehd is aimed specially at evaluating characteristics of the vehicular scenarios: the Doppler spread due to high mobility, the multipath fading due to the presence of infrastructure elements and other moving vehicles, the delay spread, the RF impairments due to nonlinear amplifiers, etc. All these channel impairments are particularly noticeable in the frequency band used by radio interfaces targeted for vehicular environments (5.2 - 5.9 GHz).

Moreover, such impairments can be artificially included to analyze their impact in system performance. Note that this is completely different from a conventional computer simulation since real analog transmission takes place and, thus, the actual distortion caused by radio hardware is included in the received signal. This is a very convenient manner of obtaining preliminary experimental results that are very difficult to obtain by field trials on the road with moving vehicles. In essence, this approach constitutes an intermediate stage between computer simulations and field trials.

This Chapter presents an example of such mode of operation consisting in the performance assessment of the IEEE 802.11p physical (PHY) layer under different vehicle speeds.

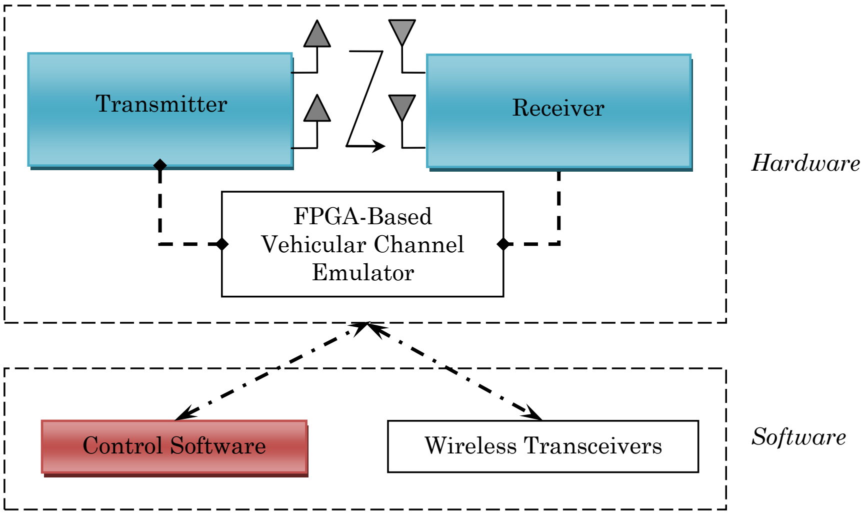

Figure 3.1: In color: components of the rapid prototyping system that are described in Chapter 3.

This Chapter is structured as follows. Sections 3.2 and 3.3 give a global description of the different system modules and a brief explanation of the synchronization, configuration and control mechanisms. The testbed hardware and software components are described in Sections 3.4 and 3.5. Section 3.6 exposes the results obtained when evaluating the testbed in artificial high-speed scenarios. Finally, Section 3.7 is devoted to the conclusions.

Like in Chapter 2, this Chapter will deal mainly with the two issues highlighted in Fig. 3.1: the hardware testbed and the control software.

3.2 FlexVehd Architecture

Fig. 3.2 shows the FlexVehd hardware hosted by two ordinary PCs, one for the transmitter (referred to as Tx PC) and another one for the receiver (called Rx PC). Fig. 3.3 illustrates the block diagram of the entire FlexVehd system. Three main parts can be distinguished: the testbed hardware that allows transmitting analog signals at an intermediate frequency (IF) of 10 MHz; the multilayer software architecture that provides convenient access to the hardware (which was described in Section 2.6) and; finally, the user application, i.e. the IEEE 802.11p PHY layer transmission and receiving chains (detailed in Chapter 4), implemented using the testbed capabilities and the architecture facilities.

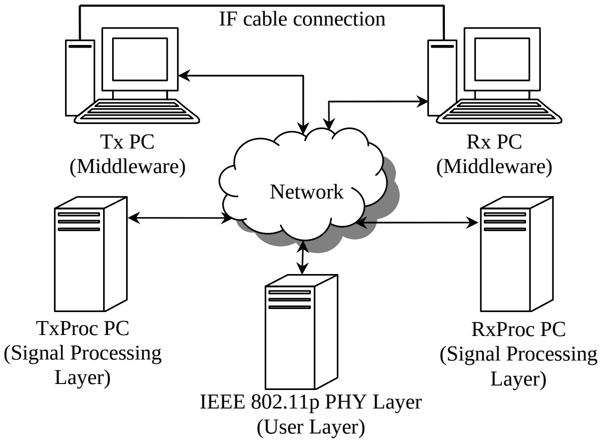

Figure 3.2: FlexVehd platform diagram.

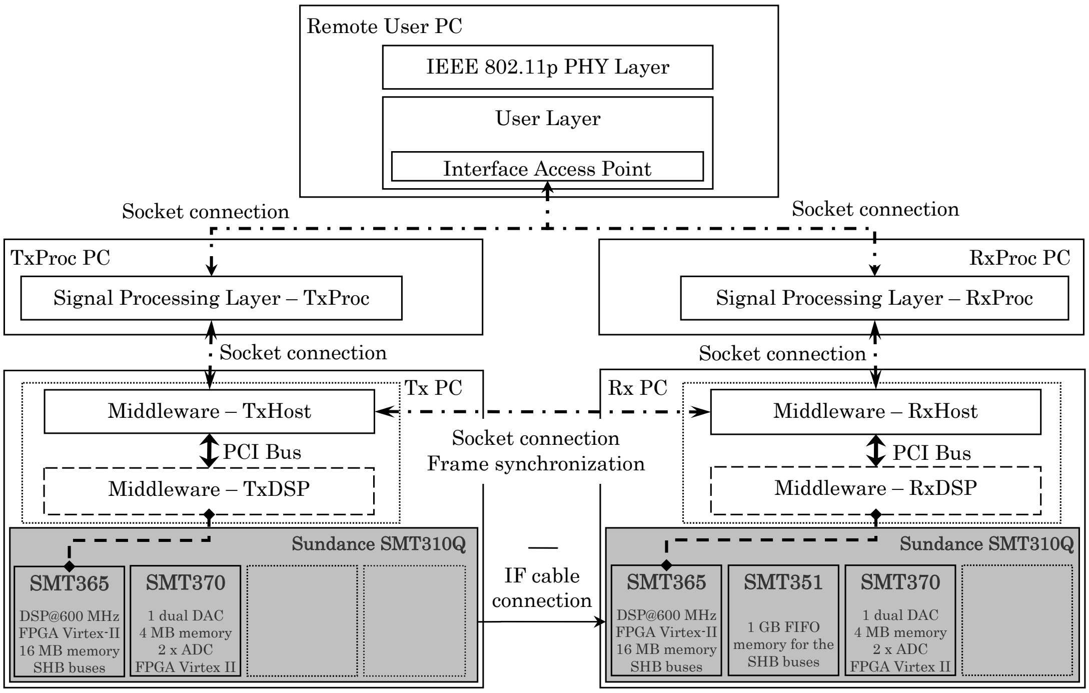

Figure 3.3: FlexVehd architecture scheme.

The lowest software level (i.e. the middleware) has to be installed in the same computers as the testbed hardware, but the other two layers can be installed in any other available PCs. In order to minimize the time the Tx-Rx PCs are busy, the rest of

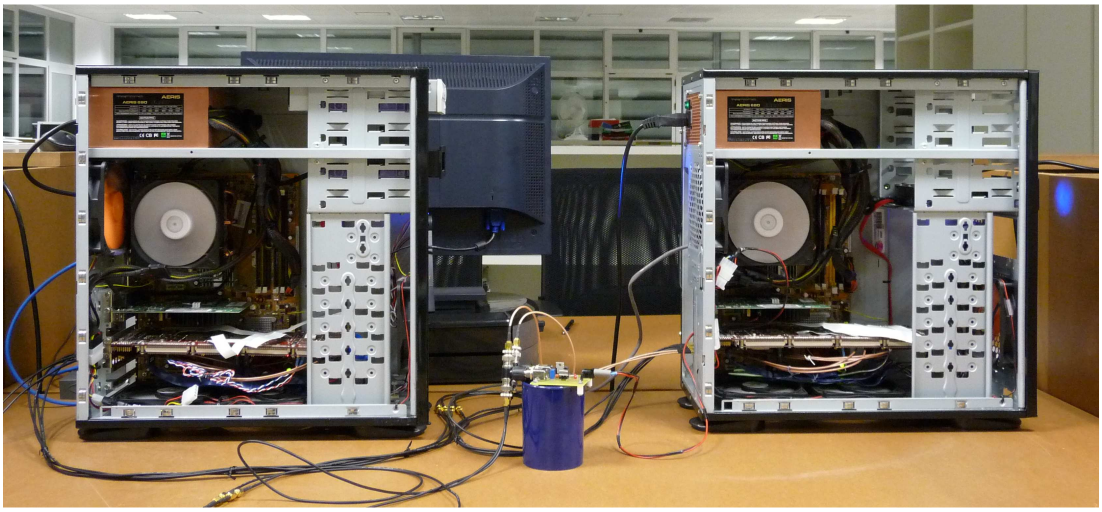

Figure 3.4: Picture of FlexVehd showing the Tx and Rx PCs.

the architecture layers are detached from them, which allows running different signal processing layer instances simultaneously in different PCs. Finally, the user layer is installed in the same computer as the IEEE 802.11p PHY transceiver.

An example of how the data frames are transmitted is exposed in Section 2.6.1.

3.3 Middleware Layer

The middleware layer fills the gap between the platform hardware and the PHY layer, allowing discrete-time signals to be transferred through the PCI bus and making possible the synchronization between the Tx PC and the Rx PC using a TCP/IP network connection. A detailed description of the characteristics and functioning of the middleware is given in Section 2.6.

3.4 FlexVehd Hardware Description

Fig. 3.4 shows a picture of the Tx and Rx PCs that constitute FlexVehd. The hardware of the testbed is very similar to the one used by the MIMO 2 × 2 platform, which is detailed in Section 2.2. It is based on a PCI carrier board (SMT310Q) and a basic processing module (SMT365) that is equipped with a Xilinx Virtex-II FPGA (Field-Programmable Gate Array) and a Texas Instruments C6416 digital signal processor (DSP) at 600 MHz. The processing module has two buses that can transfer 32-bit

Table 3.1: IEEE 802.11p PHY key parameters

Parameter	Value(s)
Data rate (Mb/s)	3, 4.5, 6, 9, 12, 18, 24 and 27
Modulation	BPSK, QPSK, 16-QAM, 64-QAM
Code rate	1/2, 1/3, 3/4
# subcarriers	48 data + 4 pilots
OFDM symbol duration	$8\mu s$
Guard time	$1.6\mu s$
FFT period	$6.4\mu s$
Preamble duration	$32\mu s$
Subcarrier spacing	0.15625 MHz

words up to 400 MB/s, allowing the connection with the SMT370 module, that contains a dual AD9777 D/A converter and two AD6645 A/D converters. The SMT370 module also has a 2 MB per-channel memory that is used to load the frames to be transmitted. Note that in contrast to the MIMO platform of Chapter 2, there are no 1 GB First-In First-Out (FIFO) memory modules in the transmitter since the 2 MB of the SMT370 are enough for the experiments to be performed.

At the receiver side, the data acquired by the analog-to-digital conversors (ADCs) (placed in a SMT370 module) are stored in real time in a 1 GB FIFO memory module (SMT351). The SMT365 module is then responsible for sending the data to the middleware through the PCI bus.

3.5 IEEE 802.11p PHY Layer Implementation

FlexVehd implements the PHY layer of the IEEE 802.11p standard (17), which is based on the 802.11a standard. IEEE 802.11p includes some specific modifications for enhanced performance in high mobility scenarios. The key parameters of the IEEE 802.11p PHY layer are summarized in Table 3.1.

The IEEE 802.11p PHY layer transmitter and receiver are implemented as Simulink models because it facilitates system design, it is easy to integrate with the MATLAB entry points of the multilayer architecture, and it constitutes a very good framework for migration to hardware (e.g. to an FPGA) of those subsystems of interest (for example, by using the tool Xilinx System Generator for DSP). A detailed description on the design of the IEEE 802.11p transceiver can be found in 4.2.2.1.

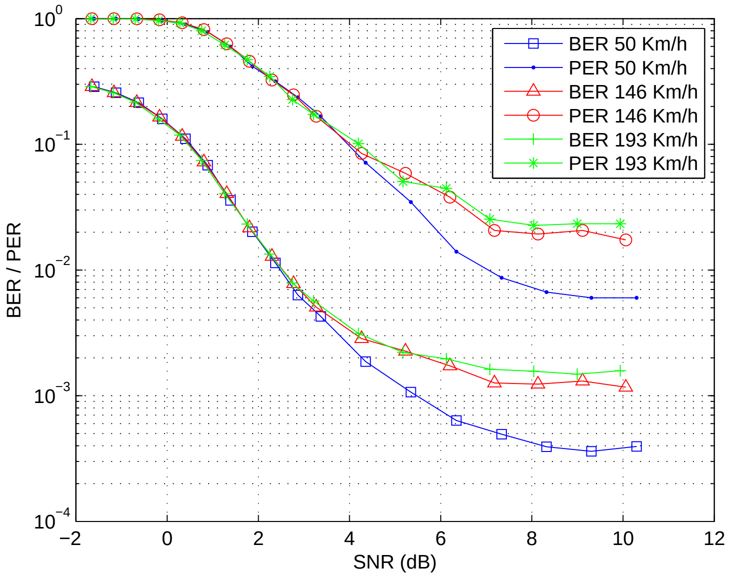

Figure 3.5: BER and PER Vs SNR of FlexVehd for different vehicle speeds. BPSK, code rate $R_c = 3/4$, 64-byte packets.

3.6 Experimental Performance Results of IEEE 802.11p PHY Under High Mobility Conditions

The first results obtained when using FlexVehd were aimed at capturing the impact of Doppler spread due to mobility taking the impairments of IF hardware into account. Note that in Chapters 5 and 6 more sophisticated channels are described, but for testing purposes, only an artificial Doppler effect was added to the channel before performing the digital-to-analog conversion. Additive white Gaussian noise (AWGN) was also added but at reception, after analog-to-digital conversion. Notice that apart from the artificial impairments included, the hardware adds a frequency offset (due to the mismatch between the clocks of the transmitter and the receiver), phase rotation and AWGN with constant power.

Fig. 3.5 shows the Bit-Error Rate (BER) and Packet-Error-Rate (PER) obtained for 64-byte (512 bits) packets when recreating vehicular speeds of 50, 146 and 193 Km/h (31, 90 and 120 mph respectively). Rayleigh fading is assumed. BPSK and a code rate $R_c = 3/4$ are used. Each point has been obtained by averaging over 500 transmitted packets. To be congruent with the ITS band for vehicular communications, a carrier

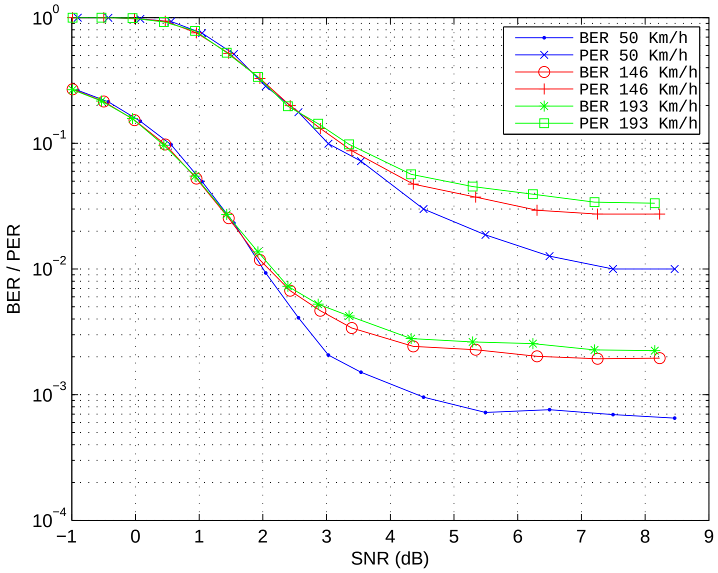

Figure 3.6: BER and PER Vs SNR of FlexVehd for different vehicle speeds. 4-QAM, code rate $R_c = 1/2$, 64-byte packets.

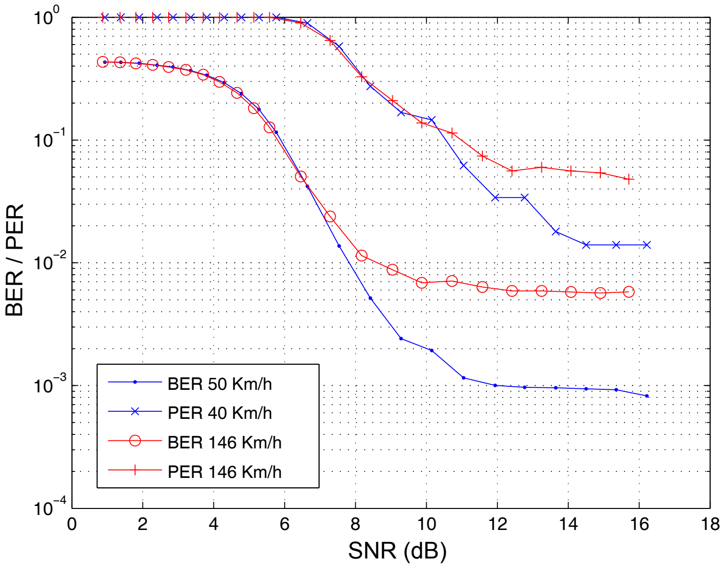

Figure 3.7: BER and PER Vs SNR of FlexVehd for different vehicle speeds. 16-QAM, rate $R_c = 1/2$, 64-byte packets.

frequency of 5.875 GHz is assumed when calculating the maximum Doppler shift related to the vehicle speed. Note that vehicular communications are likely to be performed in the 5 GHz band since both US and European authorities have reserved spectrum for Intelligent Transport Systems (ITS) at 5.9 GHz.

For the packet size indicated, it can be observed that the receiver implementation fulfills the requirements of (54) (i.e. PER is less than 10% for a vehicle speed of 193 Km/h). Fig. 3.6 shows the corresponding results when the modulation is 4-QAM and the code rate is $R_c = 1/2$. In this case, the receiver implementation again fulfills the IEEE 802.11p standard requirements.

Higher spectral efficiency can be obtained by switching to 16-QAM. Fig. 3.7 shows the corresponding performance results for 16-QAM and code rate $R_c = 1/2$. This combination results in two transmitted information bits per channel use. Once more, it can be seen that the receiver implementation suffices for being compliant with the requirements of the standard.

3.7 Conclusions

This Chapter presented FlexVehd, a radio testbed for vehicular environments that provides the maximum flexibility to researchers to test their algorithms. The platform makes use software transceivers that implement the PHY layer of the IEEE 802.11p standard. Several basic experimental results have been shown in order to demonstrate the potential of the testbed, comparing the performance of the transceiver when it worked under different high-speed Rayleigh scenarios.

4

Rapid prototyping for vehicular wireless transceivers

4.1 Introduction

Vehicular applications can be divided into two groups: safety and non-safety applications. Safety applications often demand fast message exchanges and do not use much bandwidth. Examples of such applications include collision avoidance and hard braking warnings, accident reporting or intersection announcements. The IEEE 802.11p standard is probably the best positioned to provide safety services, although other wireless standards, such as IEEE 802.16e or IEEE 802.20, have been considered (55, 56, 57). On the other hand, non-safety applications do not have such tight time constraints and may need higher data transfer rates. Non-safety applications include mobile internet access, automatic toll collection, traffic light status notification, speed limit and cruise control, roadsign recognition, travel information management, etc.

For non-safety vehicular applications, the discussion about which is the most suitable wireless access standard remains an open issue. The most cited candidates are the WiFi standards IEEE 802.11a/b/g and IEEE 802.16e (Mobile WiMAX). It is important to note that vehicular communications are likely to be performed in the 5 GHz band since both US and European authorities have reserved spectrum for Intelligent Transpor Systems (ITS) at 5.9 GHz. Due to this, IEEE 802.11b/g are not the preferred choices and the final candidates might be reduced to IEEE 802.11p, IEEE 802.11a and IEEE 802.16e. This Chapter shows how to create MATLAB/Simulink transceivers for the physical (PHY) layer of such three standards in order to use them in conjunction with the testbeds described in Chapters 2 and 3, and the vehicular channel emulators detailed in Chapters 5 and 6.

In particular, in Chapter 5 it is described a hardware evaluation system for measuring the performance of the IEEE 802.11p PHY-layer in Single-Input Single-Output (SISO) vehicular scenarios. Other authors have carried out similar studies for the IEEE 802.11a PHY-layer (58, 59). Moreover, although several papers have suggested Mobile WiMAX as an alternative to IEEE 802.11p for non-safety applications (55, 57), as of writing, there is no analysis of the performance of Mobile WiMAX when transmitting over the channels that are arguably the best references for evaluating vehicular communications at the 5 GHz band (60).

Other wireless communication standards have been proposed to be used in vehicular environments, such as HSDPA (High-Speed Downlink Packet Access) (61), IEEE 802.20 (iBurst) (62) or EDGE Evolution (63), but the peak data rates they offer for broadband communications (14.4 Mbit/s, 16 Mbit/s and 1 Mbit/s, respectively (61, 64, 65)) are lower than those theoretically provided by IEEE 802.11p, IEEE 802.11a or IEEE 802.16e (27 Mbit/s, 54 Mbit/s and 39.9 Mbit/s, respectively (54, 66, 67)). Also, there are several recently developed standards whose performance in vehicular scenarios has yet to be assessed and whose study will constitute an interesting topic for further research. Such new standards seem to offer better global performance than IEEE 802.16e, but they are either in earlier development stages (e.g. LTE (Long-Term Evolution) (68), IEEE 802.16m (69)) or have not been explicitly designed for vehicular applications (e.g. IEEE 802.11n (70)).

Note also that if IEEE 802.11p has to provide non-safety services in vehicular environments, its data transfer rates will have to be somehow improved in order to compete with standards like IEEE 802.16e or LTE. One of the best ways to increase the transmission capacity consists in using multiple antennas at transmission (known as MISO (Multiple-Input Single-Output) systems), reception (SIMO (Single-Input Multiple-Output) systems) or both at transmission and reception (MIMO systems) (35, 36). IEEE 802.11p was initially devised as a single-antenna (SISO) system, so it is of great interest from a communication system designer point of view to evaluate in realistic scenarios transceivers based on IEEE 802.11p but including multiple antennas.

It is important to remark that IEEE 802.11p is an amendment to the IEEE 802.11 standard that specifies the extensions for wireless local area networks in order to provide wireless communications in a vehicular environment. Such amendment is based on IEEE 802.11-2007 as amended by IEEE 802.11k-2008, IEEE 802.11r-2008, IEEE 802.11y-2008, IEEE 802.11n-2009 and IEEE 802.11w-2009. Hence, being IEEE 802.11n one of the amendments, the idea of applying multiple-antenna techniques in IEEE 802.11p could be easily carried out in future releases of the standard through a new amendment. In this way, besides describing SISO implementations of IEEE 802.11p, IEEE 802.11a and IEEE 802.16e transceivers, this Chapter also shows how to implement a MIMO IEEE 802.11p software transceiver, which will be later evaluated in Chapter 6.

The rest of this Chapter is structured as follows. Section 4.2 describes how the IEEE 802.11p (SISO and MIMO) and IEEE 802.11a transceivers evaluated in Chapters 5 and 6 have been built, giving a brief summary on IEEE 802.11p evaluations performed by other authors. Section 4.3 summarizes some of the latest Mobile WiMAX performance measurements and details the Mobile WiMAX transceiver used for the experiments exposed in Chapter 5. Finally, Section 4.4 is devoted to the conclusions.

This structure allows concluding that the present Chapter will deal with one important aspect of any rapid prototyping system: the software transceivers to be evaluated (highlighted in Fig. 4.1).

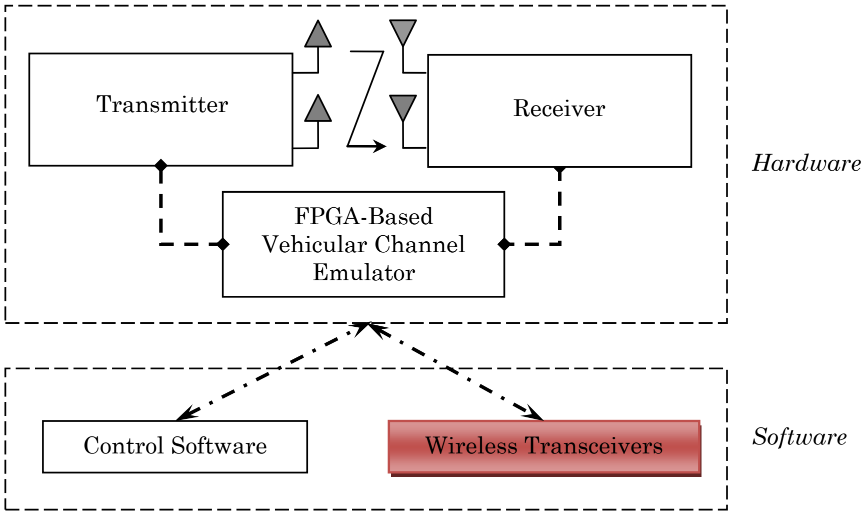

Figure 4.1: In color: components of the rapid prototyping system that are described in Chapter 4.

4.2 IEEE 802.11p and IEEE 802.11a

The IEEE 802.11p standard (17) is an amendment to IEEE 802.11-2007 (71) and is technically compatible with the specifications given by ASTM E2213-03 (54), which address the challenges that arise when providing wireless access in vehicular environments. The standard has its origin in 1999, when the Dedicated Short Range Communications (DSRC) spectrum band, a band of 75 MHz at 5.9 GHz, was allocated in the United States. At that moment, a need to define a common framework to develop hardware and applications for the DSRC band emerged. A deep description of the standard is beyond the scope of this Thesis, but the interested reader may take a look at the excellent overviews given in (72), (73) and (74).

The IEEE 802.11p Medium Access Control (MAC) and PHY layers are very similar to those used in the Wireless Local Area Network (WLAN) standard IEEE 802.11a (66), but they have lower overheads to allow faster exchanges of safety messages.

At the PHY-layer level the main difference between IEEE 802.11a and IEEE 802.11p is that the 20 MHz bandwidth used in IEEE 802.11a is reduced to 10 MHz in IEEE 802.11p. Although the mentioned bandwidth reduction results in a loss of data transfer rate, it provides an important advantage when overcoming the effects of vehicular channels: the OFDM (Orthogonal Frequency-Division Multiplexing) symbols are longer in the time domain and the system can deal with larger delay spreads, thus being able to avoid ISI (Inter-Symbol Interference). Therefore, if we ignore the IEEE 802.11p ACR (Adjacent Channel Rejection) and the SEM (Spectrum Emission Mask) requirements, the practical implementation of a basic IEEE 802.11p transceiver is straightforward: it suffices to double all the OFDM timing parameters used by IEEE 802.11a devices.

4.2.1 Background: IEEE 802.11p Evaluations

Due to the novelty of the IEEE 802.11p MAC layer, most of the evaluations of the standard in the literature are aimed at studying it by using simulators like `ns-2` (75) or following analytical approaches (76). There are also software implementations of the standard: some of them describe generic transceivers (77), but the majority are focused on the performance of the receiver by using techniques for getting better channel estimations (78, 79, 80, 81) or improved decisions (82).

Before the development of the IEEE 802.11p amendment there were several proposals to support DSRC applications. Most of the already available prototypes were designed for the particular regulations of countries like Japan (83) or Korea (84), or for the European standard (85). As the draft of the IEEE 802.11p has been evolving, more prototypes have arisen, such as (86), that the authors declare as the first platform where IEEE 802.11p and IEEE 1609 (that defines the upper MAC layer and the network pulse session layer) have been implemented.

DSRC-based processing modules have been developed as well. For instance, in (87) a fast OFDM modulator/demodulator allows performing an IFFT/FFT in less time than the symbol interval (8 μs) and in (88) a software-defined FPGA-based channel simulator was designed for testing baseband transceivers of different wireless communication systems, showing an example of its use with an IEEE 802.11p system.

4.2.2 IEEE 802.11p/a Transceivers

Fig. 4.2 shows the basic setup of a IEEE 802.11p evaluation system that makes use of the channel emulator described in Chapter 5. The key parameters of the IEEE 802.11p/a transceivers are exposed in Table 4.1. In a nutshell, it can be said that the evaluation system consists of a transmitter that generates packets that are sent to the receiver, passing through the vehicular channel emulator.

Table 4.1: Feature comparison of the IEEE 802.11p, IEEE 802.11a and IEEE 802.16e standards.

Parameter	**802.11p**	**802.11a**	**802.16e**
Carrier Modulation	BPSK, QPSK, 16-QAM, 64-QAM		
Code rate	1/2, 2/3, 3/4		
# subcarriers	48 data + 4 pilots		360 data + 60 pilots
OFDM symbol duration	8 μs	4 μs	64 μs
Guard time	1.6 μs	0.8 μs	12.8 μs
FFT period	6.4 μs	3.2 μs	51.2 μs
Total bandwidth	10 MHz	20 MHz	10 MHz
Subcarrier spacing	156.25 KHz	312.5 KHz	19.53 KHz

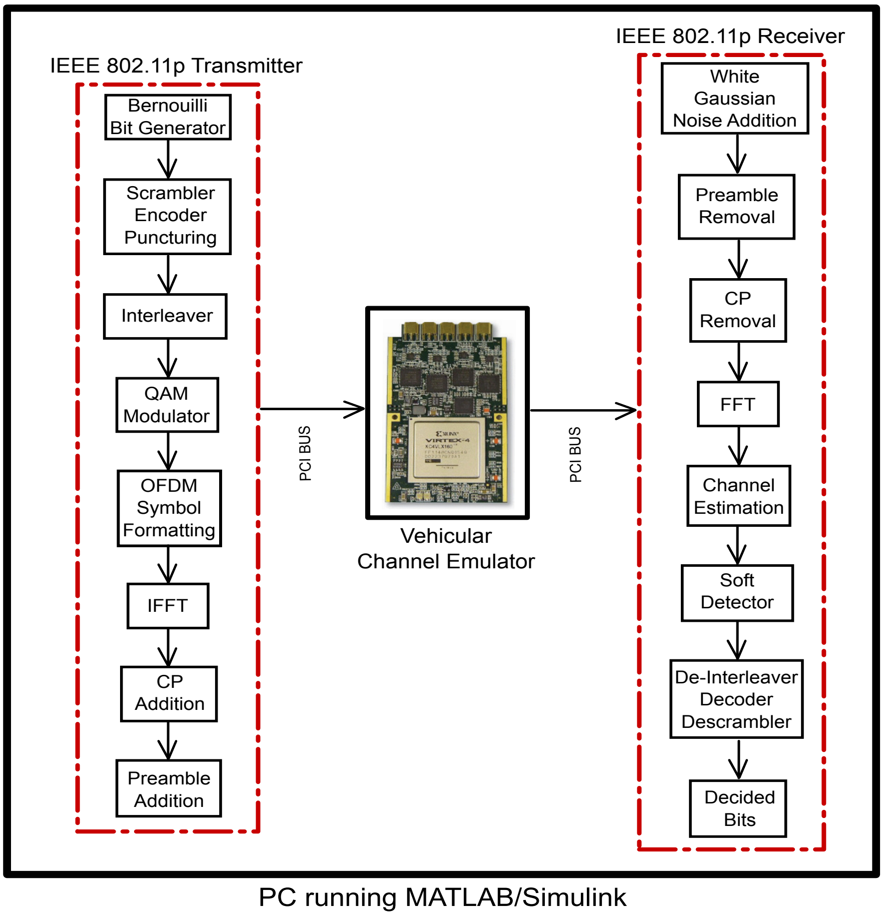

Figure 4.2: IEEE 802.11p transceiver connected to the SISO vehicular channel emulator.

4.2.2.1 SISO Transceiver

Single-Antenna Transmitter The blocks of the IEEE 802.11p transmitter are depicted at the top of Fig. 4.2. The scheme is very similar to any OFDM transmitter with coding.

First, equally likely data bits are generated using a Bernoulli binary generator. In each cycle of the model, enough bits are generated to fill a frame with the number of desired OFDM symbols to transmit. Then, the generated bits are scrambled using a 127-bit sequence obtained from the generator polynomial $S(x) = x^7 + x^4 + 1$. A different pseudo random sequence is used for each transmission.

The scrambled bits are then processed by a rate 1/2 convolutional encoder with the industry-standard generator polynomials $g_0 = 133$ and $g_1 = 177$. In case of needing higher rates (2/3 and 3/4 are supported), puncturing is employed. Afterwards, data interleaving is carried out in a two-step permutation. The first permutation ensures that adjacent coded bits are mapped onto non-adjacent subcarriers, whilst the second permutation ensures that adjacent coded bits are mapped onto less and more significant bits of the constellation to avoid long runs of low reliability.

After interleaving, the bits are Gray-mapped into QAM symbols and placed into subcarriers. Fig. 4.3 shows how the assignment of positions in each OFDM symbol is performed. Of the total 64 subcarriers, four are dedicated to pilot signals. These pilots are always situated in the same subcarriers (in positions -21, -7, 7 and 21, since subcarriers are numbered between -32 and 31). Moreover, the pilots are BPSK modulated by a pseudo binary sequence. Forty-eight of the rest of the subcarriers are used for placing the data symbols. The subcarrier 0 is reserved for the DC and the remaining subcarriers act as frequency guards. Finally, the IFFT is performed.

For each OFDM symbol, a 1/4 cyclic prefix (CP) is added to prevent ISI and, eventually, a preamble is appended to the whole frame. In our experiments, although the preamble is always transmitted in order to respect the requirements of the standard, it is not used in reception since perfect time synchronization is assumed and there are no algorithms that may make use of it. Furthermore, the vehicular channel emulator operates in baseband, so there is no IF stage, neither in the transmitter nor in the receiver.

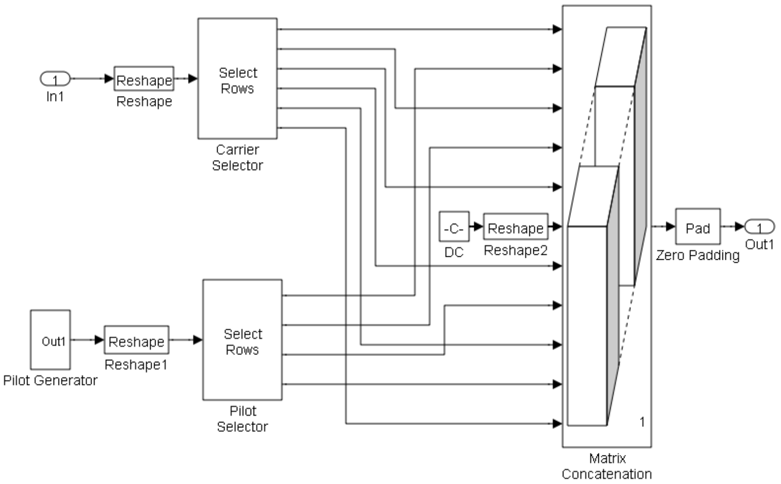

Figure 4.3: Placement of the DC, data and pilots into the IEEE 802.11p subcarriers.

Single-Antenna Receiver The receiver blocks are shown at the bottom of Fig. 4.2. The first step performed is the addition of white Gaussian noise in order to obtain BER (Bit Error Rate) and PER (Packet Error Rate) curves versus E_b/N_0 values.

After the preamble and the CP are removed, the FFT is applied to each OFDM symbol. Next, the channel has to be estimated, which is performed by extracting the four pilots and dividing them by their respective transmitted values (which are known by the receiver), obtaining the estimated channel coefficients for the pilot subcarriers. These estimates are noisy and the implemented channel inversion method can even amplify the noise, but they were used for simplicity.

Then, the four channel coefficient estimates are linearly interpolated to obtain the channel frequency response for the rest of the subcarriers. After this, an MMSE (Minimum Mean Square Error) equalizer (89) is employed.

Then, the equalized symbols are sent to a soft detector. The output LLRs are deinterleaved, inverting the permutations performed in the transmitter, and the Viterbi block decodes the data.

At the end, the decoded bits are descrambled, using the same scrambler as at the transmitter side and the final bits are obtained.

4.2.2.2 MIMO Transceiver

Fig. 4.4 shows the MIMO 4 × 4 evaluation system designed for connecting to one of the vehicular channel emulators described in Chapter 6. Each of the elements labeled as IEEE 802.11p transmitter/receiver is a modified version of the previously described IEEE 802.11p SISO transceiver but with certain differences that are exposed below.

Fig. 4.5 depicts the other MIMO evaluation system whose performance is evaluated in Chapter 6. In this case, it is an IEEE 802.11p system with two transmit and two receive antennas that makes use of Alamouti coding/decoding stages.

Multiple-Antenna Transmitter The multi-antenna IEEE 802.11p transmitter is very similar to the one described in the SISO subsection, but several changes had to be made in order to be able to make use of multi-antenna techniques.

The first element to change is the channel estimator, which is chosen to use orthogonal pilots that constitute matrices called OSTPM (Orthogonal Space-Time Pilot Matrices). Specifically, Hadamard matrices were used and they were created using

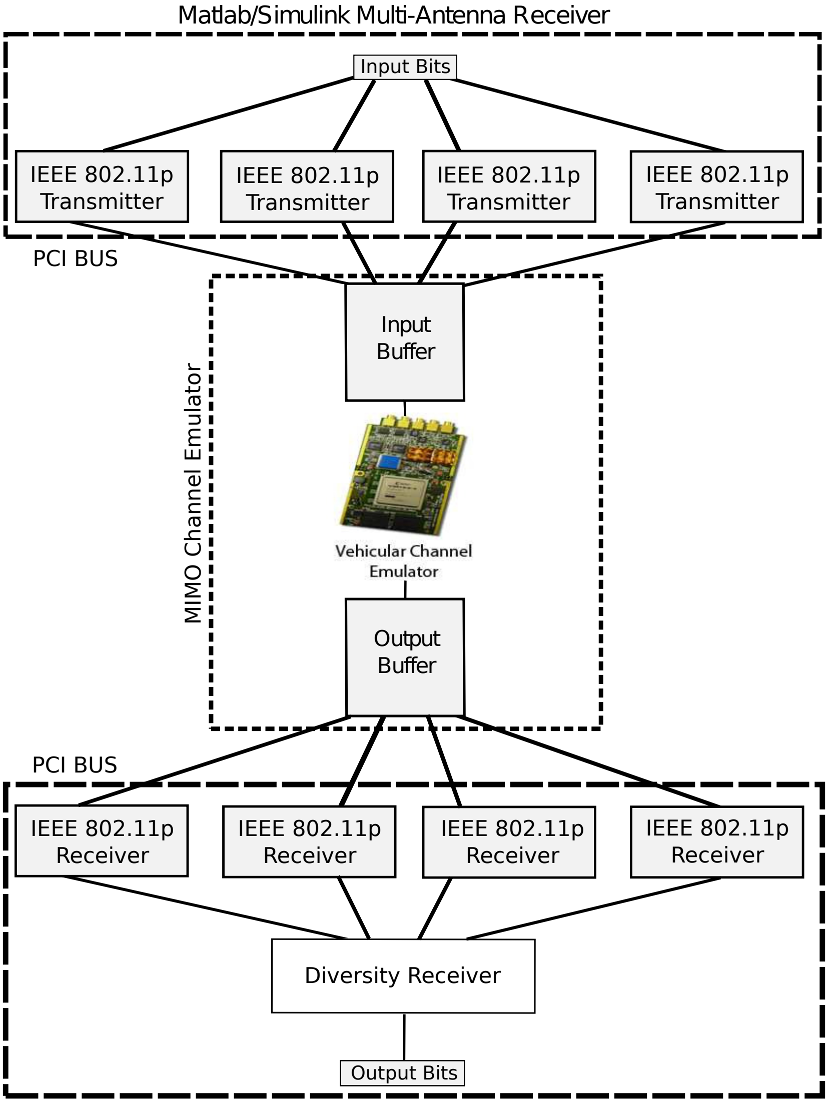

Figure 4.4: Time-multiplexed evaluation system for multiple-antenna IEEE 802.11p transceivers.

Sylvester's method, which generates a sequence of matrices that are known as Walsh matrices. Such matrices are orthogonal in space and time and, in the case of transmitting with two antennas, are generated according to (4.1),

$$\mathbf{P} = \begin{pmatrix} p_k & p_k \\ p_k & -p_k \end{pmatrix} \tag{4.1}$$

where p_k is the BPSK-modulated pilot symbol transmitted at the k-th subcarrier. Since IEEE 802.11p uses four pilots inside each OFDM symbol, the pilot matrix is generated

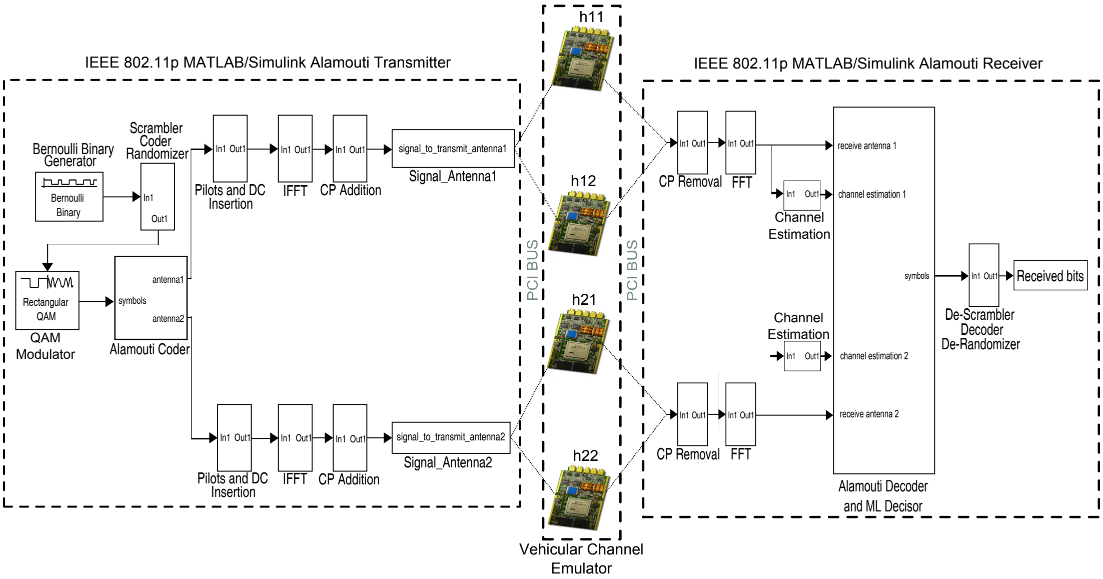
IEEE 802.11p MATLAB/Simulink Alamouti Transmitter
Bernoulli Binary Generator
Bernoulli Binary
Scrambler Coder Randomizer
Rectangular QAM
QAM Modulator
Alamouti Coder
Pilots and DC Insertion
IFFT
CP Addition
Signal_Antenna1
Signal_Antenna2
h11
h12
h21
h22
PCI BUS
Vehicular Channel Emulator
IEEE 802.11p MATLAB/Simulink Alamouti Receiver
CP Removal
FFT
Channel Estimation
Alamouti Decoder and ML Decisor
De-Scrambler Decoder De-Randomizer
Received bits

Figure 4.5: Time-multiplexed MIMO 2 × 2 emulator model without buffering.

by replicating (4.1) to obtain a 2 × 4 matrix.

Using this scheme, channel estimation only requires simple linear processing. For instance, in the case of transmitting with two antennas, the signal received at the k-th subcarrier for two consecutive OFDM symbols is:

$$\begin{aligned} y_{1,k} &= p_k h_1 + p_k h_2 + n_1 \\ y_{2,k} &= p_k h_1 - p_k h_2 + n_2 \end{aligned} \tag{4.2}$$

where n_1 and n_2 are AWGN samples and h_1 and h_2 are the channel coefficients. Then, the channel coefficient estimation can be obtained as:

$$\begin{aligned} \hat{h}_{1,k} &= \tfrac{y_{1,k}+y_{2,k}}{2p_k} \\ \hat{h}_{2,k} &= \tfrac{y_{1,k}-y_{2,k}}{2p_k} \end{aligned} \tag{4.3}$$

Note that this channel estimation method has several limitations. First, it assumes that the channel remains constant over two consecutive pilots: if there is a sufficiently high Doppler spread, performance will be degraded. The second drawback is related to the generation of the pilot matrix: it is only possible to use this pilot scheme when the number of transmit antennas is a power of two. Moreover, in order to make use of the channel estimation method, an even number of OFDM symbols has to be transmitted.

In spite of the above-mentioned issues, this method was used due to its simplicity and because the maximum Doppler shift of the MIMO channels described in Chapter

6 is 1742 Hz. This means that it can be assumed that the channel does not vary significantly within a time period of 574 μs, which is higher than the time required to transmit two consecutive OFDM symbols (16 μs).

Apart from the modifications required by the channel estimation, to exploit space-time diversity, MIMO systems need an additional coding stage. In the case of a 2 × 2 system Alamouti coding (41) is used, whereas the 4×4 transceiver implements a quasi-orthogonal code suggested by Jafarkhani (90).

Multiple-Antenna Receiver At the receiver side the main changes with respect to the SISO system are related to the support of diversity schemes. In the SIMO case the MRC (Maximum-Ratio Combining) technique was implemented, whilst the MIMO transceiver includes an Alamouti decoder (for the 2 × 2 system) and a ML detector. No more details will be given about the receivers since they use standard algorithms and techniques which are extensively covered by the existing literature (e.g. (91)).

4.3 Mobile WiMAX for Vehicular Scenarios

4.3.1 Background

In recent years the popularization and massive distribution of multimedia content has stimulated the development of broadband communication technologies with increasing data rates. Nowadays, wired-access technologies such as cable or DSL (Digital Subscriber Line) supply enough data rate for most applications. However, several standards still compete to become the one chosen by the main wireless carriers to provide the next-generation of mobile broadband services.

Vehicular scenarios are one of the most challenging scenarios for broadband wireless communications since radio interfaces face the challenge of transmitting over highly time-variant channels that often exhibit high delay spreads. Practical evaluation of vehicular communication systems can be directly performed by loading a data communications unit on a vehicle and driving through different environments. This procedure, however, is extremely expensive and, moreover, experiments can be affected by unintended side effects that may be uncontrollable.

In spite of its difficulties, the performance of Mobile WiMAX transceivers over real-world experiments has been reported in (92) and (93). The work in (92) used a MIMO

2×2 Mobile WiMAX testbed made up with Alcatel-Lucent equipment and measurements were done in the 3.5 GHz frequency band. The experiments were obtained when driving through an area with tall trees and buildings, and involved different types of traffic. A base station antenna was on the roof of the highest building in the area and a mobile station was placed inside a car that drove at a maximum speed of 40 Km/h. The experimental results show that MIMO techniques improve reliability since they reduce the number of packet losses. Furthermore, the authors compared the performances of Mobile WiMAX and HSPA (High-Speed Packet Access) concluding that in the case of light files (256 bytes) Mobile WiMAX outperforms HSPA. On the other hand, for larger files (1 MB), HSPA obtains better average throughput thanks to the use of HARQ (Hybrid Automatic Repeat Request) protocols, which significantly decrease the number of retransmissions. It must be noted that for a fair comparison both Mobile WiMAX and HSPA should have made use of HARQ, but at the time of writing the authors had not implemented HARQ for their Mobile WiMAX testbed.

Another example of performance measurements over real channels is (93). In this case, several tests were conducted using what the authors claim to be the first commercial WiBro network, which worked at 2.3 GHz and used 10 MHz of channel bandwidth. In one of the experiments WiBro performance was measured when a user traveled around Seoul in a subway and in a bus measuring the UDP/TCP goodput and the RTT (Round Trip Time), concluding that:

- The goodput in both scenarios was poor in shadowed areas (i.e. where there was a poor network coverage).

- The goodput in the subway scenario was good near the subway stations, but poor between stations. This happened because the repeaters were installed in each station.

- Although the goodput was reduced when transmitting in shadowed areas, the connectivity remained during the whole measurement time (in the worst case, the goodput was roughly 200 Kbps).

- In the bus tests, in about 80% of the measurement time the goodput was higher than 1 Mbps.

- The average RTT in both scenarios was almost the same, but the maximum RTT was larger in the case of the subway (in almost 100 ms).

- Both in bus and subway an average goodput between 1.7 and 1.9 Mbps was obtained, which lead the authors to state that the WiBro service was well provided.

Moreover, the authors compared the results obtained with WiBro with those obtained using HSDPA. In this case, one base station was deployed in a static ubication, while the user transmitted from five different spots:

- Outdoor Center. The user transmitted from the top of a building situated 200 meters away from the base station, existing a clear LOS (Line-of-Sight) path.

- Indoor Center. The user was ubicated in the fourth floor of the same building used for the Outdoor Center tests. There was no indoor repeaters.

- LOS Repeater. The user transmitted from the third floor of a building located far from the base station. Although the user received no LOS components from the base station, the channel environment was good, since the user actually connected to an indoor LOS repeater that was deployed in the vicinity of the base station.

- Cell Edge. This spot was located about 1 kilometer away from the base station. The channel quality was not good in spite of being an outdoor environment.

- Mobile Environment. The user moved with an average speed of 30 Km/h along a circular road that surrounded the campus (it was assumed that handoffs did not occur).

For these scenarios the paper concludes that:

- WiBro provides higher downlink goodput than HSDPA and a good uplink performance.

- Both WiBro and HSDPA obtain an average RTT that in most of the scenarios was far larger than those obtained by commercial WLANs (more than 100 ms, versus 10 ms for WLANs).

- WiBro-to-WiBro communications suffer from very large delays (an average of almost 300 ms), even between users located in the same cell. This is an important restriction for real-time services like VoIP (Voice over IP).
- When TCP was used as transport protocol, the large RTTs limited WiBro's maximum obtainable goodput.

Despite the interesting empirical results obtained by the papers previously mentioned, wireless transceivers can be more cost-effectively evaluated inside a laboratory, performing transmissions over either theoretical channels (i.e. channels that follow a statistical model on which the scientific community has reached a consensus) or empirical channels collected during measurement campaigns. The advantage of this approach is that channels can be used repeatedly and fair comparisons between different transmission methods can be carried out.

Mobile WiMAX transceivers have been evaluated over theoretical vehicular channels in (94) and (95). In (94), the way in which different types of terrains and vehicle movement patterns affect the propagation characteristics of 3.5 GHz Mobile WiMAX equipment is analyzed from a theoretical perspective. The authors carried out such an investigation because they consider Mobile WiMAX to be suitable for networked vehicular applications due to its inherent wide coverage, which minimizes the rate of handover and, thus, the data losses due to disrupted communications. It is also interesting to compare the effects of different terrains on a WiMAX-based vehicular network, since the terrain over which the network is deployed influences the propagation of the signal.

To perform the measurements, QualNet is used as network simulator and NCTUns as vehicular simulator. Three types of terrain were considered: flat (the terrain had no obstacles that might interfere with the transmitted signal), urban (with dense and high-rise buildings) and rural (with sparse and low-rise buildings). Furthermore, a speed of up to 50 Km/h was considered in five different speed/acceleration profiles for rural and urban environments. As a result, the authors were able to study average throughputs and packet latencies, concluding that the reduction of LOS due to the effects related to each terrain seemed to affect the network performance more than vehicular mobility.

Simulations over a theoretical channel are also performed in (95), where a FEC (Forward Error Correction) scheme for a live Mobile WiMAX-based video surveillance system is presented. The authors consider that the existing wireless technologies, including WiMAX, fail to deliver high data rates at high vehicular speeds, especially due to multipath fading. To address the dynamic nature of the mobile communications environment, they propose to use a FEC scheme whose block size is adapted to suit the current channel conditions. Their system consisted of a public train, which acted as a mobile node, and several base stations that were placed on train platforms. The simulations were conducted using `ns-2` and the implemented channel took into account that the train began to move after a stop and gradually increased its velocity, reaching a top speed of 70 Km/h. The results obtained show that the use of the FEC proposed yields higher throughputs and lower jitters than other schemes.

In contrast to the above-mentioned papers, a channel model obtained from empirical measurements is used in (57). There, the authors compare the performance of several channel estimation methods applied to an IEEE 802.16e transceiver that transmits through two VTV (vehicle-to-vehicle) channels modeled from data collected at a frequency of 5.12 GHz. Such channels are described in (96) and consider two typical vehicular environments: a high traffic density highway and an urban area. The performed simulations use the uplink PUSC (Partial Usage of Subchannels) mode and show that the proposed channel estimation techniques give a good tradeoff between performance and computational complexity.

Finally, there are several papers in the literature that compare the results obtained using empirical channels and real-world measurements. For instance, (97) develops a Mobile WiMAX downlink physical layer simulator that transmits over the 3GPP-SCM channel model for urban microcells (98). To validate the simulation results, the authors compared them with data obtained through a measurement campaign in a real urban cell scenario. In the simulated environment, the transceiver was configured to transmit data to three mobile stations which traveled at 40 Km/h, recreating a situation very similar to the one faced during the measurement campaign. After comparing the simulated and the empirical results, it is concluded that their Mobile WiMAX simulator could be used to predict the performance (in terms of packet error rate versus signal-to-noise ratio) for a range of environments, transmit power levels and antenna configurations.

Although all the references mentioned provide a good idea about the capacities of Mobile WiMAX for transmitting in certain vehicular environments, their conclusions are only partially valid for the future VTV and RTV (road-to-vehicle) communications. On the one hand, most of the papers focus on the study of transceivers that do not work at 5.9 GHz, the frequency where ITS applications are likely to be developed. On the other hand, most of the channels analyzed consider either very generic or very specific environments. There is a lack of a proper Mobile WiMAX performance evaluation through different environments representative of the most common vehicular scenarios.

All the experiments shown in this Thesis are focused on obtaining PHY-layer performance results, constituting the basis for simulations of higher layers. Regarding the PHY layer, the most important performance metrics are the BER and the Frame-Error Rate (FER) versus SNR or Signal-to-Interference plus Noise Ratio (SINR). These performance metrics adequately characterize the PHY layer and can be used as input to network simulators such as QualNet or `ns-2`. Network simulators usually implement the MAC layer of wireless communications standards, but they model the PHY layer by just using simple BER calculation expressions which do not take into account complex scenarios such as the vehicular environments considered in Chapters 5 and 6. As indicated in (99), BER results obtained through sophisticated PHY layer simulations can be incorporated in QualNet (in the form of lookup BER/FER Vs SINR tables) to obtain much more realistic network simulations.

From a general perspective, the impact on network simulation when using realistic PHY metrics was analyzed in (100). There, the authors consider a set of factors at the PHY layer that are relevant to the performance evaluations of higher layer protocols. Such factors include signal reception, path loss, fading, interference and noise computation, and preamble length. The authors quantify the impact of these factors under typical scenarios used for the performance evaluation of wireless ad hoc routing protocols. Their experimental results show that the factors at the PHY layer not only affect the absolute performance of a protocol but, because their impact on different protocols is non-uniform, it can even change the relative ranking among protocols for the same scenario.

4.3.2 Mobile WiMAX Transceiver

The implemented transceiver is focused on the PHY layer referred to in the IEEE 802.16e standard as WirelessMAN-OFDMA. Fig. 4.6 depicts the block diagram of a evaluation system where the transmitter and the receiver are connected to the vehicular channel emulator described in Chapter 5. Such figure shows that Mobile WiMAX has been defined in a similar way to other regular OFDM (Orthogonal Frequency-Division Multiplexing) communication systems. In fact, the implementation uses a variation of OFDMA (Orthogonal Frequency Division Multiple Access) called SOFDMA (Scalable OFDMA) that makes it possible to work with multiple users simultaneously and which can be adapted to different environments and circumstances.

Among the different Mobile WiMAX working modes, it was chosen to operate in a mode called Downlink PUSC. In this mode, the 512 subcarriers are divided into 360 subcarriers for data, 60 for pilots and 92 for the guards and the DC. Each fourteen adjacent subcarriers over two OFDMA symbols constitute a cluster or resource block (24 subcarriers for data and 4 for pilots). Furthermore, each OFDMA symbol is divided into fifteen subchannels but, for the sake of simplicity, all subchannels are assigned to a unique user.

4.3.2.1 Mobile WiMAX MATLAB/Simulink Transmitter

Fig. 4.6 (left) depicts the transmitter block diagram. The implementation follows closely the indications given in Section 8.4 of (101), although several modifications were made in order to simplify the design and reduce simulation time. Such differences are described below.

First, a Random Integer Generator (RIG) block generates equally likely data bits. For each transmission, the model generates enough bits to fill the desired number of OFDMA symbols to be transmitted. An IEEE 802.16e transmitter works with slots instead of symbols: one slot is equal to two consecutive OFDMA symbols. Hence, the minimum number of OFDMA symbols to be transmitted is two. Also, it should be noticed that in our tests we have considered that a fair comparison between the different standards should be performed measuring the FER (FEC block error rate). Since each FEC block contains 48 data bits and there are 720 data subcarriers for each slot (two OFDMA symbols), using QPSK and a rate 1/2 code, it is concluded that in each slot the transmitter sends 15 FEC blocks.

The generated bits are randomized using a randomizer that is initialized for each FEC block. Although the standard states that padding with 0xFF shall be added if the amount of data to be transmitted does not fit the amount of allocated data, padding is never performed since it is always generated the exact amount of data. The

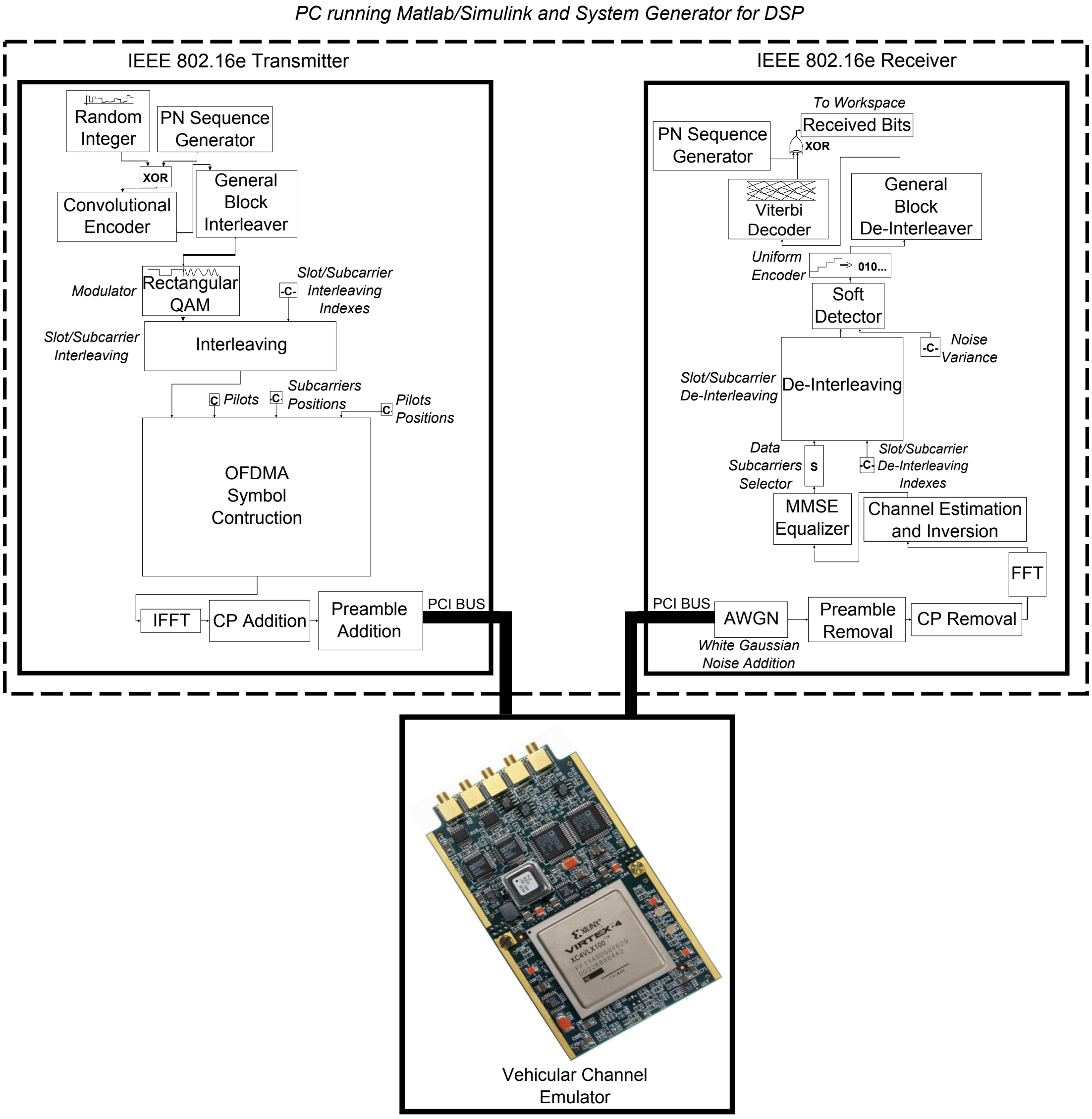

Figure 4.6: Evaluation system of a MATLAB/Simulink IEEE 802.16e transceiver using a vehicular channel emulator.

randomization process is carried out by XOR-ing the generated bits with a sequence obtained from a 15-bit randomization vector whose initialization value is indicated in Section 8.4.9.1 of (101). The randomized bits are convolutionally encoded with the industry-standard generator polynomials $g_0 = 133$ and $g_1 = 171$. In the tests a rate 1/2 encoder was used but, in the event of other rates being needed (2/3 and 3/4 are supported), puncturing can be employed using the patterns given by the standard.

Block interleaving is then carried out in a two-step permutation. The first permutation ensures that adjacent coded bits are mapped onto non-adjacent subcarriers, while the second permutation ensures that adjacent coded bits are mapped onto less and more significant bits of the constellation to avoid long runs of bits with low reliability.

After block interleaving, the bits are Gray-mapped into QAM symbols and interleaved twice subsequently. The first interleaving stage affects the QAM data symbols contained in one slot (i.e. two adjacent OFDMA symbols) while the second stage reorganizes the QAM data symbols inside each OFDMA symbol. With respect to the pilots, the standard uses a Pseudo-Random Binary Sequence (PRBS) generator during their modulation. However, since this generator is only used to provide additional security, we decided not to include it in our design in order to reduce complexity and simulation time.

The interleaved QAM data symbols are placed into their corresponding subcarriers in each OFDMA symbol. Notice that the positions of pilots and data change depending on whether they are placed into an odd or an even OFDMA symbol according to the structure imposed by the Downlink PUSC mode. Once the OFDMA symbol is built, the IFFT is performed and a 1/4 CP is added.

Finally, a preamble is appended to the whole frame. Although the preamble is always transmitted in order to comply with the requirements of the standard, it is not used in reception since perfect time synchronization is assumed and it was not implemented any algorithm that may need it. Moreover, note that the vehicular channel emulator operates in baseband, so there is no IF stage, neither at the transmitter nor at the receiver.

4.3.2.2 Mobile WiMAX MATLAB/Simulink Receiver

The receiver block diagram is shown on the right of Fig. 4.6. The first step is the addition of white Gaussian noise in order to obtain BER and FER curves versus E_b/N_0 values. The E_b/N_0 parameter is the received bit energy to noise power spectral density ratio commonly referred to as received SNR per bit.

After removing the preamble and the CP, the FFT is applied to each OFDM symbol and the channel is estimated. Although different channel estimation techniques were tested, a simple method was used in the experiments presented in Section 5.10.5: the pilots are first extracted and then divided by their respective transmitted values (which are known at the receiver), obtaining the estimated channel coefficients for the pilot subcarriers. Such estimates are linearly interpolated to obtain the channel frequency response for the remaining subcarriers. Moreover, to improve channel inversion, an MMSE equalizer (89) is applied.

The equalized symbols are de-interleaved (at slot and symbol levels) and then sent to a soft detector whose output LLRs are also de-interleaved and decoded using a Viterbi block. Finally, the decoded bits are de-randomized applying an XOR gate, using the same PN sequence generated at the transmitter side, and the final bits are obtained.

4.4 Conclusions

This Chapter has detailed the design and implementation of MATLAB/Simulink transceivers able to transmit through the channel emulators described in the next Chapters. Specifically, in Section 4.2 IEEE 802.11p (SISO and MIMO) and IEEE 802.11a transceivers were described after giving a brief summary on IEEE 802.11p evaluations performed by other authors. Section 4.3 summarized some of the latest Mobile WiMAX performance measurements and then detailed the elements that make up the implementation of the PHY layer of a Mobile WiMAX transceiver.

An extensive evaluation of these three transceivers will be given in the next Chapters, where they are connected to SISO and MIMO channel emulators able to recreate different vehicular scenarios.

5

Rapid prototyping for SISO vehicular channel emulators

5.1 Introduction

To assess the performance of IEEE 802.11p systems it is desirable to evaluate it in realistic situations. Tests can be performed directly in a vehicle, driving through different environments, but that is a time-consuming task and the experiments can be affected by unintended side effects that may be uncontrollable. It is more convenient to use a hardware channel emulator and measure the performance inside a testing lab with a testbed platform, as in (10, 102, 103). Nevertheless, commercial channel emulators are usually very expensive and may not offer enough flexibility when configuring certain wireless channel parameters.

These observations motivated the development of a low-cost and flexible alternative to evaluate the performance of an IEEE 802.11p system in vehicular environments in conjunction with the testbeds described in Chapters 2 and 3, and the software transceivers detailed in Chapter 4. The channel emulator was designed to be reconfigured fast and easily, providing enough flexibility to implement customized wireless channel models. To achieve these goals, rapid-prototyping software was used for developing the evaluation framework: MATLAB/Simulink for the transceiver and Xilinx System Generator for the FPGA-based vehicular channel emulator. System Generator is especially useful because it allows designing the channel emulator much faster than using conventional hardware description languages like VHDL or Verilog.

The implemented channel models are based on (60) and (104), where the authors obtained channel models from data acquired during a measuring campaign carried

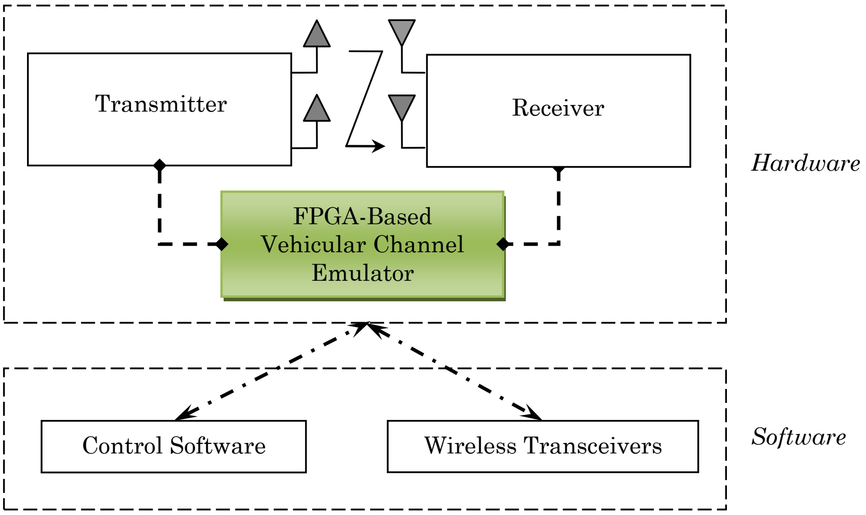

Figure 5.1: In color: components of the rapid prototyping system that are described in Chapter 5.

out in a metropolitan area at a frequency of 5.9 GHz. Six different situations were considered, including measures in common vehicular environments such as highways, urban canyons and suburban areas.

This Chapter is structured as follows. Section 5.2 details some of the latest commercial and academic Single-Input Single-Output (SISO) channel emulators. Section 5.3 describes the emulated vehicular models, whose theoretical and practical implementation is extensively covered by Sections 5.4 to 5.9. Finally, Section 5.10 exposes the results of the SISO performance measurements whereas Section 5.11 is devoted to the conclusions.

As it can be observed, the present Chapter will be mainly focused on the rapid prototyping element highlighted in Fig. 5.1: the FPGA-based channel emulator.

5.2 Background: Commercial and Ad-Hoc Channel Emulators

Channel emulation is typically used when evaluating product performance in realistic situations before commercial release. With the aid of a channel emulator the equipment manufacturers avoid unintended interferences, hence the simulation environment can be controlled. Furthermore, the tiresome task of performing successive field measurements is limited to the minimum (to obtain the channel model, if there is none already available) and the rest of the experiments are carried out inside a testing lab.

There are many commercial channel emulators that are manufactured by companies such as Spirent (105), Rhode & Schwarz (106), Azimuth Systems (107), Agilent (108)...These emulators are usually general-purposed (for instance, Spirent's SR5500 or Rhode's AMU200A), but there are some that are aimed at evaluating a specific technology, like Azimuth's 400WB Multiple-Input Multiple-Output (MIMO) Channel Emulator (for IEEE 802.11n systems) or Agilent's N5106A PXB MIMO Receiver Tester (with built-in LTE and Mobile WiMAX channels).

All these channel emulators are robust and work great for most applications, but they have two problems. On the one hand, they are usually really expensive. Although their prices have been dropping in the last years, not a long time ago some researchers (20) paid up to 150,000 € for their channel emulator (a Spirent TAS4500 FLEX), what it is clearly superior to the roughly 3,000 € that costs the FPGA development kit used to implement the emulator presented in this Chapter. The second problem of commercial emulators is that they may not offer enough flexibility to researchers when configuring certain channel parameters. To tackle these issues a number of low-cost ad-hoc channel emulators have recently been proposed.

To develop a low-cost easily-reconfigurable channel emulator, different technologies can be assessed. Due to real-time constraints, microcontrollers and affordable Digital Signal Processors (DSPs) are not valid. Among the rest of the semiconductor technologies, three of them are especially suitable for implementing a channel emulator: CPLDs (Complex Programmable Logic Devices), ASICs (Application-Specific Integrated Circuits) and FPGAs. Although ASICs offer superior performance, their development time is too high for a prototype (they are mainly used for final products). Thus, the choice has to be made between CPLDs and FPGAs. The latter are the most commonly used due to two reasons:

- FPGAs own specific resources that are more adequate for implementing a channel emulator, such as counters or arithmetic operator blocks.
- Although CPLDs can execute tasks at a faster rate, the maximum allowed complexity of the designs is inferior to that offered by an FPGA.

Therefore, most of the proposed ad-hoc emulators are based on FPGA technology. Moreover, during the last years FPGAs have become a more attractive tool to researchers: their unitary cost has been dramatically reduced and new implementation tools make the formerly tedious tasks of programming and design verification easier.

Some examples of FPGA-based channel emulators are described in (109, 110, 111). In (109) an FPGA-based AWGN (Additive White Gaussian Noise) channel emulator is implemented. The emulator is based on a hardware white Gaussian noise generator that is developed by combining the Box-Muller and Central limit theorems, and designing the whole model in VHDL.

Similarly, in (110) the authors use a Xilinx Virtex-II Pro to implement a fading channel emulator. The fading process models use sum-of-sinusoids (SOS) algorithms that allow designing and implementing Rician and Rayleigh fading channels. The final designs use only between 2% and 5% of the FPGA slices and are able to generate 201 million 16-bit complex-valued fading samples per second.

Finally, (111) presents a baseband multipath fading channel emulator implemented on a Virtex-IV using Xilinx XtremeDSP FPGA platform. The emulator is implemented using Simulink models and System Generator IP blocks. The final design is limited to a two-path channel due to the extensive use of FPGA resources; the input/output rate is set to 20 MHz; and the Doppler frequency is 5 Hz. After the verification stage, the emulator was downloaded to the FPGA board and tested with a baseband QPSK signal, achieving results similar to those obtained through MATLAB simulations.

The above mentioned developments have at least two major drawbacks. First, the use of low-level description languages such as VHDL results in slow development stages. Although in most cases VHDL allows obtaining a resource-efficient FPGA design, programming can become a cumbersome task that may consume a large amount of time and economic resources. There are new sophisticated tools like System Generator which permit to work with high-level blocks that enable to build complex designs easier and faster.

The second problem is precisely related to the use of high-level tools. These tools facilitate fast prototyping but they usually generate non-optimized large designs that may not fit into the FPGA. For instance, in (111) the authors only download a two-path channel emulator due to the lack of available hardware resources. Hence, for large designs, optimizations must be made.

Several papers have also identified and quantified the effects related to the cited drawbacks. For instance, in (112), the authors used rapid prototyping tools (C, MAT-

LAB and Simulink) to accelerate the implementation process. Additionally, in order to save space, they limited their channel emulator to 8 or 16 paths (depending on the channel model). To quantify the error introduced by this reduction of the number of paths, they compared the average RMS (Root Mean Squared) delay spread of the original and the reduced model, finding a lower delay profile in the latter, what caused an error in the channel emulation.

Another interesting paper is (113), which presents a MIMO development platform made of scalable processing boards that contain TI C6414 DSPs and Xilinx Virtex 2 FPGAs. The article is not focused on saving FPGA resources, but it constitutes a good example of how to shorten the development cycle using rapid prototyping and hardware-in-the-loop techniques. In fact, the authors were able to build fast a MIMO channel emulator based on the COST 259 recommendations, splitting the processing into two stages: the impulse response of the channel resides in the DSP while the convolution of the transmitted signal with the impulse response is performed in the FPGA.

The last example of this kind of emulators is (114). There, the authors implemented a configurable multipath fading channel emulator using MATLAB and Xilinx System Generator. They built a resource-efficient white Gaussian noise generator (it occupies less than 450 logic cells of a Xilinx Virtex-IV) and they made use of IIR (Infinite Impulse Response) Doppler filters instead of FIR (Finite Impulse Response) filters in order to avoid the utilization of a large number of taps, typically required by the latter (however, their IIR filters are less accurate when representing the original Doppler spectrum).

The vehicular emulator described in this Chapter addresses the two mentioned drawbacks: System Generator was used to develop the channel emulator faster than using VHDL and the resulting design was optimized in order to be able to implement a twelve-path channel emulator, even leaving space for future extensions.

5.3 Implemented Vehicular Channel Models

The implemented channel models are based on the excellent work of (60) and (104), which is mainly based on a measurement campaign at 5.9 GHz carried out in the spring of 2006 in Atlanta, Georgia. Thanks to those measurements the authors were able to obtain channel models for six different environments that cover some of the most common situations where VTV (Vehicle-To-Vehicle) and RTV (Road-To-Vehicle) communications may take place. The six selected environments can be grouped into three major scenarios:

- Urban canyons, with dense and high-rise buildings, where vehicles moved at speeds between 32 Km/h and 48 Km/h.
- Suburban expressways, with moderately dense, low-rise buildings, where the measurement speed was approximately 105 Km/h.
- Suburban surface streets, with moderately dense, low-rise buildings, where the driving speed was between 32 Km/h and 48 Km/h.

In those environments, the measurements were taken in two different ways depending on the situation. For VTV communications, data was exchanged between vehicles with 8 dBi omnidirectional antennas that were mounted magnetically on the roof. For RTV channels, the communications were performed between a vehicle with one of the mentioned omnidirectional antennas and a monopole or half-dome antenna mounted on a pole by the side of the road at a height of 6.1 meters.

The main characteristics of the vehicular scenarios are the following (for a more detailed description of the measurements see Section 5.2 in (104)):

- **VTV-Expressway Oncoming**. Two oncoming vehicles approached at high speeds in a stretch of highway without middle wall. Both vehicles were synchronized to enter the highway at the same time and then each vehicle accelerated to 105 Km/h. During the modeling, the authors only considered the measurements taken when the vehicles were at a distance of 300-400 meters, where they observed a fixed LOS (Line-of-Sight) Doppler shift in the first path.
- **VTV-Urban Canyon Oncoming**. Two oncoming vehicles at 32-48 Km/h approached in an urban canyon. During the modeling, only the data segments obtained with a separation of 100 meters were selected.
- **RTV-Suburban Street**. A vehicle at 32-48 Km/h received packets from a monopole antenna placed next to a suburban street intersection. The vehicle approached the intersection from any of the four possible directions. Only the

data acquired at 100 meters were taken into account in the model. Moreover, the data segments with a signal level lower than a threshold were discarded from the modeling process.

- **RTV-Expressway**. The communications were performed between a half-dome antenna mounted on a pole off the side of an expressway and a vehicle at 105 Km/h. The measurements were taken at a distance of 300-400 meters as the vehicle approached from both directions of the expressway.
- **VTV-Expressway Same Direction With Wall**. For this scenario, the data were obtained in different locations along several expressways where there was a middle wall between oncoming lanes. In all these situations, two vehicles at 105 Km/h carried out communications when travelling in the same direction. Only the measurements at a distance of 300-400 meters were used during the modeling of the channel.
- **RTV-Urban Canyon**. A transmitting monopole antenna was mounted on a pole near an urban intersection and a vehicle was approaching at a speed of 32-48 Km/h. For the model, only the measurements at a distance of 100 meters were taken into account. To avoid deep fades, the data with a signal level lower than an arbitrarily set threshold were not used in the modeling of the channel.

The values of the parameters of the vehicular models are clearly summarized at the end of (60). Although the measurement campaign was performed at 105 Km/h in expressways and 32 Km/h to 48 Km/h for surface streets, the authors scaled the models to make their Doppler frequencies consistent with vehicle speeds of 140 Km/h and 120 Km/h, respectively. Furthermore, the distances were also made consistent with 100 meters for surface streets and approximately 400 meters for expressways, what seem to be typical ranges where vehicles carrying IEEE 802.11p transceivers are expected to operate.

It must be mentioned that the authors have also proposed a modification of the *RTV-Expressway* channel with Doppler frequencies consistent with a speed of 200 Km/h (104). Such channel has not been implemented in the FPGA-based channel emulator because of the similarities with the model of the same scenario at 140 Km/h and to carry out a fair comparison at the same speed among the different expressway channels.

Table 5.1: Main characteristics of the SISO vehicular models.

Vehicular Channel	Distance TX - RX (m)	Speed (km/h)	Path Modulation (number of paths)	Maximum Delay Spread (ns)	Rician K (dB)	Overall K Factor (dB)	Maximum Freq. Shift (Hz)	Maximum Fading Doppler (Hz)	LOS Doppler (Hz)
VTV-Expressway Oncoming	300-400	140	Rician (1) / Rayleigh (10)	302	-1.6	-3.6	1466	858	1452
RTV-Urban Canyon	100	120	Rician (1) / Rayleigh (11)	501	7.5	6.7	720	994	654
RTV-Expressway	300-400	140	Rician (1) / Rayleigh (11)	401	-5.3	4.3	769	813	770
VTV-Urban Canyon Oncoming	100	120	Rician (1) / Rayleigh (11)	401	4.0	3.0	1145	936	1263
RTV-Suburban Street	100	120	Rician (1) / Rayleigh (11)	700	3.3	2.1	648	851	635
VTV-Express. Same Dir. with Wall	300-400	140	Rician (2) / Rayleigh (10)	701	23.8, 5.7	3.3	-561	1572	-60, 40

The key characteristics of the vehicular channels are summarized in Table 5.1. For each model, the following parameters are shown: distance between the transmitter and the receiver, speed of the vehicle, number of paths of the channel and their modulation (Rician or Rayleigh), maximum delay spread, Rician K for the Rician paths, overall K factor (i.e. the ratio of the deterministic power over the total random power of all taps), maximum frequency shift for all paths, maximum fading Doppler (i.e. maximum half-width of the fading spectral shapes of all the paths of each channel) and LOS Doppler of the Rician paths.

Table 5.1 also gives an idea of the complexity involved in the implementation of the channels. These high speed and high delay spread scenarios own large Doppler shifts that force the emulator to interpolate and rapidly update each path coefficients. Moreover, although the amount of required FPGA hardware is reduced by working with the baseband IQ components at 10 MHz, it is not possible to implement the six channels into a single FPGA: as it can be seen in Table 5.2, 76% of the slices are occupied in the best case and, regarding the rest of the resources (flip-flops, LUTs (Look-Up Tables), FIFO16/RAMB16 memory blocks and DSP48 blocks), between 19% and 65% are used. Thus, six independent .bit files are generated, though in practice, only three different FPGA designs are needed due to the model similarities:

- One design is exclusively dedicated to the channel *VTV-Expressway Oncoming* which is the only one with eleven paths.
- Another model is used for *VTV-Expressway Same Direction with Wall* because it requires the existence of two Rician and ten Rayleigh paths, while the rest of the channels (apart from *VTV-Expressway Oncoming*) consists of just one Rician path and eleven Rayleigh paths.

- One design for the other four channels, which differ in their configuration parameters but share all their FPGA resources.

Table 5.2: General parameters and resources occupied by the SISO vehicular channel emulator.

Vehicular Channel	Coefficient generation rate [Effective rate] (Hz)	Interpolation rate	Occupied Slices (%)	Occupied Slice Flip-Flops (%)	Occupied LUTs (%)	Occupied FIFO16 / RAMB16s (%)	Occupied DSP48s (%)
VTV-Expressway Oncoming	3484 [4000]	x2500	76%	36%	50%	19%	60%
RTV-Urban Canyon	2194.6 [2500]	x4000	84%	39%	57%	20%	65%
RTV-Expressway	2168 [2500]	x4000	84%	39%	57%	20%	65%
VTV-Urban Canyon Oncoming	3314 [4000]	x2500	84%	39%	57%	20%	65%
RTV-Suburban Street	1988 [2000]	x5000	84%	39%	57%	20%	65%
VTV-Express. Same Direction With Wall	3170 [4000]	x2500	85%	40%	57%	24%	65%

Finally, it should be mentioned that all models experience the same latency in the channel emulator (the latency is the time delay between the input and output signals). In the measurements performed the value of the latency is always 2.813 ms. The latency is important when testing real transceivers although in the experiments shown in this Chapter this information is not necessary, since the channel emulator is evaluated using a software transceiver.

5.4 Theoretical Model

For the generation of each channel coefficient at the i-th path in the time instant t, the following model was used

$$\begin{aligned} h(i,t) &= \sqrt{K_i P_i/(K_i+1)}\,\overline{h}(i,t) \\ &\quad +\sqrt{P_i/(K_i+1)}\,h_w(i,t) \end{aligned} \tag{5.1}$$

where

- K_i: Rice factor of the i-th path.
- P_i: power of the i-th path.
- $h_w(i,t)$: represents the contribution of the NLOS (Non Line-of-Sight) component to the i-th path at the time instant t. It is a random variable that follows a complex Gaussian distribution with mean zero and unit variance.

- $\overline{h}(i,t)$: contribution of the LOS component to the i-th path at the time instant t. It is determined by

$$\overline{h}(i,t) = e^{j(2\pi f_{D,i} cos(\theta_i)t+\phi_i)} \tag{5.2}$$

  where

  - $f_{D,i}$: maximum Doppler spread of the i-th path.
  - θ_i: angle of arrival of the i-th path.
  - ϕ_i: phase of the LOS component of the i-th path.

To decrease the number of input configuration parameters, several of the operations involved in (5.1) and (5.2) are calculated offline. As it can be viewed in Fig. 5.2, the emulator only needs five parameter blocks:

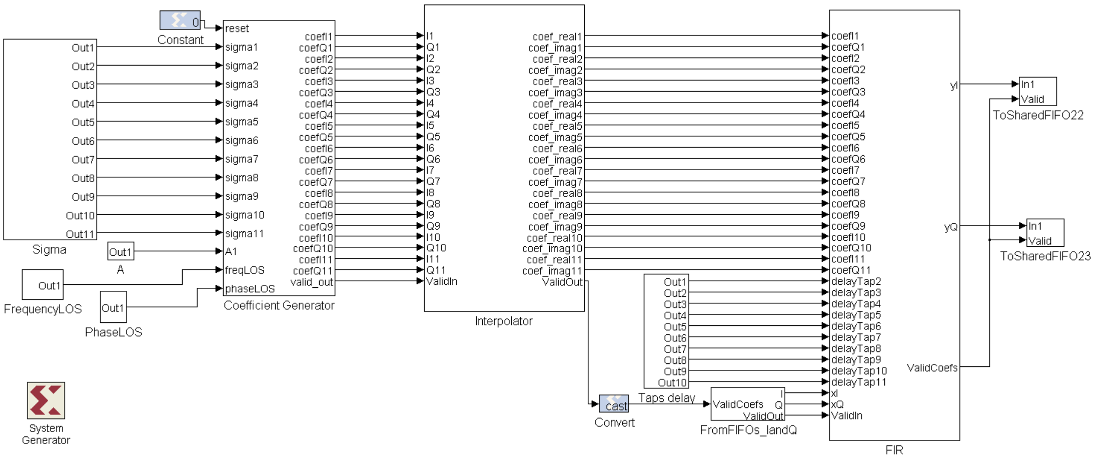

Figure 5.2: General view of the System Generator model optimized for the vehicular SISO channel *VTV-Expressway Oncoming.*

- `Sigma` block contains the power factors of the NLOS components:

$$\sigma = \sqrt{P_i/(K_i+1)}$$

- `A` block holds the power factors of the LOS components:

$$A = \sqrt{K_i P_i/(K_i+1)}$$

- `FrequencyLOS` block contains part of the exponent of $\overline{h}(i,t)$:

$$f_{\text{LOS}} = 2\pi f_{D,i} cos(\theta_i)$$

- `PhaseLOS` block is simply ϕ_i.
- `Taps delay` block holds the normalized delays of the different paths. Note that the delays need to be normalized since the FPGA sampling period is 100 ns, but the model has delay values that differ in only one nanosecond (for instance, paths 6 to 8 of *VTV-Expressway Oncoming* are delayed 200 ns, 201 ns and 202 ns respectively). The authors of the model justify this in Section III in (60): to avoid problems with their commercial channel emulator (Spirent SR5500), paths comprising a single tap were separated in delay by one nanosecond. Instead, the FPGA-based channel emulator described in this Chapter does not present such problem and, thus, the one-nanosecond artificial delays in the models are suppressed. That means that the previously mentioned delays of 200 ns, 201 ns and 202 ns, will occur at the same time instant (200 ns).

5.5 Hardware and Software

The vehicular channel emulator is implemented on an FPGA using Nallatech's BenADDA-IV development kit, which has the following features:

- It contains a Virtex IV (XC4VSX35-10FF668) that allows using Xtreme-DSP slices of up to 400 MHz.
- It has two 14-bit ADCs (Analog-to-Digital Conversors) able to work up to 105 MS/s and two 14-bit DACs (Digital-to-Analog Conversors) that can run up to 160 MS/s.
- Dedicated internal clock up to 105 MHz (although it can use an external clock).
- 4 MB of 166 MHz ZBTRAM (Zero-Bus Turnaround RAM).

- Possibility to connect the kit using a PC (via the PCI bus) or in stand-alone mode.

In order to decrease the amount of time required to implement the channel model on the FPGA, System Generator for DSP was used since it enables to design and program the Virtex IV faster. It allows using libraries of high-level blocks and can interact with MATLAB and Simulink. Moreover, another advantage of this software is its ability to exchange data between a design running on the FPGA and a software implementation that is executed on a PC. In fact, in the tests exposed in Section 5.10 a MATLAB/Simulink IEEE 802.11p transceiver interacts with the vehicular emulator, which is running on the FPGA.

5.6 Hardware Set-up to Evaluate Physical Transceivers

Although in this Chapter it is described and shown performance results obtained using a software transceiver, it is straightforward to connect the emulator with a hardware testbed or with commercial transceivers. Depending on the transceiver outputs, there are two ways of setting up the hardware.

On the one hand, if the transceiver provides IF (Intermediate Frequency) or RF (Radio Frequency) outputs, we need a hardware setup like the one depicted in Fig. 5.3. The transmitter is connected to a downconverter that obtains the IQ representation of the signal, which is sent to the emulator ADCs. After the emulator applies the channel model in real time to the IQ digitized signal, the output samples are forwarded to the emulator DACs. Then, the analog signal is sent to the upconverter, which, finally, transmits it to the receiver.

Since the emulator does not perform down or up conversions, additional hardware is required for this purpose. In the case shown in Fig. 5.3, both operations are carried out by commercial devices. A spectrum analyzer equipped with a downconversion module (Agilent E4404B) is used to downconvert signals of up to 6.7 GHz. The upconversion is performed by using a vector signal generator (Agilent E4438C) that receives the IQ signal from the emulator and upconverts it to frequencies up to 6 GHz.

On the other hand, if the physical transceiver is able to provide the emulator with IQ outputs, then the up and downconversion stages are not needed. The signals would be acquired by the emulator ADCs and sent directly from the DACs to the receiver.

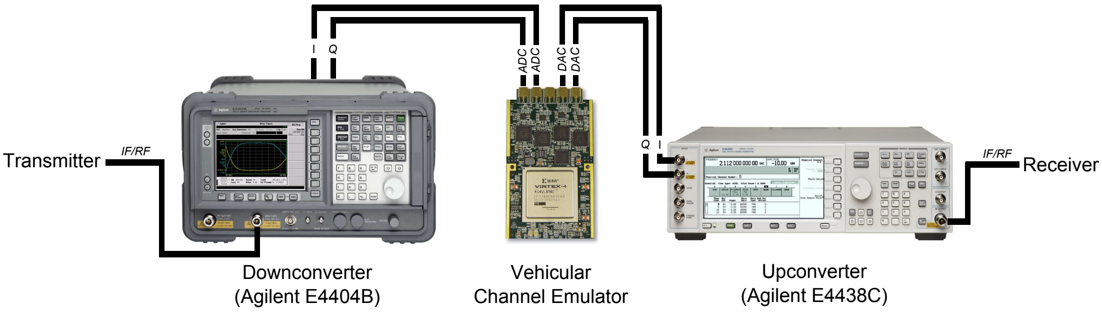

Figure 5.3: Hardware set-up for testing an IEEE 802.11p physical transceiver with IF/RF outputs on the vehicular channel emulator.

5.7 FPGA Design Overview

Fig. 5.2 shows a general view of the hardware design. Several blocks represented in such figure contain sub-blocks which are shown throughout this Chapter: the `Coefficient Generator` block includes Figs. 5.4, 5.5 and 5.6, the `Interpolator` block contains a number of interpolators like the one shown in Fig. 5.7 and the `FIR` block is shown in Fig. 5.8.

Notice that the design presented in Fig. 5.2 is optimized for a specific channel (*VTV-Expressway Oncoming*) in order to minimize the amount of required hardware, although the rest of channels share most of the hardware resources, as it was mentioned at the end of Section 5.3.

The design can be divided into different parts that carry out six different major tasks: acquisition of the channel parameters, Gaussian noise generation, Doppler filtering, LOS Doppler generation, interpolation and FIR filtering.

5.7.1 Acquisition of the Channel Parameters

The generation of the configuration parameters of the vehicular channel is performed offline to reduce the amount of used hardware and due to the fact that such parameters remain constant throughout the whole emulation process. The parameters are stored into registers that will be read by the FPGA-based emulator. Notice that every channel has its own peculiarities and all the parameters equal to zero can be removed from the design to save hardware. For example, all the channels but *VTV-Expressway Same Direction With Wall* have only one Rician component, so in these channels we only need one register to store each of the LOS parameters detailed in Section 5.4.

5.7.2 White Gaussian Noise Generation

To obtain NLOS coefficients, first, we need to use the System Generator's White Gaussian Noise Generator (WGNG) block that generates i.i.d. samples from a Gaussian distribution with zero mean and unit variance. Since the maximum number of complex paths is twelve, we need twenty-four real values of such noise samples that will be filtered depending on the Doppler shift experienced by each path.

Fig. 5.4 shows the way it was reduced the amount of FPGA resources consumed by the 24-output Gaussian noise generator: instead of using twenty-four independent WGNG blocks, the samples produced by only one WGNG block are multiplexed in time. This optimization is crucial since each WGNG block consumes an important amount of FPGA resources.

For this specific vehicular emulator, the use of only one WGNG block together with a multiplexer results in a functionality that is equivalent to making use of twenty-four parallel WGNG blocks. Since each WGNG block runs at 10 MHz, using a twenty-five output multiplexer like the one shown in Fig. 5.4 (twenty-five is the integer divider of 10 MHz closest to twenty-four), produces twenty-four noise samples at a frequency of 400 KHz that still is several orders of magnitude higher than the desired channel coefficient generation effective rates (see Table 5.2).

To quantify the amount of resources saved thanks to this optimization, the 24-output Gaussian noise generator was isolated in a different design (to measure the resources that it requires) and it was compared the optimized and the non-optimized versions (the latter uses twenty-two WGNG blocks as it was aimed at emulating the channel *VTV-Expressway Oncoming*). We would like to show the real savings but, unfortunately, our PC (Intel$^{®}$ Core 2 Duo E8500@4.33 GHz, 2 GB of RAM, Windows$^{®}$ XP) always ran out of memory while performing the synthesis of the non-optimized model due to the large amount of resources it requires. However, it is possible to obtain an estimation of the required resources using System Generator's Resource Estimator block. Table 5.3 shows the estimated amount of resources occupied by both versions. It can be clearly observed that WGNG blocks should be used carefully in order to avoid wasting FPGA resources.

Table 5.3: Estimated savings obtained thanks to the optimization of the Gaussian generator for the SISO channel emulator.

Resource type	Optimized 24-output Gaussian generator	Non-optimized 24-output Gaussian generator	Estimated Savings (%)
Slices	1347	20355	93.3%
Flip-flops	1729	19607	91.2%
Block RAMs	8	176	95.4%
LUTs	997	21850	95.4%
Embedded Multipliers	4	88	95.4%

If we needed to reduce even more the number of occupied resources, it would be possible to generate the channel coefficients in MATLAB and then transfer them to the FPGA. However, this solution is not desirable due to the following reasons:

- If the channel coefficients were only transmitted from MATLAB during the initialization phase, due to the limited amount of memory on the FPGA, there would be a time when the channel coefficients would have to be used again. Therefore, correlation in the output signal would appear.

- If we transfer a new set of channel coefficients from MATLAB at fixed intervals, we would not be able to use the emulator in stand-alone mode since we always would relay on having a computer running MATLAB linked to the FPGA.

5.7.3 Doppler Filtering

To generate the actual NLOS components, the generated white Gaussian noise samples have to be filtered according to each path's Doppler spectrum. This spectrum is determined by a fading spectral shape, a frequency shift and a maximum Doppler shift. Table 5.1 shows these latter two parameters for the considered channel models. Four different spectral shapes are considered: *round, flat, classic 3 dB and classic 6 dB* (104).

Fig. 5.5 shows the blocks that allow applying the Doppler spectrum to each Rayleigh path. Each Doppler filter consists of 256 coefficients. This filter size provides a good tradeoff between precision and hardware complexity. The coefficients were hard coded in each of the six `.bit` files since each filter is unique for each path of each vehicular channel. If we desired to build a more generic vehicular emulator, it would be possible

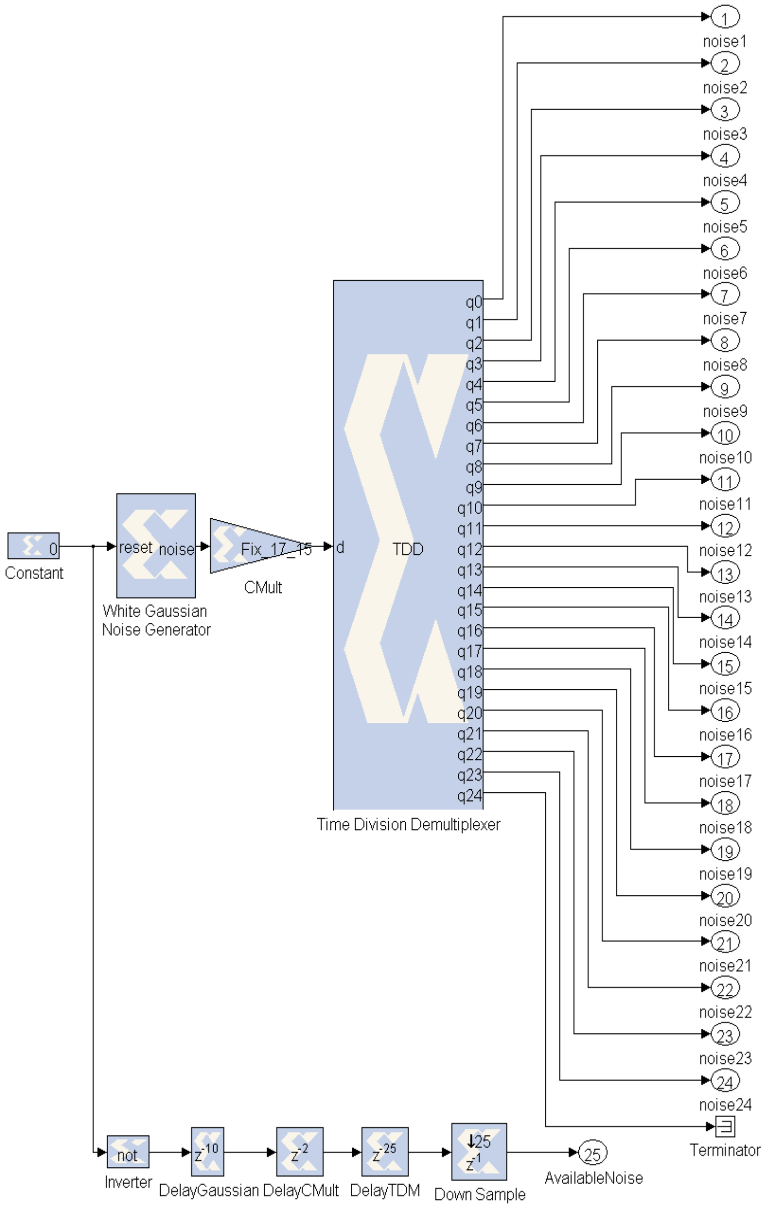

Figure 5.4: Resource-efficient 24-output Gaussian noise generator.

to reload dynamically the coefficient sets, but each filter block would need additional control lines.

To reduce to a half the required hardware, it is exploited the fact that the real and the imaginary parts of the filter can be used twice for each path to perform the complex FIR filtering (see Fig. 5.6). This optimized block can be seen in Fig. 5.5, which is

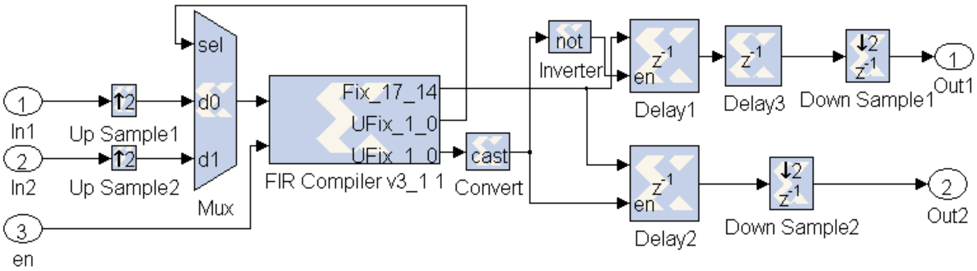

Figure 5.5: Optimized blocks for applying the Doppler Spectrum in the SISO vehicular channel emulator.

contained under the block *Doppler Filter* shown in Fig. 5.6.

Table 5.4 shows some of the resource savings achieved when reducing to a half the number of filters in a vehicular channel with eleven paths. Like in the previous subsection with the Gaussian generators, the optimized and non-optimized versions of the filter block were isolated and it was obtained the size they occupy. In this case, it was possible to synthesize both designs and obtain the actual hardware occupation without using the Resource Estimator block. It can be observed in Table 5.4 that, although the optimized block uses slightly more slices, the savings occur in the DSP48 and the RAMB16 blocks, that are reduced a 50%. This is important, since the lack of this type of blocks is a bottleneck to keep on designing the rest of the emulator.

Table 5.4: Savings due to the optimization of the Doppler filter block in the SISO vehicular channel emulator.

Resource type	**Optimized Doppler Filter block**	**Non-optimized Doppler Filter block**	**Total resources in the FPGA**	**Savings (%)**
Slices	7382	7239	15360	-1.9%
DSP48 blocks	88	176	192	50%
RAMB16 blocks	44	88	192	50%

Finally, LOS Doppler has to be also taken into account in the vehicular channels and must be applied to each Rician path according to Eq. (5.2). To achieve this, the System Generator's DDS (Direct Digital Synthesizer) block was used, which generates a sine and a cosine with the required phase and frequency parameters. Since the angle of arrival of the LOS component has not been considered in (60), it is always set to zero, what means that the received Rician paths arrive straight from the driving direction. This implies that the LOS Doppler is always equal to the path's maximum Doppler spread.

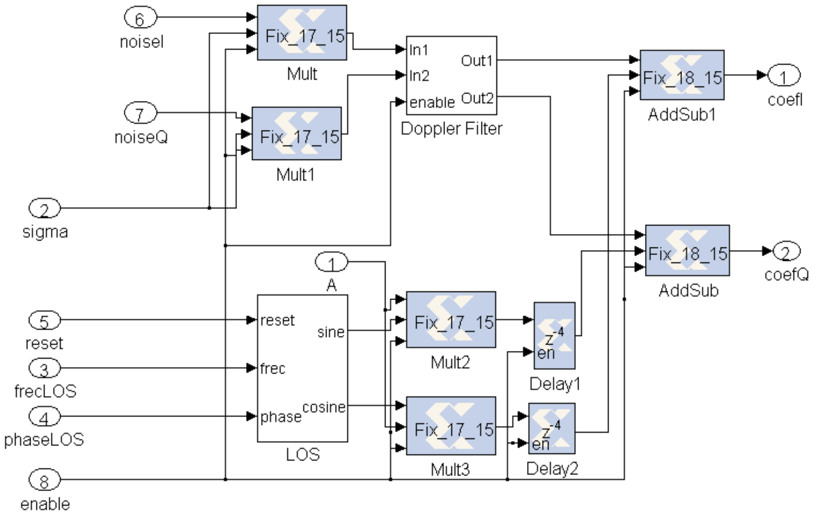

Figure 5.6: Generation and addition of the LOS and NLOS components of each path in the SISO emulator.

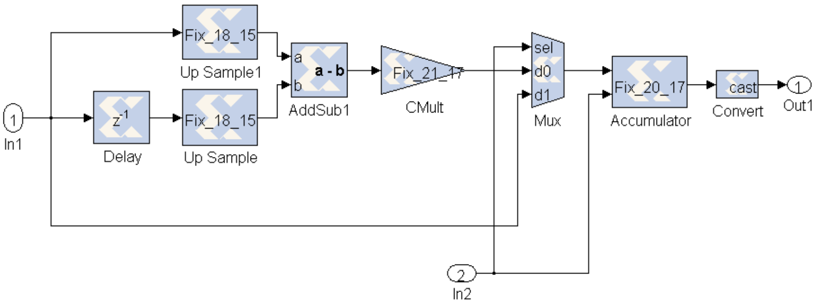

Figure 5.7: One path's linear interpolator of the SISO emulator.

5.7.4 Interpolation

After adding the LOS and NLOS components according to Eq. (5.1) (see Fig. 5.6), the coefficients must adapt their rate to the rate of the incoming signal (i.e. the signal from the transmitter that arrives at 10 MHz). These coefficients are generated at a rate that depends on the maximum Doppler shift and that is much lower than the FPGA frequency. Indeed, in a specific vehicular channel, the implicit sample rate is twice the maximum Doppler shift of all paths. In the implemented vehicular channel models, this rate fluctuates between 1,988 Hz and 3,484 Hz (see Table 5.2). To avoid designing a complex resampling stage, instead of using the original coefficient generation rate, an effective sample rate was applied, which is equal to the nearest superior integer divider of 10 MHz. Thus, it is only required to interpolate between two already generated coefficients. For doing this, two cascaded linear interpolators are used, each one having its own interpolation factor. The product of both interpolation factors gives the global interpolation rate of the corresponding channel (shown in Table 5.2).

Every single cascaded linear interpolator (see Fig. 5.7) obtains p intermediate values following these steps:

- At the time instant $t = 1, 2...$ the current coefficient, s_t, and the one generated at the previous time instant, s_{t-1}, are copied p times, being p the interpolation factor. As a result, we have two vectors of upsampled coefficients $\mathbf{s}_t = [\overbrace{s_t, s_t, ..., s_t}^{p}]$ and $\mathbf{s}_{t-1} = [\overbrace{s_{t-1}, s_{t-1}, ..., s_{t-1}}^{p}]$. It is assumed the initial condition $s_0 = 0$.
- Next, the vectors $\mathbf{s}_t$ and $\mathbf{s}_{t-1}$ are subtracted and divided by p to produce the difference vector
$$\mathbf{\Delta}_t = \frac{\mathbf{s}_t - \mathbf{s}_{t-1}}{p}$$
Obviously, $\mathbf{\Delta}_t = [\overbrace{\Delta_t, \Delta_t, ..., \Delta_t}^{p}]$ where
$$\Delta_t = \frac{s_t - s_{t-1}}{p}$$
- Finally, $\mathbf{\Delta}_t$ inputs an accumulator, that recursively computes the output as $y_t = y_{t-1} + \Delta_{\lfloor \frac{t}{p} \rfloor}$ for $t = 1, 2...$ where $\lfloor \cdot \rfloor$ denotes the integer value (floor) operator. Notice that provided that $y_0 = 0$, $y_t = s_{t/p}$ for $t = p,\ 2p,\ 3p...$ and the accumulator output corresponds to the output of a linear interpolator.

5.7.5 FIR Filtering

Finally, the signal from the transmitter is filtered with the interpolated coefficients. Fig. 5.8 depicts a general view of the developed complex FIR filter (this figure shows twenty-two paths instead of twenty-four because it is optimized for the *VTV-Expressway Oncoming* channel as in Fig. 5.2). In the diagram, two groups of blocks can be distinguished: blocks aimed at delaying the incoming signal (`DelayBufferI` and `DelayBufferQ`) and blocks for applying the complex FIR filter to the delayed signals (`FIR_Taps1` to `FIR_Taps4`).

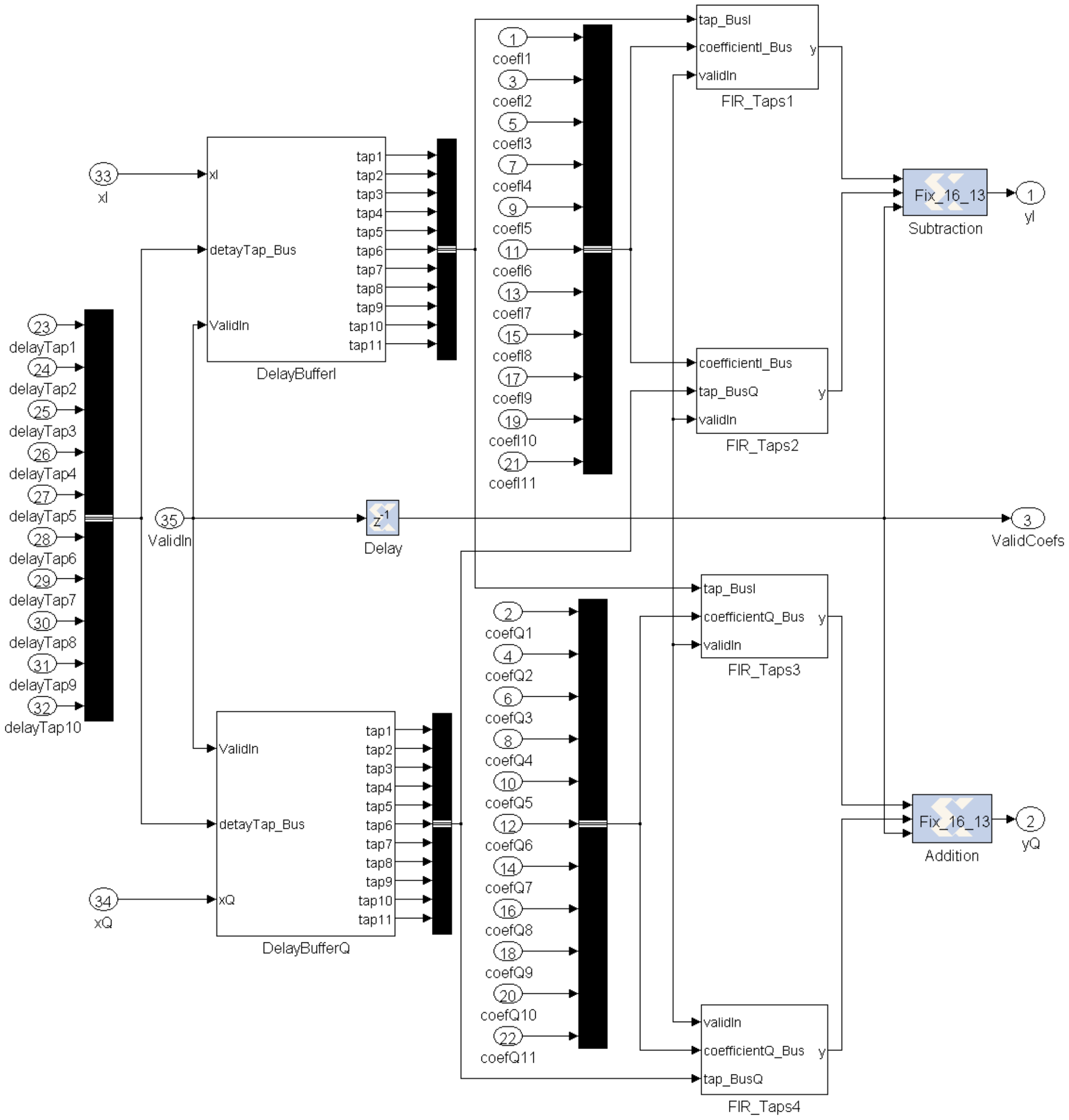

Figure 5.8: General view of the complex FIR filter of the SISO channel emulator.

5.8 Emulator Basic Operation

The emulator operation can be summarized as follows:

1. The configuration parameters of the vehicular channel are initially read from registers (shown in Fig. 5.2).
2. The emulator starts to generate channel coefficients, both for the LOS and the

NLOS components (illustrated in Figs. 5.2 to 5.6).

3. The coefficients are interpolated to have their rate adapted to the incoming signal rate, passing each path through linear interpolators (like the one shown in Fig. 5.7). The interpolation is carried out in two stages, whose interpolation factors depend on the effective generation rates shown in Table 5.2. For instance, in the channel *RTV-Urban Canyon*, the coefficients have an effective generation rate equal to 2,500 Hz. Since the incoming signal rate is 10 MHz, the coefficients need to be interpolated with a global factor of 4,000, which can be applied in two stages with interpolation factors 2^5 and 5^3.

4. Finally, the incoming signal is applied to a complex FIR filter that uses the channel coefficients generated (in Fig. 5.8).

5.9 Emulator input/output interface

The exchange of data between the software transceiver and the FPGA is performed through the PCI bus, making use of two different kinds of memory blocks:

- Registers. The configuration parameters are set at the beginning of the emulation using registers. The values of these parameters do not change throughout the emulation, but, if desired, they could be modified dynamically.

- Shared FIFO (First-In First-Out) memories. This sort of memories is used for driving the input/output IQ signals to/from the FPGA. The main advantage of using shared FIFOs is their ability to accelerate the simulation speed beyond what is typically possible with hardware co-simulation. Instead of transferring one sample per cycle, these blocks can work with frames of data that are sent in a single cycle to the FPGA. The largest available FIFO memory can hold up to 64,000 channel coefficients. Since each baseband OFDM (Orthogonal Frequency-Division Multiplexing) symbol consists of 80 values (corresponding to 64 carriers plus the cyclic prefix), up to 800 OFDM symbols could be stored at the same time in one single FIFO.

Speed enhancements are achieved thanks to using the *Free-Running Clock* mode for hardware simulation, which allows the FPGA to operate asynchronously with respect to the Simulink simulation. This means that the FPGA is not kept in lockstep with the simulation. In our experiments, the vehicular emulator clock period was 100 ns (10 MHz), while Simulink's system period was one second, what implies that the FPGA was running much faster.

Note also that the size of the shared memories can be reduced to match the simulation requirements. For instance, when evaluating the performance in Section 5.10, simulations with 64-byte packets were launched, hence the input/output memories were adapted to such amount of data. In these simulations, when all the memory requirements were taken into account, only between 19% and 24% of the FIFO16/RAMB16 memory blocks were occupied (see Table 5.2).

Moreover, the fact of transmitting just one 64-byte packet at each time becomes an advantage when averaging BER (Bit Error Rate) and PER (Packet Error Rate) during performance measurements. In such case, the channel coefficients of two consecutive packets are less correlated than when sending the same two packets together in the same frame, because while the transmitter produces the second packet, the emulator keeps generating coefficients (with a period that is 10^6 times the transmitter's period).

The data exchange process can be summarized as follows:

1. The configuration parameters of the channel to emulate are set in registers that are read by the FPGA.

2. The transmitter generates the packets to be sent and creates a single data frame with all of them.

3. The data frame to transmit is placed in a shared FIFO that is read by the vehicular emulator.

4. The emulator applies the channel coefficients to the transmitted signal and sets the output values in shared FIFOs.

5. The output shared FIFOs are read by the IEEE 802.11p software receiver that processes the transmitted packets.

5.10 Experiments: Evaluating Vehicular SISO Transceiver Performance

5.10.1 Evaluation of the IEEE 802.11p Physical layer over Vehicular Channels

To evaluate the performance of the software implementation of the IEEE 802.11p physical (PHY) layer described in Section 4.2.2.1, the signal from the transmitter was passed through the FPGA-based vehicular channel emulator. The measurements were performed using the *co-simulation mode*: the transmitter and the receiver were implemented in MATLAB and Simulink, but the channel emulator ran on an FPGA.

Fig. 5.9 and Fig. 5.10 depict, respectively, the BER and PER for the six vehicular channels presented in Section 5.3. The curve for the AWGN channel has been also added as a reference of the transceiver performance. In order to achieve a fair comparison, the same transmission parameters are set for the IEEE 802.11p transceiver, changing only the channel for each curve. A rate 1/2 FEC (Forward Error Correction) code was used, each of the carriers was QPSK modulated, and a maximum of 10,000 64-byte data packets were averaged (the simulation stops for each E_b/N_0 value when 100 erroneous packets are detected). The receiver assumed perfect time synchronization, pilot-aided channel estimation was performed and an MMSE (Minimum Mean Square Error) linear equalizer was applied, as described in Section 4.2.2.1.

Notice that (54) states that a vehicular communications system must be capable of transferring messages to and from vehicles at speeds of 193 Km/h with a PER of less than 10% for 64-byte data packets. Although the assumed speeds of the six implemented vehicular channel models are 140 Km/h or 120 Km/h (depending on the channel), the minimum E_b/N_0 to reach the 10% PER threshold constitutes a good reference of the system performance. Table 5.5 summarizes the required minimum E_b/N_0 to reach a PER of 10%. The values fluctuate between 10.1 dB a 18.2 dB, which are in an acceptable range when operating in a real scenario.

To justify these results, it is useful to compare them with the conclusions of (60). There the authors affirm that PER generally decreases with decreasing Doppler offsets and widths, and increasing K factors. Using Table 5.1 and Table 5.5, it is possible to corroborate this assertion. On the one hand, if we sort the channels by their E_b/N_0

necessary to reach the 10% PER threshold and by their Rician K (using the *Overall K Factor* for *VTV-Expressway Same Direction with Wall*), we obtain the same ranking. This allows us to conclude that the larger the Rician K factor (clearer line-of-sight), the smaller E_b/N_0 is needed to reach the threshold. On the other hand, regarding the Doppler offset, it can be seen that the vehicular channel with the largest Doppler offset (*VTV-Expressway Oncoming*) is one of the channels that require more E_b/N_0 to reach the 10% threshold. However, in other channels there is not a clear impact in the E_b/N_0 threshold. This is the case of *VTV-Urban Canyon Oncoming* which owns large Doppler shifts, fading and LOS offsets. This latter case allows stating that the degree of line-of-sight is a critical factor since it shadows other influential factors like the mentioned frequency offsets.

Finally, it is important to stress that in a real transceiver there are other sources of error (i.e. RF impairments, synchronization errors, channel estimation mismatches...) which have not been added to the software transceiver but that could be easily implemented. In that sense, the exposed IEEE 802.11p evaluation framework represents a very useful tool for transceiver designers, who can isolate the sources of error and analyze them while assessing the performance in a realistic scenario.

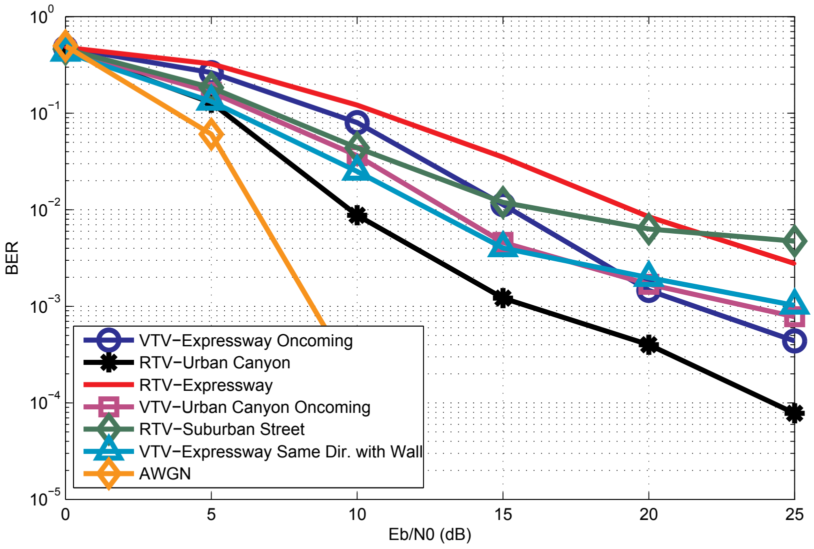

Figure 5.9: BER of the IEEE 802.11p transceiver for the different SISO vehicular channels.

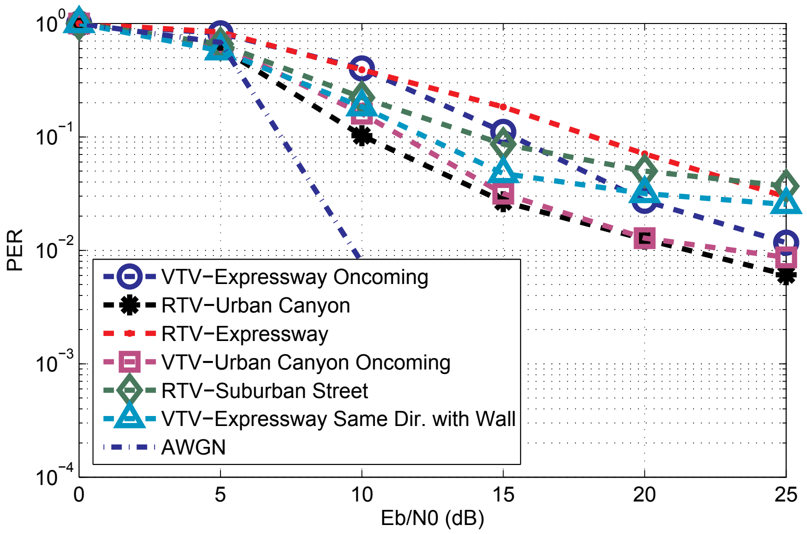

Figure 5.10: PER of the IEEE 802.11p transceiver for the different SISO vehicular channels.

Table 5.5: Minimum E_b/N_0 to obtain a PER of 10% in the SISO vehicular channels.

Vehicular Channel	E_b/N_0 (dB)
VTV-Expressway Oncoming	15.36
RTV-Urban Canyon	10.10
RTV-Expressway	18.20
VTV-Urban Canyon Oncoming	11.45
RTV-Suburban Street	14.22
VTV-Expressway Same Direction with wall	12.25

5.10.2 Observed Inter-Carrier Interference (ICI) in the Implemented Vehicular Channels

The use of a one-tap equalizer as described in Section 4.2.2.1 was intentional. Due to the Doppler effect, it was very likely that ICI appeared in the received signal, but the practical results for these channel models and the assumed transmission conditions (i.e. no synchronization errors or RF impairments) have shown that there is almost no ICI at the receiver side. A simple explanation follows: let us first assume that the OFDM signal, after the FFT (Fast-Fourier Transform), is modeled as

$$\mathbf{y}_f = \mathbf{F}\mathbf{y}_t = \mathbf{F}\mathbf{H}_t\mathbf{F}^H\mathbf{s}_f + \mathbf{F}\mathbf{n}_t = \mathbf{H}_f\mathbf{s}_f + \mathbf{n}_f$$

where $\mathbf{y}_t$ is the received signal vector, $\mathbf{F}$ and $\mathbf{F}^H$ are matrices that perform the FFT and the IFFT respectively (^{H}denotes the Hermitian or conjugate transpose), $\mathbf{H}_t$ is the matrix with the channel coefficients in the time domain, $\mathbf{s}_f$ contains the transmitted symbols, and $\mathbf{n}_t$ and $\mathbf{n}_f$ represent vectors with Gaussian noise. In this model, the receiver recovers the transmitted symbols by estimating the matrix $\mathbf{H}_f = \mathbf{F}\mathbf{H}_t\mathbf{F}^H$.

The presence of ICI depends on the characteristics of $\mathbf{H}_f$. If this matrix is diagonal, the signal received in each subcarrier depends on the transmitted symbol and only on one channel coefficient and, therefore, there will be no ICI. However, if any non-diagonal coefficient of $\mathbf{H}_f$ is different from zero, ICI will appear.

For the implemented vehicular channel models, it could be observed that $\mathbf{H}_f$ was *almost* diagonal. As a specific measure of this observation, it was considered the ratio between the energy of the coefficient in the diagonal and the sum of the energies of the rest of the values in the same row of the channel matrix. For the i-th row of one sample of the channel matrix, this ratio can be obtained as

$$q_i = \frac{|h_{ii}|^2}{\sum\limits_{\substack{j=1\\ j\neq i}}^{N} |h_{ij}|^2}$$

where h_{ij} are the coefficients of $\mathbf{H}_f$ and N is the length of the FFT ($N = 64$ for IEEE 802.11p). This ratio as averaged for 10,000 channel matrices. It is important to stress that the channel coefficients were obtained directly from the emulator, i.e. they were not estimations.

Table 5.6: Energy ratios for the SISO vehicular channels.

Vehicular Channel	**Average Energy Ratio**
VTV-Expressway Oncoming	2706.4
RTV-Urban Canyon	1151.4
RTV-Expressway	1746.3
VTV-Urban Canyon Oncoming	2769.6
RTV-Suburban Street	770.3
VTV-Expressway Same Direction with wall	940.1

Table 5.6 shows the average energy ratios for the six vehicular channels. Such Table confirms that, on average, the energy of the diagonal element is at least 700 times higher

than the sum of the energies of the rest of the coefficients in the same row. With such difference, the contribution to the ICI of the non-diagonal coefficients, though existent, can be ignored and, hence, a simple one-tap equalizer is enough to compensate the channel effects.

As said above, the considered emulation setup only accounted for channel estimation errors. In this case, the use of a one-tap equalizer was valid for overcoming the effects of the implemented vehicular channels. When testing real transceivers, however, additional sources of error appear (e.g. synchronization errors, RF impairments...). In addition, the considered vehicular channel models do not cover all the situations that may appear in real operation. Thus, transceivers designed to operate in real environments should, of course, include more sophisticated equalizers to compensate larger amounts of ICI.

5.10.3 Importance of Coding over Vehicular Channels

To show the importance of using coding when transmitting over vehicular channels, two different channels were chosen to have the performance of the IEEE 802.11p transceiver measured. Transmissions were carried out using the same configuration as described in Section 5.10.1.

Fig. 5.11 and Fig. 5.12 show, respectively, the BER and PER for *RTV-Expressway* and *VTV-Expressway Same Direction with Wall* when transmitting with and without coding. In both channels a PER of 10% is achieved with an E_b/N_0 of more than 25 dB when coding is not used. Notice that performance is dramatically improved with the utilization of coding. The BER curves of the experiments that use coding remain, at low E_b/N_0 values, above the non-coded versions due to the generation of new errors when decoding. Once the decoder reaches a BER of a little less than a 10%, the decoder is able to correct errors and the versions with coding outperform the non-coded versions.

5.10.4 IEEE 802.11p Vs IEEE 802.11a in Vehicular Channels

In this subsection the performances of the PHY layers of IEEE 802.11a and IEEE 802.11p are compared. For a comparison of these transceivers against IEEE 802.16e, see Section 5.10.5.

As it was mentioned in Section 4.2, the main difference between IEEE 802.11p and IEEE 802.11a the physical level is the bandwidth: IEEE 802.11a uses 20 MHz, while

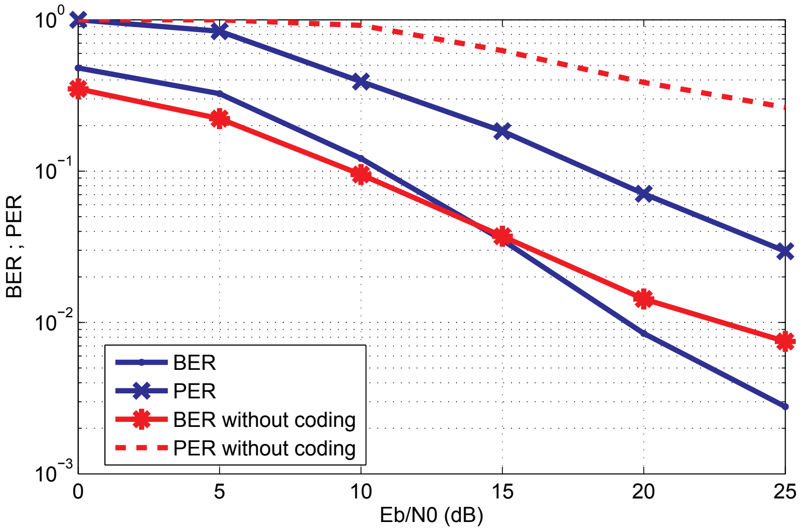

Figure 5.11: BER/PER when transmitting with and without coding through *RTV-Expressway.*

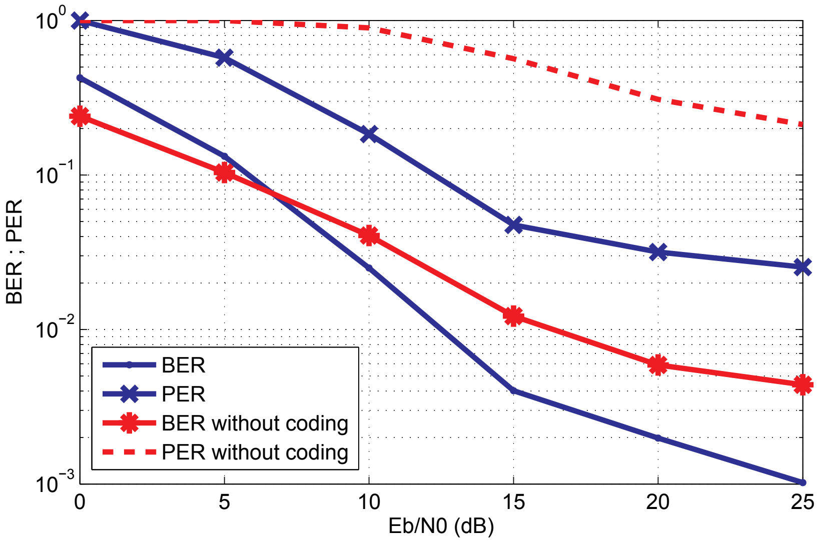

Figure 5.12: BER/PER when transmitting with and without coding through *VTV-Expressway Same Direction with Wall.*

IEEE 802.11p only 10 MHz. The reduction in bandwidth translates into larger OFDM symbols and halves the transmission rates.

Fig. 5.13 shows a good example of the advantages of using IEEE 802.11p when transmitting through vehicular channels. In the channels chosen, the IEEE 802.11a transceiver performs worse and experiences error floors, starting at an E_b/N_0 of 15 dB.

Furthermore, if we obtain the 10% PER thresholds for the IEEE 802.11a transceiver, we can see that in *RTV-Urban Canyon* it requires 12.87 dB (while IEEE 802.11p only needs 10.1 dB). In *VTV-Expressway Oncoming*, the IEEE 802.11a transceiver never reaches the 10% threshold. This difference is due to the high delay spread of the vehicular channels that introduce interference among symbols in the case of IEEE 802.11a. Therefore, there exists an important improvement in using IEEE 802.11p instead of IEEE 802.11a when transmitting over vehicular environments.

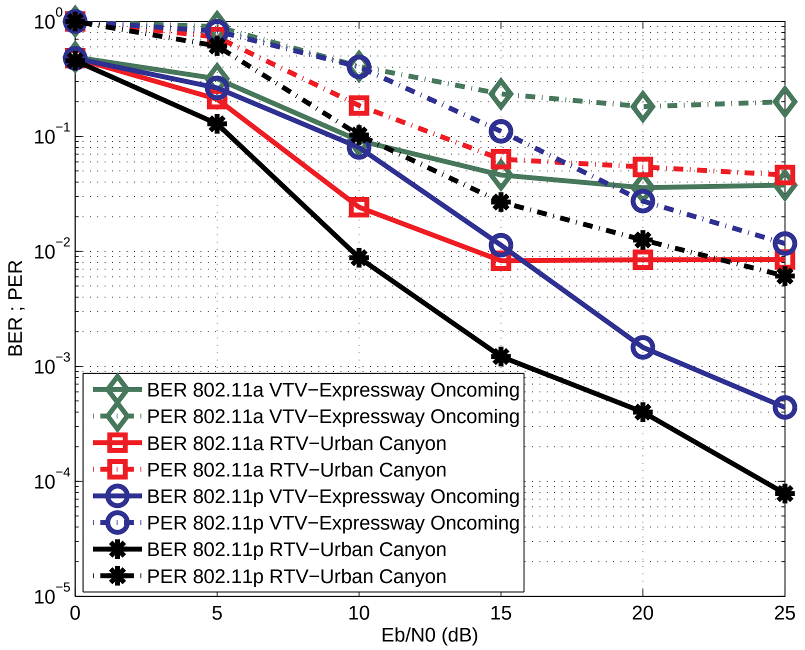

Figure 5.13: BER/PER comparison between IEEE 802.11p and IEEE 802.11a when transmitting over the channel emulator.

5.10.5 IEEE 802.11p Vs IEEE 802.11a Vs IEEE 802.16e

In this subsection the IEEE 802.16e is added to the previous performance comparison. To achieve a fair comparison all the transceivers set the same transmission parameters (previously indicated in Section 5.10.1). The configuration parameters of the transceivers are shown in Table 4.1. However, for the sake of fairness, instead of comparing performances in terms of PER, it was obtained the FEC block Error Rate (FER) when all the transceivers made use of the same FEC block size. Thus, a maximum of

100,000 48-bit FEC blocks were averaged for different E_b/N_0 values (the simulation stopped for each E_b/N_0 value when 100 erroneous FEC blocks were detected).

5.10.5.1 Performance over AWGN and Rayleigh Fading Channels

In order to obtain a performance reference for the implemented transceivers, they were evaluated and compared over two non-vehicular environments. Fig. 5.14 and Fig. 5.15 show, respectively, the transceivers' performance over an AWGN channel and a frequency-flat Rayleigh block fading channel whose coefficients were constant during 15 FEC blocks (i.e. one Mobile WiMAX slot).

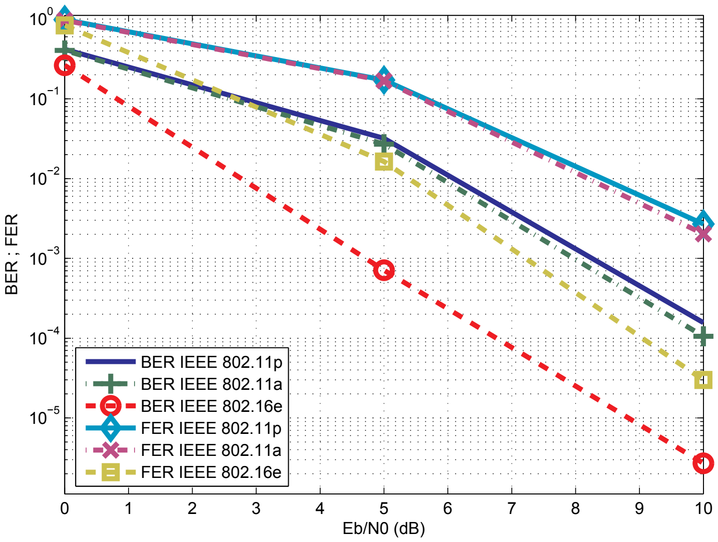

Figure 5.14: Performance comparison of IEEE 802.11p, IEEE 802.11a and IEEE 802.16e transceivers when transmitting over an AWGN channel.

For both channels, IEEE 802.11p and IEEE 802.11a produce roughly the same results. This behavior was to be expected since the transceiver is the same in all aspects apart from the bandwidth. However, the IEEE 802.16e transceiver yields much better results, especially in the AWGN channel, even though the transmission parameters are the same in all transceivers. This is because channel estimation is far more accurate in the case of the IEEE 802.16e transceiver. Indeed, Fig. 5.16 shows the Mean Squared Error (MSE) between the estimated and the true channel for the IEEE 802.11p (the same results were obtained with IEEE 802.11a and are therefore omitted) and IEEE

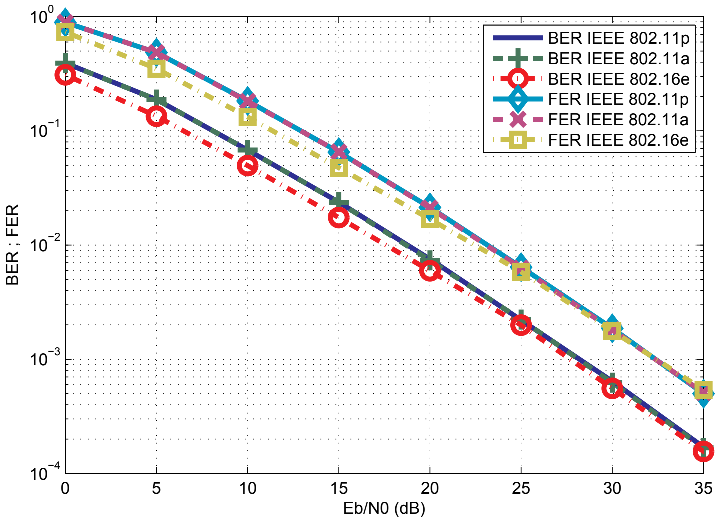

Figure 5.15: Performance comparison of IEEE 802.11p, IEEE 802.11a and IEEE 802.16e transceivers when transmitting over a block-fading Rayleigh channel.

802.16e transceivers when considering an AWGN channel. IEEE 802.16e better estimates the channel thanks to the use of sixty pilots (i.e. for 360 data subcarriers one pilot is used for each group of six data subcarriers), while IEEE 802.11p makes use of only four pilots (since there are 48 data subcarriers, there is one pilot for each group of twelve data subcarriers).

5.10.5.2 Performance over Vehicular Channels

Figs. 5.17 to 5.22 depict the BER and FER curves for the three transceivers when transmitting over the six vehicular channels described in Section 5.3. In general, it can be observed that the IEEE 802.16e transceiver produces better results (both in terms of BER and FER) than IEEE 802.11p, while the IEEE 802.11a transceiver obtains the worst global results. Also, notice that due to the limited coherence time of the vehicular channels, there is always a residual channel estimation error which produces the error floor that can be observed in all figures. This is in contrast to the performance curves corresponding to the static frequency-flat channels used in Figs. 5.14 and 5.15, where there are no error floors.

In urban environments (channels *VTV-Urban Canyon Oncoming* and *RTV-Urban Canyon*) Mobile WiMAX outperforms IEEE 802.11p/a in terms of BER and FER for

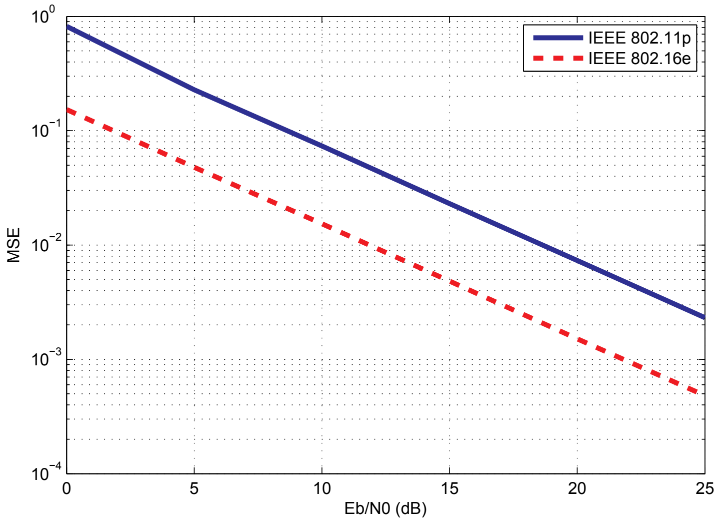

Figure 5.16: Channel estimation error of the IEEE 802.11p and IEEE 802.16e transceivers when transmitting over an AWGN channel.

E_b/N_0 values below 20 dB (see Figs. 5.17 and 5.18). Also, notice that Mobile WiMAX requires the lowest E_b/N_0 values to reach a target FER of 10%.

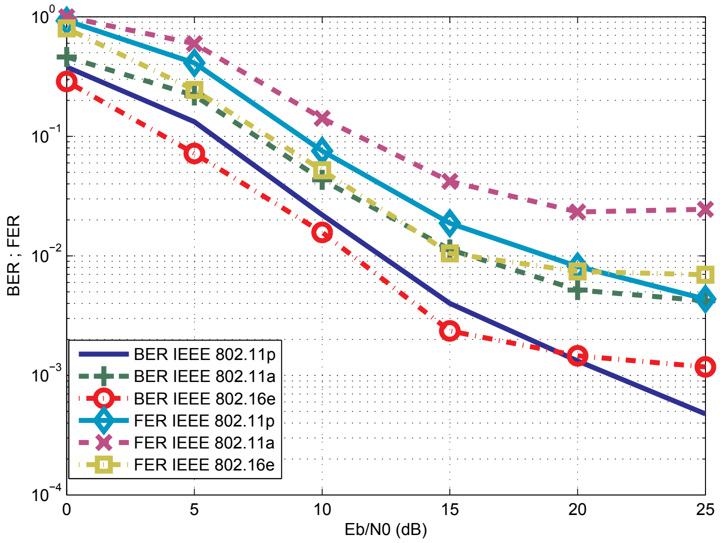

Figure 5.17: Performance comparison of IEEE 802.11p, IEEE 802.11a and IEEE 802.16e when transmitting over *VTV-Urban Canyon Oncoming*.

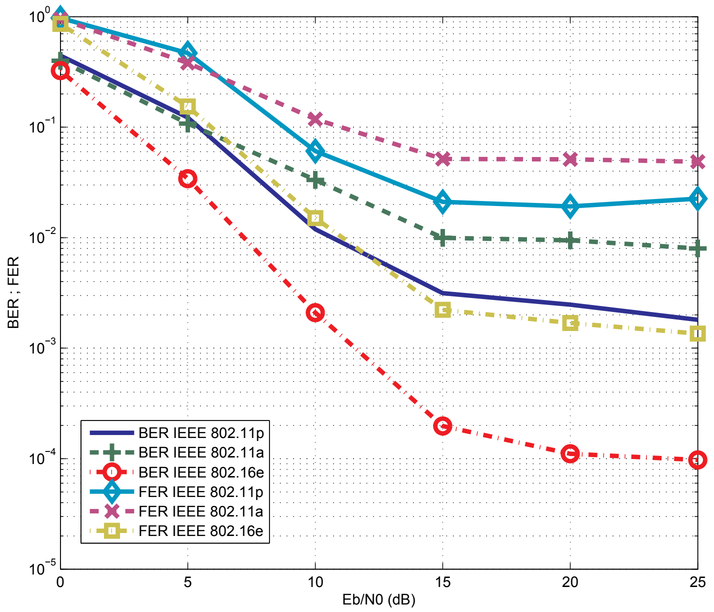

Figure 5.18: Performance comparison of IEEE 802.11p, IEEE 802.11a and IEEE 802.16e when transmitting over *RTV-Urban Canyon*.

In surface streets (*RTV-Suburban Street*) Mobile WiMAX also performs better than the other standards (see Fig. 5.19). For instance, to reach an FER of 10% Mobile WiMAX requires 8.6 dB, while IEEE 802.11p and IEEE 802.11a need, respectively, 11.1 dB and 14.6 dB.

In expressways (*VTV-Expressway Oncoming*, *RTV-Expressway*, *VTV-Expressway Same Direction With Wall*) the results depend on the channel. In *VTV-Expressway Oncoming* (see Fig. 5.20) IEEE 802.11p clearly outperforms Mobile WiMAX at both low and high values of E_b/N_0. IEEE 802.11p and Mobile WiMAX both exhibit a similar behavior when considering *RTV-Expressway* channels (see Fig. 5.21): Mobile WiMAX is slightly better than IEEE 802.11p for low E_b/N_0 values while the situation reverses for high E_b/N_0 values. Finally, in the case of *VTV-Expressway Same Direction With Wall*, Mobile WiMAX clearly obtains a major gain over IEEE 802.11p/a: for example, it requires an E_b/N_0 of 6.2 dB less than that of IEEE 802.11p (7.6 dB vs 13.8 dB) to obtain an FER of 10% (see Fig. 5.22).

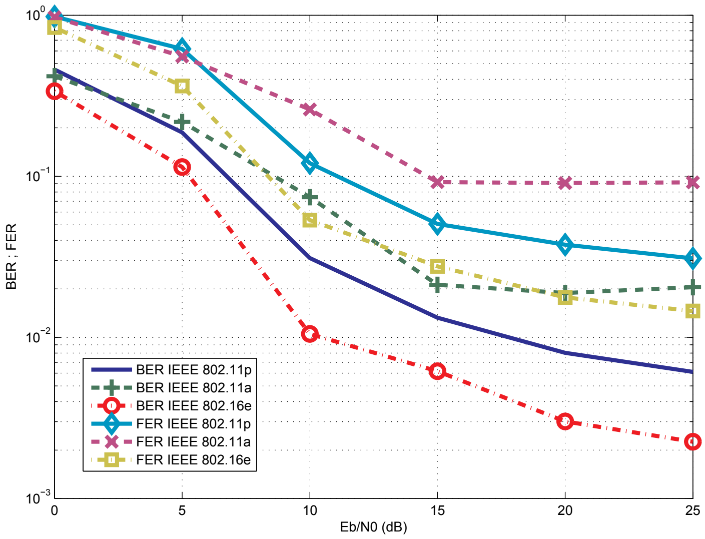

Figure 5.19: Performance comparison of IEEE 802.11p, IEEE 802.11a and IEEE 802.16e when transmitting over *RTV-Suburban Street*.

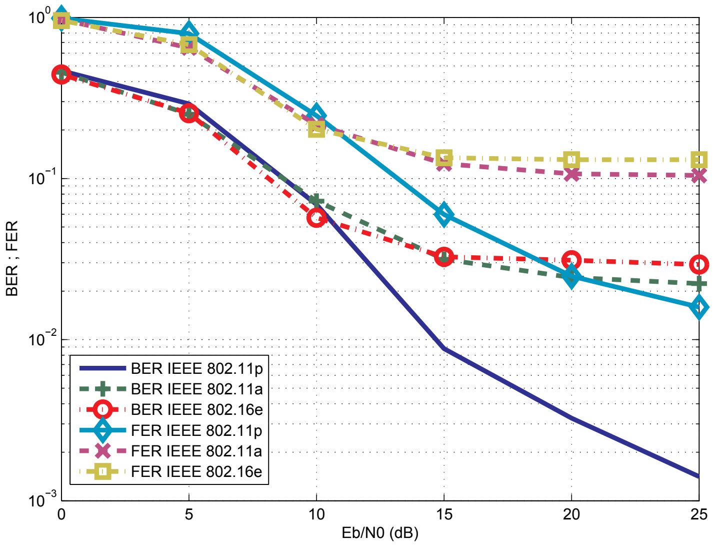

Figure 5.20: Performance comparison of IEEE 802.11p, IEEE 802.11a and IEEE 802.16e when transmitting over *VTV-Expr. Oncoming*.

5.10.5.3 Discussion: IEEE 802.16e or IEEE 802.11p?

The BER/FER versus E_b/N_0 curves depicted indicate that the PHY Layer of Mobile WiMAX outperforms that of IEEE 802.11p in most of the reference channel models

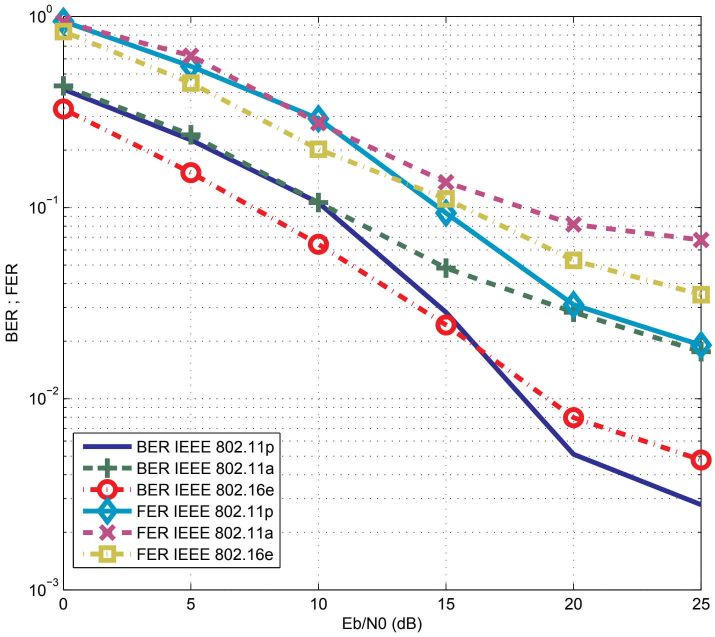

Figure 5.21: Performance comparison of IEEE 802.11p, IEEE 802.11a and IEEE 802.16e when transmitting over *RTV-Expressway*.

used as benchmarks in vehicular communications. The only clear exception is the performance over the *VTV-Expressway Oncoming* channel where the performance of IEEE 802.11p is much better than that of Mobile WiMAX. For the *RTV-Expressway* channel, the PHY Layer performance of both standards is quite similar. In the remaining four scenarios the PHY Layer of Mobile WiMAX is considerably better than that of IEEE 802.11p.

An explanation of this behavior is the superior robustness to high channel delay spreads of the Mobile WiMAX PHY Layer. In IEEE 802.11p, assuming a bandwidth of 10 MHz and 64 subcarriers, a 1/4 cyclic prefix will lead to a guard time of 1.6 μs. In the case of Mobile WiMAX, a transceiver that uses 10 MHz of bandwidth and 512 subcarriers has a 1/4 cyclic prefix that lasts 12.8 μs. Thus, the OFDM symbols used in Mobile WiMAX can equalize channels with a larger delay spread.

Another advantage of the Mobile WiMAX PHY Layer is that the maximum data rate that can be reached is 39.9 Mbits/s, while this value is only 27 Mbits/s in the case of IEEE 802.11p. Regarding bandwidth, both PHY Layers admit 10 MHz, but Mobile

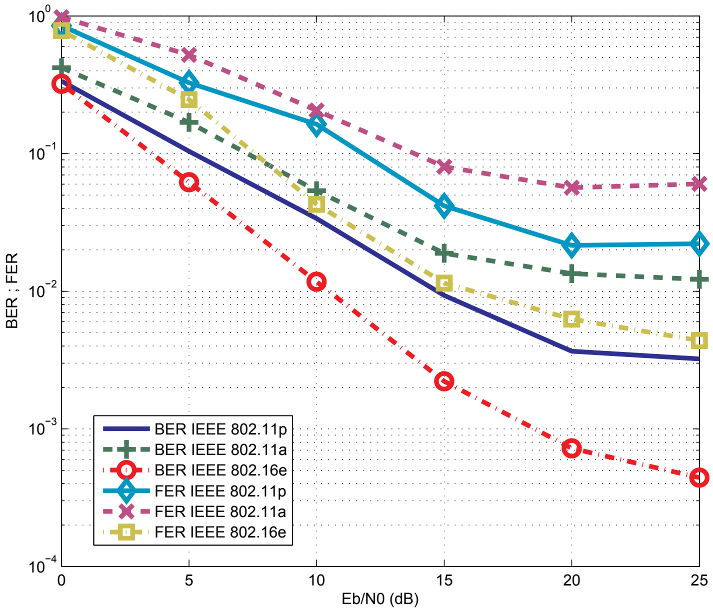

Figure 5.22: Performance comparison of IEEE 802.11p, IEEE 802.11a and IEEE 802.16e when transmitting over *VTV-Expressway Same Direction with Wall.*

WiMAX shows more flexibility by offering different bandwidth sizes. Depending on the band class, a bandwidth of 3.5, 5, 7, 8.75, 15 or 20 MHz can be selected, it thus being more suitable for adapting its transmission to the various radio spectrum regulations that exist all around the world.

On the other hand, it should be mentioned that the PHY layer of IEEE 802.11p supports larger vehicle speeds. Indeed, IEEE 802.11p is designed to transmit 1,000 byte data packets with an FER lower than 10% at a maximum Doppler shift of ± 2,100 Hz, what means that a top speed of roughly 385 Km/h can be reached when using the 5 GHz band (the transmitter and receiver would drive at almost 193 Km/h). In contrast, the WiMAX Forum mobility profile requirements (67) specify that each Mobile WiMAX terminal should be able to exchange data at a maximum speed of 120 Km/h (which already coincides with the maximum permitted traveling speed in many countries). Therefore, it can be stated that, theoretically, IEEE 802.11p is more appropriate for high speed environments. This explains the superior performance of IEEE 802.11p over the *VTV-Expressway Oncoming*, which is the channel with maximum Doppler shift (see

Table 5.1).

Finally, notice that these experiments are focused on the PHY-layer performance of the standards analyzed. A final decision about which is the best standard for a specific vehicular network will depend on many other factors that are beyond the scope of this work, such as infrastructure deployment costs, support of the automotive industry, energy consumption, handover efficiency, regulations of the radio spectrum, higher layer constraints and so on. However, the results presented herein provide a set of PHY-Layer metrics that can be used as the PHY model for a network simulator, which in turn can be used to carry out analyses of the whole vehicular network.

5.11 Conclusions

In this Chapter it was described a flexible, reconfigurable and cost-effective solution for evaluating wireless transceivers through the real-time emulation of vehicular channels. After presenting a comprehensive review of the current state of the art of real-time vehicular channel emulators, it was detailed the design and implementation of a vehicular channel emulator that is based on the use of FPGA technology and rapid prototyping software tools. The channel emulator considers six different scenarios that include common environments such as highways, urban canyons and suburban areas.

Several examples of performance evaluation of the IEEE 802.11p, IEEE 802.11a and IEEE 802.16e PHY layers were obtained when transmitting over the six high-speed scenarios. It was also studied the effects of using coding, which obtains important gains in BER/PER, and the performance of the three transceivers was compared in Rayleigh and AWGN channels.

The results obtained in non-vehicular channels illustrate the importance of obtaining a good estimation of the channel coefficients: Mobile WiMAX attains better results than IEEE 802.11p and IEEE 802.11a thanks to doubling the number of pilots per data subcarrier.

Regarding the vehicular measurements, it can be concluded that:

- The use of IEEE 802.11p improves dramatically the performance respect to IEEE 802.11a.

- In the chosen experimental conditions, Mobile WiMAX outperforms IEEE 802.11p and IEEE 802.11a in most vehicular channels, although further research on non-PHY layer related factors should be performed to determine which standard is better for each specific vehicular scenario.

6

Rapid Prototyping for MIMO Vehicular Channel Emulators

6.1 Introduction

As it was mentioned in previous Chapters, there are basically two sorts of vehicular applications: those dedicated to providing safety services and those that do not. The former require to exchange messages fast in order to obtain a swift reaction from the car or the driver in dangerous situations, like when a sudden braking occurs or when two cars approach an intersection. Due to the needed speed, transceivers tend to send short packets and therefore small bandwidths are demanded. IEEE 802.11p (17) is probably the best positioned standard for providing safety services since it has been explicitly optimized for such kind of communications.

On the contrary, non-safety services do not have so tight time restrictions and usually require higher data transfer rates. Non-safety applications include, for instance, mobile internet access, roadsign recognition or travel information management. If IEEE 802.11p has to provide such kind of services in vehicular environments, its data transfer rates will have to be somehow improved in order to compete with standards like IEEE 802.16e or LTE.

One of the best ways to increase the transmission capacity consists in using multiple antennas at transmission (MISO systems), reception (SIMO systems) or both at transmission and reception (MIMO systems) (35, 36). IEEE 802.11p was initially devised as a single-antenna (SISO) system, but as it is an amendment to the IEEE 802.11 based on IEEE 802.11n-2009 (among others), the possibility of applying multiple-antenna techniques in IEEE 802.11p could be easily carried out in future releases of the standard through a new amendment.

To assess the performance of a multiple-antenna IEEE 802.11p transceiver it is desirable to use software channel emulation, which is widely used by researchers when the transceiver whose performance is being assessed is also software. The price to be paid is that software simulations may take a long time if the channel to be emulated consists of several paths for each receive/transmit antenna, as occurs in realistic wireless MIMO channel models. Moreover, researchers usually have to average the results simulating the transmission and reception of thousands or millions of data packets in order to obtain valid conclusions. Additionally, such averaging process requires using loops that some development environments (e.g. MATLAB/Simulink) do not handle fast, leading to slow performance evaluations.

As it could be seen in Chapter 5, the concept of software-hardware co-simulation suggests a new hybrid approach: the transceivers can be implemented in software, giving researchers absolute configuration control and flexibility, whereas the channel emulation can be run on hardware, thus accelerating channel emulation and the overall simulation process. Obviously, the time required for developing a hardware emulator is larger than in the software-based case, but if the channel emulation task has to run an extremely high number of times (as required for a proper evaluation of the performance of a transceiver), the time savings related to simulation eventually compensate for the development time.

These reasons motivated the development of two different channel emulators that follow this software-hardware co-simulation approach: they consider wireless communication testbeds where the transceivers (described in Chapter 4) are entirely software-based but the channel emulators run on an FPGA. Like in Chapter 5, in order to reduce the development time, rapid-prototyping tools were used, both for software (MATLAB/Simulink) and hardware (Xilinx System Generator), which allowed building and refining the channel emulator really fast in comparison with other traditional tools (e.g. hardware description languages).

In Chapter 5 it was assessed transceiver performance in co-simulation mode when different vehicular communication transceivers operated over six scenarios that include highways, urban canyons and suburban areas. In this Chapter, we will depart from the implementations described in such Chapter and we will see how to upgrade the whole system with the goal of carrying out performance comparisons for multiple-antenna transceivers. Furthermore, it will be presented a second MIMO 2 × 2 channel emulator based on novel channel parameters recently proposed in (115). Thanks to the exposed FPGA-based channel emulators it will be possible to reduce the simulation time required to obtain performance measurements between 6 and 209 times with respect to a conventional entirely software-based simulation approach.

This Chapter is structured as follows. Section 6.2 gives an extensive overview of latest vehicular channel models and multiple-antenna channel emulators. Section 6.3 describes the design and implementation of two different channel emulators able to recreate up to seven different vehicular scenarios. Finally, Section 6.4 is dedicated to expose a performance comparison of multi-antenna IEEE 802.11p transceivers whereas Section 6.5 is devoted to the conclusions.

Like in the previous Chapter, this one will be mainly focused on describing the implementation tasks performed to develop the FPGA-based channel emulators (highlighted in Fig. 6.1).

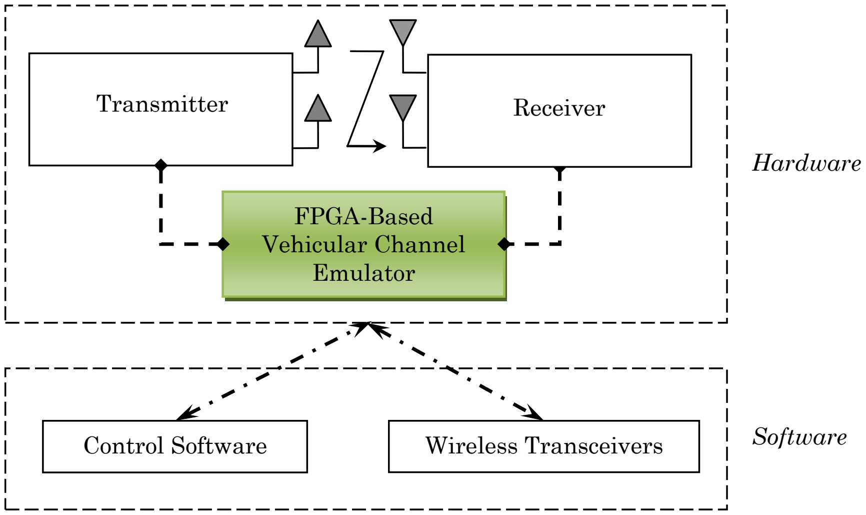

Figure 6.1: In color: components of the rapid prototyping system that are described in Chapter 6.

6.2 Background

6.2.1 Mobile and Vehicular Channel Models

In recent years several research papers have presented mobile and vehicular channel models that try to reflect the continuously changing conditions of the environment (60, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133). Most of these models have been developed for SISO transceivers (60, 116, 117, 118, 120, 121, 123, 124, 127, 128, 130, 132, 133), while a few are specific for multiple-antenna systems (119, 122, 125, 126, 129, 131). These vehicular models can be also classified depending on the way they were obtained, distinguishing physical models (116, 117, 118, 124, 125, 126, 129, 130, 131, 132, 133), empirical models (60, 121, 122, 123, 127, 128) and models that mix empirical measurements and physical developments (119, 120).

Physical models characterize an environment by analyzing the propagation of electromagnetic waves between a transmitter and a receiver. Such models can be very sophisticated and usually require indicating several parameters in order to reproduce accurately the propagation in a specific scenario. Moreover, this kind of models does not depend on the characteristics of the antenna array (number of antennas, polarization, etc.) or the system bandwidth.

Geometry-based stochastic channel models (GSCM) are probably the most popular physical channel models. A good example is (126), which describes a three-dimensional wideband channel model for MIMO transceivers that carry out mobile communications. After describing the reference model, the authors compare the theoretical results obtained with empirical MIMO vehicle-to-vehicle measurements performed in surface streets and highways, concluding that there is a close agreement between both of them.

In regard to empirical vehicular models, it can be observed that most of them have been obtained for the 5 GHz band (60, 119, 120, 121, 122, 127, 128), although there are also models for the 2.4 GHz band (e.g. (123)). Vehicular communications will mostly take place in the 5 GHz band: in 1999, the DSRC spectrum band, a band of 75 MHz at 5.9 GHz, was allocated in the United States, whereas, an equivalent band was allocated in Europe in August 2008 by the European Telecommunications Standards Institute (ETSI) (such entity reserved 30 MHz of spectrum in the 5.9 GHz band for ITS). The channel emulators presented in this Chapter cover both the 5 GHz frequency band (in Section 6.3.1) and 2.4 GHz (in Section 6.3.2).

Empirical models are created by analyzing measurements obtained when transmitting over a real environment and then modeling the channel characteristics. Sometimes researchers are not satisfied with the accuracy of their empirical model and they add analytical developments (such as the ones performed for physical channel models), creating mixed models. For instance, in (119) it is exposed a wideband MIMO vehicular channel model based on 5.2 GHz measurements performed in highways and rural areas, but due to the non-stationarity of the channel statistics, the authors decided to apply a GSCM instead of the classical tapped-delay line model. To assess the model accuracy, the authors compare theoretical and empirical antenna correlations, obtaining a good similarity between both.

From all the above-mentioned vehicular models, two empirical models were chosen to be implemented: the ones described in (115) and (60). Empirical models may not be as accurate as physical models, but they are in general far less complex and they do not involve as many parameters as physical models do. Instead of accuracy empirical models look for the ability to recreate the specific environment where a measurement campaign has been carried out. Due to this reason, many of the latest wireless communication standards, such as IEEE 802.16e, IEEE 802.11n or 3G/beyond-3G telephony, have proposed empirical models for the evaluation of their transceivers (98, 134, 135).

Finally, note that the channels modeled in (60) are based on SISO measurements, but they are used because they have become a good reference for evaluating IEEE 802.11p transceivers. Further investigation is still needed to adapt such channels to multiple-antenna environments, but when that occurs, the transceivers and the channel emulator models presented in this Chapter will continue to be valid, only requiring slight modifications or no modifications at all.

6.2.2 MIMO Channel Emulators

There are two kinds of MIMO emulators: commercial and academic. On the one hand, there are many commercial MIMO emulators manufactured by companies such as Azimuth Systems (107), Agilent (108), Rhode & Schwarz (106), Spirent (105) or Propsim (136). Most of their MIMO emulators are general-purposed (for instance, Spirent's SR5500, Rhode's AMU200A, Propsim's F8 MIMO OTA or the ACE MX family from Azimuth), although there are some aimed at evaluating specific technologies, like Azimuth's ACE-400WB/Wi-Fi (that includes the TGn channel models (134) for evaluating IEEE 802.11n transceivers) or Agilent's N5106A PXB MIMO Receiver Tester (with built-in LTE and MobileWiMAX channels).

Although commercial emulators work great in most situations, they are expensive and suffer from a clear lack of flexibility when used by researchers to study performance in state-of-the-art wireless channels. The lack of flexibility is mainly in regard to the configuration of non-standard characteristics of the channel (i.e. the number of paths and taps, the fading spectral shapes, the fading Doppler, etc.). Additionally, another important issue arises in the case of vehicular channels: several well-known researchers (115) state that the main characteristics of a VTV (Vehicle-To-Vehicle) channel resides in the non-constant characteristic of the K-factor, which current channel emulators do not support.

On the other hand, apart from commercial products, different authors have described how they built their own channel emulators. Like in the SISO case detailed in Chapter 5, among all the available technologies, FPGAs are the most convenient since they can deal with more complex designs and they own specific resources (such as counters or arithmetic operator blocks) that are more adequate for implementing a channel emulator. In fact, FPGAs are probably the most popular hardware technology for developing academic channel emulators.

Examples of academic MIMO FPGA-based channel emulators are described in (137, 138, 139, 140, 141, 142, 143, 144). Some of them are generic (137, 138, 139, 142), while others (140, 141, 143, 144) are specifically oriented towards the implementation of the IEEE 802.11n channels (134). It is important to point out that, as of writting this Thesis, none of the studied channel emulators have been explicitly developed for recreating SISO/MIMO VTV or RTV (Road-To-Vehicle) environments.

One of the main problems when implementing MIMO channel emulators in an FPGA is that they require large designs and, therefore, the use of resources has to be optimized. Although most of the studied channel emulators are able to implement the whole system into only one FPGA, there are examples of channel emulators that distribute computing among different FPGAs (144). Furthermore, to fit the design into one FPGA, researchers have to save resources using different clever tricks, being one of the most recurrent the offline generation of the channel coefficients (137, 140, 141, 143). Also, some authors (141) achieve to save up to 67% of the FPGA resources by applying the channel coefficients in the frequency domain.

At least three drawbacks can be easily detected in the above-mentioned academic developments. The first two issues have been already cited when analyzing SISO channel emulators in Chapter 5, while the third one is specific for MIMO emulators.

First, the use of low-level description languages such as VHDL results in slow development stages. Although in most cases VHDL is able to obtain resource-efficient FPGA designs, programming can become a cumbersome task that may consume a large amount of time and economic resources. There are new sophisticated tools such as Xilinx System Generator that permit the use of high-level blocks, enabling to build complex designs easier and faster. However, it must be noted that although rapid-prototyping tools increase development speed, they usually generate non-optimized large designs that may not fit into the FPGA. Hence, for large designs, optimizations must be performed.

The second problem of the emulators cited above is related to the portability of the channel emulator. An ideal channel emulator should be able to work in stand-alone mode, without needing external devices to generate and transfer channel coefficients to the FPGA. In the reviewed literature PCs are commonly used to generate coefficients (137, 140, 141), although some authors have been able to build more compact systems by using other devices (for instance, in (143) coefficient generation is performed by an ARM9 processor). Moreover, there is a bottleneck in the number of coefficients that can be transferred from the external device to the FPGA (either due to restrictions on the external device or on the communication buses), what leads to important limitations. For example, in (143) the externally generated coefficients are stored into a ZBTRAM (Zero-Bus Turnaround RAM) and then sent to the emulator, which is only able to apply such coefficients through a two-minute time interval.

The third drawback of the studied channel emulators is related to scalability. As it can be derived from the results exposed in (141), when we work with a time-domain based channel emulator, the gate count (i.e. the number of 2-NAND logic gates that would be required to implement the same number and type of logic functions) roughly doubles every time we add a transmit and a receive antenna to the system. Therefore, a scalable solution would have to be able to deal with more inputs and outputs without requiring such important hardware complexity increases.

In the next Sections of this Chapter two different approaches are proposed in order to solve the mentioned drawbacks. Both of them make use of Xilinx System Generator to develop highly-configurable channel emulators faster than using a Hardware Description Language (HDL). The System Generator designs are optimized to fit twelve to twenty-four complex path channel emulators into one FPGA. Moreover, although both channel emulators were designed bearing in mind that they had to be able to work in stand-alone mode, the emulator described in Section 6.3.1 was specifically built to cause the lowest possible impact on resource-consumption in case of requiring additional transmit/receive antennas, therefore facilitating scalability.

6.3 MIMO Performance Evaluation Systems

Two similar evaluation systems were developed. Their main components were already depicted in Figs. 4.4 and 4.5, where transmitters and receivers exchange data with the vehicular channel emulators through the PCI bus.

Both channel emulators are based on the same FPGA described in Section 5.5: a Virtex IV (XC4VSX35-10FF668) placed inside a Nallatech's BenADDA-IV development kit. Like in the SISO case, a rapid-prototyping tool (Xilinx System Generator) was used for developing the vehicular channel emulators, taking advantage of one of the System Generator features: its ability to exchange data between a design running on the FPGA and a software implementation that is executed on a PC.

6.3.1 FPGA-based MIMO Channel Emulator Built by Upgrading the SISO Vehicular Channel Emulator

The first channel models implemented on the FPGA-based vehicular emulator were briefly described in Section 5.3. There, six different 5.9 GHz high-speed environments are exposed, which cover some of the most common situations where VTV and RTV communications may take place. Such channel models can be grouped into three major scenarios: urban canyons (*VTV-Urban Canyon Oncoming*, *RTV-Urban Canyon*), expressways (*VTV-Expressway Oncoming*, *RTV-Expressway*, *VTV-Expressway Same Direction With Wall*) and suburban surface streets (*RTV-Suburban Street*). Urban canyons and suburban surface streets assume a speed of 120 Km/h, whereas the expressway measurements were made consistent with speeds of 140 Km/h. Table 5.1 shows the main characteristics of the models.

The first attempt to expand the SISO emulator to accept more input and output antennas consisted in creating a SIMO 1 × 2 system by replicating the SISO hardware. The design obtained was too large to fit into the FPGA, so optimizations were required. For the sake of conciseness, there will be cited below only the three most important optimizations carried out, whose savings are summarized in Table 6.1, where the more complex of the six designs (in terms of FPGA resources consumed) is used as a reference (*VTV-Expressway Same Direction with Wall*).

Table 6.1: Resource utilization of different versions of the SIMO/MIMO vehicular channel emulator.

Version	Slices	Slice flip-flops	LUTs	FIFO16/RAMB16	DSP48
SISO	85%	40%	57%	24%	65%
SIMO 1 × 2 (V1)	113%	74%	110%	40%	100%
SIMO 1 × 2 (V2)	107%	71%	104%	36%	100%
SIMO 1 × 2 (V3)	99%	69%	89%	37%	78%
MIMO 4 × 4	82%	43%	60%	27%	66%

6.3.1.1 First Optimization

The first optimization reduced the amount of resources dedicated to perform the Doppler filtering stage (i.e. the stage aimed at applying each path's Doppler spectrum) by using a four-output Doppler filter. This filter is a natural evolution of the SISO filter represented in Fig. 5.5. It makes use of a four-input multiplexer and has a four-output demutiplexer after the *FIR Compiler* block (see Fig. 6.2). This is possible since every path uses the same Doppler filters. The resources occupied by this optimized version of the emulator are shown in Table 6.1 in the row for *SIMO* 1 × 2 *(V1)*.

6.3.1.2 Second Optimization

The second optimization consisted in realizing that each System Generator's AWGN (Additive White Gaussian Noise) block was consuming a lot of resources (in fact, in Section 5.7.2 it was quantified the savings obtained when replacing 24 AWGN blocks with a 24-output multiplexer). Thus, one of the two existing generators was removed and it was created a 50-output Gaussian generator (in SIMO 1 × 2 we only need 48

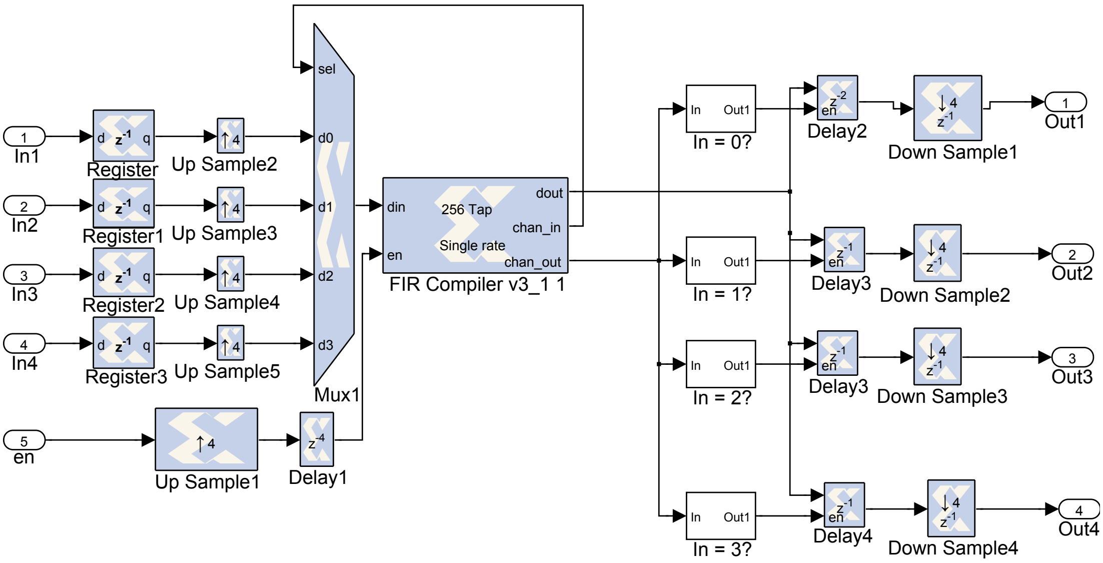

Figure 6.2: New Doppler filtering stage for the SIMO 1 × 2 channel emulator.

outputs, but 50 is the closest integer divider of 10 MHz). Note that System Generator's demultiplexers are restricted to use up to 32 outputs, so it is necessary to build an ad-hoc 50-output demultiplexer like the one depicted in Fig. 6.3. As it can be seen in Table 6.1, in the row *SIMO* 1 × 2 *(V2)*, this optimization allows saving 6% of the slices, 3% of the slice flip-flops, 6% of the LUTs and 4% of the FIFO16/RAMB16 blocks, but it was yet too large to fit into the FPGA.

6.3.1.3 Third Optimization

The third optimization consisted in allowing every path to share Doppler filters, interpolators and FIR (Finite Impulse Response) filters, being the Gaussian noise generated unique for each path. Hence, it was built a multiplexed version of the emulator that buffered incoming signals, switched the Gaussian noise source and applied the channel to each transmitted signal at the right time instants. As it is shown in the row *SIMO* 1 × 2 *(V3)* of Table 6.1, the used resources decreased substantially with respect to version V2 and permitted to save 8% of the slices, 2% of slice flip-flops, 15% of the LUTs and 22% of the DSP48 blocks. Thus, the design finally fitted into the FPGA, but it was clear that it would be very difficult to fit a MIMO 4 × 4 system (which has 8 times more paths than a SIMO 1 × 2 system) following the same optimization philosophy.

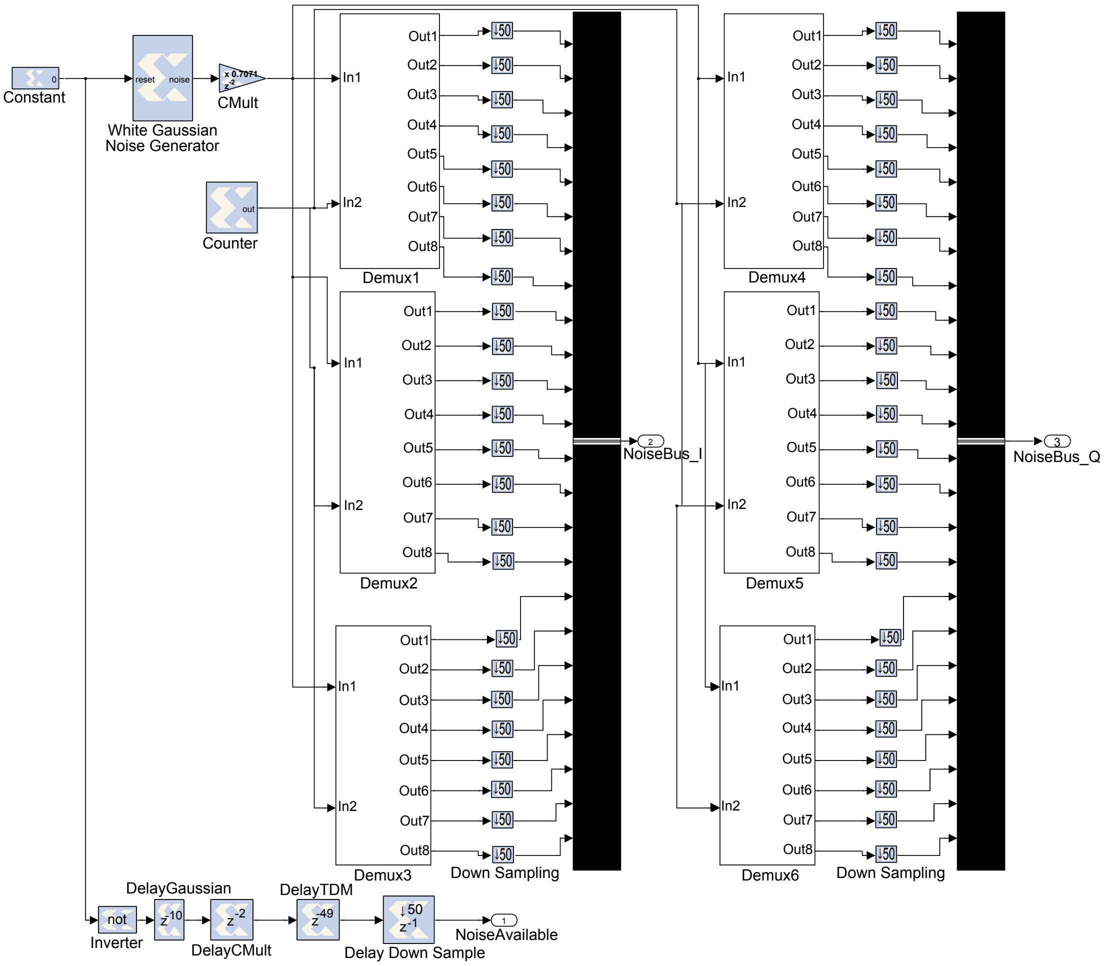

Figure 6.3: New Gaussian noise generator block for the SIMO 1 × 2 channel emulator.

6.3.1.4 MIMO Time-Multiplexed Approach

To solve the space issue there are several alternatives. The most obvious solution consists in purchasing a development kit with a larger FPGA, but that would increase the final cost of the emulator. Another solution could be the offline generation of the channel coefficients, but then we would require a PC or an embedded device every time the vehicular channel emulator was used (i.e. the emulator would not be able to work in stand-alone mode). It could be even refined the HDL code generated by System Generator, but that could take an important amount of time.

After considering all the alternatives devised, a scheme based on input and output buffers was proposed (shown in Fig. 4.4). Such buffers act similarly to two synchronized parallel-to-serial and serial-to-parallel converters. Each set of signals transmitted from an array of IEEE 802.11p transceivers are stored into a buffer and released at specific time instants, achieving a similar effect as if the parallel transceivers were executed in series. In Table 6.1, row *MIMO* 4×4, it can be seen the important resource savings attained by using the time-multiplexing approach, which even consumes 3% less slices than the SISO version thanks to the optimizations described in subsections 6.3.1.1 to 6.3.1.3.

It is important to emphasize that although the resources consumed by the time-multiplexed approach have been indicated in Table 6.1 using the term *MIMO* 4×4, such resources would be the same for any system of up to four antennas in transmission and four antennas in reception. In fact, the experiments performed in Section 6.4 used the same time-multiplexed design but for transceivers with less than four antennas in transmission/reception (SIMO 1×2, SIMO 1×3, SIMO 1×4 and MIMO 2×2). Obviously, such systems do not use the whole memory offered by the input/output buffers, which is optimized for a MIMO 4×4, but the channel emulator design is the same for all of them.

It is also important to note that the input buffer has to add zeroes between each pair of signals that were transmitted by different antennas so as to reduce time correlation. In the SISO emulator described in Chapter 5 the coefficient generation rate was set to frequencies that range between 2 KHz and 4 KHz, so the shortest time the channel would remain almost constant would be 500 μs. Therefore, if the FPGA clock is set to 20 MHz, the number of cycles that the channel coherence would remain constant would be 500 μs / 50 ns = 10,000. If we want to stay safe, we can wait for 100,000 cycles (5 ms) to guarantee minimal correlation. Hence, each pair of signals was separated by 5 ms. It was easy to test this assumption by measuring the channel coefficient autocorrelation when transmitting 10,000 consecutive 64-subcarrier OFDM symbols: no relevant correlations were found (the non-diagonal coefficients of the correlation matrix ranged approximately between 10^{-2} and 10^{-3}).

6.3.2 FPGA-based MIMO Channel Emulator for the New MIMO 2×2 Acosta's Channel Model

In 2010 some of the authors of the vehicular channels described in (60) used a similar methodology to present a novel vehicular MIMO channel model in (115). There, it is described an empirical model for an urban environment around Georgia Tech campus where a transmit vehicle travelled roughly 100 meters ahead of the receive vehicle at a nominal speed of 56 Km/h (the actual speed ranged between 40 Km/h and 56 Km/h). The vehicles conformed a MIMO 2 × 2 system that used 2.4 GHz co-located dual-polarized antennas. The authors modeled the channel and extracted characteristics for the four channel coefficients, whose main parameters are summarized in Table 6.2.

Table 6.2: Main parameters of the new Acosta's MIMO 2 × 2 empirical model.

Coefficient	**Number of Taps/Paths**	**Maximum Relative Path Loss (dB)**	**Maximum Delay (ns)**	**Maximum Frequency Shift (Hz)**	**Maximum Fading Doppler (Hz)**
$h_{1,1}$	8/24	8.8	196	-96	345
$h_{1,2}$	3/9	-6.3	56	-80	197
$h_{2,1}$	3/9	-11.2	56	38	261
$h_{2,2}$	6/18	-4.7	140	95	336

6.3.2.1 MIMO System Design

The design of this MIMO 2 × 2 emulator was very similar to each of the models of Section 6.3.1, but there are several issues that had to be modified to adapt the model to the parameters specified in (115).

First, it must be noted that the model is made up of four channel coefficients that describe channels with up to 24 complex paths (there are only up to 12 complex paths in each model described in Section 6.3.1), what increases importantly the size of the FPGA design. In Table 6.3 it is exposed the amount of FPGA resources consumed by each channel coefficient design, that in some cases reaches the 99% of the total slices and more than 90% of some of the resources (hence, a MIMO emulator design does not fit into the FPGA used). Furthermore, in this case the buffer-based time-multiplexed scheme described in the previous Section cannot be applied, since each channel coefficient does not share the Doppler filtering stage (the Doppler effect parameters differ greatly from one coefficient to another). To solve this issue a new time-multiplexed approximation (represented in Fig. 4.5) was designed where each transceiver antenna has to be synchronized in time with the emulator to send its signal when the emulator requires it.

Table 6.3: General parameters and resources occupied by the new MIMO 2 × 2 vehicular channel emulator

Channel Coefficient	Coefficient generation rate [Effective rate] (Hz)	Interpolation rate	Occupied Slices (%)	Occupied Slice Flip-Flops (%)	Occupied LUTs (%)	Occupied FIFO16 / RAMB16s (%)	Occupied DSP48s (%)
$h_{1,1}$	738 [800]	x12,500	99%	63%	93%	31%	100%
$h_{1,2}$	458 [500]	x4,000	56%	28%	42%	16%	39%
$h_{2,1}$	446 [400]	x20,000	52%	27%	39%	15%	39%
$h_{2,2}$	590 [640]	x15,625	99%	51%	92%	25%	77%

In Table 6.3 it can be also observed that, in comparison to the models of (60) and due to the lower vehicular speeds, there exist lower Doppler frequencies, what lead to lower coefficient generation rates and higher interpolation factors. Note that the coefficients must adapt their rate to the rate of the incoming signal (i.e., the signal from the IEEE 802.11p transmitter that arrives at 10 MHz). These coefficients are generated at a rate that depends on the maximum Doppler shift and that is much lower than the FPGA frequency. Indeed, in a specific vehicular channel, the implicit sample rate is twice the maximum Doppler shift of all paths. In the implemented vehicular channel models, this rate fluctuates between 446 Hz and 738 Hz (see Table 6.3). To avoid designing a complex resampling stage, instead of using the original coefficient generation rate, it is used an effective sample rate that is equal to the nearest integer divider of 10 MHz.

Besides, there is an important difference with respect to the upgraded SISO models: each of the 24 complex paths follows a Rayleigh distribution, there are no Rician components. Nevertheless, although there are no LOS components, it can be observed that the coefficient $h_{1,1}$ is noticeably stronger than the other channel coefficients (see column *Maximum Relative Path Loss* in Table 6.2, where the term *Relative Path Loss* is actually used to refer to the average power of each path).

6.3.2.2 Coefficient Generation Process

Each complex coefficient is generated following a process that is very similar to the one detailed in Section 5.8:

- First, Gaussian noise is generated by a system like the one depicted in Fig. 6.3. Note that since the noise is generated at 20 MHz, the demultiplexer can output samples at up to 20 MHz / 50 = 400 KHz, which is clearly higher than the maximum effective coefficient generation rate (800 Hz, see Table 6.3). Each complex coefficient will be then conformed by a pair of Gaussian values with zero mean and unit variance.

- Next, the coefficients of each path are filtered to apply the Doppler characteristics indicated by the model. Such characteristics are determined by the frequency shift, the fading Doppler and the fading spectral shape. The Doppler filtering stage is like the one shown in Fig. 6.2, but only for two input/output signals instead of four.
- After filtering, the coefficients are interpolated to adapt their rate to the signal that comes from the IEEE 802.11p transceivers at 10 MHz. Due to the low effective coefficient generation rates, the interpolator requires interpolation factors that range between 4,000 and 20,000. An extensive description of the components and the functioning of the interpolation stage is given in Section 5.7.4.
- Finally, a complex FIR is used to apply the coefficients to the incoming signal (the System Generator design is similar to the one shown in Fig. 5.8, but for up to 24 complex paths).

6.4 Experimental Results

6.4.1 Co-Simulation Vs. Software Simulation Lags

In order to quantify the speed improvements achieved by the FPGA-based simulation models proposed, it is interesting to compare its lags with the ones obtained when simulating using the MATLAB/Simulink version of the vehicular channel emulators. The results are exposed in Table 6.4. Note that the lag is the time interval that goes by since a sample arrives at the input of the emulator until it is processed and placed at the output. Such lag is constant for every vehicular channel implemented. The study of the lags shows that the use of the co-simulation mode (a mode where the transceiver runs in MATLAB/Simulink, whereas the emulator runs on the FPGA) dramatically increases the processing speed: each sample can be processed between 6 and 209 times faster than using the Simulink-based approach.

It is important to emphasize that such speed improvement factors are achieved even despite the fact that the co-simulation lag includes the time needed to transfer the signals through the PCI to/from the FPGA. Moreover, in the case of the channel models described in Section 6.3.1, the lag also includes 5 ms delays between each pair of signals transmitted.

Finally, note that in the MIMO 2 × 2 channel emulator co-simulation lags vary for each channel coefficient depending on the complexity of its model. Thus, $h_{1,1}$, which is made up of 24 complex paths, has the largest lag, while $h_{1,2}$ and $h_{2,1}$ obtain the smallest lags because they are made of just 9 complex paths.

6.4.2 Performance Measurements

6.4.2.1 Characteristics of the Measurements

To evaluate the performance of the software transceivers, their signals were passed through the FPGA-based vehicular channel emulators taking advantage of the co-simulation mode. In order to achieve a fair comparison the same transmission parameters were set for every transceiver and it was assumed that all of them send signals with the same transmission power. A rate 1/2 FEC (Forward Error Correction) code was used and the OFDM subcarriers were filled with QPSK-modulated symbols. The receiver assumed perfect time synchronization and the channel was estimated using the OSTPM-based method described in Section 4.2.2.2. In the SISO systems an MMSE (Minimum Mean Square Error) linear equalizer followed by an ML (Maximum Likelihood) detector was used, while SIMO transceivers implemented the MRC (Maximum Ratio Combining) technique. In the case of MIMO receivers, symbols were decoded if needed (an Alamouti decoder is used for 2×2 systems) and ML detectors were applied.

A maximum of 10,000 48-bit FEC blocks were averaged for different SNR (Signal-to-Noise Ratio) values (the simulation stopped for each SNR value when 100 erroneous FEC blocks were detected).

6.4.2.2 Performance in Frequency-flat Rayleigh Channel

In order to obtain a reference of the performance of the transceivers implemented, they were evaluated over a non-vehicular environment. Fig. 6.4 and Fig. 6.5 show, respectively, the BER (Bit Error Rate) and FER (FEC block Error Rate) of the transceivers when transmitting over a spatially-uncorrelated frequency-flat block fading Rayleigh channel whose coefficients were constant during one FEC block.

Table 6.4: Co-Simulation and Simulink lags for different channels and multiple-antena systems.

Number of paths	Channel	Software lag (s)	Co-Simulation lag (s)	Speed Improvement factor
1 (SISO)	Expressway Oncoming	44.5096	0.4256	x104.58
	Urban Canyon	70.5096		x165.67
	Expressway	70.5096		x165.67
	Urban Canyon Oncoming	44.5096		x104.58
	Suburban street	88.5096		x208.96
	Expressway Same Direction With Wall	44.5096		x104.58
2 (1 × 2, 2 × 1)	Expressway Oncoming	44.9184	0.8424	x53.32
	Urban Canyon	70.9184		x84.19
	Expressway	70.9184		x84.19
	Urban Canyon Oncoming	44.9184		x53.32
	Suburban street	88.9184		x105.55
	Expressway Same Direction With Wall	44.9184		x53.32
3 (1 × 3, 3 × 1)	Expressway Oncoming	45.3272	1.2600	x35.97
	Urban Canyon	71.3272		x56.61
	Expressway	71.3272		x56.61
	Urban Canyon Oncoming	45.3272		x35.97
	Suburban street	89.3272		x70.89
	Expressway Same Direction With Wall	45.3272		x35.97
4 (1 × 4, 4 × 1, 2 × 2)	Expressway Oncoming	45.7360	1.6776	x27.26
	Urban Canyon	71.7360		x42.76
	Expressway	71.7360		x42.76
	Urban Canyon Oncoming	45.7360		x27.26
	Suburban street	89.7360		x53.49
	Expressway Same Direction With Wall	45.7360		x27.26
4 (MIMO 2 × 2)	$h_{1,1}$	787.62	4.70	x167.58
	$h_{1,2}$	314.31	3.91	x80.39
	$h_{2,1}$	393.30	3.91	x100.59
	$h_{2,2}$	622.81	4.39	x141.87
16 (4 × 4)	Expressway Oncoming	50.6416	7.49	x6.76
	Urban Canyon	76.6416		x10.23
	Expressway	76.6416		x10.23
	Urban Canyon Oncoming	50.6416		x6.76
	Suburban street	94.6416		x12.64
	Expressway Same Direction With Wall	50.6416		x6.76

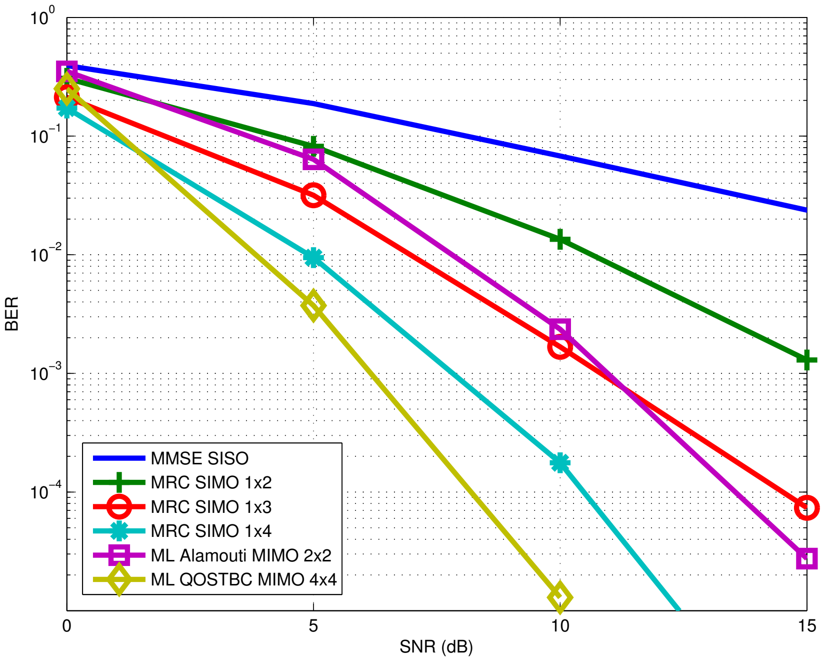

Figure 6.4: BER performance of SISO, SIMO and MIMO IEEE 802.11p transceivers for frequency-flat block fading Rayleigh channel.

It can be observed for both BER and FER that increasing the number of transmit and receive antennas greatly improves the performance of the SISO transceiver. If we take as a reference the SNR needed to obtain a FER of 10%, there exists a difference of 9 dB between the SISO system and the SIMO 1×4 or the MIMO 4×4, what implies that a SISO transceiver would require 8 times more power than a SIMO 1×4 or a MIMO 4×4 to obtain the same FER.

Finally, it is also interesting to note that there is a difference of about 3 dB between the SIMO 1×4 and the MIMO 2×2 when comparing their BER curves. This effect happens because the MIMO 2×2 uses Alamouti coding: in general, when comparing an Alamouti coding system with n_R antennas at reception with a SIMO system with $2n_R$ receive antennas, the former only provides an array gain of n_R, which translates into an SNR loss of 3 dB.

6.4.2.3 Channel Estimation Error in Rayleigh Channel

In spite of the good performance of the MIMO systems in a Rayleigh channel, their channel estimation accuracy is inferior to the one obtained by SIMO systems. This is

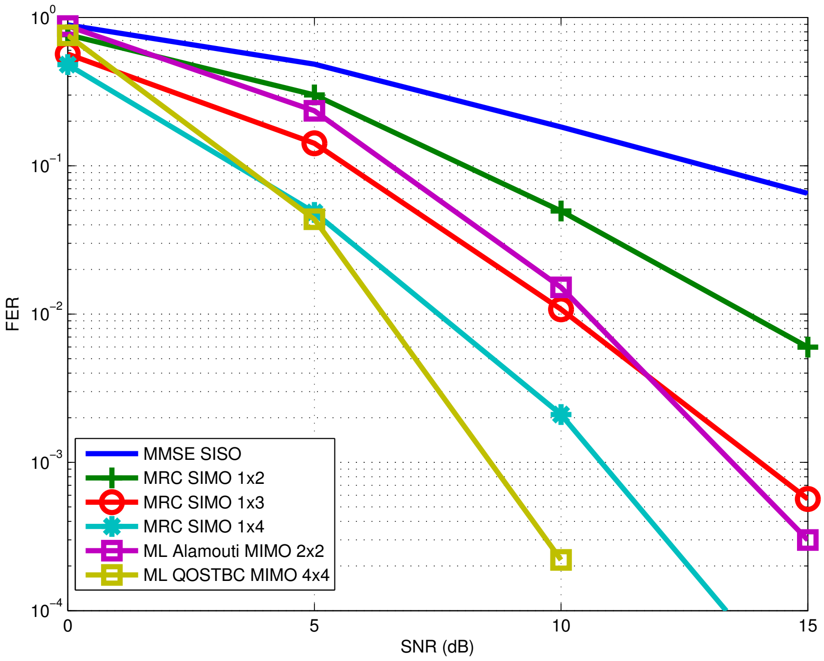

Figure 6.5: FER performance of SISO, SIMO and MIMO IEEE 802.11p transceivers for frequency-flat block fading Rayleigh channel.

due to the fact that in a SIMO system it is possible to obtain up to four channel estimations per OFDM symbol, whereas in a MIMO system, because of the orthogonality-based channel estimation method used, only two (in a MIMO 2×2) or one (in a MIMO 4×4) channel estimations can be performed per OFDM symbol.

In Fig. 6.6 it can be clearly observed that MIMO 4×4 systems obtain worse channel estimations than MIMO 2×2 and SIMO 1×4 systems. Such estimation is also worsened by the Doppler effect: the higher the Doppler frequency, the lower the accuracy of the channel estimation (see Fig. 6.7). However, note that in this specific channel the difference in the estimation error between the lowest (1 Hz) and highest (4,000 Hz) Doppler is really small.

6.4.2.4 Performance in the Vehicular Channels Built by Upgrading Acosta's SISO models

Figs. 6.8 to 6.19 show the BER and FER curves obtained for the six vehicular channels. It can be observed that in all scenarios SIMO/MIMO transceivers outperform (both in terms of BER and FER) SISO systems.

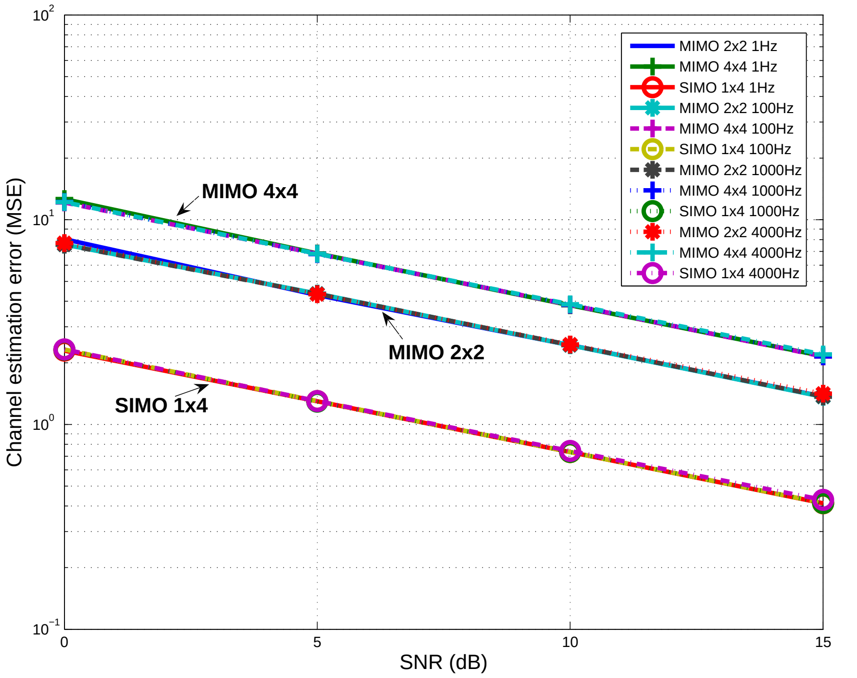

Figure 6.6: Channel estimation error for frequency-flat block fading Rayleigh channel for different Doppler frequencies for SIMO and MIMO IEEE 802.11p transceivers.

In the emulated conditions, SIMO systems seem to offer the best tradeoff between hardware complexity and performance. In almost every environment the SIMO 1 × 4 system obtains the best performance, while SIMO 1 × 3 or MIMO 4 × 4 are the transceivers that get the second best results.

SIMO 1×2 and MIMO 2×2 attain similar BER/FER performance, while the results obtained by the MIMO 4×4 differ depending on the vehicular channel. It is possible to observe that the most influential factor is the existence of low Rician K factors, which have a direct relationship with performance: in general, it can be stated that the higher the Rician K, the better the performance (Rician K factors are shown in Table 5.1). In fact, if we rank the channels by decreasing Rician K (using the overall K factor for *VTV-Expressway Same Direction With Wall*, since it has two Rician paths), we obtain the same ranking as when we rank them from best to worst SISO performance. Moreover, if we do the same but for the MIMO 4 × 4 performance, we obtain almost the same ranking but with a slight difference: *RTV-Expressway* ranks fifth instead of sixth, and *VTV-Expressway Oncoming* ranks sixth instead of fifth. This change in

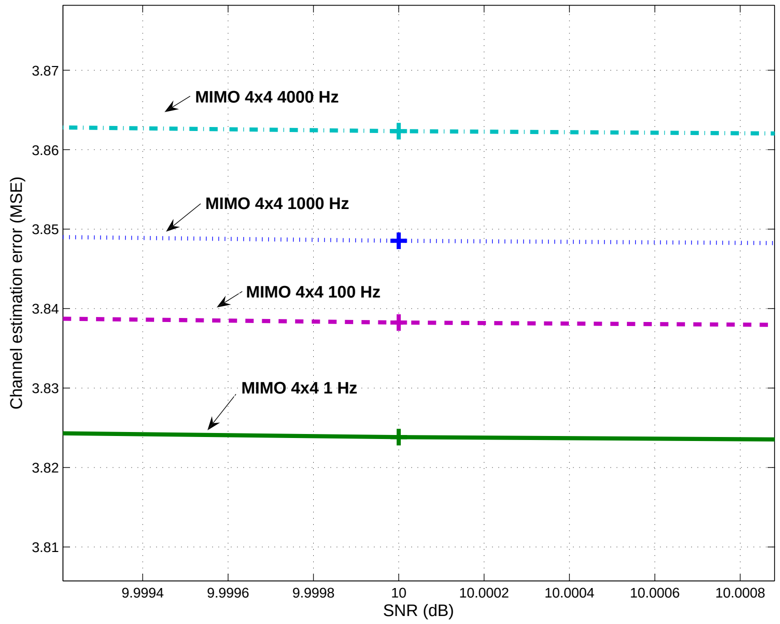

Figure 6.7: Channel estimation error for the MIMO 4 × 4 IEEE 802.11p transceivers around SNR=10 dB for frequency-flat block fading Rayleigh channel.

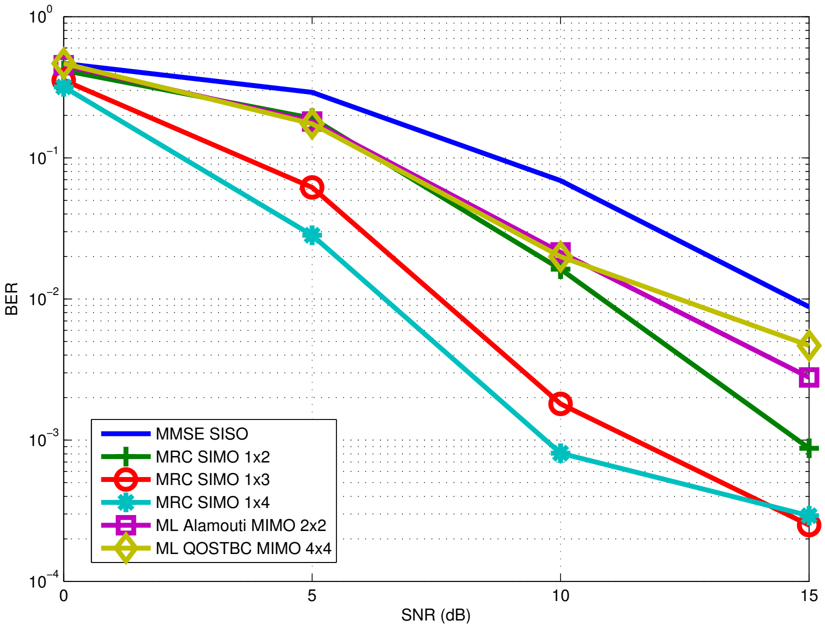

Figure 6.8: BER performance for SISO, SIMO and MIMO *VTV-Expressway Oncoming*.

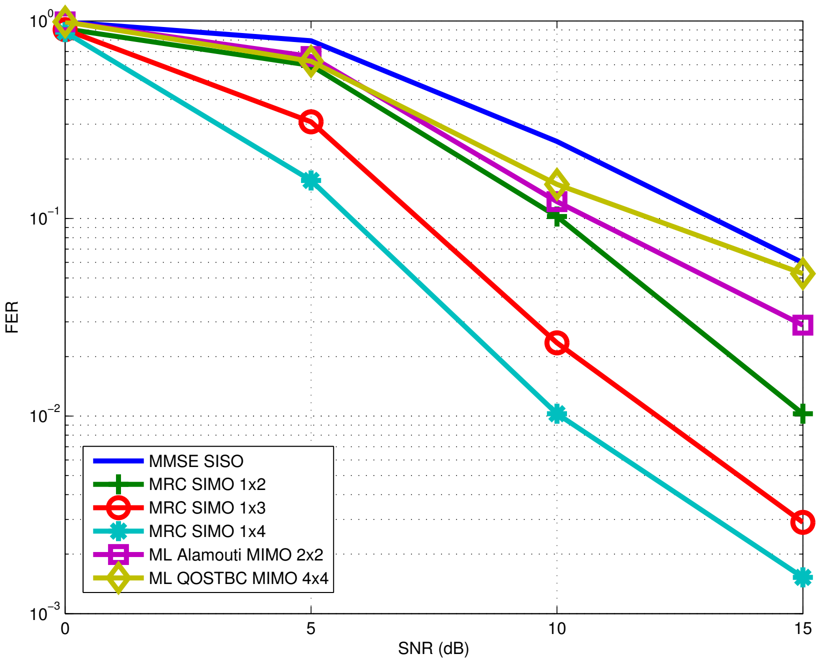

Figure 6.9: FER performance for SISO, SIMO and MIMO *VTV-Expressway Oncoming.*

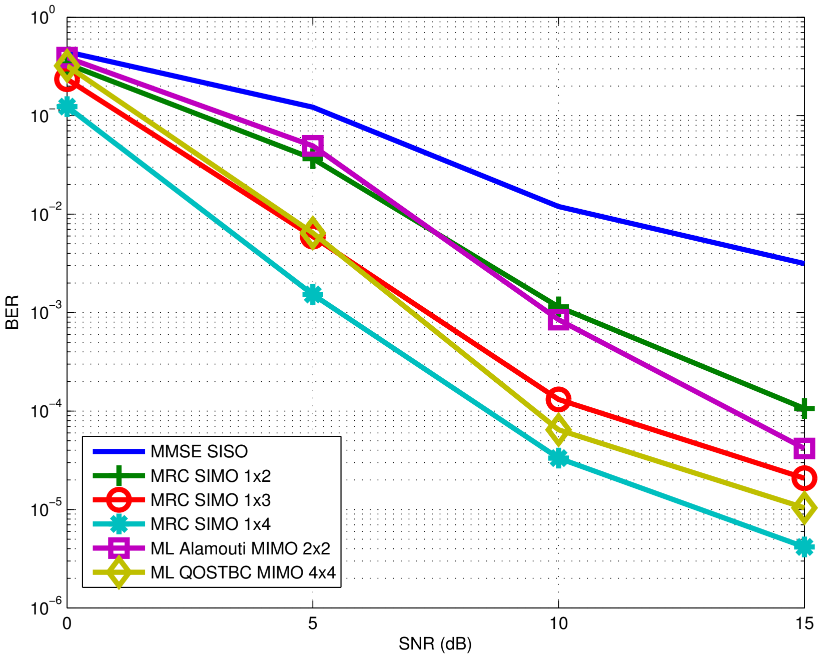

Figure 6.10: BER performance for SISO, SIMO and MIMO *RTV-Urban Canyon.*

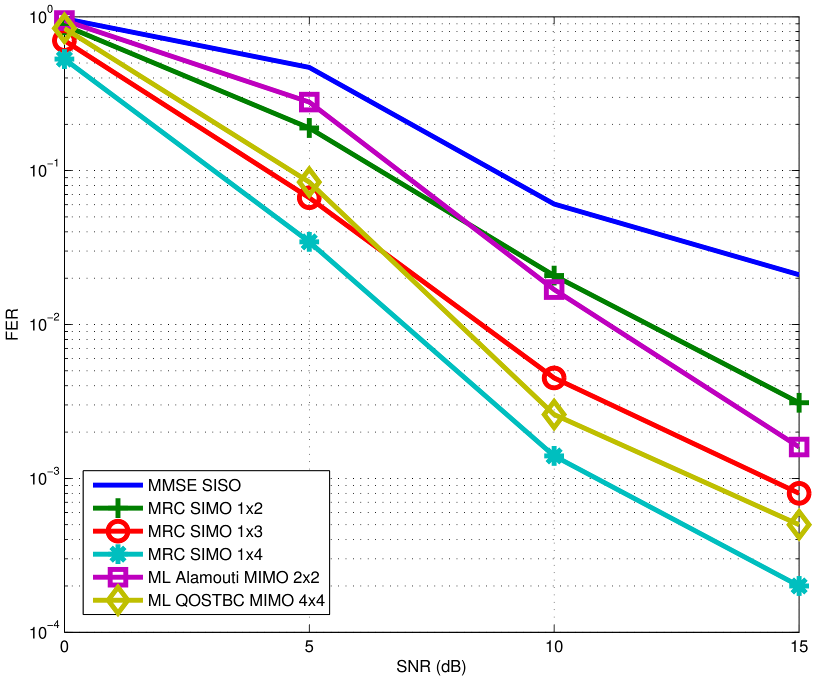

Figure 6.11: FER performance for SISO, SIMO and MIMO *RTV-Urban Canyon*.

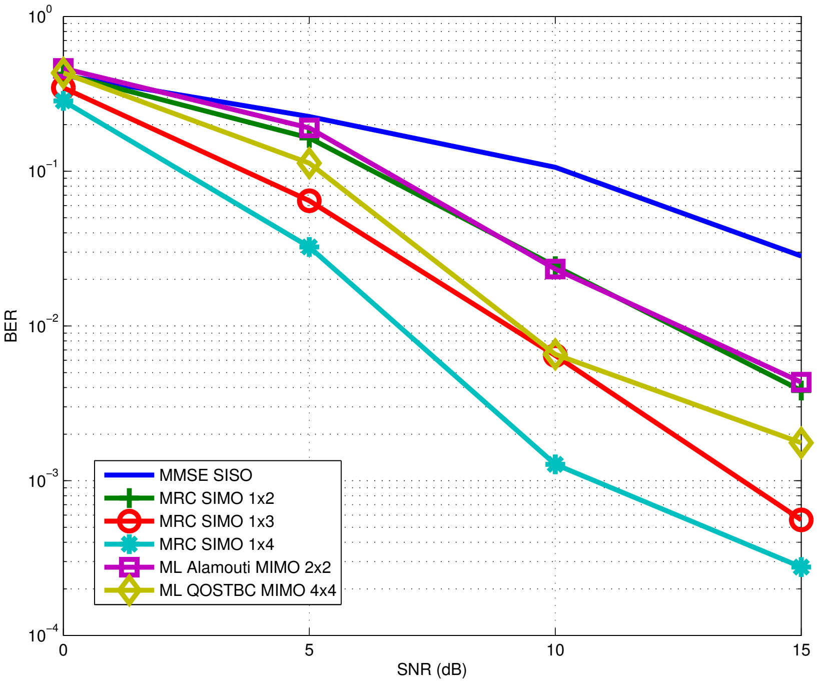

Figure 6.12: BER performance for SISO, SIMO and MIMO *RTV-Expressway*.

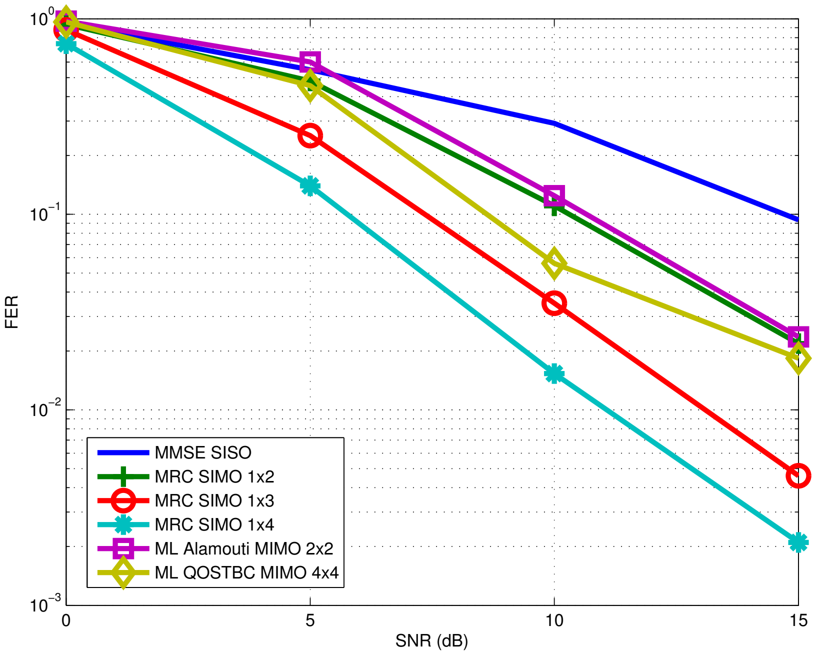

Figure 6.13: FER performance for SISO, SIMO and MIMO *RTV-Expressway.*

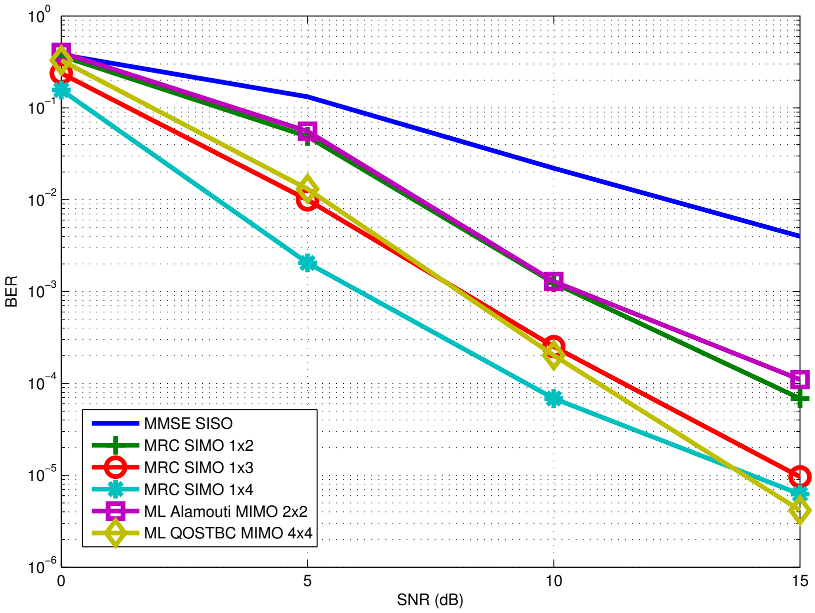

Figure 6.14: BER performance for SISO, SIMO and MIMO *VTV-Urban Canyon Oncoming.*

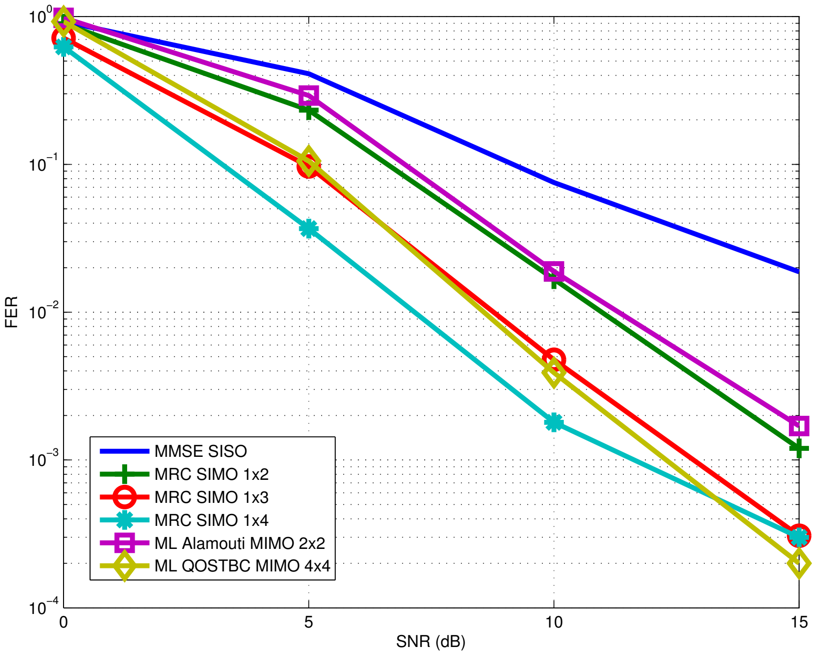

Figure 6.15: FER performance for SISO, SIMO and MIMO *VTV-Urban Canyon Oncoming.*

the ranking seems to be influenced by a second factor: the presence of high Doppler frequencies in *RTV-Expressway Oncoming*.

As it was mentioned in subsections 4.2.2.2 and 6.4.2.3, due to the channel estimation technique implemented, a high Doppler frequency might lead to a bad channel estimation and therefore to worse performance results, especially in multi-antenna systems. Note that in the case of MIMO 4 × 4 the current channel estimation technique can only make use of one pilot for the entire OFDM symbol, therefore slight channel variations throughout an OFDM symbol may lead to a bad estimation. To determine if the channel estimation technique worsens MIMO 4 × 4 performance, it is necessary to compare the channel estimation errors obtained by such system in the different vehicular scenarios. No noticeable correlation was found between performance (shown in Figs. 6.20 and 6.21) and channel estimation error (depicted in Fig. 6.22).

If a FER of 10% is set as a reference, it is possible to obtain Table 6.5, which gives a good idea of the performance of each transceiver and confirms the previous statements. The maximum differences in SNR occur when comparing the SISO and

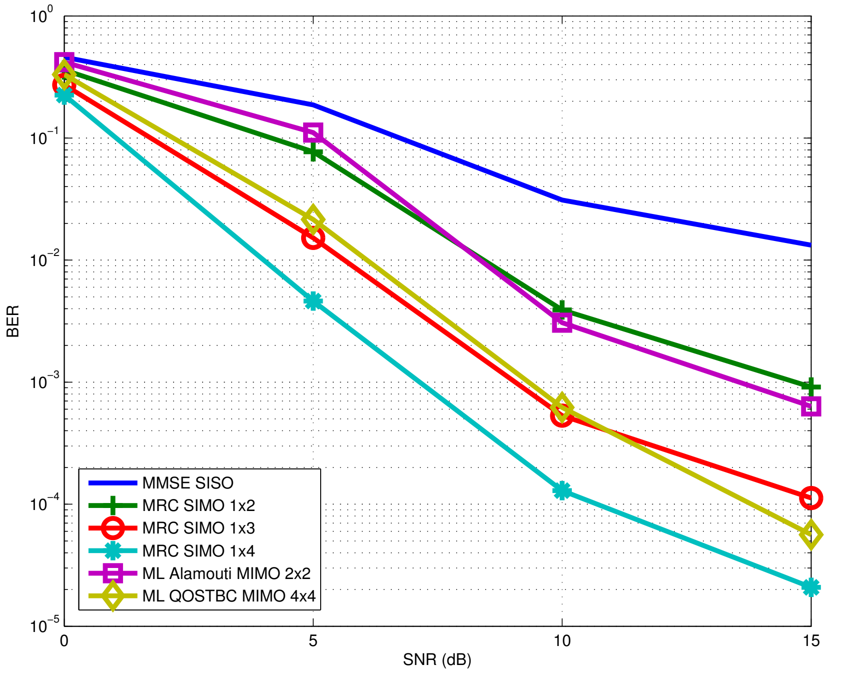

Figure 6.16: BER performance for SISO, SIMO and MIMO *RTV-Suburban Street*.

the SIMO 1×4 systems, ranging between 5.73 dB (for *RTV-Urban Canyon*) and 8.96 dB (for *RTV-Expressway*). These differences in SNR involve that a SIMO 1 × 4 system would require between 4 and 8 times less power than a SISO system to obtain the same FER.

Table 6.5: SNR (dB) required to obtain a FER of 10% in each SIMO/MIMO vehicular channel.

Channel	**SISO**	**SIMO**	**SIMO**	**SIMO**	**MIMO**	**MIMO**
		1 × 2	1 × 3	1 × 4	2 × 2	4 × 4
VTV-Expressway Oncoming	13.17	10.05	7.19	5.81	10.68	11.91
RTV-Urban Canyon	8.78	6.44	4.14	3.05	6.83	4.26
RTV-Expressway	14.71	10.30	7.35	5.75	10.65	8.62
VTV-Urban Canyon Oncoming	9.16	6.60	4.91	3.23	6.95	5.08
RTV-Suburban Street	11.08	7.65	5.37	4.12	7.84	6.10
VTV-Express. Same Dir. With Wall	11.80	6.32	4.94	3.52	6.30	6.79

Finally, it is important to emphasize that multiple-antenna transceivers obtain their largest SNR gains over SISO systems when transmitting over channels that assume high

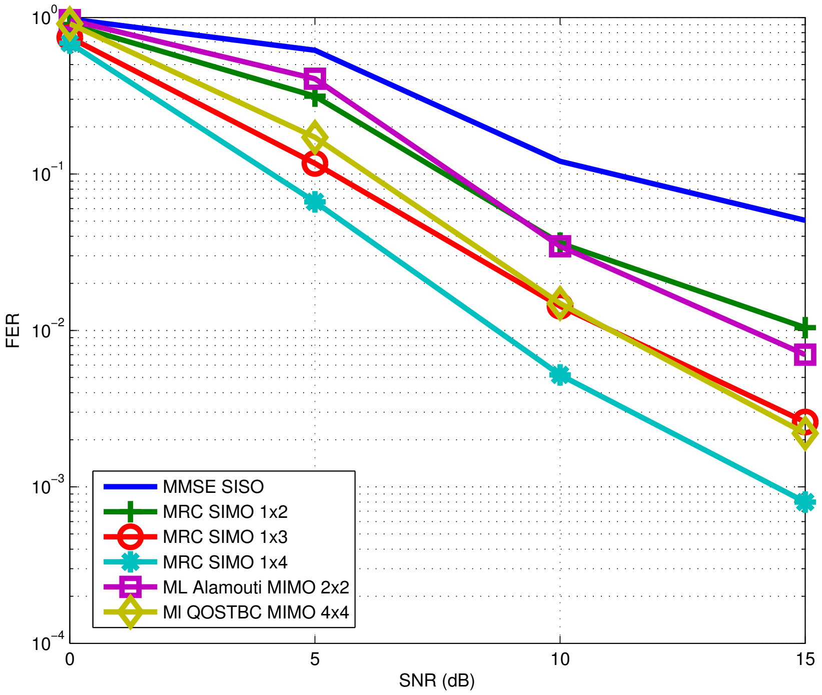

Figure 6.17: FER performance for SISO, SIMO and MIMO *RTV-Suburban Street*.

vehicular speeds (i.e. the scenarios located in expressways), achieving in the case of the SIMO 1 × 4 a gain between 7.36 and 8.96 dB. This is a quite interesting result since it means that mobile communications performed in high speed scenarios can be greatly improved by placing antenna arrays along the roadside and/or in vehicles and applying relatively simple space-time diversity techniques.

6.4.2.5 Performance in the New MIMO 2×2 Acosta's Channel Model

Figs. 6.23 and 6.24 show comparisons of BER and FER between the MIMO 2 × 2 vehicular channel (labeled as *VTV-Urban MIMO* 2 × 2) and the rest of the vehicular channels analyzed in the previous section. It can be observed that the MIMO 2 × 2 IEEE 802.11p transceiver performs worse in the *VTV-Urban MIMO* 2 × 2 both in terms of BER and FER. Despite the fact that the Doppler characteristics in the *VTV-Urban MIMO* 2 × 2 are not as tough as in the other vehicular channels, there exists a factor with deeper influence on the performance: the lack of a LOS component.

Indeed, while the upgraded SISO models are always composed of one or two Rician paths and several Rayleigh paths, in the case of the *VTV-Urban MIMO* 2 × 2 all the

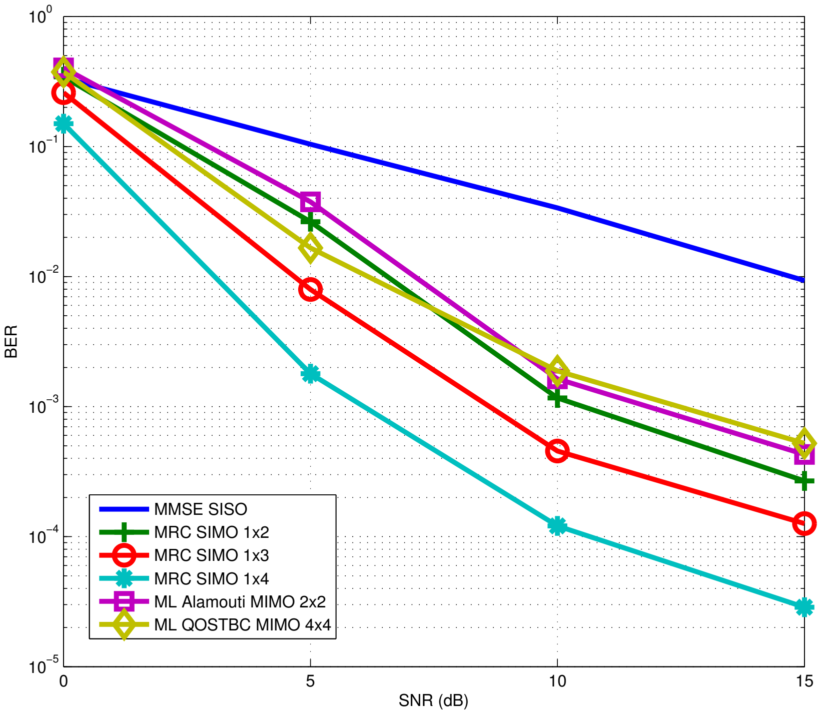

Figure 6.18: BER performance for SISO, SIMO and MIMO *VTV-Exp. Same Direction With Wall.*

paths are Rayleigh and therefore such components are assumed to be NLOS. These results corroborate one of the conclusions of Section 6.4.2.4: the lower the Rician K, the lower the performance. Thus, since a Rayleigh channel has a near-zero Rician K factor, it can be expected that such channel performs worse than others with certain degree of line-of-sight (i.e. Rician paths).

6.5 Conclusions

In this Chapter it was detailed the design and development of a multiple-antenna IEEE 802.11p performance evaluation system made up of low-cost and flexible FPGA-based MIMO channel emulators that work in conjunction with the software transceivers described in Chapter 4.

The performance of the PHY layer of SISO, SIMO and MIMO transceivers has been measured when transmitting on seven different scenarios that include situations in urban areas, suburban environments and expressways.

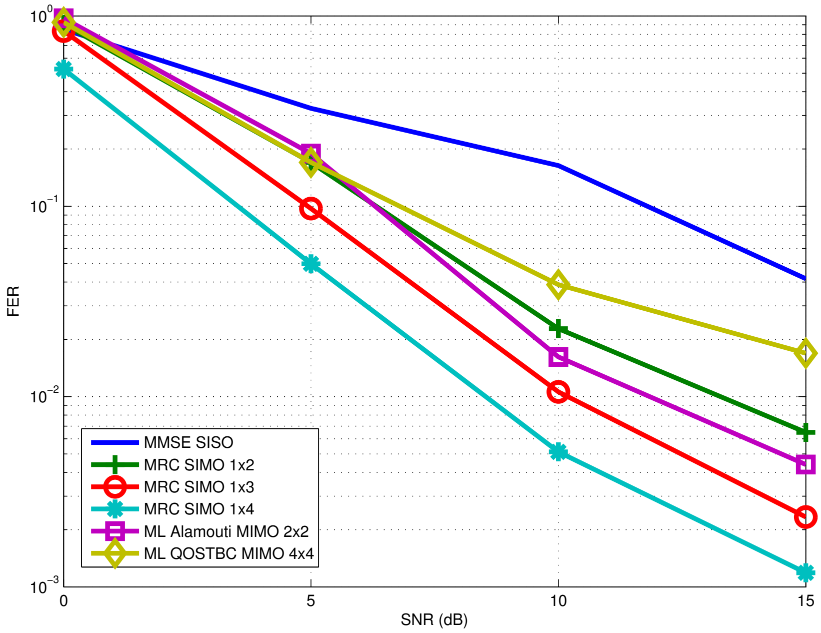

Figure 6.19: FER performance for SISO, SIMO and MIMO *VTV-Exp. Same Direction With Wall.*

With respect to the emulators, first it was described how to turn the SISO vehicular channel emulator created in Chapter 5 into a MIMO 4 × 4 system by using a time-multiplexed approach that occupies only 82% of the slices of a Xilinx Virtex IV. Additionally, it was shown how to reuse some of the stages of the upgraded SISO system to create a MIMO 2 × 2 channel emulator based on recently obtained MIMO 2 × 2 empirical data. These emulator designs are able to accelerate vehicular simulation speed between 6 and 209 times.

Regarding the upgraded SISO vehicular measurements, it can be concluded that, in the chosen experimental conditions, SIMO and MIMO transceivers outperform SISO systems: they require less transmission power to attain the same BER/FER, they can reach higher data rates and they are able to improve drastically the SISO performance in many of the most common vehicular scenarios. Among the different transceivers it can be stated that the one that performs best is the SIMO 1 × 4, whereas more complex systems, such as the MIMO 4 × 4 system, underperform in some scenarios due to low Rician K factors and the attainable channel estimation accuracy. Precisely, a near-zero Rician K is present in the new MIMO 2 × 2 channel model, what leads to observe worse

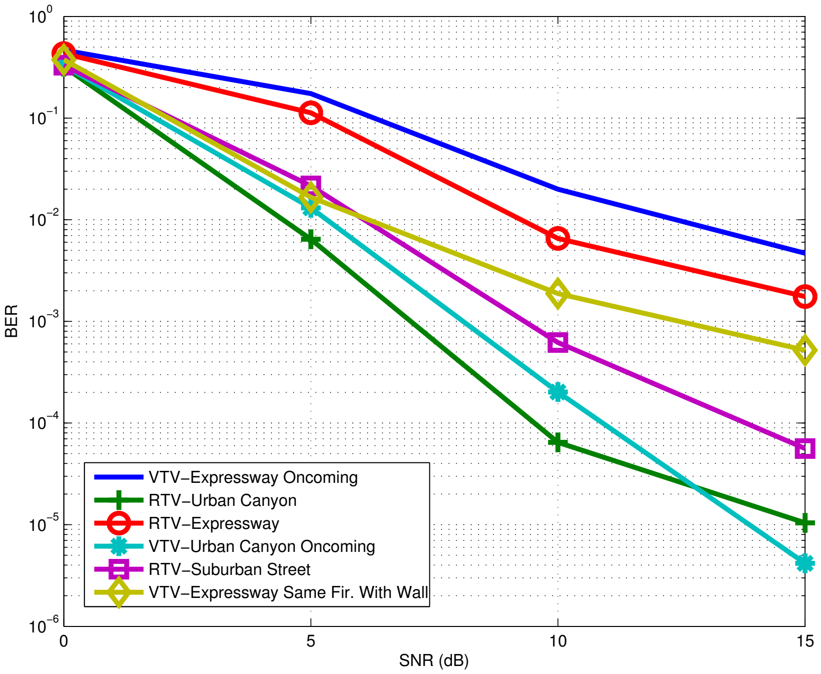

Figure 6.20: BER performance comparison among the different MIMO 4 × 4 vehicular channels.

performance results than when transmitting over the other vehicular models.

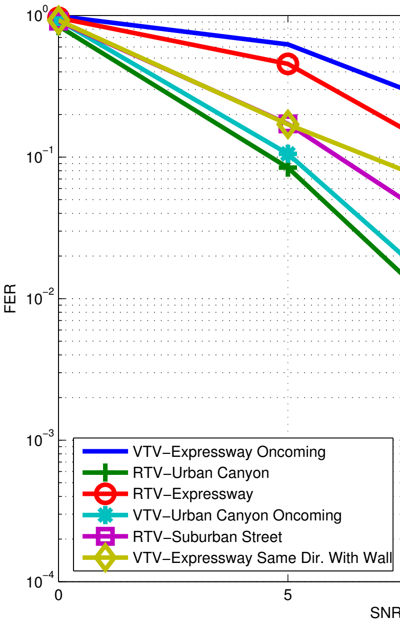

Figure 6.21: FER performance comparison among the different MIMO 4 × 4 vehicular channels.

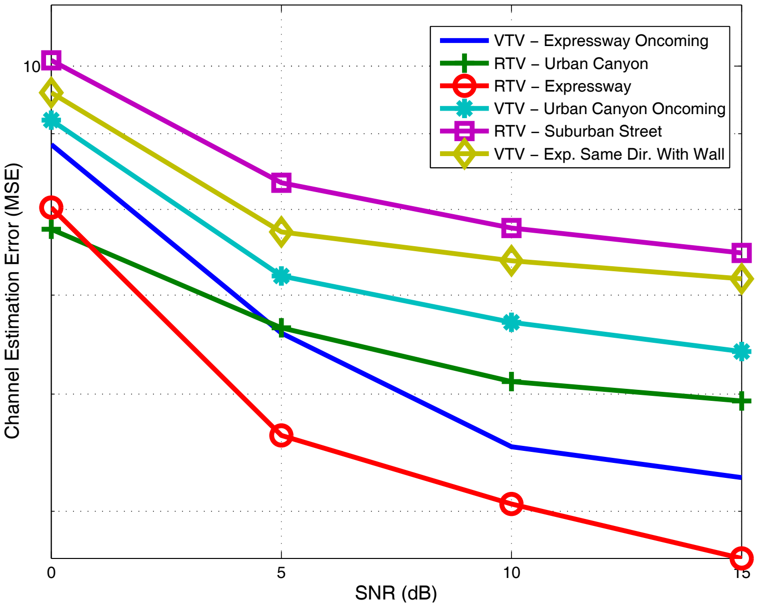

Figure 6.22: Channel estimation error in MIMO 4 × 4 vehicular scenarios.

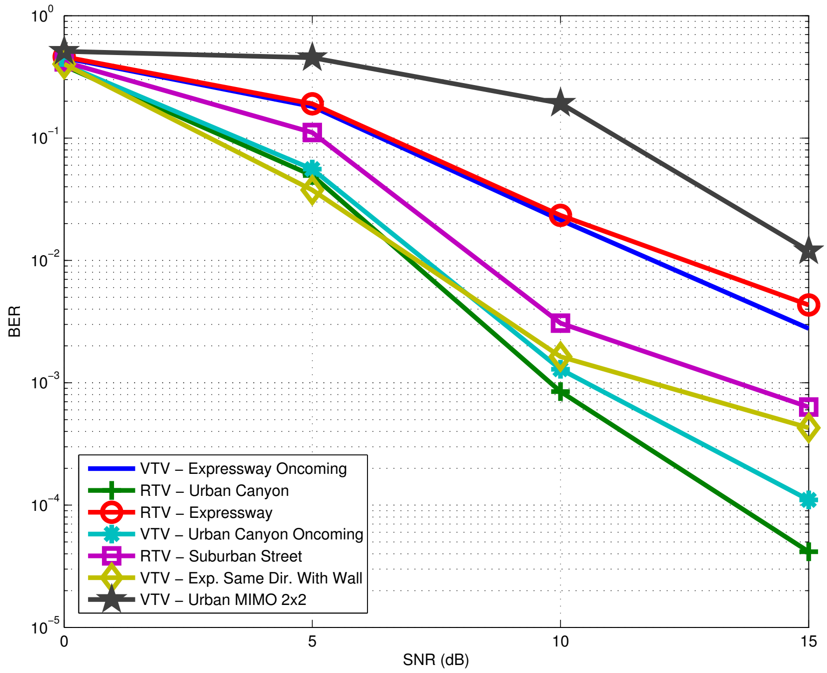

Figure 6.23: BER performance comparison among the different MIMO 2 × 2 vehicular channels.

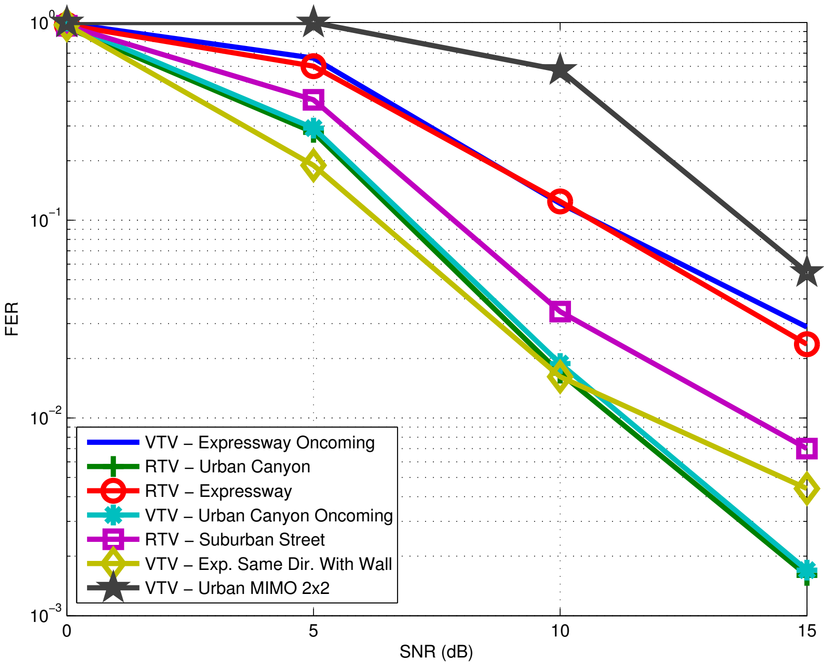

Figure 6.24: FER performance comparison among the different MIMO 2 × 2 vehicular channels.

7

Conclusions and Future Work

7.1 Conclusions

This work has presented a rapid prototyping system for evaluating vehicular communications in realistic scenarios. The evaluation system is composed of a hardware testbed, software transceivers and an FPGA-based vehicular channel emulator.

Before describing the vehicular testbed, in Chapter 2 it was described how to build a MIMO 2×2 testbed. As an example of the results that can be obtained using such platform, performance curves of several MIMO-STBC schemes were created. In particular, it was compared the Alamouti orthogonal scheme for two receive antennas with coherent and non-coherent demodulation. Two different channel estimation methods were considered: a conventional pilot-aided technique and a recently proposed blind algorithm based on second-order statistics. The results presented were obtained in three different scenarios. In all those cases the blind channel estimation technique provides BER performance similar to the pilot-aided method but with a slight increase in the effective data rate and a moderate increase in the computational complexity of the detector. On the other hand, the differential STBC presents, as expected, a 3 dB penalty with respect to coherent schemes.

Chapter 2 also exposed a multilayer software architecture for MIMO testbed end-user access. Such architecture fills the gap that currently exists between commercial hardware components and the most usual abstraction level required by researchers. The architecture layers overcome the limitations of developing software for the testbed. Instead of developing software using the application program interfaces provided by the manufacturer, the architecture supplies a high level interface access to the platform. It frees the researchers from having to acquire a deep knowledge of the testbed and its hardware. For instance, they can test easily new algorithms without having to develop completely new source code for the testbed, speeding up test tasks. To show the flexibility of the protocol designed several use cases were detailed. Such cases include usual researching tasks like the creation of performance measurements or other situations such as the transmission of data bursts, the implementation of a feedback channel or the deployment of a multiuser environment.

After building the MIMO testbed it was straightforward to optimize it to construct a testbed for vehicular environments: Flexvehd. Such platform (detailed in Chapter 3) uses hardware similar to the one that makes up the MIMO 2×2 platform, but it implements an IEEE 802.11p PHY-layer transceiver. To show the potential of the testbed, the performance of the transceiver was evaluated on several scenarios where it was artificially added typical characteristics of a vehicular environment, like Doppler shifts and Rayleigh fading.

In Chapter 4 it is shown how to create SISO and MIMO wireless software transceivers that implement the physical layers of IEEE 802.11p, IEEE 802.11a and IEEE 802.16e, which were evaluated in Chapters 5 and 6. Specifically, in Chapter 5 it was described a flexible, reconfigurable and cost-effective solution for evaluating wireless transceivers through the real-time emulation of vehicular channels. After presenting a comprehensive review of the current state of the art of commercial and academic channel emulators, it was detailed the design and implementation of a vehicular channel emulator that is based on the use of FPGA technology and rapid prototyping software tools. The channel emulator considers six different scenarios that include common environments such as highways, urban canyons and suburban areas.

Several examples of performance evaluation of the IEEE 802.11p, IEEE 802.11a and IEEE 802.16e PHY layers were obtained when transmitting over the six high-speed scenarios. It was also studied the effects of using coding, which obtains important gains in BER/PER, and the performance of the three transceivers was compared in Rayleigh and AWGN channels. The results obtained in non-vehicular channels illustrate the importance of obtaining a good estimation of the channel coefficients: Mobile WiMAX attains better results than IEEE 802.11p and IEEE 802.11a thanks to doubling the number of pilots per data subcarrier. Regarding the vehicular measurements, it can be concluded that the use of IEEE 802.11p improves dramatically the performance respect to IEEE 802.11a. Also, it could be observed that in the chosen experimental conditions, Mobile WiMAX outperforms IEEE 802.11p and IEEE 802.11a in most vehicular channels.

Finally, Chapter 6 was devoted to detail the design and development of a multiple-antenna IEEE 802.11p performance evaluation system made up of low-cost and flexible FPGA-based MIMO channel emulators that work in conjunction with the software transceivers described in Chapter 4. The performance of the PHY layers of SISO, SIMO and MIMO transceivers was measured when transmitting over seven different vehicular scenarios. With respect to the emulators, first it was described how to turn the SISO vehicular channel emulator created in Chapter 5 into a MIMO 4×4 system by using a time-multiplexed approach that occupies only 82% of the slices of a Xilinx Virtex IV. Additionally, it was shown how to reuse some of the stages of the upgraded SISO system to create a MIMO 2×2 channel emulator based on recently obtained MIMO 2×2 empirical data. These emulator designs are able to accelerate vehicular simulation speed between 6 and 209 times respect to the software-based approach.

Regarding the upgraded SISO vehicular measurements, it can be concluded that, in the chosen experimental conditions, SIMO and MIMO transceivers outperform SISO systems: they require less transmission power to attain the same BER/FER, they can reach higher data rates and they are able to improve drastically the SISO performance in many of the most common vehicular scenarios. Among the different transceivers it can be stated that the one that performs best is the SIMO 1×4, whereas more complex systems, such as the MIMO 4×4 system, underperform in some scenarios due to low Rician K factors and the attainable channel estimation accuracy. Precisely, a near-zero Rician K is present in the new MIMO 2×2 channel model, what leads to observe worse performance results than when transmitting over the other vehicular models.

7.2 Future Work

7.2.1 Implementation of Other SISO and MIMO Channels

After modeling the channels presented in (60) and (115), it is straightforward to implement other tapped-delay line models like the empirical vehicular models exposed in (127, 128, 145). The models to be implemented are not restricted to inter-vehicle communications: it would be straightforward to implement channel models for other environments like aircrafts (146) or for underwater communications (147).

Apart from tapped-delay line models, it would be also interesting to implement a GSCM model like (119). There, the authors describe extensively a geometry-based stochastic MIMO model for vehicular communications created combining empirical measurements and physical developments. With such model, it is possible to recreate vehicle-to-vehicle scenarios choosing from a wide range of vehicular parameters (coordinates of the vehicles, moving speed, direction, discrete and diffuse scatterers, etc.).

7.2.2 Evaluation of Transceivers in Vehicular Environments

This Thesis has exposed the evaluation of three different software transceivers that implement some of the standards that are more likely to be used in vehicular scenarios (IEEE 802.16e, IEEE 802.11p and IEEE 802.11a). Since IEEE 802.11p is probably the best positioned standard to be implemented in real VTV communications (at least for providing safety services), the evaluation system developed can be used as a basis for designing and testing new algorithms and techniques that improve IEEE 802.11p performance. The use of multiple antennas in transmission and/or reception was proposed in Chapter 4, but there are other areas of the current standard that might be enhanced (note that IEEE 802.11p is mostly based on the IEEE 802.11a standard, which was released in 1999 and since then a lot of research has been carried out in the field of wireless communications).

Furthermore, there are other recent standards that would be really interesting to evaluate in the implemented vehicular channels, like LTE or IEEE 802.16m (Mobile WiMAX Release 2), whose implementation would be relatively easy to perform and test in the evaluation system proposed.

7.2.3 BER/FER Input for Vehicular Simulators

The results obtained during the experiments performed in this work have allowed obtaining BER, FER and PER curves. The values of such curves can be used for further research as input parameters for simulators that are commonly used in vehicular network research (like NCTUns (148), EstiNet (149) or ns-2 (150)).

7.2.4 Evaluation of Infotainment Services Performance

Since IEEE 802.11p has been optimized for providing safety services, it is interesting to determine which non-safety services can provide. Note that such non-safety applications do not have such tight time constraints but they may need higher data transfer rates, which implies that transceivers should be enhanced.

Non-safety applications include mobile internet access, automatic toll collection, traffic light status notification, speed limit and cruise control, roadsign recognition, travel information management, etc. Among these applications video streaming is currently a hot topic, especially since the release of the H.264 codec. Nowadays H.264 is one of the most used formats for recording, compressing and distributing high definition video and, therefore, its performance evaluation in vehicular scenarios using the rapid prototyping tools presented in this Thesis can be useful for wireless carriers and video content distributors.

7.2.5 Web-Controlled Vehicular Platform

The evaluation system presented contains an FPGA-based channel emulator that can be placed into a PCI slot. Currently the system can be managed through a remote management application. The next step would consist in offering an online manager that would allow remote researchers to send and process the signals generated by their transceivers. Thus, a web interface would show several configuration options to choose the channel model and certain parameters.

A preliminary study of the state of the art on this topic has shown that there is only one similar initiative in the academic field (139). There, the authors implemented an emulation system that is integrated with a web-based GUI (Graphical User Interface) system which allows the users to control the emulator interface. Through the web-based GUI, users can specify the parameters of the channel to be emulated and start/stop the emulation process. The input and output waveform can be uploaded or downloaded by the registered users through FTP. This is a good approximation to offer remote channel emulator control, but it could easily improved thanks to the latest web technologies and the features of the control software described in Chapter 2.

References

[1] IEEE 802.11 Group Project Timelines. `http://grouper.ieee.org/groups/802/11/Reports/802.11_Timelines.htm`, Last access: July 2011. 1

[2] D. Ramírez et al. **A comparative study of STBC transmissions at 2.4 GHz over indoor channels using a 2 × 2 MIMO testbed**. *Wireless Communications and Mobile Computing*, **8**(9):1149–1164, Nov. 2008. 3

[3] D. Ramírez et al. **A flexible testbed for the rapid prototyping of MIMO baseband modules**. In *Proc. of 3rd International Symposium on Wireless Communication Systems*, Valencia, Spain, September 2006. 4

[4] J. A. García-Naya et al. **Performance of STBC transmissions with real data**. In *Proc. of the 16th IST Mobile and Wireless Communications Summit*, 2007. 4

[5] José A. García-Naya et al. **A flexible MIMO testbed developed at the University of A Coruña**. In *Proc. of CMCS*, 2007. 4, 17

[6] D. Ramírez et al. **Plataforma hardware para el desarrollo de sistemas MIMO**. In *Proc. of URSI*, 2006. 4

[7] J. A. García-Naya et al. **A distributed multilayer architecture enabling end-user access to MIMO testbeds**. In *Proc. of PIMRC*, 2008. 4

[8] T. M. Fernández Caramés et al. **MIMO testbed middleware for transmission automation**. In *Proc. of ELMAR*, 2008. 4

[9] José A. García Naya et al. **Arquitectura multicapa distribuida para demostradores MIMO**. In *Proc. of URSI*, 2008. 4

[10] T. M. Fernández Caramés et al. **FlexVehd: a flexible testbed for vehicular radio interfaces**. In *Proc. of ITST*, 2008. 4, 65

[11] T. M. Fernández Caramés et al. **Rapid prototyping for evaluating vehicular communications**. In *Rapid prototyping / Book 2, INTECH*, ISBN: 979-953-307-048-2, 2011. 4

[12] T. M. Fernández Caramés, M. González López, and L. Castedo. **FPGA-based vehicular channel emulator for real-time performance evaluation of IEEE 802.11p transceivers**. *EURASIP Journal on Wireless Communications and Networking*, **2010**:18 pages. 4

[13] T. M. Fernández-Caramés, M. González-López, and L. Castedo. **Mobile WiMAX for vehicular applications: Performance evaluation and comparison against IEEE 802.11p/a**. *Computer Networks*, 2011. 4

[14] T. M. Fernández-Caramés, M González-López, and L. Castedo. **FPGA-based vehicular channel emulator for evaluation of IEEE 802.11p transceivers**. In *Proc. of ITST*, 2009. 4

[15] Pedro Suárez Casal et al. **Diseño FPGA de un emulador de canal para WiMAX**. In *Proc. of URSI*, 2008. 4

[16] T. M. Fernández-Caramés et al. **Performance Evaluation of Multiple-Antenna IEEE 802.11p Transceivers Using an FPGA-based MIMO Vehicular Channel Emulator**. *Submitted to EURASIP Journal on Wireless Communications and Networking*, 2011. 4

[17] IEEE. *IEEE Standard for Information technology-Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC)and Physical Layer (PHY) Specifications. Amendment 6: Wireless Access in Vehicular Environments*, July 2010. 7, 41, 48, 103

[18] R.M. Rao et al. **Multi-Antenna Testbeds for Research and Education in Wireless Communications**. *Communications Magazine*, **42**(12):72–81, Dec. 2004. 7

[19] X. Nieto, L. M. Ventura, and A. Mollfulleda. **GEDOMIS: a broadband wireless MIMO-OFDM testbed, design and implementation**. In *Proc. of TRIDENTCOM*, 2006. 7, 18, 19

[20] S. Caban et al. **Vienna MIMO Testbed**. *EURASIP Journal on Applied Signal Processing*, **Vol. 2006**. 7, 18, 19, 67

[21] M. Rupp, S. Caban, and C. Mehlführer. **Challenges in Building MIMO Testbeds**. In *Proc. of the European Signal Processing Conference*, 2007. 8

[22] P. W. Wolniansky et al. **V-Blast: an architecture for realizing very high data rates over the rich-scattering wireless channel**. *In Proc. of International Symposium on Signals, Systems, and Electronics*, 1998. 18

[23] A. Adjoudani et al. **Prototype experience for MIMO BLAST over third-generation wireless system**. *IEEE Journal on Selected Areas in Communications*, **21**(3):440–451, 2003. 18

[24] D. Garrett et al. **A 28.8 Mb/s 4 × 4 MIMO 3G high-speed downlink packet access receiver with normalized least mean square equalization**. In *Proc. of IEEE International Solid-State Circuits Conference*, 2004. 18

[25] S. H. Won, D. S. Lyu, and H. J. Park. **Physical layer implementation and evaluation of multiple input multiple output - orthogonal frequency division multiplexing (MIMO-OFDM) system**. *In Proc. of International Conference on Communication Technology*, 2003. 18

[26] J.W. Wallace, B.D. Jeffs, and M.A. Jensen. **A real-time multiple antenna element testbed for MIMO algorithm development and assessment**. In *Proc. of Antennas and Propagation Society International Symposium*, 2004. 18

[27] E. Aschbacher et al. **Design of a Flexible and Scalable 4 x 4 MIMO Testbed**. In *Proc. of the 11th Digital Signal Processing Workshop*, 2004. 18, 19

[28] D. Borkowski et al. **SABA: A Testbed for a Real-Time MIMO System**. *EURASIP Journal on Applied Signal Processing*, **Vol. 2006**. 18, 19

[29] R. Morawski, T. Le-Ngoc, and O. Naeem. **Wireless and wireline MIMO testbed**. In *Proc. of CCECE*, 2003. 18

[30] P. Murphy et al. **An FPGA based rapid prototyping platform for MIMO systems**. In *Proc. of the Thirty-Seventh Asilomar Conference on Signals, Systems and Computers*, 2003. 18, 19

[31] A. Guillén i Fábregas et al. **A MIMO-OFDM Testbed for Wireless Local Area Networks**. *EURASIP Journal on Applied Signal Processing*, **Vol. 2006**. 18

[32] S. Lang, M.R. Rao, and B. Daneshrad. **Design and development of a 5.25 GHz software defined wireless OFDM communication platform**. *Communications Magazine, IEEE*, **42**(6):S6–12, 2004. 18

[33] N. Chiurtu et al. **Software-defined radio implementation of multiple antenna systems using low-density parity-check codes**. In *Proc. of the Wireless Communications and Networking Conference*, 2005. 18, 19

[34] W. Zhu, D. Browne, and M. Fitz. **An Open Access Wideband Multi-Antenna Wireless Testbed with Remote Control Capability**. In *Proc. of TRIDENTCOM*, 2005. 18, 19

[35] G. Foschini and M. Gans. **On limits of wireless communications in a fading environment when using multiple antennas**. *Wireless Personal Communications*, **6**:311–335, 1998. 26, 46, 103

[36] I. E. Telatar. **Capacity of multi-antenna Gaussian channels**. *European Transactions on Telecommunications*, **10**(6):585–595, Nov.-Dec. 1999. 26, 46, 103

[37] A. F. Naguib, N. Seshadri, and A. R. Calderbank. **Increasing data rate over wireless channels**. *IEEE Signal Processing Magazine*, **17**(3):76–92, May 2000. 26

[38] D. Gesbert et al. **From Theory to Practice: an Overview of MIMO Space-Time Coded Wireless Systems**. *IEEE Journal on Selected Areas in Communications*, **21**(3):281–302, 2003. 26, 151

[39] L. Zheng and D. Tse. **Diversity and multiplexing: A fundamental tradeoff in multiple-antenna channels**. *IEEE Transactions on Information Theory*, **49**(5):1073–1096, 2003. 26

[40] A. J. Paulraj et al. **An overview of MIMO communications - A key to Gigabit wireless**. *Proceedings of the IEEE*, **92**(2):198–218, Feb. 2004. 26

[41] S.M. Alamouti. **A simple transmit diversity technique for wireless communications**. *IEEE Journal on Selected Areas in Communications*, **45**(9):1451–1458, 1998. 26, 28, 55, 152

[42] V. Tarokh, H. Jafarkhani, and A. R. Calderbank. **Space-Time block codes from orthogonal designs**. *IEEE Transactions on Information Theory*, **45**(5):1456–1467, 1999. 26

[43] G. Ganesan and P. Stoica. **Space-Time block codes: A maximum SNR approach.** *IEEE Transactions on Information Theory*, **47**(4):1650–1656, 2001. 27

[44] A. F. Naguib et al. **A space-time coding modem for high-data-rate wireless communications**. *IEEE Journal on Selected Areas in Communications*, **16**(8):1459–1478, Oct. 1998. 27, 28, 153

[45] V. Tarokh and H. Jafarkhani. **A differential detection scheme for transmit diversity**. *IEEE Journal on Selected Areas in Communications*, **18**(7):1169–1174, Jul. 2000. 27, 28

[46] B. L. Hughes. **Differential Space-Time modulation**. *IEEE Transactions on Information Theory*, **46**(7):2567–2578, Nov 2000. 27

[47] B. Hochwald and W. Sweldens. **Differential Unitary Space-Time Modulation**. *IEEE Transactions on Communications*, pages 2041–2052, Dec. 2000. 27

[48] B. Hochwald and T. Marzetta. **Unitary space-time modulation for multiple-antenna communications in Rayleigh flat fading**. *IEEE Transactions on Information Theory*, **46**(2):543–564, Mar. 2000. 27

[49] B. Hochwald et al. **Systematic Design of Unitary Space-Time Constellations**. *IEEE Transactions on Information Theory*, **46**(6):1962–1973, 2000. 27

[50] C. Budianu and L. Tong. **Channel estimation for Space-Time Orthogonal Block Codes**. *IEEE Transactions on Signal Processing*, **50**(10):2515–2528, Oct. 2002. 27

[51] P. Stoica and G. Ganesan. **Space-Time block codes: Trained, blind, and semi–blind detection**. *Digital Signal Processing*, **13**:93–105, Jan. 2003. 27

[52] S. Shahbazpanahi, A. B. Gershman, and J. H. Manton. **Closed-form blind MIMO channel estimation for orthogonal space-time block codes**. *IEEE Transactions on Signal Processing*, **53**(12):4506–4517, Dec. 2005. 27, 28, 154, 155, 156

[53] J. Vía, I. Santamaría, and Jesús Pérez. **A Sufficient Condition for Blind Identifiability of MIMO-OSTBC Channels Based on Second Order Statistics**. In *Proc. of the Seventh IEEE Workshop on Signal Processing Advances in Wireless Communications*, 2006. 28, 154, 156

[54] ASTM Intl. *Standard Specification for Telecommunications and Information Exchange Between Roadside and Vehicle Systems 5 GHz Band Dedicated Short Range Communications (DSRC) Medium Access Control (MAC) and Physical Layer (PHY) Specifications (E2213-03)*, Sep. 2003. 44, 46, 48, 87

[55] T. L. Doumi. **Spectrum considerations for public safety in the United States**. *IEEE Communications Magazine 44*, **1**:30–37, 2006. 45, 46

[56] C. Ribeiro. **Bringing wireless access to the automobile: a comparison of Wi-Fi, WiMAX, MBWA, and 3G**. In *Proc. of the 21st annual Rensselaer at Hartford Computer Science Conference*, 2005. 45

[57] B. Wang, I. Sen, and D. Matolak. **Performance evaluation of 802.16e in vehicle to vehicle channels**. In *Proc. of VTC*, 2007. 45, 46, 59

[58] M. Wellens, B. Westphal, and P. Mähönen. **Performance evaluation of IEEE 802.11-based WLANs in vehicular scenarios**. In *Proc. of VTC-Spring*, 2007. 46

[59] L. Cheng et al. **A measurement study of time-scaled 802.11a waveforms over the mobile-to-mobile vehicular channel at 5.9 GHz**. *IEEE Communications Magazine*, **46**(5):84–91, 2008. 46

[60] G. Acosta-Marum and M. A. Ingram. **Six time- and frequency-selective empirical channel models for vehicular wireless LANs**. In *Proc. of VTC-Fall*, 2007. 46, 65, 69, 71, 75, 81, 87, 106, 107, 114, 116, 137

[61] 3GPP. *3GPP TS 25.308 v5.7.0, Technical specification group radio access network - High Speed Downlink Packet Access (HSDPA): overall description - Stage 2 (Release 5)*, Dec. 2004. 46

[62] IEEE. *Air interface for mobile broadband wireless access systems supporting vehicular mobility-physical and media access control layer specification, (IEEE 802.20-2008)*, Aug. 2008. 46

[63] 3GPP. *3GPP TS 41.101 v7.2.0, Technical specifications and technical reports for a GERAN-based 3GPP system (Release 7)*, Dec. 2008. 46

[64] W. Bolton, Y. Xiao, and M. Guizani. **IEEE 802.20: mobile broadband wireless access**. *IEEE Wireless Communications*, **14**(1):84–95, 2007. 46

[65] Ericsson. **White paper: The evolution of EDGE**. Technical report, Sep. 2009. 46

[66] IEEE. *IEEE 802.11a: Wireless LAN medium access control and physical layer specifications: high-speed physical layer in the 5 GHz band*, Sep. 1999. 46, 48

[67] WiMAX Forum. **Requirements and Recommendations for WiMAX Forum Mobility Profiles**. Technical report, Nov. 2005. 46, 100

[68] 3GPP. **3GPP TS 36.201 v8.3.0, LTE physical layer - general description (Release 8)**, March 2009. 46

[69] IEEE. *Amendment to IEEE standard for local and metropolitan area networks - Part 16: air interface for broadband wireless access systems - advanced air interface (IEEE 802.16m Draft 5.0)*, April 2010. 46

[70] IEEE. *IEEE 802.11: Wireless LAN medium access control and physical Layer specifications amendment 5: enhancements for higher throughput (IEEE 802.11n-2009)*, Oct. 2009. 46

[71] IEEE. *IEEE Standard for Information technology - Telecommunications and information exchange between systems-Local and metropolitan area networks - Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, June 2007. 48

[72] D. Jiang et al. **Design of 5.9 GHz DSRC-based vehicular safety communication**. *IEEE Wireless Communications*, **13**(5):36–43, Oct. 2006. 48

[73] D. Jiang and L. Delgrossi. **IEEE 802.11p: towards an international standard for wireless access in vehicular environments**. In *Proc. of VTC-Spring*, 2008. 48

[74] R. Uzcátegui and G. Acosta-Marum. **WAVE: A Tutorial**. *IEEE Communications Magazine*, **47**(5):126–133, 2009. 48

[75] Y. Wang et al. **IEEE 802.11p performance evaluation and protocol enhancement**. In *Proc. of International Conference on Vehicular Electronics and Safety*, 2008. 48

[76] S. Eichler. **Performance evaluation of the IEEE 802.11p WAVE communication standard**. In *Proc. of VTC-Fall*, 2007. 48

[77] M. Abbassi. *Characterization of a 5 GHz modular radio frontend for WLAN based on IEEE 802.11p*. Master's thesis, Dec. 2008. 48

[78] J. Reddy et al. **On equalization for OFDM-dedicated short range communication (DSRC) modem**. In *Proc. of ICPWC*, 2000. 48

[79] S. Sibecas et al. **Pseudo-pilot OFDM scheme for 802.11a and R/A in DSRC applications**. In *Proc. of VTC-Fall*, 2003. 48

[80] Y. Cheng, Y. Lu, and C. Liu. **Adaptive channel equalizer for wireless access in vehicular environments**. In *Proc. of ITST*, 2006. 48

[81] H. Abdulhamid, K. E. Tepe, and E. Abdel-Raheem. **Iterative channel-tracking techniques for 5.9 GHz DSRC applications**. *Research Letters in Communications*, **Vol. 2008**. 48

[82] J. Mar and C. Kuo. **Performance improvement of the DSRC system using a novel S and PI-decision demapper**. In *Proc. of ICC*, 2008. 48

[83] N. Sasho et al. **Single-chip 5.8 GHz DSRC transceiver with dual mode of ASK and $\pi/4$-QPSK**. In *Proc. of the Radio and Wireless Symposium*, 2008. 49

[84] S. Shin et al. **0.18um CMOS integrated chipset for 5.8 GHz DSRC systems with +10dBm output power**. In *Proc. of ISCAS*, 2008. 49

[85] N. Almeida et al. **Low cost transceiver for DSRC applications**. In *Proc. of the Asia-Pacific Microwave Conference*, 2006. 49

[86] T. Tsuboi et al. **Dual receiver communication system for DSRC**. In *Proc. of the International Conference on Future Generation Communication and Networking*, 2008. 49

[87] J. Mar et al. **Realization of OFDM modulator and demodulator for DSRC vehicular communication system using FPGA chip**. In *Proc. of ISPACS*, 2006. 49

[88] J. Mar et al. **Design of software-defined radio channel simulator for wireless communications: case study with DSRC and UWB channels**. *IEEE Transactions on Instrumentation and Measurement*, **58**(8):2755–2766, Aug. 49

[89] L. Rugini, P. Banelli, and G. Leus. **Simple equalization of time-varying channels for OFDM**. *IEEE Communications Letters*, **9**(7):619–621, 2005. 52, 64

[90] H. Jafarkhani. **A quasi-orthogonal space-time block code**. *IEEE Transactions on Communications*, **49**(1):1–4, 2001. 55

[91] A. Goldsmith. *Wireless Communications*. Cambridge University Press, NY, USA, 2005. 55

[92] E. Piri et al. **Empirical evaluation of mobile WiMAX with MIMO**. In *Proc. of the Fifth IEEE Broadband Wireless Access Workshop*, 2009. 55

[93] D. Kim et al. **Performance measurement over Mobile WiMAX/IEEE 802.16e network**. In *Proc. of WoWMoM*, 2008. 55, 56

[94] D. Aphiraksatyakul, B. Seet, and C. Lau. **Evaluation of terrain effects on Mobile WiMAX in a vehicular environment**. In *Proc. of ITST*, 2008. 58

[95] I. Ahmad, D. Habibi, and Z. Rahman. **An improved FEC scheme for mobile wireless communication at vehicular speeds**. In *Proc. of ATNAC*, 2008. 58

[96] D. W. Matolak, I. Sen, and W. Xiong. **Channel modeling for V2V communications**. In *Proc. of V2VCOM*, 2006. 59

[97] M. Tran et al. **Mobile WiMAX: performance analysis and comparison with experimental results**. In *Proc. of VTC-Fall*, 2008. 59

[98] 3GPP. *3GPP TR 25.996 v6.1.0, Spatial channel model for multiple input multiple output (MIMO) simulations*, Sep. 2003. 59, 107

[99] G. Yeung et al. **Detailed OFDM Modeling in Network Simulation of Mobile Ad Hoc Networks**. In *Proc. of the 18th Workshop on Parallel and Distributed Simulation*, 2004. 60

[100] M. Takai, J. Martin, and R. Bragodia. **Effects of wireless physical layer modeling in mobile ad hoc networks**. In *Proc. of the 2nd ACM international symposium on Mobile ad hoc networking and computing*, 2001. 60

[101] IEEE. *Air interface for fixed broadband wireless access systems*, May 2009. 61, 62, 153

[102] R. Mangharam et al. **A multi-hop mobile networking test-bed for telematics**. In *Proc. of Society for Automotive Engineers (SAE) World Congress*, 2005. 65

[103] J. Dulmage et al. **COTS-based DSRC testbed for rapid algorithm development, implementation, and test**. In *Proc. of WiNTECH*, 2006. 65

[104] G. Acosta-Marum. *Measurement, modelling and OFDM synchronization for the wideband mobile-to-mobile channel*. PhD thesis, May 2007. 65, 69, 70, 71, 79

[105] Spirent Communications. `http://www.spirent.com`, Last Access: July 2011. 67, 107

[106] Rhode & Schwarz. `http://www2.rohde-schwarz.com`, Last access: July 2011. 67, 107

[107] Azimuth Systems. `http://www.azimuthsystems.com`, Last access: July 2011. 67, 107

[108] Agilent Technologies. `http://www.home.agilent.com`, Last Access: July 2011. 67, 107

[109] A. Ghazel et al. **Design and performances analysis of high speed AWGN communication channel emulator**. *Analog Integrated Circuits and Signal Processing*, **34**:133–142, 2003. 68

[110] A. Alimohammad et al. **An accurate and compact Rayleigh and Rician fading channel simulator**. In *Proc. of VTC-Spring*, 2008. 68

[111] J. Hwang et al. **Fast FPGA prototyping of a multipath fading channel emulator via high-level design**. In *Proc. of ISCIT*, 2007. 68

[112] C. Mehlfürer et al. **A scalable rapid prototyping system for real-time MIMO OFDM transmissions**. In *Proc. of IEE/EURASIP Conference on DSP enabled Radio*, 2005. 68

[113] F. Kaltenberger et al. **A multi-band development platform for rapid prototyping of MIMO systems**. In *Proc. of the ITG/IEEE Workshop on Smart Antennas*, 2005. 69

[114] J. Hwang et al. **Efficient structure for FPGA implementation of a configurable multipath fading channel emulator**. In *Proc. of ISPACS*, 2006. 69

[115] G. Acosta-Marum, B. T. Walkenhorst, and R. J. Baxley. **An Empirical Doubly-Selective Dual-Polarization Vehicular MIMO Channel Model**. In *Proc. of VTC-Spring*, 2010. 105, 107, 108, 114, 115, 137

[116] A. S. Akki and F. Haber. **A statistical model for mobile-to-mobile land communication channel**. *IEEE Transactions on Vehicular Technology*, **35**(1):32–38, 1986. 106

[117] J. Fuhl, A. F. Molisch, and E. Bonek. **Unified channel model for mobile radio systems with smart antennas**. *IEE Proceedings - Radar, Sonar and Navigation*, **145**(1):32–41, 1998. 106

[118] A. F. Molisch et al. **Geometry-based directional model for mobile radio channels-principles and implementation**. *European Transactions on Telecommunications*, **14**(4):351–359, 2003. 106

[119] J. Karedal et al. **A geometry-based stochastic MIMO model for vehicle-to-vehicle communications**. *IEEE Transactions on Wireless Communications*, **8**(7):3646–3657, 2009. 106, 107, 138

[120] J. Maurer et al. **A new inter-vehicle communications (IVC) channel model**. In *Proc. of VTC-Fall*, 2004. 106

[121] X. Zhao et al. **Characterization of Doppler spectra for mobile communications at 5.3 GHz**. *IEEE Transactions on Vehicular Technology*, **52**(1):14–23, 2003. 106

[122] O. Renaudin et al. **Non-stationary narrowband MIMO inter-vehicle channel characterization in the 5-GHz band**. *IEEE Transactions on Vehicular Technology*, **59**(4):2007–2015, 2010. 106

[123] G. Acosta-Marum and M. A. Ingram. **A BER-based partitioned model for a 2.4 GHz vehicle-to-vehicle expressway channel**. *Wireless Personal Communications*, **37**(3):421–443, 2006. 106

[124] A. G. Zajic and G. L. Stüber. **Space-time correlated mobile-to-mobile channels: modelling and simulation**. *IEEE Transactions on Vehicular Technology*, **57**(2):715–726, 2008. 106

[125] A. G. Zajic and G. L. Stüber. **Three-dimensional modeling, simulation, and capacity analysis of space-time correlated mobile-to-mobile channels**. *IEEE Transactions on Vehicular Technology*, **57**(4):2042–2054, 2008. 106

[126] A. G. Zajic et al. **Wideband MIMO mobile-to-mobile channels: geometry-based statistical modeling with experimental verification**. *IEEE Transactions on Vehicular Technology*, **58**(2):517–534, 2009. 106

[127] I. Sen and D. W. Matolak. **Vehicle-Vehicle channel models for the 5-GHz band**. *IEEE Transactions on Intelligent Transportation Systems*, **9**(2):235–245, 2008. 106, 137

[128] Q. Wu, D. W. Matolak, and I. Sen. **5-GHz-band vehicle-to-vehicle channels: models for multiple values of channel bandwidth**. *IEEE Transactions on Vehicular Technology*, **59**(5):2620–2625, 2010. 106, 137

[129] X. Cheng et al. **An adaptive geometry-based stochastic model for non-isotropic MIMO mobile-to-mobile channels**. *IEEE Transactions on Wireless Communications*, **8**(9):4824–4835, 2009. 106

[130] T. Tank and J. Linnartz. **Vehicle to vehicle communications for AVCS platooning**. *IEEE Transactions on Vehicular Technology*, **46**(2):528–536, 1997. 106

[131] M. Pätzold, B. O. Hogstad, and N. Youssef. **Modeling, analysis, and simulation of MIMO mobile-to-mobile fading channels**. *IEEE Transactions on Wireless Communications*, **7**(2):510–520, 2008. 106

[132] P. Petrus, J. H. Reed, and T. S. Rappaport. **Geometrical-based statistical macrocell channel model for mobile environments**. *IEEE Transactions on Communications*, **50**:495–502, 2002. 106

[133] Y. R. Zheng. **A non-isotropic model for mobile-to-mobile fading channel simulations**. In *Proc. of Milcom*, 2006. 106

[134] V. Erceg et al. *TGn channel models (Technological report for IEEE P802.11)*, May 2004. 107, 108

[135] V. Erceg et al. *Channel models for fixed wireless applications (Contribution IEEE 802.16.3c-01/29r4 to IEEE 802.16 Broadband Wireless Access Working Group)*, June 2003. 107

[136] Propsim. `http://www.propsim.net`, Last access: July 2011. 107

[137] Z. Zhan et al. **A generalized hardware implementation of MIMO fading channels**. In *Proc. of ISCIT*, 2009. 108, 109

[138] F. Ren and Y. R. Zheng. **A novel emulator for discrete-time MIMO triply selective fading channels**. *IEEE Transactions on Circuits and Systems*, **57**(9):2542–2551, 2010. 108

[139] T. Wang, C. H. Liao, and T. D. Chiueh. **A real-time digital baseband MIMO channel emulation system**. In *Proc. of ISCAS*, 2007. 108, 139

[140] O. A. Nasr and B. Daneshrad. **Design and FPGA implementation an accurate real time 3x4 MIMO channel emulator**. In *Proc. of ACSSC*, 2009. 108, 109

[141] H. Eslami, S. V. Tran, and A. M. Eltawil. **Design and implementation of a scalable channel emulator for wideband MIMO systems**. *IEEE Transactions on Vehicular Technology*, **58**(9):4698–4709, 2009. 108, 109

[142] M. Cui, M. Hidekazu, and A. Kiyomichi. **FPGA implementation of 4×4 MIMO test-bed for spatial multiplexing systems**. In *Proc. of PIMRC*, 2004. 108

[143] A. Dassatti et al. **High performance channel model hardware emulator for 802.11n**. In *Proc. of IEEE FPT*, 2005. 108, 109

[144] D. N. Dung et al. **Implementation and evaluation of 4 4 MIMO fading simulator considering antenna characteristics**. In *Proc. of ICCE*, 2006. 108

[145] A. Paier et al. **Car-to-car radio channel measurements at 5 GHz: Pathloss, power-delay profile, and delay-Doppler spectrum**. In *Proc. of ISWCS*, 2007. 137

[146] D. W. Matolak and A. Chandrasekaran. **Aircraft intra-vehicular channel characterization in the 5 GHz band**. In *Proc. of ICNS*, 2008. 137

[147] M. C. Domingo. **Overview of channel models for underwater wireless communication networks**. *Physical Communication*, **1**(3):163–182, 2008. 137

[148] NCTUns. `http://nsl.csie.nctu.edu.tw/nctuns.html`, Last access: July 2011. 138

[149] EstiNet. `http://www.estinet.com`, Last access: July 2011. 138

[150] ns2. `http://nsnam.isi.edu/nsnam/index.php/User_Information`, Last access: July 2011. 138

[151] E. G. Larsson, P. Stoica, and G. Ganesan. *Space-Time Block Coding for Wireless Communications*. Cambridge University Press, New York, USA, 2003. 151, 152, 156, 157

[152] U. Fincke and M. Pohst. **Improved methods for calculating vectors of short length in a lattice, including a complexity analysis**. *Mathematics of Computation*, **44**:463–471, Apr. 1985. 151

[153] O. Damen, A. Chkeif, and J. Belfiore. **Lattice code decoder for space-time codes**. *IEEE Communication Letters*, **4**(5):161–163, May 2000. 151

[154] J. Jaldén, C. Martin, and B. Ottersten. **Semidefinite Programming for Detection in Linear Systems – Optimality Conditions and Space-Time Decoding**. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2003. 151

[155] A. Hottinen et al. **Industrial embrace of smart antennas and MIMO**. *IEEE Wireless Communications*, **9**:8–16, 2006. 153

[156] J. Vía and I. Santamaría. **On the Blind Identifiability of Orthogonal Space-Time Block Codes from Second Order Statistics**. *IEEE Transactions On Information Theory.* 154, 156

[157] J. Vía et al. **Blind Decoding of MISO-OSTBC Systems based on Principal Component Analysis**. In *Proc. of the IEEE International Conference on Acoustic, Speech, and Signal Processing*, 2006. 154, 155

Appendices

A

Data Model and Characteristics of the MIMO-STBC System

Notation

In this Appendix it will be used bold-faced upper case letters to denote matrices, e.g., $\mathbf{X}$, with elements $x_{i,j}$; bold-faced lower case letters for column vectors, e.g., $\mathbf{x}$, and light-face lower case letters for scalar quantities. The superscripts $(\cdot)^T$ and $(\cdot)^H$ denote transpose and Hermitian, respectively. The real and imaginary parts of a complex quantity will be denoted as $\Re(\cdot)$ and $\Im(\cdot)$, and superscript $\hat{(\cdot)}$ will denote estimated matrices, vectors or scalars.

The trace, range (or column space) and Frobenius norm of matrix $\mathbf{A}$ will be denoted as $\text{Tr}(\mathbf{A})$, $\text{range}(\mathbf{A})$ and $\|\mathbf{A}\|$, respectively. The notation $\mathbf{A} \in \mathbb{C}^{M\times N}$ and $\mathbf{A} \in \mathbb{R}^{M\times N}$ will be used to denote that $\mathbf{A}$ is a complex or real matrix of dimension $M \times N$. Finally, the identity matrix of dimensions $p \times p$ will be denoted as $\mathbf{I}_p$ (although the subindex will be omitted when confusion is not possible) and $E[\cdot]$ will denote the expectation operator.

A.1 Data Model for MIMO STBC systems

Throughout this Appendix it will be assumed a flat fading MIMO channel model with n_T transmit and n_R receive antennas. The $n_T \times n_R$ complex channel matrix is

$$\mathbf{H} = [\mathbf{h}_1 \cdots \mathbf{h}_{n_R}] = \begin{bmatrix} h_{1,1} & \cdots & h_{1,n_R} \\ \vdots & \ddots & \vdots \\ h_{n_T,1} & \cdots & h_{n_T,n_R} \end{bmatrix},$$

where $h_{i,j}$ denotes the channel response between the i-th transmit and the j-th receive antennas, and $\mathbf{h}_j$ contains the channel response associated with the j-th receive antenna.

It will be considered a space-time block code (STBC) transmitting M symbols during L time slots and using n_T antennas at the transmitter side. The transmission rate is defined as $R = M/L$ and the symbols of the n-th data block are denoted as $r_k[n]$, $k = 1, \ldots, M$. Depending on whether $r_k[n]$ is complex or real, the number of real symbols, M', transmitted in each block is

$$M' = \begin{cases} M & \text{for real constellations,} \\ 2M & \text{for complex constellations.} \end{cases}$$

For a STBC, the n-th block of data can be expressed in terms of the transmitted real symbols as

$$\mathbf{S}[n] = \sum_{k=1}^{M'} \mathbf{C}_k s_k[n],$$

where $\mathbf{C}_k \in \mathbb{C}^{L \times n_T}$, $k = 1, \ldots, M'$, are the STBC code matrices, and

$$s_k[n] = \begin{cases} \Re(r_k[n]), & k \leq M, \\ \Im(r_{k-M}[n]), & k > M, \end{cases}$$

are real symbols. In the case of real STBCs, the code matrices $\mathbf{C}_k$ and therefore the transmitted matrix $\mathbf{S}[n]$ are real.

The signal at the j-th receive antenna is

$$\mathbf{y}_j[n] = \mathbf{S}[n]\mathbf{h}_j + \mathbf{n}_j[n] = \sum_{k=1}^{M'} \mathbf{w}_k(\mathbf{h}_j) s_k[n] + \mathbf{n}_j[n],$$

where $\mathbf{n}_j[n]$ is spatial and temporally white complex noise with variance σ^2 and $\mathbf{w}_k(\mathbf{h}_j)$ represents the combined effect of the STBC and the j-th channel, which is given by

$$\mathbf{w}_k(\mathbf{h}_j) = \mathbf{C}_k \mathbf{h}_j,$$

for $k = 1, \ldots, M'$.

Taking into account the isomorphism between complex vectors $\mathbf{w}_k(\mathbf{h}_j)$ and real vectors $\tilde{\mathbf{w}}_k(\mathbf{h}_j) = \left[\Re(\mathbf{w}_k(\mathbf{h}_j))^T, \Im(\mathbf{w}_k(\mathbf{h}_j))^T\right]^T$ we can define the real-valued extended code matrices

$$\tilde{\mathbf{C}}_k = \begin{bmatrix} \Re(\mathbf{C}_k) & -\Im(\mathbf{C}_k) \\ \Im(\mathbf{C}_k) & \Re(\mathbf{C}_k) \end{bmatrix},$$

which imply

$$\tilde{\mathbf{w}}_k(\mathbf{h}_j) = \tilde{\mathbf{C}}_k \tilde{\mathbf{h}}_j, \tag{A.1}$$

with $\tilde{\mathbf{h}}_j = \left[\Re(\mathbf{h}_j)^T, \Im(\mathbf{h}_j)^T\right]^T$. Defining now the real vectors

$$\begin{aligned} \tilde{\mathbf{y}}_j[n] &= [\Re(\mathbf{y}_j[n])^T, \Im(\mathbf{y}_j[n])^T]^T \\ \tilde{\mathbf{n}}_j[n] &= [\Re(\mathbf{n}_j[n])^T, \Im(\mathbf{n}_j[n])^T]^T \end{aligned}$$

the above equation can be rewritten as

$$\begin{aligned} \tilde{\mathbf{y}}_j[n] &= \sum_{k=1}^{M'} \tilde{\mathbf{w}}_k(\mathbf{h}_j) s_k[n] + \tilde{\mathbf{n}}_j[n] \\ &= \tilde{\mathbf{W}}(\mathbf{h}_j)\mathbf{s}[n] + \tilde{\mathbf{n}}_j[n], \end{aligned}$$

where $\mathbf{s}[n] = [s_1[n], \ldots, s_{M'}[n]]^T$ contains the M' transmitted real symbols and $\tilde{\mathbf{W}}(\mathbf{h}_j) = [\tilde{\mathbf{w}}_1(\mathbf{h}_j) \cdots \tilde{\mathbf{w}}_{M'}(\mathbf{h}_j)]$. Finally, stacking all the received signals into $\tilde{\mathbf{y}}[n] = \left[\tilde{\mathbf{y}}_1^T[n], \ldots, \tilde{\mathbf{y}}_{n_R}^T[n]\right]^T$, we can write

$$\tilde{\mathbf{y}}[n] = \tilde{\mathbf{W}}(\mathbf{H})\mathbf{s}[n] + \tilde{\mathbf{n}}[n],$$

where $\tilde{\mathbf{W}}(\mathbf{H}) = \left[\tilde{\mathbf{W}}^T(\mathbf{h}_1) \cdots \tilde{\mathbf{W}}^T(\mathbf{h}_{n_R})\right]^T$, and $\tilde{\mathbf{n}}[n]$ is defined analogously to $\tilde{\mathbf{y}}[n]$.

When $\mathbf{H}$ is known at the receiver, and assuming a Gaussian distribution for the noise, the coherent Maximum Likelihood (ML) decoder amounts to minimizing the following criterion (151)

$$\hat{\mathbf{s}}[n] = \underset{\mathbf{s}[n]}{\operatorname{argmin}} \|\tilde{\mathbf{y}}[n] - \tilde{\mathbf{W}}(\mathbf{H})\mathbf{s}[n]\|^2,$$

subject to the constraint that the elements of $\hat{\mathbf{s}}[n]$ belong to a finite set $\mathcal{S}$. This is a NP-hard problem and optimal algorithms to solve it, such as *sphere decoding*, can be computationally expensive (38, 152, 153, 154).

A.2 Orthogonal STBCs

In the case of orthogonal STBCs (OSTBCs), the matrix $\tilde{\mathbf{W}}(\mathbf{H})$ satisfies

$$\tilde{\mathbf{W}}^T(\mathbf{H})\tilde{\mathbf{W}}(\mathbf{H}) = \|\mathbf{H}\|^2\mathbf{I}, \tag{A.2}$$

which reduces the complexity of the ML receiver to M' independent parallel searches to find the closest symbols to the estimated signal

$$\hat{\mathbf{s}}_{\mathrm{ML}}[n] = \frac{\tilde{\mathbf{W}}^T(\mathbf{H})\tilde{\mathbf{y}}[n]}{\|\mathbf{H}\|^2},$$

i.e., the OSTBC-MIMO channel response vectors $\tilde{\mathbf{w}}_k(\mathbf{h}_j)$ defined in (A.1) can be seen as the ML equalizers.

The necessary and sufficient conditions on the code matrices to satisfy (A.2), for $k = 1, \ldots, M'$, are given by (151)

$$\mathbf{C}_k^H\mathbf{C}_l = \begin{cases} \mathbf{I} & k = l, \\ -\mathbf{C}_l^H\mathbf{C}_k & k \neq l. \end{cases} \tag{A.3}$$

It is straightforward to prove that the above condition must also be satisfied by the real extended code matrices.

$$\tilde{\mathbf{C}}_k^T\tilde{\mathbf{C}}_l = \begin{cases} \mathbf{I} & k = l, \\ -\tilde{\mathbf{C}}_l^T\tilde{\mathbf{C}}_k & k \neq l. \end{cases}$$

The most popular OSTBC is the Alamouti code (41), which transmits $M = 2$ complex symbols in $L = 2$ time slots, so the code rate is $R = 1$. The n-th block of data for the Alamouti code is

$$\mathbf{S}[n] = \begin{bmatrix} r_1[n] & r_2[n] \\ -r_2^*[n] & r_1^*[n] \end{bmatrix}$$

and the code matrices are

$$C_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad C_2 = \begin{bmatrix} j & 0 \\ 0 & -j \end{bmatrix}$$
$$C_3 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \qquad C_4 = \begin{bmatrix} 0 & j \\ j & 0 \end{bmatrix}.$$

The tests shown in Section 2.7 are restricted to the Alamouti code because of the limitation in the number of transmitting antennas of the testbed used in the experiments. The use of a 2 × 2 platform precludes the use of more sophisticated OSTBCs. Nevertheless, most of the latest standards that use MIMO technologies for broadband wireless systems, such as IEEE 802.16e (Mobile WiMAX) or IEEE 802.20, and evolutions of third generation (3G) systems, such as 3G long-term evolution (LTE), support MIMO systems with two antennas and Alamouti coding for cost and simplicity reasons (101, 155).

A.3 Channel Estimation in MIMO-OSTBC systems

In this subsection it is described the channel estimation techniques used in the experiments of Section 2.7 for Alamouti decoding. First, it is considered the conventional pilot-based supervised technique and next it is described a recently proposed blind technique.

A.3.1 Pilot-Aided Channel Estimation.

The channel estimation method used is described in (44). Basically, for n_T transmit antennas we need to construct n_T pilot sequences. During the n-th frame transmitted from antennas one and two it is inserted a pair of pilot sequences consisting of K symbols each one

$$\mathbf{S}^{\text{pilot}} = \left[\mathbf{s}_1^{\text{pilot}}\ \mathbf{s}_2^{\text{pilot}}\right] = \begin{bmatrix} s_{1,1} & s_{2,1} & \cdots & s_{K,1} \\ s_{1,2} & s_{2,2} & \cdots & s_{K,2} \end{bmatrix}^T.$$

The pilot sequences are designed to be orthogonal

$$\left(\mathbf{s}_i^{\text{pilot}}\right)^H \mathbf{s}_l^{\text{pilot}} \propto \delta_i^l$$

where δ_i^l is the Kronecker delta. This orthogonality among the pilot sequences allows estimating independently the fading coefficient from each transmitting antenna to each receiving antenna. Specifically, the Least Squares (LS) estimate of the channel coefficient between the i-th transmit antenna and the j-th receive antenna is given by

$$\hat{h}_{i,j} = \frac{\left(\mathbf{s}_i^{\text{pilot}}\right)^H \mathbf{y}_j^{\text{pilot}}}{\left\|\mathbf{s}_i^{\text{pilot}}\right\|^2}$$

where $\mathbf{y}_j^{\text{pilot}}$ is the received signal at the j-th antenna when $\mathbf{s}_i^{\text{pilot}}$ has been transmitted.

On the other hand, the transmission of a pilot sequence provokes a reduction in the effective E_b/N_0 or, equivalently, a reduction in the effective transmission rate. For instance, when N_D data symbols and K pilots are transmitted during the n-th frame, the rate reduction factor associated to this technique is

$$R_{\text{pil}} = \frac{N_D}{N_D + K}.$$

A.3.2 SOS-based Blind Channel Estimation

Recently, a new method for blind channel estimation under OSTBC transmissions has been proposed in (52). It is based only on Second Order Statistics (SOS) and it is able to blindly identify the channel (up to a real scalar ambiguity) for most of the existing OSTBCs when the number of receive antennas is $n_R > 1$ (156). However, some OSTBCs (including the Alamouti code used in this paper) cannot be identified by this method due to an additional ambiguity, which must be eliminated by resorting to other information (e.g., linear precoding, non-white source signal, reduced rate, etc.) (52, 53, 156, 157).

In this subsection, the method proposed in (52) is first summarized and then particularized for the Alamouti scheme using the method proposed in (53) to avoid ambiguities.

Let us start by writing the $2n_RL \times 2n_RL$ correlation matrix of $\tilde{\mathbf{y}}[n]$

$$\mathbf{R}_{\tilde{\mathbf{y}}} = E[\tilde{\mathbf{y}}[n]\tilde{\mathbf{y}}^T[n]] = \tilde{\mathbf{W}}(\mathbf{H})\mathbf{R}_s\tilde{\mathbf{W}}^T(\mathbf{H}) + \frac{\sigma^2}{2}\mathbf{I}, \tag{A.4}$$

where $\mathbf{R}_s = E[\mathbf{s}[n]\mathbf{s}^T[n]]$ is the source correlation matrix.

The method proposed in (52) is the solution to the following optimization problem

$$\begin{aligned}\hat{\mathbf{H}} &= \underset{\mathbf{H}}{\operatorname{argmax}} \operatorname{Tr}\left(\tilde{\mathbf{W}}^T(\mathbf{H})\mathbf{R}_{\tilde{\mathbf{y}}}\tilde{\mathbf{W}}(\mathbf{H})\right),\\ \text{s.t. } & \tilde{\mathbf{W}}^T(\mathbf{H})\tilde{\mathbf{W}}(\mathbf{H}) = \mathbf{I},\end{aligned} \tag{A.5}$$

which is given by any channel matrix $\hat{\mathbf{H}}$ with $\|\hat{\mathbf{H}}\| = 1$ satisfying

$$\operatorname{range}(\tilde{\mathbf{W}}(\hat{\mathbf{H}})) = \operatorname{range}(\tilde{\mathbf{W}}(\mathbf{H})). \tag{A.6}$$

or, equivalently

$$\tilde{\mathbf{W}}(\hat{\mathbf{H}}) = \tilde{\mathbf{W}}(\mathbf{H})\mathbf{Q}$$

where $\mathbf{Q}$ is an orthogonal matrix.

It has been shown (157) that $\boxed{\text{A.5}}$ can also be rewritten as the following principal component analysis (PCA) problem

$$\underset{\hat{\hat{\mathbf{h}}}}{\operatorname{argmax}} \, \hat{\hat{\mathbf{h}}}^T\tilde{\mathbf{Z}}^T\tilde{\mathbf{Z}}\hat{\hat{\mathbf{h}}}, \qquad \text{s.t.} \qquad \|\hat{\hat{\mathbf{h}}}\| = 1, \tag{A.7}$$

where the data matrix is defined as $\tilde{\mathbf{Z}} = [\tilde{\mathbf{Z}}[0]^T \cdots \tilde{\mathbf{Z}}[N-1]^T]^T$, $\tilde{\mathbf{Z}}[n]$ is

$$\tilde{\mathbf{Z}}[n] = \begin{bmatrix} \tilde{\mathbf{y}}_1^T[n]\tilde{\mathbf{C}}_1 & \cdots & \tilde{\mathbf{y}}_{n_R}^T[n]\tilde{\mathbf{C}}_1 \\ \vdots & \ddots & \vdots \\ \tilde{\mathbf{y}}_1^T[n]\tilde{\mathbf{C}}_{M'} & \cdots & \tilde{\mathbf{y}}_{n_R}^T[n]\tilde{\mathbf{C}}_{M'} \end{bmatrix},$$

and $\hat{\hat{\mathbf{h}}}$ is defined as follows

$$\hat{\hat{\mathbf{h}}} = \left[\hat{\hat{\mathbf{h}}}_1^T \ , \ \ldots \ , \ \hat{\hat{\mathbf{h}}}_{n_R}^T\right]^T.$$

Once the channel has been obtained, the transmitted signal is estimated as

$$\hat{\mathbf{s}}[n] = \tilde{\mathbf{Z}}[n]\hat{\hat{\mathbf{h}}}.$$

As it was previously mentioned, for the particular case of Alamouti coding, this technique cannot be applied due to an indeterminacy caused by (A.6). Basically, this ambiguity appears when there exists an estimated channel $\hat{\mathbf{H}} \neq c\mathbf{H}$, whose associated equalization matrix $\tilde{\mathbf{W}}(\hat{\mathbf{H}})$ spans the same subspace as $\tilde{\mathbf{W}}(\mathbf{H})$. For the case of the

MIMO 2×2 testbed with Alamouti coding used in the experiments, this indeterminacy implies that the largest eigenvalue of $\tilde{\mathbf{Z}}^T\tilde{\mathbf{Z}}$ has a multiplicity of four (52, 156). This means that the true channels $\mathbf{h}_1$ and $\mathbf{h}_2$ associated to the first and second receive antenna, respectively, belong to the subspace spanned by the four eigenvectors associated to the largest eigenvalue of $\tilde{\mathbf{Z}}^T\tilde{\mathbf{Z}}$.

Several attempts have been made to overcome this ambiguity by resorting to some form of precoding of the source signal. In the tests in subsection 2.7 it is used a particularly simple method which has been recently proposed in (53). There, it is proved that any OSTBC transmitting an odd number of real symbols (i.e., M' odd) is identifiable regardless of the number of receiving antennas. Therefore, any non-identifiable complex OSTBC can be made identifiable simply by not transmitting one real symbol per OSTBC block. Obviously the transmission rate is reduced, but this rate penalty can be controlled by eliminating only one real symbol per B OSTBC blocks. In this case, the rate reduction factor is

$$R_{\text{blind}} = \frac{BM' - 1}{BM'}$$

which tends to one for $BM' \gg 1$. Finally, for a fixed number of transmitted OSTBC blocks there is a trade-off between the quality of the channel estimate and R_{blind} as a function of B, an issue that has been discussed in (53).

A.4 Differential STBCs

An alternative to blind decoding methods that also avoids the need of CSI estimation is the use of differential schemes. It is well-known, however, that differential modulations suffer a penalty of 3 dB in comparison to coherent detection. Below it is briefly presented the specific encoding and decoding of Differential STBC (DSTBC) for two transmit and two receive antennas that is included in this comparative study (for a more detailed explanation see (151)). This particular type of DSTBC is restricted to constant modulus signals ($|r_k[n]| = 1$).

A.4.1 Encoding Algorithm

Let $\mathbf{X}[n]$ be a set of unitary matrices to be transmitted. In DSTBC schemes the matrices $\mathbf{T}[n]$ are transmitted, which are constructed as follows:

$$\mathbf{T}[n] = \mathbf{X}[n]\mathbf{T}[n-1]$$

with $\mathbf{T}[0] = \mathbf{I}$. Using an OSTBC, a possible design for the matrices $\mathbf{X}[n]$ is

$$\mathbf{X}[n] = \frac{1}{\sqrt{M}}\mathbf{S}[n]$$

where the normalization factor $1/\sqrt{M}$ is necessary to obtain $\mathbf{X}[n]\mathbf{X}^H[n] = \mathbf{I}$. This particular choice of unitary matrices reduces the computational complexity of the detector.

A.4.2 Decoding Algorithm

If we transmit $\mathbf{T}[n]$, the received $L \times n_R$ matrix is

$$\begin{aligned}\mathbf{Y}[n] &= \mathbf{T}[n]\mathbf{H} + \mathbf{N}[n] \\ &= \mathbf{X}[n]\mathbf{T}[n-1]\mathbf{H} + \mathbf{N}[n]\end{aligned}$$

where $\mathbf{Y}[n] = [\mathbf{y}_1[n] \cdots \mathbf{y}_{n_R}[n]]$, and $\mathbf{N}[n]$ is defined analogously. The ML detection of $\mathbf{X}[n]$ from $\mathbf{Y}[n]$ and $\mathbf{Y}[n-1]$ amounts to maximizing the following cost function (151)

$$J(\mathbf{X}[n]) = \Re\left\{\mathrm{Tr}\left\{\mathbf{X}[n]\mathbf{Y}[n-1]\mathbf{Y}^H[n]\right\}\right\}. \tag{A.8}$$

For arbitrary unitary matrices $\mathbf{X}[n]$ this is a computationally expensive problem; however, using OSTBC matrices the detection of each symbol can be decoupled and the cost function (A.8) to be maximized takes now the form

$$J(s_1[n], \cdots, s_{M'}[n]) = \sum_{k=1}^{M'} \left(\Re\left\{\mathrm{Tr}\left\{\mathbf{C}_k\mathbf{Y}[n-1]\mathbf{Y}^H[n]\right\}\right\} s_k[n]\right) \tag{A.9}$$

B

List of Acronyms

3G	Third Generation
3GPP	Third Generation Partnership Project
ACR	Adjacent Channel Rejection
ADC	Analog-to-Digital Converter
ASIC	Application-Specific Integrated Circuit
AWGN	Additive White Gaussian Noise
BPSK	Binary Phase-Shift Keying
DAC	Digital-to-Analog Converter
BER	Bit Error Rate
BNC	Bayonet Neill-Concelman Connector
CP	Cyclic Prefix
CPLD	Complex Programmable Logic Device
CSI	Channel State Information
DC	Direct Current
DDR	Double Data Rate
DDS	Direct Digital Synthesizer
DSL	Digital Subscriber Line
DSP	Digital Signal Processor
DSRC	Dedicated Short Range Communications
DSTBC	Differential Space-Time Block Coding
EMIF	External Memory InterFace
ETSI	European Telecommunications Standards Institute
FEC	Forward Error Correction
FER	Frame Error Rate or FEC block Error Rate
FFT	Fast Fourier Transform
FIFO	First-In First-Out
FIR	Finite Impulse Response
FPGA	Field-Programmable Gate Array
FTP	File Transfer Protocol
GSCM	Geometry-based Stochastic Channel Model
GUI	Graphical User Interface
HARQ	Hybrid Automatic Repeat Request
HDL	Hardware Description Language
HOS	Higher-Order Statistics
HSDPA	High-Speed Downlink Packet Access
HSPA	High-Speed Packet Access
ICI	Inter-Carrier Interference
IEEE	Institute of Electrical and Electronics Engineers
IF	Intermediate Frequency

IFFT	Inverse Fast Fourier Transform
IIR	Infinite Impulse Response
ITS	Intelligent Transport Systems
LLR	Log-Likelihood Ratio
LOS	Line-Of-Sight
LS	Least Squares
LTE	Long-Term Evolution
LUT	Look-Up Table
MAC	Medium Access Control
MF	Matched Filter
MIMO	Multiple-Input Multiple-Out
MISO	Multiple-Input Single-Out
MMBX	Micro-Miniature Board Connector
MMSE	Minimum Mean Square Error
ML	Maximum Likelihood
MRC	Maximum-Ratio Combining
MSE	Mean Squared Error
NLOS	Non-Line-Of-Sight
OFDM	Orthogonal Frequency-Division Multiplexing
OFDMA	Orthogonal Frequency-Division Multiple Access
OSTBC	Orthogonal Space-Time Block Coding
OSTPM	Orthogonal Space-Time Pilot Matrices
PC	Personal Computer
PCI	Peripheral Component Interconnect
PER	Packet Error Rate
PHY	Physical
PN	Pseudorandom Number
PRBS	Pesudo-Random Binary Sequence
PUSC	Partial Usage of Subchannels
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase-Shift Keying
RF	Radio Frequency
RIG	Random-Integer Generator
RMS	Root Mean Squared
RTT	Round-Trip Time
RTV	Road-to-Vehicle
SDB	Sundance Digital Bus
SDRAM	Synchronous Dynamic Random Access Memory
SEM	Spectrum Emission Mask
SHB	Sundance High-speed Bus
SINR	Signal-to-Interference plus Noise Ratio
SISO	Single-Input Single-Output
SIMO	Single-Input Multiple-Output
SNR	Signal-to-Noise Ratio
SOFDMA	Scalable OFDMA
SOS	Second-Order Statistics or Sum-of-Sinusoids
STBC	Space-Time Block Coding

TCP	Transmission Control Protocol
TIM	Texas Instruments Module
UDP	User Datagram Protocol
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Integrated Circuit
VTV	Vehicle-to-Vehicle
WCDMA	Wideband Code Division Multiple Access
WGNG	White Gaussian Noise Generator
WLAN	Wireless Local Area Network
XOR	eXclusive OR