

Real-time Emulation of Vehicular Non-stationary Channels

Rute Ramalho Santos

Thesis to obtain the Master of Science Degree in
Electrical and Computer Engineering

Supervisor: Prof. Luís Manuel de Jesus Sousa Correia

Examination Committee

Chairperson: Prof. José Eduardo Charters Ribeiro da Cunha Sanguino
Supervisor: Prof. Luís Manuel de Jesus Sousa Correia
Member of the Committee: Prof. António José Castelo Branco Rodrigues

May, 2018

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practises of the Universidade de Lisboa.

Acknowledgments

I would like to start by acknowledging my dissertation supervisors Professor Luís M. Correia and Professor Christoph F. Mecklenbräuker, for their insight and support .

I would like to thank Professor Luís M. Correia, for his availability to assist me and for receiving me in GROW after coming back from Vienna. His guidance was vital, not only through sharing his advices and experience, but also the weekly meetings, and his honest and straightforward attitude.

My sincere gratitude goes to Professor Christoph F. Mecklenbräuker from the Institute of Telecommunications of TU Wien, who co-advised me, for the amazing opportunity of developing this work in Vienna and his guidance through the first steps.

To Dr. Thomas Blazek for all his help during the whole development of my thesis, for patiently answering my questions, doubts, and many emails, and for always being available to assist me, during and after my stay.

I would like to thank my family, specially my mother, for their friendship, encouragement and caring over all these years, for always being there for me through thick and thin, for their unconditional love, patience and sacrifices, and without whom this project would not be possible. To my cousin Daniela, who always supported me and understood me better than anyone. To my grandfather whom, although can not witness this moment, would undoubtedly be very proud.

To my old school group of friends, all friends in Instituto Superior Técnico, and all the amazing people I met during my stay in Vienna, for the motivation, the support and also the fun.

To each and every one of you – Thank you.

Abstract

The aim of this thesis is to describe the use of a communications channel emulator to assess the performance of modems installed in vehicles, in the field of Intelligent Transportation Systems' applications. These vehicle nodes are responsible for exchanging safety and traffic efficiency information, using the amendment to the Wi-Fi protocol known as 802.11p. This enables the simulation of a realistic vehicular channel with reconfigurable parameters and without the need of field measurements. For that, it is essential to thoroughly model the non-stationary channel, characterized by dynamic environments, high node velocities and multipath. The channel is described by a tapped-delay line model, in which the taps are determined by an optimization algorithm, based on LASSO, on real-life measurements in Lund, Sweden. The algorithm aims at minimizing the mean square error between the approximated channel description and the original measurements, and limiting the number of taps in use. The emulator is part of a project developed in the Telecommunications Institute of the Technical University of Vienna. Its further development enables the emulation of several consecutive scenarios and the configuration of the delay parameters for each magnitude and phase. The performance of the channel emulator is evaluated through the packet error ratio, for the various scenarios and different attenuations, showing that stationary channel models do not emulate vehicular scenarios realistically and, for the non-stationary model, exchanged information is almost completely lost except when the vehicles are in close proximity.

Keywords

Intelligent Transportation Systems; 802.11p; non-stationary channels; tapped-delay line; emulator; packet error ratio.

Resumo

O objetivo desta tese é descrever o uso de um emulador de canal de comunicações para avaliar o desempenho de modems instalados em veículos, no âmbito de aplicações dedicadas a Sistemas de Transporte Inteligentes. Estes nós veiculares são responsáveis pela troca de informações de segurança e eficiência de trânsito, utilizando a emenda 802.11p do protocolo Wi-Fi. Isto permite a simulação de um canal veicular realista caracterizado por parâmetros que são reconfiguráveis e sem a necessidade de medições de campo. Para isso, é essencial modelar minuciosamente o canal não-estacionário, caracterizado por ambientes dinâmicos, elevadas velocidades dos nós e o efeito do multicaminho. O canal é descrito por um modelo de linha de atraso *tapped*, no qual os *taps* são determinados através de uma otimização, baseada no algoritmo LASSO, em medidas reais em Lund, Suécia. O algoritmo visa a minimização do erro quadrático médio entre a descrição aproximada do canal e as medições originais, e limitar o número de *taps* em uso. O emulador faz parte de um projeto desenvolvido no Instituto de Telecomunicações da Universidade Técnica de Viena. O seu desenvolvimento posterior permite a emulação consecutiva de cenários e a configuração dos parâmetros de atraso, para cada magnitude e fase. O desempenho do emulador é avaliado através da taxa de erro de pacote, para diversos cenários e atenuações, mostrando que os modelos estacionários não emulam cenários veiculares de modo realista e, para o modelo não-estacionário, a informação partilhada é perdida exceto quando os veículos estão nas proximidades.

Palavras Chave

Sistemas de Transporte Inteligentes; 802.11p; canais não-estacionários; linha de atraso *tapped*; emulador; taxa de erro de pacote.

Contents

Acknowledgments	v
Abstract	vii
Resumo	ix
List of Figures	xiii
List of Tables	xvii
List of Acronyms	xix
List of Symbols	xxiii
List of Software	xxv
1 Introduction	1
1.1 Overview	3
1.2 Motivation and Concepts	6
2 Basic Concepts and State of the Art	9
2.1 IEEE 802.11p/WAVE	11
2.2 Vehicular Scenarios	15
2.3 Vehicular Channel Characterization	17
2.3.1 Channel Impulse Response	17
2.3.2 Tapped Delay Line	19
2.3.3 Influence on the Scenarios	21
2.4 Channel Emulator	22
2.5 State of the Art	24
3 Model and Implementation	27
3.1 Channel Models	29
3.1.1 Stationary Tests from Distributions	29
3.1.2 Non-stationary from Measured Data	30
3.2 Emulator Basic Description	34
3.3 Emulator Development	38
3.3.1 Transmission and Reception of Scenario Data	38
3.3.2 Data Updating	40
3.3.3 Data Writing	43
3.4 Model Assessment	45

4 Results Analysis	49
4.1 Measured Data Scenario	51
4.2 Data Processing	52
4.3 Analysis of Results	55
4.3.1 Stationary Models	55
4.3.2 Non-stationary Models	64
5 Conclusions	71
A Distributions	77
A.1 Rayleigh	79
A.2 Rice	79
B NI USRP RIO	81
C PER Traces for Stationary Scenarios	85
C.1 Pure Rayleigh	87
C.2 Pure Rice	89
C.3 Urban Approaching	91
C.4 Urban Crossing	91
C.5 Highway LOS	92
C.6 Highway NLOS	93
D PER Traces for Non-stationary Scenarios	95
D.1 Packet Size of 200 Bytes	97
D.2 Packet Size of 500 Bytes	99
D.3 Packet Size of 1000 Bytes	101
References	103

List of Figures

1.1	Phases of the evolution of connected cars.	4
1.2	Vehicular communication environment.	5
2.1	VANET environment.	11
2.2	WAVE system, with common locations of the OBUs and RSUs, the general makeup of the WBSSs, and the way a WBSS can connect to a WAN through a portal.	13
2.3	Set of channels defined in WAVE.	14
2.4	Tap Delay Line model.	20
2.5	Delay line with one tap.	20
2.6	Example of a Tap Delay Line, with an arbitrary number of taps.	20
2.7	Example of time varying discrete-time impulse response model for multipath channel. . .	21
2.8	Channel emulator system.	23
3.1	System view of the emulator.	35
3.2	Channel input vector, for all taps N_{taps} in each time instant and time duration N_t	36
3.3	Blocks diagram for the signal transmission, convolution with simulated channel and reception.	36
3.4	Tapped delay line architecture for the emulator convolution module.	37
3.5	Measurement setup, with the Tx, channel emulator and Rx. The emulator's interface is shown as well in the monitor.	37
3.6	Top down modular design for the emulator.	38
3.7	UDP transmission.	39
3.8	UDP reception.	40
3.9	Store data in FIFO.	41
3.10	Choice of FIFO for data removal.	42
3.11	Update and passing of data arrays.	42
3.12	Rotate array.	43
3.13	Test to check if the scenario to be emulated is the first to be received by the emulator. . .	44
3.14	Test to check if array with the scenario data is complete.	45
3.15	Test to see if array is ready to be written into FPGA.	45
3.16	Spectrum of the channel model before and after sudden transition.	47
3.17	Emulator output spectrum of the channel before and after sudden transition.	47

4.1	Organization of the measurements matrix.	52
4.2	Urban crossing in Lund, Sweden. A, B, C and D represent vehicles (extracted from GoogleMaps).	52
4.3	Urban crossing. The receiver D moves at speed v_{Rx} and the transmitter C moves at speed v_{Tx}	53
4.4	Magnitude of the non-stationary channel along time and delay.	53
4.5	Channel sounder results for urban crossing scenario.	54
4.6	Approximated channel results for urban crossing scenario.	54
4.7	AWGN comparison scenario, for 200 bytes packet size and several attenuations.	56
4.8	60 PER traces for Pure Rayleigh and attenuation of 12dB, for different speeds.	57
4.9	Sliding average PER results for Pure Rayleigh scenario, for 500 bytes packet size and several attenuations.	58
4.10	60 PER traces for Pure Rice and attenuation of 12dB, for different speeds.	59
4.11	Sliding average PER results for Pure Rice scenario, for 500 bytes packet size and several attenuations.	60
4.12	60 PER traces for Urban Intersection and attenuation of 12dB.	61
4.13	Sliding average PER results for ETSI urban scenarios, for 500 bytes packet size and several attenuations.	62
4.14	60 PER traces for Highway and attenuation of 12 dB.	62
4.15	Sliding average PER results for ETSI highway scenarios, for 500 bytes packet size and several attenuations.	63
4.16	General linear regression for the expected PER curve behaviour. The different regions are characterized by a different linear equation.	65
4.17	60 PER traces for packet size of 200 bytes.	66
4.18	60 PER traces for packet size of 500 bytes.	66
4.19	60 PER traces for packet size of 1000 bytes.	66
4.20	Average PER over emulation for 200 bytes packet size. The label shows the attenuations.	69
4.21	Average PER over emulation for 500 bytes packet size. The label shows the attenuations.	70
4.22	Average PER over emulation for 1000 bytes packet size. The label shows the attenuations.	70
B.1	USRP RIO System Block Diagram.	84
C.1	60 PER traces for Pure Rayleigh and speed of 10 km/h.	87
C.2	60 PER traces for Pure Rayleigh and speed of 50 km/h.	87
C.3	60 PER traces for Pure Rayleigh and speed of 100 km/h.	88
C.4	60 PER traces for Pure Rayleigh and speed of 150 km/h.	88
C.5	60 PER traces for Pure Rice and speed of 10 km/h.	89
C.6	60 PER traces for Pure Rice and speed of 50 km/h.	89
C.7	60 PER traces for Pure Rice and speed of 100 km/h.	90
C.8	60 PER traces for Pure Rice and speed of 150 km/h.	90

C.9	60 PER traces for Urban Approaching.	91
C.10	60 PER traces for Urban Crossing.	92
C.11	60 PER traces for Highway with LOS.	92
C.12	60 PER traces for Highway without LOS.	93
D.1	60 PER traces for packet size of 200 bytes.	97
D.2	60 PER traces for packet size of 200 bytes.	97
D.3	60 PER traces for packet size of 200 bytes.	98
D.4	60 PER traces for packet size of 500 bytes.	99
D.5	60 PER traces for packet size of 500 bytes.	99
D.6	60 PER traces for packet size of 500 bytes.	100
D.7	60 PER traces for packet size of 1000 bytes.	101
D.8	60 PER traces for packet size of 1000 bytes.	101
D.9	60 PER traces for packet size of 1000 bytes.	102

List of Tables

2.1	List of the protocols that compose the WAVE communications stack, with the designation of the standard that covers each one.	12
2.2	Difference in parameters for amendments 802.11a and 802.11p.	14
2.3	Average vehicular communication measures for vehicle speed and distance between nodes.	17
3.1	List of assessment tests made on the emulator to validate its performance.	46
4.1	Measurement campaign parameters (adapted from [48]).	51
4.2	Pure Rayleigh PER results average and standard deviation, for attenuation of 12 dB.	56
4.3	Pure Rice PER results average and standard deviation, for attenuation of 12 dB.	59
4.4	PER results average and standard deviation for urban scenarios and attenuation of 12 dB.	61
4.5	Highway PER results average and standard deviation, for attenuation of 12 dB.	63
4.6	Linear regression equation parameters for 200 bytes and the 3 main regions.	67
4.7	Linear regression equation parameters for 500 bytes and the 3 main regions.	67
4.8	Linear regression equation parameters for 1000 bytes and the 3 main regions.	68
4.9	Correlation coefficient between PER results and regression, PER results mean and standard deviation.	69
C.1	Pure Rayleigh PER results (over 60 traces) average and standard deviation, for attenuation of 4 dB and 15 dB.	88
C.2	pure Rice PER results (over 60 traces) average and standard deviation, for 4 dB and 15 dB.	90
C.3	Urban PER results (over 60 traces) average and standard deviation, for 4dB and 15dB.	91
C.4	Highway PER results (over 60 traces) average and standard deviation, for 4dB and 15 dB.	93
D.1	Linear regression equation parameters for 200 bytes and the 3 main regions.	98
D.2	Average parameters over the 60 traces: correlation coefficient between PER results and regression, PER results average and standard deviation, for 200 bytes packet size.	98
D.3	Linear regression equation parameters for 500 bytes and the 3 main regions.	100
D.4	Average parameters over the 60 traces: correlation coefficient between PER results and regression, PER results average and standard deviation, for 500 bytes packet size.	100
D.5	Linear regression equation parameters for 1000 bytes and the 3 main regions.	102
D.6	Average parameters over the 60 traces: correlation coefficient between PER results and regression, PER results average and standard deviation, for 1000 bytes packet size.	102

List of Acronyms

ADC	Analog-to-Digital Converter
AP	Access Point
AWGN	Additive White Gaussian Noise
BS	Base Station
BSS	Basic Service Set
BSSID	Basic Service Set Identification
CCH	Control Channel
CIR	Channel Impulse Response
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
DAC	Digital-to-Analog Converter
DC	Direct Current
DDC	Digital Down-Conversion
DSP	Digital Signal Processor
DUC	Digital Up-Conversion
EDCA	Enhanced Distributed Channel Access
ETSI	European Telecommunications Standards Institute
FDD	Frequency Division Duplex
FFT	Fast Fourier Transform
FIFO	First In First Out
FIR	Finite Impulse Response
FM	Frequency Modulation

FPGA	Field-Programmable Gate Array
GSCM	Geometric-based Stochastic Channel Model
HetNet	Heterogeneous Network
I	In-phase
IO	Input Output
IoT	Internet of Things
IP	Internet Protocol
ISI	Inter-Symbol Interference
ITS	Intelligent Transportation Systems
LOS	Line Of Sight
LTE	Long Term Evolution
MAC	Media Access Control
MIMO	Multiple-Input and Multiple-Output
MPC	Multipath Component
MPS	Multipath Propagation Simulator
MSE	Mean Squared Error
NI	National Instruments™
NLOS	Non-Line Of Sight
OBU	On-Board Unit
OFDM	Orthogonal Frequency Division Multiplexing
OFDMA	Orthogonal Frequency Division Multiple Access
PDP	Power Delay Profile
PER	Packet Error Ratio
PHY	Physical layer
Q	Quadrature
QAM	Quadrature Amplitude Modulation
RF	Radio-Frequency
RMS	Root Mean Square

RSSI	Received Signal Strength Indication
RSU	Roadside infrastructure Units
Rx	Receiver
SCH	Service Channels
SDR	Software-Defined Radio
SISO	Single-Input and Single-Output
SNR	Signal-to-Noise Ratio
TDL	Tapped Delay Line
Tx	Transmitter
UDP	User Datagram Protocol
USRP	Universal Software Radio Peripheral
V2I	Vehicle-to-Infrastructure
V2P	Vehicle-to-Pedestrian
V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-everything
VANET	Vehicular Ad-hoc Network
VC	Vehicular Communications
WAVE	Wireless Access in Vehicular Environments
WBSS	WAVE Basic Service Set
WLAN	Wireless Local Area Network
WSA	WAVE Service Advertisement
WSS	Wide Sense Stationary
WSSUS	Wide Sense Stationary with Uncorrelated Scatterers

List of Symbols

δ	unit impulse function
Θ_i	total phase shifts of i th multipath component
μ_τ	mean energy delay
$\rho_{P_p, P_p^{reg}}$	correlation coefficient between PER data and linear regression
σ	channel multipath delay
$\bar{\sigma}$	mean delay spread
σ_p	standard deviation for PER
$\sigma_{P_p^{reg}}$	linear regression standard deviation for PER
σ_{RMS}	RMS delay spread
ϕ	additional phase shifts in the channel
a_i	real amplitudes of i th multipath component
c	speed of light
d	position
F	discrete Fourier transform matrix
f_c	carrier frequency
f_D	Doppler spread
$f_{D,i}$	maximum Doppler spread for time snapshot i
h	impulse response of a multipath channel in time
$h_{i,WSS}$	realization from a WSS fading
h^{Rice}	vector of time components for a single tap using Rice distribution
h^{Ray}	vector of time components for a single tap using Rayleigh distribution

\mathbf{H}	channel matrix
\mathbf{h}_i	channel row vector for a single time slot i
\mathbf{H}_{app}	approximated channel matrix
$\mathbf{h}_{app,i}$	approximated channel row vector for a single time slot i
$\hat{\mathbf{h}}_{app,i}$	estimator for the approximated channel vector in a single time slot i
I_0	I_0 -Norm
I_1	I_1 -Norm
k_{Rice}	constant factor that guarantees Rice components are not larger than 1
k_{taps}	limiting constant factor dependent on the number of intended taps
$m_{Region\ i}$	linear regression slope for region i
N_{MPC}	total number of multipath components
$N_{packets}$	total number of packets sent for a single trace, during an emulation
N_t	total number of time snapshots
N_{taps}	total number of active taps
$p(\sigma)$	power delay profile
$P_{0,Region\ i}$	linear regression offset in time instant t_0 , for region i
P_p	PER
\bar{P}_p	mean value for PER
$P_{p,i}$	PER value per packet i
\bar{P}_p^{reg}	linear regression mean value for PER
$P_{p,i}^{reg}$	linear regression PER value per packet i
r	output stream of the emulator
s	input stream of the emulator
t	time
$t_{0,i}$	time instant Region i starts
$t_{max,i}$	time instant Region i ends
t_s	tuning parameter
v	velocity

List of Software

LabVIEW LabVIEW™ 2014

Matlab Matlab R2015a

Python Python

1

Introduction

This chapter presents an overview of the context in which the thesis was developed, taking current vehicular communications scenarios into account, and the motivations for its development.

1.1 Overview

The present increase in people and goods' mobility throughout the world has been impacting the society when it comes to traffic congestion, accidents and fatalities in vehicular contexts [1]. The number of vehicles in cities is starting to surpass population numbers. This situation, together with the new wave of urbanization and the growing pressure on improving private and public transports, have made the development of more sustainable, efficient and safer transportation systems one of the fundamental challenges of the next decade [2].

Some of the solutions already in use to prevent or mitigate the problems related with road safety and congestion comprise broadcast of information on traffic and hazardous situations via Frequency Modulation (FM) radio band, and road signs displaying variable messages, warning drivers about changing conditions [1].

The academic and industrial research communities have been investigating direct communications between vehicles in order to improve efficiency and safety of future transportation. The growing technological development regarding mobile computing, wireless communications and remote sensing are pushing interconnected driving applications forward [1]. Applications, such as Intelligent Transportation Systems (ITS), enable users to make a safer and smarter use of transport networks.

The growing sophistication of vehicle equipment, additional components of new wireless communications, computing and sensing capabilities enable the vehicles to gather data about traffic and road state information, and exchange it in real-time with other nearby vehicles and Roadside infrastructure Units (RSU). Some examples are the on-board controls and information sources that evaluate the state of the vehicle, and passive safety mechanisms that protect the passengers in adverse driving conditions. Also, autonomous sensor technologies, such as navigation systems, compasses, rear and front parking radars and cameras, evaluate the surrounding landscape, road and location of the vehicle in order to capture in real-time the traffic situation or accidents [1].

Some of the most relevant safety applications include the transmission of warnings for pre and post-crash situations. Pre-crash warnings focus on alerting drivers of hazardous vehicles with which they lack visual contact; they can be used in situations like intersection collision avoidance and cooperative merging assistance. In the post-crash case, these applications are intended to facilitate traffic flow; an example is after the occurrence of a traffic accident, by broadcasting a message with its location, and enabling the approaching vehicles to circumvent the accident [3].

The use of Vehicular Communications (VC) based solutions allows the safety and efficiency warning operation beyond the Line Of Sight (LOS) constraint, unlike what happens with traditional road signs and the already in use solutions described previously. This way, there is a cooperation between vehicles and infrastructures that perceive hazardous situations in extended space and time horizon [1].

Connected vehicles are seen as one of the key enabling technologies for more efficient and sustainable ITS, also boosting the evolution of automated driving. The connected vehicle evolution is shown in Figure 1.1.

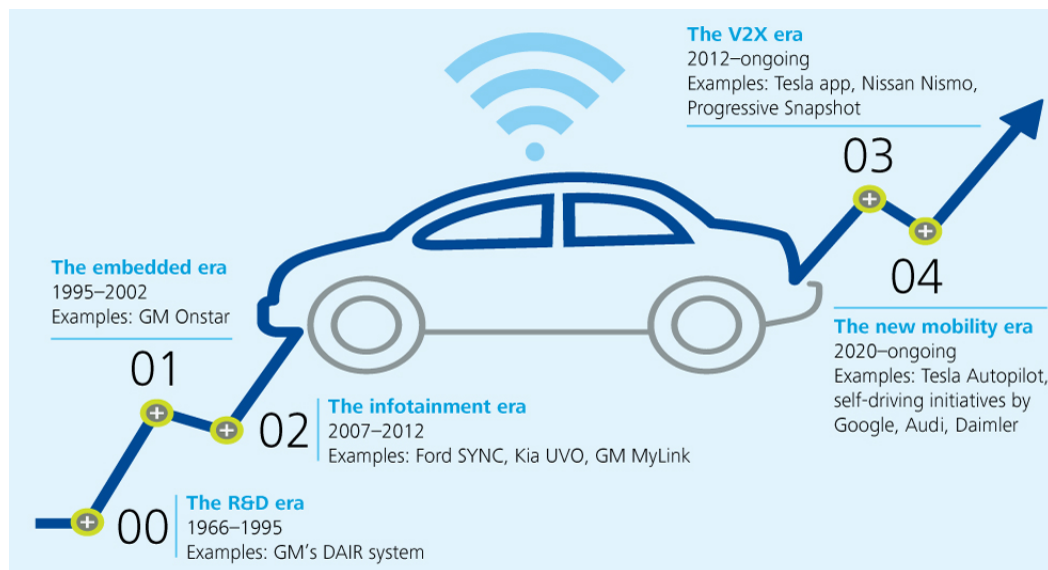


Figure 1.1: Phases of the evolution of connected cars (extracted from [4]).

Intelligent infrastructures, such as hospitals and police stations, part of the so called smart cities, should be equipped with base stations capable of communicating in a near real-time fashion with vehicles. While car drivers would travel in a more efficient, enjoyable and safer manner, smart cities would use the traffic information spread by car systems to offer best living conditions, by reducing air and noise pollutions and traffic congestions [5].

Gartner [6] points out that by 2020 there will a quarter billion connected cars, enabling new in-vehicle services and automated driving capabilities, and the proportion of new vehicles equipped with this capabilities will increase dramatically. The General Manager of IBM's Global Automotive Industry [7] suggested that "connected cars will be the ultimate Internet of Things (IoT)".

To make autonomous connected vehicles a reality, it is mandatory to develop and improve vehicular standards and design vehicular communication systems that enable road users and other actors to exchange information in real time and with high reliability.

Vehicular channels are wireless multipath links, meaning that the signal propagates from the Transmitter (Tx) through several paths, adding up all these contributions (amplitudes, phases and respective delays) at the Receiver (Rx), in different amounts and fading types. The vehicles and RSUs act as the communicating nodes, where the message transmission is done locally, i.e., without the need for fixed infrastructures or base stations, which is known as a Vehicular Ad-hoc Network (VANET).

VANETs are characterized by more diverse environments where the communication happens, the combination of different types of communication links, Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure

(V2I), Vehicle-to-Pedestrian (V2P), etc., and the type of objects, static and/or mobile, as seen in Figure 1.2 (where C2I refers to V2I and C2C refers to V2V.). The models created for cellular scenarios are not accurate enough to characterise the highly time-variant vehicular ones [8].

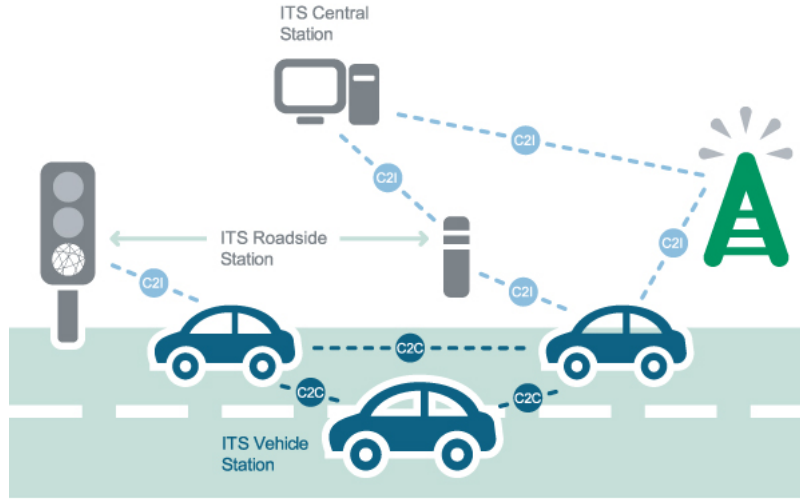


Figure 1.2: Vehicular communication environment (extracted from [9]).

In order to create reliable models for several kinds of vehicular and ITS applications, such as road-side efficiency and safety-related applications, the different types of vehicular links, such as V2I and V2V, and consequent channel characterization, have to be taken into consideration. Some messages transmitted over these networks have high latency requirements and a decision made on delayed information could be quite harmful. Adding to that, the mobility of Tx, Rx and interacting objects in the propagation environment result in highly non-stationary fading, which means the delay and Doppler spreads of each Multipath Component (MPC) change over time and frequency.

The system for VC is IEEE 802.11p [10], which is an approved amendment to the IEEE 802.11 standard to add Wireless Access in Vehicular Environments (WAVE) [11], allowing ad-hoc communication and direct exchange of information between vehicles and the vicinity, being the one used in this thesis. It takes the specific characteristics of vehicular communications into account, and it specifies physical and medium access control protocols designed for the volatile vehicular environment. The operating frequency is 5.9 GHz and the communication distance is shorter than in cellular cases, typically 100 m in Non-Line Of Sight (NLOS) and 500 m in LOS.

It is relevant to study, develop and improve the standards already existing for vehicular scenarios, namely 802.11p, so they are taken to the next level, by connecting a decentralized ad-hoc vehicular network to everything else, i.e., the network. The future of Vehicle-to-everything (V2X) communications may combine the best of 802.11p standard, which benefit from the absence of network and supports safety-critical and low latency applications, with the cellular ones, which add multimedia services, better support for V2I communications and network coverage.

Long Term Evolution (LTE) [12] with support for V2X is envisioned to improve all the aforementioned benefits, providing a viable and cost-effective solution, which is able to accelerate the adaptation of V2X communications on the way to 5G [13]. This way, the combination of both WAVE and cellular standards enables the increase of driving comfort, spreads traffic information and navigation for more efficient use of roads, submits the surroundings information to the network to optimize the infrastructures investments and make road travel more fuel efficient. In light of LTE evolution toward 5G and V2V communications, it is evident that enabling vehicles to also be connected to the cellular network provides superior capabilities [14].

1.2 Motivation and Concepts

The evolving demand for VC applications has encouraged the design and implementation of various modems, that provide reliable support systems for safety and traffic efficiency on the roads, to be mounted in ITS enabled vehicles. To test the performance of these modems, they need to be assessed under realistic conditions, creating the need for accurate models to replicate the roadside communication conditions in simulations and labs.

The small and large-scale signal statistics change very rapidly in a typical urban scenario, due to the dynamic environment, the low height of the antennas and the high mobility of the vehicles, causing the transmitted signal to interact with a large number of surrounding objects. In this multipath scenario, the simulation of each path, characterized by different fading configurations, is essential for thoroughly testing the receiver performance.

A possible solution to attest the reliability of commercial modems mounted in vehicles are measurement campaigns, which provide accurate data for evaluation of Tx and Rx. However, they are expensive and require extensive planning, which is undesirable for manufacturers. Instead, a channel emulator can be placed between the receiver and transmitter in order to emulate the fading characteristics of the channel.

An appropriate channel model needs to be used as input to the emulator for accurately evaluating vehicular protocols and applications before the actual deployment in real-life scenarios.

The work developed in this thesis is part of a project developed in the Institute of Telecommunications of Technical University of Vienna. It aims to create accurate models for vehicular communications and further develop an existing emulator, using these models as input, to recreate a non-stationary vehicular channel. To accomplish this, a powerful optimization tool, Lasso, is used to determine the taps that better represent the original non-stationary channel in a limited number of taps. Also, the functionalities added to the channel emulator allow not only to consecutively simulate different scenarios, but also to update the delay components with the update of the magnitude and phase ones. The optimized model allied to the developed emulator enable a low-complexity, fast and effective simulation of non-stationary

vehicular channels, fundamental in assessing modems for safety and efficiency purposes.

This thesis consists of 5 chapters, including this one, and a group of appendixes.

Chapter 2 starts by introducing the 802.11p system for vehicular communications, followed by a thorough characterization of the vehicular channel and relevant parameters. A description of the main functions of a channel emulator is also given, concluding with the state of the art, which presents the latest work developments on the subject of the thesis.

Chapter 3 analyses the stationary and non-stationary channel models to be implemented into the emulator, as well as the used methods to generate them, from probability distributions and from real-life measurements respectively. It ends with the further development of the emulator, implemented on a National Instruments™ (NI)-Universal Software Radio Peripheral (USRP) RIO platform [15], so that it can replicate vehicular non-stationary conditions.

Chapter 4 consists of the analysis of the obtained results. It begins with the description of the real-life measurements scenario and its processing. It ends with an evaluation of the emulator's performance through a Packet Error Ratio (PER) analysis of both stationary and non-stationary models.

Chapter 5 has the conclusions of this thesis, showing the most important outcomes and results, also addressing some suggestions for future work on this topic.

At the end, a group of appendixes is presented in order to give auxiliary information. Appendix A presents the relevant probability distributions that characterise the fading process for analyzed stationary channel models. Appendix B contains the explanation of how the platform of the emulator works. Appendix C exhibits the PER results for the several stationary channel models, while Appendix D presents the PER results for the non-stationary one.

2

Basic Concepts and State of the Art

This chapter focuses on the description of the Wi-Fi system that enables communications in vehicular ad-hoc networks, 802.11p, as well as the characterization of the vehicular channel, its modelling by a tapped delay line and the typical vehicular scenarios. The definition of a channel emulator and an analysis of the state of the art is also presented.

2.1 IEEE 802.11p/WAVE

The IEEE 802.11 standard collects and specifies the Physical layer (PHY) specifications and Media Access Control (MAC) for the implementation of Wireless Local Area Network (WLAN) in the 2.4, 3.6, 5 and 60 GHz frequency bands. The standard and its amendments provide the fundamentals of Wi-Fi technology [16].

An IEEE 802.11 WLAN architecture can be divided into Basic Service Sets (BSSs), in which several wireless devices are connected to a single redistribution point, Access Point (AP). It is identified by a Basic Service Set Identification (BSSID), shared by all the unit members.

VANETs impose a set of new requirements on existing wireless communication systems, such as low latency and high data rate. Vehicular safety applications are intolerant to long connection establishment delays before being enabled to communicate with other vehicles encountered on the road [17].

The entities of a WAVE system, displayed in Figure 2.1, consist of RSUs, fixed and installed on the road, and On-Board Units (OBUs), mounted in vehicles and functioning while moving.

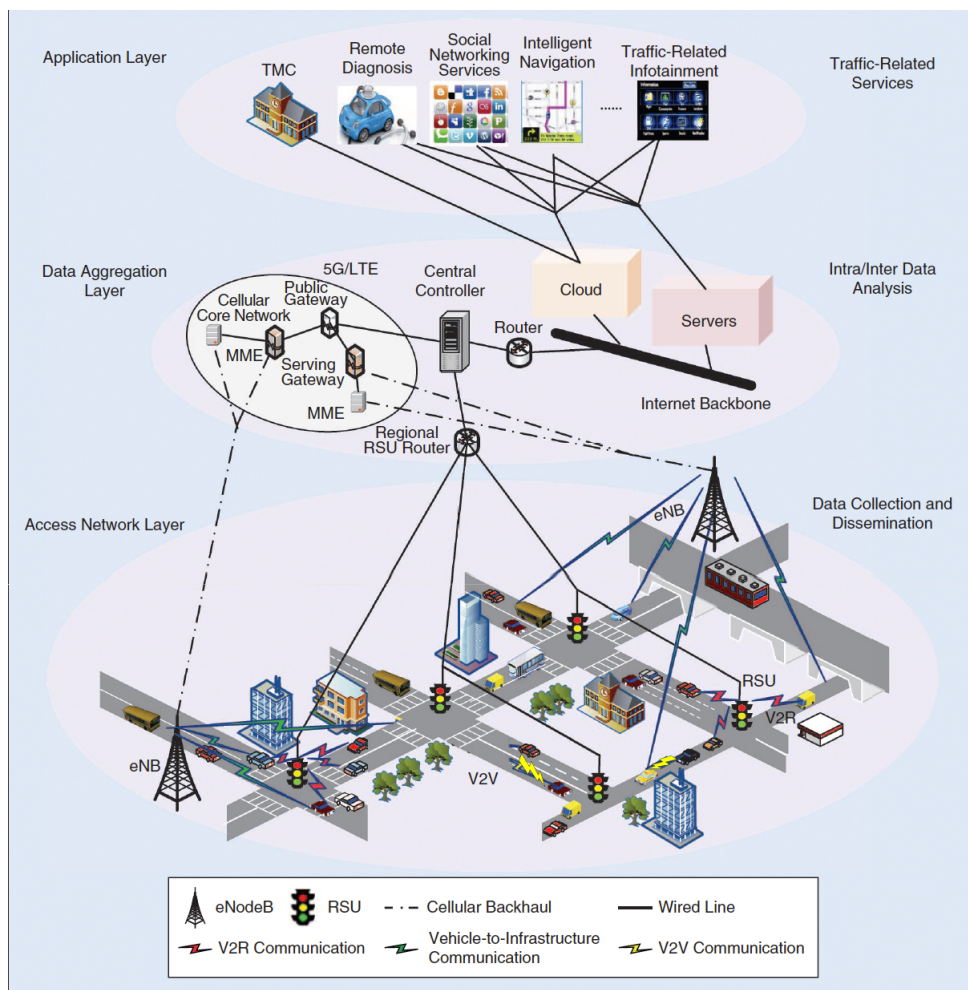


Figure 2.1: VANET environment (extracted from [18]).

Given the differences between the operating environment of an 802.11 WLAN and a vehicular environment, the addition of an amendment to the standard is required. The IEEE 802.11, 1609, European Telecommunications Standards Institute (ETSI) and ITS technical committees framed a group of protocol families collectively known as WAVE, described in Table 2.1, required to support ITS applications, such as data exchange among high-speed vehicles, and between vehicles and the RSU in the licensed ITS band.

Table 2.1: List of the protocols that compose the WAVE communications stack, with the designation of the standard that covers each one (adapted from [19]).

Protocols	Standard document	Purpose of the standards	OSI model layer
WAVE PHY and MAC	IEEE 802.11p	Specifies the PHY and MAC functions required of an IEEE 802.11 device to work in the rapidly varying vehicular environment.	1 and 2
Multichannel operation	IEEE 1601.4	Provides enhancements to the IEEE 802.11p MAC to support multichannel operation.	2
WAVE networking services	IEEE 1609.3	Provides addressing and routing services within a WAVE system.	2, 3 and 4
WAVE resource manager	IEEE 1609.1	Describes an application that allows the interaction of OBUs with limited computing resources and complex processes running outside, to give the impression that they are running in the OBUs.	N/A
WAVE security services	IEEE 1609.2	Covers the format of secure messages and their processing.	N/A

This chapter focuses only on the 802.11p amendment, which specifies layers one and two of the OSI model of the protocol stack, i.e., the PHY and MAC layers. This amendment needs to be stable, guaranteeing interoperability among vehicles made by different manufacturers, and the roadside infrastructures in different geographic locations.

The IEEE 802.11p WAVE architecture foresees the presence of one Control Channel (CCH), reserved for system control and safety messages, and up to six Service Channels (SCHs), used to exchange non-safety data packets (e.g., Internet Protocol (IP) traffic) and WAVE mode short messages [20].

The channel time is divided into synchronization intervals with a fixed length of 100 ms, each consisting of a 50 ms CCH and a 50 ms SCH interval. The amendment introduces an operation called multichannel, in which the vehicular devices monitor the CCH during common time intervals (the CCH intervals), and can optionally switch to one SCH (during the SCH intervals). This way, safety warning messages are transmitted on CCH, while non-safety data applications may run over SCH.

Regarding the **MAC layer**, the operations for the establishment of the connection are less time-consuming than the original 802.11, since vehicular safety applications demand instantaneous data exchange capabilities. Its core mechanism is the same as the one specified in 802.11e, Enhanced Distributed Channel Access (EDCA), which is based on the Carrier Sense Multiple Access with Collision Avoid-

ance (CSMA/CA) scheme.

For the WAVE mode, introduced in the amendment, the data packets transmission is only allowed to occur within a WAVE Basic Service Set (WBSS), which consist of only OBUs or a mix of OBUs and RSUs, seen in Figure 2.2. This way, a station can transmit and receive data frames without the need to belong to a BSS a priori, known as ad-hoc, opposite to what happens with 802.11.

A node that initiates the WBSS is called provider, while a node that joins a WBSS is known as user. To establish a WBSS, the provider periodically broadcasts on CCH a WBSS announcement message, which includes the WAVE Service Advertisement (WSA). This WSA contains the information identifying WAVE applications and respective network parameters, necessary to join a WBSS (e.g., the ID of the WBSS, the SCH it will use and timing information for synchronization). For the joining process, a node needs to monitor all WSAs on the CCH to learn about the existence and the operational parameters of available WBSSs. Then, the node may join the WBSS by switching to the SCH used by the WBSS [20]. This approach offers extremely low overhead for WBSS setup by discarding all association and authentication processes.

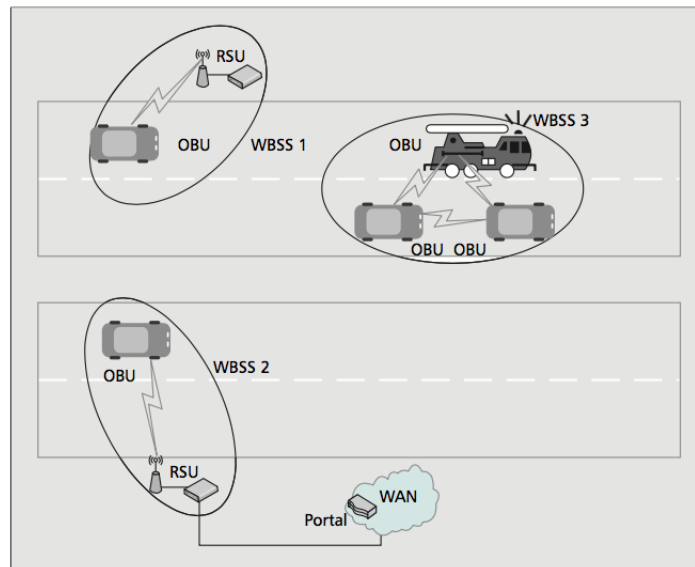


Figure 2.2: WAVE system, with common locations of the OBUs and RSUs, the general makeup of the WBSSs, and the way a WBSS can connect to a WAN through a portal (extracted from [19]).

By default, WAVE units operate independently, exchanging information over CCH. However, the nodes of a particular WBSS exchange information through one or several SCHs [19].

Regarding **PHY**, data will be received from the upper layer, which represents data link layer frames. This layer follows the same frame structure, modulation scheme and training sequences as the IEEE 802.11a PHY one.

Compared to cellular communications, the amendment provides higher transfer rates and smaller communication latencies, for small communication zones.

According to [21], ETSI outlines the frequency ranges allocated to support cooperative ITS in Europe to be a 70 MHz band, ranging from 5.855 GHz to 5.925 GHz, where:

- 5.855 GHz to 5.875 GHz is used in ITS non-safety applications,
- 5.875 GHz to 5.905 GHz is used in ITS road safety related applications,
- 5.905 GHz to 5.925 GHz is used in general ITS applications.

The distribution of the channels in the respective frequency bands can be seen in Figure 2.3.

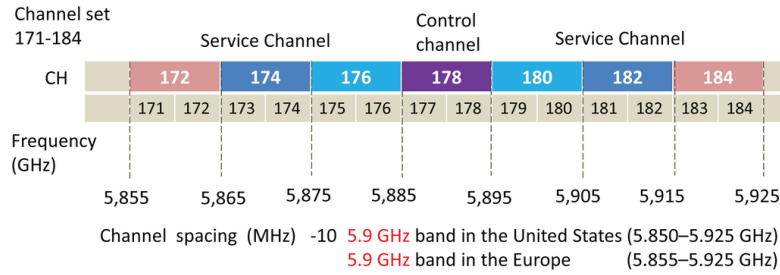


Figure 2.3: Set of channels defined in WAVE (extracted from [22]).

The parameters that characterize 802.11p/WAVE and 802.11a standard, which was used as basis, as well as a comparison between the two, are organized in Table 2.2.

Table 2.2: Difference in parameters for amendments 802.11a and 802.11p (adapted from [23]).

Amendment	802.11a	802.11p	Differences
Radio frequency band [GHz]	5.150 – 5.250, 5.250 – 5.350, 5.470 – 5.725, 5.725 – 5.825	5.855 – 5.875, 5.875 – 5.905, 5.905 – 5.925	
Channel spacing [MHz]	20	10	Half
Data rates [Mbit/s]	6, 9, 12, 18, 24, 36, 48, 54	3, 4.5, 6, 9, 12, 16, 24, 27	Half
Modulation	OFDM	OFDM	No change
Number of subcarriers	52	52	No change
Subcarrier modulation	BPSK, QPSK, 16QAM, 64QAM	BPSK, QPSK, 16QAM, 64QAM	No change
Error control	FEC	FEC	No change
Coding rate	1/2, 2/3, 3/4	1/2, 2/3, 3/4	No change
OFDM symbol duration [μs]	4	8	Double
Guard time [μs]	0.8	1.6	Double
Preamble duration [μs]	16	32	Double
Subcarrier spacing [MHz]	0.3125	0.15625	Half

The 802.11p amendment, as it happens with 802.11a, uses the Orthogonal Frequency Division Multiplexing (OFDM) modulation scheme to multiplex data, which works by splitting the radio signal into multiple smaller sub-signals. The main reason for using it is the high spectral efficiency, i.e., all orthogonal subcarriers are simultaneously transmitted. Orthogonal Frequency Division Multiple Access (OFDMA) provides shared access to multiple users by assigning subsets of subcarriers to individual users.

The OFDM channel bandwidth is halved in order to keep abreast the requirements of VANETs, resulting in a 10 MHz bandwidth (instead of 20 MHz in 802.11a). This way, OFDM symbols are longer in the time domain, its length being twice ($8\mu\text{s}$) the one of 802.11a ($4\mu\text{s}$), enabling the system to deal with large delay spreads and avoid Inter-Symbol Interference (ISI).

Also, carrier spacing is reduced by half, typically 0.15625 MHz, compared to 802.11a, whose spacing is 0.3125 MHz. In summary, the change in bandwidth affects all OFDM timing parameters used in the regular 802.11a transmissions, by doubling them. Consequently, the transmission rate is reduced by half [24].

The 802.11p/WAVE offers data exchange among vehicles, V2V, and between vehicles and roadside infrastructure, V2I, within a range up to 1 km and a vehicle velocity up to 260 km/h.

For the work developed in this thesis, the used operation frequency lies in the 5.9 GHz band, and its importance is highlighted by the European decision to use the 5.875 GHz to 5.905 GHz frequency band for ITS applications related to safety.

In summary, this amendment is characterized by the following features, improvements, requirements and/or changes:

- communication in a highly mobile environment,
- half the data rates of 802.11 with 10 MHz channels,
- one CCH and six SCHs,
- a unique ad-hoc mode,
- random MAC access,
- high accuracy for the Received Signal Strength Indication (RSSI),
- 16 Quadrature Amplitude Modulation (QAM) used in the high speed mobile environment,
- option for a more severe operating environment,
- priority control,
- power control.

2.2 Vehicular Scenarios

The important characteristics that distinguish vehicular communications from the other types of wireless communications consist of several diverse environments, the existence of different communication types, namely V2V, V2I, V2P, etc., and the scatterers, static or mobile objects between Tx and Rx, that affect vehicular communications [8].

In a V2V scenario, unlike what happens in a cellular one, there is neither an AP nor a Base Station (BS). The Rx and the Tx may move at high velocities, the antennas are mounted at a relatively low height (1 m

to 2 m), the relevant scatterers may be highly mobile (e.g., other vehicles) and the distances between Tx, Rx and these scatterers are high (in the range of a few hundred meters).

In the case of a V2I scenario, the propagation channel will depend on the height of the BS and the AP. If both are elevated high, the V2I can be approximated by a cellular (micro cell) scenario. On the other hand, if this height is low, the V2I becomes similar to the V2V case.

Regarding the surrounding environment where communication happens, propagation is mostly influenced not only by objects like buildings, vehicles, both static and mobile, and several types of vegetation, but also their number, size and density.

The most common vehicular environments have also been widely investigated for cellular systems, being the following:

- **Urban:** streets are generally wide, with one or more lanes in each way, depending on the region. Houses are close to the curb and traffic density is often high. In this scenario, two relevant and specific situations can occur, such as the **urban crossing**, where the two communication node vehicles are both considered to be approaching the crossing, and the **urban approaching**, where the receiving and transmitting vehicles are placed on the same road.
- **Suburban:** streets are typically narrower than in urban areas and mostly have one or two lanes in each way. Houses are more set back from the curb and traffic density is lighter.
- **Highways:** usually have two to six lanes in each way, with a divider separating each one, and there are no houses in the vicinity. Traffic density ranges from very high on urban highways to considerably lower on rural ones.
- **Rural:** generally have one lane in each way and few or no buildings nearby. Typically with hills and vegetation that can provide additional MPCs. Traffic density is normally low.

This propagation environment classification is sufficient for most of the vehicular network applications, such as approaching emergency vehicle warnings, hazardous location notifications or traffic condition warnings, since these applications are useful for almost any type of road [3].

However, signal propagation is strongly impacted by the varying presence, location and density of road-side objects, as well as the velocity and density of vehicular traffic. Therefore, this classification is insufficient for application specific scenarios, like pre and post-crash warnings, where a more dedicated channel characterization is needed.

Even though intersections are part of the aforementioned scenarios, they are examples of a set of specific channel characteristics that need to be properly assessed, e.g., how the LOS path is obstructed (by buildings) when two cars approach an urban intersection from perpendicular directions.

The average vehicle node speeds and distance between the nodes, where communication happens, is detailed in Table 2.3.

Table 2.3: Average vehicular communication measures for vehicle speed and distance between nodes.

	Average speeds [km/h]	Inter-node distances [m]
inside urban areas	20 – 50	few meter - 100
inside rural areas	10 – 40	few meters - 400
highways	50 – 120	~ 1000

The thorough description of the previously presented scenarios, associated with vehicular communications, is relevant since it provides critical and important situations that need to be further modelled individually.

2.3 Vehicular Channel Characterization

2.3.1 Channel Impulse Response

Vehicular channels are wireless multipath links, meaning the signal at the receiver results from contributions from several propagation paths. As a result, the signal experiences fading, variations of the received signal power with time, and signal echoes with different delays. The complex characteristics of a vehicular channel result in complex propagation environments, which are difficult to model [3].

In a vehicular scenario, the receiver moves along the route at some constant velocity, v . For a fixed position, d , the channel between Tx and Rx can be modelled as a linear time invariant system. However, the signal encounters several scatterers, mobile or static, and the spatial location of the receiver changes over time, causing the appearance of different multipath waves with different propagation delays, amplitudes and phases.

The mobile radio channel can then be modelled as a linear time varying channel that changes with time, frequency and distance. This way, the channel impulse response can be described in the time, frequency and spatial domains. This thesis focuses mainly on the time description and, therefore, the impulse response of a multipath channel in time, h .

The key component that characterises the channel response is fading, namely large-scale and small-scale. **Large-scale fading** characterises the mean received signal strength. It is the result of signal attenuation due to signal propagation over large distances (between Tx and Rx), path loss and diffraction around large objects in the propagation path, known as shadowing [25].

Small-scale fading is characterized by rapid fluctuation of the received signal strength and phase, causing time and frequency dispersion. Depending on the nature of the transmitted signal, the channel

and the velocity, the time and frequency dispersion lead to four possible distinct effects: flat fading and frequency selective fading, based on multipath time delay spread, and fast fading and slow fading, based on Doppler spread. The small-scale variations of a mobile radio signal can be directly related to the impulse response of the mobile radio channel.

The impulse response, $h(t, \sigma)$, is a wideband channel characterization, and contains all information necessary to simulate or analyze any type of radio transmission through the channel. It is function of both the time variations due to motion, t , and the channel multipath delay for a fixed value of time, σ [26].

The impulse response is expressed in (2.1) [26], where the unit impulse function δ determines the specific MPCs that are non-zero at time t and excess delays σ_i . The phase term $2\pi f_c \sigma_i(t) + \phi(t, \sigma)$ represents the phase shift due to free space propagation of the i th MPC plus possible additional shifts encountered in the channel. This term can be substituted by a single variable $\Theta_i(t, \sigma)$, which lumps together all phase shifts of a single MPC.

$$h(t, \sigma) = \sum_{i=0}^{N_{MPC}-1} a_i(t, \sigma) \cdot e^{[j(2\pi f_c \sigma_i(t) + \phi(t, \sigma))]} \delta(\sigma - \sigma_i(t)) \quad (2.1)$$

where:

- a_i : real amplitudes of the i th multipath component at time instant t ,
- σ_i : excess delays of the i th multipath component at time instant t ,
- f_c : frequency of the carrier,
- N_{MPC} : total possible number of multipath components,
- δ : unit impulse function.

In a wireless link, the transmitted signal may encounter several mobile or static objects (such as buildings, vegetation or other vehicles), creating several propagation paths. This is known as **multipath**. The signal copies at the receiver arrive with a certain delay, σ , and may suffer phase shifting.

For small-scale channel modelling, the power delay profile, $p(\sigma)$, of the channel can indicate the intensity of a signal received through a multipath channel, as a function of time delay. It is given by the spatial average over a local area, and its calculation is shown below [26].

$$p(\sigma) = |h(t, \sigma)|^2 \quad (2.2)$$

The **delay spread** measures the difference between the time of arrival of the earliest significant MPC (typically the LOS component, if exists) and the time of arrival of the latest one. The mean delay of the channel is given below [27].

$$\bar{\sigma} = \frac{\int_0^\infty \sigma p(\sigma) d\sigma}{\int_0^\infty p(\sigma) d\sigma} \quad (2.3)$$

The RMS delay spread, σ_{RMS} , which is typically used, is given below [27].

$$\sigma_{RMS} = \sqrt{\frac{\int_0^\infty (\sigma - \bar{\sigma})^2 p(\sigma) d\sigma}{\int_0^\infty p(\sigma) d\sigma}} \quad (2.4)$$

The delay spread may change considerably depending on the environment, and also over time in a single environment, particularly due to the movements of Tx, Rx and scatterers in a V2V scenario.

The movement of scatterers along the propagation path changes the transmitted signal given frequency, due to the different incident angle, creating **Doppler effect**. The received signal has slight frequency variations from the original signal due to this interaction with the moving objects. The signals travelling along different paths have different rates of change in phase, corresponding to Doppler shifts. The difference in Doppler shifts between the direct signal and the largest Doppler shift from a multipath corresponds to the **Doppler spread**, f_D , which is given below [27].

$$f_D = \frac{v \cdot f_c}{c} \quad (2.5)$$

where:

- v : scatterer velocity,
- c : speed of light.

The maximum values for Doppler spread on a vehicular scenario can be up to four times higher than on a cellular one for the same velocity. The speed of a scatterer that approaches the Tx contributes to the Doppler shift value twice the value of its speed. If in this case the scatterer also approaches the Rx, there is a second contribution, meaning that the total Doppler shift will be four times higher than the speed of the scatterer vehicle. The Doppler spectrum can change rapidly in a vehicular channel, meaning that the conditions for wide-sense stationary are violated.

The mobility of the Rx and Tx, along with the movements of the interacting objects in the propagation environment, result in significant non-stationary channel fading. The vehicular channel impulse response exhibits significant delay and Doppler spread, which also change over the time and frequency axes. The channel statistics change with the evolution of the Rx, Tx and interacting objects geometry over time.

2.3.2 Tapped Delay Line

The channel model described in the previous Subsection 2.3.1 can be represented by a delay line with a number of taps equal to exactly the number of MPCs, N_{MPC} . This is known as **Tapped Delay Line (TDL)**, represented in Figure 2.4, where each impulse describes an MPC, associated with a delay σ_i , provoked by a group of one or more scatterers. The TDL model allows to represent the channel impulse response by an approximated version, using a delay line with N_{taps} taps, where each tap is a scatterer group and c_i is the channel coefficient in instant i .

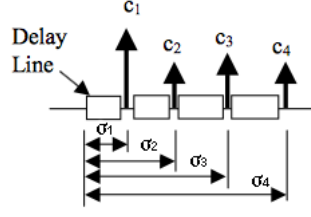


Figure 2.4: Tap Delay Line model (adapted from [25]).

A TDL is a delay line with at least one "tap". A delay line, exemplified in Figure 2.5, extracts a signal $x(n)$ from somewhere within the delay line, optionally scales it, and allows it to be delayed by a certain number of samples M_i , by summing the signal with other taps, forming an output signal $y(n)$.

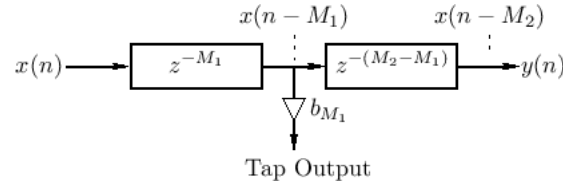


Figure 2.5: Delay line with one tap (extracted from [28]).

The delay is represented by the discrete Fourier Z-transform parameter, in which the amount of delay is given by the amount of samples M_1 and M_2 . Each tap releases a signal scaled by b and delayed by the amount defined by z^{-M} , as shown by the signals $x(n - M_1)$ and $x(n - M_2)$.

A more general example of a tap delay line with an arbitrary number of taps is exemplified in Figure 2.6, where the signal is delayed in each tap by a single sample and scaled by b_i , adding up at the output $y(n)$.

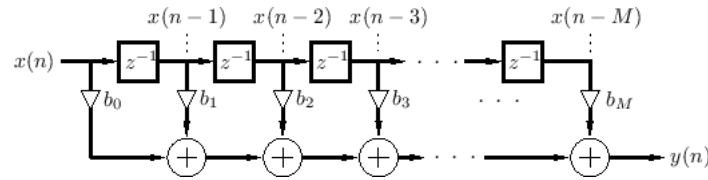


Figure 2.6: Example of a Tap Delay Line, with an arbitrary number of taps (extracted from [28]).

The TDL discrete representation of the channel impulse response in (2.1) is shown in (2.6) [29], for a number of taps equal to the number of all MPCs, N_{MPC} .

$$h(t, \sigma) = \sum_{i=1}^{N_{MPC}} a_i h_{i,WSS} \delta(\sigma - \sigma_i) \quad (2.6)$$

where:

- a_i : gain of the i th tap,
- σ_i : delay of the i th tap,
- $h_{i,WSS}$: realization from a WSS fading process.

To determine the delay parameters from discrete data, the RMS delay spread, σ_{RMS} is given below [30].

$$\sigma_{RMS} = \sqrt{\frac{\sum_{i=0}^{N_{MPC}-1} \sigma_i a_i^2}{\sum_{i=0}^{N_{MPC}-1} a_i^2} - \mu_\tau^2} \quad (2.7)$$

The mean energy delay, μ_τ is expressed below [30].

$$\mu_\tau = \frac{\sum_{i=0}^{N_{MPC}-1} \sigma_i a_i^2}{\sum_{i=0}^{N_{MPC}-1} a_i^2} \quad (2.8)$$

To simulate the non-stationary process inherent to vehicular channels in a low-complexity manner, local scattering regions in time and frequency are implemented, where each one is assumed to be stationary and, therefore, modelled by a distinct local WSS scattering function.

Figure 2.7 shows an example of a time varying discrete-time impulse response of a multipath channel, where t varies into the page, N_{MPC} are the number of significant MPCs, provoked by groups of one or more scatterers, and the time delay bins are quantized to widths given below.

$$\Delta\sigma = \sigma_{i+1} - \sigma_i \quad (2.9)$$

In case more than one multipath signal arrives within an excess delay bin, their combination yields the instantaneous amplitude and phase of a single MPC. This way, several sets of scatterers can be grouped into the same MPC, and hence, associated with a single tap.

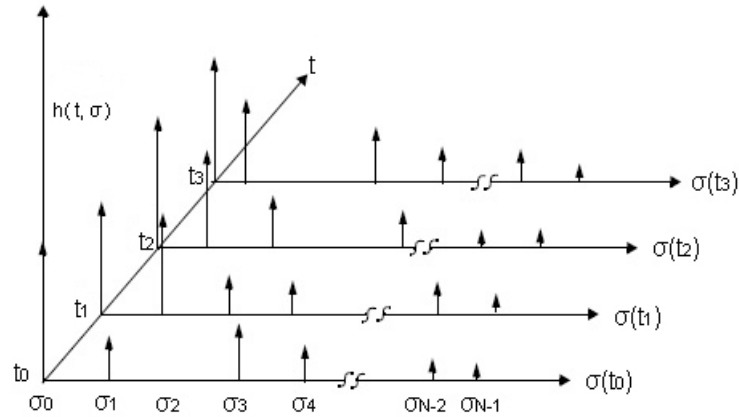


Figure 2.7: Example of time varying discrete-time impulse response model for multipath channel (adapted from [26]).

2.3.3 Influence on the Scenarios

Considering the environments discussed in Section 2.2 and the channel characterization in Subsection 2.3.1, it is relevant to discuss how propagation parameters influence vehicular environments, where communication can happen.

For a **rural** scenario, NLOS MPCs are characterized by moderate Doppler shifts and strong attenuation parameters, whereas the LOS one, for stationary scenarios, is approximated by an Additive White Gaussian Noise (AWGN) channel.

In the **urban** scenario, two specific situations need to be taken into account: the urban crossing, which represents the NLOS scenario, and the urban approaching, which addresses the LOS one.

For **urban crossing**, the NLOS component is justified by the fact that the direct path is barely free of obstacles, leading to low power relative to backscattered paths. Taking the heterogeneous nature of the scatterers into account, they can be arranged in sets (to be characterized afterwards by a tap each).

The first set can be the moving cars positioned between the communication nodes. The multipath signal will be characterized by small delays and small Doppler shifts, which is due to the scatterer exhibiting positive relative velocities in respect to one of the communication nodes and negative to the other.

A possible second scatterer set is cars that are retreating from both vehicles, which will result in negative Doppler shifts and larger delays.

Finally, a third set of scatterers may be vehicles that are far away, and are being approached by both communication nodes. This is undoubtedly the weakest group and has the largest positive Doppler shifts.

Regarding the **urban approaching** case, since there is no crossing travelling by the vehicles, the backscattering happens earlier than in the urban crossing case. The relative speeds are higher resulting in higher relative Doppler shifts.

The final scenario is the **highway**. Due to higher velocities and an open setting, the Doppler shifts and delays will be higher. Vehicles will travel in the same or parallel lanes, meaning that LOS scatterers will either approach one of the communication nodes and retreat from the other, or will approach both. The NLOS component is due to possible existence of large obstacles in the way of the communication, which results in larger delay spreads.

2.4 Channel Emulator

With the increasing development of technology that takes advantage of more complex channel characteristics, channel modelling necessary to accurately emulate the radio environment becomes even more critical to a test setup.

Using real hardware in real environments to perform experiments and measurements is the most realistic approach to characterize vehicular channels. However, the cost and repeatability issues make this

approach unfeasible for large-scale evaluations [8].

One possible solution is channel simulation, which ensures high repeatability. However, the design of a realistic channel simulator is challenging, since either the simulation environment has to be highly detailed to account for all aspects of a real system, or it has to make certain assumptions about the real-world.

A solution that lies between real-world channel experimentation and simulation is channel emulation. A radio channel emulator, also called fading simulator, is a tool for air interface testing in wireless communications. In a test environment, a radio channel emulator replaces the real radio channel between radio transmitter and receiver, providing to the receiver a faded representation of the transmitted signal.

In emulation, some components of the real communication systems are used together with simulated ones. The goal is maintaining the repeatability and configurability inherent of simulation environments, combined with the high level of realism of a testbed.

The channel emulator takes as input the signals generated by real devices, subjects them to simulated realistic signal propagation models and finally feeds them back into the real devices. The procedure is illustrated in Figure 2.8.

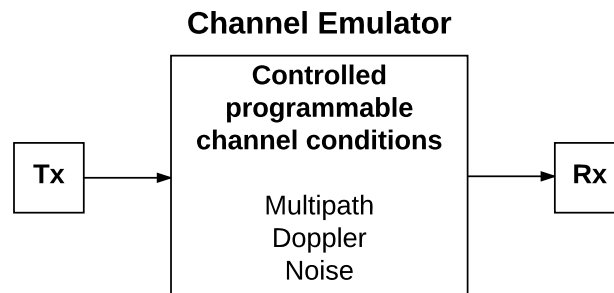


Figure 2.8: Channel emulator system (extracted from [31]).

The use of emulators keeps to a minimum the tiresome task of performing successive field measurements, relying only on obtaining the channel model, if there is not an available one already. The rest of the experiments are carried out inside a testing lab.

Although emulators and simulators are sometimes used interchangeably, the emulator mimics the internal behaviour of the device and its hardware.

Some commercial channel emulators, such as the ones developed by Keysight Technologies, Anritsu and Azimuth, enable the reproduction of a real-world radio wave propagation environment of mobile wireless scenarios.

Keysight Technologies [32] developed a Radio-Frequency (RF) fading channel emulator to test wireless devices and the network infrastructure, by simulating the characteristics of real-world radio channel

conditions, within a laboratory environment. This characteristics include pathloss, multipath fading, delay spread, Doppler spread, polarization, correlation and spatial parameters, which are critical for Multiple-Input and Multiple-Output (MIMO) and multi-radio system performance.

Azimuth [33] offers a family of RF channel emulators for link performance testing in 2G/3G, Wi-Fi, LTE, LTE Advanced, Heterogeneous Network (HetNet) and beyond.

In academia, Jamison et al. [34] developed a highly configurable and repeatable multipath channel emulator, which accounts for IoT environments, expected to have large signal path losses due to NLOS conditions, reflective surfaces and/or moving components in the environment. To generate channels, the emulator uses a database of over 2000 pre-defined channel conditions, ranging from Ricean to hyper-Rayleigh distributions. A reconfigurable antenna array is used in order to obtain over 200 million different configurations that are controlled by LabVIEW™ software [35], which also controls the attenuators and saves the measurements.

2.5 State of the Art

The use of an appropriate channel model is critical for creating an accurate evaluation of vehicular protocols and applications before the actual deployment in real-life situations.

Sen et al. [30] performed channel measurements to develop V2V channel models for 5 GHz. Measurements were conducted for several different environments, varying antenna locations and at different times of the day, for small and large cities. A subsequent statistical characterization of the Channel Impulse Response (CIR) and propagation path loss was performed via the collection of Power Delay Profiles (PDPs). The amplitude statistics were modelled using flexible Weibull distribution. The number of taps was calculated using the Root Mean Square (RMS) delay spread's maximum value, and only the samples above given thresholds for autocorrelation and noise were considered. To reduce complexity, a persistence process based on Markov chains was implemented to include only the taps that contributed 99% of the time, which provided a faithful non-stationary and non Wide Sense Stationary with Uncorrelated Scatterers (WSSUS) representation of the channel.

An emulator created to replicate the physical behaviour of the vehicular wireless channel was developed by Acosta-Marum et al. [36]. Six models, three for V2V and other three for V2I, were developed based on measurements at 5.9 GHz made in Atlanta. The considered model was TDL, where each tap process was described as having Rice or Rayleigh fading. Each model captured the joint Doppler-delay characteristics of a different environment. An analysis of PER for each model showed that it generally decreases with decreasing Doppler offsets and widths. Since the channel with highest PER was V2V on a highway, the article concludes that this is the channel that should be used for certification tests.

Fernández-Caramés et al. [37] developed a Field-Programmable Gate Array (FPGA) based vehicular channel emulator aimed at assessing the performance of IEEE 802.11p transceivers, using the models described previously in [36]. The channel impulse response in a given time instant and path resulted from the contribution of LOS and NLOS components, generated, respectively, by Doppler spread, phase and angle of arrival, and by a Gaussian distribution. The generation of these parameters was performed offline to reduce the amount of required FPGA resources, and they remained constant throughout the emulation. The transmitted signal was filtered with the interpolated coefficients, Finite Impulse Response (FIR) filter. The article concludes that a system must be capable of transferring messages to and from vehicles at speeds of 193 km/h with a PER of less than 10%, which happens for a Signal-to-Noise Ratio (SNR) of 15 dB, in highway, and 10 dB, in urban. FIR based emulators are computationally efficient, but complexity rapidly becomes impractical for larger array sizes (MIMO) [38].

Spirent [39] provides commercial solutions for channel emulation, and developed a V2X testbed that can be used for validation and performance benchmarking of V2X applications. Commercial emulators are usually aimed at evaluating a specific technology and, although most of them are robust and work well for most of the applications, they are generally expensive and may not offer enough flexibility to researchers when configuring the channel parameters.

Nilsson et al. [40] describes a Multipath Propagation Simulator (MPS), which simulates the wireless channel using multiple antennas, long fibre-optic delay lines and phase shifters. The MPS consists of an array of antennas, each of which represents one particular delay.

Vlastaras et al. [41] addresses the problem of stress testing and evaluating modem transceivers by implementing a 1x1 Single-Input and Single-Output (SISO) real-time wireless channel emulator, using a Software-Defined Radio (SDR). The model implements a two-tap delay line channel model, where both taps have equal amplitudes but the second's Doppler shift and delay vary. The test is applied on two different IEEE 802.11p modem implementations, for several packet sizes, and a characterization of PER performance is made for different Doppler combinations. The article concludes that PER increases with the increase in packet size, and a good modem performance is achieved with small packet sizes.

3

Model and Implementation

This chapter describes the methods to determine the tap components, both for stationary models, through probability distributions, as well as for non-stationary, through an optimization algorithm, and its implementation on a further developed real time channel emulator, the ground basis for the development of this thesis.

3.1 Channel Models

3.1.1 Stationary Tests from Distributions

The emulator used as basis for the work developed in this thesis [29, 42] takes as input the channel impulse response values in form of an array, with a pre-defined number of delay taps per time snapshot. The channel models are TDL models, described in Subsection 2.3.1.

The scenarios considered for the emulator are Rural, Urban and Highway, and the results in [29] were used to generate the data for the channel models, for up to 4 taps. The rural scenario is used as a baseline for the other ones, being approximated by an AWGN channel.

For a time instant i , each tap is characterised by the scenario deterministic parameters, such as attenuation a_i , RMS delay σ_{RMS} , and the stochastic characterization of the small-scale fading process h_i .

The first tap is static, meaning the delay, gain and maximum Doppler frequency are zero, being used as normalization for the other taps. The taps realizations for each time instant and scenario, stored in vector \mathbf{h}_i , were generated according to Rayleigh distribution

Taking the maximum Doppler frequency $f_{D,i}$ as the design parameter, the choice for the shape of the Power Spectral Density of the fading process is a Half-bathtub [29], due to the fact that backscattered waves are not uniformly distributed in power. This can be caused by non-omnidirectional antenna patterns or if scatterers are only located in front or behind the receiving communication node, which are characteristic situations of vehicular communications, mostly due to the asymmetric placement of scatterers and communication happening predominately along the road.

In order to test the emulator, the models for the used scenarios (rural, urban and highway) employ stationary processes with Rayleigh statistics, meaning that the small-scale fading trace, generated for each tap individually, are complex valued. The channel models present some shortcomings due to their simplicity, namely the pure LOS static tap in every scenario, which is not always motivated by the propagation model.

Given that when there is a dominant stationary non-fading signal component present, such as an LOS propagation path, the small-scale fading envelope distribution is Rice, the static tap may be more accurately represented by this distribution, which takes possible objects obstructing the Fresnel ellipsoids into account.

The random MPCs arriving from different angles are superimposed on a stationary dominant signal. At the receiver, this has the effect of adding a Direct Current (DC) component to the random multipath. This way, the use of the Rice distribution facilitates the use of the channel models on scenarios with dense

traffic, and when cars at relative low speeds contribute with a non-negligible MPC to the LOS.

When the dominant signal gets weaker, the composite signal resembles a noise signal, which has a Rayleigh distributed envelope, evidenced by the probability density function in Appendix A.1. Taking into account that the Rice distribution degenerates to a Rayleigh one when the dominant component fades away, as seen by the probability density function in Appendix A.2, a method to generate Rice samples from a Rayleigh method can be deducted.

The Rayleigh procedure to generate channel coefficients can be adapted to produce coefficients using the Rice distribution, by multiplying the resulting tap components, for the whole time duration, with a normalization factor and summing a DC component equal to one. This is expressed in (3.1), which needs to be determined for each of the N_{taps} intended taps.

$$\mathbf{h}^{\text{Rice}} = \mathbf{1} + \mathbf{h}^{\text{Ray}} \cdot \frac{1}{\sqrt{k_{\text{Rice}} \cdot \overline{|\mathbf{h}^{\text{Ray}}|^2}}} \quad (3.1)$$

where:

- $\mathbf{h}^{\text{Rice}} \in \mathbb{C}^{N_t \times 1}$: vector containing all the time components for Rice distribution, for a single tap,
- $\mathbf{h}^{\text{Ray}} \in \mathbb{C}^{N_t \times 1}$: vector containing all the time components for Rayleigh distribution, for a single tap,
- $\mathbf{1} \in \mathbb{C}^{N_t \times 1}$: array of ones the same dimension as the Rayleigh one,
- $\overline{|\mathbf{h}^{\text{Ray}}|^2}$: mean value of the Rayleigh components array in time,
- k_{Rice} : factor to assure complex Rice components are not larger than 1,
- N_t : number of time instants.

The mean value used in the previous expression is determined using Matlab, and its calculation can be seen below.

$$\overline{|\mathbf{h}^{\text{Ray}}|^2} = \frac{1}{N_t} \sum_{j=1}^{N_t} |\mathbf{h}^{\text{Ray}}|^2 \quad (3.2)$$

With all distributions defined, the generation of random samples to form the coefficients of each scenario taps vector, along time, is possible.

3.1.2 Non-stationary from Measured Data

Vehicular channels are characterized by a non-stationary fading process, and so, local regions in time and frequency are defined where the process is assumed WSS.

The model is described by a time-variant TDL that groups sets of scatterers, each characterized by one of the N_{taps} taps. This solves the problem of modelling multiple scatterer groups (MPCs), characterized by different fading statistics, with low implementation complexity.

In order to feed the non-stationary channel data into the emulator, these data need to be organized in vectors of magnitude, phase and respective delays.

To use the TDL simplification to emulate the desired channel, it is mandatory to use a method to choose the most important MPCs of the channel without compromising the quality of the emulation.

The data used to calculate the taps is organized in a matrix, shown in (3.3), where rows represent time slots and columns represent delayed MPCs.

$$\mathbf{H} = \begin{bmatrix} - & \mathbf{h}_1 & - \\ - & \mathbf{h}_2 & - \\ & \vdots & \\ - & \mathbf{h}_i & - \\ & \vdots & \\ - & \mathbf{h}_{N_t} & - \end{bmatrix} \quad (3.3)$$

where:

- $\mathbf{H} \in \mathbb{C}^{N_t \times N_{MPC}}$: channel matrix,
- $\mathbf{h}_i \in \mathbb{C}^{1 \times N_{MPC}}$: channel vector with delayed multipath components for each time instant i .

The problem then becomes finding the approximated channel matrix, $\mathbf{H}_{app} \in \mathbb{C}^{N_t \times N_{MPC}}$, in which MPCs in each row are all zero, except the N_{taps} (intended number of taps) strongest ones, whose values of magnitude and phase are such that the energy between the original matrix, \mathbf{H} , and the approximated one is the same.

For an arbitrary number of taps N_{taps} , the determination of the optimum components (magnitude, phase and delay) is done by minimizing the error between the real channel matrix and the approximated one. The Mean Squared Error (MSE) is a measure of the quality of an estimator and, therefore, the chosen method.

The optimum estimator, for a single timeslot, can be given by determining the argument that minimizes the MSE, in (3.4). The calculation is made in frequency, hence the presence of the frequency channel response by calculating the Fast Fourier Transform (FFT) of the time channel matrix.

$$\hat{\mathbf{h}}_{app,i} = \arg \min \text{MSE}(\text{FFT}\{\mathbf{h}_{app,i}\}, \mathbf{h}_i) \quad (3.4)$$

where:

- $\hat{\mathbf{h}}_{app,i} \in \mathbb{C}^{1 \times N_{MPC}}$: estimator for the approximated channel vector in a single time slot i ,
- MSE : Minimum Square Error calculation,
- FFT : Fast Fourier Transform operator.

Using a similar nomenclature as described for the channel in (3.3), the calculation of the approximated channel matrix is made for each time slot, for efficiency purposes, which means each approximated

channel matrix row, $\mathbf{h}_{app,i} \in \mathbb{C}^{1 \times N_{MPC}}$, is calculated one at a time, by using each row of the original data matrix, \mathbf{h}_i .

The result of the error minimization for the optimum estimator is an approximated channel matrix in which each row has all elements equal to zero, except the intended N_{taps} ones, which are the chosen taps. Also, it guarantees that the transformation from the original channel matrix to the approximated one is unitary, meaning the energy remains the same.

A possible first approach is choosing the strongest among all MPCs in each row, each associated with a delay, and calculate their magnitudes so that the total energy of the signal is the same. This is accomplished by the following expression, where the calculation is made for each time instant i .

$$\frac{1}{N_{MPC}} \sum_{j=1}^{N_{MPC}} |h_{i,j}|^2 = \frac{1}{N_{MPC}} \sum_{j=1}^{N_{MPC}} |h_{app,i,j}|^2 \quad (3.5)$$

However, the choice of the main peaks of the signal will be random for each time slot, mainly in the case of highly time varying channels as vehicular ones. This makes the correlation between the chosen peaks in each time slot unknown (and possibly low). Another problem arises with the calculation of phases, since the preservation of the total energy between the channel matrix and the approximated one only foresees the calculation of the magnitudes. This leads to a rough approximation for each tap phase by equating them, for example, to the mean among all phases in the original channel time snapshot vector.

Instead, the Lasso [43] algorithm solves the problem by shrinking some coefficients and setting others to zero, leading to stable-low variance (ridge regression) and interpretable results (subset selection).

The results in [43] were the starting point to solve the optimization problem in (3.4). The method used for calculating the taps was based on c-Lasso [44] for complex values. In order for the algorithm to work better, the data arrays used in the optimization are normalized, meaning the obtained results will not have the same energy as the original data. This problem is solved after the results are obtained.

Considering arbitrary data (\mathbf{x}^i, y_i) , for $i = 1, 2, \dots, N$, where $\mathbf{x}^i = (x_{i1}, \dots, x_{ip})^T$ are the predictor variables and y_i are the responses, the Lasso estimate $(\hat{\alpha}, \hat{\beta})$ is defined by (3.6). The parameter $t_s \geq 0$ determines the amount of shrinkage applied to the estimates.

$$(\hat{\alpha}, \hat{\beta}) = \arg \min \left\{ \sum_{i=1}^N (y_i - \alpha - \mathbf{x}_i^T \beta)^2 \right\} \quad \text{subject to } \sum_{j=1}^p |\beta_j| \leq t \quad (3.6)$$

where:

- $\hat{\beta} = (\hat{\beta}_1, \dots, \hat{\beta}_p)^T$: intended estimate,
- $t_s \geq 0$: tuning parameter.

For all t_s , the solution for α is $\hat{\alpha} = \bar{y}$, which means it can be assumed without loss of generality that $\bar{y} = 0$, and hence omit α . The computation of the solution in (3.6) is a quadratic programming problem

with linear inequality constraints [43].

Developing (3.4), for a single time instant, it takes the form of (3.6), as shown below.

$$\hat{\mathbf{h}}_{app,i} = \arg \min \left\{ \sum_{i=1}^{N_{MPC}} (\mathbf{h}_i - \mathbf{F} \cdot \mathbf{h}_{app,i})^2 \right\} \quad (3.7)$$

where:

- $\mathbf{F} \in \mathbb{C}^{N_{MPC} \times N_{MPC}}$: Fast Fourier transform matrix.

As previously explained, the constraint of the problem under study focuses on limiting the number of non-zero components, in a single time slot, to the number of intended taps.

In order to do a variable selection in each row of the matrix, the I_0 -Norm is used, as described in [45]. Although I_0 lacks the homogeneity property (thus, is not strictly speaking a norm), most mathematicians and engineers use the definition of I_0 as the number of non-zero coefficients in a vector, as expressed below, for an arbitrary vector.

$$\|\mathbf{c}\|_0 \stackrel{def}{=} \# \{i : c_i \neq 0\} \quad (3.8)$$

where:

- \mathbf{c} : arbitrary vector,
- $\#$: cardinality function,
- i : arbitrary index of \mathbf{c} .

Taking this I_0 definition into consideration, the constraint in (3.6), applied to the problem in study, evolves to the one in (3.9). The number of intended taps, N_{taps} , is the tuning parameter described above for Lasso.

$$\|\mathbf{h}_{app,i}\|_0 \leq N_{taps} \quad (3.9)$$

However, I_0 is non-convex and its solutions are computationally difficult to solve, due to the non-convexity of the corresponding objective function. This way, a possible solution is to solve a similar convex problem.

The solutions of the I_1 -Norm are a good approximation to the ones of I_0 . The norm I_1 of an arbitrary vector \mathbf{c} is defined below.

$$\|\mathbf{c}\|_1 \stackrel{def}{=} \sum |c_i| \quad (3.10)$$

Before applying I_1 in the problem constraint, the inequality in (3.9) has to be adapted to take the form of the one in (3.6).

Given the definition in (3.10) of a sum of coefficients (in a vector), a satisfying approximation is given in

the following expression.

$$\|\mathbf{h}_{app,i}\|_1 \leq t_s \quad \text{for} \quad t_s = k_{taps} \cdot \|\mathbf{h}_i\|_1 \quad (3.11)$$

where:

- k_{taps} : constant dependent on the number of intended taps .

The tuning parameter constant k_{taps} in (3.11) was obtained by trial-and-error. It was assumed that, instead of limiting the number of non-zero coefficients in each time instant channel vector by the number of intended taps, the constraint would be limiting the I_1 of the estimated vector, sum of all coefficients, to a certain k_{taps} (less than zero) amount of the original channel vector I_1 .

Ultimately, the problem becomes, at last, a quadratic programming problem with a convex inequality constraint, which can be solved computationally rather easily. The used software is the CVX modelling system for Matlab [46].

Since the algorithm performs better on a normalized array, when the optimization problem is solved, the obtained complex values for the chosen taps do not have the same energy as the original data. To solve this, a least squares fit is performed between the approximated model and the original non-normalized data, taking the chosen taps into consideration. This way, the results are an approximated model with the same energy as the original one and, therefore, it is ready to be directly fed into the emulator.

3.2 Emulator Basic Description

This section introduces the emulator as the ground basis for the work, described in [42], and its further development by adding functionalities that allow efficient and consecutive scenario data emulation along with varying delays associated with each tap.

The emulator in [42] solves the problem of simulating a channel characterized by significant non-stationary fading, in which not only the channel exhibits high delays and Doppler spreads, but also these parameters change over the time and frequency axes. i.e., the channel statistics change as the geometry of Rx, Tx and interacting objects change over time.

The definition of local scattering regions, in time and frequency, defining WSSUS regions, where the fading process is assumed stationary, is essential. This way, each region is modelled distinctly by a local scattering function.

The emulator architecture aims at an optimized and low complexity (hardware) architecture, while providing flexibility for reconfiguring the scattering model for each local stationary region. This allows the calculation of the non-stationary channel responses and, at the same time, updating the local scattering

functions, while switching among different ones.

The platform for the emulator is NI USRP Rio 2953R [15], and optimizes the trade-off between reconfigurability of the system and the lowest possible computational complexity. Additional information about the platform is found in Appendix B.

NI USRP RIO works with frequencies up to 6 GHz and the emulation bandwidth is 20 MHz. Fading traces are pre-generated and loaded at runtime. The emulator takes the TDL models directly as input.

Regarding the system level architecture, the channel emulator is divided into three modules: propagation, antenna and convolution, as seen in Figure 3.1.

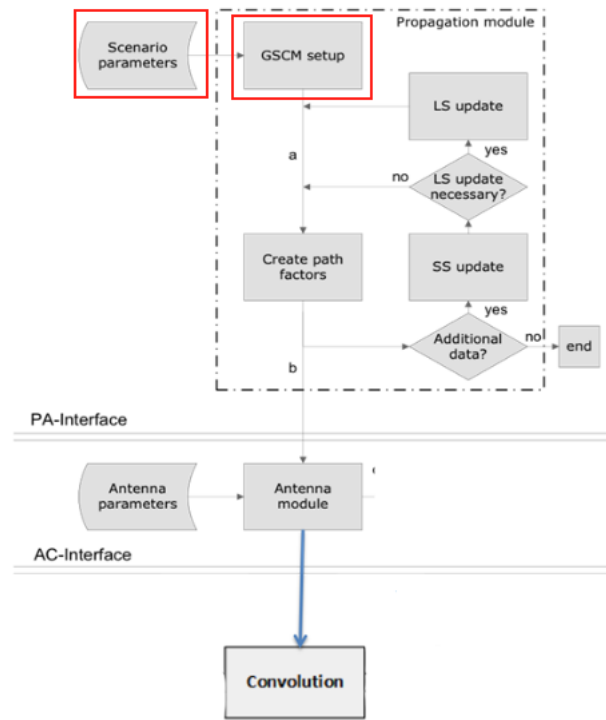


Figure 3.1: System view of the emulator (adapted from [42]).

The **propagation module** is firstly responsible for acquiring the scenario specific input, in the form of a vector seen in Figure 3.2, containing the channel values for all taps in all time instants, and setting the global parameters, such as center frequency and sampling rate. Secondly, this module computes and passes on the parameters associated with each MPC, such as the attenuation coefficients, path delays and angles of arrival and departure. The implementation emulates a Geometric-based Stochastic Channel Model (GSCM) [47], which takes non-stationary properties into account, and assumes local stationary regions, switching between them.

To account for the rapid environmental changes characteristic of vehicular communications, the propagation module updates the previous parameters frequently. The parameters do not vary at the same rate, pathloss parameters being slow varying comparatively to those associated with the relative position

Time instant	1			...				N_t			
Index	1	2	...	N_{taps}	$N_{taps}(N_t - 1) + 1$	$N_{taps}(N_t - 1) + 2$...	$N_t N_{taps}$
Tap	1	2	...	N_{taps}	1	2	...	N_{taps}

Figure 3.2: Channel input vector, for all taps N_{taps} in each time instant and time duration N_t .

of scatterers.

There are two updates rates that can be defined: large scale (LS) and small scale (SS). The geometry updates are calculated by the LS rate, determined by the stationary time of the channel. The SS fading update is determined by the Doppler bandwidth and is necessary for accurately modelling Doppler contributions.

The **antenna module** deals with the calculation of the attenuation parameters for each MPC, according to the orientation of the antenna. These results, along with the corresponding path delays, are stored in a fast memory, which feeds the corresponding parameters to the FPGA, on to the convolution module (through the AC interface). Both propagation and antenna modules are implemented in the host PC.

With all the necessary parameters obtained in the previous modules, it is possible to calculate the discrete time varying impulse response of the channel, $h(t, \sigma)$.

The **convolution module** has the role of real-time calculating the output stream, $r(t)$, through the convolution of the channel discrete time-varying impulse response with the input stream, $s(t)$. The convolution procedure, as well as the signal transmission and reception, is shown in the block diagram of Figure 3.3.

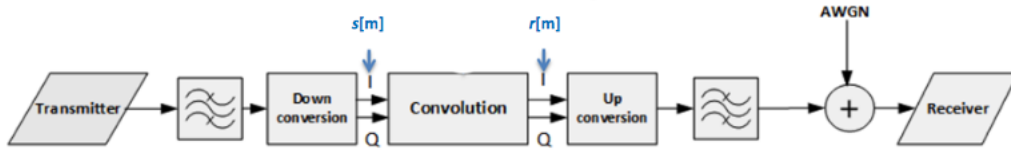


Figure 3.3: Blocks diagram for the signal transmission, convolution with simulated channel and reception (adapted from [42]).

In order to perform a low complexity and accurate convolution and consequent calculation of a discrete version of the output stream, $r(t)$, the chosen architecture is that of a TDL, with equally spaced taps, explicit in Figure 3.4. Doppler frequency shifts are implemented in real-time.

The number of taps for this architecture is 68, so that it covers the contributions of the reflectors that are 1 km away, and the resolution is 50 ns. Only a maximum of 10 taps can be active.

The impulse response samples are updated through the host, in the host PC, and passed on, to the emulator (where it is connected). The host PC, running LabVIEW™ 2014, is responsible for loading the compiled bit files into the FPGA, where the emulator is implemented. The emulator is also connected to

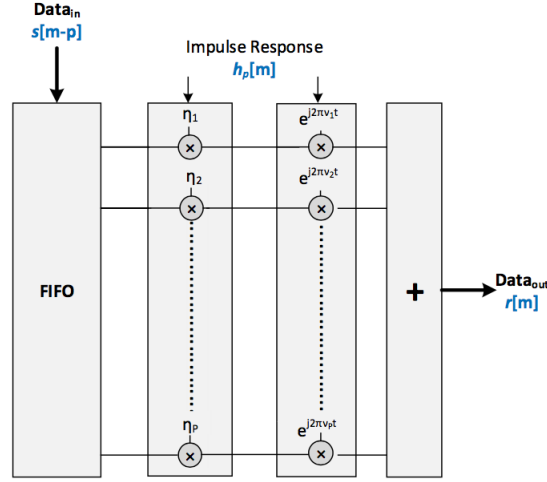


Figure 3.4: Tapped delay line architecture for the emulator convolution module (extracted from [42]).

two IEEE802.11p modems, both set up as Tx and Rx, as seen in the measurement setup in Figure 3.5.

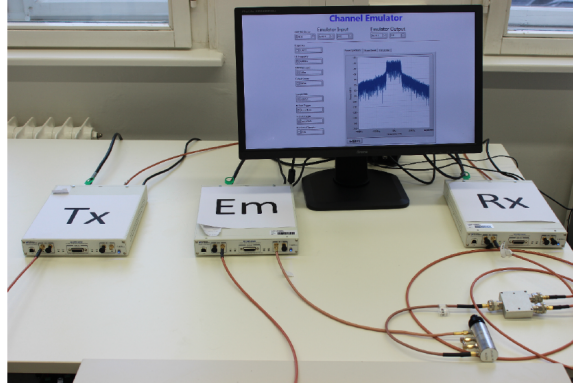


Figure 3.5: Measurement setup, with the Tx, channel emulator and Rx. The emulator's interface is shown as well in the monitor (extracted from [29]).

In summary, the data generated by the Tx modem is input to the FPGA emulator, processed through the FPGA, which acts as the channel and, finally, the signal is output of the FPGA emulator and received by the Rx modem.

The architecture of the emulator should be capable of reproducing the behaviour (input and output) of a non-stationary N_{taps} - tap wireless vehicular propagation channel, although it only allows the introduction of fixed delays for all time instants.

Figure 3.1, shown previously, highlights in red the emulator sections to which this thesis is dedicated to: scenario parameters calculation, shown in Section 3.1, and GSCM setup, where parameters are fed to the emulator, updated during runtime, and passed on to be written into the FPGA, explained in the next section.

3.3 Emulator Development

3.3.1 Transmission and Reception of Scenario Data

The problem of channel emulation implementation can be decomposed into sub-modules, whose top-down modular design is seen in Figure 3.6.

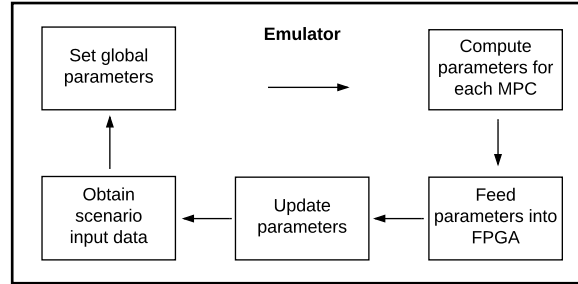


Figure 3.6: Top down modular design for the emulator.

The main host code is responsible for the capture, processing and passing on of data to be written into the FPGA that performs channel emulation. The problems to be solved in this thesis are part of the host code, namely data collection and forwarding sub-modules. The delay components are sent, received, and written the same way the magnitude and phase ones, and are updated once every 10 time instants the other ones are, enabling the updating of its values along the channel duration.

First of all, channel data need to "enter" the main program in an automated and efficient manner, so that the testing of multiple (consecutive) scenarios, propagation environments and conditions, such as attenuations, packet size, etc., is possible. Also, this makes possible the transmission of the channel delays, associated with each time instant taps, which vary in time. This way, a characterization of a vehicular non-stationary environment is possible.

Thereunto, since the processing and writing of data into the FPGA emulator is done inside the main program cycle, data is sent continuously through an User Datagram Protocol (UDP) link, run in the same machine, and received in the host code. The program responsible for sending the data was implemented in Python and the routine is explained in Figure 3.7.

A UDP connection is established, in the same machine, between the Python transmitter program and the receiver in LabVIEW main program.

Data is arranged in arrays of magnitude, phase and delay, where the $N_{taps} = 16$ consecutive components correspond to the taps in a time instant. Since the arrays dimension is generally very big and there are three types of data in form of array to be sent, they have to be divided in sub-arrays in order to be sent through a UDP link without being lost.

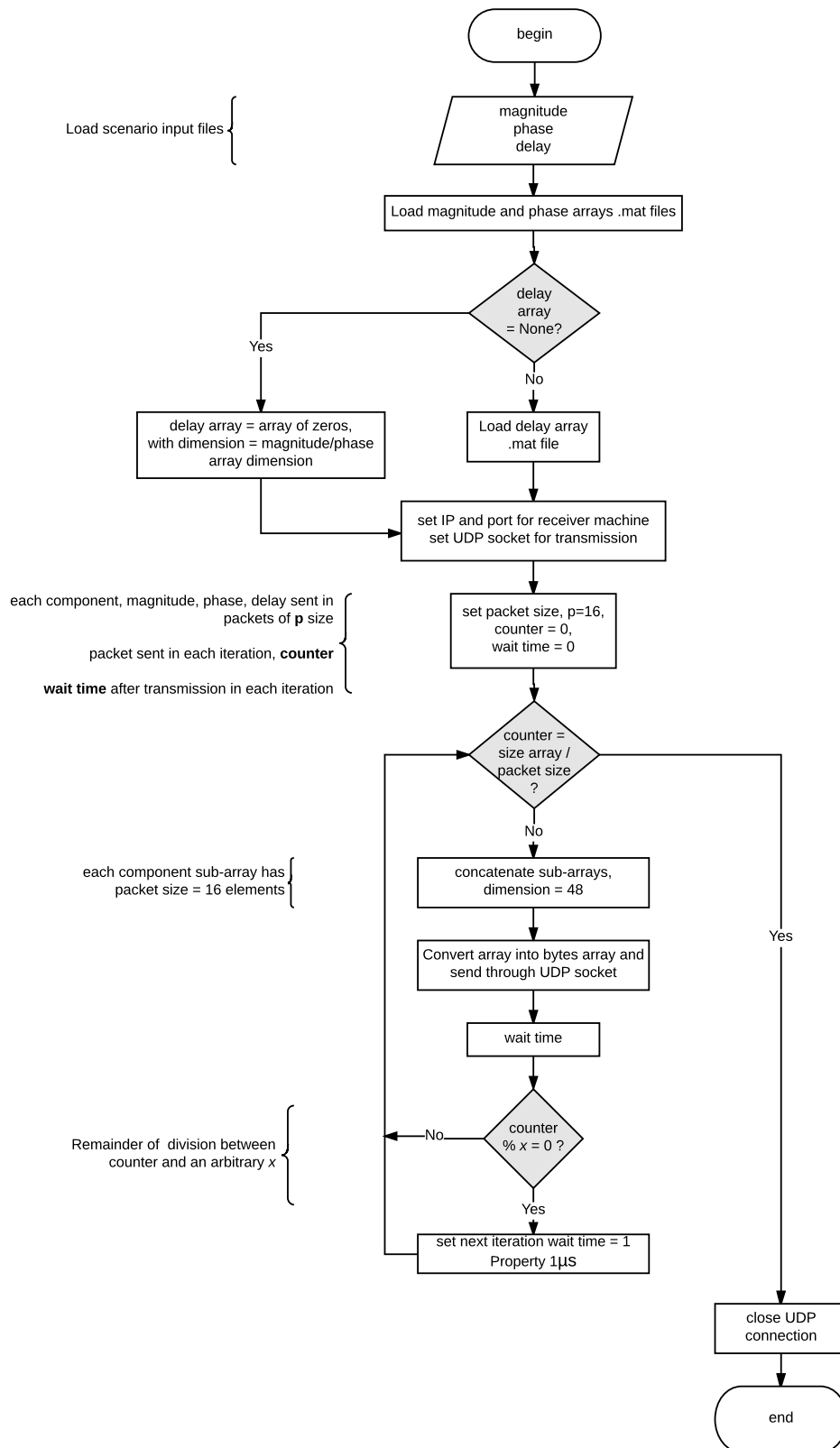


Figure 3.7: UDP transmission.

The sub-arrays for the three channel parameters, magnitude, phase and delay, are concatenated in a single array, whose size, the biggest possible without any data being lost, was chosen to be 48 coefficients, 16 for each parameter.

A delay in sending data was also added, since the processing time of the host code was much slower than the one for sending, and an overload of data in the receiver made the main program cycle slower.

Regarding the reception of the data module in the host code, its procedure is shown in Figure 3.8.

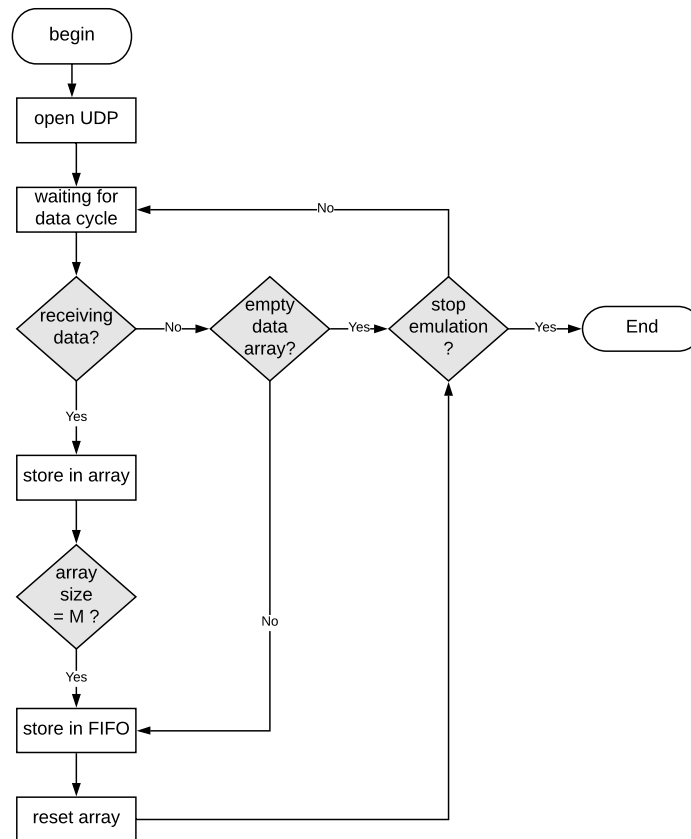


Figure 3.8: UDP reception.

After the UDP connection is established between the transmitter code in Python and the receiver function in LabVIEW, outside the main code processing cycle, the program begins a parallel cycle to the main processing/writing one, where it stays waiting for new data until the emulation is stopped.

3.3.2 Data Updating

Since the host code capability of processing data in the main cycle is slower than receiving it, the sub-arrays (of size 16) in the UDP receiving cycle are concatenated into bigger sub-arrays. Only when these arrays reach the size of 1600 they are passed on into the main cycle, by storing the arrays in a First In First Out (FIFO) type of queue and retrieving them from it in the main cycle. The procedure, explained

in Figure 3.9, is the continuation of the UDP receiving section.

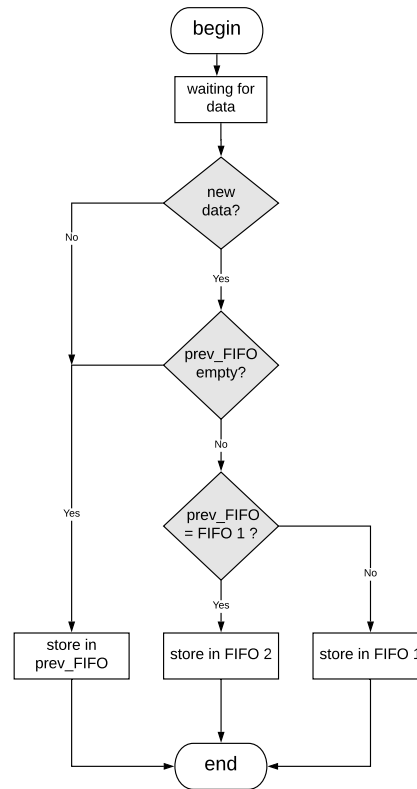


Figure 3.9: Store data in FIFO.

In order to run consecutive scenarios, and since the UDP transmission of data is faster than processing it, two FIFO queues were created. In the case of the first scenario data still being passed on into the main cycle and new data arriving in the program at the same time, the second FIFO is used for the new data while the first is emptied. This way, when the first scenario finishes running, the second is immediately retrieved from the second FIFO, without mixing data.

Moving to the host code cycle, the removal of data arrays from FIFO is detailed in Figure 3.10. The aim is to test which of the FIFOs has elements and remove the data from the non-empty one. If both have elements, in order not to mix data, the program retrieves it from the FIFO chosen in the previous cycle iteration, i.e, the one that stores data from the scenario that was sent first. This way, all data sent first is retrieved, and only then the program switches to the other FIFO, storing the next scenario.

As the magnitude, phase and delay sub-arrays are retrieved from FIFO, in each cycle iteration, they are continuously concatenated in arrays and saved in array variables until they are complete. The procedure is detailed in the pseudo-code of Figure 3.11.

After deciding between the two possible FIFO queues, if the program is not receiving any data at a given time and the chosen FIFO is empty, or data arrays are complete and the chosen FIFO was switched, meaning new data is being received from a new scenario, the complete array variables are erased so that they are ready to receive new data. On the other hand, if no data is being received but the FIFO still

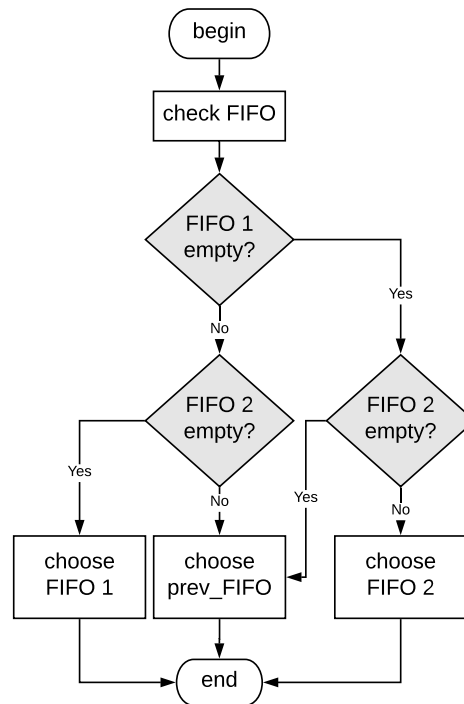


Figure 3.10: Choice of FIFO for data removal.

```

1: procedure UPDATE
2:   if not receiving data or switched FIFO then
3:
4:     if switched FIFO or chosen FIFO is empty then
5:       Reset data arrays
6:       return empty data arrays
7:     else
8:       Remove data from FIFO
9:       Append new data into data arrays
10:      return updated data arrays
11:    end if
12:  else
13:
14:    if chosen FIFO is empty then
15:      return data array from previous clock
16:    else
17:      Append new data into data arrays
18:      return updated data arrays
19:    end if
20:  end if
21: end procedure
  
```

Figure 3.11: Update and passing of data arrays.

has elements, then they are retrieved and added to the array variables. In both cases, the empty and updated arrays are passed on to the writing sub-module.

If the program is receiving data or the chosen FIFO did not change (no new scenario), then two situations can happen depending on whether FIFO is empty or not. In case it is, the array variables from the previous cycle iteration are passed on, until some data is put on queue. Otherwise, they are retrieved, the array variables are updated and passed on.

3.3.3 Data Writing

After the arrays are updated, they need to be passed on to the writing sub-module. However, given the nature of the LabVIEW platform, the array variables may be updated faster than the time needed to write the previous data into the FPGA, when new data from a new scenario arrives. Therefore, data is delayed a cycle iteration by a flip-flop type structure that stores data. This way, if the program switches FIFO (new data is stored in queue), the flip-flop passes the complete data array variables from the previous scenario, without passing the ones ready to receive new data.

The way arrays are written into the FPGA depends if it is the first scenario being run by the emulator. The program is implemented in a way that in the first case scenario, the updated data arrays are passed on to be written in all iterations, being constantly updated. To avoid writing the same data in every iteration, arrays are rotated so that elements already read are put in the end of the array, as seen in pseudo-code of Figure 3.12.

```
1: procedure ROTATE
2:   if not receiving data then
3:     Rotation = 0
4:     return original array
5:   else
6:
7:     if previous rotation  $\geq$  size of current data array then
8:       previous rotation  $\div$  size of data array
9:
10:    if Remainder = 0 then
11:      Rotation = 0
12:      return original array
13:    else
14:      Rotation = quotient  $\times$  size of data array
15:      return data array rotated
16:    end if
17:  else
18:    Rotation = previous rotation
19:    return data array rotated
20:  end if
21: end if
22: end procedure
```

Figure 3.12: Rotate array.

For the rest of the cases, the program waits until the next scenario's data arrays are complete, and only then are they sent to be written. In these cases, there is no need to rotate the arrays, because they are passed on already complete.

In order to test if data arrays are complete, the program waits until the receiver cycle stops receiving new data from the scenario. Until that happens and data is completely retrieved from the queue, the array is still incomplete. When all data is removed from the given scenario chosen FIFO (even if new data arrives and put on the other FIFO), the array is complete.

Finally, when a given scenario data is completely received, the aim of the program is to write its complete data arrays continuously and repetitively, in a loop, until new scenario data is received and its arrays are complete. The variable state is introduced to decide on whether data arrays are allowed to pass into the writing sub-module or not. If the chosen FIFO switched thanks to a new scenario, the state variable only allows data to be passed on when the arrays are complete. In the remaining cases, for the same FIFO, the state variable depends on being or not the first case. If yes, then the updated data arrays are always allowed to be continuously written, otherwise, it depends on the arrays being complete or not.

The aforementioned tests, checking the first case, the arrays state of completeness or if data is allowed to be passed on to the writing sub-module are explained in pseudo-code of Figures 3.13, 3.14 and 3.15, respectively.

```
1: procedure FIST CASE
2:   Initialization: it is the first case
3:   if it is the first case then
4:
5:     if data array is complete then
6:       Not first case any more
7:       return first case = false
8:     else
9:       It is still the first case
10:      return first case = true
11:    end if
12:  else
13:    Not the first case
14:    return first case = false
15:  end if
16: end procedure
```

Figure 3.13: Test to check if the scenario to be emulated is the first to be received by the emulator.

```

1: procedure STATE OF ARRAY
2:   if stopped receiving data then
3:
4:     if switched FIFO or empty FIFO then
5:       return Data array is complete
6:       return state of array = true
7:     else
8:       Data array is incomplete
9:       return state of data array = false
10:    end if
11:  else
12:    Data array is incomplete return state of data array = false
13:  end if
14: end procedure

```

Figure 3.14: Test to check if array with the scenario data is complete.

```

1: procedure STATE
2:   if switched FIFO then
3:     return state = state of array (complete or incomplete)
4:   else
5:
6:     if it is first case then
7:       return state = true
8:     else
9:       return state = state of array (complete or incomplete)
10:    end if
11:  end if
12: end procedure

```

Figure 3.15: Test to see if array is ready to be written into FPGA.

3.4 Model Assessment

The performance evaluation of the emulator was an essential step in its development process, in order to ensure data was correctly sent, received and passed, and to assess the proper functioning of the different steps of the added functionalities on the main code and the results for the different scenarios.

The several assessment tests performed on the emulator are shown in Table 3.1, where one describes

the nature of each one, which step of the main code it belongs to, and how it was performed.

Table 3.1: List of assessment tests made on the emulator do validate its performance.

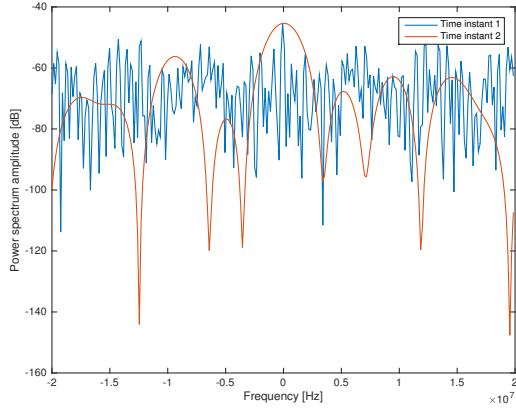
Nr.	Description	Method
1	Validation of the main code scenario input, by verifying if the decoded data, sent by UDP is correct.	Decode the input, knowing data are magnitude, phase and delay sub-arrays, of type double and fixed size.
2	Validation of data arrays passed on to main code cycle to be written, by checking if sub-arrays were correctly appended.	Send a single-active tap array with consecutive values for each time instant, e.g. 100020003000..., for a 4-tap case.
3	Validation of the special cases, by examining if data passed on is written, to be emulated, at the correct time.	Examine if main code receives and writes data continuously on the first scenario, and waits until all data is received to emulate the next scenario.
4	Validation of the emulated scenario, by confirming if the data spectrum obtained in the emulation is correct.	Comparison between the emulator and Matlab simulated spectrums, for a fast-varying scenario with a sudden slow-varying and smooth jump.

The first three tests were made to assess the performance of the emulator during its development by checking if data that was being sent through UDP to the main code was being correctly received, stored in the arrays, passed on and written into the FPGA.

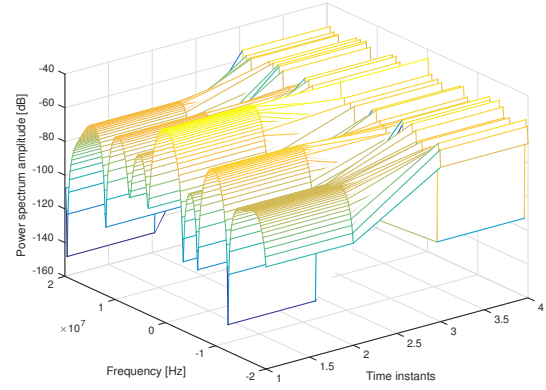
To assess the correct functioning of the completed emulator, the final test was performed directly in the output results by comparing them to others from simulations made on Matlab.

The use of a fast-varying stationary channel model with fixed delay values and a sudden jump, in time, to a slow-varying one, characterized by different delays, was implemented and compared with the spectrum obtained in the emulator. The behaviour of the spectrums in both cases, obtained though simulation on Matlab, can be seen in Figure 3.16(a), and the sudden transition is very evident in Figure 3.16(b).

The obtained spectrums from the emulator are shown in Figure 3.17, where 3.17(a) represents the fast-varying channel and 3.17(b) represents the sudden slow-varying one. This test enables to easily check that the emulator is performing correctly by evidencing plainly the sudden transition in the magnitude behaviour for this channel scenario.

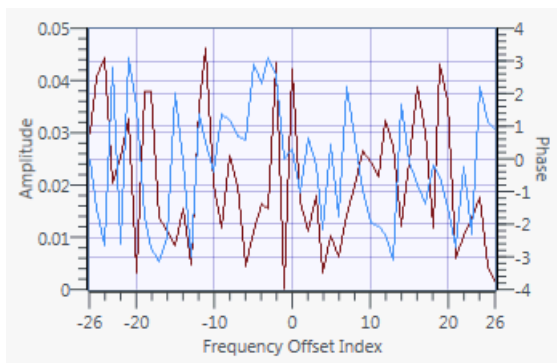


(a) Before transition, time instant 1 and after transition, time instant 2.

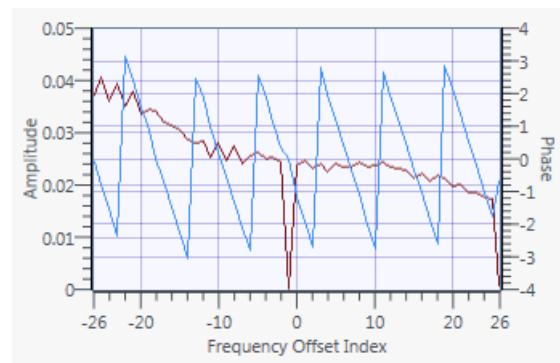


(b) Spectrum for transition during an arbitrary number of instants.

Figure 3.16: Spectrum of the channel model before and after sudden transition.



(a) Magnitude and Phase before transition.



(b) Magnitude and Phase after transition.

Figure 3.17: Emulator output spectrum of the channel before and after sudden transition.

4

Results Analysis

This chapter introduces an analysis of the chosen taps, obtained through optimization, in the non-stationary scenario, comparing them with the original data. The emulator output results, the packet error ratio, are discussed for both stationary and non-stationary scenarios, for different packet sizes and attenuations.

4.1 Measured Data Scenario

The used data was obtained by means of a measurement campaign conducted in 2009 called DRIVE-WAY [48], in the city of Lund, Sweden. The measurement campaign considered a dedicated subset of communication and traffic scenarios, and was conducted in order to perform a realistic characterization of propagation channels for safety related ITS applications.

The key features include performance in traffic situations that are of interest for these applications, such as intersections, traffic congestion and merge lanes. The use of realistic antennas designed for vehicular usage and realistic antenna mounting, the investigation of important propagation mechanisms, such as LOS obstruction and the benefits of multi-antenna usage are also analyzed. The measurement parameters are displayed in Table 4.1.

Table 4.1: Measurement campaign parameters (adapted from [48]).

Center frequency [GHz]	5.6
Measurement Bandwidth [MHz]	240
Sample rate [MHz]	640
Test signal length [μs]	3.2
Number of Tx antenna elements	4
Number of Rx antenna elements	4
Number of channels	16
Number of snapshots	32500
Time snapshot resolution [μs]	307.2
Frequency/Delay resolution [MHz]	312.5
Recording time [s]	~ 10

The channel sounder, used to obtain the measurements, provides the sampled transfer function data in form of several consecutive three-dimensional matrices, $H[m, q, p]$ of size $[500 \times 769 \times 16]$. The rows, m , are the time instants (snapshots), being 500 per matrix; the columns, q , are the 769 delayed components per time snapshot, in frequency; and p represents the 16 several possible links between the 4 transmitter antennas and the 4 receiver antennas. The analysis is made for the link between the first receiver and transmitter antennas, and the matrix organization is seen in Figure 4.1.

This thesis focuses on the urban road crossing scenario, on a single lane, with obstructed LOS, surrounding buildings and parked cars on all sides of the intersection. The scenario is shown in Figure 4.2, where vehicle C is the transmitter and vehicle D is the receiver.

Figure 4.3 shows that vehicle D moves on a straight line, at a speed of 40 km/s, towards position B. Meanwhile, vehicle C moves in a different lane, without seeing vehicle D, and slows down from a speed of 40 km/s to 0 km/s, changing direction on the left towards vehicle D.

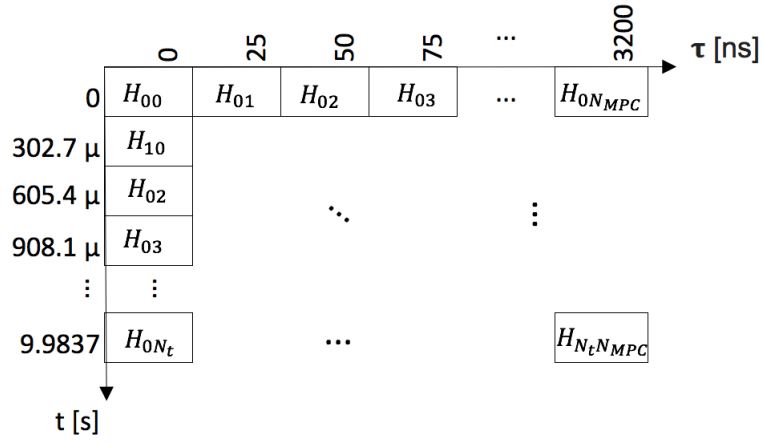


Figure 4.1: Organization of the measurements matrix.



Figure 4.2: Urban crossing in Lund, Sweden. A, B, C and D represent vehicles (extracted from GoogleMaps).

The measurements end after the crossing of the vehicles, moment when they both move away from each other, on the same lane, and gradually slow down.

4.2 Data Processing

For the non-stationary measurements analyzed in the previous section, the original channel frequency data representation can be seen in Figure 4.4(a), and the frequency response magnitude of the approximated channel, obtained using the model in Section 3.1.2, can be seen in Figure 4.4(b).

When comparing the two channel behaviours, it is possible to infer that the energy of the signal in the approximated case is concentrated around the delay component (in frequency) of 0 Hz while the remaining components are weaker. Overall, both channels have a similar behaviour, since the normalization

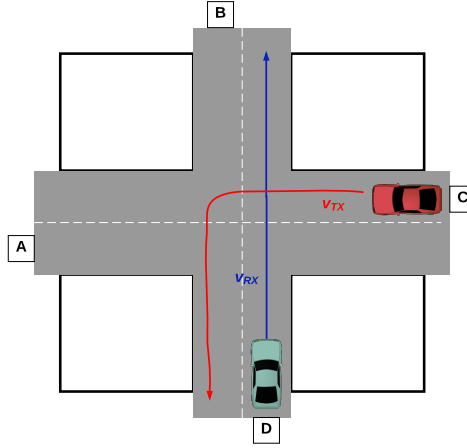


Figure 4.3: Urban crossing. The receiver D moves at speed v_{Rx} and the transmitter C moves at speed v_{Tx} .

performed on the model taps enables both signals to have the same energy. The strongest component is therefore the one with the smallest delay and the signal's energy is concentrated in this area.

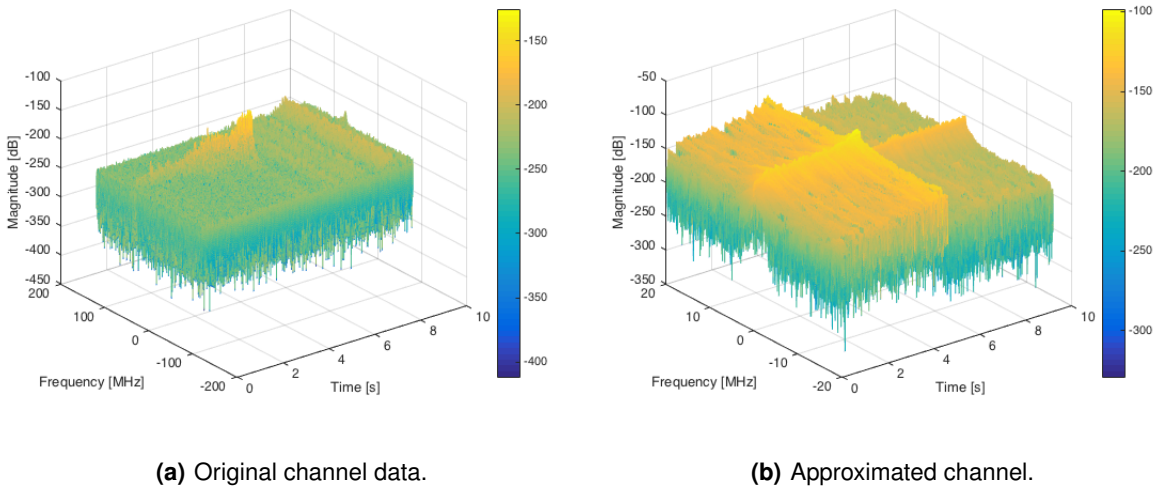


Figure 4.4: Magnitude of the non-stationary channel along time and delay.

The approximated frequency response in Figure 4.4(b) shows the high correlation between the chosen taps for the MPCs along the time axis, demonstrating that the algorithm to choose the taps is highly efficient. It is also evident the highly irregular behaviour of the channel, which is inherent to the non-stationarity of vehicular channels, mainly the urban ones.

The channel sounder results of the scenario measurements, such as its frequency response and PDP, can be seen in Figure 4.5. The beginning of the scenario is characterized by an NLOS component, result of the reflections on buildings and/or other objects of the surrounding two roads, as the Tx and the Rx vehicles approach the crossing. This is evident in the figures since the received power is low until the channel reaches time instant of approximately 7 s.

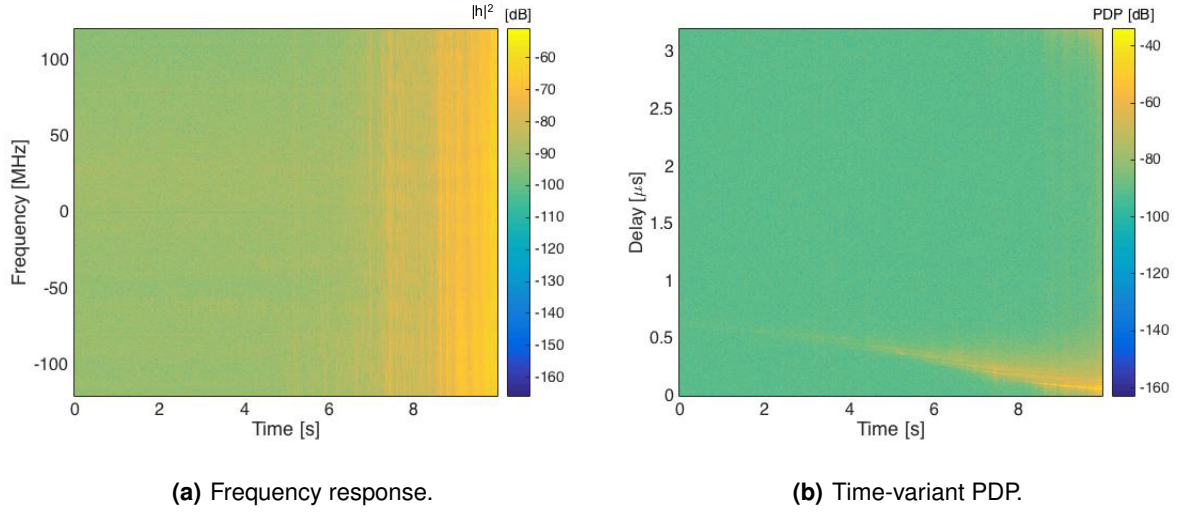


Figure 4.5: Channel sounder results for urban crossing scenario.

After the initial NLOS behaviour, a strong component appears at around 8 s, evidenced by the high increase in power, which indicates the presence of an LOS path. This situation occurs when the two vehicles are in the crossing and the closest to each other. After that, they both move away.

Figure 4.6 shows the channel frequency response and PDP for the approximated version of the channel matrix, which keeps the main contributions of the ones in the original data channel. The image noise is due to the plot determination being made on a matrix with a high number of zero-components.

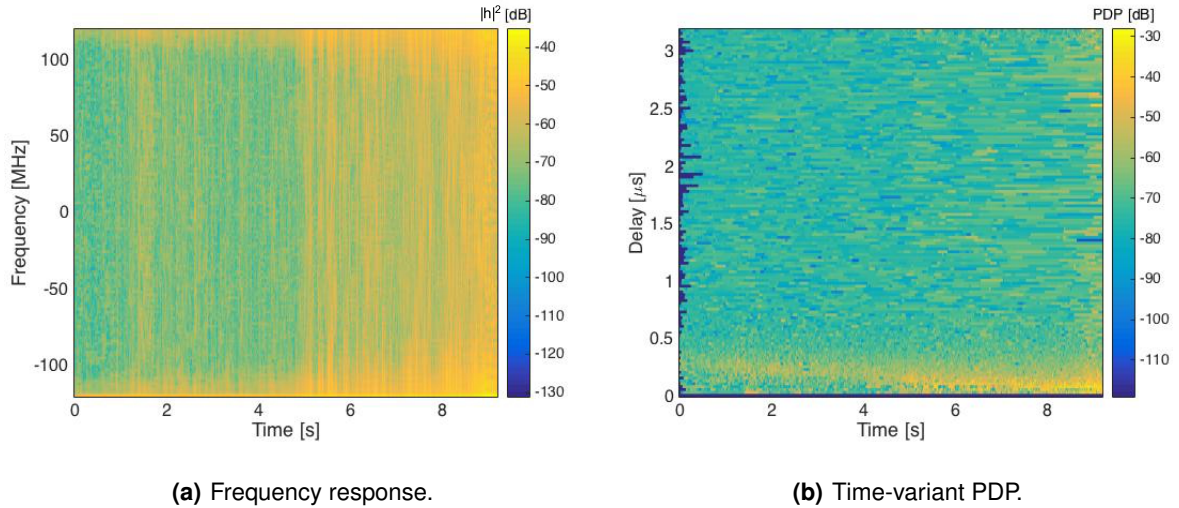


Figure 4.6: Approximated channel results for urban crossing scenario.

4.3 Analysis of Results

4.3.1 Stationary Models

This section presents the emulator results, regarding PER analysis of the received packets, after passing through the channel emulator.

In this subsection, the stationary channel models, such as pure Rayleigh, pure Rician and the standardized ETSI models, are taken into account. Each model is presented with a number of taps varying from 1, for the pure distribution models, to 4, for the more critical scenarios regarding vehicular traffic, cities and highways. In addition to that, the influence attenuation has on the results is analyzed, for a payload packet size of 500 bytes.

The performance of each channel model is assessed through the analysis of PER, by comparing the number of packets successfully received with the total number of transmitted ones.

The channel, for each scenario, packet size and attenuation, is assumed to last for approximately 10 s and the emulation takes 10 min, meaning there are 60 PER traces of the same scenario generated over the measurement time. The generation of multiple traces, during the measurement time, enables the comparison between PER results for the same scenario and to verify how stable the behaviour of the emulator is.

The rate at which packets are transmitted from Tx, through the emulator, on to Rx is 500 *packet/s*, meaning that each trace contains $N_{\text{packets}} = 5000$ PER data points, over the channel time axis, plotted and analyzed below. The average PER of all traces, during a single emulation, is also presented.

The AWGN scenario is included for comparison purposes and can be seen in Figure 4.7, where Figure 4.7(a) presents the 60 PER realizations for a single attenuation, and 4.7(b) shows the average PER for each time instant, for all attenuations. As expected, PER results are very low in this all-noise scenario and, for the lowest attenuations, the curve can not even be seen.

The situation is different for the single tap distribution scenarios. Although the channel is only represented by one tap, the high variances of PER during the emulation, for the different attenuations and speeds, are noticeable.

Firstly, PER traces for the pure Rayleigh scenarios and an attenuation of 12 dB are seen in Figures 4.8 (more attenuations in Appendix C.1). Rayleigh fading occurs due to multipath and it is applicable when there is no dominant propagation along a LOS between Tx and Rx, as seen in the probability density function in Appendix A.1.

As expected, the figures show relatively high PER results, due to multipath and the NLOS characteristic

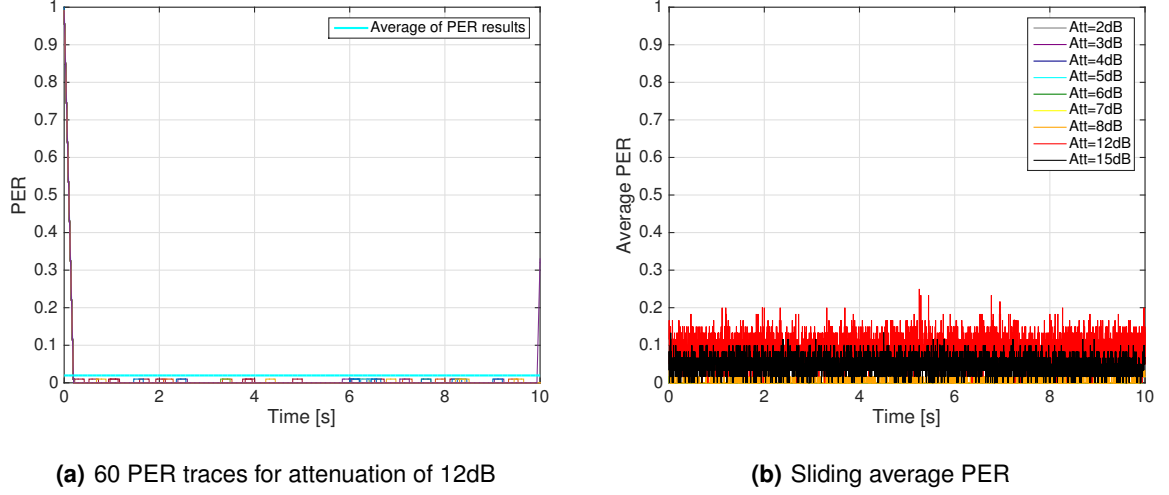


Figure 4.7: AWGN comparison scenario, for 200 bytes packet size and several attenuations.

channel. Given that there is only one contribution, from a single static tap, PER curves have a relative stable behaviour, and the increase in PER values with the increase in speed and attenuation is very evident.

Table 4.2 shows the 60 PER channel realization traces' mean value, \bar{P}_p , and standard deviation, σ_p , over the whole measurement time, for speeds of 10 km/s, 50 km/s, 100 km/s, 150 km/s, packet size of 500 bytes and attenuation of 12 dB. The mean value and standard deviation for PER, both determined using Matlab, are respectively defined in (4.1) and (4.2). Since all these parameters were determined for each trace, the presented results are the average among all of them.

$$\bar{P}_p = \frac{1}{N_{packets}} \sum_{i=1}^{N_{packets}} P_{p,i}. \quad (4.1)$$

$$\sigma_p = \sqrt{\frac{\sum_{i=1}^{N_{packets}} |P_{p,i}|^2 - \frac{|\sum_{i=1}^{N_{packets}} P_{p,i}|^2}{N_{packets}}}{N_{packets} - 1}}. \quad (4.2)$$

Table 4.2: Pure Rayleigh PER results average and standard deviation, for attenuation of 12 dB.

Speed [km/h]	\bar{P}_p	σ_p
10	0.468	0.097
50	0.557	0.074
100	0.698	0.055
150	0.848	0.039

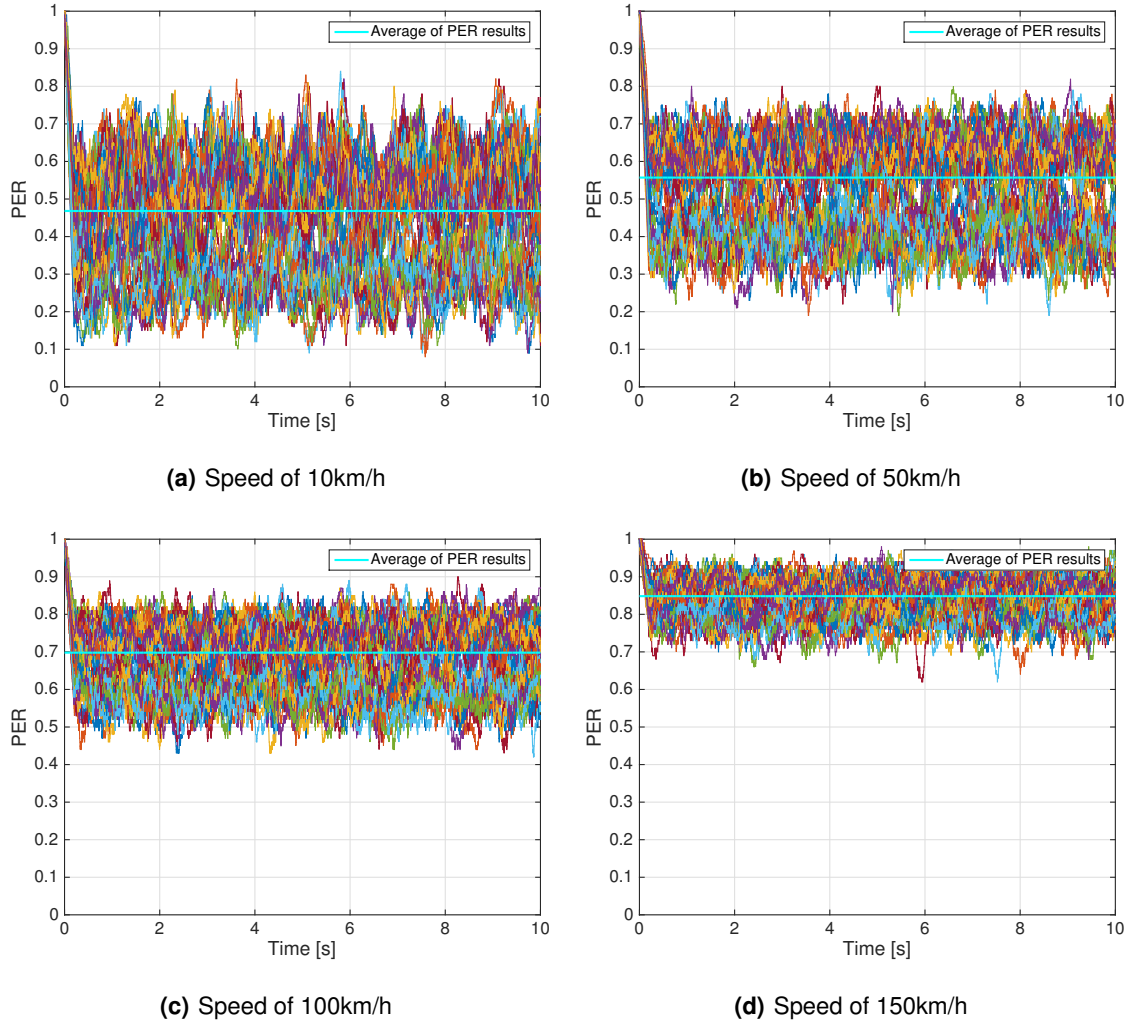


Figure 4.8: 60 PER traces for Pure Rayleigh and attenuation of 12dB, for different speeds.

The parameters for the mean values and standard deviations confirm the previous analysis, with the mean PER increasing and the standard deviation decreasing with the increase in vehicle speed. Also, when analyzing the other attenuations, it is clear that the channel's behaviour degrades with its increase, and results become closer to the mean value.

Taking the 60 PER traces already analyzed, it is possible to determine the average PER through ensemble means during a whole emulation, for each scenario. This was done by averaging all the events that are separated exactly 10 s. The ensemble average can be found in Figure 4.9. This average, over all traces, converges to the error probability of the transmission, since it takes all variations of the single PER traces into account, and gives a much more accurate presentation of the channel's behaviour when it comes to the errors in reception.

When analyzing the pure Rice scenarios, it is interesting to compare its results with Rayleigh ones. PER traces for the Rice scenarios can be seen in Figure 4.10 for an attenuation of 12 dB (more results in Appendix C.2).

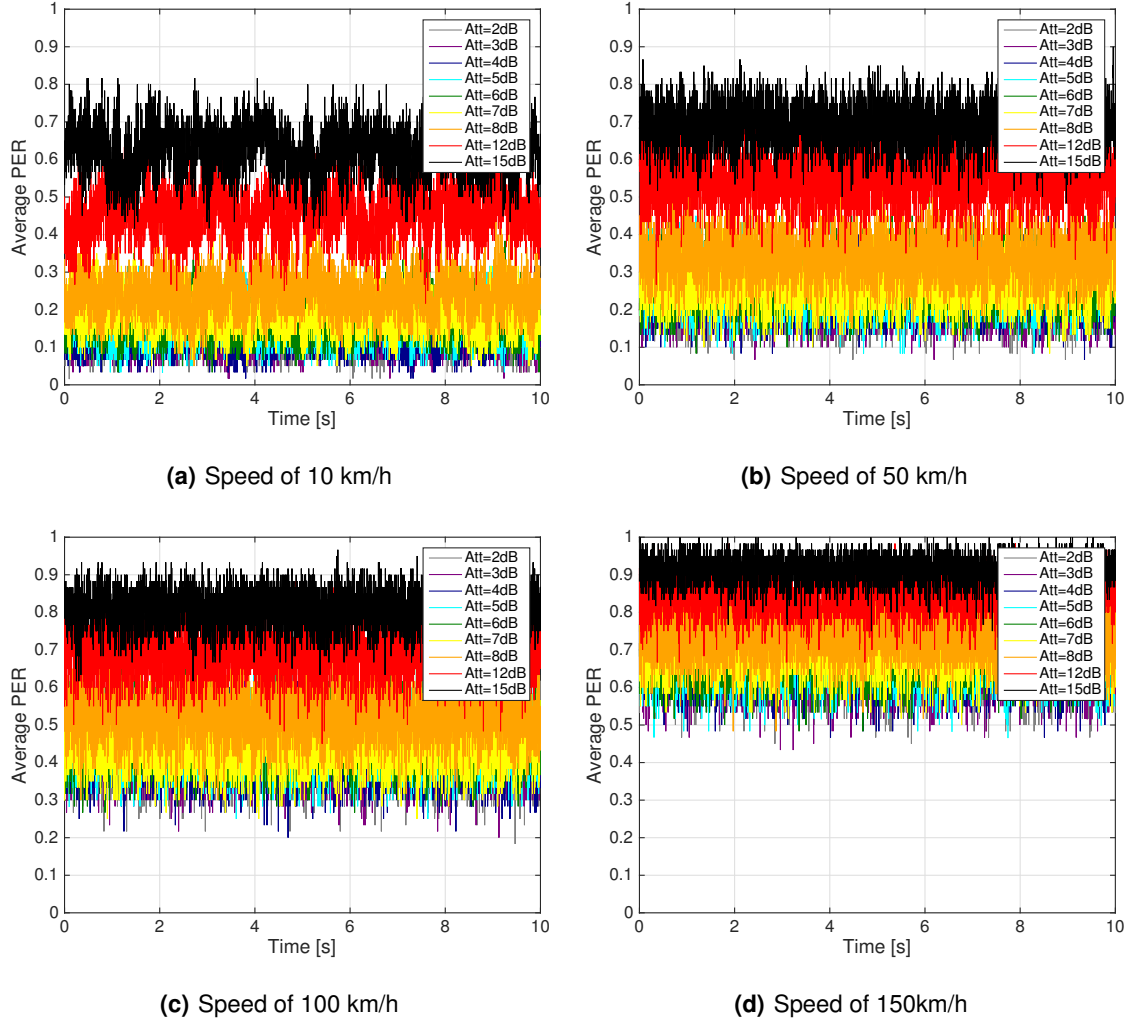


Figure 4.9: Sliding average PER results for Pure Rayleigh scenario, for 500 bytes packet size and several attenuations.

Rice fading, as seen in Appendix A.2, includes the Rayleigh fading with the introduction of a dominant and strong LOS in the environment. This way, multipath signals are weaker in comparison with the dominant one.

The analysis of the Rice scenarios is very similar to the Rayleigh one, although, overall, PER values are considerably lower than the pure Rayleigh scenario, proving the previous assumption in Subsection 3.1.1 of Rice distribution behaving better in dense traffic and low speed situations.

When analyzing the mean values and standard deviations in Table 4.3, although PER values are low for higher attenuations, this assumption is less clear for higher speeds.

The ensemble average for Rice PER results are presented in Figure 4.11, with similar conclusions as the ones drawn for the Rayleigh case.

When comparing Rice with Rayleigh results, which present lower values for PER for the several attenu-

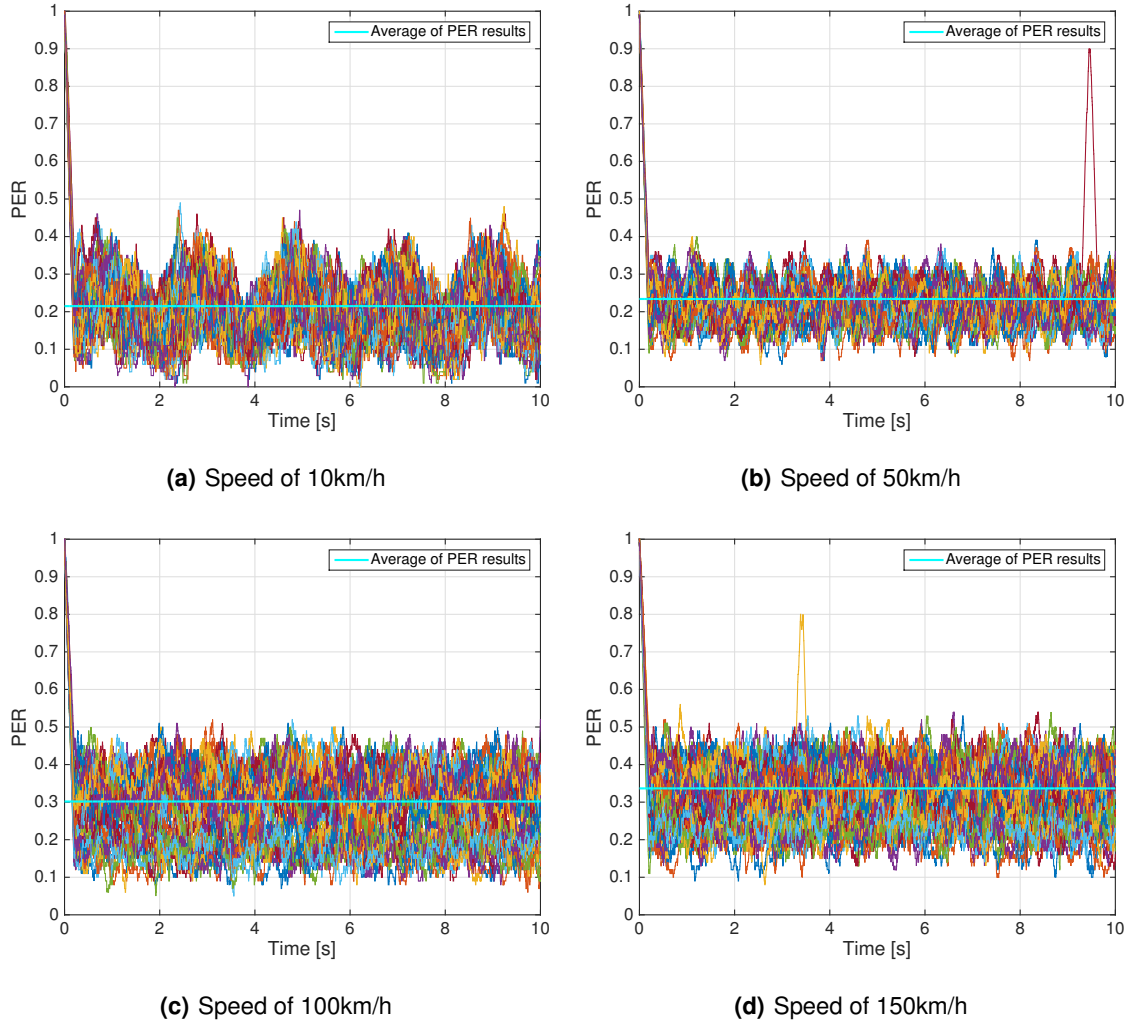


Figure 4.10: 60 PER traces for Pure Rice and attenuation of 12dB, for different speeds.

Table 4.3: Pure Rice PER results average and standard deviation, for attenuation of 12 dB.

Speed [km/h]	\bar{P}_p	σ_p
10	0.215	0.111
50	0.234	0.096
100	0.302	0.092
150	0.337	0.088

ations, it becomes apparent that the static tap is clearly better represented by the Rice distribution, due to the fact that this distribution is suited for LOS scenarios. Also, Rice traces seem more stable with increasing speed, seen by the decreasing standard deviations.

Overall, the increase in speed does not seem to significantly influence the values of PER for the Rice distribution, mainly for lower attenuations. This can be explained by the fact that the model is single-tap, which can be associated with lower traffic and having a LOS component. Given the simplicity of the

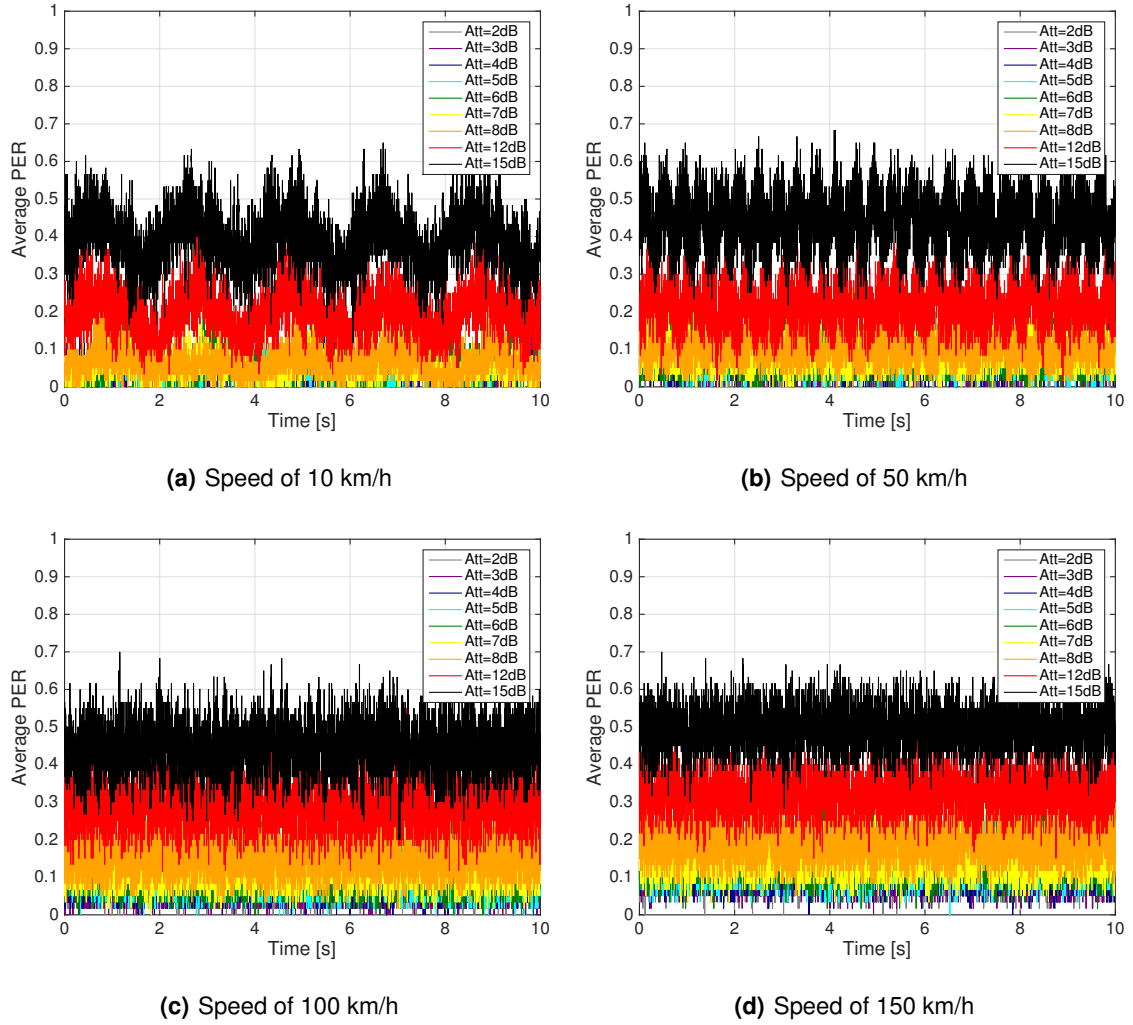


Figure 4.11: Sliding average PER results for Pure Rice scenario, for 500 bytes packet size and several attenuations.

model, Rice distribution presents better results for PER, as expected.

Analyzing the stationary ETSI models, such as the urban and highway scenarios, with and without LOS, the curves behaviour is less clear in the distinction between the different attenuations.

The urban scenarios consist of two particular situations. The first one is the urban approaching, in which two vehicles approach each other in LOS, on the same road, with nearby buildings. The second one is the urban crossing, in which two vehicles approach a blind intersection with other traffic and nearby objects, such as buildings, present on all corners. PER traces for both scenarios can be seen in Figure 4.12 for an attenuation of 12 dB (more attenuations in Appendixes C.3 and C.4).

The urban approaching scenario is characterized by an LOS component, since the vehicles see each other on the same road. The signal reaching the receiving vehicle is characterized by a direct signal and other multipath contributions from nearby buildings and other objects.

The urban crossing scenario is characterized by an NLOS component, because the vehicles move in

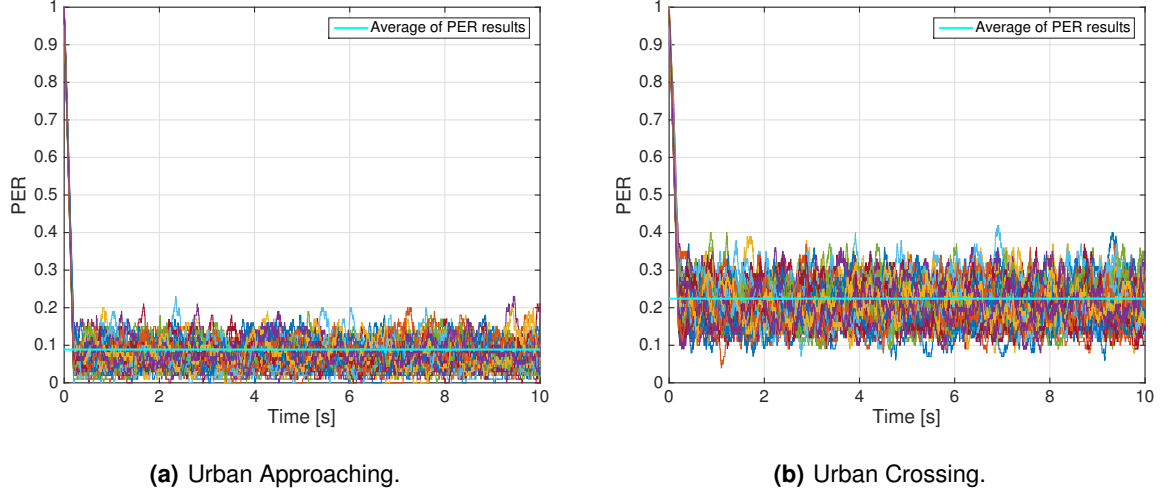


Figure 4.12: 60 PER traces for Urban Intersection and attenuation of 12dB.

different lanes and do not see each other until they cross the intersection. The received signal is mainly characterized by the MPCs from nearby objects, such as buildings.

The analysis of the several traces in the figures already referenced, for several attenuations, indicates that small-scale variations are seen strongest in the urban crossing scenario, since it has stronger MPCs, which in turn result in more interference in the signal at the receiver. This is also caused by the long correlation times due to low Doppler speeds.

Indeed, when consulting the parameters in Table 4.4 and in the appendixes, it is evident that the measured PER results present a worse behaviour for the urban crossing scenario, since the mean value of the signals is higher for every attenuation. The average standard deviation also supports it and, although, the traces are close to the mean value, both urban scenarios present a highly variable behaviour due to the strong multipath contributions to the signal, resulting in high interference at the receiver.

Table 4.4: PER results average and standard deviation for urban scenarios and attenuation of 12 dB.

Scenario	\bar{P}_p	σ_p
Urban Approaching	0.089	0.108
Urban Crossing	0.224	0.097

The ensemble average PER plots for the urban scenarios are presented in Figure 4.13, showing a variant behaviour for the higher attenuations, due to more uncertainty in receiving the transmitted packets. PER values are higher for the urban crossing, as already mentioned, due to MPCs being stronger for this scenario.

Finally, PER traces for the highway scenario, divided into LOS, where vehicles see each other, and

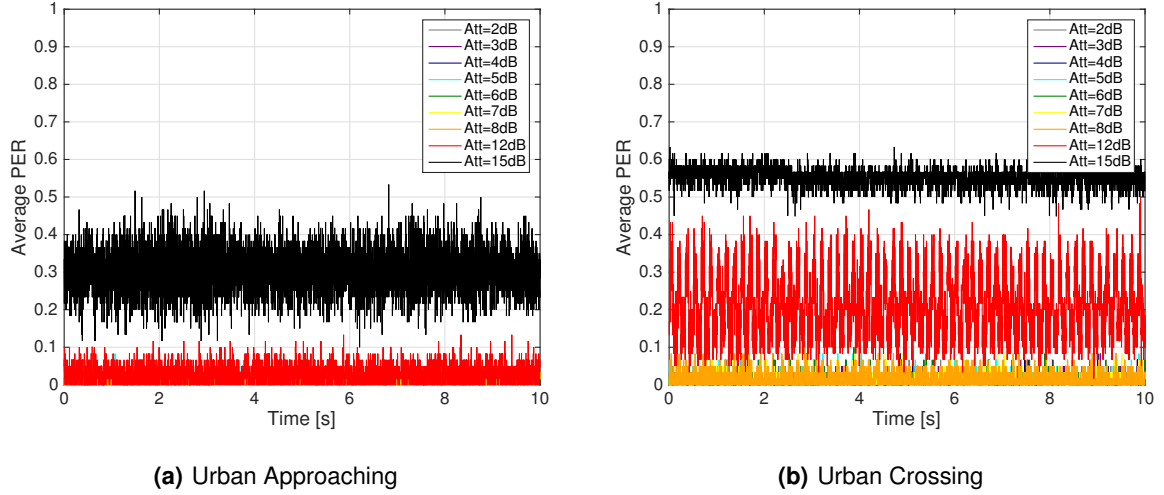


Figure 4.13: Sliding average PER results for ETSI urban scenarios, for 500 bytes packet size and several attenuations.

NLOS, where vehicles are blocked from each other with other vehicles, are presented in Figure 4.14 for an attenuation of 12 dB. It is possible to identify higher PER for the NLOS scenario, as expected, given the vehicles are blocked from each other and the multipath contributions are relatively weak, due to the scenario being open and, therefore, the number of obstacles being low.

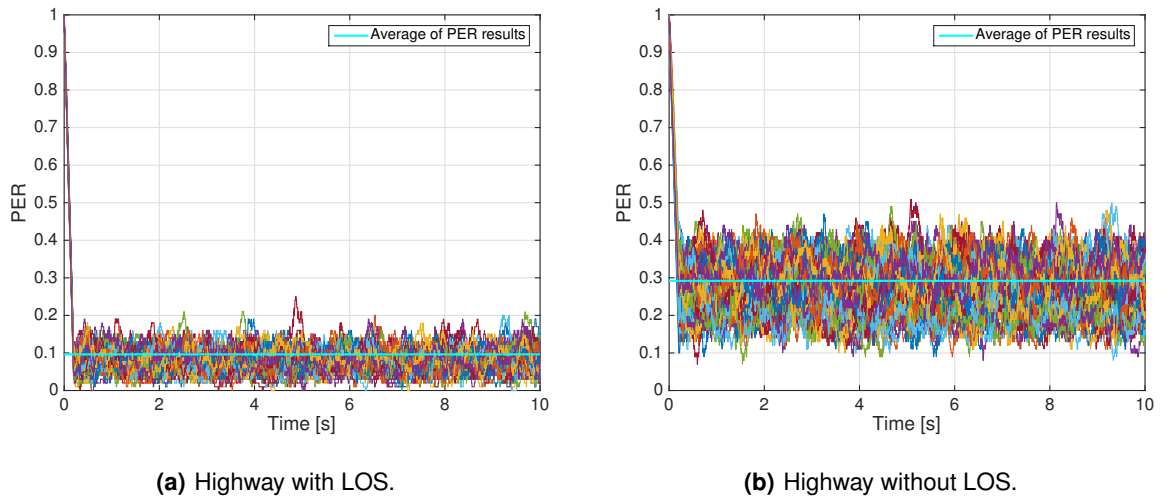


Figure 4.14: 60 PER traces for Highway and attenuation of 12 dB.

The parameters for both the scenarios presented in Table 4.5 show that indeed, for the majority of time, the NLOS scenario has a very bad performance, losing practically all the packets during the transmission, seen by the very high mean value and very low standard deviation. Although the vehicles move at high speeds, PER results for the LOS scenario are much better given there is a strong direct contribution, seen by the considerably lower mean values.

When analyzing the scenarios for different attenuations, in Appendixes C.5 and C.6, results do not vary

Table 4.5: Highway PER results average and standard deviation, for attenuation of 12 dB.

Scenario	\bar{P}_p	σ_p
Highway LOS	0.096	0.107
Highway NLOS	0.292	0.092

very much and the average standard deviations are relatively low, given the considerably lower presence of nearby scatterers comparing with the urban scenario.

Overall, and when analyzing the ensemble PER average in Figure 4.15, the random high variances are cut out and the results show PER values that are higher for the NLOS component.

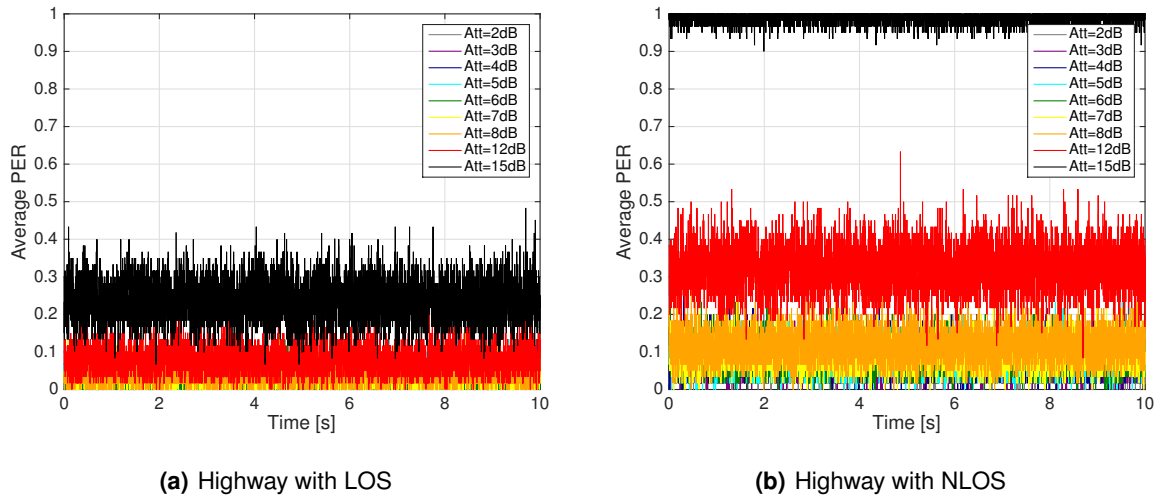


Figure 4.15: Sliding average PER results for ETSI highway scenarios, for 500 bytes packet size and several attenuations.

Comparing the highway scenarios with the urban ones, it is clear that the MPCs are less strong, seen by lower variations in PER traces, and the Doppler effect is higher, due to the higher speeds. Also, the distinction between LOS and NLOS is very evident, both in the urban and highway scenarios. It is clear that the highway NLOS model performs worse than the urban one, since the high Doppler effect highly degrades the signal. However, the highway LOS performs better than the urban NLOS, since the signal reaches the receiver with less interference.

This way, the LOS component is indeed a very influential characteristic in the behaviour and performance of the channel and the absence of a direct line at the receiver can significantly degrade the signal and result in higher PER due to the higher loss of the packets during transmission. Adding to that problem, the high vehicle speeds during the transmission can cause the majority of the packets to be lost during the measurement time.

Before concluding, a comparison between the high speed pure single tap distributions scenarios, such as Rice and Rayleigh for 100 km/s and 150 km/s, and the highway ones can be drawn, due to the similarity

between the vehicle speed in both scenarios and the lack of many surrounding objects (given by the single pure tap distributions). Overall, PER results are similar among these scenarios, mainly for the higher attenuations. However, single-tap distributions are optimistic and present unrealistic results.

It is clear that for the stationary models, PER curves vary around a constant value. The variations are seen strongest in the urban NLOS scenario, which has stronger MPCs and long correlation times due to low Doppler speeds. The ensemble average PER is able to show the more pronounced short-term variations that are neglected in PER traces. The results also show the fading impact in the variation of PER and, therefore, can not be neglected a priori.

Concluding, when analyzing the urban and highway scenarios, these results show that the non-stationarity of vehicular channels, due to moving obstacles, Tx and Rx and high speeds, is not completely well represented by the stationary models, mainly because it lacks accuracy in vehicular specific situations, like a vehicle obstacle disappearing from the direct line between Tx and Rx, low and high speed nodes and obstacles, among others. These scenarios result in unrealistic PER values and behaviour that do not evidence the several complexities and variance of the channels.

4.3.2 Non-stationary Models

The non-stationary scenario channel duration is 10 s and the emulation for each scenario takes 10 min, which means each channel is repeated 60 times during the emulation, as with the stationary scenarios. This enables averaging PER curves, over the measurement time.

The 60 PER traces were determined for several attenuations and 3 different payload packet sizes: 200, 500 and 1000 bytes. The attenuations presented in this subsection are 2 dB and 12 dB. The remaining attenuation plots are presented in Appendix D.

Taking the channel behaviour analyzed in Section 4.2, it is expected that the PER curve approximated behaviour will be similar to the one described in Figure 4.16. This approximated curve presents a not-realistic linear behaviour and it is used as comparison with the real PER results.

The figure is divided in 4 different regions, which characterize the overall packet transmission behaviour of the vehicular channel during the 10 s. In Region 0, from $t = 0s$ until t_1 , PER between the transmitter and receiver vehicles is practically 1, meaning the majority of the transmitted packets is lost. This happens because the two vehicles are far away and do not see each other, meaning the transmission is made by NLOS components from the nearby objects. In Region 1, from t_1 to t_2 , the vehicles get closer until they reach the intersection, which happens in Region 2, from t_2 to t_3 . The vehicles transition to LOS with each other, explaining why the rating at which PER decreases is higher. The peak represents the moment they are the closest. In region 3, from t_3 to $t_{max} = 10s$, PER increases as the vehicles move away from each other.

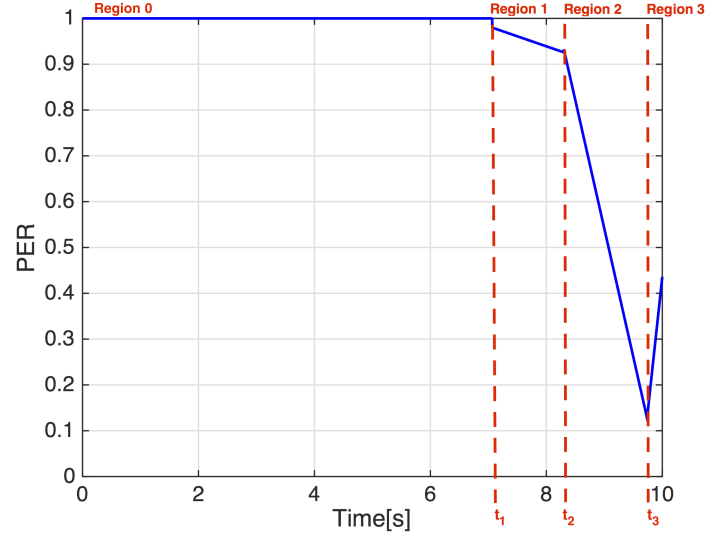


Figure 4.16: General linear regression for the expected PER curve behaviour. The different regions are characterized by a different linear equation.

The general regression line equation for each region is given below.

$$P_p = m_{Region\ i} [s^{-1}] (t_{[s]} - t_{0,i} [s]) + P_{0,Region\ i} \quad (4.3)$$

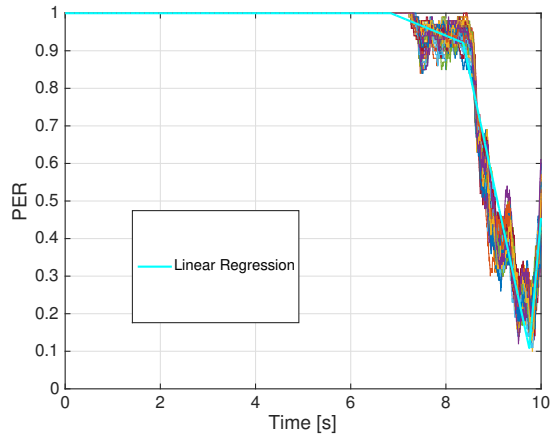
where:

- P_p : PER linear values,
- t : time axis values,
- $t_{0,i}$: time instant the Region i starts,
- $m_{Region\ i}$: linear regression slope for region i ,
- $P_{0,Region\ i}$: linear regression offset, in time instant t_0 , for region i .

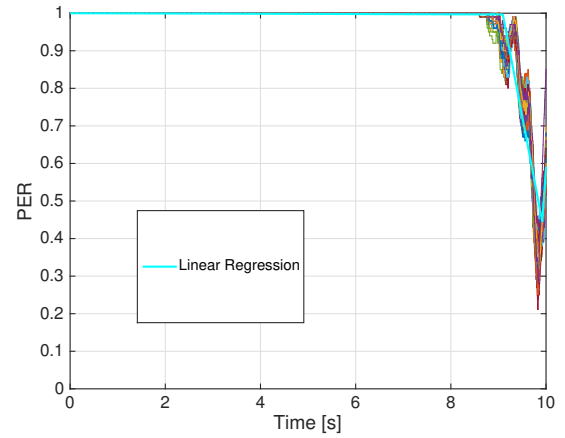
When analyzing PER measurement results, Figures 4.17, 4.18 and 4.19 show that PER traces for the 60 channel realizations spread out over the measurement time show a relatively similar behaviour.

As expected, PER values for most of the duration of the channel are close to 1, meaning that the majority of the transmitted packets are lost during the first 7 s, moment in which PER starts decreasing slightly. During this time, while the vehicles do not reach the intersection, the transmission of packets is made mainly by MPCs from nearby obstacles. At this point, the decrease in PER indicates the appearance of a LOS component, that grows stronger until it reaches its peak at around 9.5 s, when the vehicles are the closest to each other. From this point on, PER increases due to the vehicles driving away from each other along the lane.

The behaviour of the traces is very similar during the emulation time, which demonstrates the stability of the emulator over the 10 min measurement. This way, the results show that the emulator enables a high repeatability of tests.

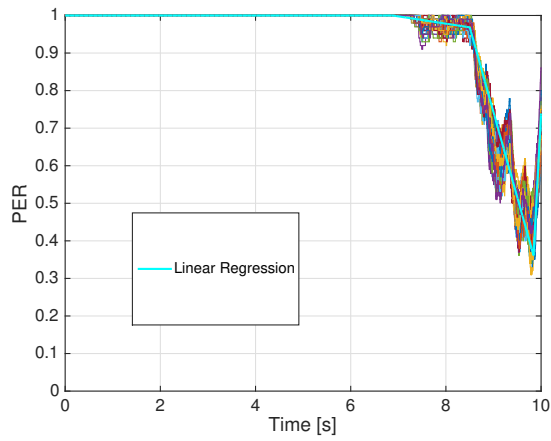


(a) Attenuation of 2 dB

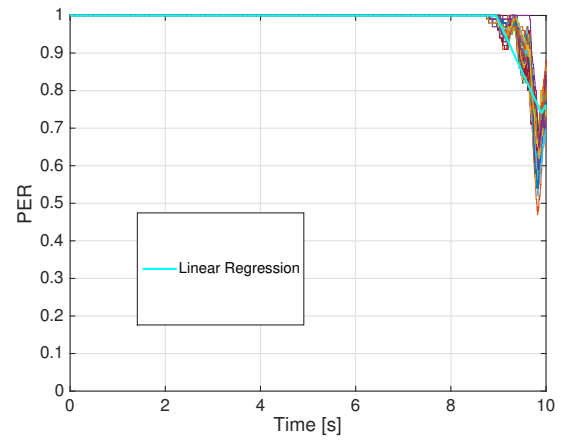


(b) Attenuation of 12 dB

Figure 4.17: 60 PER traces for packet size of 200 bytes.

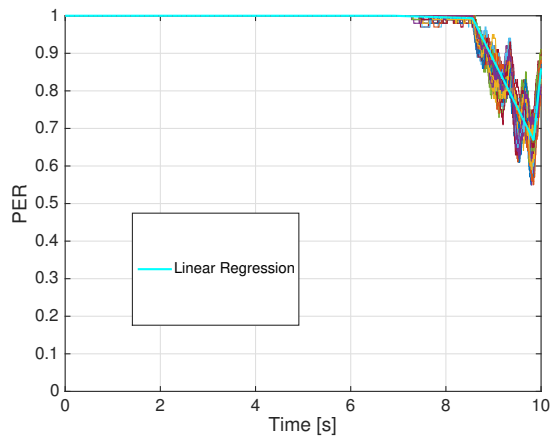


(a) Attenuation of 2 dB

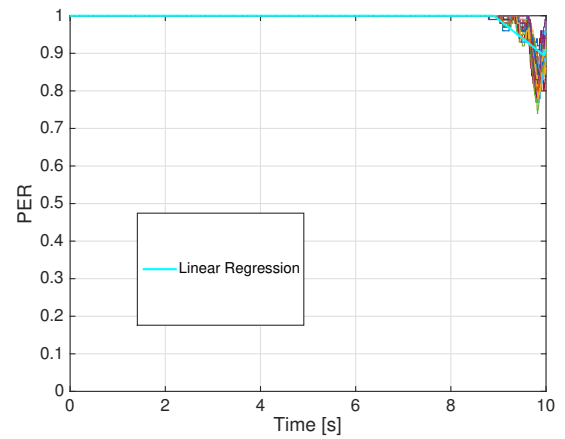


(b) Attenuation of 12 dB

Figure 4.18: 60 PER traces for packet size of 500 bytes.



(a) Attenuation of 2 dB



(b) Attenuation of 12 dB

Figure 4.19: 60 PER traces for packet size of 1000 bytes.

The linear regression parameters for each packet size and attenuation are shown in Tables 4.6, 4.7 and 4.8, where the parameter $t_{max,i}$ is the time instant a certain Region i ends. When analyzing their values, it is evident that the increase in attenuation results in the "disappearance" of Region 1, meaning that for sufficiently high attenuations, packets are almost completely lost during the transmission until the vehicles are close enough. The increase in packet loss with the increase in packet size shows that the disappearance of Region 1 happens sooner, for lower attenuations.

Table 4.6: Linear regression equation parameters for 200 bytes and the 3 main regions.

		Attenuation [dB]	
		2	12
Region 1	t_0 [s]	6.840	-
	t_{max} [s]	8.360	-
	m [%/s]	-0.048	-
	P_0 [%]	1.000	-
Region 2	t_0 [s]	8.360	8.690
	t_{max} [s]	9.752	9.960
	m [%/s]	-0.554	-0.448
	P_0 [%]	0.883	1.000
Region 3	t_0 [s]	9.752	9.960
	t_{max} [s]	10.000	10.000
	m [%/s]	1.173	1.476
	P_0 [%]	0.132	0.523

Table 4.7: Linear regression equation parameters for 500 bytes and the 3 main regions.

		Attenuation [dB]	
		2	12
Region 1	t_0 [s]	7.020	-
	t_{max} [s]	8.500	-
	m [%/s]	-0.017	-
	P_0 [%]	1.000	-
Region 2	t_0 [s]	8.500	8.950
	t_{max} [s]	9.840	9.900
	m [%/s]	-0.381	-0.373
	P_0 [%]	0.975	0.999
Region 3	t_0 [s]	9.840	9.900
	t_{max} [s]	10.000	10.000
	m [%/s]	1.684	1.072
	P_0 [%]	0.384	0.730

Table 4.9 shows the correlation coefficient between the linear regression equations and the 60 PER channel realization traces, $\rho_{P_p, P_p^{reg}}$, the traces mean value, \bar{P}_p , and standard deviation, σ_p , over the

Table 4.8: Linear regression equation parameters for 1000 bytes and the 3 main regions.

		Attenuation [dB]	
		2	12
Region 1	t_0 [s]	7.000	-
	t_{\max} [s]	8.560	-
	m [%/s]	-0.003	-
	P_0 [%]	1.000	-
Region 2	t_0 [s]	8.560	8.900
	t_{\max} [s]	9.840	9.960
	m [%/s]	-0.226	-0.146
	P_0 [%]	0.994	1.000
Region 3	t_0 [s]	9.840	9.960
	t_{\max} [s]	10.000	10.000
	m [%/s]	1.219	0.492
	P_0 [%]	0.634	0.876

whole measurement time, for the attenuations of 2 dB and 12 dB and all packet sizes. The correlation coefficient between the measured traces and the linear regression, determined using Matlab [49], is defined in (4.4). The parameters were averaged between all traces.

$$\rho_{P_p, P_p^{reg}} = \frac{1}{N_{\text{packets}} - 1} \sum_{i=1}^{N_{\text{packets}}} \left(\frac{P_{p,i} - \bar{P}_p}{\sigma_p} \right) \left(\frac{P_{p,i}^{reg} - \bar{P}_p^{reg}}{\sigma_{P_p^{reg}}} \right). \quad (4.4)$$

where:

- $\rho_{P_p, P_p^{reg}}$: correlation coefficient between channel realization traces,
- $P_{p,i}$: PER value for packet i ,
- $P_{p,i}^{reg}$: linear regression PER value for packet i ,
- \bar{P}_p^{reg} : linear regression mean value for PER,
- $\sigma_{P_p^{reg}}$: linear regression standard deviation for PER.

The correlation coefficients for the several scenarios are very high, meaning the behaviour of the channel is very close to the expected. When analyzing the mean value of PER averaged over all the 60 traces and the standard deviation, the higher the attenuation and the packet size, the higher the mean PER is, almost close to 1, and the closest the trace is to a straight line equal to the mean.

Taking the PER traces already analyzed, it is possible to determine the average PER through ensemble means. This is shown in Figures 4.20, 4.21 and 4.22, which were calculated among all the 60 channel realizations spread along the whole emulation, for several packet sizes and attenuations. This was done by averaging all the events that are separated exactly 10 s.

It is clear from analyzing the figures that the average behaviour shows more detail than the independent curves, which, for example, can not detect the short dip at approximately 8.5 s. This way, for non-

Table 4.9: Correlation coefficient between PER results and regression, PER results mean and standard deviation.

Packet size [bytes]	Attenuation [dB]	$\rho_{P_p, P_p^{reg}}$	\bar{P}_p	σ_p
200	2	0.989	0.908	0.212
	12	0.969	0.973	0.099
500	2	0.989	0.943	0.139
	12	0.957	0.986	0.054
1000	2	0.983	0.973	0.071
	12	0.949	0.995	0.022

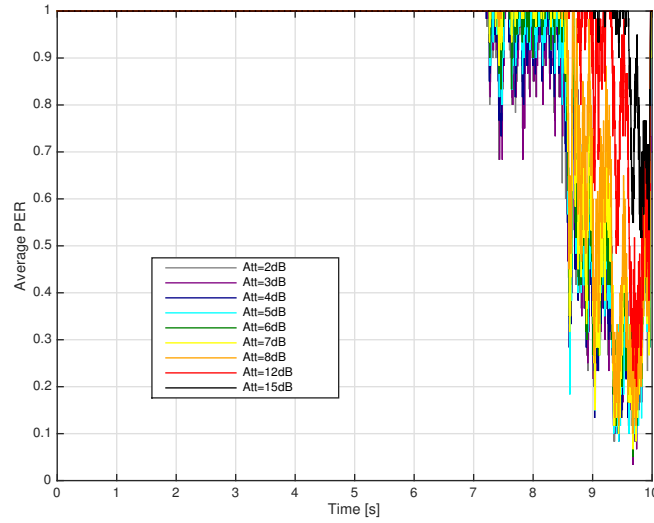


Figure 4.20: Average PER over emulation for 200 bytes packet size. The label shows the attenuations.

stationary scenarios, this estimate should be used to detect fast changes. This average PER shows an approximation to the error probability.

Along the emulation measurement time, the non-stationary channel dependence with the LOS is highly evident. Analyzing the appearance of the LOS component in Figure 4.5, at approximately 8 s, PER curves evidence this by varying from close to 1 down strongly, depending on packet length.

When varying packet size, it becomes clear that packets are better received when divided into 200 bytes, and the channel behaves the worst for the 1000 bytes packets. The most common packet size is 500 bytes. Regardless of the packet size, the qualitative behaviour of the channel is very similar.

Comparing the non-stationary results with the analyzed stationary crossing ones, it is evident that the latter presents a optimistic realization of a vehicular channel, given the low value for PER opposed to the almost 1 case of the non-stationary one. Also, they do not show the variations that occur in this specific situation, such as appearance or disappearance of LOS when the vehicles reach the intersection and start seeing each other, and the slow increase of PER as vehicles move away.

The non-stationary data-based vehicular channels present pessimistic but realistic results, since the

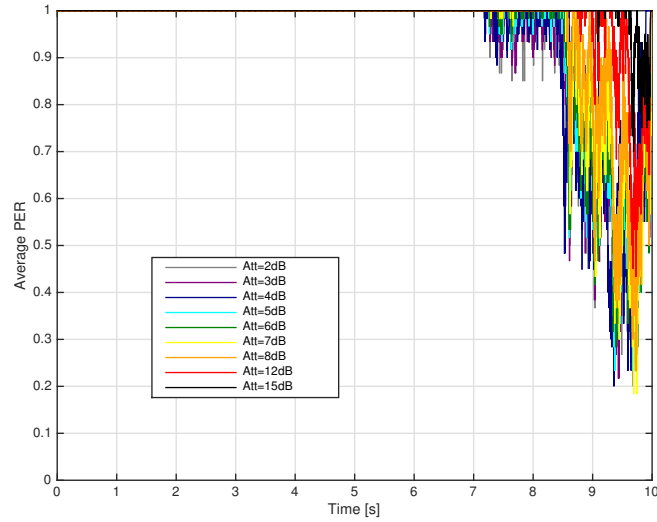


Figure 4.21: Average PER over emulation for 500 bytes packet size. The label shows the attenuations.

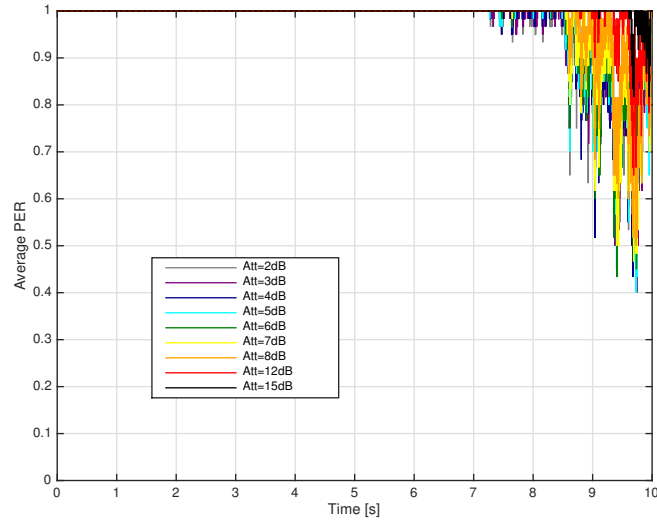


Figure 4.22: Average PER over emulation for 1000 bytes packet size. The label shows the attenuations.

majority of the packets are lost. The non-stationarities of the channel are well represented and the model of the channel is a very close approximation to the real measured data. The distinctions between LOS and NLOS are also very clear.

The results obtained for the non-stationary urban intersection are the expected ones and are in line with the applications in safety vehicular contexts. Since communication happens in an ad-hoc manner, vehicles exchange information about their surroundings very fast, in a range of a few meters from one another. When the vehicles are far away, not seeing each other, the almost-1 PER results make sense, since communication for these applications only happens when the vehicles are close. This way, information is spread continuously from vehicle to vehicle.

Concluding, the emulator presented in this thesis is stable and capable of accurately simulating the behaviour of a real-life vehicular channel, and it is reliable to be used in testing the performance of vehicle modems, at low-complexity and cost.

5

Conclusions

This chapter finalises this report and presents the conclusions.

The main purpose of this thesis was to develop a low-complexity, flexible, and reprogrammable method to emulate a vehicular channel, characterized by a highly non-stationary behaviour. This way, several different modems installed in vehicles, for applications like ITS or autonomous driving, can be tested accurately in a low-cost manner, without the need for real-life measurements. These services provide safety, traffic efficiency or entertainment services on board.

In the first chapter, one addressed how the growing mobility of people and vehicles on the road, the increase in the number of terminals connected to the network, and the imminent launch of 5G, shows the relevancy of this problem and how this approach may benefit future work in this area. The motivations for this work were illustrated, as well as an introductory description of the vehicular channel and relevant scenarios.

Chapter 2 introduced the Wi-Fi standard and the amendment IEEE 802.11.p, which enables communications in vehicular ad-hoc networks. The typical scenarios for this type of communication were also described, as well as how they affect the channel. The vehicular channel was characterized and a TDL representation was offered, for discrete time. The chapter finished with the definition of a channel emulator and the state of the art, which presented some different approaches on obtaining vehicular channel models and how to emulate them.

Chapter 3 introduced the TDL channel models for the emulator, both stationary and non-stationary. The stationary models included scenarios described by up to 4 taps. Single-tap models were generated using the Rayleigh distribution, for several different vehicle speeds, characterizing NLOS scenarios. For better results, tap components were generated using Rice distribution, which characterizes LOS scenarios. Given that the Rice distribution degenerates into a Rayleigh one, tap components were determined using Rayleigh results. The 4-tap channel models were the standardized ETSI models, rural, urban and highway, which were characterized by a first static tap, and each component was generated using the Rayleigh distribution. These stationary models were analyzed to test their capability of reproducing the real-life vehicular behaviour and to compare it with the non-stationary one.

For the non-stationary models, the used method to obtain the taps was based on the optimization algorithm, Lasso, adapted for complex channel components, c-Lasso. This powerful tool allowed to obtain an approximated channel representation, in matrix form, by minimizing the MSE between the approximation and the original measured channel data, limiting the number of taps that represent the channel. The constraint on the number of taps, in each time slot, was obtained by approximating the I_0 norm, used to limit the number of non-zero components, to the closest norm, I_1 . This way, the channel was represented by a low-complexity TDL matrix, with a fixed number of taps, and the chosen delay taps were highly correlated in time.

The stationary and non-stationary channel model components were then fed as input to the emulator. The basis version of the emulator was presented, enabling only the simulation of stationary models, one scenario at a time, and the same fixed delays for the whole measurement time. The added functionalities

to the emulator were described, allowing to emulate consecutively and automatically several scenarios, and to update the delays with the magnitude and phase. The scenarios' data consecutive reception by the emulator was done by a UDP connection with a Python script, which enabled the transmission of the magnitude, phase and delay components in sub-array form and its assemble on the main code, running the LabVIEW emulator. The arrays were then passed onto the FPGA, which updated the magnitude and phase components at a certain rate. The delay components were updated once every 10 cycles the magnitude and phase ones were updated. This transmission, reception, passing and writing of the arrays procedure allowed a scenario to be emulated the moment the main code started receiving its data.

The chapter finished with the model assessment of the emulator to test its performance. Several simple tests were performed in the emulator to test if data were being correctly received through the UDP link, assembled in the main code and finally written, by comparing data in the several stages with those that were transmitted. The implementation was finally validated by testing the measured frequency response of a fast-varying channel model, with fixed delays, with a sudden jump in behaviour to a slow-varying one, with different delays. Since the behaviour was the expected one, it was possible to proceed for any scenario to be emulated and analyzed.

Chapter 4 started by providing a description of the scenario, in Lund, Sweden, for the non-stationary channel model, and described the matrix in which data were organized. The scenario represented the urban intersection scenario, combining the moment the vehicles approach the intersection without seeing each other, and when the vehicles reach it and are in LOS with each other before moving away.

This is evidenced in the next section, in which the PDP of the data measurements was analyzed and compared with the approximated one, evidencing the transition from a NLOS scenario to a LOS one at approximately 8 s. The high correlation between the real data and the approximation demonstrated the efficiency of the algorithm in choosing the best taps, so that it was perfectly capable of representing the original channel.

When analyzing the measurement results of the emulator, the number of packets successfully received was compared with the total number of transmitted ones, in order to determine the PER of the channel under different scenarios and conditions like packet size and attenuation. Each scenario was emulated for 10 min. Since each channel duration was assumed to be 10 s, the emulator generated 60 realization traces for each one, over the measurement time. The emulator's stability and capacity to reprogramme easily for different scenarios was confirmed by the high similarity shown among traces for the same scenario, during the measurement time.

PER results for the single-tap stationary scenarios, Rayleigh and Rice, showed that the increase in attenuation leads to a degradation in the channel behaviour, meaning packets are more easily lost. The increase in vehicle speed also deteriorated the overall performance of the emulator, which was more evident in Rayleigh scenarios. The Rayleigh distribution characterizes a NLOS transmission, while the

Rice distribution targets LOS. Rice results, mainly for the lower speeds and lower attenuations, were better than the Rayleigh ones, since the LOS component was better represented.

ETSI channel models contained more information about the channel since they were characterized by more taps. The urban situation considered two characteristic scenarios, being the urban approaching scenario, in which vehicles in LOS approach each other on the road, and the urban crossing one, characterized by the two vehicles reaching an intersection without seeing each other.

The urban approaching scenario presented better results, regardless of attenuation, explained by the fact communication happens in LOS, for the whole channel duration. The results proved this behaviour, since they showed higher PER for the urban crossing scenario when comparing with the urban approaching case, in which the signal in the receiver suffered considerably less losses. The ensemble PER averages for the several attenuations showed that the average behaviour when changing the attenuation was consistent with what was expected.

The overall variances among the different attenuations showed the high multipath contributions characteristic to both the urban scenarios, specially for the urban crossing case, which has worse behaviour given the NLOS nature of the received signals.

The highway scenario was characterized by a LOS component, when the vehicles see each other, or a NLOS component, when the vehicles direct line is blocked by nearby vehicles. Although both scenarios suffer from high Doppler spreads, the LOS one behaved a lot better than the NLOS case because the absence of a direct line significantly degrades transmission and results in extremely high losses.

For the non-stationary scenario, PER results showed that LOS is a highly influential parameter, since the channel loses most of the transmitted packets until the vehicles are in direct line with each other. At this point, the packet loss started decreasing until it reached its minimum, when the vehicles are the closest, before they move away, increasing PER values again until they are far away enough the transmission is practically impossible. Packet transmission is made essentially when the vehicles are close enough to each other so that the packets are not lost due to nearby objects interference.

The attenuation, which can be compared with the amount of nearby obstacles or their density, strongly influences PER results, as already analyzed for the stationary urban scenarios. For sufficiently high attenuation, the correct reception does not happen until the vehicles are close enough to each other, even if the signal is an LOS one.

The payload packet size also was a very strong influence on the correct reception of packets, with 200 bytes being too optimistic and 1000 bytes too pessimistic. For a 200 bytes packet size, a larger amount of packets was received during the measurement time, with reception being possible much sooner than the 1000 bytes one, in which, for higher attenuations, reception was practically negligible. The best and most realistic packet size to consider is indeed 500 bytes, situation in which the curves behaviour was

closer to reality.

When analyzing the obtained results for both stationary and non-stationary channel models, it was clear that the presence of an LOS component made a huge difference in data transmission, and its absence may result in a near impossibility to establish communication between nodes.

Given that communication in a vehicular context, being safety, efficiency or entertainment related, is performed in an ad-hoc manner between vehicle nodes, the measured results were consistent with the expectation due to reception only happening when the vehicles are close enough. Indeed, in this context, vehicles communicate with each other and exchange information about their surroundings with other nearby vehicles, enabling the information to be exchanged gradually as vehicles approach each other.

The high flexibility in changing the parameters, attenuation, payload packet sizes or even scenarios, enables high repeatability and flexibility when performing an emulation. Also, the low-cost and complexity architecture of the emulator is very appealing when performing measurements, specially in commercial contexts.

The size of the stationary regions also offered a good trade-off, since it is small enough so that the emulator performs realistically, and high enough so that the emulator is not very complex.

One of the problems presented by this analysis was the need of using real life measurements to test its performance. This problem can be solved by the use of standardized non-stationary models in literature. Since the emulator performed in a stable and expected way, the emulation of non-stationary scenarios by using pre-defined channel models was possible, given that delay values were also provided as input.

When performing a scenario analysis with this emulator, the higher the duration, the more it will take for the emulation to be completed, since each channel is measured more than once in a single emulation. This problem can be solved by dividing the channel into smaller chunks, emulating each one separately, or repeating each channel less times, since it has already been proven the emulator is stable.

For future work, several vehicle modems can be tested, taking the NI USRP as comparison, in order to provide an in depth analysis not only of the emulator, but also of modems performance. It is also important to make an extensive analysis of the emulator when taking no real-life measurements based channel models, and compare the results with the ones considered in this thesis.

With the increasing evolution of the mobile networks and mobility of terminals connected to the network, these results can be used as a basis for future 5G studies, and used as comparison. The growing industry of autonomous cars may benefit from this study in order to test future modems installed in vehicles.



Distributions

Description of the Rayleigh and Rice distributions, relevant for the stationary channel models.

A.1 Rayleigh

Rayleigh distribution is a continuous probability distribution for positive valued random variables. The probability density function is given below, as a function of time.

$$f_{Ray}(x, \sigma_{Ray}) = \frac{x}{\sigma^2} e^{-x^2/(2\sigma_{Ray}^2)}. \quad (\text{A.1})$$

where:

- f_{Ray} : Rayleigh probability density function,
- σ_{Ray} : scale parameter of Rayleigh distribution,
- $x \geq 0$: time.

In mobile radio channels, Rayleigh distribution is commonly used to describe the received envelope of an individual multipath. The component σ is defined as the RMS value of the received voltage signal before envelope detection and σ^2 is the time average of the received signal before envelope detection [26].

Rayleigh fading is the specialized model for stochastic fading when there is no LOS signal, and the amplitude gain is characterized by a Rayleigh distribution [50].

A.2 Rice

Rice distribution is the probability distribution of the magnitude of a circular bivariate normal random variable with potentially zero-mean. The probability density function is given below.

$$f_{Rice}(x|\nu, \sigma_{Rice}) = \frac{x}{\sigma_{Rice}^2} e^{-\frac{(x^2 + \nu^2)}{2\sigma_{Rice}^2}} I_0\left(\frac{x\nu}{\sigma_{Rice}^2}\right), \quad (\text{A.2})$$

where:

- f_{Rice} : Rice probability density function,
- σ_{Rice} : scale parameter of Rice distribution,
- $\nu \geq 0$: noncentrality parameter,
- I_0 : modified Bessel function of the first kind with order zero.

The parameter ν is often interpreted in mobile radio channels as the peak amplitude of the dominant signal [26].

Rician fading is a stochastic model for radio propagation in which the signal arrives at the receiver by several different paths, hence exhibiting multipath interference. It occurs when one of the paths, typically a LOS signal, is much stronger than the others. In Rician fading, the amplitude gain is characterized by a Rician distribution [51].



NI USRP RIO

Description of the NI URSP platform in which the emulator is implemented.

NI USRP are computer-hosted RF transceivers used for development and exploration of SDRs. NI USRP transceivers can transmit and receive RF signals in several bands. Paired with NI LabVIEW software, NI USRP transceivers provide access to real-world signals, offering flexibility, functionality and affordability to deliver an ideal SDR prototyping platform [52].

LabVIEW, running on a Gigabit Ethernet-connected host computer, provides the signal processing engine for the modulation and demodulation of signals streaming to and from NI USRP hardware.

The NI USRP hardware implements a direct conversion analog front end with high-speed Analog-to-Digital Converter (ADC)s and Digital-to-Analog Converter (DAC)s, featuring a fixed-personality FPGA for the Digital Down-Conversion (DDC) and Digital Up-Conversion (DUC) steps. The receiver chain begins with a highly sensitive analog front end capable of receiving very small signals and digitizing them using direct DDC to In-phase (I) and Quadrature (Q) baseband signals. The down-conversion is followed by high-speed analog-to-digital conversion and a DDC that reduces the sampling rate and packetizes I and Q for transmission to a host computer using Gigabit Ethernet for further processing.

The transmitter chain starts with the host computer, where I and Q signals are generated and transferred over the Ethernet cable to the NI USRP hardware. A DUC prepares the signals for the DAC, after which I and Q mixing occurs to directly up-convert the signals to produce an RF signal, which is then amplified and transmitted.

The NI USRP RIO is an SDR that provides wireless communications designers an affordable SDR with unprecedented performance for developing next generation 5G wireless communication systems [15].

The USRP RIO has a 2x2 MIMO RF transceiver with a LabVIEW programmable Digital Signal Processor (DSP) oriented Kintex 7 FPGA. The system block diagram is in figure B.1.

The NI USRP RIO software defined radio platform combines 2 full-duplex transmit and receive channels, with 40 MHz channel of real-time bandwidth.

Each RF channel includes a switch allowing for time division duplex (TDD) operation on a single antenna using the TX1 and RX1 port, or Frequency Division Duplex (FDD) operation using two ports, TX1 and RX2.

There are 6 different USRP RIO devices with frequency options that span from 50 MHz to 6 GHz with user programmable digital Input Output (IO) lines for controlling external devices.

The Kintex 7 FPGA is a reconfigurable LabVIEW FPGA target that incorporates DSP48 co-processing for high-rate, low latency applications. PCIe Express x4 connection back to the system controller allows up to 800 Mbit/s of streaming data transfer back to the PC desktop, and 200 Mbit/s to a laptop.

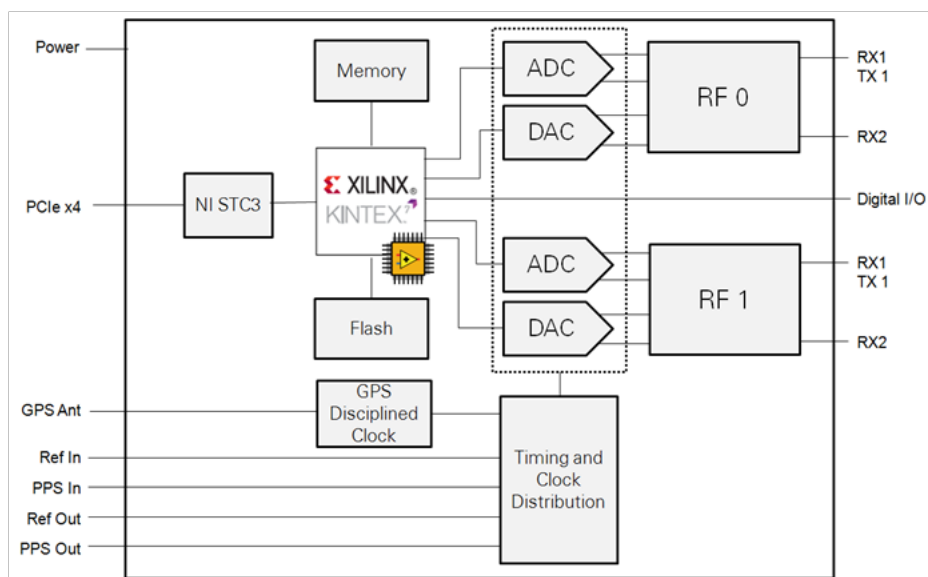
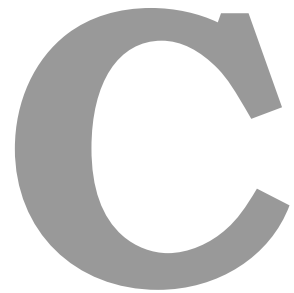


Figure B.1: USRP RIO System Block Diagram (extracted form [15]).



PER Traces for Stationary Scenarios

Plots for the 60 PER traces of stationary scenarios, for packet size of 500 bytes and several attenuations.

C.1 Pure Rayleigh

This subsection presents the sliding average PER traces for the single-tap pure Rayleigh scenario, for the vehicle speeds of 10 km/s, 50 km/s, 100 km/s and 150 km/s, in Figures C.1, C.2, C.3 and C.4, respectively.

Table C.1 shows the mean values and standard deviation, averaged over the 60 PER traces, for attenuations of 4 dB and 15 dB.

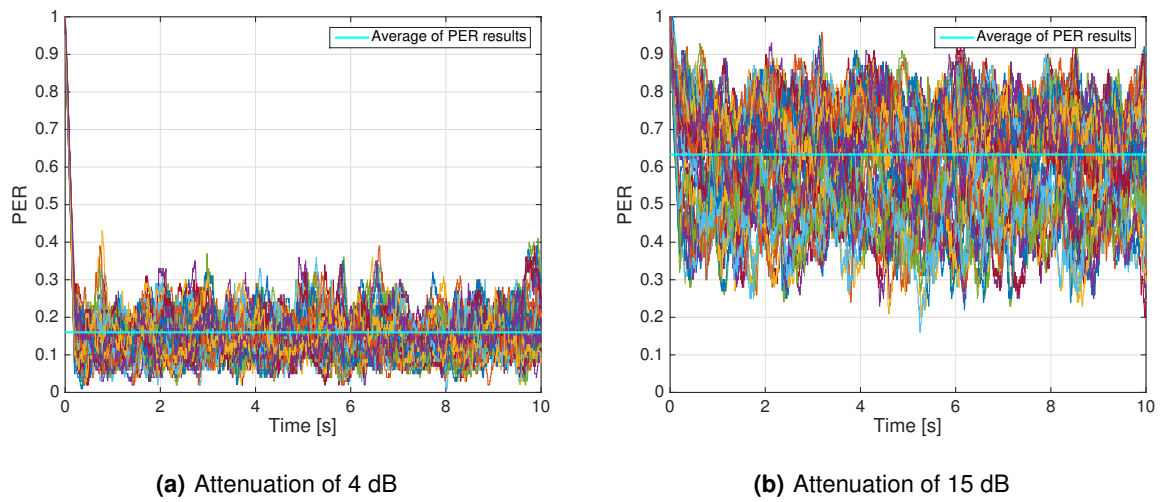


Figure C.1: 60 PER traces for Pure Rayleigh and speed of 10 km/h.

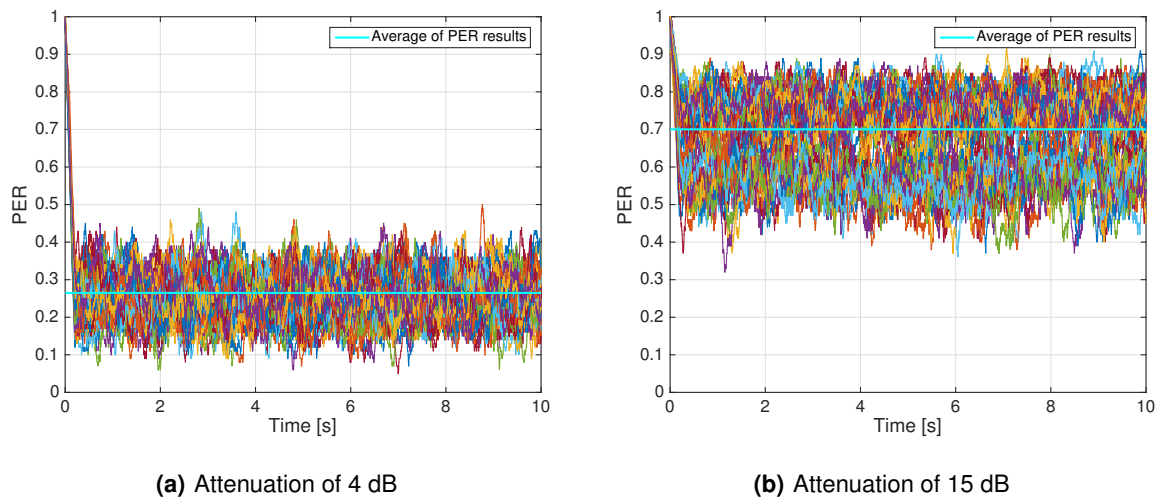
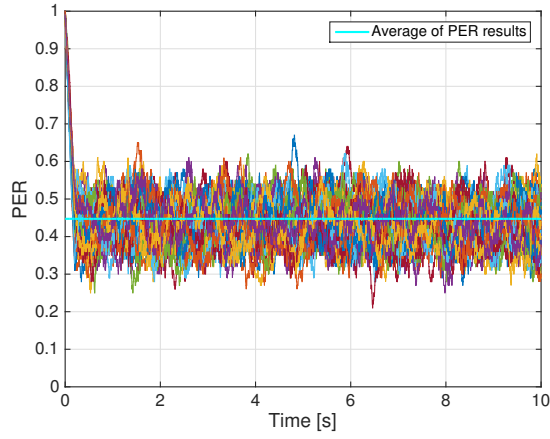
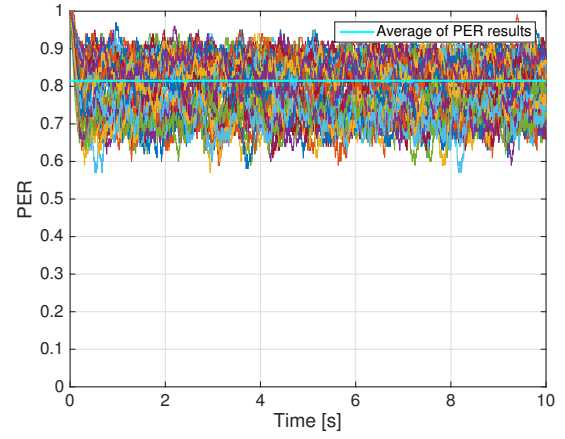


Figure C.2: 60 PER traces for Pure Rayleigh and speed of 50 km/h.

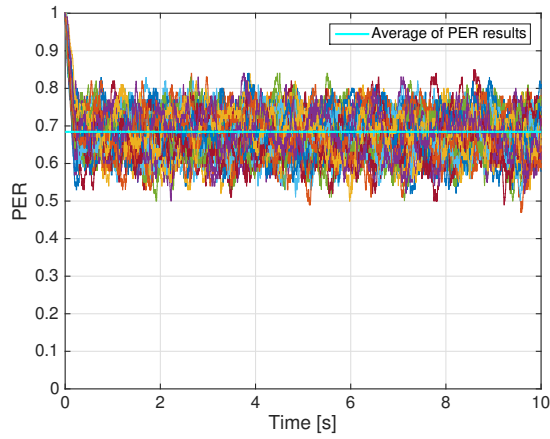


(a) Attenuation of 4 dB

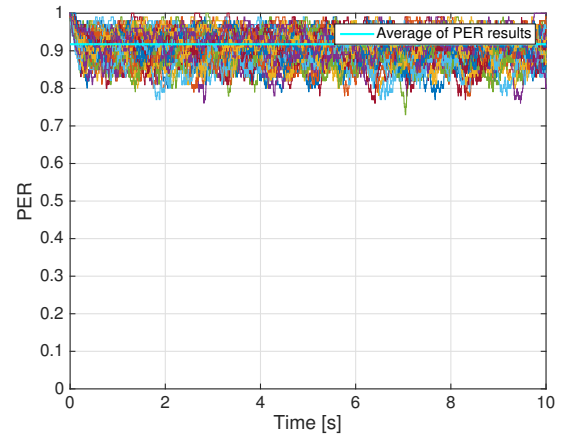


(b) Attenuation of 15 dB

Figure C.3: 60 PER traces for Pure Rayleigh and speed of 100 km/h.



(a) Attenuation of 4 dB



(b) Attenuation of 15 dB

Figure C.4: 60 PER traces for Pure Rayleigh and speed of 150 km/h.

Table C.1: Pure Rayleigh PER results (over 60 traces) average and standard deviation, for attenuation of 4 dB and 15 dB.

Speed [km/h]	Attenuation [dB]	Average PER	PER standard deviation
10	4	0.160	0.108
	15	0.633	0.090
50	4	0.265	0.098
	15	0.700	0.060
100	4	0.447	0.080
	15	0.815	0.042
150	4	0.684	0.059
	15	0.918	0.028

C.2 Pure Rice

This subsection presents the sliding average PER traces for the single-tap pure Rice scenario, for the vehicle speeds of 10 km/s, 50 km/s, 100 km/s and 150 km/s, in Figures C.5, C.6, C.7 and C.8.

Table C.2 shows the mean values and standard deviation, averaged over the 60 PER traces, for attenuations of 4 dB and 15 dB.

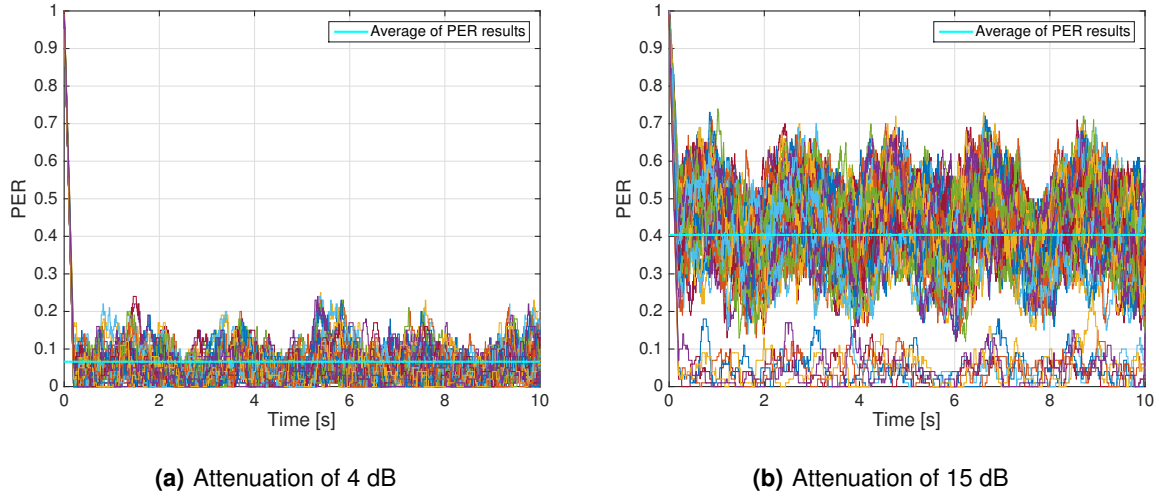


Figure C.5: 60 PER traces for Pure Rice and speed of 10 km/h.

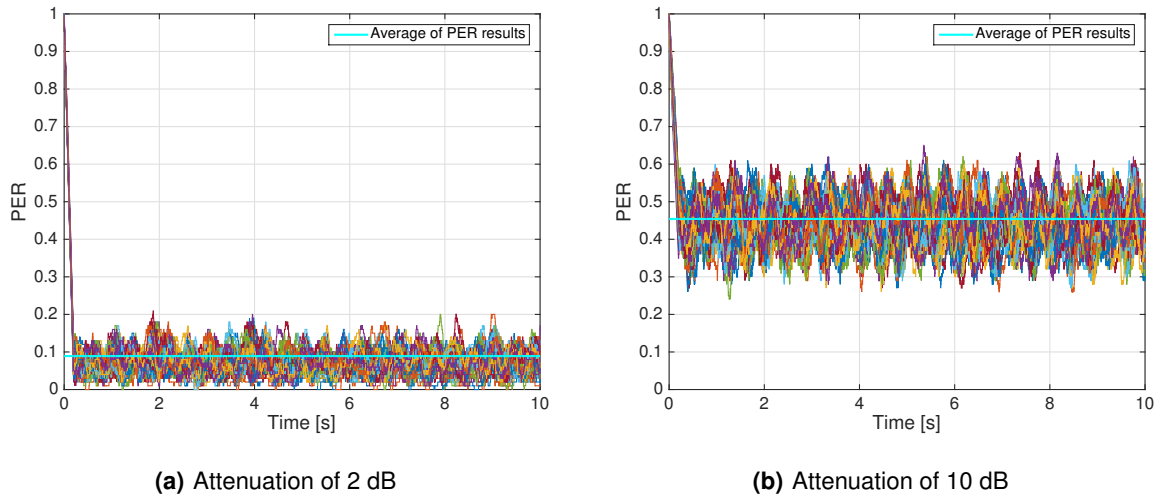
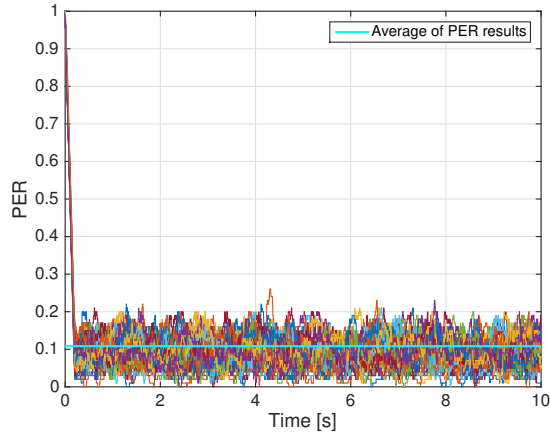
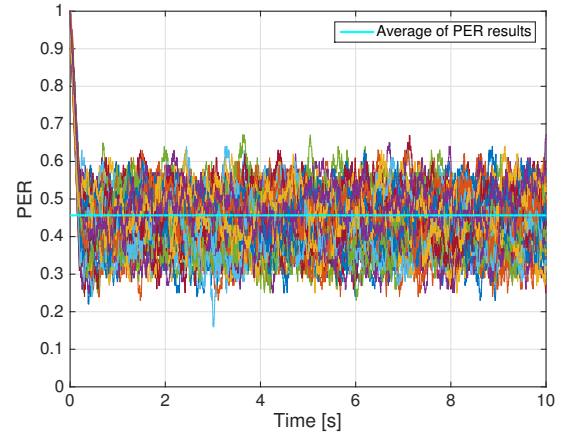


Figure C.6: 60 PER traces for Pure Rice and speed of 50 km/h.

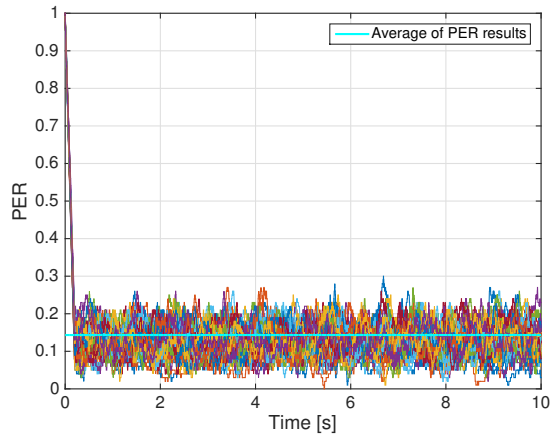


(a) Attenuation of 2 dB

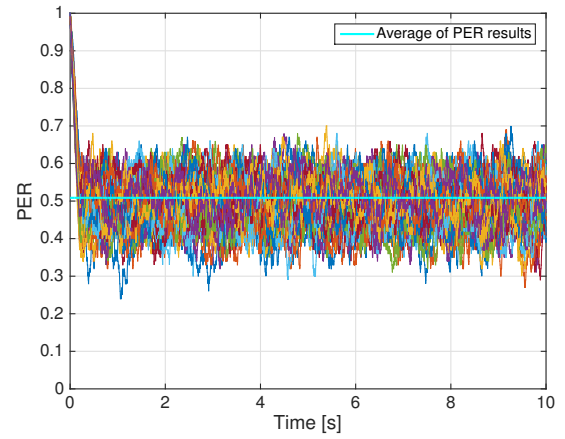


(b) Attenuation of 10 dB

Figure C.7: 60 PER traces for Pure Rice and speed of 100 km/h.



(a) Attenuation of 2 dB



(b) Attenuation of 10 dB

Figure C.8: 60 PER traces for Pure Rice and speed of 150 km/h.

Table C.2: pure Rice PER results (over 60 traces) average and standard deviation, for 4 dB and 15 dB.

Speed [km/h]	Attenuation	\bar{P}_p	σ_p
10 km/h	4	0.066	0.113
	15	0.404	0.105
50 km/h	4	0.089	0.107
	15	0.454	0.079
100 km/h	4	0.108	0.106
	15	0.457	0.079
150 km/h	4	0.143	0.103
	15	0.509	0.072

C.3 Urban Approaching

This subsection presents the sliding average PER traces for the vehicles approaching each other in the urban intersection, known as Urban Approaching in Figure C.9.

Table C.3 shows the mean values and standard deviation, averaged over the 60 PER traces, for attenuations of 4 dB and 15 dB.

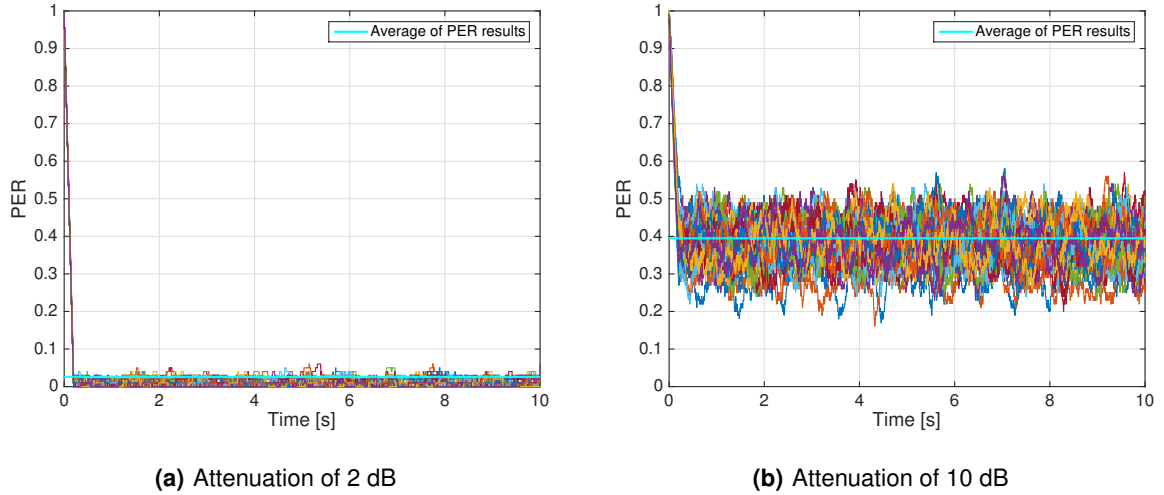


Figure C.9: 60 PER traces for Urban Approaching.

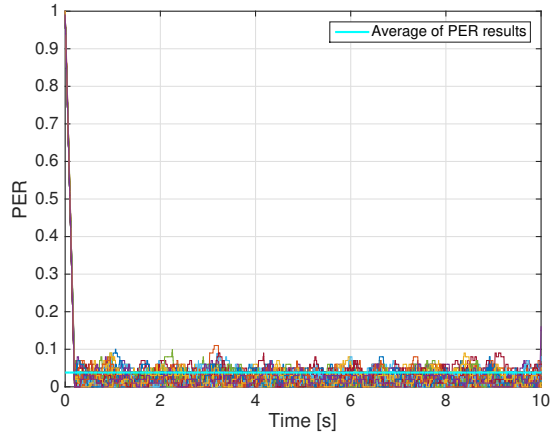
C.4 Urban Crossing

This subsection presents the sliding average PER traces for the vehicles crossing each other in the urban intersection, known as Urban Crossing in Figure C.10.

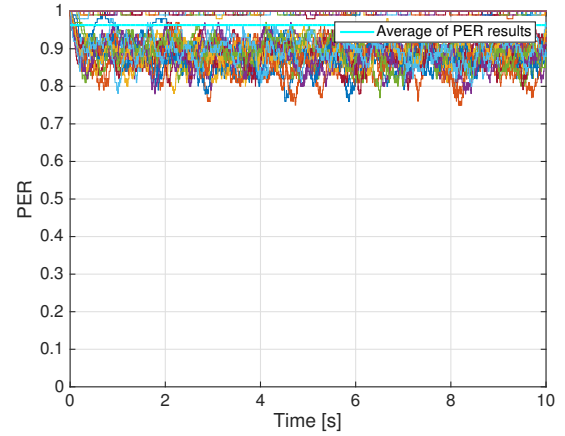
Table C.3 shows the mean values and standard deviation, averaged over the 60 PER traces, for attenuations of 4 dB and 15 dB.

Table C.3: Urban PER results (over 60 traces) average and standard deviation, for 4dB and 15dB.

Scenario	Attenuation	\bar{P}_p	σ_p
Urban Approaching	4	0.026	0.111
	15	0.395	0.083
Urban Crossing	4	0.038	0.111
	15	0.963	0.013



(a) Attenuation of 4 dB



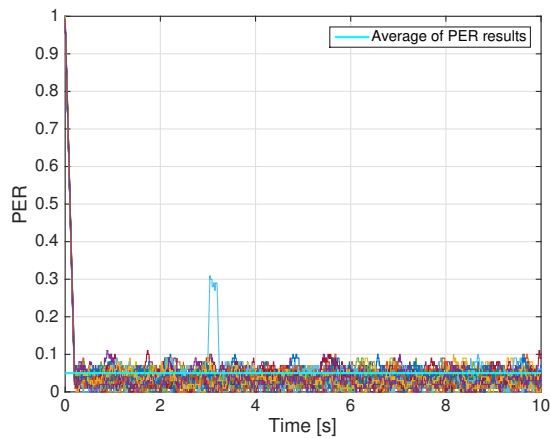
(b) Attenuation of 15 dB

Figure C.10: 60 PER traces for Urban Crossing.

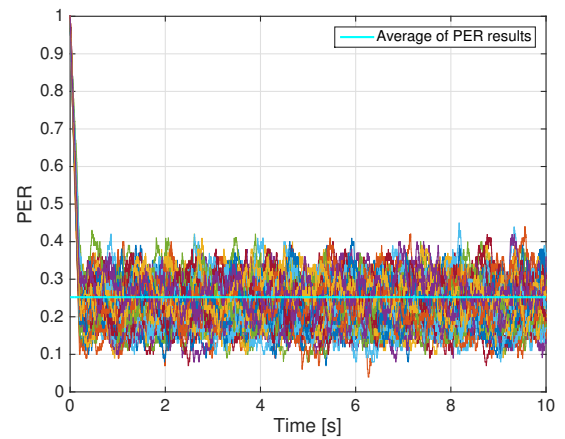
C.5 Highway LOS

This subsection presents the sliding average PER traces for the vehicles in LOS with each other in a Highway scenario in Figure C.11.

Table C.4 shows the mean values and standard deviation, averaged over the 60 PER traces, for attenuations of 4 dB and 15 dB.



(a) Attenuation of 2 dB



(b) Attenuation of 10 dB

Figure C.11: 60 PER traces for Highway with LOS.

C.6 Highway NLOS

This subsection presents the sliding average PER traces for the vehicles that are not in LOS with each other in a Highway scenario in Figure C.12.

Table C.4 shows the mean values and standard deviation, averaged over the 60 PER traces, for attenuations of 4 dB and 15 dB.

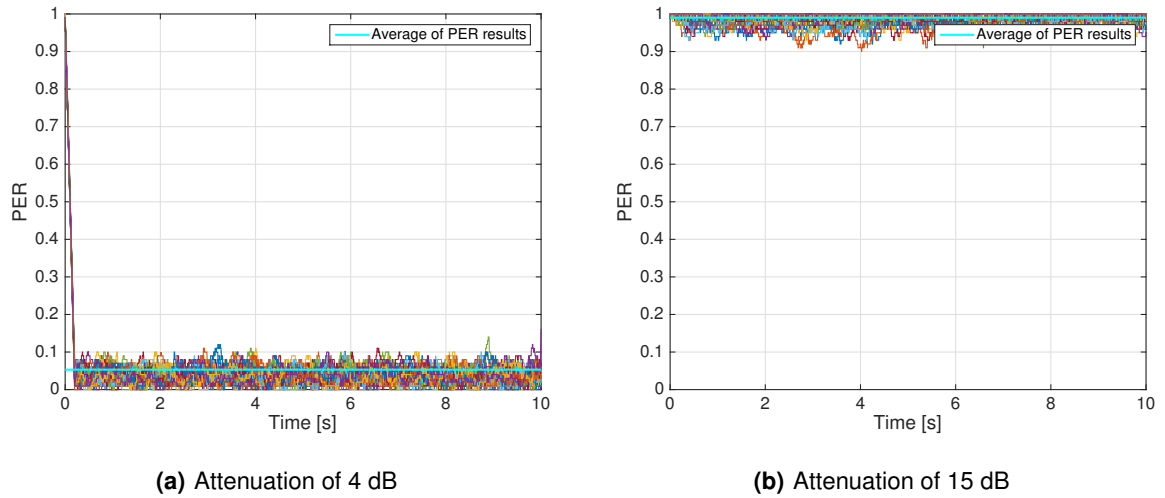


Figure C.12: 60 PER traces for Highway without LOS.

Table C.4: Highway PER results (over 60 traces) average and standard deviation, for 4dB and 15 dB.

Scenario	Attenuation	\bar{P}_p	σ_p
Highway LOS	4	0.050	0.110
	15	0.252	0.095
Highway NLOS	4	0.053	0.110
	15	0.990	0.009



PER Traces for Non-stationary Scenarios

Plots for the 60 PER traces of non-stationary scenarios, for various packet sizes and attenuations.

D.1 Packet Size of 200 Bytes

This subsection presents the sliding average PER traces for non-stationary scenario, 200 bytes packet size and attenuations of 3 dB, 4 dB, 5 dB, 6 dB, 8 dB and 15 dB in Figures D.1, D.2 and D.3.

Table D.1 shows the linear regression parameters, slope m , offset P_0 , and boundaries t_0 and t_{max} , for the 3 regions.

Table D.2 shows the correlation coefficient between the linear regression and original PER traces, the mean value and standard deviation for PER, for all different attenuations.

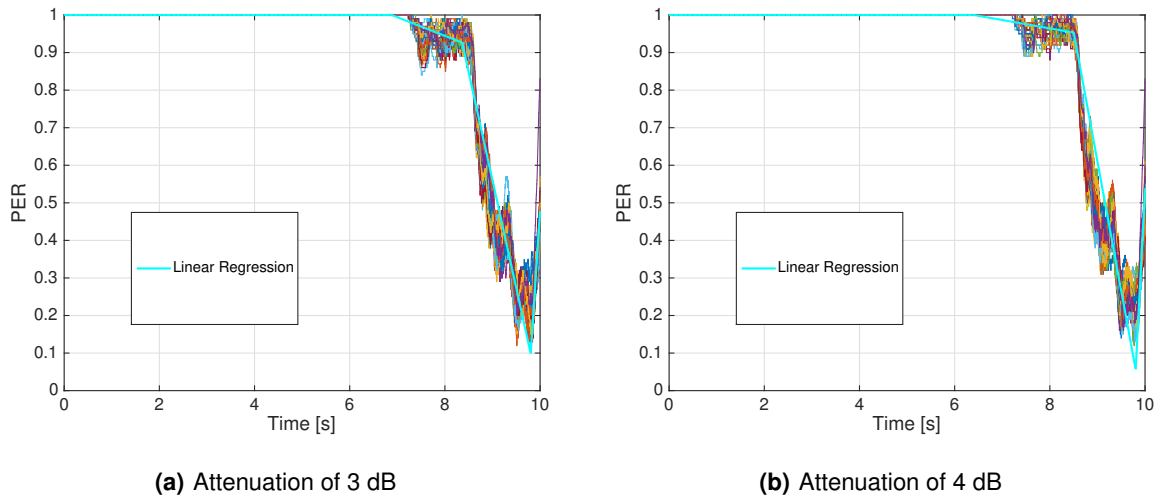


Figure D.1: 60 PER traces for packet size of 200 bytes.

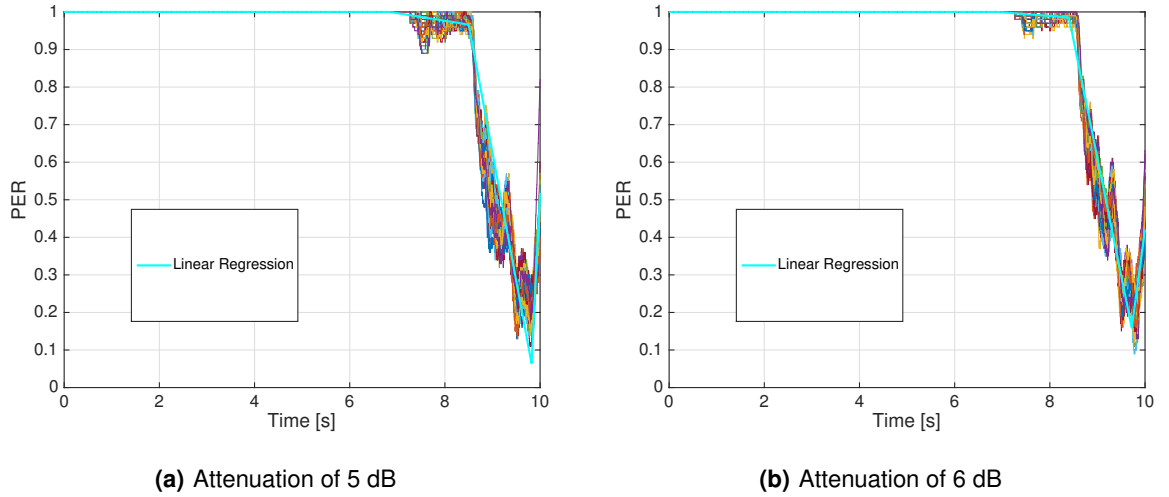
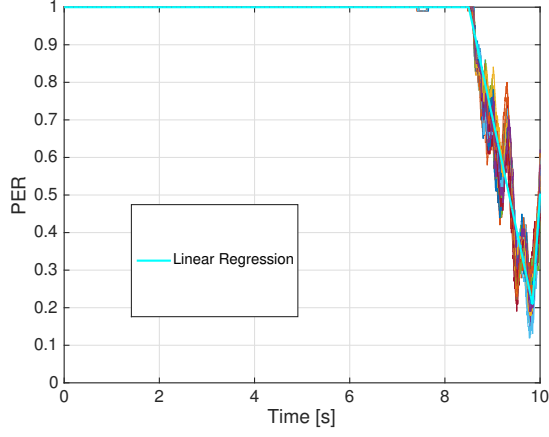
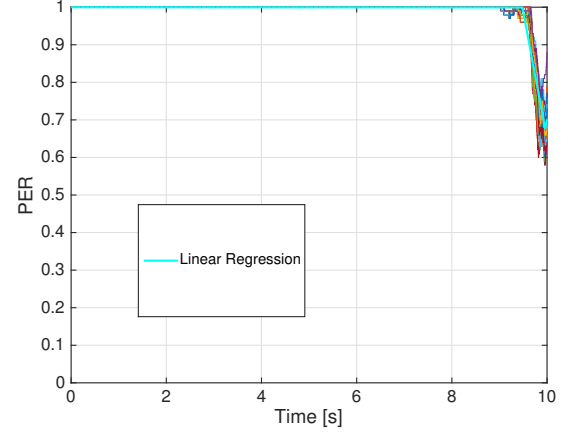


Figure D.2: 60 PER traces for packet size of 200 bytes.



(a) Attenuation of 8 dB



(b) Attenuation of 15 dB

Figure D.3: 60 PER traces for packet size of 200 bytes.

Table D.1: Linear regression equation parameters for 200 bytes and the 3 main regions.

		Attenuation [dB]					
		3	4	5	6	8	15
Region 1	t_0 [s]	6.880	6.400	6.840	6.900	-	-
	t_{\max} [s]	8.360	8.500	8.500	8.400	-	-
	m [%/s]	-0.044	-0.026	-0.015	-0.005	-	-
	P_0 [%]	1.000	1.000	1.000	1.000	-	-
Region 2	t_0 [s]	8.360	8.500	8.500	8.400	8.500	9.480
	t_{\max} [s]	9.802	9.802	9.820	9.720	9.840	9.960
	m [%/s]	-0.549	-0.519	-0.533	-0.565	-0.587	-0.832
	P_0 [%]	0.908	0.932	0.974	0.987	1.000	0.999
Region 3	t_0 [s]	9.802	9.802	9.820	9.720	9.840	9.960
	t_{\max} [s]	10.000	10.000	10.000	10.000	10.000	10.000
	m [%/s]	1.310	1.595	1.528	0.945	1.593	1.394
	P_0 [%]	0.098	0.098	0.135	0.169	0.181	0.601

Table D.2: Average parameters over the 60 traces: correlation coefficient between PER results and regression, PER results average and standard deviation, for 200 bytes packet size.

Attenuation [dB]	$\rho_{P_p, P_p^{reg}}$	\bar{P}_p	σ_p
3	0.990	0.910	0.211
4	0.991	0.913	0.209
5	0.991	0.916	0.206
6	0.991	0.921	0.200
8	0.991	0.936	0.174
15	0.989	0.991	0.045

D.2 Packet Size of 500 Bytes

This subsection presents the sliding average PER traces for non-stationary scenario, 500 bytes packet size and attenuations of 3 dB, 4 dB, 5 dB, 6 dB, 8 dB and 15 dB in Figures D.4, D.5 and D.6.

Table D.3 shows the linear regression parameters, slope m , offset P_0 , and boundaries t_0 and t_{max} , for the 3 regions.

Table D.4 shows the correlation coefficient between the linear regression and original PER traces, the mean value and standard deviation for PER, for all different attenuations.

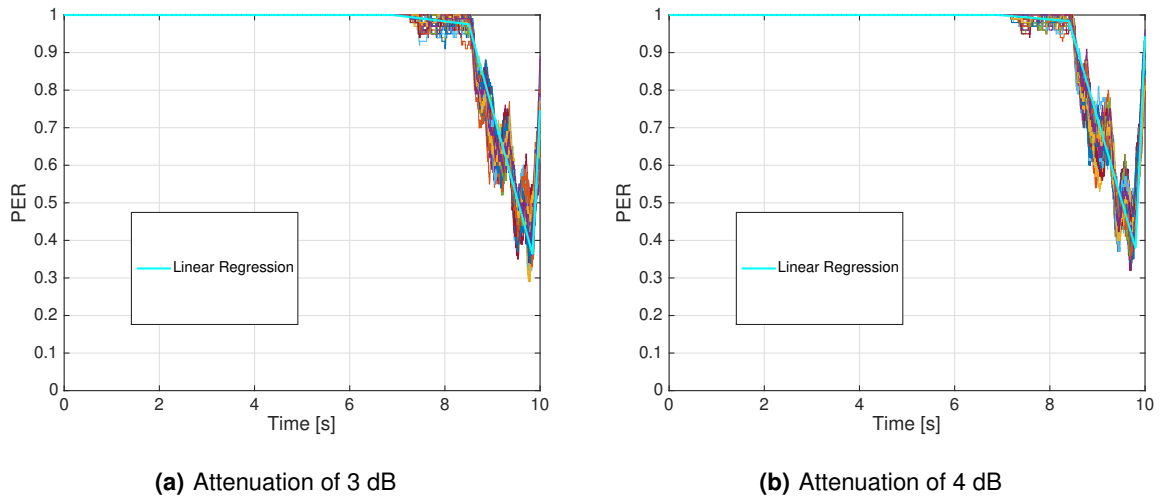


Figure D.4: 60 PER traces for packet size of 500 bytes.

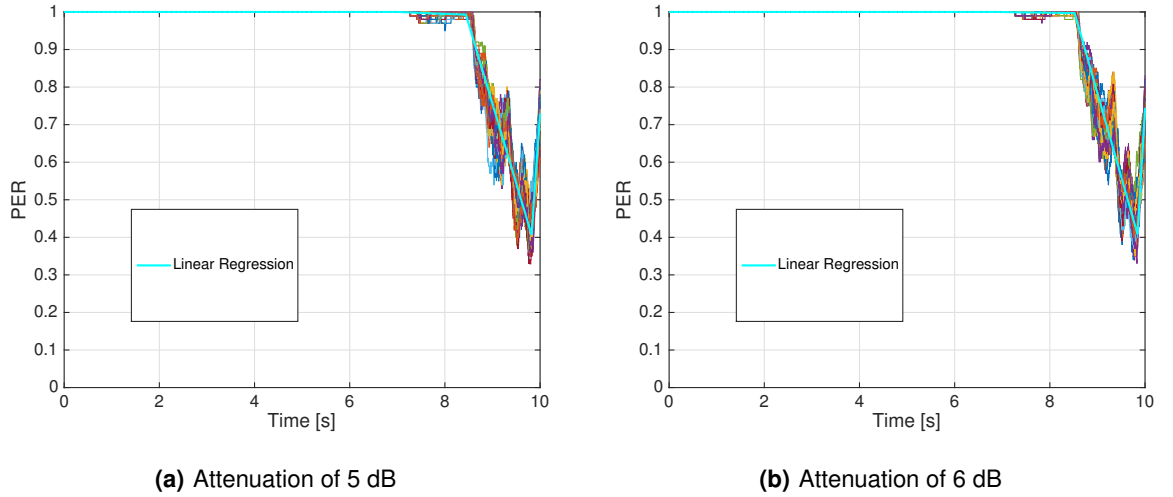


Figure D.5: 60 PER traces for packet size of 500 bytes.

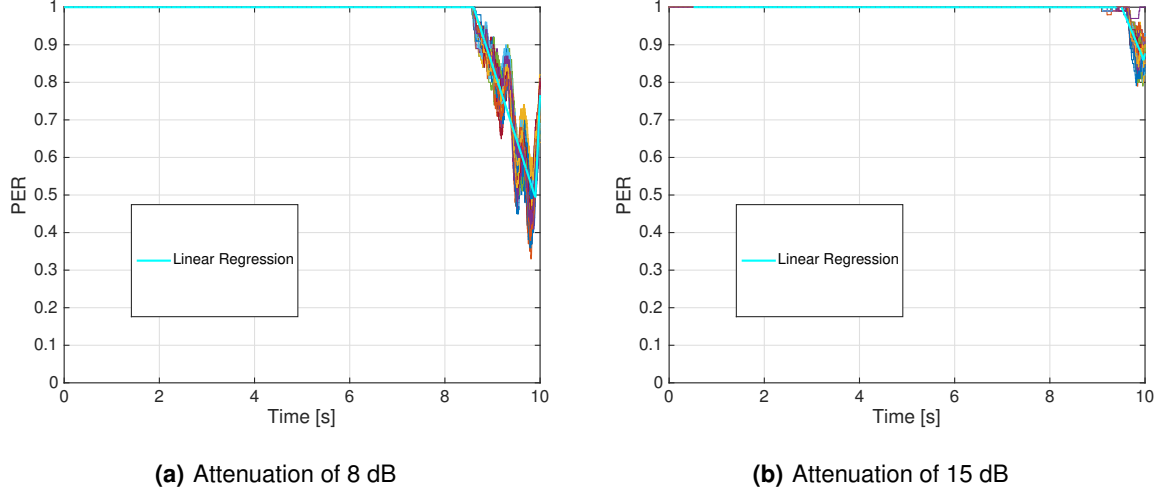


Figure D.6: 60 PER traces for packet size of 500 bytes.

Table D.3: Linear regression equation parameters for 500 bytes and the 3 main regions.

		Attenuation [dB]					
		3	4	5	6	8	15
Region 1	t_0 [s]	6.900	6.940	7.000	6.900	-	-
	t_{\max} [s]	8.500	8.400	8.440	8.520	-	-
	m [%/s]	-0.013	-0.007	-0.002	-0.001	-	-
	P_0 [%]	1.000	1.000	1.000	1.000	-	-
Region 2	t_0 [s]	8.500	8.400	8.440	8.520	8.600	9.400
	t_{\max} [s]	9.840	9.800	9.820	9.840	9.900	9.960
	m [%/s]	-0.391	-0.380	-0.407	-0.396	-0.373	-0.301
	P_0 [%]	0.961	0.982	0.990	0.994	0.999	1.000
Region 3	t_0 [s]	9.840	9.800	9.820	9.840	9.900	9.960
	t_{\max} [s]	10.000	10.000	10.000	10.000	10.000	10.000
	m [%/s]	1.647	1.827	1.676	1.606	1.072	0.477
	P_0 [%]	0.369	0.415	0.389	0.409	0.730	0.809

Table D.4: Average parameters over the 60 traces: correlation coefficient between PER results and regression, PER results average and standard deviation, for 500 bytes packet size.

Attenuation [dB]	$\rho_{P_p, P_p^{reg}}$	\bar{P}_p	σ_p
3	0.990	0.945	0.138
4	0.990	0.946	0.134
5	0.989	0.950	0.132
6	0.989	0.952	0.127
8	0.985	0.963	0.105
15	0.927	0.996	0.018

D.3 Packet Size of 1000 Bytes

This subsection presents the sliding average PER traces for non-stationary scenario, 1000 bytes packet size and attenuations of 3 dB, 4 dB, 5 dB, 6 dB, 8 dB and 15 dB in Figures D.7, D.8 and D.9.

Table D.5 shows the linear regression parameters, slope m , offset P_0 , and boundaries t_0 and t_{max} , for the 3 regions.

Table D.6 shows the correlation coefficient between the linear regression and original PER traces, the mean value and standard deviation for PER, for all different attenuations.

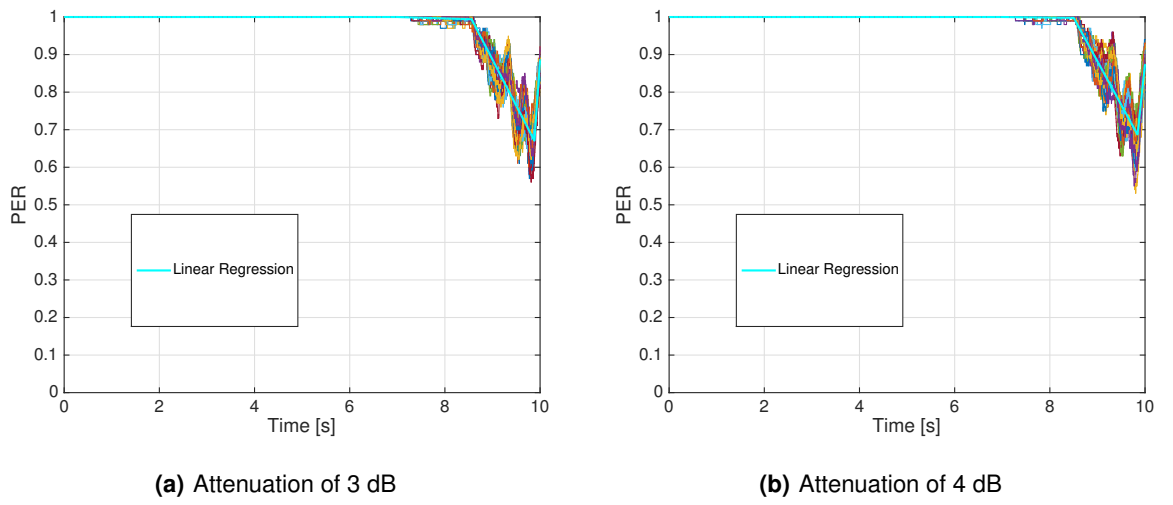


Figure D.7: 60 PER traces for packet size of 1000 bytes.

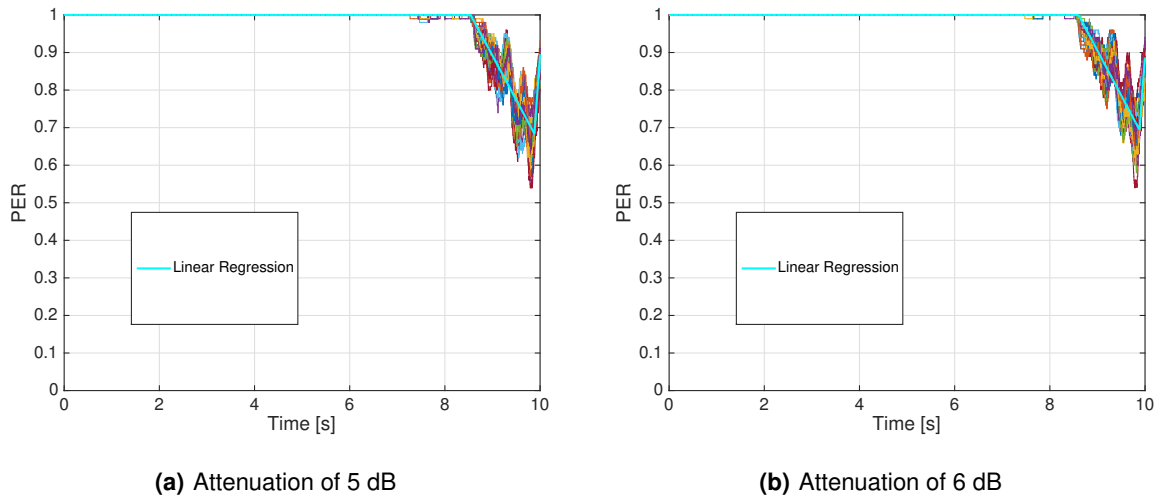
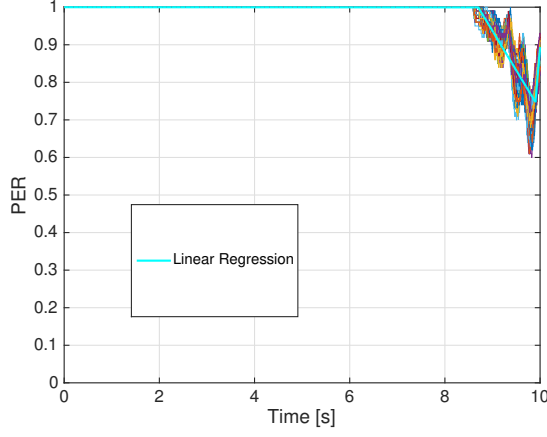
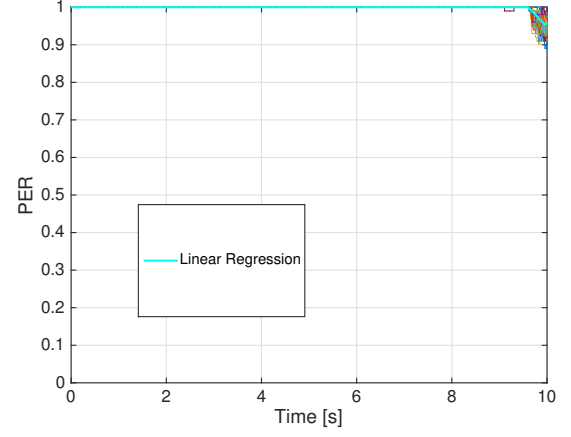


Figure D.8: 60 PER traces for packet size of 1000 bytes.



(a) Attenuation of 8 dB



(b) Attenuation of 15 dB

Figure D.9: 60 PER traces for packet size of 1000 bytes.

Table D.5: Linear regression equation parameters for 1000 bytes and the 3 main regions.

		Attenuation [dB]					
		3	4	5	6	8	15
Region 1	t_0 [s]	7.000	7.000	-	-	-	-
	t_{\max} [s]	8.560	8.500	-	-	-	-
	m [%/s]	-0.003	-0.001	-	-	-	-
	P_0 [%]	1.000	1.000	-	-	-	-
Region 2	t_0 [s]	8.560	8.500	8.540	8.600	8.700	9.600
	t_{\max} [s]	9.880	9.840	9.900	9.880	9.900	9.960
	m [%/s]	-0.229	-0.233	-0.227	-0.225	-0.228	-0.105
	P_0 [%]	0.996	0.999	0.999	0.999	0.999	1.000
Region 3	t_0 [s]	9.880	9.840	9.900	9.880	9.900	9.960
	t_{\max} [s]	10.000	10.000	10.000	10.000	10.000	10.000
	m [%/s]	1.171	1.246	1.159	1.085	1.028	0.061
	P_0 [%]	0.636	0.639	0.699	0.749	0.723	0.965

Table D.6: Average parameters over the 60 traces: correlation coefficient between PER results and regression, PER results average and standard deviation, for 1000 bytes packet size.

Attenuation [dB]	$\rho_{P_p, P_p^{reg}}$	\bar{P}_p	σ_p
3	0.982	0.974	0.069
4	0.981	0.975	0.068
5	0.980	0.976	0.067
6	0.980	0.978	0.063
8	0.973	0.983	0.051
15	0.971	0.999	0.006

References

- [1] P. Papadimitratos, A. La Fortelle, K. Evenssen, R. Brignolo, and S. Cosenza, "Vehicular communication systems: Enabling technologies, applications, and future outlook on intelligent transportation," *IEEE Communications Magazine*, vol. 47, no. 11, pp. 84–95, Nov. 2009. [Online]. Available: <http://ieeexplore.ieee.org/document/5307471/>
- [2] Universidade de Coimbra, "Smart Vehicles 2017 » Universidade de Coimbra," in *4th International Workshop on Smart Vehicles: Connectivity Technologies and ITS Applications*, Macau, China, June 12, 2017. [Online]. Available: <http://smartvehicles.uc.pt/>
- [3] C. F. Mecklenbrauker, A. F. Molisch, J. Karedal, F. Tufvesson, A. Paier, L. Bernado, T. Zemen, O. Klemp, and N. Czink, "Vehicular Channel Characterization and Its Implications for Wireless System Design and Performance," *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1189–1212, jul 2011. [Online]. Available: <http://ieeexplore.ieee.org/document/5710954/>
- [4] S. Ninan, B. Gangula, M. von Alten, and B. Sniderman, "The Internet of Things in the automotive industry — Deloitte Insights," *Deloitte Insights*, 2015. [Online]. Available: <https://dupress.deloitte.com/dup-us-en/focus/internet-of-things/iot-in-automotive-industry.html>
- [5] A. Laouiti, A. Qayyum, M. Naufal, and M. Saad, *Vehicular Ad-hoc Networks for Smart Cities, First International Workshop*, 2014, vol. 306. [Online]. Available: <http://link.springer.com/10.1007/978-981-287-158-9>
- [6] Gartner, "Gartner Says By 2020, a Quarter Billion Connected Vehicles Will Enable New In-Vehicle Services and Automated Driving Capabilities," 2015. [Online]. Available: <https://www.gartner.com/newsroom/id/2970017>
- [7] Dirk Wollschlaeger, "What's Next? V2V (Vehicle-to-Vehicle) Communication With Connected Cars — WIRED," 2014. [Online]. Available: <https://www.wired.com/insights/2014/09/connected-cars/>
- [8] W. Viriyasitavat, M. Boban, Hsin-Mu Tsai, and A. Vasilakos, "Vehicular Communications: Survey and Challenges of Channel and Propagation Models," *IEEE Vehicular Technology Magazine*, vol. 10, no. 2, pp. 55–66, Jun. 2015. [Online]. Available: <http://ieeexplore.ieee.org/document/7108160/>

- [9] Anthony C. Onuorah, "Collision Prediction and Accident Reconstruction using Vehicular Communication : Smart Systems Laboratory." [Online]. Available: <http://smartlab.herts.ac.uk/project/collision-prediction-and-accident-reconstruction-using-vehicular-communication/>
- [10] I. Standard, "INTERNATIONAL STANDARD ISO / IEC / IEEE Telecommunications and information," vol. 2012, 2012.
- [11] M. Standards Committee of the IEEE Computer Society, "IEEE Std 802.11p-2010 (Amendment to IEEE Std 802.11-2007) IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan networks—Specific requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications— Amendment 6: Wireless Access in Vehicular Environments." [Online]. Available: <https://www.ietf.org/mail-archive/web/its/current/pdfqf992dHy9x.pdf>
- [12] 3GPP, "LTE." [Online]. Available: <http://www.3gpp.org/technologies/keywords-acronyms/98-lte>
- [13] Nokia, "Vehicle-to-Everything (V2X) communication," 2016. [Online]. Available: <https://networks.nokia.com/vehicle-to-everything>
- [14] 5G Americas, "V2X Cellular Solutions," 2016. [Online]. Available: <http://www.5gamericas.org/files/2914/7769/1296/5GA{-}V2X{-}Report{-}FINAL{-}for{-}upload.pdf>
- [15] "Overview of the NI USRP RIO Software Defined Radio - National Instruments." [Online]. Available: <http://www.ni.com/white-paper/52119/en/>
- [16] X. Ma, X. Chen, P. Hightower, M. Abdul-hak, N. Al-Holou, U. Mohammad, K. Sjöberg-Bilstrup, E. Uhlemann, E. G. Strom, A. M. S. Abdelgader, W. Lenan, S. M. Nazir, and R. Rastogi, "The Physical Layer of the IEEE 802 . 11p WAVE Communication Standard : The Specifications and Challenges," *Electric Vehicles - Modelling and Simulations*, vol. II, no. 3, pp. 1–5, 2014. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5503941{%}5Cnhttp://www.intechopen.com/books/electric-vehicles-modelling-and-simulations/predictive-intelligent-battery-management-system-to-enhance-the-performance-of-electric-vehicle{%}5Cnhttp://>
- [17] D. Jiang and L. Delgrossi, "IEEE 802.11p: Towards an international standard for wireless access in vehicular environments," *IEEE Vehicular Technology Conference*, pp. 2036–2040, may 2008.
- [18] C. Chen, T. H. Luan, X. Guan, N. Lu, and Y. Liu, "Connected Vehicular Transportation: Data Analytics and Traffic-Dependent Networking," *IEEE Vehicular Technology Magazine*, vol. 12, no. 3, pp. 42–54, sep 2017. [Online]. Available: <http://ieeexplore.ieee.org/document/7994670/>
- [19] R. A. Uzcátegui, U. Nacional, E. Politécnica, and A. José, "Topics in Automotive Networking WAVE : A Tutorial," *IEEE Communications Magazine*, no. May, pp. 126–133, 2009.

- [20] M. Amadeo, C. Campolo, A. Molinaro, and G. Ruggeri, "A WAVE-compliant MAC protocol to support vehicle-to-infrastructure non-safety applications," *Proceedings - 2009 IEEE International Conference on Communications Workshops, ICC 2009*, no. April, 2009.
- [21] Its, "EN 302 663 - V1.2.0 - Intelligent Transport Systems (ITS); Access layer specification for Intelligent Transport Systems operating in the 5 GHz frequency band," 2012. [Online]. Available: <http://www.etsi.orghttp://portal.etsi.org/tb/status/status.asphttp://portal.etsi.org/chaircor/ETSI{ }support.asp>
- [22] P. Sahoo, M.-J. Chiang, and S.-L. Wu, "SVANET: A Smart Vehicular Ad Hoc Network for Efficient Data Transmission with Wireless Sensors," *Sensors*, vol. 14, no. 12, pp. 22 230–22 260, 2014. [Online]. Available: <http://www.mdpi.com/1424-8220/14/12/22230/>
- [23] V. D. Khairnar and S. N. Pradhan, "Simulation Based Evaluation of Highway Road Scenario between DSRC/802.11p MAC Protocol and STDMA for Vehicle-to-Vehicle Communication," *Journal of Transportation Technologies*, vol. 03, no. 01, pp. 88–104, jan 2013. [Online]. Available: <http://www.scirp.org/journal/doi.aspx?DOI=10.4236/jtts.2013.31009>
- [24] A. M. S. Abdelgader and W. Lenan, "The Physical Layer of the IEEE 802.11p WAVE Communication Standard: The Specifications and Challenges," 2014. [Online]. Available: <http://www.iaeng.org/publication/WCECS2014/WCECS2014{ }pp691-698.pdf>
- [25] R. Jain, "Channel models: A tutorial," Feb., 2007. [Online]. Available: http://www.cse.wustl.edu/~jain/cse574-08/ftp/channel_model_tutorial.pdf
- [26] T. S. Rappaport, *Wireless communications : principles and practice*. Prentice Hall PTR, 2002. [Online]. Available: <http://www.ni.com/white-paper/14909/en/>
- [27] A. Goldsmith, *Wireless communications*. Cambridge: Cambridge University Press, 2005.
- [28] DSPRelated.com, "Tapped Delay Line (TDL) — Physical Audio Signal Processing," p. dsprelated.com. [Online]. Available: https://www.dsprelated.com/freebooks/pasp/Tapped_Delay_Line_TDL.html
- [29] T. Blazek, M. Ashury, C. F. Mecklenbrauker, D. Smely, and G. Ghiaasi, "Vehicular channel models: A system level performance analysis of tapped delay line models," in *2017 15th International Conference on ITS Telecommunications (ITST)*. IEEE, may 2017, pp. 1–8. [Online]. Available: <http://ieeexplore.ieee.org/document/7972222/>
- [30] I. Sen and D. W. Matolak, "Vehicle-vehicle channel models for the 5-GHz band," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 2, pp. 235–245, 2008.
- [31] octoScope, "Fundamentals of Wireless Channel Emulation," 2012. [Online]. Available: <http://ww1.prweb.com/prfiles/2012/12/19/10257432/fundamentals-of-channel-emulation21212.pdf>

- [32] Keysight Technologies, “Propsim Channel Emulation Solutions — Keysight (formerly Agilent’s Electronic Measurement),” 2016. [Online]. Available: <https://www.keysight.com/en/pc-2697334/propsim-channel-emulation-solutions?pm=LB{&}nid=-32200.0{&}c=206005.i.1{&}to=79830.g.0{&}cc=PT{&}lc=eng>
- [33] Azimuth Systems, “MIMO RF Channel Emulators for LTE-A, LTE, Legacy.” [Online]. Available: <https://www.azimuthsystems.com/products/ace-channel-emulators/>
- [34] J. Jamison, B. Hewgill, and J. Frolik, “A configurable and repeatable multipath channel emulator,” in *2017 IEEE International Symposium on Antennas and Propagation & USNC/URSI National Radio Science Meeting*. IEEE, jul 2017, pp. 913–914. [Online]. Available: <http://ieeexplore.ieee.org/document/8072499/>
- [35] NI, “What is LabVIEW? - National Instruments.” [Online]. Available: <http://www.ni.com/en-us/shop/labview.html>
- [36] G. Acosta-Marum and M. A. Ingram, “Six time- and frequency- selective empirical channel models for vehicular wireless LANs,” *IEEE Vehicular Technology Magazine*, vol. 2, no. 4, pp. 4–11, 2007.
- [37] T. M. Fern, “FPGA-based Vehicular Channel Emulator for Real-Time Performance Evaluation of IEEE 802 . 11p Transceivers,” vol. 2010, pp. 592–597, 2010.
- [38] 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*, vol. 58, no. 9, pp. 4698–4709, 2009.
- [39] Spirent, “V2X Emulator.” [Online]. Available: <https://www.spirent.com/Products/V2X-Emulator>
- [40] M. Nilsson, P. Hallbjörner, N. Arabäck, B. Bergqvist, and F. Tufvesson, “Multipath Propagation Simulator for V2X Communication Tests on Cars,” *European Conference on Antennas and Propagation (EuCAP) Multipath*, pp. 1342–1346, 2013.
- [41] D. Vlastaras, S. Malkowsky, and F. Tufvesson, “Stress Test of Vehicular Communication Transceivers Using Software Defined Radio,” *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, vol. 41, no. 2, pp. 1–4, 2015. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7146111>
- [42] G. Ghiaasi, M. Ashury, D. Vlastaras, M. Hofer, Zhinan Xu, and T. Zemen, “Real-time vehicular channel emulator for future conformance tests of wireless ITS modems,” in *2016 10th European Conference on Antennas and Propagation (EuCAP)*. IEEE, apr 2016, pp. 1–5. [Online]. Available: <http://ieeexplore.ieee.org/document/7481226/>
- [43] R. Tibshirani, “Regression Selection and Shrinkage via the Lasso,” pp. 267–288, 1996. [Online]. Available: <http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.35.7574>
- [44] T. Blazek and C. F. Mecklenbräuker, “Sparse Time-Variant Impulse Response Estimation for Vehicular Channels Using the c-LASSO,” pp. 0–4, 2017.

- [45] J. Weston JASONWESTON, T. B. Bernhard Schölkopf, and M. Tipping, "Use of the Zero-Norm with Linear Models and Kernel Methods André Elisseeff," *Journal of Machine Learning Research*, vol. 3, pp. 1439–1461, 2003. [Online]. Available: <http://www.jmlr.org/papers/volume3/weston03a/weston03a.pdf>
- [46] CVX Research, "CVX: Matlab Software for Disciplined Convex Programming — CVX Research, Inc." 2012. [Online]. Available: <http://cvxr.com/cvx/>
- [47] F. Kaltenberger, G. Steinbock, G. Humer, and T. Zemen, "Low-complexity geometry based MIMO channel emulation," in *2006 First European Conference on Antennas and Propagation*. IEEE, nov 2006, pp. 1–8. [Online]. Available: <http://ieeexplore.ieee.org/document/4584747/>
- [48] A. Paier, L. Bernado, J. Karedal, O. Klemp, and A. Kwoczek, "Overview of Vehicle-to-Vehicle Radio Channel Measurements for Collision Avoidance Applications," in *2010 IEEE 71st Vehicular Technology Conference*. IEEE, 2010, pp. 1–5. [Online]. Available: <http://ieeexplore.ieee.org/document/5493975/>
- [49] MathWorks, "Linear Regression - MATLAB & Simulink." [Online]. Available: https://www.mathworks.com/help/matlab/data_analysis/linear-regression.html
- [50] B. Sklar, "Rayleigh fading channels in mobile digital communication systems .I. Characterization," *IEEE Communications Magazine*, vol. 35, no. 7, pp. 90–100, jul 1997. [Online]. Available: <http://ieeexplore.ieee.org/document/601747/>
- [51] A. Abdi, C. Tepedelenlioglu, M. Kaveh, and G. Giannakis, "On the estimation of the K parameter for the Rice fading distribution," *IEEE Communications Letters*, vol. 5, no. 3, pp. 92–94, mar 2001. [Online]. Available: <http://ieeexplore.ieee.org/document/913150/>
- [52] "What Is NI USRP Hardware? - National Instruments." [Online]. Available: <http://www.ni.com/white-paper/12985/en/>