

# Implementation of the COST 2100 MIMO Channel Model in C++ / IT++



Jiang, Weiwei

Department of Electrical and Information Technology  
Faculty of Engineering, LTH, Lund University, Sweden

A thesis submitted for the degree of

*Master of Engineering*

Apr. 6, 2012



## Acknowledgements

This master thesis project was done with reference to previous works in the COST 2100 channel model, and it is performed as a part of the requirements for completing my master studies to get the degree in the Wireless Communications program of Lund University.

I would like to appreciate my supervisor Dr. Fredrik Tufvesson and the examiner Professor Ove Edfors for giving support to finish this challenging work. They provided me a pleasant environment to finish my master thesis.

Besides, many thanks are given to Miss Merifang Zhu and Mr Lingfeng Liu. They helped me a lot to understand the concept of the channel model and the MATLAB code. They also shared their experiences with me which guided me a efficient way to accomplish this project. Their supports and suggestions are inevitable for me to complete the work.

I would also humbly say thank you to Mr. Johan Löfgren and Peter Hammarberg for the discussions and guidance during the system simulation stage to verify my work result.

## Abstract

With contributions from previous works such as COST 259 and COST 273, the COST 2100 multi-link Multiple-Input-Multiple-Output (MIMO) channel model inherits many concepts from them.

Based on the implementation in MATLAB, a module of the COST 2100 multi-link MIMO channel model in C++ with IT++ plugin is implemented, with the aim to build a general module which can be integrated into a simulation system for efficient tests and verifications. The implementation loads an external configuration file in XML format with the required parameters and provides interfaces with output of channel responses such as Double Directional Impulse Response (DDIR) or transfer functions.

In current version, it supports the features for both single-link and multi-link cases, and the structure is also flexible with future updates to support new features like autocorrelation, Dense Multipath Components (DMC), etc.

The differences between the two versions of implementations were verified and the module integration in the simple transmission system was also studied.

# Contents

<b>Acronyms and Abbreviations</b>	<b>v</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.1.1 Brief Introduction to COST . . . . .	1
1.1.2 COST 2100 and Previous Actions . . . . .	2
1.2 Objectives and Methodologies . . . . .	3
1.3 Structure of the Report . . . . .	4
<b>2 COST 2100 Multi-link MIMO Channel Model</b>	<b>6</b>
2.1 Introduction . . . . .	6
2.2 General Environments . . . . .	7
2.3 MIMO Systems . . . . .	8
2.4 Generic Channel Model . . . . .	9
2.4.1 Parameters . . . . .	9
2.4.2 Channel Representation: DDIR Model . . . . .	10
2.4.3 Clusters . . . . .	11
2.4.4 LoS Occurrence . . . . .	15
2.4.5 Other Factors . . . . .	16
2.5 Multi-link extension . . . . .	18
2.5.1 Multi-BS and Single MS Communication . . . . .	18
2.5.2 Single BS and Multi-MS Communication . . . . .	19

## Contents

---

2.5.3	Multi-BS / MS Communication . . . . .	21
<b>3</b>	<b>Implementation of the COST 2100 Channel Model</b>	<b>22</b>
3.1	Implementation in Matlab . . . . .	22
3.2	Implementation in C++ with IT++ . . . . .	24
3.2.1	Introduction to IT++ . . . . .	24
3.2.2	Module Framework Structure . . . . .	24
3.2.3	Class Design . . . . .	26
3.2.4	Functions Implementation . . . . .	27
3.3	Conclusions . . . . .	32
<b>4</b>	<b>Verifications and System Simulation Results</b>	<b>33</b>
4.1	Module Verifications . . . . .	33
4.1.1	Unit Tests Flow . . . . .	34
4.1.2	Module Overall Tests . . . . .	36
4.2	System Simulations . . . . .	39
4.2.1	Introduction to the RS-Simulation Chain . . . . .	41
4.2.2	Introduction to Alamouti Transmission Scheme . . . . .	41
4.2.3	Simulation Results . . . . .	43
4.3	Conclusions . . . . .	44
<b>5</b>	<b>Final Conclusions</b>	<b>45</b>
5.1	Summaries and Conclusions . . . . .	45
5.2	Future Works . . . . .	46
<b>A</b>	<b>External Parameters</b>	<b>47</b>
<b>B</b>	<b>Stochastic Parameters</b>	<b>49</b>
	<b>References</b>	<b>53</b>

# Acronyms and Abbreviations

- **3GPP** 3rd Generation Partnership Project
- **AoA** Azimuth of Arrival
- **AoD** Azimuth of Departure
- **BER** Bit Error Rate
- **BPSK** Binary Phase Shift Keying
- **BS** Base Station
- **BSCC** Base Station Common Cluster
- **CDF** Cumulative Distribution Function
- **CDMA** Code Division Multiple Access
- **COST** European Cooperation in the Field of Scientific and Technical Research
- **DDIR** Double Directional Impulse Response
- **DECT** Digital Enhanced Cordless Telecommunications
- **DMC** Bense Multi-path Component
- **DoA** Direction of Arrival

## Acronyms and Abbreviations

---

- **DoD** Direction of Departure
- **EIT** Department of Electrical and Information Technology
- **ETSI** European Telecommunications Standards Institute
- **EoA** Elevation of Arrival
- **EoD** Elevation of Departure
- **GBSCM** Geometric Based Stochastic Channel Model
- **GNU GPL** GNU General Public License
- **GSM** Global System For Mobile Communication
- **HIPERLAN** High Performance Radio LAN
- **ICT** Information Communication Technology
- **ITU-R** International Telecommunication Union Radiocommunication Sector
- **LTH** Lunds Tekniska Högskola
- **LoS** Line of Sight
- **MIMO** Multiple Input Multiple Output
- **MPC** Multiple Path Component
- **MS** Mobile Station
- **MSCC** Mobile Station Common Cluster
- **MoU** Memorandum of Understanding
- **OOP** Object-oriented programming
- **PDDP** Power Directional Delay Profile



- **PDP** Power Delay Profile
- **RX** Receive
- **SISO** Single Input Single Output
- **STBC** Space Time Block Code
- **TDMA** Time Division Multiple Access
- **TX** Transmit
- **UCL** Université Catholique de Louvain
- **UMTS** Universal Mobile Telecommunications System
- **VR** Visibility Region
- **XML** eXtensible Markup Language



# Chapter 1

## Introduction

*With the development of the communication industry, we are now capable to achieve fairly high data rates and low bit-error-rates for better wireless transmission efficiency. Undoubtedly, it is then of great importance to model and study with the wireless channel, which is used to convey the information from the transmitter and the receiver. Statistically a communication channel is usually modeled as a triple consisting of an input stream, an output stream, and a transition relationship for each pair of input and output elements. The COST 2100 Channel Model, which is mainly studied in this thesis project, is a multi-link MIMO channel model.*

### 1.1 Background

#### 1.1.1 Brief Introduction to COST

COST - European Cooperation in the Field of Scientific and Technical Research [2], is one of the longest-running European instruments supporting cooperation among scientists and researchers across Europe since 1971. It is an intergovernmental framework for European Cooperation in Science and Technology, allowing the coordination of nationally-funded research on a Eu-

ropean level, in order to ensure that Europe holds a strong position in the field of scientific and technical research for peaceful purposes.

COST contributes to reducing the fragmentation in European research investments and opening the European Research Area to cooperation worldwide. COST is based on Actions, which are defined by a Memorandum of Understanding (MoU) signed by the Governments of the COST states wishing to participate in the Action, and it has around 200 running COST Actions in nine scientific domains, including ICT - Information and Communication Technologies.

### **1.1.2 COST 2100 and Previous Actions**

COST 2100 is the Action on Pervasive Mobile & Ambient Wireless Communications and belongs to the ICT Domain [1]. This Action basically addresses the various topics that are currently emerging in the area of mobile and wireless communications and was proposed by the group of researchers who participated in the previous COST Actions on mobile communications.

The main objective of this Action is to increase knowledge of mobile and wireless network technologies, by exploring and developing new methods, models, techniques, strategies and tools that will facilitate the implementation of next generation mobile radio communication systems and will foster the development of the paradigms of pervasive and ambient wireless communications.

The Telecommunications area of COST has already in its tradition a stream of very successful Actions that dealt with mobile and wireless communications, and have contributed to the development and standardization of commercial systems:

- COST 207, “Digital Land Mobile Radio Communications”, March 1984 - September 1988, which contributed to the development of GSM, providing the channel model;
- COST 231, “Evolution of Land Mobile Radio (Including Personal) Com-

munications”, April 1989 - April 1996, which contributed to the deployment of GSM1800 and to the development of DECT, HIPERLAN 1 and UMTS, and defined propagation models for the GSM band that are recommended by 3GPP and ITU-R;

- COST 259, “Wireless Flexible Personalized Communications”, December 1996 - April 2000, which contributed to the deployment of DECT and HIPERLAN 1 and to the development of UMTS and HIPERLAN2, conducted simulations over the dispute on the access technique for UMTS (TDMA versus CDMA), supporting ETSI on its decisions, and supported 3GPP in the development of a channel model;
- COST 273, “Towards Mobile Broadband Multimedia Communications”, May 2001 - June 2005, which contributed to standardization efforts in 3GPP and to the deployment of UMTS networks, provided channel models for MIMO systems and initial inputs to the research on B3G generation systems.

## 1.2 Objectives and Methodologies

In this project, COST 2100 MIMO model is studied, and the goal is to implement a general channel model module of COST 2100 in the form of C++ with IT++ plugin. The module will be transplanted into a simulation system for efficiency tests and verifications.

The implementation of the model in C++ is based on the previous measurements and work in MATLAB by Liu et al. Unlike the MATLAB application, a C++ implementation is more flexible and efficient, besides, C++ is commonly used in transmission systems, especially large ones, a C++ implementation for the channel model is more compatible and can be integrated with those systems.

Hence, during the process of the project, the MATLAB code is studied

and verified for the preparation of the C++ / IT++ framework design and programming. The following is a brief overview of the approach that was adopted through this master thesis project:

- Preparation Stage:
  - Study and understand the structure of COST 2100;
  - Study the previous related channel model: COST 273 and COST 259;
  - Study the MATLAB implementation and the simulation environment.
- Implementation Stage:
  - Simulate with the MATLAB code and design the C++ / IT++ framework;
  - Implement the COST 2100 channel model in C++ / IT++.
- Simulation Stage:
  - Module unit verifications and compare the result with the MATLAB simulation result;
  - System integration and verification in the RS Simulation Chain and the Walati Transmission System.
- Final Stage:
  - Generate documentation of the implementation using Doxygen;
  - Finish the master thesis project report.

### 1.3 Structure of the Report

This report consists of 5 chapters, mainly focuses on the the C++ / IT++ framework design, implement and simulation of the COST 2100 MIMO channel model.

It begins with the background information introduction and then follows with the theoretical ideas of the COST 2100 channel model in Chapter 2. In Chapter 3 introduces the design of the channel module implementation

in details. Simulation and test results are followed in Chapter 4. In the final Chapter, conclusions are derived and possible future works, such as upgrades and improvements are also mentioned.

## Chapter 2

# COST 2100 Multi-link MIMO Channel Model

*COST 2100 is an Action related to previous Actions such as COST 259 and COST 273, in fact, many concepts used in COST 2100 are inherited from these previous works. To understand the structure of COST 2100 Channel Model, basic knowledge of COST 259 and COST 273 are required, therefore, the following chapter will also go through some concepts from the earlier COST 259 Directional Channel Model and COST 273 MIMO Channel Model.*

### 2.1 Introduction

The COST 2100 Channel Model emerges from its predecessor COST 273 / COST 259, hence many similarities can be observed. Previous COST Actions have made important contributions that are in worldwide use now: COST 231 extended the well-known Okumura-Hata model for the channel attenuation to frequency ranges and environments of interest to modern cellular systems; COST 207 derived models for the delay dispersion in cellular macrocells; these models were instrumental for the development of the GSM system. COST 259 finally derived a model for the directional characteristics in macro-, micro- and picocells, and merged this approach with a description



of the attenuation and delay dispersion that is more general than the previous models. COST 273 then derived a general channel model for MIMO systems, which uses the same generic channel model for all types of environments.

In fact, the COST 2100 model is to support more propagation specificities of multihop and relay networks, multi-user access, and sensor networks. It has the similar generic model as COST 273, which works with large number of different parameters and different environmental scenarios for MIMO systems.

## 2.2 General Environments

The channel model is parameter based, and parameters vary from scenarios to scenarios. As a first step, the channel environment has to be determined so that the parametrization of the model is to be done. Main environmental categories here in COST 2100 are the same as COST 273 [8]:

- **Macrocells**

Macrocellular environments are generally defined as environments where the BS is placed above the rooftop height of the surrounding buildings. It is the most “conventional” scenario for cellular applications.

- **Microcells**

Microcellular environments are defined as environments where the BS height is at or below the level of the surrounding rooftops, but outdoors. The MS can be located either indoor or outdoor.

- **Picocells**

Picocellular environments are typically defined as environments covering a small area, such as in-building (offices, homes, halls, corridors, etc.), or other areas as tunnels, mines. Picocells are used to extend coverage to indoor areas where outdoor signals do not reach well, or to add network capacity in areas with very dense phone usage, such as train stations.

- **Ad hoc networks**

Ad hoc network environments are characterized by the following properties:

- (i) all transceivers are at approximately the same height,
- (ii) all transceiver stations show nomadic mobility, i.e. remain static for an extended period of time, before being “dropped” to a new location.

Other specified scenarios can also be valid in this channel model if the basic necessary parameters are proposed.

## 2.3 MIMO Systems

Nowadays, it's very clear that MIMO techniques are of great necessity to increase system capacity with diversity, beamforming and spatial multiplexing.

Considering a MIMO channel with  $N_{\text{TX}}$  antennas at the TX, and  $N_{\text{RX}}$  antennas at the RX, the time-variant channel impulse response between the  $j$ th TX antenna and the  $i$ th RX antenna is denoted by  $h_{ij}(t, \tau)$ , where  $t$  is absolute time and  $\tau$  is the delay in time. The MIMO channel response is given by the  $N_{\text{RX}} \times N_{\text{TX}}$  MIMO channel matrix [7]:

$$\mathbf{H}(t, \tau) = \begin{bmatrix} h_{11}(t, \tau) & h_{12}(t, \tau) & \cdots & h_{1N_{\text{TX}}}(t, \tau) \\ h_{21}(t, \tau) & h_{22}(t, \tau) & \cdots & h_{2N_{\text{TX}}}(t, \tau) \\ \vdots & \vdots & \ddots & \vdots \\ h_{N_{\text{RX}}1}(t, \tau) & h_{N_{\text{RX}}2}(t, \tau) & \cdots & h_{N_{\text{RX}}N_{\text{TX}}}(t, \tau) \end{bmatrix}, \quad (2.1)$$

or in the frequency domain, the MIMO channel matrix of transfer functions is:

$$\mathbf{H}(t, f) = \begin{bmatrix} H_{11}(t, f) & H_{12}(t, f) & \cdots & H_{1N_{\text{TX}}}(t, f) \\ H_{21}(t, f) & H_{22}(t, f) & \cdots & H_{2N_{\text{TX}}}(t, f) \\ \vdots & \vdots & \ddots & \vdots \\ H_{N_{\text{RX}}1}(t, f) & H_{N_{\text{RX}}2}(t, f) & \cdots & H_{N_{\text{RX}}N_{\text{TX}}}(t, f) \end{bmatrix}. \quad (2.2)$$

If the MIMO channel matrix has full rank, it allows for estimating the transmitted signal in a straightforward way, and is of high channel capacity. However, when the channel coefficients are correlated, it's not the case. These channel coefficients are determined by the underlying wave propagation in the radio channel. Thus, MIMO systems truly offer significant advantages, but it is still the radio channel which limits the performance bounds of a MIMO system.

## 2.4 Generic Channel Model

The generic channel model that can be used for the generation of DDIR and then the transfer function matrix can be derived from it for arbitrary antenna configurations of the MIMO systems.

### 2.4.1 Parameters

The model distinguishes between external parameters, which are fixed for a simulation run and describe the simulation environment; and the stochastic parameters, which are chosen according to a certain probability density function and are parameterized for the different environments. The external parameters also enter the parametrization of the stochastic impulse responses.

#### 2.4.1.1 External Parameters

The external parameters are parameters that remain fixed for a simulation run. They are changing according to the simulated scenarios, and also the geographical regions. The parameters are chosen to obtain maximum compatibility with the COST 259 model.

#### External parameters for all environments

- Carrier frequency  $f_c$ : [Hz];
- Base station height  $h_{BS}$ : [m];

- Mobile station height  $h_{\text{MS}}$ : [m];
- Base station position  $\vec{r}_{\text{BS}}$ : [m];
- Mobile station position  $\vec{r}_{\text{MS}}$ : [m];
- Antenna configurations: number of antennas and pattern;
- Path-loss model  $P_L$ : [dB/m].

### Additional external parameters for macro- and microcells

- Average rooftop height  $h_B$ : [m];
- Width of roads  $w_r$ : [m];
- Distance between buildings  $w_b$ : [m];
- Road orientation with respect to direct path  $\varphi_R$ : [degree];

### Additional external parameters for picocells and ad hoc networks

- Room size  $l_l \times l_w$ : [m×m];
- Number of floors between BS and MS  $N_{\text{floor}}$ : [integer];
- Building opposite to the building of BS and MS: [yes/no].

The table of external parameters with theoretical suggested values for different scenarios is shown in **Appendix A: External Parameters**.

#### 2.4.1.2 Stochastic Parameters

The stochastic parameters describe the different locations and radio environments in which the MS might be. The parametrization is influenced by the external parameters mentioned above. The table of stochastic parameters with theoretical suggested values for different scenarios is shown in **Appendix B: Stochastic Parameters**.

#### 2.4.2 Channel Representation: DDIR Model

Following the concepts of COST 259 [6], with the parameters, we can calculate clusters and Multi-Path Components (MPCs) in each of them. The

total DDIR can thus be written as the sum of the cluster DDIRs, which in turn can be formulated as [10]

$$P(\tau, \theta_{BS}, \varphi_{BS}, \theta_{MS}, \varphi_{MS}) = P_{\tau}(\tau) P_{\theta}^{BS}(\theta_{BS}) P_{\varphi}^{BS}(\varphi_{BS}) P_{\theta}^{MS}(\theta_{MS}) P_{\varphi}^{MS}(\varphi_{MS}). \quad (2.3)$$

It is assumed in this model that within one cluster, azimuth / elevation angular spread, and delay spread are independent both at the BS and MS. For the independence is per cluster, there can still be significant coupling between Direction of Arrivals (DoAs) and Direction of Departures (DoDs) overall.

### 2.4.3 Clusters

Same as COST 273 model, COST 2100 model also has three kinds of clusters: local clusters around BS and/or MS; clusters incorporating single interaction; and finally a twin-cluster concept allowing for multiple interacting objects [9]. Different kinds of clusters can exist in different scenarios. In macrocells, single cluster is the dominant propagation mechanism, whereas in indoor environments multiple interaction processes account for most of the energy of the arriving radiation. The preset “selection parameter”  $K_{sel}$  gives the ratio of single cluster to multiple clusters.

#### 2.4.3.1 Local Clusters

MS always has a local cluster while BS local cluster occurs only in certain environments, such as in picocell or ad hoc scenarios. For the local cluster as in Fig. 2.1, which results in a large angular spread, a pure single-scattering approach is used. The size is given by their delay spread, specified in the model.

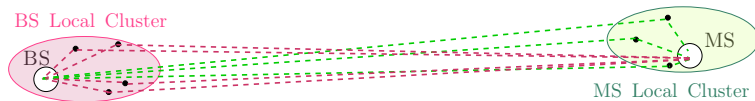


Fig. 2.1. Definition of local cluster.

### 2.4.3.2 Single Clusters

Single clusters also provides a single-scattering between the BS and MS as shown in Fig. 2.2. Different single clusters can be active or inactive to the BS and MS, while the MS is moving, contributions from multiple single clusters have a smooth transition from inactive to active state, and vice versa. To model this change of the cluster activeness, the concept of *Visibility Regions* is introduced.

#### A. Visibility Regions [6]

Each cluster is associated with a visibility region, which is a circular region of the 2-D space characterized by

$$\begin{aligned} R_C &: \text{ size of the visibility region } [\text{m}], \\ L_C &: \text{ size of the transition region } [\text{m}]. \end{aligned}$$

The number of visibility regions is linked to the expected number of clusters in the channel, i.e., if the MS is in a visibility region, then a cluster is active and contributes to the impulse response. In order to give a constant expectation for the number of clusters that equals  $N_C$ , the probability density function of the visibility regions is defined as:

$$\rho_C = \frac{N_C - 1}{\pi(R_C - L_C)^2} [\text{m}^{-2}]. \quad (2.4)$$

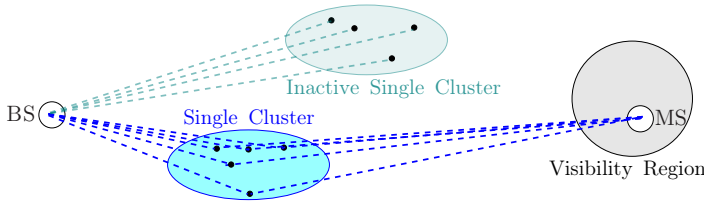


Fig. 2.2. Definition of single cluster.

## B. Cluster Position

For the single-interaction case, the actual cluster position associated with a particular visibility region is determined in the following geometric way:

1. Draw a line from the BS to the center of the visibility region. The cluster position will be determined relative to that connection line.
2. The radial distance from the BS is determined from an exponential distribution

$$f(r) = \begin{cases} 0 & r < r_{\min} \\ \frac{1}{\sigma_r} \exp\left(-\frac{r - r_{\min}}{\sigma_r}\right) & \text{otherwise} \end{cases} \quad (2.5)$$

3. The angle of the cluster center relative to the line is drawn at random from a Gaussian distribution with a standard deviation  $\sigma_{\varphi,C}$ .
4. The delay (assumed to be the minimum delay), angle as seen from the BS, and angle as seen from the MS, are then computed via simple geometrical relationships.
5. Here we get the position of the cluster. The size of the cluster is also drawn from respective distributions<sup>1</sup>.

### 2.4.3.3 Twin / Multiple Clusters

Especially in pico cells, but also in micro-cell scenarios, contributions from multiple clusters are dominant. The frequency of occurrence of the multiple-interaction clusters, as shown in Fig. 2.3, with respect to single-interaction clusters is inversely proportional to the  $K_{\text{sel}}$ .

Similar to the case of single clusters discussed above, visibility regions are placed along the trajectory of the MS, but there is a different way to set the position of the multiple-interaction clusters.

---

<sup>1</sup>The parameters for the distribution are determined in the internal parameters.

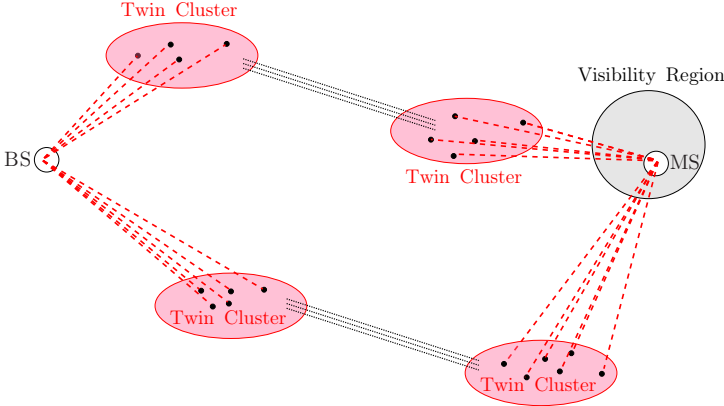


Fig. 2.3. Definition of twin clusters.

The angular spectrum, i.e., the mean AoA, AoD, and minimum delay (delay spread) are computed as random realizations from the marginal distributions of those quantities, which are given in the internal parameters<sup>1</sup>. Then, both cluster azimuth spreads (seen from BS and MS) are drawn.

In this case each cluster is divided into a cluster corresponding to the transmitter part at BS side and the receiver part at MS side. In order to limit complexity, assume clusters look like twins (Twin Clusters), and they have the same interaction objects distributions and long-term behaviors.

The distance to the BS is determined by the cluster delay spread and cluster azimuth spread at the BS, while the distance to the MS is determined in the same way by the cluster delay spread and the cluster azimuth spread at the MS:

$$d_{C,BS/MS} = \frac{d_\tau}{2 \tan \varphi_{C,BS/MS}}. \quad (2.6)$$

where  $d_{C,BS/MS}$  is the distance of the cluster from BS / MS,  $\varphi_{C,BS/MS}$  is the azimuth spread at BS / MS and  $d_\tau = 0.5\tau c_0$ . This makes sure that the dimension in the  $x$  and  $y$  directions is circular and identical, hence the same relative positions can be used for both the BS and MS cluster in the  $x - y$

<sup>1</sup>The internal parameters refer to both the external and stochastic parameters of the channel environments.



plane. To adjust for independent elevation spreads, the elevation height of the scattering objects in the MS cluster  $h_{C,MS}$  is refined by a ratio compared to the BS side  $h_{C,BS}$ .

$$h_{C,BS/MS} = \frac{d_{C,BS/MS}}{\tan \theta_{C,BS/MS}}. \quad (2.7)$$

These two BS / MS parts are connected by an additional cluster-link delay:

$$\tau_C = \tau_{C,BS} + \tau_{C,link} + \tau_{C,MS}, \quad (2.8)$$

which ensures that the total delay of the cluster corresponds to the definitions of the scenario. This could be an artificial case<sup>1</sup> when the total delay is shorter than the delay corresponding to the Line-of-Sight (LoS) connection as a negative value, and it should be avoided by a repositioning of the cluster.

According to the delay, we can get the power of each cluster (with respect to the LoS or quasi-LoS component). The longer the delay, the smaller is the power that it carries. If the attenuation becomes too high, the cluster does not have an impact on the impulse response, thus, there is a limit to the cluster attenuation to avoid the cluster being dropped from consideration. The power of the  $m$ th cluster is

$$P_m = P_0 \max(\exp[-k_\tau(\tau_m - \tau_0)], \exp[-k_\tau(\tau_B - \tau_0)]), \quad (2.9)$$

where  $k_\tau$  is the attenuation coefficient [dB/ $\mu$ s],  $\tau_0$  is the delay of the LoS component [ $\mu$ s],  $\tau_m$  is the delay of the  $m$ th cluster [ $\mu$ s] and  $\tau_B$  is the cut-off delay [ $\mu$ s].

#### 2.4.4 LoS Occurrence

Similar to the visibility region for the clusters, the occurrence of LoS for some environments is modeled stochastically. However, the major difference is that the probability for LoS decreases strongly with the increase of distance

---

<sup>1</sup>The situation occurs mainly when the clusters are originally far away from the MS but subsequently the MS moves close to them.

between the MS and the BS, and is zero after a cut-off distance  $d_{\text{co}}$ . Thus, the model needs an additional parameter as described below:

$$\begin{aligned} d_{\text{co}} &: \text{ cut-off distance for LoS} & [\text{m}], \\ R_L &: \text{ size of the visibility region for LoS} & [\text{m}], \\ L_L &: \text{ size of the transition region for LoS} & [\text{m}]. \end{aligned}$$

Furthermore, the LoS power factor<sup>1</sup> varies due to the existence of an LoS connection. In either case it is modeled as a log-normal random variable, but the mean  $\mu_K$  and variance  $\sigma_K$  are different whether there is an LoS or a quasi-LoS<sup>2</sup>.

## 2.4.5 Other Factors

### 2.4.5.1 Shadow Fading

Usually each cluster undergoes shadow fading, which is modeled as log-normally distributed with standard deviation  $\sigma_S$  [dB]. Besides, the mean of the shadowing variance is correlated with the path loss.

### 2.4.5.2 Auto-correlation and Cross-correlation

The delay spreads and angular spreads are correlated random variables as well as the shadow fading mentioned above, i.e., for cluster  $m$ ,

$$S_m = 10^{\sigma_S X_m / 10} \quad (2.10)$$

$$\sigma_{\tau, m} = \mu_{\tau} \left( \frac{d}{1000} \right)^{\varepsilon} 10^{\sigma_{\tau} Z_m / 10} \quad (2.11)$$

$$\sigma_{\varphi_{\text{BS}}, m} = \mu_{\varphi_{\text{BS}}} 10^{\sigma_{\varphi_{\text{BS}}} Y_m / 10} \quad (2.12)$$

where  $S_m$  denotes the shadow fading,  $\sigma_{\tau, m}$  is the delay spread, and  $\sigma_{\varphi_{\text{BS}}, m}$  stands for the angular spread in azimuth for the BS side.  $X_m$ ,  $Y_m$ , and  $Z_m$

---

<sup>1</sup>LoS power factor is the power of the first component, compared to the power of all other components

<sup>2</sup>The quasi-LoS case describes an NLoS scenario with Ricean fading of the local cluster and therefore a weak LoS component is modeled.

are correlated normal random variables with zero mean and unit variance. Correlated random processes can be computed using the Cholesky factorization [17]. The elevation spread, and for the angular spreads at the MS, can be calculated easily by simply replacing  $\varphi_{\text{BS}}$  with  $\theta_{\text{BS}}$ ,  $\varphi_{\text{MS}}$ , and  $\theta_{\text{MS}}$ , respectively. Altogether, there are six log-normal random variables that are characterized by their cross-correlation coefficients  $\rho_{ij}(i, j = 1 \cdots 6)$ .

Furthermore, if the MS moves over too large a distance, these values will be therefore characterized by a exponential spatial autocorrelation function with the autocorrelation distances  $L_x$ :

$$ACF(x, x') = \exp(-|x - x'|/L_x). \quad (2.13)$$

### 2.4.5.3 Polarization

It is assumed that orthogonal polarizations suffer from statistically independent small-scale fading, while all other statistical parameters (e.g. shadowing, delay spread, etc.) are identical. In the polarization matrix

$$\begin{pmatrix} P_{\text{VV}} & P_{\text{VH}} \\ P_{\text{HV}} & P_{\text{HH}} \end{pmatrix},$$

the entries characterize the powers, averaged over the small-scale fading, and the ratio XPD is a log-normally distributed with its mean  $\mu_{\text{XPD}}$  [dB] and the standard deviation  $\sigma_{\text{XPD}}$ . Besides XPD, all other polarization coefficients are modeled log-normally as well.

$$\text{XPD} = \frac{P_{\text{VV}} + P_{\text{VH}}}{P_{\text{HV}} + P_{\text{HH}}}. \quad (2.14)$$

From some measurements result, the XPD finds its dependencies on the distance between TX and RX, as well as on the delay and the angles of the MPCs [16].

### 2.4.5.4 Diffuse Scattering

Diffuse scattering is the part of the measured signal which cannot be resolved in the angular domain [19]. The diffuse component can be described

by the ratio of power in the discrete components vs the diffuse power with mean  $\mu_{\text{diff}}$  and standard deviation  $\sigma_{\text{diff}}$ . The PDP of the diffuse component is modeled uniformly in azimuth and exponentially in decay.

## 2.5 Multi-link extension

The single-link COST 2100 channel model is developed from a BS-centric, double directional, and geometric based stochastic channel model (GBSCM). With the multi-link extension, which in general refers to concurrent communication between multiple BSs and MSs [14].

Clusters connect a BS-MS link and the scatterers inside contribute signal waves to the link. In multiple BS-MS links, far clusters can be visible to them and the common clusters are observed simultaneously in different BS-MS links. There are two types of common cluster: the commonness with respect to the connection to the BSs (BS-common) and the commonness with respect to the connection to the MSs (MS-common).

### 2.5.1 Multi-BS and Single MS Communication

A Multi-BS and Single MS communication is a single MS communicates with multiple BSs simultaneously. If the MS is located at a certain area, which enables it to link to different BSs, the cluster link to the MS becomes visible with respect to multiple BSs in the BS-MS links.

#### **BS-common Clusters (BSCC)**

The cluster that the MS reaches multiple BSs via, is a common cluster defined as BS-common cluster (BSCC). And in the COST 2100 channel model, the visibility of the BSCC is determined by its own VR, which has the same physical description of an ordinary non-common VR. Once the MS is inside that common VR, it will be connected with multiple BSs via the related BSCC,

as in Fig. 2.4. At the BS side, the commonness is described as the number of the BSs sharing the BSCC.

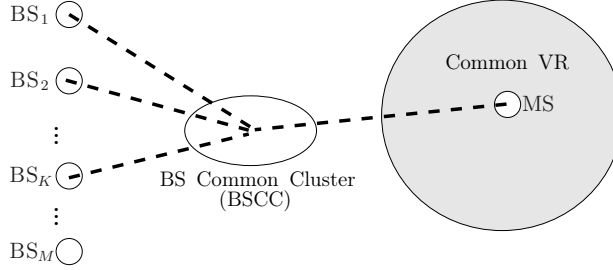


Fig. 2.4. Definition of BS Common Cluster.

For example, in a multi-link communication with  $M$  BSs, if  $K$  BSs share a BSCC ( $1 < K < M$ ), it's defined as  $K$ -BSCC with a ratio:

$$\rho_{K\text{-BSCC}} = \frac{N_{K\text{-BSCC}}}{N_C}, \quad (2.15)$$

$$\sum_{K=1}^M \rho_{K\text{-BSCC}} = 1, \quad (2.16)$$

where  $N_{\text{BSCC}}$  is the total number of  $K$ -BSCC and  $N_C$  is the total number of clusters, in all links.

### 2.5.2 Single BS and Multi-MS Communication

A Single BS and Multi-MS communication is defined the same process as multiple MSs at distinct locations communicate simultaneously with a same single BSs on the contrary. But the case of MSs within the same VR is excluded, because the communication happens actually in the same link. Here different MSs are apart located at a farther distance, compared to the size of VR to each other, and thus are usually linked with the BS via different clusters in the environment.

### MS-common Clusters (MSCC)

However, at some special areas distributed in the environment, there can be a same common cluster for different MSs to link with the BS. Such common cluster is defined as MS-common cluster (MSCC) as shown in Fig. 2.5. The MSCC is related with multiple VRs, defined as a VR group, and a MS located in any one of the VRs in the VR group can reach the BS-MS link via this MSCC, and these VRs also have the same physical description as the single-link COST 2100 channel model.

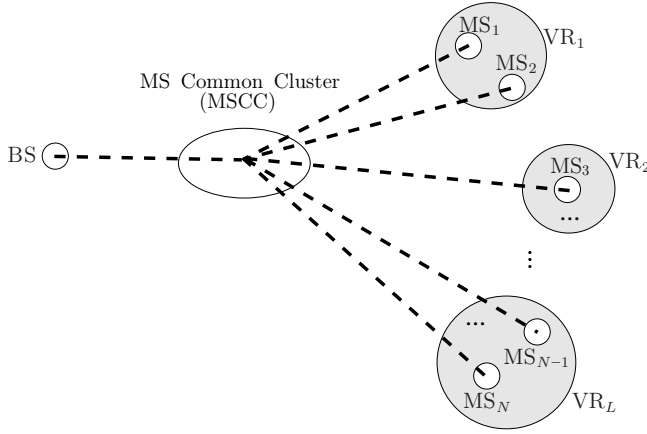


Fig. 2.5. Definition of MS Common Cluster.

It is observed from measurements that as MS moving across these areas common cluster which can discontinuously be observed in the BS-MS link for a total lifetime  $T_{\text{MSCC}}$  larger than the average lifetime of an ordinary far cluster  $T_{\text{C,avg}}$ . The number of VRs,  $L$ , in the VR group of a MS-CC is then calculated by:

$$L_{\text{MSCC}} = \frac{T_{\text{MSCC}}}{T_{\text{C,avg}}}, \quad (2.17)$$

and  $L_{\text{MSCC}}$  is modeled as a Poisson distributed random variable with its mean value extracted from the measurements.

### 2.5.3 Multi-BS / MS Communication

In Multi-BS and Multi-MS communication, it's a combination of the previous two common clusters, i.e., a cluster can exhibit both BS-commonness and MS-commonness, and these two kinds of commonness are described respectively without any obligatory dependencies in the COST 2100 multi-link model.

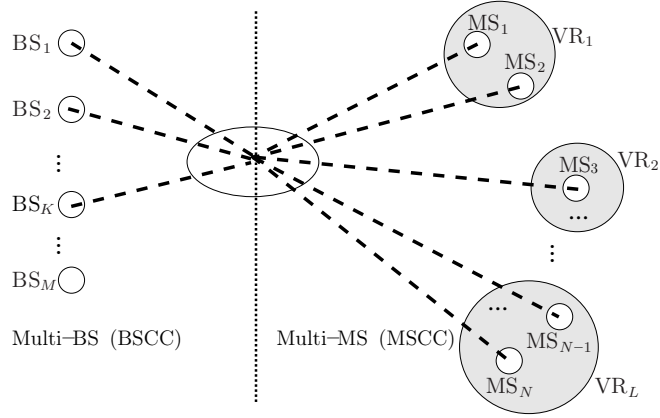


Fig. 2.6. Multi-BS / MS Communication with BS and MS Common Cluster.

In Fig. 2.6, the common cluster becomes a BSCC for BS<sub>1</sub> and BS<sub>2</sub> with MS<sub>1</sub> at common VR<sub>1</sub>, while it also can become a MSCC with the VR group including VR<sub>1</sub> and VR<sub>2</sub>. The freedom of connection to BSs and to MSs for a common cluster in both sides can allow more complicated commonness than the single BSCC / MSCC described above. Moreover, in reality, the VR density for each BS in Multi-BS and Multi-MS communications fluctuates, but it's modeled as identical for different BSs for simplicity here.

## Chapter 3

# Implementation of the COST 2100 Channel Model

*The following chapter delves into the design and implementation of the COST 2100 channel model and the associated details, with all the concepts introduced in Chapter 2. It starts with a brief introduction of the MATLAB implementation and the IT++ library on the C++ platform. Then a detail list of class design and each important functions are discussed, and it ends with a short conclusion.*

### 3.1 Implementation in Matlab

Under the framework of COST 2100 “Pervasive Mobile & Ambient Wireless Communication”, the channel model with the multi-link extension is initially implemented in MATLAB by Liu from ICTEAM, UCL, Belgium [15]. The program package consists of the list of all necessary functions as following:

- *Main Function*

```
function [paraEx paraSt link env BS MS]=  
COST_2100(network,freq,snapRate,snapNum,BSPos,MSPos,MSVelo,antBS,antMS)
```

Generate the channel information in the form of DDIR.



- *Important Functions*

```
function [paraEx paraSt]=
```

```
get_para(network,freq,snapRate,snapNum,posBS,posMS,veloMS,antBS,antMS)
```

Get the preset parameters according to different network scenarios.

```
function VRtable=get_VRtable(paraEx,paraSt)
```

```
function VRLOS=get_VRLOS(BS,paraEx,paraSt)
```

```
function VR=get_VR(VRtable,paraEx,paraSt)
```

Generate the common cluster distribution and VRs.

```
function cluster=get_cluster(VR,VRtable,paraEx,paraSt)
```

```
function cluster=get_cluster_local(type,ind,paraEx,paraSt)
```

```
function mpc=get_mpc(cluster,paraSt)
```

Generate the local and far cluster information and the MPCs inside each cluster.

```
function channel=
```

```
get_channel(BS,MS,VRtable,VR,cluster,mpc,dmc,paraEx,paraSt)
```

```
function h_loos=get_channel_loos(channel,BS,MS,paraEx,paraSt)
```

Get the channel link between each pair of BS and MS.

```
function MS_new=update_chan(MS,paraEx,paraSt)
```

Update the channel with the movement of MS.

- *Utility Functions and Plot Functions*

```
function matrix=rotate_matrix(phi,theta)
```

```
function pathloss=calc_pathloss(pos_BS,pos_MS,paraEx)
```

```
function l=calc_dist(a,b)
```

```
function visual_pddp(channel,paraEx,paraSt)
```

```
function visual_channel(link,env,paraEx,paraSt)
```

## 3.2 Implementation in C++ with IT++

With the help of the previous work in MATLAB, the project is mainly devoted to transplant it into C++ platform with IT++ as a general package for more convenient application in channel modeling and system simulation. The implementation will include all the features in the existing MATLAB program and provide general interfaces for certain systems integration.

### 3.2.1 Introduction to IT++

IT++[\[4\]](#), which originates from the former department of Information Theory at the Chalmers University of Technology, Gothenburg, Sweden, is a C++ library composed of classes and functions for mathematical calculations, linear algebra (matrices and vectors), signal processing and telecommunication systems. It can also be used in areas such as machine learning and pattern recognition as well. It is also a main application in simulation of communication systems aiding research. Templated vector and matrix classes are the core of the IT++ library, making its functionality similar to that of MATLAB and GNU Octave. The library package is a well defined and mature framework based on object oriented programming.

Here in this project, the main core of IT++ is used for all the computations and the implementation of COST 2100 channel model is a supplementary solution to channel models of the current IT++ library in the communication features.

### 3.2.2 Module Framework Structure

Similar to the MATLAB implementation, according to the generic model of the COST 2100 channel, the structure of this C++ implementation is designed under General Public License (GPL) [\[3\]](#) as following in Fig. [3.1](#):

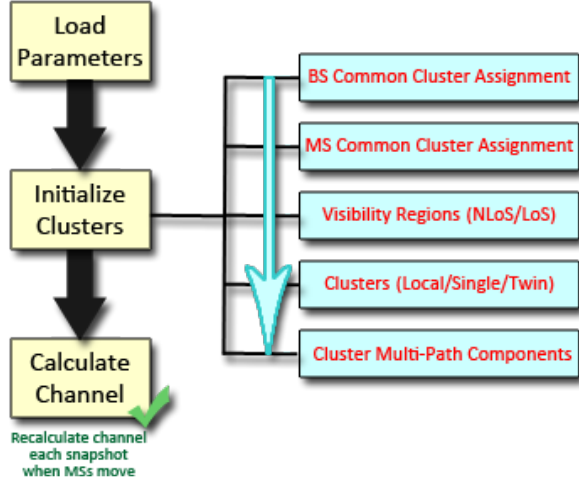


Fig. 3.1. Module Structure Flow of the Model in C++ Implementation.

- Step 1. Load and calculate all the necessary preset parameters for the parameter-oriented COST 2100 channel model [12];
- Step 2. Generate the common cluster distribution to support the multi-link extension;
- Step 3. Generate the possible VRs according to the possible cluster commonness;
- Step 4. Generate the clusters related to the VRs;
- Step 5. Generate the MPCs for each cluster with the preset MPC number;
- Step 6. Calculate the DDIR for each possible MPC if it's active;
- Step 7. Sum up all the DDIRs and calculate the channel matrix;
- Step 8. Due to the movement of the MS at different snapshots, check whether the MPCs is still valid in the local MS cluster, if not, generate a new MPC and update the local cluster;
- Step 9. Repeat Step 6 to 8 to calculate the channel for all snapshot.

With the channel matrix got from the module, some other detailed simulations based on this channel model can be done, and this will be discussed later in the next chapter.

### 3.2.3 Class Design

To achieve the module functions mentioned above, the classes of the program are designed as following:

- **class Channel\_Specification**

The main class, which simulates the channel model with the given parameters as the input data;

- **class BS\_Info**

Object class of BS, stores the information of all BSs participating the channel simulation, such as number of BSs, positions, VRs, common cluster distributions and local clusters information;

- **class MS\_Info**

Object class of MS, stores the information of all MSs participating the channel simulation, such as number of MSs, positions, velocities, common cluster distributions and local clusters information;

- **class Cluster**

Object class of Cluster, stores all the necessary information of each cluster, i.e., cluster positions to BS / MS, delay spread, angular spread, shadow fading, link delay and the information of MPCs inside it.

- **class MPC**

Object class of MPC set, which is related to the cluster, includes the position and amplitude information of each single MPC among the whole MPC set.

- **class Transfer\_Function**

Object class of Transfer Function for one pair of BS / MS link at one single snapshot, includes the channel coefficients and the frequency bins. For MIMO case, the channel coefficients are divided into each link of every TX and RX antenna pair.

- **class Impulse\_Response**

Object class of Impulse Response for one pair of BS / MS link at one single snapshot, is similar to Transfer Function, and in time-delay domain.

### 3.2.4 Functions Implementation

#### 3.2.4.1 void getParamsFromXML(const char fileName[])

As there are quite a few parameters need to be preset, for easy deployment, an independent configuration file of xml format with the defined "file-Name" is used in the project, which enables users to change possible simulation environments without rebuilding the whole module. Thus, the parameters in the configuration file should follow the rules strictly in order to avoid any possible exceptions [5].

By parsing the XML-format configuration file and loading the preset parameters as described in Step 1, all the preset variables of the program are set or calculated according to corresponding formulas. The entity structure of the xml file is shown as follows:

- **Root Node** <ChannelSpecification>
  - *BS Node* <BSInfo>
    - \* <position>
    - \* <commonRatio>
    - \* <antennaPattern>
  - *MS Node* <MSInfo>
    - \* <position>
    - \* <velocity>

- \* <commonCluster>
- \* <antennaPattern>
- *Channel Node* <Channel>
  - \* <profile>
  - \* <frequency>
  - \* <snapRate>
  - \* <snapNum>
  - \* <overSampleRate>
  - \* <snapRbandTypeate>
  - \* <external>
    - <bsHeight>
    - <msHeight>
    - <cellRadius>
    - <rooftopHeight>
    - <buildingSeparation>
    - <roadOrientation>
    - <roomSize>
    - <floorNum>
  - \* <stochastic>
    - <vrRadius>
    - <trRadius>
    - <clusterPower>
    - <excessDelay>
    - <cutoffDistLOS>
    - <vrRadiusLOS>
    - <factorLOS>
    - <singleClusterRatio>
    - <averageLocalCluster>
    - <averageFarCluster>
    - <activeLocalCluster>
    - <mpc>
    - <diffuseRadiation>
    - <shadowing>

```

· <spread> [<delay> <AoD> <EoD> <AoA> <sprEoAead>]
· <crossCorrelation>
· <cluster> [<azimuth> <elevation> <distance> <linkDelay>]
· <polarization>
· <dmc>

```

#### 3.2.4.2 void init\_BSCC() / void init\_MSCC()

For multi-link extension as in Step 2, we need to assign the BS / MS common clusters according to the BS / MS common ratios given in the preset parameters beforehand. With reference to the idea of common clusters in Chapter 2, we take the algorithms below to finish the assignment.

##### BSCC Generation

When the BS number is  $N_{BS}$ , the average VR density is  $P_{VR}$  and the BS common ration given:

1. Generate a temporal VR slots table of length  $N_{BS} \times P_{VR}$ ;
2. Calculate the number of  $K$ -BSCC ( $K$ -common VRs),  $N_{KBSCC}$ , with the BS common ratio;
3. Assign the common VRs from  $K = N_{BS}$  to  $K = 2$  sequentially, start with the first VR slot;
4. Randomly assign the left VRs to BSs until each of them reach  $P_{VR}$  VRs when  $K = 1$ .
5. Truncate the unused temporal VR slots to the length of real existing VRs  $N_{VR}$ .

##### MSCC Generation

When the number of existing VRs  $N_{VR}$  and average MS common cluster grouping factor  $q$  given:

1. Generate a VR slots table of length  $N_{\text{VR}}$ ;
2. Generate the poisson random number of cluster grouping  $L$  with  $q$  for a MSCC;
3. From the VR slots table, randomly choose  $L$  VRs and group them to the current MSCC;  
(If the length of the left available VR slots  $N_{\text{VR-AVL}} < L$ , just assign them all to the current MSCC)
4. Remove the grouped VRs from the VR slots table;
5. Repeat for the next MSCC from Step 2 until all the VRs are assigned.

### 3.2.4.3 void init\_VR()

After common cluster allocation, by the concept of the VR, we get the positions of all the VRs, including the LoS case as described in Step 3. With BSCC known, every VR position is then randomly generated with the positions of all the BSs that can see it.

### 3.2.4.4 void init\_cluster()

With common cluster assignment and VRs generated, clusters can be generated as in Step 4. Cluster position is located according to the reference BS and VR position; once the cluster position is known, the distance from BS / MS to the cluster  $d_{\text{C,BS/MS}}$  can be calculated, then the angular spread can also be derived [11].

### 3.2.4.5 void get\_MPC()

For each cluster, we need to get its related MPCs. MPC scatterers are positioned randomly Gaussian in the cluster. We calculated the MPCs information for each cluster as in Step 5. Rayleigh fading is taken to account the attenuation of the MPCs.



### 3.2.4.6 void init\_channel()

When all the preparations of channel modeling are done, as from Step 6 to Step 8, we just generate the channel simulation samples at every snapshot point for each pair of links between the BS and MS. From all the visible VRs and cluster for each BS, activeness is validated, if the cluster is active, then all the MPCs in that cluster is counted for channel response. Also, **function update\_MS()** here is used to check and update the information of the MS positions and local clusters while MS is moving during the whole process of channel response calculation. The channel is created in form of both DDIR and Channel Matrix (Transfer Functions).

#### DDIR

For each single MPC in the cluster, with its corresponding position and amplitude attenuation, both azimuth and elevation angles at both BS and MS can be calculated from the coordinate transformation as well as the total delay.

#### Channel Matrix

The channel matrix is a result of the sum of all DDIRs, together with the impact of the antenna radiation patterns. It can be presented in the form of transfer functions in the frequency domain or impulse responses in the delay domain [18].

For impulse response, according to the sample rate, we have the delay resolution and then group the MPCs of different delays into delay bins. In each delay bin, all the possible impulse responses introduced by those MPCs are added up and we get the  $h(t, \tau)$ .

For transfer functions, frequency range is divided into certain length of frequency bins, and a Fourier transformation of impulse response gives that transfer function, i.e., in our case:

$$H(t, f) = a_1 \exp[-j2\pi f \tau_1] + a_2 \exp[-j2\pi f \tau_2] + \cdots, \quad (3.1)$$

where  $a_i$  is the complex amplitude.

Since it's a MIMO channel model, antenna calibration is counted also to get the final channel matrix. If the antenna measurements data is provided, loaded as an IT-file, AoD / AoA and EoD / EoA of each MPC to TX / RX are required to get the antenna responses. To grab the values wanted, there should be a rotate transformation of the angles to get the relative useful angular index in the measurements data.

### **3.2.4.7 `get_all_DDIRs()` / `get_all_transfer_functions()`**

This is the getter functions to output the simulated channel model to be used in the specified transmission system. Return the final result of the channel realization for other systems using this channel model module.

## **3.3 Conclusions**

The architecture is designed to be a pluggable module interface with all the preset parameters for various scenario environments in a stand-alone configuration file to fix in routines different kinds of system simulation. The module however is able to customized to work for the simulation system which will be discussed in the next chapter. Besides it may also work as an add on to the existing IT++ library.

In our case, all the classes are packaged under the “penux” so as to have a smoother test in our simulation system, while the fundamental routines of the IT++ namespace are used extensively throughout with no doubts.

## Chapter 4

# Verifications and System Simulation Results

*With all the concepts and theories above, we have the COST 2100 channel model implemented, in this chapter, several tests and a simulation in the system level that were carried out to verify the module functions and show its accesses to the system performance evaluation. The unit test flow is illustrated for the stand-alone module verifications; for the system simulation, a brief introduction to the simulation system is provided up ahead, and it's followed by a system simulation with the Alamouti Transmission scheme. A brief summary of the results is given with conclusions in the end, to ensure that our implementation framework of COST 2100 channel model is workable.*

### 4.1 Module Verifications

In Chapter 3, it's clearly listed that the whole module structure is divided into several unit stages, and each stage consists of the required information for the next several stages, thus, the unit tests of the module go also along this guideline.

### 4.1.1 Unit Tests Flow

The unit tests flow of the module can be separated into steps as following:

- Step 1: Check the parameter loading unit: list all the preset parameters and parameters used in the simulation generated from the loaded values and make sure they are all in correct formats. Test exception reports if exit invalid entries in the configuration file.
- Step 2: Check the common cluster assignment: with the parameters needed, test whether the generated assignment follows the required preset common rate.
- Step 3: Check the VRs: validate all the random generated VRs are of reasonable positions. Test performance of the loop of valid VR re-generation.
- Step 4: Check the clusters: first ensure the random parameters initialization of the clusters, and avoid any unreasonable results of the cluster generation.
- Step 5: Check the MPCs: verify all the MPCs are generated with correct positions and amplitudes.
- Step 6: Check the channel initialization: first check the cluster activeness judgement and the result DDIR values; then check the LoS case; finally check the antenna response calculation and the final channel matrix.
- Step 7: Check the channel update due to the movement of MS: make sure all the recalculated MPCs are of valid values.

#### 4.1.1.1 Unit Test for `void init_BSCC()` / `void init_MSCC()`

Among all the functions, the algorithm of common cluster allocation is not standardized, thus, an individual test for this function is taken in case of any possible errors.

**Multi-BS case of 2, 3, and 4 BSs:**

1. For 2 BSs with the common ratio of 0.7 and 0.3:  
In all it's 467 VRs;  
1st BS: 303 VRs visible;  
2nd BS: 304 VRs visible;  
VRs in common: 140, i.e.,  $140/467 = 0.2998 \approx 0.3$ .
2. For 3 BSs with the common ratio of 0.5, 0.3 and 0.2:  
In all it's 1611 VRs;  
1st BS: 913 VRs visible;  
2nd BS: 912 VRs visible;  
3rd BS: 913 VRs visible;  
2 VRs in common: 483, i.e.,  $483/1611 = 0.2998 \approx 0.3$ .  
3 VRs in common: 322, i.e.,  $322/1611 = 0.1999 \approx 0.2$ .
3. For 4 BSs with the common ratio of 0.35, 0.25, 0.1 and 0.3:  
In all it's 517 VRs;  
1st BS: 304 VRs visible;  
2nd BS: 304 VRs visible;  
3rd BS: 304 VRs visible;  
4th BS: 303 VRs visible;  
2 VRs in common: 129, i.e.,  $129/517 = 0.2495 \approx 0.3$ .  
3 VRs in common: 52, i.e.,  $52/517 = 0.1006 \approx 0.2$ .  
4 VRs in common: 155, i.e.,  $155/517 = 0.2998 \approx 0.2$ .

**Multi-MS case of 4, 6 and 10 MSs grouping:**

1. Average common cluster group of 4:  
In all it's 517 VRs, and 129 clusters,  
i.e.,  $517/129 = 4.0078 \approx 4$ .
2. Average common cluster group of 6:

In all it's 1554 VRs, and 262 clusters,  
i.e.,  $1554/262 = 5.913 \approx 6$ .

3. Average common cluster group of 10:

In all it's 1034 VRs, and 101 clusters,  
i.e.,  $1034/101 = 10.2376 \approx 10$ .

From the test results, we can find the generated common ratio of the assignment is very close to what is required, due to the inaccuracy of random functions when generating the stochastic parameters according to the generic model, such as Gaussian distribution or Poisson distribution. By applying a more general statistics test with more samples, it shows the same result, which implies the validity of the algorithm.

### 4.1.2 Module Overall Tests

With all the single units tested, an overall module test is required to compare the result of the IT++ implementation to the MATLAB version. By taking the same set of channel preset parameters for the simulations performed and observing the PDDP of the generated channel, the result is proposed as following:

#### Power Delay Profile

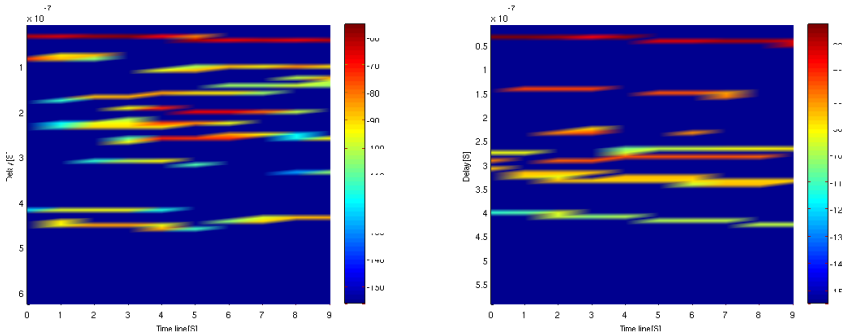


Fig. 4.1. Power Delay Profile: a) in MATLAB, b) in C++.

Fig. 4.1 shows an example of the simulated power delay profile (PDP) of the single channel realizations, where it's clearly shown at low delay time, power is higher. With the delay increases, the channel power is decreased as expected. Clearly the dominant power in both simulations are nearly the same since it is determined by the LoS component. The other scatterers are determined by the random process, which is changed in different simulation process. The results reflected by the C++ implementation are well matched with the results in MATLAB.

### Power Direction Profile

Both the azimuth and elevation power direction profiles at BS / MS of the single channel realizations are shown in Fig. 4.2 to 4.5. For the azimuth power profile, the angle varies randomly within a certain angle range for both the C++ and MATLAB implementation. The randomness of the angle distribution is determined by the randomly placed scatterers, which are different between each C++ / MATLAB simulation. For the elevation profile, the angle remains almost constant around  $90^\circ$  in both cases.

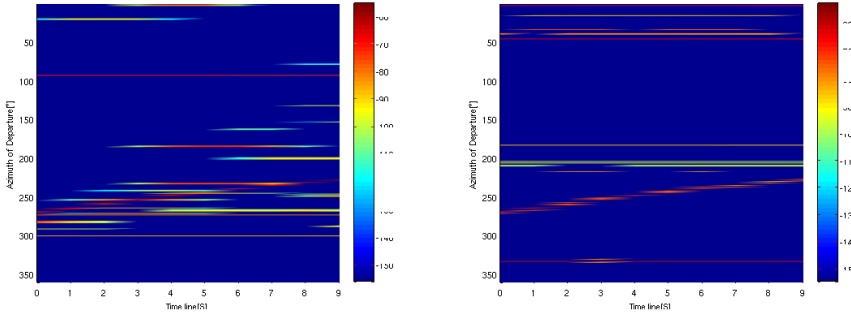


Fig. 4.2. Azimuth Power Direction Profile [BS]: a) in MATLAB, b) in C++.

All the results from the simulation given by C++ implementation are agreed with the MATLAB plots quite well respectively. Thus we can conclude the C++ implementation is of proper simulation results and the randomization

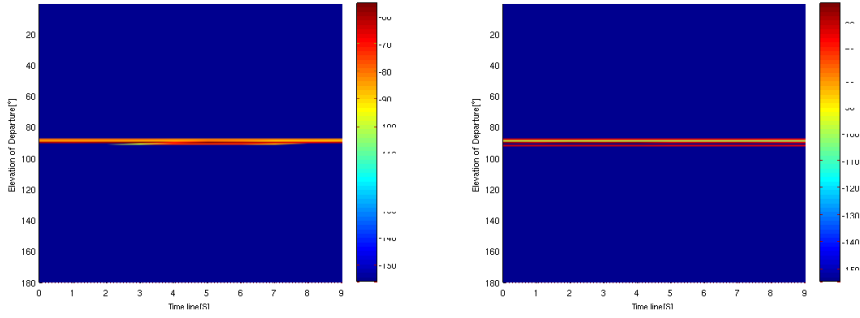


Fig. 4.3. Elevation Power Direction Profile [BS]: a) in MATLAB, b) in C++.

of the parameters leads to the difference between every single run, but we may still find the commonness of each run.

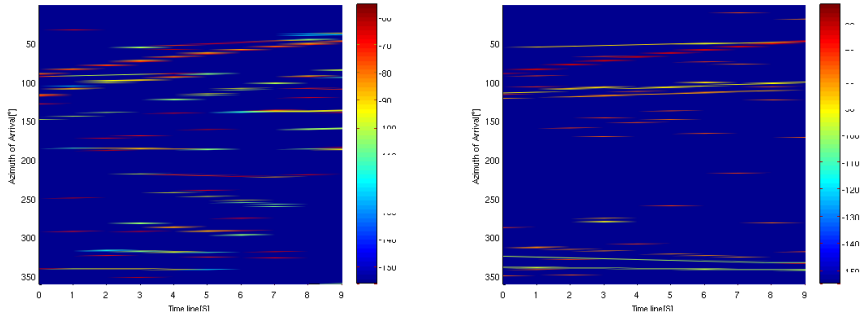


Fig. 4.4. Azimuth Power Direction Profile [MS]: a) in MATLAB, b) in C++.

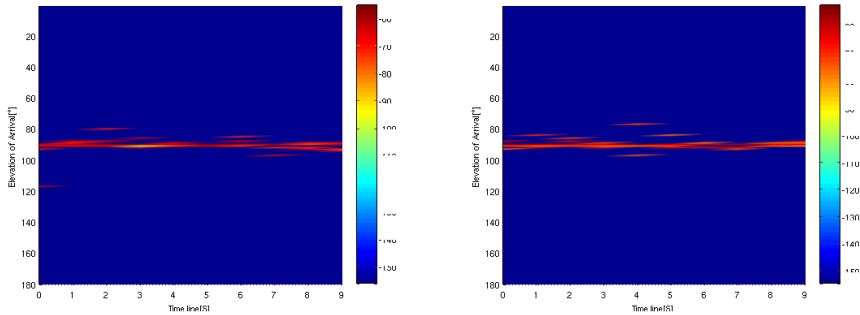


Fig. 4.5. Elevation Power Direction Profile [MS]: a) in MATLAB, b) in C++.

For a more accurate comparison between the C++ implementation and



MATLAB version, the statistics of the simulations is studied. With 100 re-



Fig. 4.6. Average Power Delay Profile Comparison.

alizations of the simulation for both implementations, the average PDP is calculated and plotted as shown in Fig. 4.6. It indicates the same trend for both versions: high power at low delay, and the power drops with increasing delays. The fading dips at high delays could be different, but the difference would be smaller if the averaging calculation is performed on a larger sample base.

Besides the CDF plots of power, delay, AoD / EoD and AoA / EoA are listed below, which also provide more evidence to show the correctness of the analysis above as well as the channel model implementation in C++.

## 4.2 System Simulations

The system simulation of the COST 2100 channel model implementation is carried out based on the RS-Simulation Chain development by EIT of LTH,

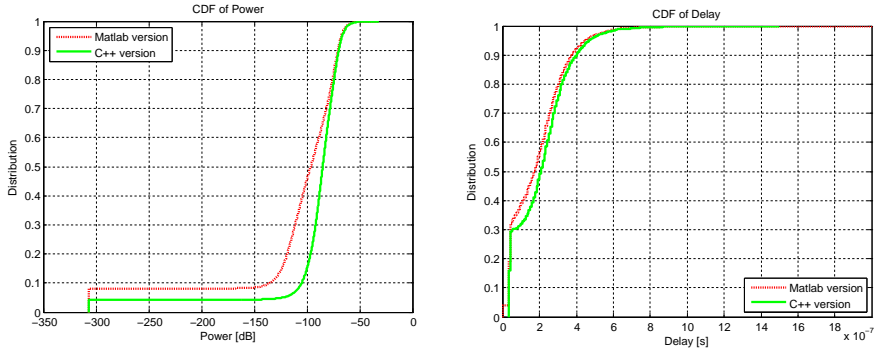


Fig. 4.7. Power and Delay Distribution (CDF).

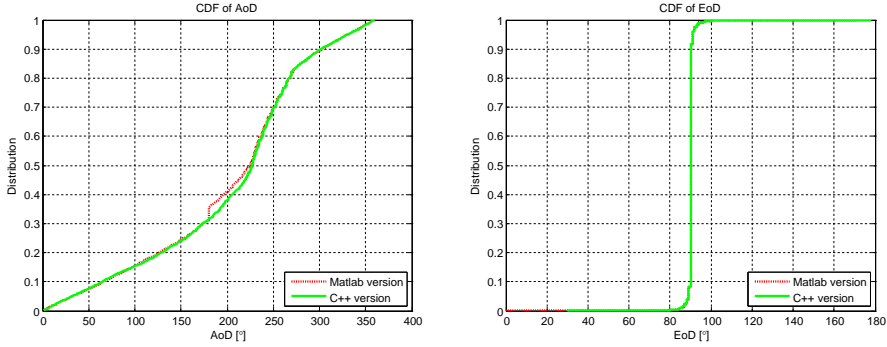


Fig. 4.8. AoD / EoD Distribution (CDF).

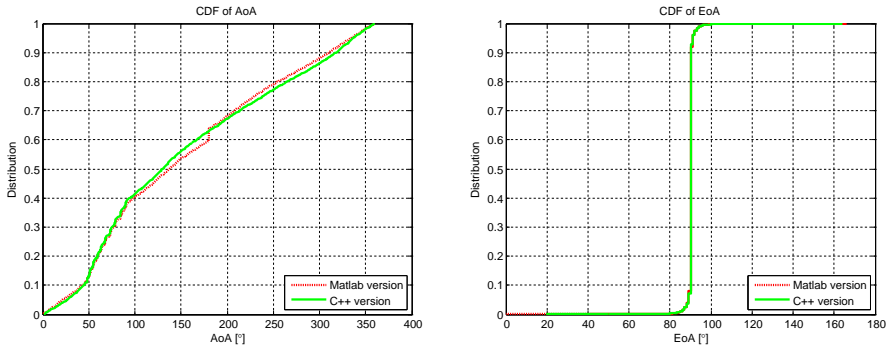


Fig. 4.9. AoA / EoA Distribution (CDF).

Sweden. The channel model module is used for a MIMO system transmission simulation, here in the project, a MIMO transmission simulation using Alamouti Transmission Scheme is adopted for the system verification of the model.

### 4.2.1 Introduction to the RS-Simulation Chain

RS-Simulation Chain is a integrated communication simulation system using C++, IT++ and perl, with transmitter, channel and receiver all included. Each stage of the communication is modeled into blocks with specified functions, together with a global general controller, managing the whole process of the simulation.

The blocks in the system is inherited from a virtual class `iProcessBlock` with input and output ports. A general realization of the interface is carried out by class `BasicBlock`, and all the detailed blocks are the sub classes of it. All the necessary blocks are added to the simulation environment and connected between each other with proper ports. Here shows the block diagram of the system flow in Fig. 4.10.

The transmitter part has different blocks of data generation, coding, modulation and etc., and they altogether provide the data stream to go through the channel. In the channel block, in which the COST 2100 channel model module is placed, a realization of the channel model is created in this block and the input data is processed with channel properties given in the channel matrix generated and also the noise. At the receiver, the channel is estimated and received data is sampled and demodulated. The controller traces and schedules the whole running process, and finally the BER is calculated to evaluate the system performance.

### 4.2.2 Introduction to Alamouti Transmission Scheme

For a MIMO channel model, Alamouti scheme is chosen in our simulation. Alamouti's code is the simplest of all the Space-Time Block Codings (STBCs) [7]. In the Alamouti scheme, which is a system of two transmit antennas, i.e.,  $M_{TX} = 2$ , two different symbols  $s_1$  and  $s_2$  are transmitted simultaneously from antennas 1 and 2 respectively during the first symbol period, followed by signals  $-s_2^*$  and  $s_1^*$  from antennas 1 and 2 respectively during the next symbol period, or equivalently the coding

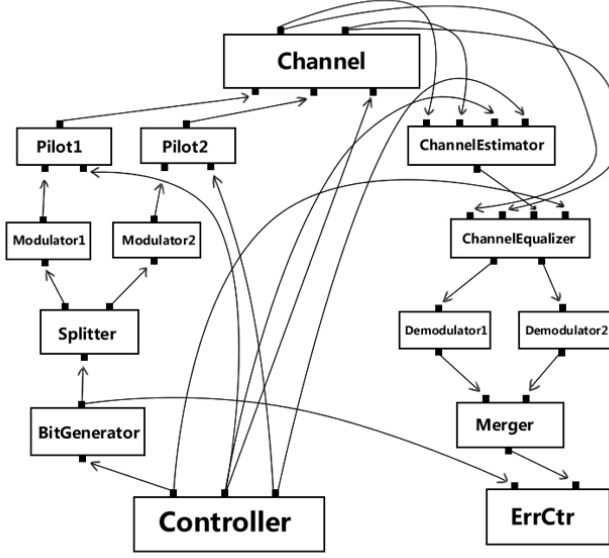


Fig. 4.10. System Flow of RS-Simulation Chain

matrix is

$$C_2 = \begin{bmatrix} c_1 & c_2 \\ -c_2^* & c_1^* \end{bmatrix}.$$

Assume that the channel remains constant over consecutive symbol periods and the channel is also frequency flat, for a  $2 \times 2$  MIMO system, the channel matrix  $\mathbf{H}$  is

$$\mathbf{H} = \begin{bmatrix} h_{1,1} & h_{1,2} \\ h_{2,1} & h_{2,2} \end{bmatrix},$$

then the received signals at the receive antenna array over consecutive symbols periods are

$$\begin{aligned} \mathbf{y}_1 &= \sqrt{\frac{E_s}{2}} \begin{bmatrix} h_{1,1} & h_{1,2} \\ h_{2,1} & h_{2,2} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}, \\ \mathbf{y}_2 &= \sqrt{\frac{E_s}{2}} \begin{bmatrix} h_{1,1} & h_{1,2} \\ h_{2,1} & h_{2,2} \end{bmatrix} \begin{bmatrix} -s_2^* \\ s_1^* \end{bmatrix} + \begin{bmatrix} n_3 \\ n_4 \end{bmatrix}, \end{aligned} \quad (4.1)$$

where  $n_1, n_2, n_3$  and  $n_4$  are uncorrelated ZMCSCG noise samples with  $E\{|n_i|^2\} =$

$N_0(i = 1, 2, 3, 4)$ . Then the receiver forms a signal vector  $\mathbf{y}$  according to

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2^* \end{bmatrix} = \sqrt{\frac{E_s}{2}} \begin{bmatrix} h_{1,1} & h_{1,2} \\ h_{2,1} & h_{2,2} \\ h_{1,2}^* & -h_{1,1}^* \\ h_{2,2} & -h_{2,1}^* \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \\ n_3^* \\ n_4^* \end{bmatrix} = \sqrt{\frac{E_s}{2}} \mathbf{H}_{eff} \mathbf{s} + \mathbf{n}, \quad (4.2)$$

] where  $\mathbf{H}_{eff}$  is orthogonal irrespective of the channel realization. Then the receiver forms the antenna outputs  $\mathbf{z}$  of the effective channel for data symbols  $\mathbf{s}$ ,

$$\mathbf{z} = \mathbf{H}_{eff}^H \mathbf{y} = \sqrt{\frac{E_s}{2}} \|\mathbf{H}\|_F^2 \mathbf{I}_2 \mathbf{s} + \tilde{\mathbf{n}}, \quad (4.3)$$

and for either data symbol  $s_i (i = 1, 2)$ , the antenna output is

$$z_i = \sqrt{\frac{E_s}{2}} \|\mathbf{H}\|_F^2 s_i + \tilde{n}_i, (i = 1, 2), \quad (4.4)$$

where  $\|\cdot\|_F$  is the Frobenius norm.

The Alamouti scheme is the only orthogonal STBC that achieves rate-1, i.e., it can achieve its full diversity gain without needing to sacrifice its data rate. This property also gives Alamouti's code a significant advantage over the higher-order STBCs.

### 4.2.3 Simulation Results

The channel model is integrated into RS-Simulation Chain for simulations on the system level. A simple system level simulation for the single-link case is adopted with the simple Alamouti transmission scheme. Here a  $2 \times 2$  indoor system is used, the frequency range is from 2 GHz to 2.02 GHz, and the antenna pattern is  $7 \times 7$  in circular array.

The simulated channel has 667 frequency bins,  $1 \times 1$  SISO case and  $2 \times 2$  MIMO case are the subset of the whole channel matrix. The channels are normalized, and used in the Channel block of the system flow shown in Fig. 4.10. The transmission system uses BPSK modulation, and the Alamouti coding is done with the Modulation block before adding the Pilots. At the receiver side, the ChannelEqualizer for MIMO case does the Alamouti decoding with the estimated channel from the ChannelEstimator.

The bit-error-rate (BER) performance is shown in Fig. 4.11, and the Alamouti scheme achieves an obvious diversity gain in this simulation case. For different channel

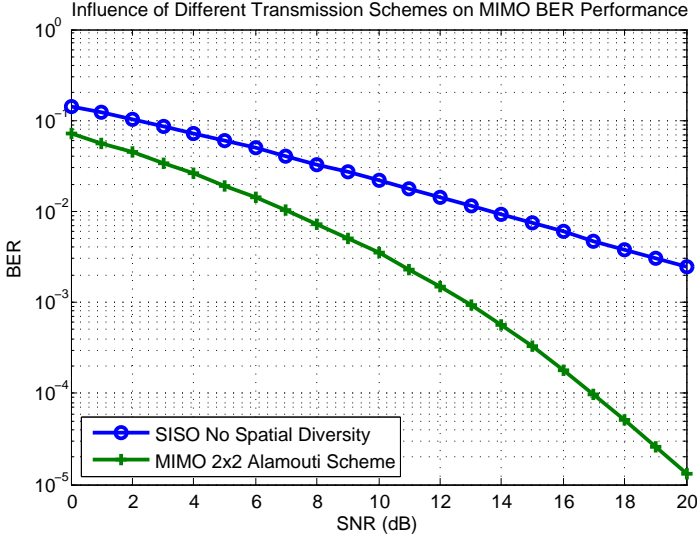


Fig. 4.11. Channel Model BER Performance in Different Transmission Scheme.

realizations, as described before, the randomization of the channel parameters can cause different diversity orders, but they are all within a reasonable range.

Besides RS-Simulation Chain, another simulation verification based on the Walati Transmission System is also performed. The result also shows that this COST 2100 channel model module can be integrated in the system properly.

### 4.3 Conclusions

All the tests and simulations above were implemented step-wisely, besides the unit tests were carried out through this thesis project. During the simulation tests, problems and bugs were also found in the MATLAB implementation and this IT++ implementation also helped the correction.

According to the simulation results, it can be believed that the COST 2100 channel model module implemented in IT++ / C++ is of good validity and application use.

## Chapter 5

# Final Conclusions

*As mentioned in the first chapter, the main object of this thesis project is to implement a IT++ / C++ COST 2100 channel model module for better application and simulation experience. but at the start, as the channel model is not developed completely, it was hard to estimate the amount of work and possible updates of the implementation. Finally, the design and implementation module framework was done with several simulations. This chapter summarize the work and discuss about the possible future works.*

### 5.1 Summaries and Conclusions

With the results from the tests and simulations, it can be believed that the goal of the project is achieved and the COST 2100 channel model module in C++ with IT++ support works fine and returns reasonable results. The source code package is available at <http://www.channelmodels.org/tiki-index.php>.

The implementation of COST 2100 channel model provides me a deeper understanding of the channel modeling and gives me a clear idea on how this model is defined and simulated. Besides COST 2100, knowledge about the previous related actions, such as COST 259 and COST 273, are also gained from this project.

By programming with IT++ / C++, many built-in functions in MATLAB are still not supported, hence, extra work in utility functions realization have been done as well. Besides, the data structure in C++ and MATLAB is of great difference, especially for these multi-dimensional matrices for channel information storage, but still the problem was solved as C++ is a Object Oriented Programming (OOP)

language and customized objects were designed to make up for this weakness in the IT++ library.

In addition, the system simulation with the RS Simulation Chain helped me to understand the whole flow of communication systems, and it's also a great learning experience to go through the structure and details of the simulation chain.

## 5.2 Future Works

For the COST 2100 channel model is not finally standardized, there will still be possible updates in the channel model itself, and this IT++ / C++ implementation is of both capacity and simplicity for those future updates.

In this version, the following features are not included, but it's considered in the framework design for the future:

- Spatial autocorrelation due to the movement of BS;
- Polarization effects;
- Diffuse scattering components;
- Dense Multi-path Components(DMC) [13].

One more thing, for COST 2100 channel model is a MIMO channel, how to import the different antenna configurations of the specific system and make it compatible for system simulations still needs some future works beyond the current implementation.



# Appendix A

## External Parameters

### External parameters for all environments

- Carrier frequency  $f_c$ : [Hz]  
For most cellular applications the carrier frequency is in the 2 GHz band; and for picocell and ad hoc applications (wireless LAN, etc.) the 5 GHz band is more common;
- Base station height  $h_{BS}$ : [m]  
For macro- and microcellular environments, height signifies the height above ground, and for picocells, it denotes the height above the floor the MS is in, negative heights imply that the BS is on a lower floor;
- Mobile station height  $h_{MS}$ : [m]  
For outdoor environments,  $h_{MS}$  signifies the height above street level, and is typically 1.5 m when the MS is at street level; while for outdoor-to-indoor applications,  $h_{MS}$  can be much larger, and for picocellular applications,  $h_{MS}$  signifies the height over the floor the user is on, and is therefore always 1.5 m, and for ad hoc applications, the height of the MS is 0.8 m (tabletop height);
- Base station position  $\vec{r}_{BS}$ : [m]  
The distance of the BS from the origin of the coordinate system, *e.g.*  $(0, 0, h_{BS})$ ;
- Mobile station position  $\vec{r}_{MS}$ : [m]  
The distance of the MS from the origin of the coordinate system, *e.g.*  $(0, 0, h_{MS})$ ;
- Antenna configurations  
Antenna number, antenna spacing, antenna pattern, antenna orientation;
- Path-loss model  $P_L$ : [dB/m]  
Path-loss depends on the distance between BS and MS, i.e.,  $|\vec{r}_{BS} - \vec{r}_{MS}|$ .

### Additional external parameters for macro- and microcells

- Average rooftop height  $h_B$ : [m]  
Suggested values are 15 m for typical urban, 30 m for bad urban; and 8 m for suburban;
- Width of roads  $w_r$ : [m]  
Suggested values are 15 m for typical urban, and 25 m for bad urban;
- Distance between buildings  $w_b$ : [m]  
Suggested values are 25 m for typical urban, and 100 m for bad urban;
- Road orientation with respect to direct path  $\varphi_R$ : [degree]  
Suggested value is  $45^\circ$ .

### **Additional external parameters for picocells and ad hoc networks**

- Room size  $l_l \times l_w$ : [m×m]  
For residential environments,  $3 \times 5 \text{ m}^2$  is recommended; for office environments with separate offices,  $2 \times 4 \text{ m}^2$ ; for offices with cubicle environments  $10 \times 10 \text{ m}^2$ ; and for halls  $50 \times 20 \text{ m}^2$ ;
- Number of floors between BS and MS  $N_{\text{floor}}$ : [integer]  
For standard situations, this number is 0, i.e. BS and MS are on the same floor;
- Building opposite to the building of BS and MS: [yes/no]  
Assume "Yes" by default.

# Appendix B

## Stochastic Parameters

### Stochastic parameters for macrocells

- Visibility Region:  
Poisson distributed;  $R_C = 100[\text{m}]$ ;  $L_C = 20[\text{m}]$ .
- Cluster movement:  
No cluster movement.
- Cluster power:  
 $k_\tau = 1[\text{dB}/\mu\text{s}]$ ;  $\tau_B = 10[\mu\text{s}]$ .
- Line of sight:  
 $d_{\text{co}} = 500[\text{m}]$ ;  $R_L = 30[\text{m}]$ ,  $L_L = 20[\text{m}]$ ;  
 $\mu_K = (26 - EPL)/6[\text{dB}]$ ;  $\sigma_K = 6[\text{dB}]$ ;  
 $K_{\text{sel}} = 1$  (Only single interacting clusters).
- Average number of local clusters:  
 $N_C = 1.18$ .
- Average number of add. clusters:  
 $N_{C,\text{add}} = 1$  (MS).
- Number of MPCs per cluster:  
 $N_{\text{MPC}} = 20$ .
- Rice factor of additional clusters:  
 $K_{\text{MPC}} = 0$ .
- Diffuse radiation:  
 $\mu_{\text{diff}} = 0.05$ ;  $\sigma_{\text{diff}} = 3.4$ ;  
 $\text{PDP}_{\text{diff}}$ : uniform in azimuth,  $\exp(t/\tau)$ .
- Delay spread, goes with  $d^{0.5}$ :  
 $\mu_\tau = 0.4[\mu\text{s}]$ ;  $\sigma_\tau = 3[\text{dB}]$ .

- Angular spread at BS,  $\sigma_{\text{AS}} = 10^{\epsilon x + \mu}$ ,  $x = N(0, 1)$ :  
 $\mu_{\phi\text{BS}} = 0.81[\text{dB}]$ ;  $\sigma_{\phi\text{BS}} = 0.34[\text{dB}]$ ;  
 $\mu_{\theta\text{BS}} = 0.5[\text{dB}]$ ;  $\sigma_{\theta\text{BS}} = 3[\text{dB}]$ .
- Angular spread at MS:  
 $\mu_{\phi\text{MS}} = 35[\text{dB}]$ ;  $\sigma_{\phi\text{MS}} = 0[\text{dB}]$ ;  $\text{pdf}(\theta_{\text{MS}}) = [0, 45]$ .
- Shadowing:  
 $\sigma_{\text{S}} = 6[\text{dB}]$ .
- Autocorrelation distances:  
 $L_{\text{S}} = 100[\text{m}]$ ;  $L_{\tau} = 100[\text{m}]$ ;  
 $L_{\phi, \text{BS}} = 100[\text{m}]$ ;  $L_{\theta, \text{BS}} = 100[\text{m}]$ ;  
 $L_{\phi, \text{MS}} = 100[\text{m}]$ ;  $L_{\theta, \text{MS}} = 100[\text{m}]$ .
- Cross-correlations:  
 $\rho_{\tau-\phi\text{BS}} = 0.5$ ;  $\rho_{\tau-\phi\text{MS}} = 0$ ;  $\rho_{\tau-\text{S}} = 0.6$ ;  
 $\rho_{\phi\text{BS}-\phi\text{MS}} = 0$ ;  $\rho_{\phi\text{BS}-\text{S}} = 0.6$ ;  $\rho_{\phi\text{MS}-\text{S}} = 0.6$ ;  
 $\rho_{\text{BS}-\text{BS}} = 0.5$ .
- Polarization:  
 $\mu_{\text{XPD}} = 6[\text{dB}]$ ;  $\sigma_{\text{XPD}} = 2[\text{dB}]$ ;  
 $\mu_{\text{VVHH}} = 0[\text{dB}]$ ;  $\sigma_{\text{VVHH}} = -\infty$ ;  
 $\mu_{\text{VHHV}} = 0[\text{dB}]$ ;  $\sigma_{\text{VHHV}} = -\infty$ .

### Stochastic parameters for microcells

- Visibility Region:  
Poisson distributed;  $R_{\text{C}} = 50[\text{m}]$ ;  $L_{\text{C}} = 20[\text{m}]$ .
- Cluster movement:  
No cluster movement, but the MS moves.
- Cluster power:  
 $k_{\tau} = 40[\text{dB}/\mu\text{s}]$ ;  $\tau_{\text{B}} = 0.5[\mu\text{s}]$ .
- Line of sight (Values from 2 and 5 GHz LoS measurements):  
 $d_{\text{co}} = 300[\text{m}]$ ;  $R_{\text{L}} = 50[\text{m}]$ ,  $L_{\text{L}} = 50[\text{m}]$ ;  
 $\mu_{\text{K}} = 7[\text{dB}]$ ;  $\sigma_{\text{K}} = 2.3[\text{dB}]$ ;  
 $K_{\text{sel}} = 0.5$ .
- Average number of local clusters:  
 $N_{\text{C}} = 1$ .
- Average number of add. clusters:  
 $N_{\text{C,add}} = 3$ .
- Marginal DoA pdf:  
Equal, random user orientation.
- Marginal DoD pdf:  
Equal in  $[-\pi/8, \pi/8]$ .

- 
- Number of MPCs per cluster:  
 $N_{\text{MPC}} = 7$ .
  - Rice factor of additional clusters:  
 $K_{\text{MPC}} = 2[\text{dB}]$ .
  - Diffuse radiation:  
 Percentage of power in the diffuse radiation: 10%  
 $\text{PDP}_{\text{diff}} = -40[\text{dB}/\mu\text{s}]$ .
  - Delay spread:  
 $\mu_{\tau} = 13[\mu\text{s}]; \sigma_{\tau} = 14[\text{dB}]$ .
  - Angular spread at BS:  
 Same values as the angular spread at MS.
  - Angular spread at MS:  
 $\mu_{\phi\text{MS}} = 2.3[\text{dB}]; \sigma_{\phi\text{MS}} = 3.4[\text{dB}];$   
 $\mu_{\theta\text{MS}} = 1.3[\text{dB}]; \sigma_{\theta\text{MS}} = 3.3[\text{dB}]$ .
  - Shadowing:  
 $\sigma_{\text{S}} = 2.9[\text{dB}]$ .
  - Autocorrelation distances:  
 $L_{\text{S}} = 5[\text{m}]; L_{\tau} = 5[\text{m}];$   
 $L_{\phi,\text{BS}} = 50[\text{m}]; L_{\theta,\text{BS}} = 50[\text{m}];$   
 $L_{\phi,\text{MS}} = 25[\text{m}]; L_{\theta,\text{MS}} = 25[\text{m}]$ .
  - Cross-correlations:  
 $\rho_{\tau-\phi\text{BS}} = 0.1; \rho_{\tau-\phi\text{MS}} = 0.1; \rho_{\tau-\text{S}} = 0.04;$   
 $\rho_{\phi\text{BS}-\phi\text{MS}} = 0; \rho_{\phi\text{BS}-\text{S}} = -0.2; \rho_{\phi\text{MS}-\text{S}} = -0.2$ .
  - Polarization:  
 $\mu_{\text{XPD}} = 8.5[\text{dB}]; \sigma_{\text{XPD}} = 1.8[\text{dB}];$   
 $\mu_{\text{VVHH}} = 0.3[\text{dB}]; \sigma_{\text{VVHH}} = 3.2[\text{dB}];$   
 $\mu_{\text{VHHV}} = -0.5[\text{dB}]; \sigma_{\text{VHHV}} = 1.8[\text{dB}]$ .

### Stochastic parameters for picocells

- Visibility Region:  
 Poisson distributed.
- Cluster movement:  
 No cluster movement.
- Cluster power:  
 $k_{\tau} = 50 \sim 100[\text{dB}/\mu\text{s}]$ .
- Line of sight:  
 $K_{\text{sel}} = 0$  (Only multiple interacting clusters).

# References

- [1] About COST 2100. (Online).  
Available: <http://www.COST2100.org/index.php?page=about-COST2100>. 2
- [2] European Cooperation in Science and Technology. (Online).  
Available: <http://www.cost.esf.org/>. 1
- [3] GNU General Public License. (Online).  
Available: <http://www.gnu.org/licenses/gpl.html>. 24
- [4] IT++ Documentation. (Online).  
Available: <http://itpp.sourceforge.net/current/>. 24
- [5] Reference Manual for libxml2. (Online).  
Available: <http://xmlsoft.org/html/index.html>. 27
- [6] R. HEDDERGOTT M. STEUNBAUCER A. F. MOLISCH, H. ASPLUND AND T. ZWICK. The COST 259 directional channel model - I. Philosophy and General Aspects. *IEEE Trans. Wireless Comm.*, 5(12), December 2006. 10, 12
- [7] R. NABAR A. PAULRAJ AND D. GORE. *Introduction to Space-Time Wireless Communications*. Cambridge University Press, 2003. 8, 41
- [8] L. CORREIA. *Mobile Broadband Multimedia Networks*, pages 365–367. Academic Press, June 2006. 7
- [9] N. CZINK AND C. OESTGES. The COST 273 MIMO Channel Model: Three Kinds of Clusters. *IEEE Trans. Wireless Comm.*, September 2008. 11
- [10] A. F. MOLISCH K. I. PEDERSEN H. ASPLUND, A. A. GLAZUNOV AND M. STEUNBAUCER. The COST 259 directional channel model - II. Macrocells. *IEEE Trans. Wireless Comm.*, 5(12), December 2006. 11
- [11] A. F. MOLISCH H. HOFSTETTER AND N. CZINK. A Twin-Cluster MIMO Channel Model. In *Proc. ‘EuCAP 2006’*, Nice, France, 6-10 November 2006. 30

- 
- [12] P. VAINIKAINEN LINGFENG LIU C. OESTGES J. POUTANEN, K. HANEDA AND F. TUFVESSON. COST 2100 TD(10) 10067: Inputs for the COST 2100 Multi-link Channel Model. In *EURO-COST*, Athens, Greece, 3-5 February 2010. 25
  - [13] V. M. KOLMONEN P. VAINIKAINEN F. TUFVESSON LINGFENG LIU J. POUTANEN, K. HANEDA AND C. OESTGES. COST 2100 TD(10) 11033: Adding Dense Multipath Components to the COST 2100 MIMO Channel Model. In *EURO-COST*, Aalborg, Denmark, 2-4 June 2010. 46
  - [14] K. HANEDA P. VAINIKAINEN F. TUFVESSON L. LIU, J. POUTANEN AND C. OESTGES. COST 2100 TD(10) 11012: A multi-link extension of the COST 2100 MIMO channel model. In *EURO-COST*, Aalborg, Denmark, 2-4 June 2010. 18
  - [15] L. LIU. A Public COST 2100 Channel Model MATLAB Source Code. (Online), August 2010. Available: [http://ftp.COST2100.org/WG2.3\\_Model/](http://ftp.COST2100.org/WG2.3_Model/). 22
  - [16] A. L. MOUSTAKAS P. J. SMITH A. F. MOLISCH F. TUFVESSON M. SHAFI, M. ZHANG AND S. H. SIMON. Polarized mimo channels in 3d: Models, measurements and mutual information. *IEEE JSAC*, **Special Issue on 4G Wireless Systems**, 2005. 17
  - [17] P. MOGENSEN AND T. KLINGENBRUNN. Modelling cross-correlated shadowing in network simulations. *VTC99*, pages 1407–1411, 1999. 17
  - [18] A. F. MOLISCH. *Wireless Communications*, pages 96–99. Wiley Press, 2005. 31
  - [19] A. RICHTER AND R. S. THOMA. Joint maximum likelihood estimation of specular paths and distributed diffuse scattering. *IEEE VTC 2005-spring*, 1, June 2005. 17