HELSINKI UNIVERSITY OF TECHNOLOGY
Department of Electrical and Communications Engineering

UNIVERSIDAD POLITÉCNICA DE MADRID
E.T.S.I. Telecomunicación

F. Javier Sanz

Wideband radio channel simulation using experimental data

Thesis submitted in partial fulfilment of the requirements for the degree of Master of Science in Engineering in Espoo, Finland.

________________________, 1998

Supervisor Professor Pertti Vainikainen

Instructor MSc Kimmo Kalliola

Helsinki University of Technology Universidad Politécnica de Madrid

Author: F. Javier Sanz

Name of the thesis: Wideband Radio Channel Simulation Using Experimental Data

Date: 16th November 1998 **Number of Pages:** 103

Department: Department of Electrical and Communications Engineering

Professorship: S-26 Radio Engineering

Supervisor: Professor Pertti Vainikainen

Instructor: MSc Kimmo Kalliola

In this Master's Thesis the simulation of the wideband radio channel using measured data is investigated and implemented. In radio interface simulations the radio channel is commonly implemented as a statistical model, often derived from measurement campaigns. In this work the methods and performance of using measured data in the simulations are studied, along with all the processing that has to be done to the measured data for adapting it to the simulation. This method is intended to be used for modelling the wideband radio channel in the study of WCDMA system, the multiple access technique chosen for the UMTS by ETSI.

The main tasks of this work were establishing the link between measured data and the simulation model, and comparing the results of this approach with the traditional ways of simulating the radio channel. The simulation tool used was COSSAP, widely used for link level simulations. The comparison has been done considering the duration of the simulations with both models, because the radio channel takes large part of the time of the overall system simulations.

The results observed were a significant reduction, more than 6 times, in the duration of the simulation when using experimental data, and a slight increment of simulation time (some seconds), compared to the statistical models (tens of seconds), when the number of taps or the speed of the mobile were increased. Another advantage of this modelling method is that it reproduces in a more realistic way the behaviour of the radio channel, because it is based on actual data, and not on statistics.

In this thesis a basic link between measured data and simulation model was studied. The future work for upgrading this approach will include the implementation of superresolution methods, the study of noise and interference implementation in the overall system simulation and methods for the interpolation of IRs.

Keywords: Impulse response, Wideband Multipath Radio Channel, Channel modelling, Sounder ,WCDMA.

Helsinki University of Technology Universidad Politécnica de Madrid

Autor:	Francisco Javier Sanz Virto		
Nombre del proyecto:	Simulación de Canales de Radio de Banda Ancha Mediante el Uso de Datos Experimentales		
Fecha:	16 Noviembre 1998	**Número de páginas:**	103
Facultad:	Departmento de Ingeniería Eléctrica y Comunicaciones (Universidad Tecnológica de Helsinki, Finlandia)		
Departamento:	Ingeniería de Radio		
Supervisor:	Profesor Pertti Vainikainen		
Instructor:	Ingeniero Superior Kimmo Kalliola		

Gran parte de las comunicaciones móviles actuales hacen uso de canales de radio de banda estrecha, pero el desarrollo de las mismas hacia los nuevos servicios como, entre otros, acceso a Internet o transmisión de video en tiempo real requerirán una velocidad de transmisión de datos mayor que la actual, y por lo tanto un ancho de banda mayor. Con este objetivo se están desarrollando la tercera y cuarta generación de telefonía móvil. Para el estudio de las diferentes técnicas de modulación, codificación, recepción y demás elementos presentes en un sistema de comunicación para estas nuevas tecnologías es imprescindible el uso de simulaciones. Uno de los elementos más importantes en estas simulaciones es el canal de radio, debido principalmente a su naturaleza variante e impredecible y a que consume el mayor tiempo de la simulación debido a su complejidad.

Tradicionalmente el modelado del canal de radio para los diferentes sistemas (como por ejemplo GSM o CDMA IS-95) se hace de una forma estadística. Estos modelos se establecen mediante la realización de largas campañas de medidas y posterior proceso y estudio de los datos obtenidos, sobre diferentes sistemas y entornos. El proceso anterior deriva en recomendaciones, tal y como ocurre en la recomendación GSM 05.05 o en el IS-95 para CDMA, en modelos para utilizar en los proyectos donde se llevan a cabo estas medidas, como ocurrió en los proyectos CoDiT y ATDMA o en modelos definidos por los cuerpos de standarización, como es el caso de los modelos de la ITU. De este modo se puede reproducir el canal de radio de una forma fácil, pero de estructura compleja, ajustada a los datos medidos y ser utilizado de este modo en las simulaciones.

En este proyecto se investiga otra forma de modelar el canal de radio de banda ancha: utilizar datos medidos en las simulaciones. Para ello es necesario someter a los datos medidos a una serie de procesamientos para adaptarlos a la simulación, tal y como será explicado posteriormente.

Con este modelo se pretende lograr las siguientes ventajas:

- Reproducción más realista del comportamiento del canal de radio
- Reducción del tiempo empleado en la simulación del canal de radio
- Fácil extensión del modelo a cualquier tipo de datos medidos, independientemente del sistema (GSM, CDMA IS-95 o WCDMA) o tipo de entorno (urbano, semi-urbano, montañoso, etc …)

Los datos utilizados para este estudio han sido obtenidos con el medidor de canales de radio de banda ancha, desarrollado en el Instituto de Radiocomunicaciones (IRC) de la Univesidad Tecnológica de Helsinki (HUT). Las Respuestas Impulsivas (RIs) del canal de radio son obtenidas tras someter los datos medidos a un proceso. Tras este proceso es necesario adaptar las RIs al simulador y al modelo de simulación. La característica principal del canal de radio de banda ancha es la existencia de multitrayecto, y por tanto de la llegada de replicas de la señal tranmitida con diferentes retardos y amplitudes provocados por los diferentes trayectos seguidos, sumándose todos para formar la RI del canal. Este efecto provoca que el canal pueda tener desvanecimiento y ser selectivo en frequencia.

Los estudios de este proyecto se han realizado sobre el simulador COSSAP, un simulador ampliamente utilizado para simulaciones a nivel de enlace, y el modelo para implementar el canal de radio de banda ancha está basado en un filtro de respuesta impulsiva finita (FIR), ampliamente utilizado para este fin. Los coeficientes de este filtro representan los diferentes retardos de la RI y sus valores la amplitud de los mismos. La naturaleza variante del canal es implementada actualizando cada cierto tiempo, dictado por los parámetros de la simulación y los de los datos medidos, los coeficientes del filtro que representan la RI.

La adaptación de los datos medidos experimentalmente se basa fundamentalmente en la obtención de las RIs a partir de los datos medidos, interpolación de RIs, muestreo de cada RI a una frequencia diferente, definición de los coeficientes que representan el retardo y cálculo de la frequencia de actualización de RIs en la simulación. A continuación la necesidad e implementación de cada uno de estos pasos serán resumidos.

Obtención de las RIs: Depende de la configuración del sistema de medición con el que se han obtenido los datos, si es mediante un correlador deslizante, los datos obtenidos representan las RIs, mientras que si es mediante muestreo directo han de someterse a un filtro adaptado a las sequencia transmitida. Con estos modos de funcionamiento se obtiene una serie inicial de RIs, cuya resolución para distinguir señales que llegan retardadas en instantes de tiempo muy próximos está limitada por el sistema de medición.

Por la razón anterior hay otros modos de obtener una serie inicial de RIs, con mayor resolución, a las que posteriormente se las someterá a una serie de complejos procesamientos para obtener la resolución requerida. Estas técnicas se conocen como métodos de superresolución.

Remuestreo: Cada RI obtenida está formada por una serie de muestras separadas cada una por un retardo que depende de la frequencia de muestreo del sistema de medida. En el modelo implementado (filtro FIR) esta separación está fijada por la frequencia de muestreo de los datos a simular (dado por el sistema a simular), por lo tanto es necesario volver a muestrear cada RI a la frequencia requerida, previo filtrado para evitar solapamiento.

Interpolación de RIs: Debido al sistema de medición utilizado se suele medir un número de RIs por unidad de espacio relativamente pequeño (generalmente dado en unidades de RIs por longitud de onda). Medir un número mayor de RIs haría la campaña de medición larga y tediosa. Por esta razón el resto de RIs necesarias son introducidas mediante interpolación sobre la medidas. Este proceso es también realizado en los modelos de canal estadístico, donde el número de RIs por longitud de onda generadas es constante y el número final de RIs, dependiente de la velocidad del móbil a simular, establecido mediante interpolación . Previos trabajos han demostrado que el número de RIs por longitud de onda usado en las simulaciones tiene un efecto importante en los resultados finales y que su valor mínimo necesario se sitúa cerca de las 30. Este parámetro influirá en el índice de actualización de las RIs, como se expondrá más adelante.

Definición de los coeficientes: Una vez seguidos los pasos anteriores los datos obtenidos podrían ser utilizados en la simulación, pero estaríamos considerando en ellos el ruido introducido por el canal y por el sistema de medida. Por lo tanto es necesario una previa definición de los coeficientes. Para ello se consideran dos posibles técnicas.

- Superresolución: La cual conlleva resultados buenos a costa de un complejo y largo proceso.
- Umbral de ruido: Modo más sencillo pero con peores resultados que el anterior, consistente en considerar las muestras que no superen el umbral como ruido. La dificultad radica en establecer cuál es el umbral adecuado.

Todos los procesamientos descritos hasta ahora se llevan a cabo en este trabajo en MATLAB, aunque cualquier otra herramienta similar puede ser utilizada. En este punto los datos están preparados para poder ser usados en la simulación, lo único que hay que hacer es convertirlos al formato del simulador, con este propósito un programa en lenguaje C ha sido implementado para convertir estos datos de MATLAB a COSSAP. También se ha creado en COSSAP el modelo de canal de radio, consistente en un bloque que lea las RIs y el filtro FIR modelando el canal, con coeficientes y datos de entrada complejos. El siguiente paso consiste en definir la *frequencia de actualización* de las RI en la simulaciones, esta depende de las características de los datos medidos, exactamente del número de RIs por longitud de onda de los datos usados, de la longitud de onda, de la velocidad del móbil y de la frequencia de muestreo de los datos a simular.

En este trabajo además del estudio y realización de todos los procesamientos expuestos anteriormente se han hecho una serie de simulaciones para comprobar que este modelo es efectivamente realizable y una comparación con los otros modelos estadísticos que pueden ser utilizados en términos de duración de la simulación. Los resultados indican que este modelo es mucho más rápido que los modelos estadísticos (seis veces más rápido en termino medio). Otro efecto a considerar es que el impacto de aumentar la velocidad del móvil o el número de trayectos seguido por la señal transmitida en el modelo de canal de radio es bastante menor en el modelo con datos experimentales que en el modelo estadístico.

Como un primer estudio sobre la utilización de datos experimentales en simulaciones de canales de radio de banda ancha en el Instituto de Radiocomunicaciones de la HUT, este trabajo representa un estudio básico, exponiendo las múltiples vías de investigación sobre el tema en el futuro. Estas vías no han sido investigadas en este proyecto por su complejidad, la cual haría durar el proyecto fin de carrera mucho más tiempo del requerido y disponible, entre estas se encuentran:

- Aplicación de métodos de superresolución
- Estudio del método de interpolación de RIs
- Estudio sobre la implementación de ruido e interferencia en simulaciones donde se utilicen datos experimentales para modelar el canal de radio.

La investigación de estos puntos seguirá a cabo en el mismo lugar donde se realizó el proyecto una vez presentado éste, sirviendo como una base sobre la que seguir investigando.

Teknillinen Korkeakoulu Universidad Politécnica de Madrid

Tekijä: F. Javier Sanz

Työn nimi: Mitattuun dataan perustuva laajakaistaisen radiokanavan simulointi

Päivämäärä: 16. Marraskuuta 1998 **Sivumäärä:** 103

Osasto: Sähkö- ja tietoliikennetekniikan osasto

Professuuri: S-26 Radiotekniikka

Työn valvoja: Professori Pertti Vainikainen

Työn ohjaaja: DI Kimmo Kalliola

Tässä diplomityössä tutkitaan mitattuun dataan perustuvaa laajakaistaisen radiokanavan simulointia. Radiorajapintasimuloinnissa radiokanava on tavallisesti implementoitu tilastollisella mallilla, joka on usein johdettu mittauskampanjasta. Tässä työssä tutkitaan mitatun radiokanavan käyttämistä sellaisenaan kanavan mallina, tämän vaikutusta simulaattorin suorituskykyyn, sekä esiprosessointia, joka tarvitaan datan muuttamiseksi simulaattorin hyväksymään muotoon.

Tarkasteltavaa menetelmää on tarkoitus käyttää laajakaistaisen radiokanavan simulointiin WCDMA-järjestelmässä, joka on ETSI:n UMTS:ään valitsema monipääsymenetelmä.

Tämän työn päätehtävät olivat peruslinkin muodostaminen mitatun datan ja simulointimallin välille sekä saatujen tulosten vertailu tilastollisella mallilla saatuihin tuloksiin. Käytettävä simulointiohjelma on COSSAP, joka on yleisesti käytetty linkkitason simulaattori. Tuloksia vertaillaan tarkastelemalla mallien simulointiin kuluvaa aikaa, koska radiokanavan simulointi edustaa suurinta osaa koko järjestelmän simulointiin kuluvasta ajasta.

Tulokset osoittavat, että mitattuun dataan perustuva malli johtaa merkittävästi (yleensä yli kuusi kertaa) lyhyempään simulointiaikaan kuin tilastollinen malli, ja ero korostuu kasvatettaessa nopeutta tai monitiekomponenttien lukumäärää. Mitattuun dataan perustuvan mallin simulointiaika kasvaa tällöin joitakin sekunteja (tilastomallilla kymmeniä sekunteja). Toinen käytettävällä mallilla saavutettava etu on, että radiokanavan käyttäytyminen toistuu realistisemmalla tavalla, koska se perustuu mitattuun dataan eikä tilastoihin.

Tässä diplomityössä toteuttiin peruslinkki mitatun datan ja simulointimallin välille. Tulevaisuudessa tätä tullaan täydentämään toteuttamalla superresoluutiomenetelmiä, menetelmiä impulssivasteen interpoloimiseksi, sekä tutkimalla kohinaa ja interferenssiä kokonaisen järjestelmän simuloinnissa.

Avainsanat: Impulssivaste, laajakaistainen monitieradiokanava, kanavamalli, kanavaluotain, WCDMA

FOREWORD

I would like to thank Professor Pertti Vainikainen for his excellent tutoring during the realisation of this thesis, and Kimmo Kalliola for his helpful and good instructive chats whenever it was necessary. I also would like to thank Jari Mattila and Esa Malkamäki, from the Communications Laboratory, for their really useful hints and comments.

Thanks as well to all the Radio Laboratory, especially to Antti Räisänen, who firstly gave me the chance to make this work here, and the rest of the people in the laboratory that made a comfortable place to work at. To all of you *paljon kiitoksia!!*.

Last I would like to thank my family for their support *¡gracias familia!*. It has been an unforgettable experience to come here to make my Master's Thesis, enjoying Finland's nature and character, and giving me the opportunity to improve my knowledge about the rich Finnish culture.

Espoo, 16th November 1998

F. Javier Sanz

Table of Contents

ABBREVIATIONS AND SYMBOLS

List of abbreviations

AGC	Automatic Gain Control
ARMA	Autoregressive Moving Average
ATDMA	Advanced Time Division Multiple Access
AWGN	Additive White Gaussian Noise
BDE	Block Diagram Editor
BER	Bit Error Rate
BPSK	Binary Phase Shift Keying
CoDiT	Code Division Testbed
COSSAP	Communications Simulation and System Analysis Program
COST	European Cooperation in the field of Scientific and Technical Research
CPU	Central Processing Unit
DOA	Direction Of Arrival
DP	Dominant Path
DS	Direct Sequence
DSP	Digital Signal Processor
ESPRIT	Estimation of Signal Parameters via Rotational Invariance Techniques
ETSI	European Telecommunications Standards Institute
FDMA	Frequency Division Multiple Access
FIR	Finite Impulse Response
GSM	Global System for Mobile communications
IIR	Infinite Impulse Response
ILPF	Ideal Low Pass Filter
IR	Impulse Response
IRC	Institute of Radio Communications at the Helsinki University of Technology
IS	Interim Standard
ITU	International Telecommunications Union
LOS	Line Of Sight
LSE	Least Square Error
MF	Matched Filter
MFD	Matched Filter Deconvolution
MUSIC	Multiple Signal Classification
MWRC	Multipath Wideband Radio Channel
NLOS	Non-Line Of Sight
PCS	Personal Communications System
PDP	Power Delay Profile
PRBS	Pseudo Random Binary Source
RACE	EU research programme "Research and development into Advanced Communications technologies for Europe"
SAW	Surface Acoustic Wave
SBT	Sequential Bit Tuning
SDS	Stream Driven Simulator
SIRCIM	Simulation of Indoor Radio Channel Impulse Response Models
SNR	Signal to Noise Ratio

SR	Superresolution
STDCC	Swept Time Delay Cross-Correlator
TDMA	Time Division Multiple Access
ULA	Uniform Linear Array
UMTS	Universal Mobile Telecommunications System
WCDMA	Wideband Code Division Multiple Access
WIRCS	Wideband Impulse Response Channel Simulator
WMRCS	Wideband Multielement Radio Channel Sounder

List of symbols

$\otimes$	Convolution sum
a_m	Amplitude
$a(t)$	Autocorrelation function of the probing signal
B_d	Doppler spread
c	Speed of the light
d	distance
d_0	close-in reference distance
$F_d(f)$	Fading spectrum
Δf	Clock frequency difference (sliding correlator)
f_c	Carrier frequency
$(\Delta f)_c$	Coherence bandwidth
f_{chip}	Chip frequency
f_d	Maximum Doppler shift
f_m	Doppler shift
f_N	Normalised fading frequency
f_s	Simulation sampling frequency
f_{st}	Time-domain simulated sampling frequency
f_{st}^{m}	Time-domain measured sampling frequency
$f_{s\tau}$	Delay-domain sampling frequency
$G(w)$	Ideal Low Pass Filter frequency response
$H_{MF}(f)$	Matched filter frequency response
$H_R(f)$	Receiver frequency response
h_i	Impulse Response
I	In-phase component
K	Oversampling rate
k	Scaling factor (sliding correlator)
L	Length of the code
L_f	Loss of the floors
L_w	Loss of the walls
M	Number of bits per symbol
$M(t)$	Variable number of taps
m	Length of the pseudo noise sequence
$m(t)$	Output of the matched filter
N_f	Number of floors

N_w	Number of walls
n	Path Loss exponent
P	Individual path output power
PL	Path Loss
$p(t)$	Pulse shape
Q	Quadrature component
R_{int}	Interpolation factor
R_u	Update rate
Δr	Delay domain resolution
$S(f)$	Doppler spectrum
$S(f)$	Frequency response of the probing signal
$S(f)$	Frequency response input to the deconvolution filter
$s(t)$	Probing signal
T_{chip}	Chip duration
T_m	Multipath spread
T_s	Sampling Period
$(\Delta t)_c$	Coherence time
t_{IR}	Duration of an impulse response
$u(t)$	Probing signal low pass equivalent
V_i	Amplitude
v	Speed of the mobile
v_f	Fixed mobile speed
$v(t)$	Effective noise component
$Y(f)$	Frequency response output of the deconvolution filter
$\alpha_n(t)$	n-th tap attenuation factor
$\delta(t)$	Dirac Impulse
$\Phi(f)$	Pre-deconvolution filter frequency response
ϕ_m	Phase
γ	Deconvolution constant
λ	Wavelength
σ	Standard deviation
τ_m	Delay
ω_s	Ideal Low Pass filter bandwidth

1.- INTRODUCTION

Today's mobile radio communication uses mainly narrow bandwidth, but the recent demand of new services, such as full-motion video, Internet access or video conferencing, will require higher bandwidths. Before the implementation of these new services, study of the different system options is needed. Therefore simulations play an important role in analysing the performance of the possible communication techniques. Simulators are useful for assessing the effectiveness of modulation, coding, equalisation and diversity techniques to combat the channel impairments.

In the wideband wireless environment one of the most important elements is the radio channel. Because of its variable and unpredictable behaviour it is usually difficult to model it in a correct way. This is done commonly by extensive measurement campaigns and further study of the results, yielding to recommendations (such as GSM, IS-95 or ITU models for the radio channel) that later will be used in the simulations.

In every simulation of radio communication systems one of the most important blocks, regarding time consumption and difficulty, is the radio interface. For the simulation of this block a channel model is needed . There are several models in use, but the most popular nowadays is the *transversal Finite Impulse Response (FIR) filter*, that is the one used in this work. It has been used extensively for the simulation of the radio interface, regardless of the system concepts, radio environments, different bit rates or different access methods.

Usually the tap coefficients of this FIR model are taken from statistical studies based on measured data or from recommendations (such as 5.05 in GSM or IS-95 in CDMA). This way the radio channel can be easily reproduced. One major drawback is that simulating the radio channel in this way takes most of the time of the system simulation, and as a statistical model it doesn't always represent the real behaviour of the channel. It is just an approximation agreed to be used and yielding to acceptable result.

A different approach is presented in this work, trying to overcome all these drawbacks: the coefficients are taken from experimental data. The data is measured by the IRC wideband radio channel sounder and adapted to the channel model and the simulation via signal processing. The simulator used is COSSAP, a stream driven simulator, where the driving force is the stream of data, not time (as happens in the time driven simulators). This type of simulators has been used previously for studying the performance of other systems like GSM and CDMA.

The goal of this work is to present the way how the measured data is used in the channel model for the simulations and study the performance of this approach in comparison with the classical methods for radio channel simulation, regarding speed and the actual way of reproducing the radio channel. It is thought that the use of actual measured data in the simulations may yield to a better performance of the system and to an improvement of the effectiveness of the options under study (such as modulation schemes, coding, equalisation, …) when they are implemented in the working system, because it represents in a more realistic way the actual behaviour of the radio channel.

The use of measured data for modelling the radio channel is also expected to yield to higher simulation speeds in comparison with the classical statistical methods applied. This way it will not be difficult to incorporate wideband models of multipath channel behaviour measured in different mobile and radio portable environments and at different frequencies as empirical data become available.

First in Chapter 2 a general review about the different ways of modelling the radio channel will be presented. In chapter 3 the description of the different sounding techniques will be presented, along with the sounder system, developed in the Institute of Radio Communications (IRC) of the Helsinki University of Technology, and used for measuring the data that will be used in this approach. In Chapter 4 the processing that has to be done to the measured data for adapting it to the simulation will be explained. In Chapter 5 the implementation of radio channel models in COSSAP will be described for both approaches: the statistical model and the one presented in this work, using measured data (WIRCS model). Finally in Chapter 6 the testing done for evaluating the performance of the WIRCS model in comparison with the classical statistical models will be presented, followed by the conclusions in Chapter 7.

2.- CHANNEL MODELLING

The radio interface of the future wideband radio transmission systems is becoming complex because of the dispersive, time variant nature of the radio channel. For narrowband case the radio channel can sometimes be evaluated analytically, under some simplifications, but for wideband case usually a simulation is needed due to the complexity. In these simulations the implementation and choice of the channel is one of the most important tasks in the overall process, that's why this field have been extensively researched, yielding to different models, as it will be presented next.

There are several mobile radio channel propagation models, distinguished by the travel distances and the different effects on the transmitted signal. Mainly two different propagation cases can be considered: Large-Scale Path Loss, and Small-Scale Fading, or fast fading. A more detailed description of these models is given in [1] and [2], in this section a brief description of the different models will be presented.

2.1. Large-Scale Path Loss models.

The Large - Scale Path Loss propagation models predict the mean signal strength for an arbitrary separation distance between transmitter and receiver (T-R). They are useful in estimating the radio coverage area of a transmitter, since they characterise signal strength over T-R separation distances varying from several hundreds of meters to thousands.

The main parameters that will affect the different models will be:

- T-R distance.
- Height of the T-R antennas.
- Environment.

This models are as well divided into two different cases: Log - distance Path Loss Model and Log-normal Path Loss Model, whose main difference is the consideration of the environment features.

The Log-distance Path Loss Models state, from theoretical and measurement based propagation models, that the average received signal power decreases logarithmically with distance, both in indoor and outdoor radio channels. The average large-scale path loss for an arbitrary T-R separation is expressed as a function of distance by using a path loss exponent, *n*.

$$\overline{PL}(d) \propto \left(\frac{d}{d_0}\right)^n \tag{2.1}$$

or

$$\overline{PL}(dB) = \overline{PL}(d_0) + 10n\log(\frac{d}{d_0}) \tag{2.2}$$

where n is the path loss exponent which indicates the rate at which the path loss increases with distance, d_0 is the close-in reference distance which is determined from measurements close to the transmitter, d is the T-R separation distance, and the bar indicates the ensemble average of all possible path loss for a given value of d. When plotted on a log-log scale, the modelled path loss is a straight line with a slope equal to 10n dB per decade. The value of n depends on the specific propagation environment. For example, in free space, n is equal to 2, and when obstructions are present n will have a larger value (typically 3-5, depending on the environment).

The Log-distance Path Loss model, given in Eq. (2.2), doesn't consider that the surrounding environmental clutter may be vastly different at two different T-R locations, being separated by the same distance. This yields to measured signals which are very different from the average value predicted by Eq. (2.2). Measurements have shown that at any given distance d the path loss at a particular location is random and distributed log-normally (normal in dB) about the mean distance dependent value:

$$PL(d) = \overline{PL}(d) + X_\sigma = \overline{PL}(d_0) + 10n\log(\frac{d}{d_0}) + X_\sigma \tag{2.3}$$

where X_σ is a zero-mean Gaussian distributed random variable (in dB) with standard deviation σ (also in dB).

This phenomenon is known as log-normal shadowing, or Log-normal Path Loss Model. The log-normal distribution describes the random shadowing effects which occur over a large number of measurement locations, where the T-R distance is the same. Simply put, log-normal shadowing implies that measured signal levels at a specific T-R separation have a Gaussian (normal) distribution about the distance-dependent mean given by Eq. (2.2), where the measured signal levels have values in dB units.

2.1.1 Outdoor propagation models.

Any single model does not predict every variation that will take place in the environment, the methods used vary widely in their approach, complexity and accuracy. To overcome this obstacle sometimes a combination of several models is used, depending on the environmental conditions relevant to the situation.

In this section the most common outdoor propagation models used nowadays for prediction of path loss will be described, as they are presented in [1,2].

Free space.

Free space path loss is usually the reference point for all the path loss models employed. The equation that is used for determining free space path loss is based on the relation $1/R^2$, or 20 dB/decade path loss.

Okumura model.

Okumura's model is one of the most widely used models for signal prediction in urban areas. Okumura developed a set of curves giving the median attenuation relative to free space in an urban area. This model is wholly based on measured data and does not provide any analytical explanation.

Hata model.

The Hata model is an empirical formulation of the graphical path loss data provided by Okumura. Hata presented the urban area propagation loss as a standard formula and supplied correction equations for applications to other situations.

Cost231 Hata.

The European Co-operative for Scientific and Technical research (EURO-COST) formed the COST-231 working group to develop an extended version of Hata model. It is very similar to Hata model with the exception of extended frequency range and correction factors added based on the geography that the model is applied to.

Cost231 Walfisch/Ikegami.

The Cost231 Walfisch/Ikegami propagation model is used for estimating the path loss in an urban environment for cellular communications, it is a combination of empirical and deterministic modelling for estimating the path loss in an urban environment over the frequency range 800 to 2000 MHz. It takes into account the impact of rooftops and building height.

Wideband Personal Communication Services (PCS) Microcell Model.

Work by Feuerstein [3] described statistics for path loss, multipath, and coverage area, developed from extensive measurements in line-of-sight (LOS) and obstructed line-of-sight (NLOS) environments. This work revealed that a 2-ray ground reflection model is a good estimate for path loss in LOS microcells, and a simple log-distance path loss model holds well for NLOS microcell environments.

Out of these models, most cellular operators use a version of the Hata model for conducting propagation characterisation, while the PCS operators are using either Hata or Cost231Hata model as their primary method to determine path loss. The Cost231 models (both Hata and Walfisch/Ikegami) are used primarily in Europe for GSM modelling and in some propagation models for cellular systems in the United States.

Other models found in the literature [1,2] are Carey, Elgi, Longley-Rice, Durkin, Bullington and Lee.

2.1.2.- Indoor propagation models.

This section outlines models for path loss inside buildings. The indoor radio channel differs from the traditional mobile radio channel in two aspects:

- The distances covered are much more smaller.
- The variability of the environment is much greater for a much smaller range of T-R separation distances.

The models presented in [1] are:

Partition Losses

These models take into account the different partitions existing inside buildings (e.g. brick or plaster board walls, wood or metal reinforced or nonreinforced concrete between floors, etc …) which have effect on the path loss. Researches have formed extensive data bases of losses for a great number of different types of partitions.

Log-distance Path Loss Model.

Works by many researchers developed the indoor path loss equation given by:

$$PL(dB) = PL(d_0) + 10n\log(\frac{d}{d_0}) + X_\sigma \qquad (2.4)$$

where the value of n depends on the surroundings and building type, and X_σ represents a normal random variation variable in dB having a standard deviation of σ dB. Notice that equation (2.4) is identical in form to the log-normal shadowing model of equation (2.3).

Ericsson Multiple Breakpoint Model.

The Ericsson radio system model was obtained by measurements in a multiple floor office building. The model has four path loss breakpoints (in a log-log scale) and considers both an upper and lower bound on the path loss.

Attenuation Factor Model.

The Attenuation Factor Model is an in-building propagation model that includes the effect of building type as well as the variations caused by obstacles. This model provides flexibility and was shown to reduce the standard deviation between measured and predicted path loss compared to the case when only a log-distance model was used.

Motley-Keenan Model.

The Motley-Keenan Model is an empirical model which takes into account the attenuation due to walls and floors in the T-R way. This way it could be considered as a Partition Losses model or as a Attenuation Factor Model, but it is widely accepted itself for modelling indoor path losses. The path loss (dB) predicted by this model is given by:

$$PL(dB) = PL(d_0) + 10n\log(x) + \sum_{j=1}^{J} N_{w_j} L_{w_j} + \sum_{i=1}^{I} N_{f_i} L_{f_i} \qquad (2.5)$$

where $PL(d_0)$ denotes the loss at the reference point, *n* is the power decay index and *x* represents the transmitter to receiver path length, N_{w_j} and N_{f_i} denote the number of walls and floors of different kinds that are traversed by the transmitted signal and L_{w_j}(dB) and L_{f_i} (dB) represent their corresponding losses.

Although this model is widely used it has to be mentioned that it yields to excessive losses predictions when a corridor is involved. This is due to a certain waveguide effect present in corridors.

2.2.- Small Scale Fading.

Small scale fading is used to describe the rapid fluctuation of the amplitude of a radio signal over a short period of time or travel distance, usually some tens or hundreds of wavelengths, so that large-scale path loss may be ignored. Fading is caused by the interference of two or more versions of the transmitted signal travelling through different paths and arriving at different times This effect may cause wide changes in the amplitude and phase of the received signal.

Depending on the relation between the signal parameters (such as bandwidth, symbol period, etc.) and the channel parameters (such as rms. delay spread and Doppler spread), different transmitted signals will undergo different types of fading [1]. If the reciprocal bandwidth of the transmitted signal is much larger than the multipath time delay spread of the channel, then the channel can be approximated as having no excess delay, these channels are called *flat fading, amplitude varying* or *narrowband channels.* The distribution of the instantaneous gain of flat fading channels is important for designing radio links, and the most common amplitude distribution is the Rayleigh distribution.

On the other hand, if the reciprocal bandwidth of the transmitted is much smaller than the multipath time delay spread of the channel, the received signal includes multiple versions of the transmitted waveform which are attenuated (faded) and delayed in time, and hence the received signal is distorted. These channels are called *wideband channels,* or *frequency selective channels,* and will be described later in section 4.4.

The most frequently used models for the small-scale fading will be presented next.

2.2.1. Tapped Delay Line.

The small scale variations of a mobile radio signal are described by the Impulse Response (IR) of the mobile radio channel. The IR is a wideband characterisation and contains all the information necessary to simulate or analyse any type of radio transmission through the channel. This is caused by the fact that a mobile radio channel can be modelled as a linear filter with a time-variant impulse response, where the time variation is caused by the relative motion between transmitter and receiver, as it is shown in [1,4,5,10].

The most commonly used model for the implementation of this linear filter is the *transversal Finite Impulse Response (FIR) filter,* also called *tapped delay line*, as it is depicted in Fig. 2.1.

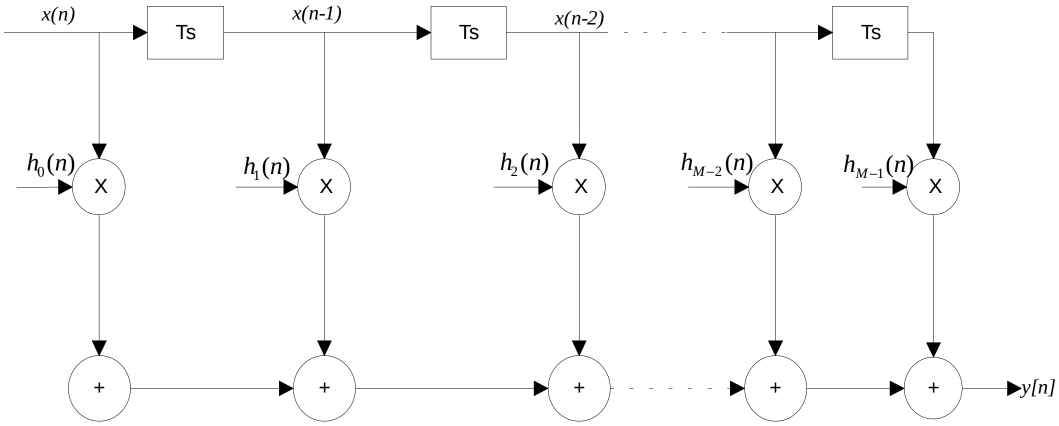


Figure 2.1. *Transversal Finite Impulse Response (FIR) filter model for wideband channels.*

The discrete-time channel input and output are respectively *x[n]* and *y[n]* and the unit tap delay will be equal to the system sampling interval T_s . The filter coefficients will consist of *M* complex-valued numbers h_i *[n], i =0,1,…,M-1,* representing the IR of the channel at delay instants iT_s . Thus:

$$y[n] = \sum_{i=0}^{M-1} x[n-i]h_i[n] \tag{2.6}$$

where the time-variant nature of the channel is represented by the index *n*, and the dispersive nature by the index *i*, representing the delay instants. The structure of the IR data is:

$$h\ (t)=\begin{bmatrix} h_0(t_n) & h_1(t_n) & \cdots & h_{M-2}(t_n) & h_{M-1}(t_n) \\ h_0(t_{n-1}) & h_1(t_{n-1}) & \cdots & h_{M-2}(t_{n-1}) & h_{M-1}(t_{n-1}) \\ h_0(t_{n-2}) & h_1(t_{n-2}) & \cdots & h_{M-2}(t_{n-2}) & h_{M-1}(t_{n-2}) \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ h_0(t_0) & h_1(t_0) & \cdots & h_{M-2}(t_0) & h_{M-1}(t_0) \end{bmatrix} \tag{2.7}$$

where each row represents an IR at a given time instant and each column a delay instant. The Eq. (2.7) would this way represent the behaviour of the radio channel since the time instant t_o to t_n according the model followed. The time step value $t_i - t_{i-1}$ will be given by the IRs sampling rate, i.e. the number of IRs measured per time unit.

This transversal FIR filter structure is very often used for modelling the Multipath Wideband Radio Channel (MWRC), described in section 4.4, with software simulation, regardless the different system concepts (GSM, IS-95, UMTS, etc ...) or the different access methods (CDMA,TDMA,FDMA).

2.2.2. Statistical Wideband FIR Channel Model.

In the statistical modelling the time dependent amplitudes $a_m(t)$, phases $\phi_m(t)$, delays $\tau_m(t)$, and possibly Doppler frequencies $f_m(t)$ of the different taps are assumed to be randomly distributed, and must be generated from probability distributions. The wideband channel IR is presented in the form:

$$h(\tau,t)=\sum_{i=1}^{M} a_m(t)e^{j\phi_m(t)}\delta\left[\tau-\tau_m(t)\right] \tag{2.8}$$

In [1] several statistical models are presented The common basis for calculating the probability distributions is either from known probability distributions, as it's done in the Two-Ray Rayleigh Fading model, or from empirical data, as it's done in Saleh and Valenzuela Indoor Statistical model, SIRCIM and SMRCIM. These models differ from each other in the locations where the empirical data are taken, thus in the type of environment in which are valid.

2.2.3. Wideband Channel Measurement Database.

The wideband channel measurement database consists of a two-dimensional matrix containing a sequence of measured IRs, where each IR consists of a number of complex samples, with uniform increments in delay domain given by the measurement equipment. For using these measured data in the FIR model it's just necessary to process it in delay and time-space domain.

2.2.4. Ray-tracing Wideband Radio Channel Model.

The Ray-tracing Model is obtained in the following way:

From a database modelling the geometry of the radio environment (streets, buildings, walls, ceilings, inventory, equipment, etc …), and including the exact position of transmitter and receiver, the ray-tracing algorithm calculates the path loss and exact propagation delay for each ray. The different rays are calculated considering the reflection, diffraction and scattering of the transmitted electromagnetic ray, and added to the LOS ray (if existing).

If the location of transmitter or receiver is changed, the differential phase shifts are also recorded. This model is very accurate in urban or indoor radio environment, when the structure is regular, but no so good in rural or irregular indoor environments.

The Ray-Tracing model can be as well divided into two different types, regarding the information it provides about the channel:

- 2-dimensional: when it's produced 2-dimensional information about the channel. Delay instants and azimuth angle of every path are calculated.
- 3-dimensional: When the information is 3-dimensional, delay instants, azimuth and elevation angles of every path are calculated.

It is important to note that this kind of model gives already complete information about the channel, both in large and small scale.

2.2.5. Relationship to FIR models.

Somehow the models presented above can be considered as FIR filter models, for all of them give information about the delay instants of the different existing paths and their correspondent amplitude value, but the main difference with the model presented in section 2.2.1. is that the delay instants do not match and a conversion is necessary.

- In the statistical and ray-tracing models the delay instants don't have to be necessarily equally spaced, as they are in the FIR filter model. In this case a path-tap conversion is necessary. This way different model approaches, such as GSM, IS-95, UMTS or ITU, where the delay instants are variable and unequally spaced, can be converted into a FIR filter model.

- In the wideband channel measurement database model the uniform increments in delay domain are much smaller than the ones in the FIR filter model, thus it has to be adapted to the increment T_s (see Fig. 2.1) of the FIR filter model through data decimation.

2.3. Practical Channel Models.

In this section different channel models will be presented, all of them based on statistical radio channel model, for different systems. The systems presented will be GSM and CDMA, the RACE projects CoDiT and ATDMA [19], the ITU models [27], and the SIRCIM developed in [21]. For allowing simulation the propagation models are presented as:

- A discrete number of taps, each determined by their time delay and average power.

- The Rayleigh or Rice distributed amplitude of each tap, varying accordingly to a Doppler spectrum.

The most common way of simulating the radio channel is through a statistical model. This model can be derived from measured data, but this measured data is not usually used in the simulator, it is just used to obtain the statistical model that will be implemented, as it is presented in this section.

In this work a different approach is studied, using the measured data in the simulation. The main reasons leading to this are:

- Realistic model of the radio channel, thus more accurate behaviour.
- Comparison of the speed of the simulations with the statistical models.

2.3.1. GSM.

The channel model presented for the GSM system is based on the recommendation GSM 05.05 for different types of terrain. The recommendation gives the number of taps, the delay value of each tap and correspondent average power, and the type of Doppler spectrum of each tap. (see COSSAP GSM library).

Table 2.1. *Recommendation GSM 05.05. Typical Urban (TU) terrain.*

Tap	Delay [μs]	Power [dB]	Doppler Spectrum
1	0	-3	CLASS
2	0.2	0	CLASS
3	0.5	-2	CLASS
4	1.6	-6	CLASS
5	2.3	-8	CLASS
6	5.0	-10	CLASS

Table 2.2. *Recommendation GSM 05.05. Hilly Terrain (HT).*

Tap	Delay [μs]	Power [dB]	Doppler Spectrum
1	0	0	CLASS
2	0.1	-1.5	CLASS
3	0.3	-4.5	CLASS
4	0.5	-7.5	CLASS
5	15	-8	CLASS
6	17.2	-17.7	CLASS

Table 2.3. *Recommendation GSM 05.05.Rural Area terrain (RA).*

Tap	Delay [μs]	Power [dB]	Doppler Spectrum
1	0	0	RICE
2	0.1	-4	CLASS
3	0.2	-8	CLASS
4	0.3	-12	CLASS
5	0.4	-16	CLASS
6	0.5	-20	CLASS

The power of each tap is normalised to the greatest value. The Doppler spectra are classified either as CLASS or RICE, defined as:

- Classical (CLASS):

$$S(f) = \frac{A}{\sqrt{1-\left(\frac{f}{f_d}\right)^2}} \qquad |f| \le f_d \tag{2.9}$$

- Rician (RICE), this spectrum represents the combination of a classical spectrum and a direct path. For GSM:

$$S(f) = \frac{0.41}{2\pi f_d\sqrt{1-\left(\frac{f}{f_d}\right)^2}} + 0.91\delta(f - 0.7f_d) \tag{2.10}$$

where $f_d = \frac{f_c v}{c}$ and:

- v is the velocity of the mobile.
- c is the speed of the light ($3.0*10^8$ m/s)
- f_c is the transmitting frequency (900 MHz for GSM).
- f_d is the maximum Doppler shift.

2.3.2. CDMA.

The radio channel model for IS-95 CDMA system is implemented according to the propagation model defined in the COSSAP CDMA library for this model. It corresponds to the 3-tap model, as it is described in Table 2.4.

Table 2.4. *3-Tap channel description for CDMA as described in IS-95*

Tap Number	**Delay [µs]**	**Power [dB]**	**Doppler Spectrum**
1	0	0	CLASS
2	1.5	-3	CLASS
3	14.5	-6	CLASS

2.3.3. CoDiT and ATDMA.

Both CoDiT (*Co*de *Di*vision *T*estbed) and ATDMA (Advanced Time Division Multiple Access) were research projects within the European RACE programme (1992-95) that had the objective to explore the potential of CDMA and TDMA, respectively, as access methods for a third generation mobile telephone system.

For the definition of the propagation models at the very beginning of the project ATDMA, wideband propagation measurements in the 2 GHz frequency band were carried out by using three channel sounders adapted to three cell types pico-, micro- and macrocells. Furthermore, an analysis tool was developed to evaluate this data and finally the wideband channel models were implemented in hardware and software channel simulators.

The approach taken by CoDiT for channel modelling in the simulations is statistical-physical modelling. The channel models are obtained by means of a statistical spatial scatterer distribution. Every different environment is characterised in terms of a number of scatterers (less than 20) and some parameters relative to each of the scatterers, like their relative amplitudes, time delays, Doppler variations and incidence angles of the wavefronts. The values for these parameters are sorted out from statistical distributions. A complete and detailed report of the work can be found in [18].

In the case of ATDMA the so called *stored channels* were used for the purpose of link level simulation and transceiver performance analysis. The stored channels consist of a set of complex IRs corresponding to different environments and chosen from a testbed of measurements. As stated in the final report of the project:

- There was no well established method to derive wideband propagation models from a large set of measurements files. A new method based on the statistical evaluation and classification of measured data files was developed leading to the definition of the recorded channels as typical and atypical. These models, which have the format of the sampled data, reproduce the full properties of the radio channel including time-variability, frequency-selectivity and long-term fading.

- The recorded channel models did not fit completely with the software tools used in the simulations. Consequently an other simpler to use model, the synthetic model, consisting of a set of discrete paths with delay, amplitude and fading distributions, was derived from the recorded channel model.

Both CoDiT and ATDMA projects have delivered the channel model realisations in COST 207 format. These models are simplified versions of the actual channel models that are produced within the projects. The amplitude distributions, for example, are assumed to be either Rayleigh or RICE. An example of the results for macrocellular environment is shown in Tables 2.5. and 2.6. For further information, see [19].

Table 2.5. *CoDiT Channel model in COST 207 Format. Macrocellular environment.*

Tap	Delay [ns]	Power [dB]	Doppler Spectrum
1	100	-3.2	CLASS
2	200	-5.0	CLASS
3	500	-4.5	CLASS
4	600	-3.6	CLASS
5	850	-3.9	CLASS
6	900	0.0	CLASS
7	1050	-3.0	CLASS
8	1350	-1.2	CLASS
9	1450	-5.0	CLASS
10	1500	-3.5	CLASS

Table 2.6.*ATDMA Channel model in COST 207 Format. Macrocellular environment.*

Tap	Delay [ns]	Power [dB]	Doppler Spectrum
1	0	0	CLASS
2	380	-10.0	CLASS
3	930	-22.7	CLASS
4	1940	-24.7	CLASS
5	2290	-20.7	CLASS
6	2910	-22.1	CLASS

2.3.4. ITU Models.

The International Telecommunication Union (ITU) in [27] specifies a channel impulse response model for terrestrial test environment based on a tapped delay model. The model is characterised by the number of taps, the time delay relative to the first tap, the average power relative to the strongest tap, and the Doppler spectrum of each tap. It is important to note that a small variation, +/- 3 %, in the relative time delay is allowed so that the channel sampling rate can be made to match some multiple of the link simulation sample rate.

The values of the delays and average powers for these models are given in Tables 2.7, 2.8 and 2.9 for different propagation environments. As delay spread variability cannot be captured using a single tapped delay line, two multipath channels are defined for each test environment: channel A for low delay spread case, and Channel B, for median delay spread.

Table 2.7. *Indoor Office Test Environment Tapped-Delay-Line Parameters*

Tap	**Channel A**		**Channel B**		**Doppler Spectrum**
	Rel. Delay (nSec)	**Avg. Power (dB)**	**Rel. Delay (nSec)**	**Avg. Power (dB)**	
1	0	0	0	0	FLAT
2	50	-3.0	100	-3.6	FLAT
3	110	-10.0	200	-7.2	FLAT
4	170	-18.0	300	-10.8	FLAT
5	290	-26.0	500	-18.0	FLAT
6	310	-32.0	700	-25.2	FLAT

Table 2.8. *Outdoor to Indoor and Pedestrian Tapped-Delay-Line Parameters*

Tap	**Channel A**		**Channel B**		**Doppler Spectrum**
	Rel. Delay (nSec)	**Avg. Power (dB)**	**Rel. Delay (nSec)**	**Avg. Power (dB)**	
1	0	0	0	0	CLASSIC
2	110	-9.7	200	-0.9	CLASSIC
3	190	-19.2	800	-4.9	CLASSIC
4	410	-22.8	1200	-8.0	CLASSIC
5	-	-	2300	-7.8	CLASSIC
6	-	-	3700	-23.9	CLASSIC

Table 2.9. *Vehicular Test Environment, High Antenna, Tapped-Delay-Line Parameters*

Tap	**Channel A**		**Channel B**		**Doppler Spectrum**
	Rel. Delay (nSec)	**Avg. Power (dB)**	**Rel. Delay (nSec)**	**Avg. Power (dB)**	
1	0	0	0	-2.5	CLASSIC
2	310	-1.0	300	0	CLASSIC
3	710	-9.0	8900	-12.8	CLASSIC
4	1090	-10.0	12900	-10.0	CLASSIC
5	1730	-15.0	17100	-25.2	CLASSIC
6	2510	-20.0	20000	-16.0	CLASSIC

2.3.5. SIRCIM.

SIRCIM is a computer simulator based upon models presented by Rappaport, Seidel and Takamizawa in [21]. This paper presents statistical impulse response models that were derived from measurements. A *statistical* model was used mainly for two reasons:

- The number of potential reflectors was too large to incorporate into a purely deterministic model.

- The locations of potential scatterers varied considerably as transmitters and receivers were moved within a building and from building to building. While it should have been possible to incorporate a limited number of specular reflectors by a ray tracing and diffraction theory, careful measurements which yield scattering parameters for walls, ceilings, and other common objects in buildings hadn't been yet reported in the literature.

The models presented describe:

- The distribution of the number of multipath components in a particular multipath delay profile.
- The distribution of the number of multipath components received within local area.
- The probability of receiving each multipath component at a particular excess delay.
- The distributions of the amplitudes, phases and time delays of multipath components received within a local area.
- The probability of receiving a multipath component at small scale locations.

SIRCIM creates multipath channel power delay profiles with the same ensemble and local statistics as the measured data. The multipath channels are modelled as a linear filter with a complex baseband impulse response. With this simulation arriving signals at a receiver from one (or many) transmitters can be predicted by performing convolutions of the transmitted signals with simulated channels.

3.- CHANNEL SOUNDER

In this chapter the sounding and technology of the IRC Wideband Multielement mobile Radio Channel Sounder (WMRCS) will be explained. The WMRCS is used for acquiring measured data for channel characterisation.

First the common wideband channel measurements techniques will be explained, followed by the description of the IRC-WMRCS and the different operation procedures it has.

3.1.- Wideband channel sounding techniques.

Different channel sounding techniques are possible for investigating the properties of the wideband mobile radio channel. A detailed description is given in [6] and [7], thus here only a brief overview will be given. The measurement techniques can be divided into frequency and time domain, as it is shown in Fig. 3.1, determining the characterisation achieved.

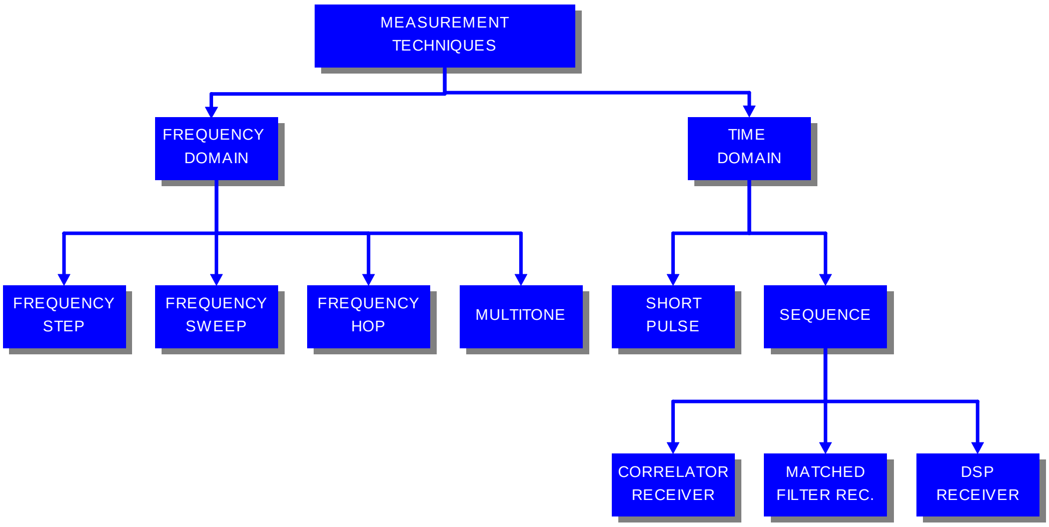


Figure 3.1. *Wideband radio channel measurement techniques*

3.1.1. Frequency-Domain Techniques.

Frequency step, frequency sweep and frequency hop methods use a device (e.g. frequency synthesiser) capable of producing several carrier frequencies. The carrier frequency is either stepped or swept over a specified frequency band. In frequency hopping this is done in a random manner, while in frequency step and sweep a constant frequency variation is applied.

The result of these methods is the channel transfer function, which can be obtained by evaluating the changes of the complex envelope of the transmitted signal. One drawback of these methods is the slowness in the measurement process, which allows the time varying channel to change during the measurement.

In multitone techniques several different carrier frequencies are transmitted at the same time, thus shortening the measurement time but making the system more complex.

3.1.2.- Short Pulse Technique.

In the short pulse technique a very short-duration pulse is transmitted, which can be considered as a pseudo-impulse. Thus the received signal represents the convolution of the sounding pulse with the channel IR [7]. In order to calculate the time-varying nature of the channel, the pulse is transmitted periodically, providing a series of "snapshots" of the multipath channel. Successive snapshots form a "motion picture" representation of the multipath propagation between transmitter and receiver.

The major problem of this technique is the high peak-to-mean power ratio, necessary to detect the weak echoes as well. Trying to solve this, techniques using pulse compression are employed.

3.1.3. Sequence Techniques.

In a Direct Sequence (DS) measurement system the transmitted carrier is modulated with a high-rate Pseudo-Random-Binary-Source (PRBS) which spreads the signal over a large bandwidth. The different receivers used give way to the different sequence time techniques existing: Correlator Receiver, Convolution Matched Filter Receiver and DSP Receiver.

3.1.3.1. Correlator receiver.

In a correlator receiver the incoming waveform is correlated with a locally generated reference code for all possible relative phase shifts. During the correlation the value of the convolution of the channel with the autocorrelation function of the PRBS sequence is calculated. The output of this process represents the lowpass equivalent estimate of the channel IR. This process of scanning through all possible relative phase shifts causes time scaling, thus the channel IR estimation at the correlator output is a time average over several transmitted code sequences.

The most extended correlator receiver used is the Sliding Correlator, also called swept time-delay crosscorrelator (STDCC), where the correlation processing is done using a swept time-delay technique in which the incoming signal is correlated with a *m*-sequence identical to that used in the trasmitted, but clocked at a slightly slower rate. The scaling factor is determined by the difference in clock rates at transmitted and receiver.

3.1.3.2. Convolution matched filter.

In the convolution matched filter receiver a filter matched to the sounding waveform is used. The system operates in real time because the output of the matched filter is a series of snapshots of the channel response. The filter can be implemented by a Surface Acoustic Wave (SAW) matched to the specific transmitted sequence.

This method presents several disadvantages: the real time information cannot be recorded without very expensive equipment, the performance of the practical SAW devices is limited by deficiencies, and any matched filter device is only configured appropriate for one specific sequence.

3.1.3.3. DSP receiver.

Digital Signal Processing (DSP) can be used to produce matched filtering by calculating the IR. The received signal is converted into digital and then the matched filtering is done in the DSP unit in a post processing phase. The result is stored in a buffer and then transferred to a mass storage device, together with the Automatic Gain Control value. Performing the match filtering with DSP methods provides an improved dynamic range of about 15-20 dB over the SAW matched filter sounders, and allows flexibility as well, it is easily possible to adjust the receiver to different transmitter sequences, by just changing the software.

3.2. IRC Wideband Mobile Multielement Radio Channel Sounder.

The choice of the channel sounding technique will usually depend on the application foreseen for the propagation data. To produce meaningful and accurate data consideration to the system factors should be given. In this section an overview of the IRC-WMRCS is given, followed by the different operation parameters.

3.2.1. Overview.

The IRC-WMRCS consists of transmitter, receiver and a data acquisition and control unit, as shown in Fig.3.1. Deeper information about the sounder can be found in [9,29,30].

In the transmitter a maximum length pseudo-random binary sequence, also known as m-sequence (for details see [8]), is used to BPSK modulate the carrier frequency (currently 2.154 GHz.). Several code sequence lengths and chip frequencies can be selected, thus enabling adaptation to the radio channel environment under investigation.

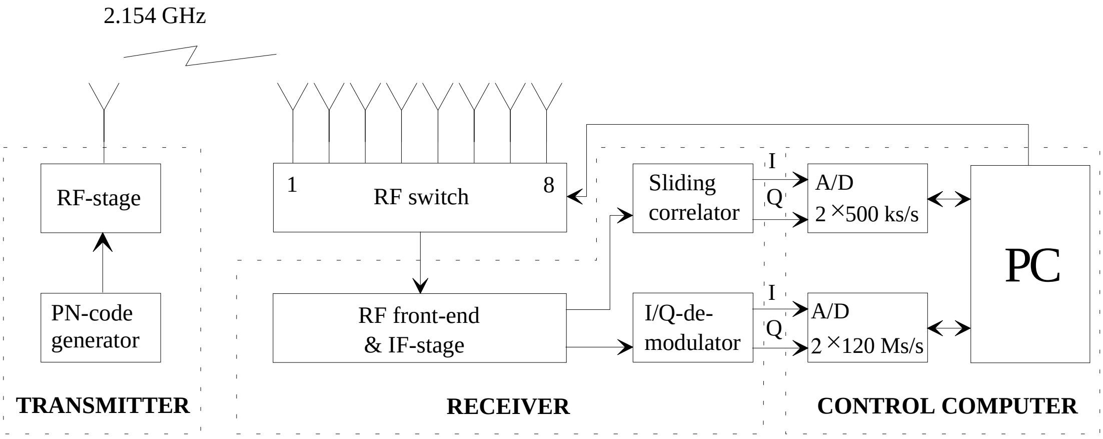


Figure 3.2. *IRC-Wideband Multielement Radio Channel Sounder as presented in [9]*

In the receiver the received modulated carrier is downconverted in the radio frequency (RF) front end to an intermediate frequency (IF) of 300 MHz. In the IF stage the signal is adjusted via digital programmable step attenuators to a reference level appropriate for the input levels of the following circuits. The IF signal is fed, depending on the receiver configuration, to one of the following two blocks.

- Sliding correlator receiver.
- Direct sampling receiver.

As it has been previously explained, the sliding correlator unit produces a time averaged estimate of the channel IR. Therefore sampling of the sliding correlator output signals (inphase & quadrature components) can be done with a low speed A/D converter. During the measurements the sampled data is continuously written to the hard disk.

Real time signals are produced by the direct receiver, which only splits the IF signal to its inphase (I) and quadrature (Q) components and downconverts them to the baseband. To obtain the channel IR from these signals further post processing is needed.

Due to the high frequencies in the output signals of the direct sampling matched filter unit, very high speed A/D converters are employed to sample the I and Q signals. Here the sampled data is stored on fast onboard buffer memory and is transferred to a disk after the completion of the measurement, i.e. when the onboard memory is full.

The IRC-WMRCS allows as well multielement antenna configuration, where each antenna element IR is sampled in a fast succession and stored individually as explained above. In [9] an antenna array of 8 elements was used, later it has been extended to 64 elements.

3.2.2. System Operation and Functionality.

The interrelationship between all the system's factors must be considered in order to produce meaningful and accurate results. Consideration should be given to the *m*-sequence length *m*, the chip rate and the scaling factor *k* of the swept correlator (when used). All these factors are adjustable by the IRC-WMRCS equipment and will be straightforward related to the measured data.

3.2.2.1. Code Length and Chip Frequency.

An appropriate code length and chip frequency has to be selected to fit the radio propagation environment under investigation. Code length L and chip frequency f_{chip} determine the time duration t_{IR} of the code as:

$$t_{IR} = \frac{L}{f_{chip}} \qquad (3.1)$$

The following basic rules must be considered when selecting these parameters:

1.- The sequence duration has to be long enough to cover all possible echo paths arriving at the receiver with different values of time delay. If sequence is chosen too short, delay paths from the previous IRs will overlap into the begin of the next IR, causing signal distortion.

2.- On the other hand, it is advisable to keep the code length as short as possible. This avoids wasting storage capacity as a too long sequence would have a long tail containing only noise. This is especially important in the direct sampling receiver, where the data is first stored in an onboard memory which has limited capacity.

In the sliding correlator system the IR is not obtained in real-time but is time scaled by a factor *k*, defined by:

$$k = \frac{f_{chip}}{\Delta f} \qquad (3.2)$$

where Δf is the difference between the chip frequencies of the transmitted and regenerated chip frequency. Thus the IR is a time averaged estimate and *k* times longer than the real-time one:

$$t_{IR} = \frac{L}{f_{chip}} \cdot k \qquad (3.3)$$

The following tables list all the possible combinations of code length and chip frequencies, and give the resulting time durations of the IR. Note that for the sliding correlator only the combination L=1023 chips and f_{chip} = 53.85 MHz exists, resulting in a code duration of 19 μs. Table 3.1 lists the possible k factors and the resulting duration of the scaled IR, while Table 3.2. shows the real-time duration of all possible length-frequency combinations in the direct sampling receiver system.

Table 3.1. *Sliding correlator k factors and resulting sequence durations.*

k	***t_{IR}* [ms]**
1077	20.46
2154	40.92
2691	51.12
5385	102.30

Table 3.2. *Direct sampling receiver system code combinations and resulting sequence durations.*

	IR DURATION [μs]			
L/f_{CHIP} (MHz)	**2.5**	**10**	**30**	**60**
13	5.20	1.30	0.43	0.22
31	12.40	3.10	1.03	0.52
63	25.20	6.30	2.10	1.05
127	50.80	12.70	4.23	2.12
255	102.00	25.50	8.50	4.25
511	204.40	51.10	17.03	8.52
1023	409.20	102.30	34.10	17.05
2047	818.80	204.70	68.23	34.12

Note from Table 3.2. that the 13 chip code is not an *m*-sequence but a Barker code.

3.2.2.2. Slow and Fast Systems.

The sampling frequencies of the slow system, i.e. the one using the sliding correlator receiver, and fast system, using the direct sampling receiver, are of very different magnitude (order of hundreds of kHz compared to around 100 MHz). This demands for totally different approaches of data handling in the acquisition process. The main differences between the two systems reside in the timing and AGC control:

Timing:

In the *slow system* there's only the possibility of one fixed sequence length, therefore adaptation to the expected maximum channel IR duration is achieved by sampling only the first part of the IR and ignoring the long noise tail. Then the system waits as long as this IR is expired plus the number of additional IR, specified by the Doppler frequency setting.

When the sampling cycle is achieved the data is transferred to the PC memory and finally stored in the hard disk, thus the only limitation for the measurement run is the available hard disk space.

In the *fast system* a continuous data stream to the hard disk is not feasible due to the high sampling frequencies used. Here the data is stored on a fast onboard memory and the transfer of the whole data set is done when the onboard memory is full. The IRC-WMRCS sampling card is equipped with a 2x4 MBytes onboard memory for both I and Q components.

AGC control:

In the *slow system* the time between two sampling sets is used to calculate a new value for the AGC, based on the peak value of the last detected IR. This value is output to the attenuators. The currently set attenuation value is stored together with the sample data in the file.

In the *fast system* there's no online data available during the measurement, so the AGC is implemented in a different way: an analogue voltage proportional to the power level in the IF stage is sampled by the slow sampling unit. The control program compares this voltage with a reference level and determines a new value for the step attenuators. The attenuation values are stored in a vector during the measurement and after completion assembled to the data file in the appropriate way.

3.2.2.3. Channel Sounder File Utilities.

In the IRC-WMRCS four different types of files are generated to store the sampled IRs, as well as system configuration information, descriptive information entered by the user, and temporary data.

- *Configuration file.* Stores the current system configuration in order to use it in further measurements.

- *Sample file.* The measurement data is stored in a raw and unscaled data (native format of the sampling cards).

- *Info file*. An *Info* file containing the system configuration settings is generated automatically for each measurement and the user interface allows to enter additional descriptive text.

- *Temporary file*. A temporary file is employed to store the serial number of the last generated Sample file of the day. This file assures that the channel sounder can keep track of the daily serial numbers.

- *Matlab file.* In order to analyze the measured data the raw sample file can be converted into a format readable by other type of software. Here Matlab file format was chosen and a file converter program enables to create Matlab files of user specified size.

4.- SYSTEM DESCRIPTION.

In this chapter the whole system operation will be presented, i.e. the necessary processing that has to be done to the measured data in order to adapt them to the simulation, and the different ways for achieving this.

4.1.- Measured Data and Simulation Parameters.

In this section a description of the parameters that define the measured data is given, followed by the required simulation parameters. This is necessary in order to explain all the signal processing that should be done to the measured data before being used in the simulation.

The basic parameters of the measured set of data are:

- Carrier frequency (f_c).
- Space (Time) sampling rate: The number of IRs *measured* per wavelength (IRs/λ), represented by f_{st}^{m}.
- Delay sampling rate: Sampling rate of the measurement system, i.e. the time-delay spacing between two consecutive samples of an IR, represented by $f_{s\tau}$.
- Chip rate of the sounding signal (f_{chip}).

A typical IR measured and processed through a matched filter is depicted in Fig. 4.1 where these parameters were f_c =2.154 GHz., f_{st}^{m}=50 IRs/λ, $f_{s\tau}$ =120 MHz, f_{chip} = 30 MHz.

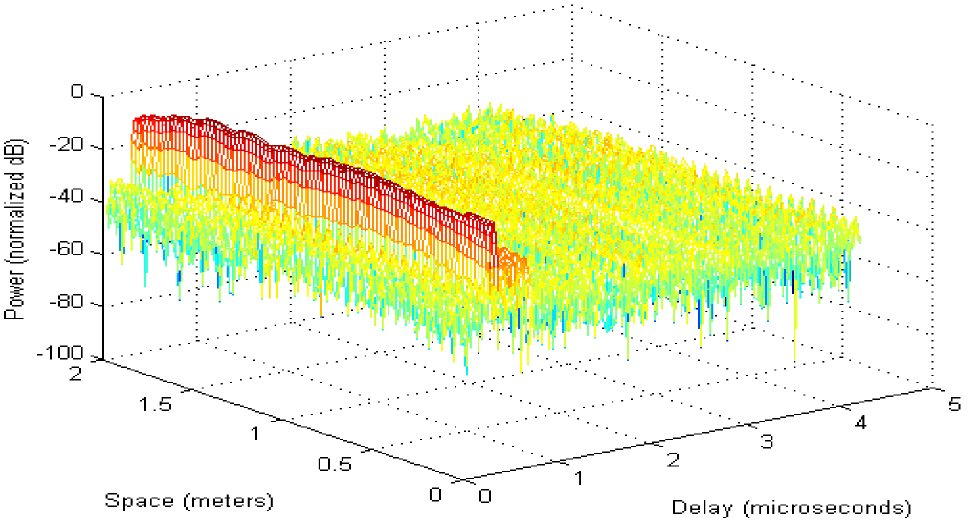


Figure 4.1. *Measured IRs*

It is important to make clear the difference between time domain and delay domain, even though both are a measure of time. When it is referred to delay domain it means the time axis of an individual IR, i.e. the delay instants of an IR. On the other hand the term time domain refers to the different number of IRs measured in a time unit. This difference can be clearly seen in Fig.4.1. where the time domain is represented in the Space axis and the delay domain in the Delay axis.

The ready-to-simulate set of data will be extracted from the measured data after a signal processing defined by these measurement parameters and the simulation parameters. The basic simulation parameters will be:

- Speed of the mobile (v).
- Sampling rate of the simulation system (f_s).
- Signal to noise ratio (SNR).
- Number of IRs/λ required for the simulation (f_{st}).

Given all these parameters (measurement and simulation) a concrete signal processing procedure will be applied to the measured data to obtain the data that can be used in the simulations. In the next sections the theory of these parameters based signal processing will be explained.

4.2. Overall Description.

In Fig. 4.2 the description of the whole process can be seen, involving the measurement equipment and the further signal processing that has to be done for adapting the data to the simulation. The transmitter contains a Pseudo Random Binary Source (PRBS), whose binary signal is coded and modulated and then transmitted over the radio channel. The received signal is measured by the sounder. If the mode in which the sounder is working is "Sliding correlator" (see section 3.2), then this signal will be the IR and there is no need of processing in the IR estimator block, otherwise the data have to be processed in order to obtain the estimated IRs

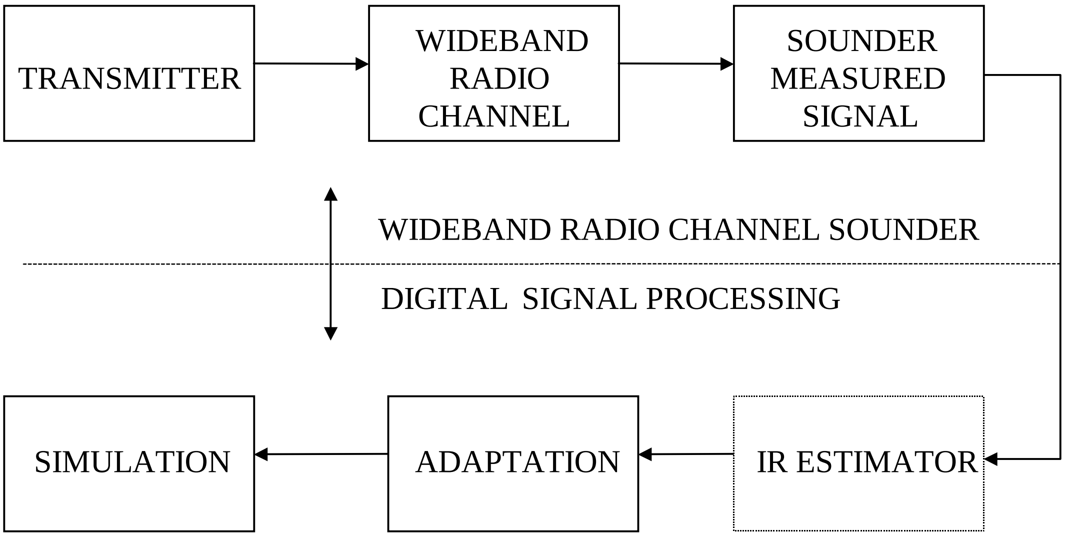


Figure 4.2. *Overall description.*

Once the IRs have been calculated, they have to be adapted to the FIR filter simulation model, explained in Chapter 2. This adaptation will consist mainly of three parts:

- Interpolation of IRs to the simulation requirements.
- Decimation of each IR to the simulation sampling rate.
- Definition of the taps, i.e. removal of wrong or noisy taps.

These three blocks will be explained in the last part of this chapter. After these processes the IRs of the wideband radio channel are ready to be used in the simulations, representing the dispersive time-variant behaviour of the channel.

4.3. Transmitter.

The structure of the transmitter is as depicted in Fig. 4.3. The binary source produces maximal length pseudo random binary sequences, known as *m-sequences* (see [8]). The *m-sequences* were chosen due to their good autocorrelation properties, close to an ideal impulse, thus allowing resolving the multipaths. This sequence is coded, modulated onto a RF carrier and transmitted over the channel.

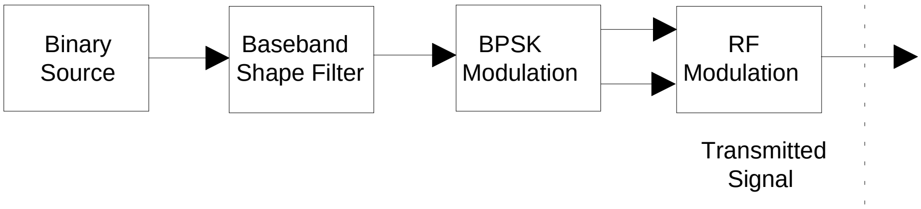


Figure 4.3. *Transmitter structure*

4.4. Multipath Wideband Radio Channel.

A description of the Multipath Wideband Radio Channel (MWRC) will be presented, followed by its effect on the transmitted signal.

In order to describe the behaviour of the MWRC, its effect on a transmitted signal will be analysed, representing then the impulse response of a generic channel. The main effects that will characterise the channel will be:

- *Time spread*: If an ideal pulse transmitted over the time-variant multipath channel, the received signal appears as a train of pulses at different time instants, each one corresponding to a different path followed by the transmitted signal.

- *Time-variant behaviour*: The physical medium that forms the channel is not constant, thus the characteristic of the channel will be variant. This can be observed if the sounding code is repeated several times: the received signal will have changes in the amplitude, phase and relative delays between the received pulses.

- *Random frequency modulation* due to varying Doppler shifts on different multipath signals, caused by the relative movement between transmitter and receiver and by the movement of surrounding objects.

Considering a general transmitted signal and based on [5]:

$$s(t) = \Re\left[u(t)e^{j2\pi f_c t}\right] \tag{4.1}$$

where *u(t)* is the complex-valued equivalent low-pass signal of *s(t),* $\Re$ denotes the real part, and f_c the carrier frequency, and the effects of the channel explained above, the received signal is:

$$y(t) = \sum_{n=0}^{M(t)-1} \alpha_n(t) s\left[t - \tau_n(t)\right] \tag{4.2}$$

where *M(t)* denotes the total number of paths, whose number varies with the time, $\alpha_n(t)$ the attenuation factor on the *n*th path and $\tau_n(t)$ is the delay for the *n*th path.

Substituting (4.1) in (4.2) yields to:

$$y(t) = \Re\left[\left\{\sum_{n=0}^{M(t)-1} \alpha_n(t) e^{-j2\pi f_c \tau_n(t)} u\left[t - \tau_n(t)\right]\right\} e^{j2\pi f_c t}\right] \tag{4.3}$$

Analysing the equivalent low-pass yields to:

$$r(t) = \sum_{n=0}^{M(t)-1} \alpha_n(t) e^{-j2\pi f_c \tau_n(t)} u\left[t - \tau_n(t)\right] \tag{4.4}$$

Since *r(t)* represents the low-pass equivalent received signal when *u(t)* is transmitted, it can be inferred that the received signal when an impulse is transmitted is:

$$h(\tau; t) = \sum_{n=0}^{M(t)-1} \alpha_n(t) e^{-j2\pi f_c \tau_n(t)} \delta\left[t - \tau_n(t)\right] \tag{4.5}$$

where $\delta(\bullet)$ is the Dirac delta function.

The continuous time-variant impulse response of the channel, resulting from excitation pulses at time instances $t - \tau$, is described by Eq. (4.5) If the channel impulse response is assumed to be wide sense stationary over a small-scale time or distance travel, then the impulse response may be simplified as:

$$h(\tau) = \sum_{n=0}^{M-1} \alpha_n e^{-j2\pi f_c \tau_n} \delta(\tau - \tau_n) \tag{4.6}$$

Most practical radio channels can be regarded as quasi wide sense stationary, thus each impulse response may be assumed as in Eq.(4.6) [7]

The characteristics of a fading multipath channel will be defined by correlation functions and power spectral density functions. The main functions and parameters described by them are [5]:

- *Delay Power Spectrum*. It's the autocorrelation function of the equivalent low pass impulse response of the channel. The range of values where it is essentially nonzero is called *multipath spread* of the channel, and is denoted by T_m.

- *Autocorrelation function of the PDP in the frequency domain.* It provides a measure of the *coherence bandwidth* of the channel $(\Delta f)_c$. Thus two sinusoids with frequency separation greater than $(\Delta f)_c$ are affected differently by the channel.

As a result of the Fourier transform realationship between the two functions above, it can be considered that:

$$(\Delta f)_c \approx \frac{1}{T_m} \tag{4.7}$$

- *Doppler power spectrum of the channel.* The range of values over which the Doppler power spectrum function is essentially nonzero is called the *Doppler spread B_d of the channel.* The reciprocal of the Doppler spread is a measure of the coherence time of the channel. That is:

$$(\Delta t)_c \approx \frac{1}{B_d} \tag{4.8}$$

where $(\Delta t)_c$ denotes the *coherence time*.

This way the time dispersive nature of the channel in a local area will be described by the delay spread and coherence bandwidth, while the time variant nature of the channel in a small-scale region will be given by the Doppler spread and the coherence time.

4.5. IR Estimator.

Different IR estimator approaches can be used for estimating the IR of the wideband radio channel. Usually channel estimation is done directly by a Matched Filter (MF), matched to the transmitted sequence.

The MF is the optimum detection method in Gaussian linear channel [5], but an important drawback is that its time-delay resolution is limited to the sequence chip duration. In addition, possible subchip delay components have a tendency to shift peak positions and amplitudes.

When higher resolution is needed one possibility is increasing the transmitted sequence rate, but it requires additional cost and hardware complexity. Thus different techniques are used to improve the resolution without increasing the chip rate. These techniques are known as superresolution techniques and will be presented later in this chapter.

It is important to note that the IR estimator will give a first estimation of the IRs (number of taps, delay instants and amplitude of each tap). For improving the estimation of the channel further post processing techniques will have to be applied, such as superresolution, both in delay and angular domain.

In order to analyse the IR estimator configuration, firstly it will be shown how the transmitted signal is corrupted by the channel to form the received signal that will be processed by the IR estimator.

The transmitted *m*-sequence, consisting on *K* chips, has a LP model:

$$s(t) = \sum_{i=1}^{K} V_i \, p(t - t_c) \tag{4.9}$$

where $p(t)$ is the pulse shape, V_i is the amplitude depending on the modulation scheme (e.g. for BPSK $V_i = \pm 1$) and t_c is the chip duration. Considering the channel given by Eq.(4.6), and additive white Gaussian noise, the received signal is

$$r(t) = s(t) \otimes h(t) + n_c(t) = \sum_{n=0}^{M-1} \alpha_n e^{-j2\pi f_c \tau_n} s(t - \tau_n) + n_c(t) \tag{4.10}$$

where $\otimes$ represents the convolution sum.

The different processes to estimate an IR from the received signal will be presented next.

4.5.1. Matched Filter.

The output of the matched filter *m(t)* can be expressed as the convolution of the autocorrelation function *a(t)* of the probing signal *s(t)* with the channel IR, i.e.,

$$m(t) = s(-t) \otimes r(t) = s(-t) \otimes s(t) \otimes h(t) + s(-t) \otimes n_c(t) = a(t) \otimes h(t) + v(t) \tag{4.11}$$

where $v(t) \equiv s(-t) \otimes n_c(t)$ denotes the effective noise component. Since the autocorrelation function of the probing signal is *a priori* known, the output of the matched filter can be deconvolved to estimate the channel IR.

As stated in [20] matched filtering provides a resolution limited by the shape of the probing signal's autocorrelation peak

4.5.2. Deconvolution Techniques.

Here the restoring of the channel estimate by deconvolution is considered. Due to the assumption of the linear time-invariant property of the channel, one can solve the channel transfer function estimate by Fourier transform

$$\hat{h}(t) = \int_{-\infty}^{\infty} H(f) e^{j2\pi ft} df$$

where the channel frequency response can be calculated by deconvolution

$$\hat{H}(f) = \frac{Y(f)}{S(f)H_R(f) + \gamma} \tag{4.12}$$

where $H_R(f)$ is the frequency response of the receiver filter, *Y(f)* is the output of the filter, *S(f)* is the input and γ is the deconvolution constant. Depending on the filter used there are several possibilities used for improving the unfiltered deconvolution SNR, as it is shown in [11]. The deconvolution constant provides better stability on the frequency response when values close to zero are used for γ in the denominator of Eq. (4.12).

Matched Filter Deconvolution.

Considering the restoring of the channel estimate by deconvolution, the channel frequency response can be calculated by:

$$\hat{H}(f) = \frac{Y(f)}{S(f)H_{MF}(f) + \gamma} \tag{4.13}$$

where $H_{MF}(f) = \mathcal{F}(h_{MF}(t))$, $\mathcal{F}$ denotes the Fourier transform and $h_{MF}(t) = s(-t)$ the response of the matched filter.

This method of (direct) inverse filtering yields a channel estimate whose resolution is the sample resolution of the signal $s_y(t)$ [11]. Thus, the resolution is in principle independent of the chip rate.

A price to be paid for increasing the resolution is the noise sensitivity leading to limitation in the dynamic range. Adequate values of the deconvolution constant γ have to be used to make it as noise insensitive as possible. In order to improve the deconvolution performance matched filtering is applied.

Wiener Filtering.

Besides matched filtering, one can improve deconvolution performance by Wiener filtering. Here the aim is to design a pre-deconvolution filter such that it can remove as much as possible of out-of-band noise before the deconvolution is calculated. Thus defining the filter frequency response as $\Phi(f)$ yields:

$$\hat{H}(f) = \frac{\Phi(f)Y(f)}{S(f)H_R(f)+\gamma} \tag{4.14}$$

where the filter $\Phi(f)$ is calculated by minimizing the least-square error of the ideal and estimated channel impulse response, yielding

$$\Phi(f) = \frac{|S(f)|^2}{|S(f)|^2 + |N(f)|^2} \tag{4.15}$$

This method requires the knowledge of *S(f)* and *N(f)*, they can be determined as presented in [11].

4.6. Adaptation to Simulation.

In this section all the necessary processing that the measured IR signal has to go through for being adapted to the simulations will be explained. This section will be focused on adapting the measured IR to a FIR filter radio channel model, also known as tapped delay line, as it was presented in Section 2.2.1. This model is widely used for wideband radio channel simulations.

Mainly two procedures have to be performed:

- Interpolation of IRs in time domain.
- Resampling of each IR in delay domain to match the sampling moments of the simulation model and the IRs.

Initially the order is arbitrary, but it is a task out of the scope of this thesis to test if there is any significant effect on the simulation due to the order in which these procedures are applied. Next they will be explained deeper.

4.6.1. Interpolation in Time Domain.

In this section the necessity of IR interpolation will be presented first, followed by the different existing methods.

As stated in Section 4.1. one of the parameters of the measured data is the number of IRs measured per wavelength (f_{st}^{m}). This parameter is usually limited by the sounder system. As it was explained in the previous chapter the onboard buffer memory of the data acquisition unit in the IRC-WMRCS is limited to 8 MB (4 MB for both I and Q components, sampled separately but in a synchronised way) so that when it is full the measurement is stopped, the data downloaded to the hard disk and the sounder is ready for another measurement. This way several "buffered measurements" can be concatenated to form a set of measurements representing a spatial distance, whenever the environment can be considered as stable between two "buffered measurements".

In order not to repeat this downloading of the measured data too many times for a wanted spatial distance, the parameter f_{st}^{m} is set to a reasonable value (commonly 2-5 IRs/λ), and the required number of IRs for the simulations (given by f_{st}) is achieved via interpolation.

Further versions of the sounder will make the data transfer to the hard disk automatically and in a reduced time (order of milliseconds or less). In this case the interpolation would seem as unnecessary, if it would be done only for the reasons explained above, but sampling at large f_{st}^{m} rate would create a huge amount of data, and it's always easier to handle smaller amount of data and generate by interpolation the rest of the data needed for the simulation.

In the next sections the interpolation methods will be explained and an example of one of them with actual measured data will be presented.

4.6.1.1. Methods.

Interpolation is a commonly used procedure for reconstructing a function either approximately or exactly from samples. As presented in [26] for a bandlimited signal, if the sampling instants are sufficiently close, then the signal can be reconstructed exactly, i.e. through the use of an ideal lowpass filter exact interpolation can be carried out between the sample points. The effect in time domain of this filtering can be considered as an interpolation using the sinc function. This interpolation using the sinc function is commonly referred to as *bandlimited interpolation* since it implements exact reconstruction if the signal is bandlimited and the sampling frequency satisfies the conditions of the sampling theorem.

Since a very good approximation to an ideal low pass filter is relatively difficult to implement, in many cases it is preferable to use a less accurate but simpler method, or equivalently interpolating function, as will be presented next.

Interpolation is a subject widely known and studied, thus in this section the different methods will be briefly quoted, with the drawbacks and the advantages of each one. For the description of each method, refer to [12].

Polynomial interpolation.

The interpolating polynomial *P* is the polynomial of least degree that satisfies, for a function *f*,

$$P(x_i) = f(x_i) \text{ for each } i = 0,1,..,n$$

in the case presented in this thesis, this function will correspond to the one presented in the example.

Although there is a unique interpolating polynomial, it can take many different forms. The *Lagrange* form is most often used for interpolating tables when *n* is small. *Neville's* method is used for evaluating several interpolating polynomials at the same value of *x*. *Newton's* forms of the polynomial are more appropriate for computation. However *polynomial interpolation* has the inherent weakness of oscillation, particularly if the number of nodes is large. In this case there are other methods that can be better applied.

The *Hermite* polynomials interpolate a function and its derivative at the nodes. These polynomials are very accurate, but they require more information about the function to be approximated. For large number of nodes, the Hermite polynomials also exhibit oscillation weakness.

The most commonly used form of polynomial interpolation is *piecewise polynomial interpolation*. *Cubic Hermite* is recommended when the function and derivative values are available. When only the function values are available, as it will be the case of the measured data used for interpolating, *free cubic spline* interpolation is recommended. This spline forces the second derivative to be zero at the endpoints. Other cubic splines require additional information, such as the *clamped cubic spline*, that needs the values of the derivative function at the endpoints of the interval.

There are other methods of interpolation that are commonly used. *Trigonometric* interpolation is used with large amounts of data when the function has a periodic nature. Interpolation by *rational functions* is also used. If the data are suspected to be inaccurate, smoothing techniques can be applied.

4.6.1.2. Practical Example.

An example of the interpolation method will be presented, applied to measured data. This example was used to test the accuracy of the interpolation method that was going to be used in the first phase of this work. The interpolation method under investigation was the *free cubic spline* method. As recommended in [12], this method should be used when we only have the values of the function to interpolate, as it happens in this case.

In similar works that had to deal with the same task presented here this method was used for the interpolation [13]. It is as well used in the simulators, when the interpolation is done online, as it will be presented in Chapter 5.

The testing procedure consisted on measuring IRs at a f_{st}^{m} = 50 IRs/λ rate, decimating them to 5 IRs/λ, by just taking one IR out of ten from the original data, interpolating the new IRs by the method under study, and comparing the original data with the generated. The interpolation was made by separately interpolating the real and imaginary part of each IR. The results can be seen in the next figures.

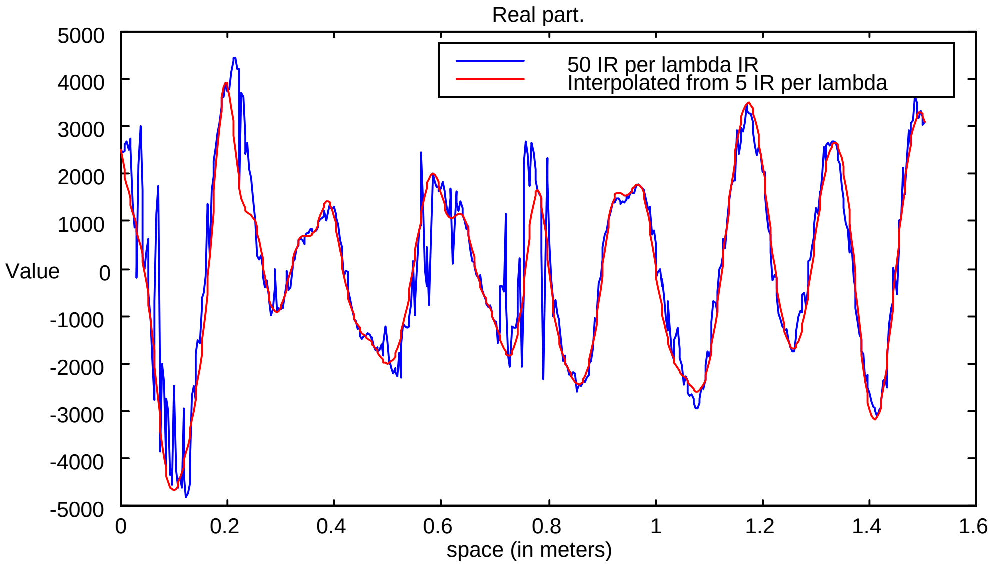


Figure 4.4. *Original and Interpolated real part. Cubic spline interpolation, from 5 IR per wavelength to 50. Strongest path signal.*

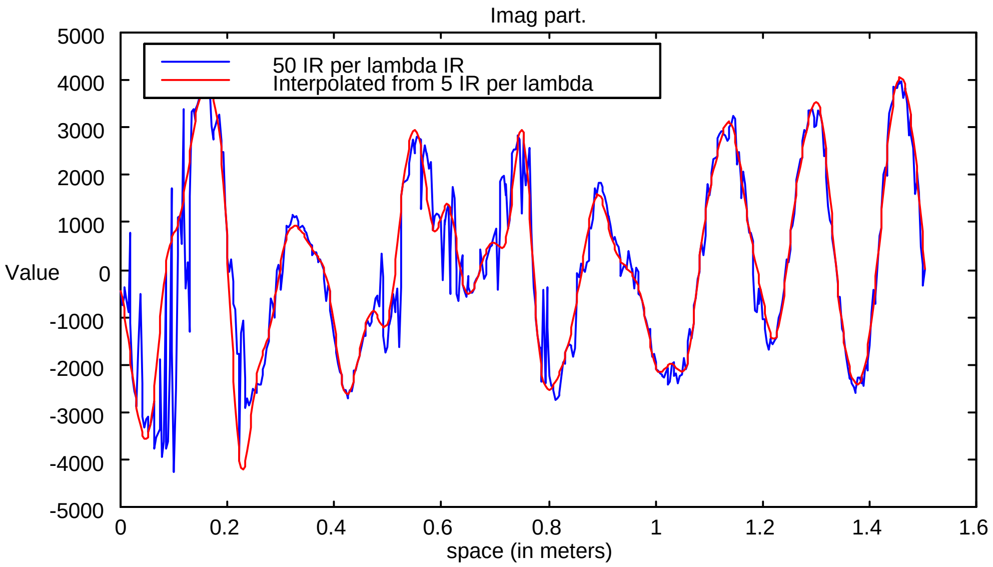


Figure 4.5. *Original and Interpolated imaginary part. Cubic spline interpolation, from 5 IR per wavelength to 50. Strongest path signal.*

As it can be seen from figures 4.4 and 4.5, corresponding to the strongest path component, the interpolated signal matches quite well the measured signal, except in the abrupt variations, that were caused by the high sampling rate of the sounder system, and may be considered as measurement errors. Actually, the interpolation smoothens these errors. So as a starting point this method seems to be valid, further research should investigate other methods.

4.6.2. Resampling.

As described in the beginning of this chapter the measured IRs consist of a discrete sequence that has been taken at a sampling rate of $f_{s\tau}$. In delay domain this represents a delay spacing of *1/* $f_{s\tau}$ between two consecutive samples.

On the other hand the input sequence of the data that will be used in the system simulation has a delay spacing corresponding to $1/f_s$, where f_s is the sampling rate of the input data. As a stream driven simulator is going to be used, the samples must correspond to the same time instant, so there is a need of resampling the IR sequence to the f_s sampling rate in order to adapt the time instants of both the IR sequence and the simulation input sequence.

Summarising, the delay axis of the IRs doesn't match the delay axis of the sampling moments, so there is a need for resampling to match them. This must be done with decimation of the taps to the sampling instants (usually the resampling will consist on a decimation of the data, i.e. sampling at a lower rate, for $f_{s\tau}$ is normally greater than f_s).

Now that the resampling necessity is explained, the effect that this operation will have on the signal will be seen, both in time-delay domain and in frequency domain. The most important block in the decimation process will be the anti aliasing filter used before decimation. In the next section the different options and the effects of each one will be presented.

4.6.2.1. Anti Aliasing Filter.

Before designing the anti aliasing filter the most important thing that should be considered is that the overall frequency response of the simulation system must not be modified by this filter used in the resampling process. So first of all this overall response should be considered.

The first thing to take into account is the bandwidth of this overall frequency response. This will be given by the minimum bandwidth filter used in the simulation, and it corresponds to the one used in the receiver, whose bandwidth corresponds to the symbol rate. But if oversampling is used in the simulations, with a given factor *K*, then the sampling rate of the signal at the input of the radio channel block will be *K* times greater than the one at the input of the receiver block, and this means that the bandwidth of the resampling filter is *K* times greater than the one in the receiver. So the resampling filter has to be designed in such a way that inside this bandwidth the frequency response is flat.

Now that the first part of the filter design is presented, the conditions for designing the rest of the filter have to be set, i.e. the part outside the frequency band of interest. For this task a criterion about the “goodness” of the filtering should be established in order to discard the wrong approaches.

The main requirement, added to the one explained before, is that the filtering has to keep the information contained in the IR.. Here two different types of filters may be mainly used for this purpose:

- *Ideal Low-Pass Filter* because it has flat frequency response and zero phase.

- *FIR filter* because it has linear phase, and in the band of interest its frequency response may be considered as flat. (Here different kind of FIR filters can be designed).

So now the problem is reduced to decide the criterion that should be used for comparing these two possibilities.

Criterion for the filter design.

In filtering changes will be done to the shape of the signal that will affect the delay domain and frequency domain responses. So it has to be found out which one of this changes is going to be critical for the simulations.

The first thing that can be clearly seen is that when the sampling to the desired rate is done, after filtering the values of the new taps don’t match the ones we would have in the original signal because the delay domain signal is now rather different from the original (see Figure 3, with ideal low-pass filtering). So it could be considered as an error because of this mismatching.

But here a question arises, what represents the “original” signal?. The “original” signal represents the actual IR, filtered by the sounder pre-sampling filter, and then sampled at a rate of $f_{s\tau}$. But if the actual IR would be sampled at a rate of f_s, these data could be used directly in the simulations because there would be no need for resampling, and the values would be the correct ones, so this signal should be the one to compare with.

The shape of this new signal would be different from the one obtained at a rate of $f_{s\tau}$, because a different pre-sampling filter would be used (adapted to the f_s rate), so the mismatching referred to cannot be considered as a mistake because of this. This way a criterion for the goodness of the filter cannot be stated beforehand, the only way would be comparing the performance of the different options.

Next the features of these two filters and examples of filtered IRs will be presented.

Ideal Low-pass filter.

An ideal low-pass filter (ILPF) frequency response is given by:

$$G(\omega) = \begin{cases} \dfrac{1}{\omega_s} & -\omega_s/2 \le \omega \le \omega_s/2 \\ 0 & \text{otherwise} \end{cases} \qquad (4.16)$$

where the gain factor $\dfrac{1}{\omega_s}$ is used for convenience.

The equivalent representation in time-delay domain is:

$$g(\tau) = \mathcal{F}^{-1}(G(\omega)) = sinc(\tau f_s) \qquad (4.17)$$

where $\mathcal{F}^{-1}$ represents the inverse Fourier transform, sinc(x)=sin(πx)/πx and $\omega_s = 2\pi f_s$.

Considering the measured IR in the delay domain (sampling rate $f_{s\tau}$)

$$h(\tau) = \sum_{k=1}^{M} \alpha(\tau_k)\delta(\tau-\tau_k) \qquad (4.18)$$

where α represents the amplitude function, $\delta(.)$ the dirac-delta function, defined as:

$$\delta(\tau - \tau_k) = \begin{cases} 1 & if \quad \tau = \tau_k \\ 0 & \text{otherwise} \end{cases} \qquad (4.19)$$

and $\tau_k = kT_{s\tau} = k/\ f_{s\tau}$, M the number of samples per IR.

The filtering of the IR by an ideal low-pass filter of bandwidth ω_s results:

$$R(\omega) = H(\omega)G(\omega) \qquad (4.20)$$

where $H(\omega) = \mathcal{F}(h(\tau))$, Fourier transform of the IR.

The equivalent in delay domain is:

$$r(\tau) = \mathcal{F}^{-1}((R(\omega)) = h(\tau)g(\tau) = h(\tau) \otimes \mathrm{sinc}(\tau f_s) =$$

$$= \sum_{j=1}^{M} \left[\alpha(\tau_j)\mathrm{sinc}(\frac{\tau - \tau_j}{T_s}) \right] \qquad (4.21)$$

where $\otimes$ denotes the convolution sum and $T_s = 1/f_s$.

The main effect is the creation of side-lobes caused by the shape of the sinc function as can be seen in Figure 4.6.

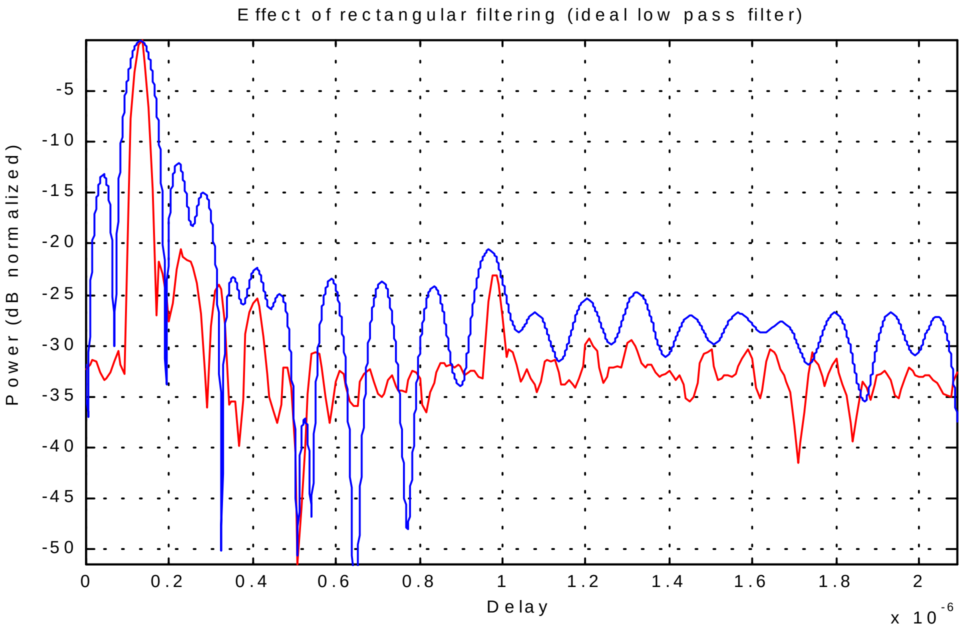


Figure 4.6. *Effect of the ideal low-pass filter in the delay domain resampling.*

This filtering has already been used in simulations with a similar purpose as in this one (resampling) [14] .

FIR filter.

For the design of a FIR filter several parameters have to be specified:

- Order of the filter (with FIR filters if a strong attenuation outside the frequency band of interest is desired, a large order FIR filter is needed).
- Cut-off frequency.
- Window (when windowing method is used).

There are several options for implementing this filter, but the most reasonable is the one that has the best frequency response for our purposes This would be the Ideal Low-Pass Filter (ILPF). But its ideal feature makes it not realisable, and it causes as well the effects mentioned in the former section. So another possibility is presented here.

The tap coefficients of the FIR filter are calculated using a time-delay domain windowing over the ILPF response in time domain (i.e. sinc-pulse in time-delay domain). This is done because a finite duration frequency response, such as the ILPF, results in a infinite duration in time-delay domain (e.g. sinc-pulse). To limit its duration it is multiplied by a finite time-duration window function. This is just a way for converting the ideal frequency response into a realisable one.

The effect of this truncation in frequency domain is an undesirable overshoot that we will try to minimise by using different window functions. In this design there is a trade-off between band-pass region smoothness, stop-band attenuation and width of the transition region from pass to stop-band. The different options are presented in [15].

In Figures 4.7 and 4.8 an example of the filtering and further resampling can be seen (from 120 MHz to 120/7=17.1429 MHz) using a 100th order FIR filter, with Hamming window and 17.1429 MHz bandwidth.

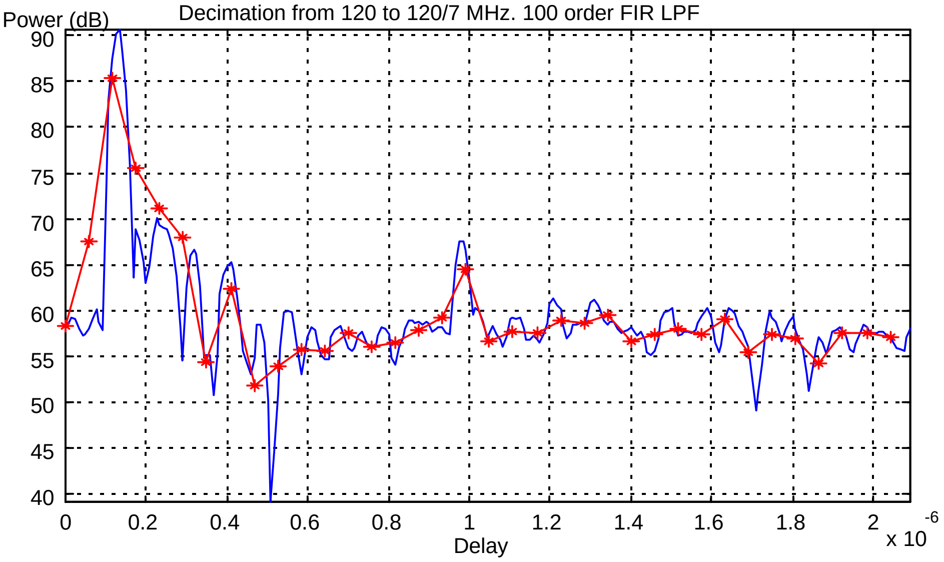


Figure 4.7. *Filtering and resampling from 120 to 17.1429 MHz. 100th. order FIR*

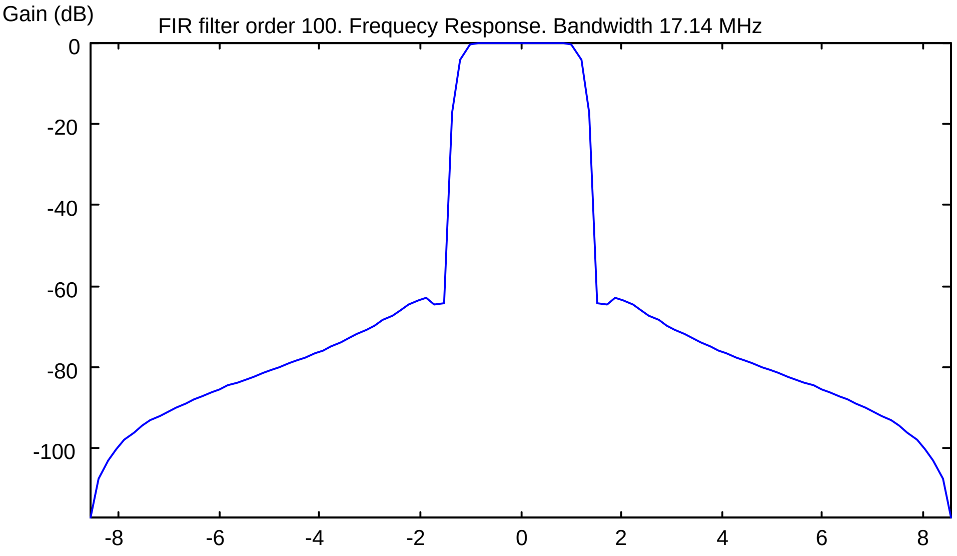


Figure 4.8. *100th. order FIR filter. Frequency representation.*

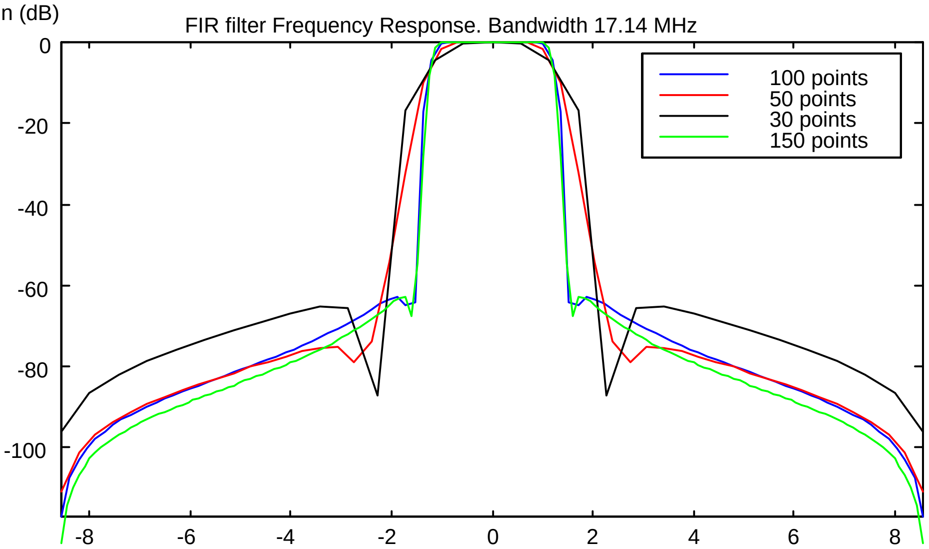


Figure 4.9. *Effect of the filter order in the frequency response*

In Fig.4.9 the effect of the filter order in the frequency response can be seen. The higher the order is, the sharper is also the frequency response. It can be seen that increasing the order from 50 to 100 produces a clear improvement, while increasing from 100 to 150 there is only a very slight improvement.

Further possibilities to investigate about FIR filtering, out of the scope of this thesis, would be combining different filter orders with different windows.

From the figures above it can be seen that the frequency response can be considered flat in the region of interest. But as it was said there are many possibilities when creating a filter of this type, so the different options should be investigated.

4.7. Tap Definition.

After the processing of the measured data explained throughout Chapter 4, the available data at this point consists on a set of IRs interpolated to a certain number of IRs/λ and each one consisting of a certain number of taps whose separation matches the simulation sampling interval. These taps could be already used in the simulation but wrong channel behaviour would be included because the noise effect of the measured channel and the noise of the measurement system are included in this data (note here that the measured channel noise is inherent to the channel and if one wants to represent the real behaviour of the channel it should be removed, but in the simulations the noise will be included after the radio channel as an Additive White Gaussian Noise (AWGN)). Thus a tap defining method is necessary in order to improve the accuracy of the processed data.

Two possibilities arise for defining the taps:

- Establishing a noise threshold and removing all the taps under this threshold.
- Superresolution methods in delay or angular domain.

Superresolution methods are usually very complex to implement, but yield to good results, while noise threshold is the most simple one, but not with so good results. As a first approach for testing the system simulation presented in this thesis a noise threshold is suitable to be implemented, as it will be done in this work. Further investigation should include superresolution methods, which will be presented in the next section.

Noise threshold technique consists on considering the samples below a certain threshold from the strongest component as noise. The technique is simple, but the criteria for establishing a certain threshold value is not easy because of the random nature of the noise, typical values used are 10-20 dB. Anyway it is very difficult to separate noise from signal, and when applying this technique it is normal that signal samples are removed and noise samples are left. An example of a noise threshold technique can be seen in figure 4.10, where the threshold was 20 dB. For clarity in the representation the noisy samples have been set to the threshold value (where the value is in dB), but the actual value to be set when the data is going to be used in the simulations is zero.

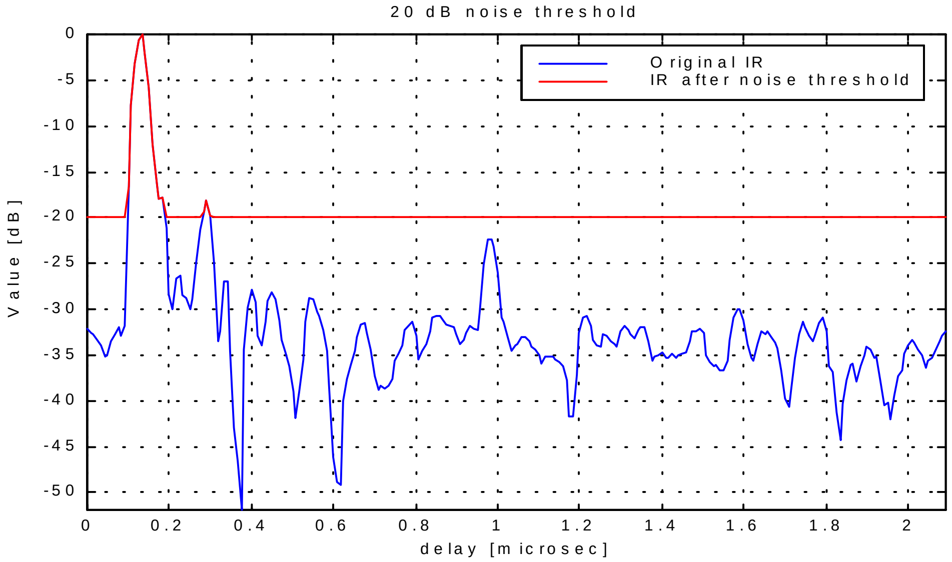


Figure 4.10. *Noise threshold technique.*

4.7.1. Superresolution Methods.

As it was stated in Section 4.5. the resolution of the measurement system is limited by the chip frequency of the sounding code. The nominal resolution is defined as the chip duration, thus in the IRC-WMRCS the best resolution is:

$$\Delta r = c * T_{chip} = \frac{c}{f_{chip}} = \frac{3*10^{8}\mathrm{m/s}}{60*10^{6}\mathrm{Hz}} = 5\ \mathrm{m} \tag{4.22}$$

For outdoor environments the reflected rays causing multipath are usually separated by distances greater than 5 m, thus this resolution is enough for resolving the multipath, but in indoor environment these distances are smaller, thus high-resolution IR should be calculated. For this purpose two options can be applied:

- Increasing the chip rate.
- Techniques for achieving sub-chip resolution by signal processing and keeping the same chip rate.

The first option requires more complex sounder hardware structure, and consequently additional cost. The second one are the techniques known as superresolution (SR) methods. These methods are widely studied for achieving SR at both delay and angular domain. In this work several of the proposed methods will be briefly described for both delay and angular SR.

Traditionally the SR methods have been applied to the angular domain, to achieve more accurate values of the Direction Of Arrival (DOA), given by the azimuth and elevation angles. Because of the similar relationship between Array domain-DOA and Delay domain-Frequency domain these methods can be applied to the delay domain resolution. Due to the extensive study in this field the techniques will be briefly described, along with the pros and cons.

The main goal of these methods is the estimation of parameters by fusing temporal and spatial information, captured via sampling a wavefield with a set of judiciously placed antenna sensors. They are known as *parameter estimation techniques*. As presented in [22], they can be classified into two main categories: *Spectral-based* and *Parametric* approaches.

In the ***spectral-based*** approach a spectrum-like function of the parameter of interest is formed. The locations of the highest (separated) peaks of the function in question are recorded as the parameter estimates. Spectral-based methods can be classified into:

- Beamforming techniques:
  - Conventional Beamforming.
  - Capon's beamformer.

- Subspace-based methods:
    - MUltiple SIgnal Classification (MUSIC) Method.
    - Extensions to MUSIC.
    - Resolution-Enhanced Spectral Based Methods.

A solution proposed for achieving better resolution is to couple MUSIC algorithm with some spatial prefiltering, to result in what's known as *Beamspace Processing*. Other attempts to improve MUSIC are quoted in [22], based on different modifications of the criterion function.

The ***Parametric*** approach requires a simultaneous search for all parameters of interest. While the spectral-based methods are computationally attractive, they do not always yield to the required accuracy. In particular, for scenarios involving highly correlated signals, the performance of spectral-based methods may be insufficient. An alternative is to more fully exploit the underlying data model, leading to the parametric array processing methods. The price to pay for the increased efficiency and robustness is that the algorithms usually require a multidimensional search to find the estimates. The most commonly used algorithms are:

- Deterministic Maximum Likelihood.
- Stochastic Maximum Likelihood.

The main difference between these two methods is in the assumptions about the transmitter signals, considering the transmitter signal waveforms as deterministic in the first case, and as a Gaussian random processes in the second. Parametric estimation methods are computationally quite complex. However, for Uniform Linear Arrays (ULA) a number of less demanding algorithms are known, as described in [22]. One of the most popular is the Estimation of Signal Parameters via Rotational Invariance Techniques (ESPRIT) algorithm.

The methods presented above are essentially limited to processing narrowband data. For wideband data a natural extension of all these methods is to employ narrowband filtering, for example using FFT. An optimal exploitation of the data entails combining information from different frequency bins. A different suboptimal approach is to process the different FFT channels separately using a standard narrowband method, whereafter the parameter estimates at different bins must be combined in some appropriate way. Another approach is to explicitly model the array output as a multidimensional time series, using Autoregressive Moving Average (ARMA) model.

Besides the methods described above a novel algorithm approach for delay domain superresolution will be presented next . The motivation for presenting this approach is the probable implementation of this technique in further stages of this work, or at least a suboptimal approach. This implementation is out of the scope of this work, for only the basic link between the measured data and the simulator will be presented here.

MFD-SBT Algorithm.

The Matched Filter Deconvolution (MFD) - Sequential Bin Tuning (SBT) algorithm is presented in [23]. Its performance is compared to the Maximum Entropy Method (MEM) [24] and to the MUSIC method [25], observing in the MFD-SBT a significant accuracy improvement. The simulations suggest that the algorithm is applicable in processing direct sequence correlator - based data when the SNR>20 dB and comparable to the model tap dynamic range.

In the MFD-SBT algorithm the tap delays and amplitudes are estimated in two main steps. The Dominant Paths (DPs) are extracted first, this guarantees that their bin delay times are likely to be correct and thus the SBT works efficiently. The second step is the addition of the smaller taps, as it will be explained later.

The extraction of the dominant paths is done in the following way: an initial estimate of the IR is calculated by MFD and by MF and a combined function of both estimates is formed. The DPs are calculated by applying a moving amplitude threshold to this function and minimising the least square error (LSE) of the resulting estimate. Then the SBT tunes the tap amplitudes of the DP in sequence by starting from the largest.

Adding smaller taps is achieved by reinforcement tuning in two steps. First a reduced MFD estimate is calculated by removing the DP bins and the zero entry bins (assumed to be produced by noise) from the MFD estimate. Then the SBT is applied to the reduced MFD estimate but now the entries of the reduced MFD are fetched to the algorithm one by one.

The comparison made to MEM and MUSIC spectrum analysis techniques shows a significant performance improvement using the MFD-SBT method, as it can be seen in the graphical results given in [23].

4.8. Update Rate.

The Update Rate (R_u) is one of the most important parameters to be investigated in the simulations, as it will determine the performance of the approach presented in this work. Mainly it can be defined as how many simulated data symbols are used for the same IR.

In the classical simulation based on statistical channel models an IR is generated for each simulated data sample, but two, or even sometimes much more than two, consecutive IRs are practically identical to each other, i.e. the channel remains mainly constant for a certain number of simulated data symbols. When using measured data this property should be applied, and it is controlled by the updating rate. This value is calculated as:

$$R_u = \frac{cf_s}{Mvf_c f_{st}} \quad \text{IRs / symbol} \tag{4.23}$$

where:

- c is the speed of light.
- f_s is the sampling rate of the simulated data.
- M is the number of bits per symbol.
- f_{st} is the IRs/λ rate of the data used for modelling the radio channel, corresponding to the original measured IRs at a f_{st}^{m} IRs/λ rate, plus the interpolated IRs.
- v is the speed of the mobile.
- f_c is the carrier frequency.

The performance of the simulation will be strongly influenced by the updating rate and the IRs/λ rate. This can be clearly seen if a low IRs/λ rate is used (e.g. 4-5 IRs/λ), then the leap between one IR and the next one would be like a step, while with a rate of , e.g. 50 IRs/λ, the leap would be smoother, as can be seen in Fig. 4.11 where the signals represent the time variation of the strongest path-delay.

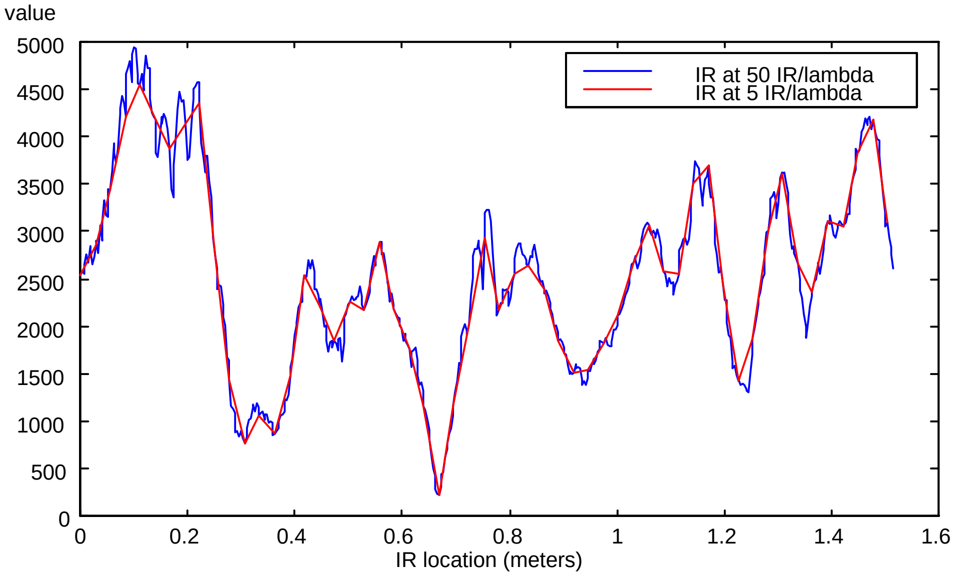


Figure 4.11. *Effect of the IRs/λ rate in the strongest path component.*

Thus simulations can be done for comparing the different suitable values. Typically the effect can be seen in the Bit Error Rate (BER) of the overall system, for it is adapted to smooth variations of consecutive channel IRs, not abrupt as would happen with a low IRs/λ rate. This effect depends on the overall simulation chain, mainly on the type of modulation used and on the receiver configuration. Implementing the overall communication system taking into account the wideband multipath nature of the radio channel is out of the scope of this thesis, thus this parameter should be tested in a further stage when this system is available. It has been noticed in hardware system simulations that the suitable value is about 30 IRs/λ.

4.9. Noise Generation.

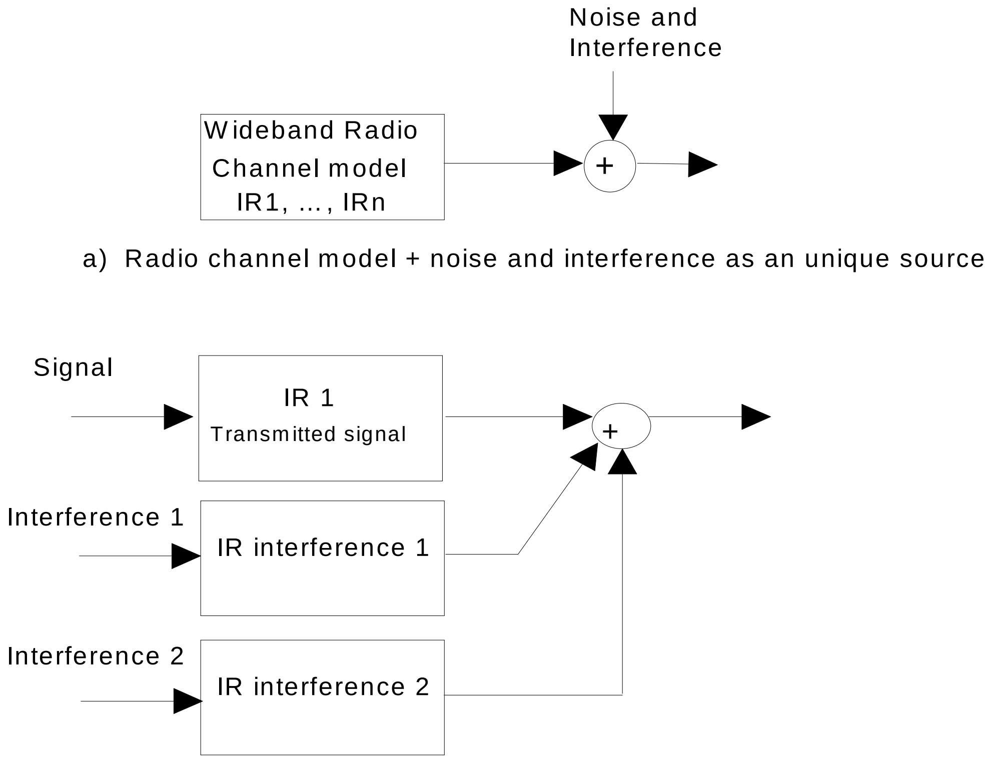
The performance of the system will be given in the last step by the SNR-BER curves of the overall system. The noise comes usually from two sources:

- The one produced by the radio channel, that is usually implemented as an Additive White Gaussian Noise (AWGN).

- The interference produced by other sources, generally considered only the co-channel interference.

Given a concrete set of IRs used for modelling the radio channel, this noise should be modelled to achieve a desired SNR value. This can be done in several ways depending on the power and gain control of the simulated system, and it is thus application dependent. One first approach for the simulations is to off-line calculate the average tap powers of the whole set of IRs being used in the simulation and then applying the correspondent value to the noise generator for achieving the desired SNR.

Further work, not presented in this thesis, will yield to different noise generation approaches dependent on the application being simulated and the mechanisms for power and gain control in the overall simulation. At this point two possible models can be implemented, as they are shown in Fig. 4.12.



a) Radio channel model + noise and interference as an unique source

b) Radio channel model for the transmitted signal and for the separate interference signals. IR chosen, e.g. randomly

Figure 4.12. *Noise and interference implementation schemes.*

5.- IMPLEMENTATION.

In this chapter different options for implementing the radio channel block in system simulations will be presented. Two different approaches are presented, the classic one based on statistical properties of the radio channel, and the one studied in this work using the measured data. This data represents the time-variant dispersive nature of the radio channel in the simulations.

As this work will be tested in COSSAP simulator, a description of it will be given, along with the description of statistical radio channel models implemented in the same simulator.

5.1. Statistical Radio Channel Model Simulations.

The main procedure for creating a statistical radio channel model is analysing measured data and extracting the statistical properties to be implemented later in the model. This method is used in a straightforward manner in the SIRCIM model, as was presented in Chapter 2.

Sometimes these statistical properties for different systems and environments are reported to standardisation bodies or study groups that publish standard models or recommendations about the radio channel models for being used in the simulation. Examples of this are GSM, where the models are based on the recommendation GSM 05.05, CDMA, where the models are based on IS-95 (as shows the COSSAP block representing the radio channel for CDMA), or in the models developed in the RACE projects CoDiT and ATDMA [19], presented in Chapter 2.

In the next section the statistical channel models implemented in COSSAP will be presented, these are the GSM and CDMA models.

5.1.1. COSSAP Simulator.

COSSAP is a complete design environment that allows to specify, analyse, simulate and implement complex DSP and communications systems. The tools of the COSSAP design environment will be categorised and briefly explained next, for further information refer to the COSSAP manuals [31].

The tools of the COSSAP simulator can be described as:

Design specification

- Block Diagram Editor (BDE)
- Blocks from the COSSAP Model Libraries.
- Own block written by the user.
- Filter design tools.

Simulation and analysis

- Stream Driven Simulator (SDS).
- Utilities to control and display a running simulation.
- Report generator tool.
- Graphical analyser that depicts simulation results.

Hardware implementation

- HDL Code Generator (VCG).
- HDL Simulation Interface (VSI).

Software implementation

- DSP Code generator (DCG).
- DSP Developer Kits (DDKs).

The entire system is specified using the COSSAP Block Diagram Editor (BDE), where blocks and interconnects represent functions and data flow. A wide selection of standard blocks is available from COSSAP's extensive model libraries. Own models can also be defined. Complex block systems can be grouped into one block by creating hierarchical blocks.

Once the design has been captured using the BDE, it can be simulated using the Stream Driven Simulator (SDS). The SDS is the key to the signal processing design system. The main features of the SDS are:

- Supports modelling at both the behavioural and architectural levels.
- Provides full support for asynchronous and multi-rate designs.
- Provides complete debugging facilities.

The driving force of the simulation is the stream of data, not time. Thus a block is not scheduled for invocation till it has sufficient signal elements available at its input port(s). When a simulation is scheduled it is not needed to "flatten" the hierarchical block diagrams into their components or combine pieces of source code for the signal flow of a flat block diagram. System simulations can be created and run within an appropriate intuitive modular environment for the job at hand.

The COSSAP HDL Code Generator (VCG) creates HDL code for direct synthesis of Integrated Circuits (IC), Field Programmable Gate Arrays (FPGA) or ASICs. HDL code generation is based on the original block diagram defined for the SDS.

The COSSAP HDL Simulation interface (VSI) provides a complete co-simulation environment for systems that are specified as a mix of high-level blocks and hardware blocks written in VHDL or Verilog. This interface supports the Synopsys VHDL System Simulator (VSS) and the Cadence Verilog XL simulator for verification of behavioural, Register Transfer Level (RTL) and gate-level designs.

The COSSAP DSP Code Generator (DCG) creates programs that execute on programmable digital signal processors. In addition to generic C code, the generator produces code targeted for floating-point DSPs. From the block diagram the DCG can automatically create a range of program architectures. The code size, throughput and latency of the generated program can be controlled by selecting a scheduling strategy, specifying the degree of macro pipelining, enabling or disabling code in-lining, and selecting a memory allocation scheme. There is a library of DSP blocks available, or own blocks can be written in C or assembly language.

The COSSAP DSP Developer Kits (DDKs) allow executing and assembling code programs in a system-level simulation environment. Co-simulation with some fixed-point DSPs is supported. The assembly code blocks are executed at the instruction level by using an instruction-set simulator, while the rest of the system is simulated in the COSSAP SDS.

5.1.1.1. COSSAP GSM.

The different radio channel models presented in COSSAP for the GSM system will be presented. They are based on the model presented in Chapter 2.

The model gives the number of taps, the delay value of each tap and its correspondent average power, and the type of Doppler spectrum of each tap. The COSSAP block only takes the propagation profiles from the recommendations, the carrier frequency and the sampling rate are given as parameters. The way how these parameters are implemented in COSSAP will be presented next.

Delay

The delay is achieved by inserting a *N* number of zero-valued samples at the beginning of the signal to be delayed. If the input rate of the signal to be delayed is a given factor f_s and the desired delay is given by τ then the number of zero-valued samples to insert will be:

$$N = \mathrm{NINT}\left[f_s * \tau \right] \tag{5.1}$$

where NINT represents the nearest integer value.

The smallest delay that can be achieved equals $1/f_s$.

Rayleigh Distribution and Average Power.

The Rayleigh Distributed Amplitude is provided as defined in GSM Recommendations 05.05, Version 3.13.0.

The input sampling rate is supposed to be $f_s = 4/T_s$ where T_s is the symbol duration. The Rayleigh distributed amplitude is modelled as a multiplicative distortion $f_m(t)$, which has an adjustable average amplitude a_m, i.e.

$$f_m(t) = a_m f_d(t) \tag{5.2}$$

where $f_d(t)$ is a filter function with the fading spectrum $F_d(f)$ as it is presented in Chapter 2, given by:

$$F_d(f) = \frac{1}{\sqrt{1-\left(\frac{f}{f_d}\right)^2}} \tag{5.3}$$

The filter function $f_d(t)$ is implemented for a fixed velocity by an IIR filter. The 8th order IIR-Filter modelling the Doppler spectra is originally described in [16], and the filter coefficients used in the simulation are listed in [17]. Each path is independently faded.

The filter has the normalized fading frequency f_N

$$f_N = \frac{2\pi f_d}{f_s} = 0.05686 \tag{5.4}$$

where f_N is defined as[1]

$$f_N = \frac{2\pi}{f_{st}} \tag{5.5}$$

where f_{st} represents the IRs/λ rate. Thus the COSSAP GSM model generates IRs corresponding to a sampling rate of :

$$f_{st} = \frac{2\pi}{f_N} = \frac{2\pi}{0.05686} = \frac{f_s}{f_d} = 110.5 \text{ IRs}/\lambda$$

This corresponds to the Doppler spectrum of a mobile with velocity such that:

[1] Note that the normalised fading frequency is fixed in the model, thus f_{st} is fixed as well. The value of f_d will depend on the sampling rate parameter f_s.

$$f_N = \frac{2\pi v_f f_c}{c f_s} = 0.05686 \qquad (5.6)$$

(For GSM the values of these parameters would be f_c= 900 MHz, f_s=4*270833 Hz, yielding to a v_f value of 4*2941 Km/h.)

The filter input is white Gaussian noise with variance adjusted so that the average output power after filtering equals one. The appropriate energy of the path is obtained by multiplying the filtered noise with a corresponding scalar *k*:

$$k = 10^{\frac{\mathrm{P}}{20}}$$

where *P*, in dB, represents the required output power of the corresponding tap.

The output of a single source of white Gaussian noise is multiplexed onto the different taps in order to avoid unwanted cross correlations.

The different velocities of the mobile are produced by interpolation over the fixed number of IRs generated by the model (given by Eq. (5.5)). The larger the interpolation rate, the slower the velocity of the mobile will be. The interpolation rate is defined by:

$$R_{\mathrm{int}} = \frac{v_f}{v} \qquad (5.7)$$

where v_f is defined by Eq. (5.5) and *v* is the speed of the mobile.

This way it can be clearly seen that the necessary number of IRs for simulating the radio channel when the mobile is moving at a certain speed are generated by interpolation between the original IRs created by the model, with the interpolation rate given by Eq. (5.7). The interpolator rate is achieved by two interpolators, first a Sinc interpolator and then a linear, as it can be observed from the block diagrams given in Appendix B1.

5.1.1.2. COSSAP CDMA.

The radio channel model is implemented according to the propagation model defined in Chapter 2. It is adapted to a carrier frequency of 881 MHz. The delay, Rayleigh distribution and average power are implemented in a similar way as the one explained above for GSM, with the only difference of a fractional interpolation of IRs instead of a Sinc and Linear interpolation, as it is done in GSM (v. Appendices B1 and B2). The interpolation rate is the same as in Eq. (5.7) with the difference that in the fractional interpolator the interpolation rate does not have to be an integer value. It also allows interpolation rates smaller than 1, which allows *v* to be larger than v_f. The maximum for *v* is thus determined to where the sampling rate reduction produces aliasing of the fading spectra. Additionally, the interpolation algorithm may be specified, the options are *Linear*, *Quadratic* or *Cubic*.

5.2. Measurement-Based Radio Channel Model Simulations.

The most common way of simulating the radio channel is through a statistical model. This model can be derived from measured data, but this measured data is not usually used in the simulator, it is just used to obtain the statistical model that will be implemented, as it was presented in the previous section.

In this work a different approach is studied, using the measured data in the simulation. The main reasons leading to this are:

- Realistic model of the radio channel, thus more accurate behaviour.
- Comparison of the speed of the simulations with the statistical models.

This approach has already been used in [28] to test under realistic conditions a sequential algorithm for joint parameter estimation and multiuser data detection in DS/CDMA system with multipath propagation. The total number of IRs used was 500 and for each simulation run the channel IR of a single user was chosen randomly from this set of IRs.

5.2.1. WIRCS.

With the purpose presented in Chapter 2 in mind the IRC-Wideband Impulse Response Channel Simulator (WIRCS) was developed in this work. The simulator used is COSSAP (see Section 5.1.1). The measurement system is explained in Section 3.2. and the necessary processing that has to be done for using measured data in a simulation of the radio channel is explained in Chapter 4.

All the pre-processing of the measured data before being used in the simulation is done in MATLAB, while the simulation is done in COSSAP, thus a program for converting MATLAB files into COSSAP datafiles has been created. The main COSSAP blocks of the simulation consist on:

- A block that reads the IR from the created input dataset. The parameters of this block are the number of taps of each IR and the location of the dataset. The real and imaginary parts of the IRs are outputted separately and inputted to the radio channel block.

- A block representing the radio channel, consisting of a FIR filter with complex variable coefficients and complex data input. The parameters of this block are the number of coefficients of the filter and the number of input signals to be used with the same filter coefficients before updating them.

An accurate knowledge of the IRs is necessary to set these parameters, yielding otherwise to wrong matching of the measured data and the channel model.

5.3. Measured Data-Simulation Link.

The main steps that have to be followed for using measured data in a simulator are depicted in Fig. 5.1. This is applicable for a general simulator, and the necessity of each step was presented throughout Chapter 4. In this work the concrete signal processing done to the measured data in order to adapt it to the different simulations performed was:

- Matched filter impulse response estimator
- No implementation of superresolution techniques
- Time domain interpolation of IRs using a cubic spline method. The interpolation rate was established to achieve a rate of f_{st}= 100 IRs/λ from the measured data rate f_{st}^{m}= 5 IRs/λ
- Decimation rate for achieving the different simulations sampling rates from the original measured data at f_s=120 MHz. The anti-aliasing filter was a 100th order FIR filter, designed using the widowing method, and choosing a Hamming window.
- The taps were defined by a noise threshold technique, using a threshold of 18 dB.

All the steps above were done in MATLAB, but any other similar software tool could be used following the same steps. The conversion from MATLAB to COSSAP format was done using the converter program described in Appendix A.

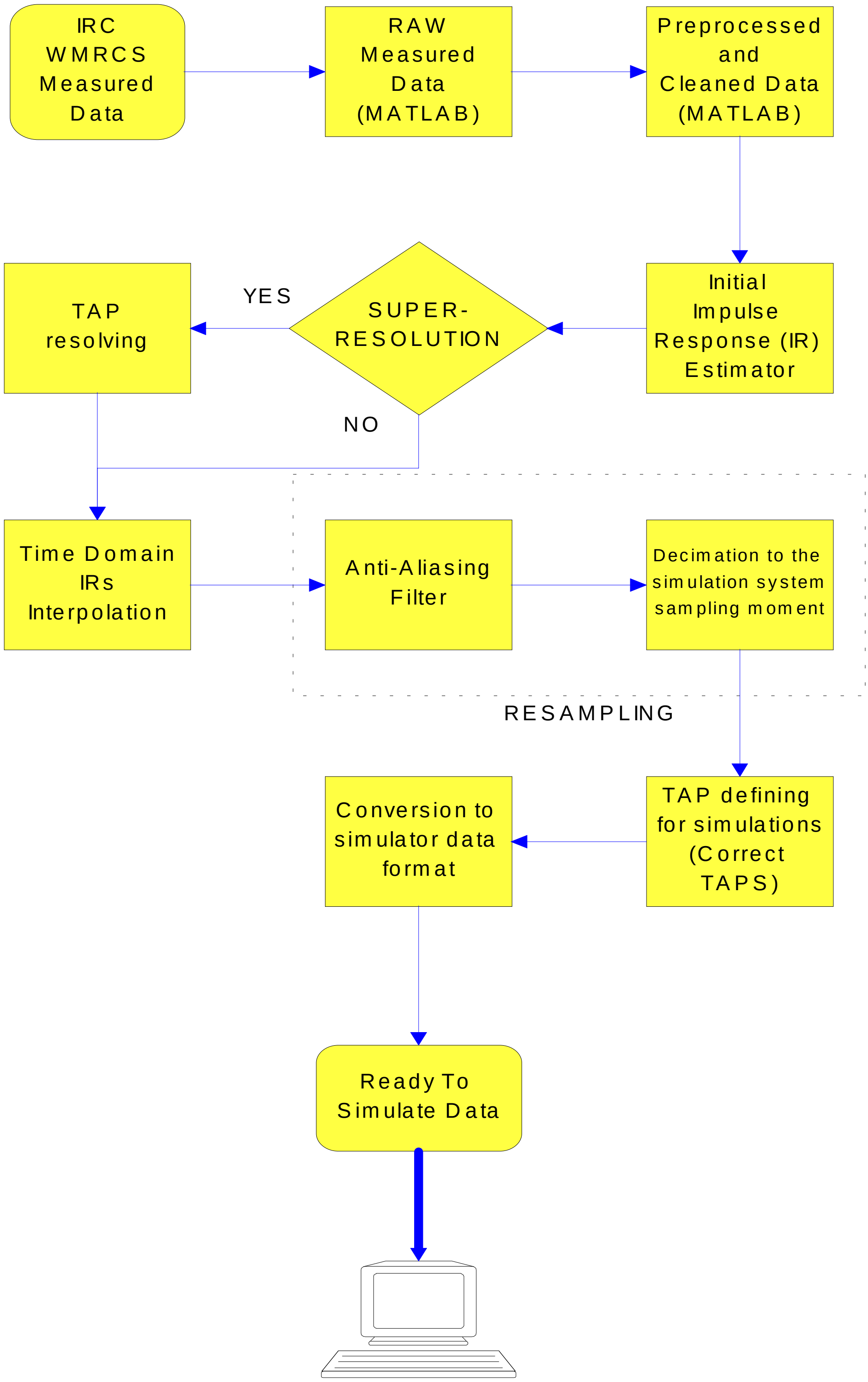

IRC WMRCS Measured Data
RAW Measured Data (MATLAB)
Preprocessed and Cleaned Data (MATLAB)
Initial Impulse Response (IR) Estimator
SUPER-RESOLUTION
YES
TAP resolving
NO
Time Domain IRs Interpolation
Anti-Aliasing Filter
Decimation to the simulation system sampling moment
RESAMPLING
TAP defining for simulations (Correct TAPS)
Conversion to simulator data format
Ready To Simulate Data

Figure 5.1. *Measured Data-Simulator Link*

6. TESTING

In this chapter the description of the different simulations done and the results will be presented. The purpose of the simulations were:

- Testing the feasibility of the WIRCS model in a practical simulation. This was done to check that the model really works and can be applied in a real system.
- Comparing the speed of the WIRCS model to the statistical radio channel models implemented in COSSAP.

In this work all the necessary steps for implementing the WIRCS model from a set of measured data have been presented. Thus the first testing of this work consisted on checking that the WIRCS radio channel model can really be implemented and will work. For this purpose several simulations were done, as it will be presented. The next aim of this work was to compare the statistical models that can be implemented in COSSAP (as they are used in the GSM and CDMA radio channel models) with the WIRCS model, once the previous test showed that it worked.

There were several possibilities for setting the simulations parameters, all of them depending on the characteristics of the simulated input data, and thus on the system to be simulated. These parameters were set to give an estimation on the duration of a simulation in a system where these models may be used. In this case the goal was that the parameters suited the IS-95 system, as it will be presented later.

The way in which the measured data is adapted to each simulation will be explained as well throughout this chapter. The measured data chosen consisted on a set of IRs measured in an urban microcellular environment over a track of 120 meters, with both NLOS and LOS sections. For comparing the speed of the different models in some simulations the required simulated distance was greater than 120 meters. In these cases this set of IRs was used again and again, because using the same set of IRs has no influence in the speed, and taking and adapting the measured data in this required distance has an extra effort with no sense for this testing purpose.

6.1 System input data.

As explained above the simulations were done to give an estimation about how long it would take to simulate the radio channel in an overall simulation chain. For this reason it had no sense to establish random parameters that just would tell the differences of the two different models (WIRCS and statistical) and nothing about the duration in a practical system (such as GSM or IS-95), thus the parameters were set accordingly to a particular system, in this case to the CDMA IS-95 system, with the following parameters.

- Sampling frequency $f_{s\tau}$ = 1.2288 MHz
- Carrier frequency f_c = 2.154 GHz (this frequency is given by the measured data, and is not a variable parameter in the WIRCS model, but it is in the statistical model. For this reason it was set to the same value in the two models, besides in WCDMA the carrier frequency will be in the band of 2 GHz).
- Variable number of taps, delay and power of each tap, speed of the mobile and number of samples to simulate.

The most important variable parameter (from the ones listed above) regarding the duration of the simulation is clearly the number of samples to simulate. This number is dependent on the BER value aimed to be defined in a simulation, the number of errors used for calculating this BER (i.e. number of errors to have adequate statistics about the BER), the length of the code used in the system (i.e. number of chips per symbol) and the oversampling rate of the code (i.e. the number of samples per symbol). These parameters vary depending on the system application and, of course, on the programmer decision. As an usual way of establishing this parameters:

- The BER depends on the overall system to simulate, and may vary from 10^{-3} to 10^{-9}(the lowest value is usually required for some data services with heavy coding). [1]
- The number of symbol errors required at minimum BER is dependent on the programmer's choice, but usually varies from 10 to 100 error samples. The higher this number is the better statistics will be obtained.
- The oversampling rate depends on the system to simulate. For example to model and simulate a system with nonlinearities an oversampling rate of 32 or greater is usually needed, but for other purposes usually it is enough to use 4, 8 or maximum 16.
- The length of the code depends as well on the system to be simulated, it represents the number of chips per symbol used.

For example if the simulation system has the following characteristics:
BER = 10^{-3}, thus $\text{BER}_{\text{min}} = 10^{-4}$, 100 errors required for each BER value. 4 samples per symbol and 127 chips per symbol, then the number of samples required to obtain this BER point would be:

$$4 \times 127 \times 100 \frac{1}{10^{-4}} = 508 \times 10^{6} \text{ samples}$$

Thus for obtaining a single point in the BER curve in this case 508 million samples would be needed .

[1] For this value it is important to note that when simulating for achieving a certain BER value, e.g. 10^{-3}, a value ten times smaller must be considered for the minimum BER, thus the minimum BER would be from 10^{-4} to 10^{-10} in this case.

6.2. Feasibility testing.

As explained in the beginning of this section a set of initial simulations were done in order to check that the WIRCS model really works. The blocks to test consisted of:

- Converter from MATLAB to COSSAP format, creation of COSSAP datasets
- COSSAP Block reading the impulse responses
- FIR channel model with complex input data and complex coefficients (this block was adapted from the COSSAP existing model for complex input data and real coefficients)
- Working of these blocks together, as it is shown in Appendix B (Initial simulations schematic)

For testing the converter function a simple test was done. It consisted on converting a set of IRs to COSSAP ASCII format and comparing visually both. This way it was checked that the function worked in the desired way (as it is explained in Appendix A). Then the IR reader was checked by just viewing in COSSAP the data read from an input dataset. Finally the way of testing that the blocks worked together was to input a train of delta functions to the radio channel and check that the output was the same as the IRs converted from MATLAB to the COSSAP input dataset. The period of the train of delta functions was set to avoid overlapping between consecutive IRs, so that each delta function would produce the output of a single IR without causing overlapping with the next one. Then the data coming from the output of the WIRCS radio channel model was browsed and compared to the original MATLAB data. This way it was observed that the overall WIRCS model worked in a correct way.

6.3. Simulation time comparison.

A simulation speed comparison parameter must be defined before comparing the two approaches. From the simulator point of view there are two possible simulation duration parameters:

- Elapsed time
- CPU time

Elapsed time is the total duration of the simulation, and is strongly influenced by the load of the system, and thus by the priority of access to the CPU compared to the other users that might be using the system at the same time. In this situation it is impossible to recreate the same system conditions in the various simulations to be compared, therefore this parameter doesn't seem suitable to be used for the comparison. On the other hand the CPU time is the effective time used by the simulation, and this is the parameter that will be used for comparing the speed of the different simulations, because it is independent of the system load.

This is clearly seen when the same simulation is run several times in different system load conditions, it is observed that the elapsed time is always different, while the CPU time remains more or less constant.

The simulations were done in a HP 9000/780 machine, installed in the HP UNIX workstation HPUX B.10.20, equipped with the PA-RISC[2] processor model 160 MHz clock frequency PA-7300LC, equipped with an external cache of 1 MB and having a performance of SPECint95 = 7.75 and SPECfp95 = 7.56.[3]

Once the criterion for measuring the speed of the simulation has been established, the two radio channel model simulation's results (WIRCS and statistical COSSAP model) will be presented along with all the parameters, preceded by the description of the different simulations. It has been taken into account that both radio channel models have the same characteristics (considering the common parameters such as the value of the carrier frequency, the value of the tap delays and the simulation sampling rate).

6.3.1. Simulations description.

The purpose of these simulations was to determine the simulation time of both models under certain parameters and with the results presented make it possible to extrapolate the simulation time under other values of these parameters. With this motivation several set of simulations were done, varying the different parameters:

- Number of taps
- Speed of the mobile
- Sampling rate
- Number of samples to simulate

The first set of simulations was done in order to find out the effect of varying the different parameters when the number of samples to simulate remains constant. The next simulations were done in order to see the effect of using an AWGN generator, because in the practical use of the radio channel there will always be a noise generator in the simulations. Finally all the parameters but the number of samples to simulate and the number of taps were kept constant.

This last set of simulations was done under the most likely conditions in which the radio channel will be used, i.e. under the CDMA parameters presented earlier in this section. The sampling rate and the speed of the mobile were kept constant and the number of taps and the number of samples to simulate were varied. With these simulations it was observed how long a simulation of the radio channel might take, for various number of taps in the model (6, 12 and 16), when the speed of the mobile and the sampling rate are kept constant (respectively 3 m/s and 1.2288 MHz) and varying the number of samples (from 10 to 80 millions).

[2] Precision Architecture-Reduced Instruction Set Computing

[3] SPEC= Standard Performance Evaluation Corporation

The number of taps was chosen to match the number of taps given in some of the practical channel models presented in Section 2.3, while the number of samples was chosen to have a significant amount of data without making the simulation time very long.

The motivation for doing all these simulations was not only comparing both WIRCS and COSSAP statistical models, but giving an idea about the time required to simulate the radio channel with these two models under a given set of parameters. This can be clearly done by extrapolating from the results obtained here, as it will be presented in the last part of this chapter.

Next the simulation results for both models, with some explanations about them, will be presented.

6.3.2. COSSAP Statistical Models.

Three different COSSAP statistical models where implemented in order to reproduce the different number of taps: 6, 12 and 16 taps radio channel models where implemented, where each tap if modelled as a classical Doppler spectrum, in the same way as it was showed for GSM and CDMA. A scheme of the 12 taps model is presented in Appendix B4. The parameters of these models are:

- Carrier frequency
- Delay of each tap
- Power of each tap
- Input and Output sampling rate
- Speed of the mobile
- Random seed and variance of the noise generator
- Method for interpolating the IRs.

As explained before these parameters were set so that they reproduce the common parameters of IRs generated from the measured data and the generated statistically (regarding carrier frequency, delay of each tap and the input and output rate, as it will be shown in the numerical example of the parameters given later). The simulation model consisted on inputting a train of Dirac samples and calculating the absolute value at the output of the radio channel. As it can be observed in the schematic given in the appendix B4, at this stage no noise generator was included. Different simulations were done varying

- Number of taps of the radio channel
- Speed of the mobile
- Number of samples simulated (i.e. distance simulated)
- Simulation sampling rate[4]

[4] The sampling rate was set to match the sampling rate of the measured data model. This number is different in the measured data model when the number of taps is changed.

The tables presented next give the results of the simulations, first Table 6.1 shows the results of different simulations varying the speed and the number of taps. Table 6.3 shows the results of various simulations when varying the number of samples simulated and keeping constant the rest of the parameters.

Table 6.1. *COSSAP statistical radio channel model simulation. Results for f_c = 2.154 GHz and 20 million samples*[5] *with different speed and number of taps. Model without noise generator.*

TAPS	**SPEED [m/s]**	$f_{s\tau}$ **[MHz]**	**CPU time [mm.ss]**	**Distance [m]**
6	3	1.2288	03.13	48.8
6	7.5	1.2288	03.40	122
6	15	1.2288	04.26	244
12	3	2.8	06.18	21
12	7.5	2.8	06.41	53.6
12	15	2.8	07.28	107
16	3	4	08.34	15
16	7.5	4	09.02	37.5
16	15	4	10.05	75

From Table 6.1 it can be clearly seen that the simulation time increases with the number of taps, as it was expected because of the higher complexity of the model. Another expected effect is that as the speed increases, the distance increases and the number of interpolated IRs decreases. It can be seen in Table 6.1 that in this case the CPU time increases. The main reason for this behaviour is found from the simulation reports (see Appendix C): if the speed is increased the interpolation rate is smaller, thus each activation of the interpolator model generates less IRs. For this reason, if the same number of generated IRs is required for all the simulations then the higher the speed is the more times the interpolator block has to be activated, resulting in a greater CPU time. An example is shown in Table 6.2, representing the report of the interpolator block for different simulated speeds in the 16 tap model.

Table 6.2. *Interpolator block reports. 20 million samples simulated.*

SPEED [m/s] →	**3**	**7.5**	**15**
Interpolation factor	1680	672	336
Activations	11912	29764	59517

In the results presented above noise was not included. Usually the generation of random numbers, such as happens when implementing noise generators, take an important part of the simulation time. Because of this additional simulations were done in order to describe this effect. The noise consisted on an AWGN generator.

[5] The number of samples is counted at the output of the radio channel block

Table 6.3. *16 tap statistical model, noise generator included. 20 million samples simulated. $f_{s\tau}$ = 1.2288 MHz. f_c = 2.154 GHz. Duration of the simulation*

SPEED [m/s]	DISTANCE [m]	CPU time [mm.ss]
3	15	9.56
7.5	37.5	10.25
15	75	11.13

Next the result of the simulations done in order to see the effect of incrementing the number of samples simulated in different models will be presented in Table 6.4.

Table 6.4. *Duration of the simulation [mm.ss] depending on the number of samples $f_{s\tau}$ = 1.2288 MHz. v = 3 m/s. f_c = 2.154 GHz. COSSAP statistical model including noise generator.*

Samples[millions] Taps	10	20	40	80
6	02.17	04.35	09.16	18.10
12	03.58	08.01	15.48	31.33
16	05.28	10.42	21.17	42.33

From Table 6.4 it can be seen that if the number of samples is doubled, then the CPU time is doubled also, as it was expected considering that doubling the number of samples means doubling the number of operations to be done in the CPU, and thus doubling the CPU time. This effect is observed to be the same independently from the number of taps. As it will be shown later, the effect is the same when using experimental data.

Comparing Tables 6.1, 6.2 and 6.4 it can be clearly seen that the increment of time when including a noise generator and simulating the same number of samples (20 millions in this example) is constant regardless the number of taps of the radio channel model and the speed of the mobile, and about 1 minute and 20 seconds for the simulations done. The same effect was observed in the WIRCS model, as it will be presented later.

6.3.3. WIRCS model.

The results of the different simulations using the WIRCS model will be presented in this section, preceded by an explanation about how the measured data were measured and adapted to the simulations.

The data used in these simulations was measured by the IRC radio channel sounder in an outdoor urban microcellular environment along a track of 120 meters with NLOS and LOS. The sounder parameters were:

- f_c = 2154 MHz
- PN code length: 127
- f_{chip} = 30 MHz
- Configuration: Direct sampling
- f_{st} = 5 IRs/λ
- $f_{s\tau}$ = 120 MHz

The IRs measured in this way consisted of 508 samples each one. The adaptation to the different simulation was done in the following order (explained in more detail in Chapter 4).

- AGC applied to the measured IRs (future adaptations may not include it)
- Decimation to the simulation sampling rate (this parameter is dependent on the number of taps to be used in the simulation, e.g. for the 6 taps model a decimation factor of 85 was applied, obtaining from the original 508 samples the number of taps required: 6, consequently the sampling frequency of this generated data was 120/85 ≅ 1.4 MHz). This was done for just comparing the two models in a similar way, but in the radio channel simulations using WIRCS the number of taps will be determined by the simulation system sampling rate and the noisy tail taps that can be removed.
- Interpolation of IRs from 5 to 100 IRs/λ
- Noise thresholding of 15 dB

In this procedure the decimation was done before the interpolation due to the limitation of the software used. The size of the data created when interpolation was done first occupied more memory than the software allowed, thus the data was first decimated (reducing this way its amount) and then interpolated.

Simulations were performed under the same conditions as the statistical channel model, yielding to the results presented in the following tables.

Table 6.5. *WIRCS radio channel model simulation. Results for f_c = 2.154 GHz and 20 million samples with different speed and number of taps. Model without noise.*

TAPS	SPEED [m/s]	$f_{s\tau}$ [MHz]	UPDATE RATE	NUMBER OF IRs	CPU time [mm.ss]	D [m]
6	3	1.2288	565	35398	00.22	48.8
6	7.5	1.2288	226	88425	00.22	122
6	15	1.2288	113	176991	00.23	244
12	3	2.8	1287	15540	00.28	21
12	7.5	2.8	515	38834	00.30	53.6
12	15	2.8	257	77821	00.31	107
16	3	4	1839	10875	00.33	15
16	7.5	4	735	27210	00.34	37.5
16	15	4	367	54495	00.34	75

Some explanatory comments about this table are:

- As it was explained above, the measured data consisted of a track of 120 meters, for simulations needing a greater distance the same set of IRs was used again. This was just done for testing the speed and doesn't have any effect in the results presented in this section.
- Update rate means how many simulated samples are used for the same channel IR, and it is calculated as in Eq. (4.23).
- The number of IRs used is calculated from the number of samples simulated and the updating rate.

The results observed in Table 6.5 are mainly:

- The simulation time is much smaller than in the statistical model, but the main effort is done in the off-line processing.
- For a given model with a fixed number of taps increasing the speed has no effect in the simulation time. This can be explained because when the speed is increased the only parameter changed is the updating rate, and thus more IRs have to be loaded, but the number of operations to be done in the CPU remains the same. Because of this result it can be seen that the load of new IRs takes unappreciable time from the simulation.
- When the number of taps is incremented, the effect on the duration is not as large as in the statistical channel model.

The results of the simulations done considering the noise are presented in the next table.

Table 6.6. *Duration of the simulation: 16 taps WIRCS model, noise generator included. 20 million samples simulated.* $f_{s\tau}$*=1.2288 MHz,* f_c*=2.154 GHz.*

SPEED [m/s]	**DISTANCE [m]**	**UR [samples/IR]**	**Num. IRs**	**CPU time [mm.ss]**
3	15	1839	10875	01.46
7.5	37.5	735	27210	01.47
15	75	367	54495	01.49

It can be observed that the effect of including a noise generator is the same as in the statistical model, an increment of about 1 minute and 20 seconds over the duration in the model without noise. So it can be inferred that the influence of the noise generation in the simulations will be the same for both models.

Other simulations were performed in order to test the effect of the simulated distance (or equivalently, amount of simulated data) in the duration of the simulation. The results are presented in Table 6.7. The simulations were done for a simulated speed of 3 m/s and models with different number of taps.

Table 6.7. *Duration[mm.ss] of the simulations with different numbers of samples* $f_{s\tau}$*=1.2288 MHz. v=3 m/s.* f_c*=2.154 GHz. WIRCS model including noise generator.*

Samples[millions] **Taps**	**10**	**20**	**40**	**80**
6	00.48	01.36	03.10	06.26
12	00.51	01.43	03.28	06.56
16	00.54	01.48	03.36	07.12

As it was expected and in the same way as in the statistical channel model, the time is doubled when the simulated amount of data is doubled.

6.3.4. Comparison results

After presenting the results of the different simulations done for the different models, an example about the matching of the parameters of both models will be presented, followed by the graphical representation of some of the results.

The parameters of the statistical model were set so that they matched the same parameters obtained in the measured data. Considering the simulations done with the 6 taps model, $f_{s\tau}$ *= 1.2288 MHz. v = 3 m/s* and f_c *= 2.154 GHz,* the adapted data consisted on 6 taps where the delay spacing corresponded to *1/*$f_{s\tau}$, the same delay spacing of the different taps was set in the statistical model. In both models a cubic interpolation of IRs is implemented.

In figures 6.1-6.3 results comparing the duration of the simulations will be presented, where the number of samples was kept constant and the speed and number of taps varied. In these graphics the effect of the noise generator is not considered because it has the same effect on both models.

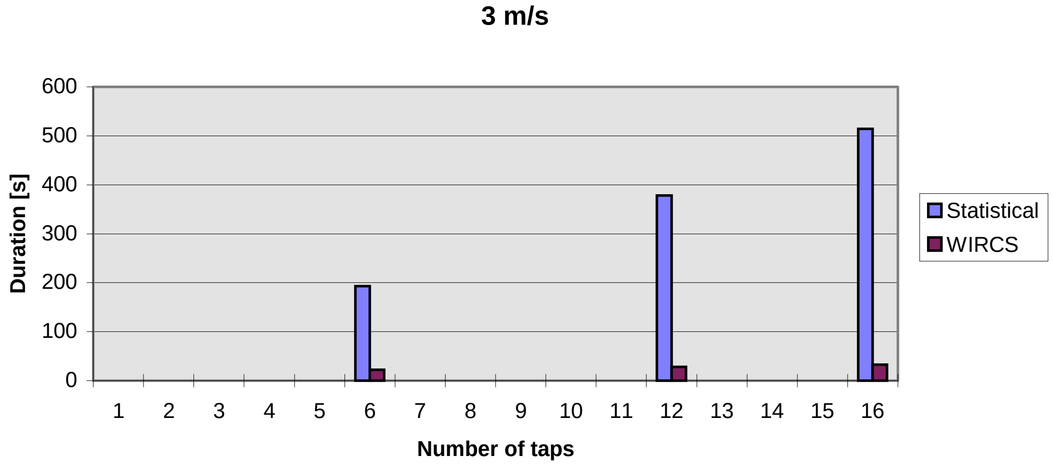


Figure 6.1. *WIRCS vs. STATISTICAL model. Simulation time of 20 million samples at 3 m/s. Comparison with 6, 12 and 16 taps.*

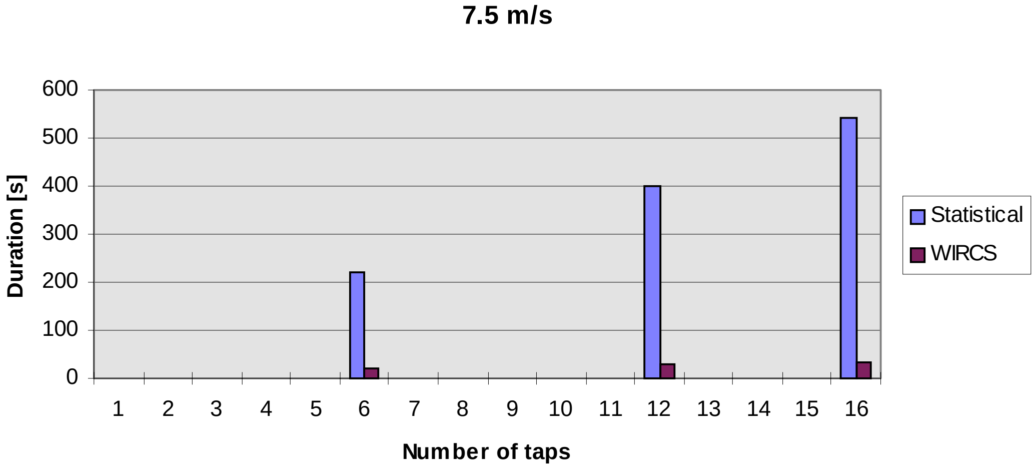


Figure 6.2. *WIRCS vs. STATISTICAL model. Simulation time of 20 million samples at 7.5 m/s. Comparison with 6, 12 and 16 taps.*

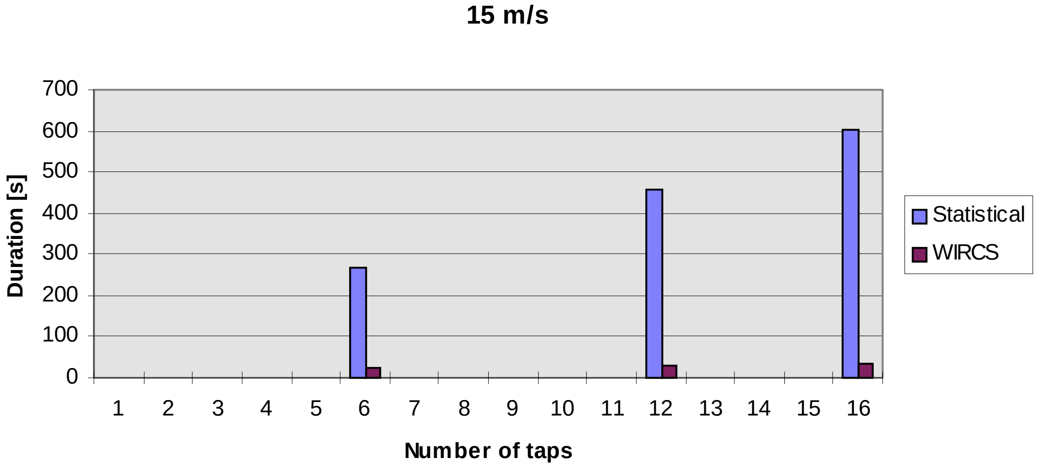


Figure 6.3. *WIRCS vs. STATISTICAL model. Simulation time of 20 million samples at 15 m/s. Comparison with 6, 12 and 16 taps.*

7.- CONCLUSIONS

In this section the conclusions about the study of using measured data in a radio channel simulation model and the results of the different simulations performed will be presented, along with the further field of investigation in the subject presented in this work.

The conclusions extracted from the study of the different signal processing that has to be done to the measured data for adapting it to the simulations are mainly:

- This approach offers a good flexibility for modelling different radio channels as adaptation is done by just changing parameters in the off-line signal processing. This way the same measured data can be used with several parameters (e.g. different sampling rates, number of taps of the model, speed of the mobile, etc.).
- It is not necessary to measure a great amount of IRs. The necessary number of IRs can be generated by interpolation. A cubic spline interpolation method was tested, obtaining good results, for this purpose it seems to be good enough.
- More precise radio channel models can be achieved with the implementation of superresolution methods in the off-line processing of the measured data.

The main conclusions extracted from the examination of the results are:

- The duration of the simulation time using the WIRCS model is significantly reduced.
- The effect of augmenting the number of taps in the model is almost insignificant in the WIRCS model compared with the classical statistical models.
- Simulating higher speed in the WIRCS model makes the increase of the simulation time much smaller than in the statistical model.
- The effect of increasing the number of samples to simulate is the same in both models, the simulation time is increased at the same rate as the number of samples, thus if the number of samples is doubled, the simulation time is doubled as well. This effect is much more critical in the statistical model, because the simulation times are greater.
- The effect of including random number generators (such as AWGN blocks) is an equal absolute increment of time in both models.

Before making any complex simulation one should ask about how long it would take. In some cases this duration would make the simulation unrealisable. The GSM 05.05 recommendation can be considered as an example. The models are usually defined by 12 taps settings, but according to the simulators it might not be possible to simulate the complete model, for this reason a reduced configuration of 6 taps is given.

As the different simulations to be done become more complex the model has less chances to be implemented due to the duration it might take, an example of this would be the simulation of multielement antenna configurations, where each element represents a different channel. If these channels would be implemented as a statistical model, they might not be realisable because of the very large duration of the simulation, depending, of course, on the parameters of the simulation. On the other hand using a model which would take less time, such as WIRCS, would make it possible.

Another advantage that the WIRCS model has is the recreation of a more realistic behaviour of the radio channel, based on the use of experimental data. Traditionally this data is used for extracting the statistics on which the different radio channels will be based. When the WIRCS model is used there is no need for this studies about the measured data. The data can be used in a straightforward manner, after some processing. This model is as well more flexible than the existing ones, allowing an easy adaptation for the different types of environments and systems to be studied.

One of the major drawbacks of the WIRCS model is the signal processing that has to be done for the adaptation of the measured data to the simulations. This processing is the main subject to investigate in further stages about this subject, as it has been presented during this work.

The main points for further study are:

- Effect of the number of impulse responses per wavelength used in the simulations. This effect can be observed in the BER curves obtained from a communications system, where the overall system is configured for working in a wideband multipath environment. This parameter is important in order to know basically two things: the minimum number of IRs per wavelength measured that is allowed for recreating the radio channel and the number of IRs that must be interpolated from the original number measured.
- Implementation of noise and interference in the overall simulation system
- Implementation of superresolution methods yielding to more accurate radio channel models.
- Study the different methods for interpolating IRs.
- Study the different ways in which the resampling of each IR can be done.

This work just establishes the basic link between measured data and the simulator, checking its feasibility and comparing the results with the traditional methods for simulating the radio channel. It also mentions the further work to be done in order to improve its performance.

REFERENCES

[1] Rappaport, Theodore S. "*Wireless Communications, principles and practice*", Prentice Hall, New Jersey, 1996. 641 p.

[2] Clint Smith, P.E., "*Practical Cellular and PCS Design*", McGraw-Hill Telecommunications, New York, 1998.

[3] Feuerstein, M.J., Blackard, K.L, Rappaport, T.S., Seidel, S.Y., and Xia, H.H. "Path Loss, Delay Spread, and Outage Models as Functions of Antenna Height for Microcellular System Design", *IEEE Transactions on Vehicular Technology*, Vol.43, No.3, pp. 487-498, August 1994.

[4] Jeruchim, Michel C., Balaban, Philip and Shanmugan, K.Sam, "*Simulation of Communication Systems*", Plenum Press, New York, 1992. 731 p.

[5] Proakis, John G., "D*igital Communications*", McGraw-Hill, Second Edition, New York, 1989, 905 p.

[6] Prosch, Markus, "*Generation of Pseudo-Random Code Sequences for a Wideband Mobile Channel Measurement System*", Master's Thesis, 1994, Helsinki University of Technology, Department of Electrical and Communications Engineering, Communications Laboratory.

[7] Parsons, J.D., Demery, D.A., Turkmani, A.M.D., "Sounding techniques for wideband mobile radio channels: a review" *IEE Proceedings-I,* Vol.138, No.5, pp.437-446, October 1991.

[8] Dixon, R.C., "*Spread Spectrum Systems*", John Wiley and sons, New York, 1984, 422 p.

[9] Kimmo Kalliola and Pertti Vainikainen, "Characterization System for Radio Channel of Adpative Array Antennas" *Proceedings of the 8th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC),* September 1997, Helsinki, Finland. Vol.1, pp. 95-99.

[10] Parsons, J.D., "*The Mobile Radio Propagation Channel*", Pentech Press, London 1992, 316 p.

[11] T. Korhonen, M. Hall, S.-G. Häggman. "Resolution enhancement techniques in mobile channel measurements" *Proceedings of Nordic Radio Symposium (NRS) '95,* Saltsjöbaden, Sweden, April 1995. pp. 127-132.

[12] Richard L. Burden, J. Douglas Faires, "*Numerical Analysis*" Fifth edition, PWS Publishing company, Boston, 1993. 768 p.

[13] Victor Fung, Theodore S. Rappaport, "Bit Error Simulation for π/4 DQPSK Mobile Radio Communications using Two-Ray and Measurement-Based Impulse Response Models" *IEEE Journal on Selected Areas in Communications,* Vol. 11, No.3, April 1993. pp. 393-405.

[14] Suvi Ruohomaa "*Investigation and comparison of FRAMES air interface proposals for UTMS*". Master's Thesis. Department Of Electrical and Communications Engineering. Helsinki University of Technology. September 1997. 66 p.

[15] T. W. Parks, C. S. "*Digital Filter Design*" . John Wiley & Sons. New York 1987. 341 p.

[16] H. Brehm, H. Stammler, W. Werner, "Design of a high flexible digital simulator for narrowband fading channels", *Signal Processing III: Theories and Applications, I.T. Young et. al. (ed.), Elsevier Science Publishers;* pp. 1113-1116, 1986.

[17] R. Haeb, "*Kohaerenter Empfang bei Datenuebertragung ueber nichtfrequentselective Schwundkanaele*" Ph. D. Thesis, Aachen University of Technology, 1988.

[18] Ed: J. Jiménez, "Final Propagation Model" *CoDiT Deliverable: R2020/TDE/PS/P/040/b1.* June 1994.

[19] Ed: M. Pizarroso, J. Jiménez, "Common basis for Evaluation of ATDMA and CoDiT Systems Concepts" *Deliverable: MPLA/TDE/SIG5/DS/P/001/b1.* September 1995. 40 p.

[20] Zoran Kostic, M. Ibrahim Sezan, Edward L. Titlebaum "Estimation of the Parameters of a Multipath Channel Using Set-Theoretic Deconvolution" *IEEE Transactions on Comunnications,* Vol. 40, No. 6, June 1992, pp. 1006-1011.

[21] T.S. Rappaport, S.Y. Seidel, K. Takamizawa, "Statistical Channel Impulse Response Models for Factory and Open Plan Building Radio Communication System Design", *IEEE Transactions on Communications*, Vol. 39, No.5, May 1991. pp. 794-807.

[22] H. Krim, M.Viberg, "Two Decades of Array Signal Processing Research: The Parametric Approach" *IEEE Signal Processing Magazine,* July 1996, pp. 67-94.

[23] T. Korhonen, M. Hall, S.-G. Häggman, "Superresolution Multipath Channel Parameter Estimation by Matched Filter Deconvolution and Sequential Bit Tuning", *2nd International Workshop on Multi-Dimensional Mobile Communications (MDMC'96),* Seoul, Korea, 1996, pp. 66-70.

[24] Burg J. P., "Maximum Entropy Spectral Analysis", *Proc. of the 37th meeting of the Society of Exploration Geophysicists*, 1967.

[25] Ralph O. Schmidt, "Multiple Emitter Location and Signal Parameter Estimation" *IEEE Transactions on Antennas and Propagation,* Vol. AP-34, pp. 276-280, March 1986.

[26] Alan V. Oppenheim, Alan S. Willsky, Ian T. Young, "*Signals and Systems*" Prentice-Hall, New Yersey 1983, 795 p.

[27] ITU Radiocommunication study groups "*Guidelines for evaluation of radio transmission technologies for FPLMTS (Draft New Recommendation ITU-R.M. [FPLMTS.REVAL])*", 26th April 1996, 80 p.

[28] Dirk Dahlhaus, Bernard H. Fleury and Andrej Radovic, " A Sequential Algorithm for Joint Parameter Estimation and Multiuser Detection in DS/CDMA Systems with Multipath Propagation" *Wireless Personal Communications 6, pp. 161-178. Kluwer Academic Publishers. 1998.*

[29] J. Kivinen, *Nopea laajakaistaisen impulssivasteen mittaus (Rapid broad band radio channel impulse response measurements, in Finnish),* Master's Thesis, Helsinki University of Technology, June 1995.

[30] P. Aikio, *Radiokanavaluotaimen kehittäminen (Development of the radio channel sounder, in Finnish),* Master's Thesis, Helsinki University of Technology, February 1996.

[31] COSSAP simulator online manuals, version 1998.02.

APPENDIX A

MATLAB-COSSAP DATA CONVERTER

In this appendix the program created in C-language for converting data from MATLAB to COSSAP will be described. First the manual with the considerations for compiling and using the program will be described, along with all the solutions to the problems that the author found when creating it. Finally the code of the program will be listed.

This appendix is thought to give the reader who is familiar with the subject a better comprehension about what has been done, and how to reproduce the same steps for creating and executing this program without facing the inherent problems of the initial steps.

APPENDIX A.1. MANUAL

The task of this program is to translate the data generated in MATLAB format into COSSAP Datasets that can be used for the simulations. For this reason firstly an explanation about the MATLAB data ready to use in the simulations will be given, in order to achieve a better understanding about this program. Finally the description of the program and its use will be given.

MATLAB data

The first MATLAB data available consists of the raw measured data and the AGC values. The IRs are obtained after processing the raw measured data, in this processing including the AGC value or not is the user's choice. Finally the IRs are adapted to the simulations via the signal processing explained throughout Chapter 4. At this point a variable containing all the IRs from one antenna element is saved into a MATLAB file, the structure of this variable is a two dimensional matrix where the rows are the different samples of an IR and the columns are the different IRs, containing either real or complex numbers. This file will be the source for the data conversion.

Program call

The call syntax is as follows:

mat2coss <matlab file> <cossap file> <cossap format>

where:

- <*matlab file*> is the complete path where the **.mat* file is stored.
- <*cossap file*> is the complete path where new file will be generated. The extension of this file is usually set so that it gives the format in which the data it contains is stored, where this is the one specified in the syntax (see Table 1). According to this, a file containing, e.g. COSSAP_ASCII data, would be *filename.ca*)
- <*cossap format*> is the allowed COSSAP data type (see Table 1).

Table 1. *COSSAP file formats*

COSSAP FORMAT	SYNTAX
COSSAP_BINARY	cb
COSSAP_ASCII	ca
ASCII_MATRIX	am
ASCII_LF_EOR	alf
ASCII_BL_EOR	abl
RAW_8BIT	r8
RAW_16BIT	r16
RAW_32BIT	r32

The MATLAB file can contain one variable (where the **.mat* file has been created by saving the MATLAB workspace variable with the *"save"* command) of real or complex data. The data from the variable will be converted into the COSSAP format. According to the usage of the data in the simulations, and following the recommendations of the COSSAP manuals, the data will be stored in the dataset as follows, e.g. a set of *N* given complex values:

Real(element n.1)
Imag(element n.1)
Real(element n.2)
Imag(element n.2)
--
--
Real(element n.N)
Imag(element n.N)

In the simulations it should always be considered that the dataset has been generated in this way, so if the original MATLAB data doesn't have real or imaginary part, its correspondent value is saved as zero, to be consistent with the structure of the dataset in the simulations.

The program will give information about what has been done, and will display the type of errors that might happen during its execution. It will give a complete information about the variable contained in the MATLAB file

- Number and name of the variable stored in the MAT-file.
- Number of elements converted from the variable (floating point numbers).
- COSSAP format into which the conversion has been done.
- File generated.

Compiling directives

When compiling the C-code of the converter program a few steps should be followed, in this case when working in the operating system HPUX, with the architecture HP700:

- In order to use the MATLAB API functions (*mat** and *mx** functions), their correspondent libraries have to be compiled and linked to the code. This must be done in two steps:

1.- Telling the system where the MATLAB API shared libraries reside. This is done by appending its path to the environment variable *SHLIB_PATH* (for HP700 architectures). The command is:

```
setenv  SHLIB_PATH  <matlab>/extern/lib/hp700:$SHLIB_PATH   (A1.1)
```

where <*matlab*> is the MATLAB root directory. In order to avoid writing this command every time when the user logs in, it should be placed in a start up script such as *.cshrc*.

2.- Telling the compiler where the headers and the object code for these functions are. This is done by including the header files from:

<*matlab*>*/extern/include* (option -*I* in the compiler).

and by linking the **mat* and **mx* libraries from:

<*matlab*>*/extern/lib/hp700* (options -*L* and -*l* in the compiler).

- And in order to use the COSSAP *cfh* library, link from:

$COSSAP_DIR/lib

the following libraries:

cfh.a *cb.a* *ca.a* *ma.a*
fa1.a *fa2.a* *raw.a* *cmisce.a*
misce.a *fil.a* *unix.a*

where *$COSSAP_DIR* is the path of the COSSAP root directory.

This link is done by compiling these libraries together with the program, writing after the name of the file the path of all these libraries (see example of compiling). The compiler had the options -*L* and -*l* for this linking purpose, but when both MATLAB and COSSAP libraries were linked at the same time the compiler didn't create the executable file, so it was done as explained in the beginning of this paragraph.

In the architecture used, the ANSI C compiler is invoked by the command *"cc"*. This command has several options that have to be used, apart from those that have been explained before. These are:

- option -*Ae* for the extending mode (ANSI C compiler, that allows as well the use of environment variables in the compiler options)

- option *-g* for generating debugging code that can be used later with the system debuggers (such as *xdb* and *dde*) when necessary.

One example of the command to execute would be:

cc -Ae -g *example.c $COSSAP_DIR/lib/cfh.a …$COSSAP_DIR/lib/unix.a*
-I*$MATLAB/extern/include* **-L***$MATLAB/extern/lib/hp700* **-l***mat* **-l***mx*
-o*mat2coss.exe*

Considerations

COSSAP FORMAT

Regarding the different COSSAP formats that can be used, a few considerations should be made before the simulations in order to solve the tradeoff between memory space and data organisation.

RAW data format. This is the most effective in memory management, for it converts each data element to *signed char, signed short* and *signed long* (res. 8,16 and 32 formats) and it doesn't write formatting data or additional information into the file as the other formats do. This format can be used to read and write unstructured binary data, but when using it we should take into account that:

- It is not record oriented, i.e. the data stored in this format is not structured in records, so the data organisation that the other formats have is lost . (For the definition of record see COSSAP *"File Organisation and Formats"* manual [31]).

- When writing a file in RAW format all bits that cannot be represented are lost.

With this format all the functions from *"cfh"* for reading and writing are supported (for more information about these functions refer to the COSSAP *"Stream driven simulation user guide"* manual [31]).

ASCII data formats. These formats are record oriented, but the overhead caused by the additional information is such that sometimes it makes the resulting COSSAP file even larger than the original MATLAB file. For large amount of data these formats are not recommended.

BINARY data format. This format is record oriented, where each record consists of elements of the same type. This seems to give the best trade-off between data organisation and memory space, for the overhead is smaller than the one in ASCII and it is record oriented.

One example of the memory used by these different formats is shown on Table 2 where the original MATLAB file consisted of data stored in a 252x100 matrix of complex values.

Table 2. *Memory space example.*

FORMAT	**SIZE (bytes)**
MATLAB	403.392
COSSAP_BINARY	204.880
COSSAP_ASCII	1.159.234
ASCII_MATRIX	1.159.200
ASCII_LF_EOR	1.108.800
ASCII_BL_EOR	1.159.200
RAW_32BIT	201.600

The most suitable COSSAP format when simulating large amount of data seems to be the *binary* format, the main reasons for this are:

- Good memory management.
- Record oriented.
- Faster simulations.

EXTENSION

The conversion of MATLAB files containing only one variable has been explained so far in this appendix because this is the type of files that will be used in this work, but the MATLAB file can contain several variables saved in it. The converter function takes this into account and is designed to convert several variables, even though this option is not used in this work.

Each variable contained in the MATLAB file will be translated into a different record in the COSSAP file in the same way as explained throughout this appendix for files containing only one variable. Thus the COSSAP file created will consist of as many records as variables had the original MATLAB file. The program will give information about all the variables during the conversion:

- Number of variables contained in the MATLAB file
- Name and number of elements of each variable
- Order of conversion, thus record generation order
- COSSAP format used for the conversion
- Created COSSAP file

COSSAP FILE GENERATED

When using the datasets created it's important to take into account the way that the several variables are stored in the COSSAP file. This is important for the way in which the datasets are going to be read in the simulations, and a decision should be made about whether the reading will be done sequentially or record oriented, depending on the number of records and their size.

Typically one file containing one variable storing a large amount of data, as the ones used in this work, will be converted into one record, and here the sequential way of reading is recommended. The main reason is that if the record oriented reading is used, it might be possible that the whole file cannot be loaded into the buffer. The different ways of reading are determined by the COSSAP macro function used, for further details see *"Stream driven simulator user guide"* in the COSSAP manuals [31].

DATA CONVERSION

Another feature of this program is that it converts the data from double precision, as they are stored in MATLAB, to floating point precision, i.e. from 8 to 4 bytes. This is mainly done because in the simulations double precision is not needed and this way memory save is achieved for enabling use of larger amount of data.

EXECUTION

The last step for running the program is just invoking the command. In order to make it in the proper way it should be checked that the operating system knows where the shared libraries reside, i.e. that command (A1.1) has been executed.

In some UNIX machines the act of invoking the command of the executable code is not recognised, for this reason it has to be invoked in the following way:

```
./mat2coss <matfile> <cossap file> <cossap format>
```

(dot, slash followed by the name of the executable code and the parameters of the program <…>, where ./ indicates to the operating system that the executable code has to be found in the current directory)

APPENDIX A.2. Code

In this section the C code of the converter function will be listed, as it was compiled for the executable function.

```
/*
 * Created by J.Sanz, original idea by J. Mattila.
 *  Last modified 8th-July-98 for multiple variables.
 */

#include <stdio.h>
#include <stdlib.h>  /* exit() */
#include <string.h>
#include <mat.h>     /* from MATLAB */

void main (int argc, char *argv[])
{

      char mat_file[1024];            /* path and name of Matlab file */
      char coss_file[1024];           /* path and name of Cossap file */
      char coss_format[1024];         /* cossap file format */
      MATFile *pmat;                  /* pointer to the MATLAB file */
      mxArray *pmx;                   /* pointer to a mxArray */
      long mn;                        /* number of data elements */
      int size;                       /* size (in bytes) of each element */
      double *pr;                      /* pointer to real data */
      double *pi;                      /* pointer to imag data */
      float *pcoss;                    /* pointer to the writing vector */
      float *pc;                       /* auxiliar */
      double *pmr,*pmi;                /* auxiliar pointers for writing */
      FILE *cossfp;                    /* Cossap file handler pointer */
      long ccg_lp0;                    /* "for" loop index */
      long status;                     /* controlling files in "cfh" functions */
      long file_format;                /* Cossap file format */
      int i;                           /* for the different variables stored */
      char **pdir;
      int numdir;

        /* Check command line parameters */
        if (argc != 4)  {
          fprintf(stderr,"LINE COMMAND IS NOT CORRECT !! \n");
          fprintf(stderr,"Syntaxis when calling the function: \n");
          fprintf(stderr,"\t converter <matlab file> <cossap file> <cossap format> \n");
       exit(1);
       }
```

```
        strcpy (mat_file, argv[1]);          /* source file - Matlab file */
        strcpy (coss_file, argv[2]);         /* destination file - Cossap file */
        strcpy (coss_format, argv[3]);     /* Cossap format */

       /* open Matlab file for reading */
         pmat=matOpen(mat_file,"r");
         if (pmat==NULL) {
           fprintf(stderr,"Problems opening:\t%s\n",mat_file);
           exit(1); /* the 1 indicates abnormal program termination */
       }
        /* check that the file format is correct */
        file_format = 0;
        if ( !strcmp (coss_format, "cb") )  {
          file_format = 1;         /* COSSAP_BINARY */
          }
        if ( !strcmp (coss_format, "ca") )  {
          file_format = 2;         /* COSSAP_ASCII */
          }
        if ( !strcmp (coss_format, "am") )  {
          file_format = 3;         /* ASCII_MATRIX */
          }
        if ( !strcmp (coss_format, "alf") )  {
          file_format = 6;         /* ASCII_LF_EOR */
       }
        if ( !strcmp (coss_format, "abl") )  {
          file_format = 7;         /* ASCII_BL_EOR */
          }
        if ( !strcmp (coss_format, "r8") )  {
          file_format = 8;         /* RAW_8BIT */
          }
        if ( !strcmp (coss_format, "r16") )  {
          file_format = 9;         /* RAW_16BIT */
          }
        if ( !strcmp (coss_format, "r32") )  {
          file_format = 10;       /* RAW_32BIT */
          }
        if (!file_format)  {
          fprintf (stderr, "\nFile format - %s - is not correct!\n\n", coss_format);
          fprintf (stderr, "Correct formats are listed below:\n\n");
          fprintf (stderr, "\t cb \t - \t COSSAP_BINARY\n");
          fprintf (stderr, "\t ca \t - \t COSSAP_ASCII\n");
          fprintf (stderr, "\t am \t - \t ASCII_MATRIX\n");
          fprintf (stderr, "\t alf \t - \t ASCII_LF_EOR\n");
          fprintf (stderr, "\t abl \t - \t ASCII_BL_EOR\n");
          fprintf (stderr, "\t r8 \t - \t RAW_8BIT\n");
          fprintf (stderr, "\t r16 \t - \t RAW_16BIT\n");
          fprintf (stderr, "\t r32 \t - \t RAW_32BIT\n\n");
          exit(1);
}
```

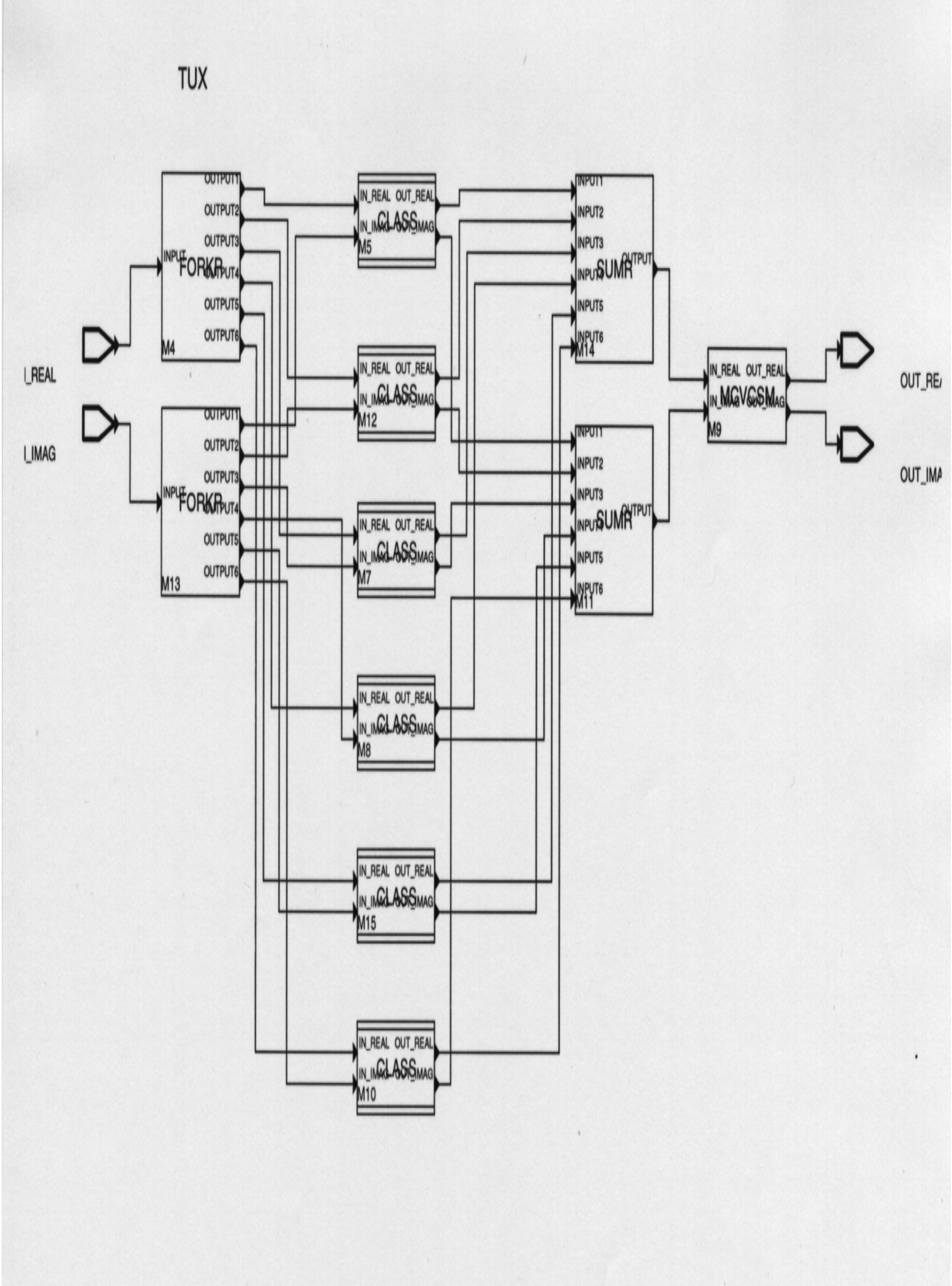
```
            /* open Cossap file in format "file_format" for writing */
            status = cfhopen (file_format, &cossfp, coss_file, "WRITE");
            if (status)  {
              fprintf (stderr, "\n Cannot Open Cossap file:\t%s\n",coss_file);
              fprintf (stderr, " for writing in the cossap %s format\n",coss_format);
              exit(1);
              }

           printf("\nWRITING DATA FROM MATLAB FILE <%s> TO COSSAP FILE
             …    <%s>\n\n",mat_file,coss_file);
           pdir=matGetDir(pmat, &numdir);
           if (pdir == NULL && numdir < 0) {
            fprintf(stderr,"Can't read directory from <%s> \n",mat_file);
           exit(1);
          }

            printf("\n\tDirectories of MAT-file: <%s>\n\n",mat_file);
            for (i = 0; i < numdir; i++) {
                printf("\tNumber of variables stored: %i \n\n",numdir);
                printf("\tVariable Number: %i corresponds to <%s>\n",i+1,pdir[i]);
                pmx=matGetArray(pmat,pdir[i]);

        /* Retrieving MATLAB file information
            pmx=matGetNextArray(pmat);
            if (pmx==NULL){
                printf("Error getting mxArray from file %s",mat_file);
                exit(1);
            } */

            /* writing variables one by one */

            mn = mxGetM(pmx)*mxGetN(pmx); /* Number of elements in the Array */
            size = mxGetElementSize(pmx); /* Size, in bytes, of each element */

            /* check that the size of FLOAT number is four bytes */
            if (sizeof(float) != 4)  {
              fprintf (stderr, "\nThe size of FLOAT variable is not four bytes as assumed
here.\n");
              fprintf (stderr, "\nYou probably need to change FLOAT type arrays to
DOUBLE type arrays!\n");
              exit(1);
              }
```

```
/* Memory allocation */
pcoss = (float *) malloc ((size_t) ((2*mn +1)*sizeof(float)));
if (pcoss==NULL) {
  fprintf(stderr,"\n Memory allocation failure !!! \n");
  }

/* Getting data from the mxArray */
pr = mxGetPr(pmx);
if (pr==NULL && (!mxIsComplex(pmx))) {
  fprintf(stderr,"Error reading real data from: %s\n",coss_file);
  exit(1);
  }

pi = mxGetPi(pmx);
if (pi==NULL && mxIsComplex(pmx)) {
  fprintf(stderr,"Error reading Imag data from: %s\n",coss_file);
  exit(1);
  }

/* writing data vector (pcoss) */
/* auxiliar pointers initialization */
pc=&pcoss[0];
pmr=&pr[0];
pmi=&pi[0];

/* writing data to the vector for different data type */
/* Type cast for converting double type to float */
/* Several possibilities available */

/* First possibility: Real & Imag part */
/* OR Imag part (in this case the real part is set to zero by MATLAB) */
if (pr!=NULL && pi!=NULL) {
for (ccg_lp0=0;ccg_lp0<mn;ccg_lp0++) {
   *pc++ = (float) (*pmr++);
   *pc++ = (float) (*pmi++);
   }
}

/* Second One: Real */
if (pr!=NULL && pi==NULL) {
for (ccg_lp0=0;ccg_lp0<mn;ccg_lp0++) {
   *pc++ = (float) (*pmr++);
   *pc++ = 0;    /* setting Imag part to zero */
   }
}

status = cfhswrtfloat (file_format, cossfp, pcoss, 2 * mn);
if (status)  {
  fprintf (stderr, "\n Cannot write data: %s in %s\n",coss_file,coss_file);
```

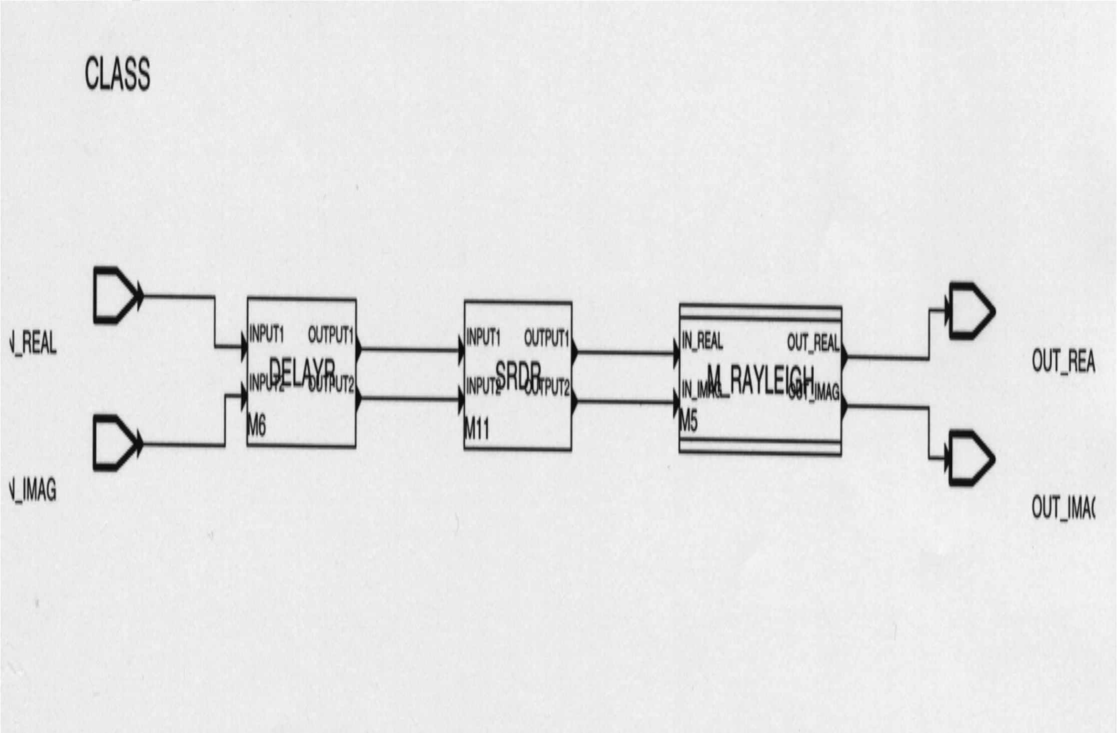
```
          exit(1);
          }


        /* Report of what has been done */
        printf("\tNumber of elements from MATLAB %s variable are
%i\n",pdir[i],2*mn);
        printf("\tConverted to <%s> COSSAP format\n\n",coss_format);

       /* Deallocate Memory */
       mxFree(pmx);
       mxFree(pr);
       mxFree(pi);
       free((char*) pcoss);

       } /*FOR LOOP*/

       /* close all files */
       status = cfhclose (file_format, cossfp, "KEEP");/* Cossap file */
       if (status)  {
         fprintf (stderr, "\n Problems closing COSSAP <%s> file!\n",coss_file);
         exit(1);
         }


       if (matClose(pmat) != 0) {
         fprintf(stderr,"\n Problems closing MATLAB <%s> file!\n",mat_file);
         }

       mxFree(pdir);
       printf(" CONVERSION DONE !!\n");
       return;
}
```

APPENDIX B.1.

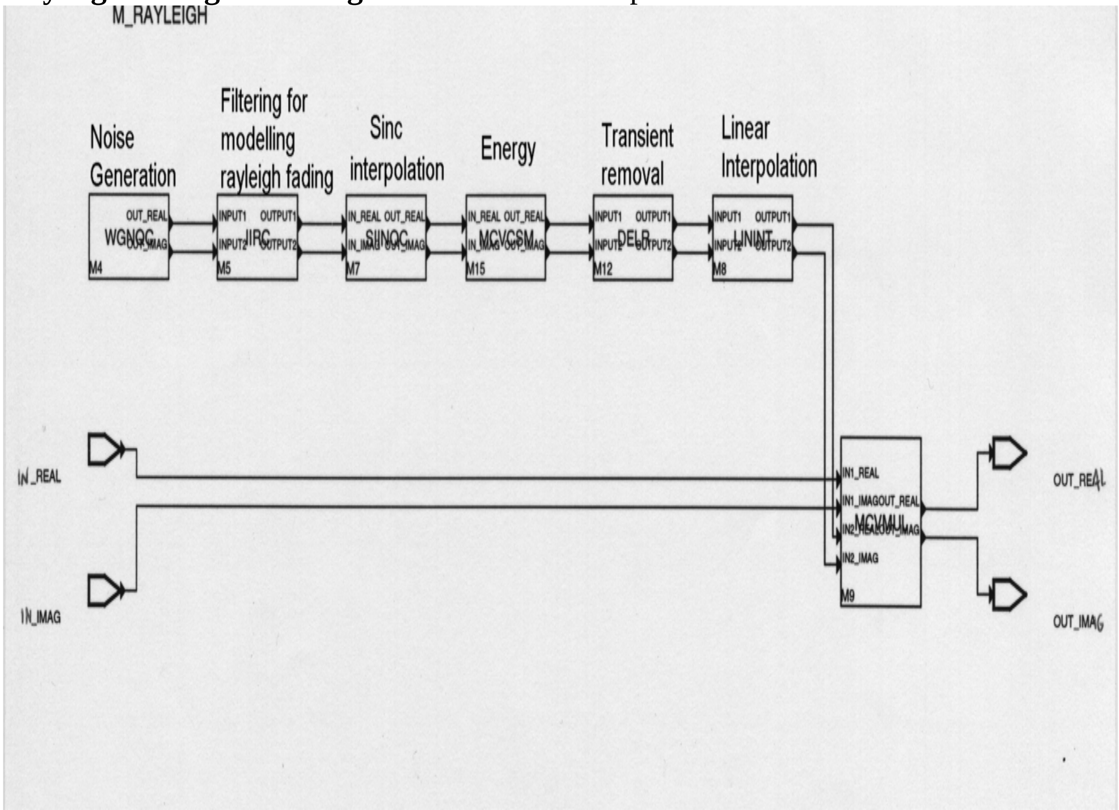
COSSAP GSM model
TYPICAL URBAN TERRAIN, 6 TAPS MODEL. RECOM. GSM 05.05



1 TAP implementation: Classical Rayleigh fading

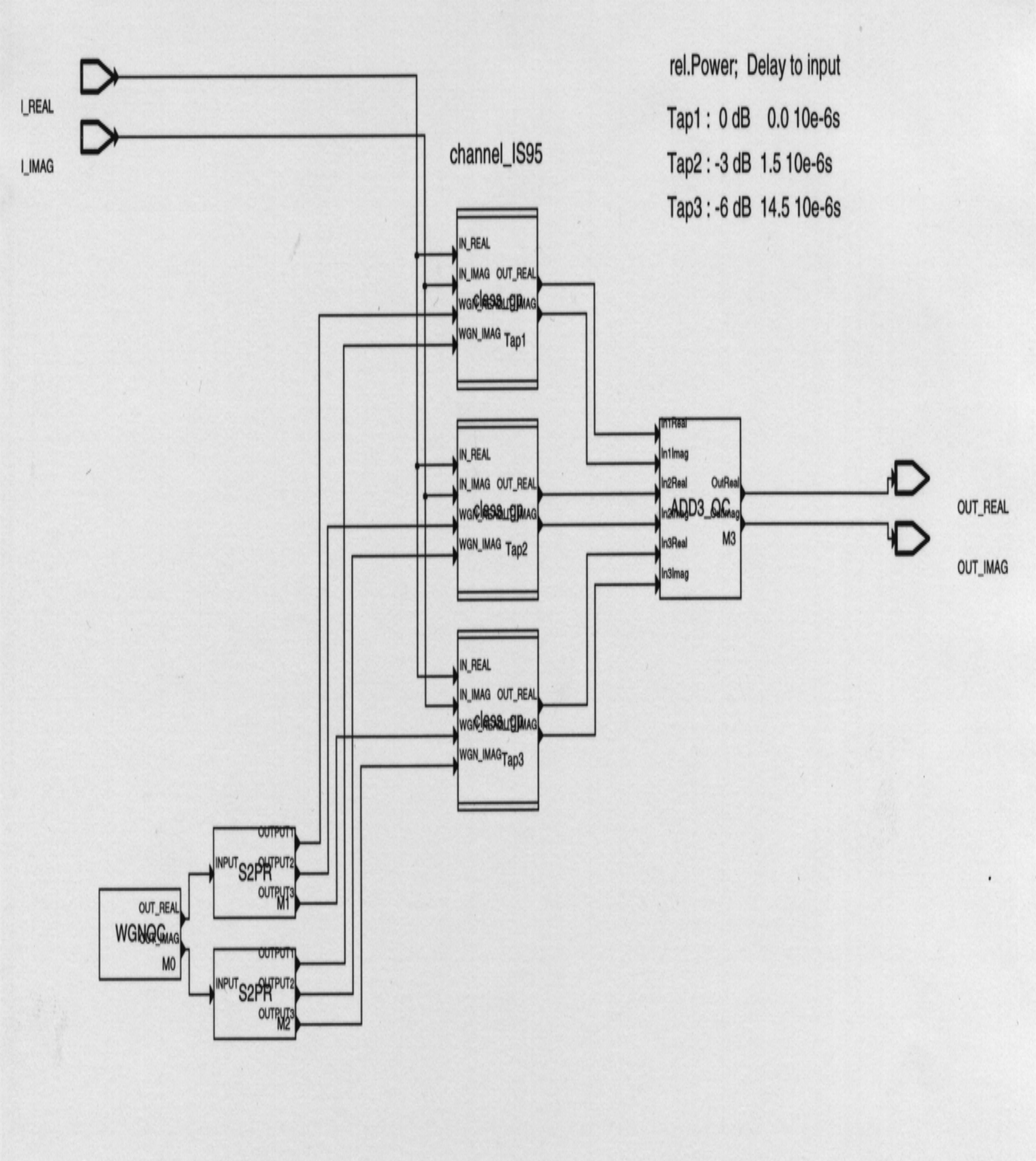


Rayleigh fading modelling: Sinc & Linear Interpolation.

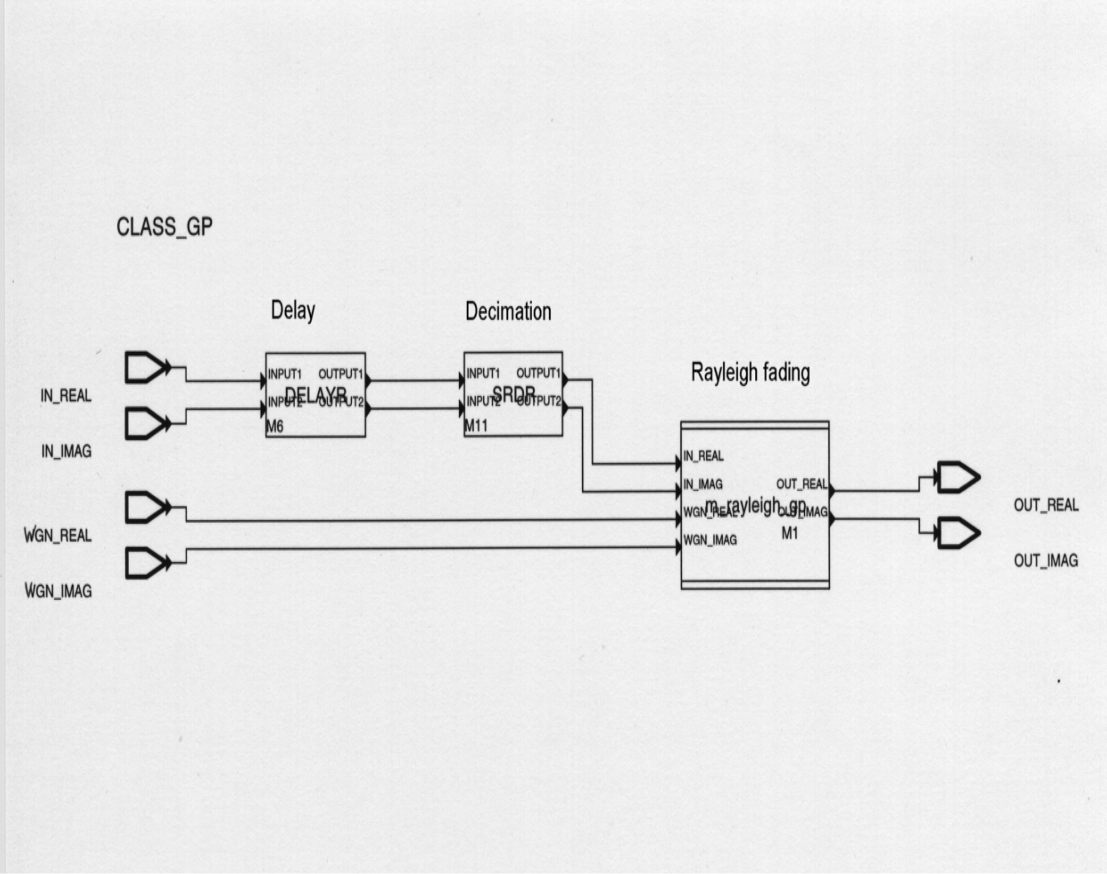


APPENDIX B.2.

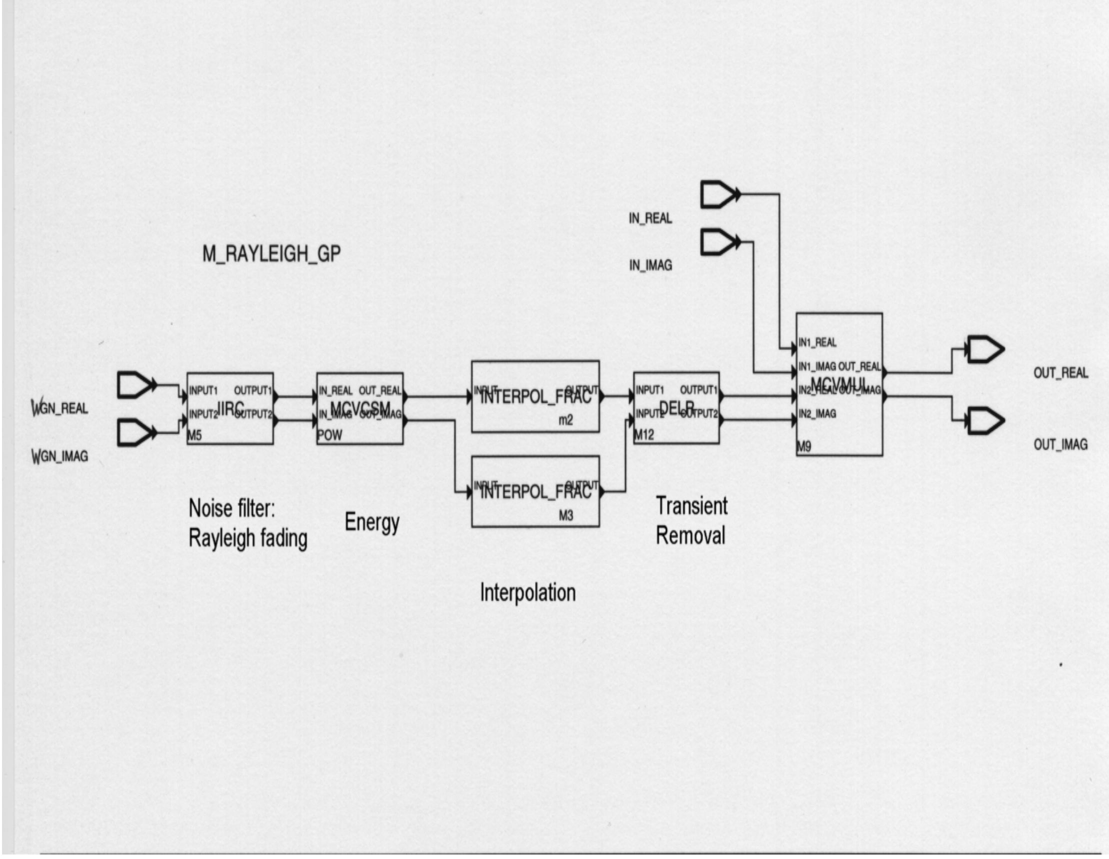
CDMA IS-95 MODEL



1 TAP implementation: Classical Rayleigh fading

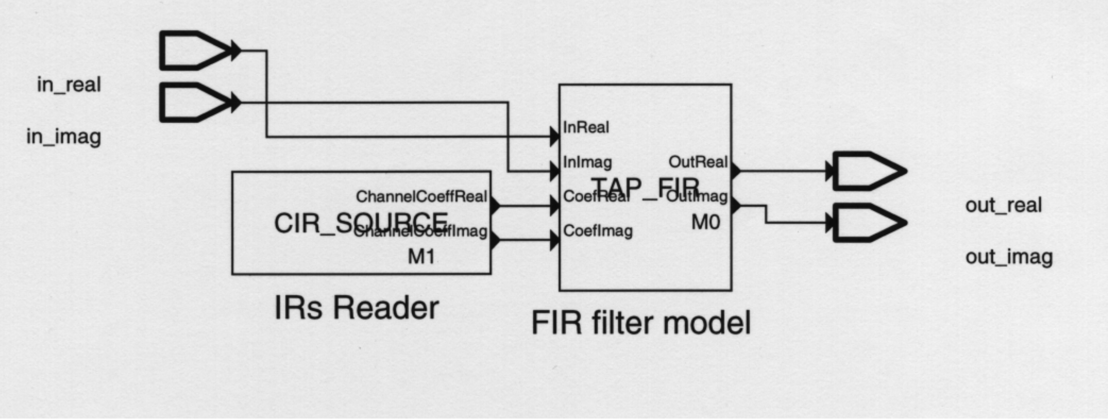


Rayleigh fading modelling: Fractional Interpolation



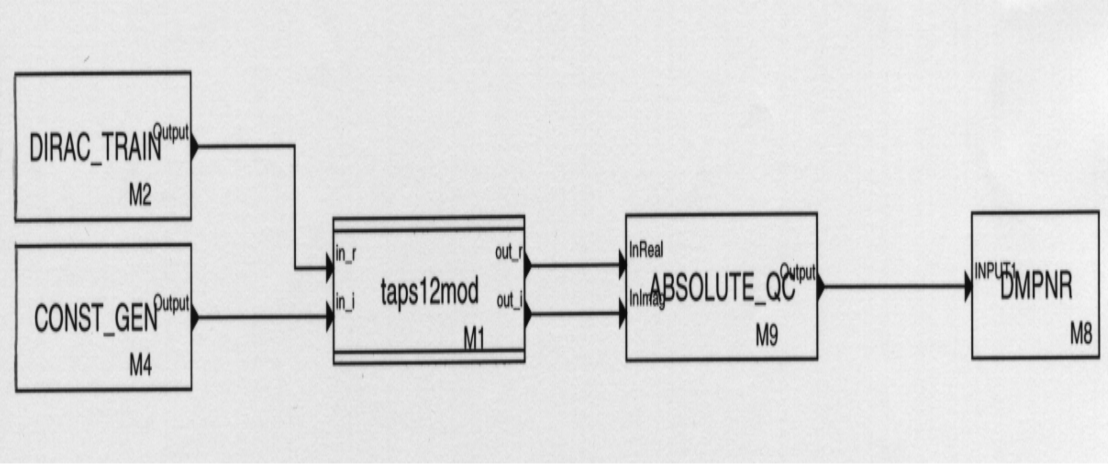
APPENDIX B.3.

WIRCS MODEL

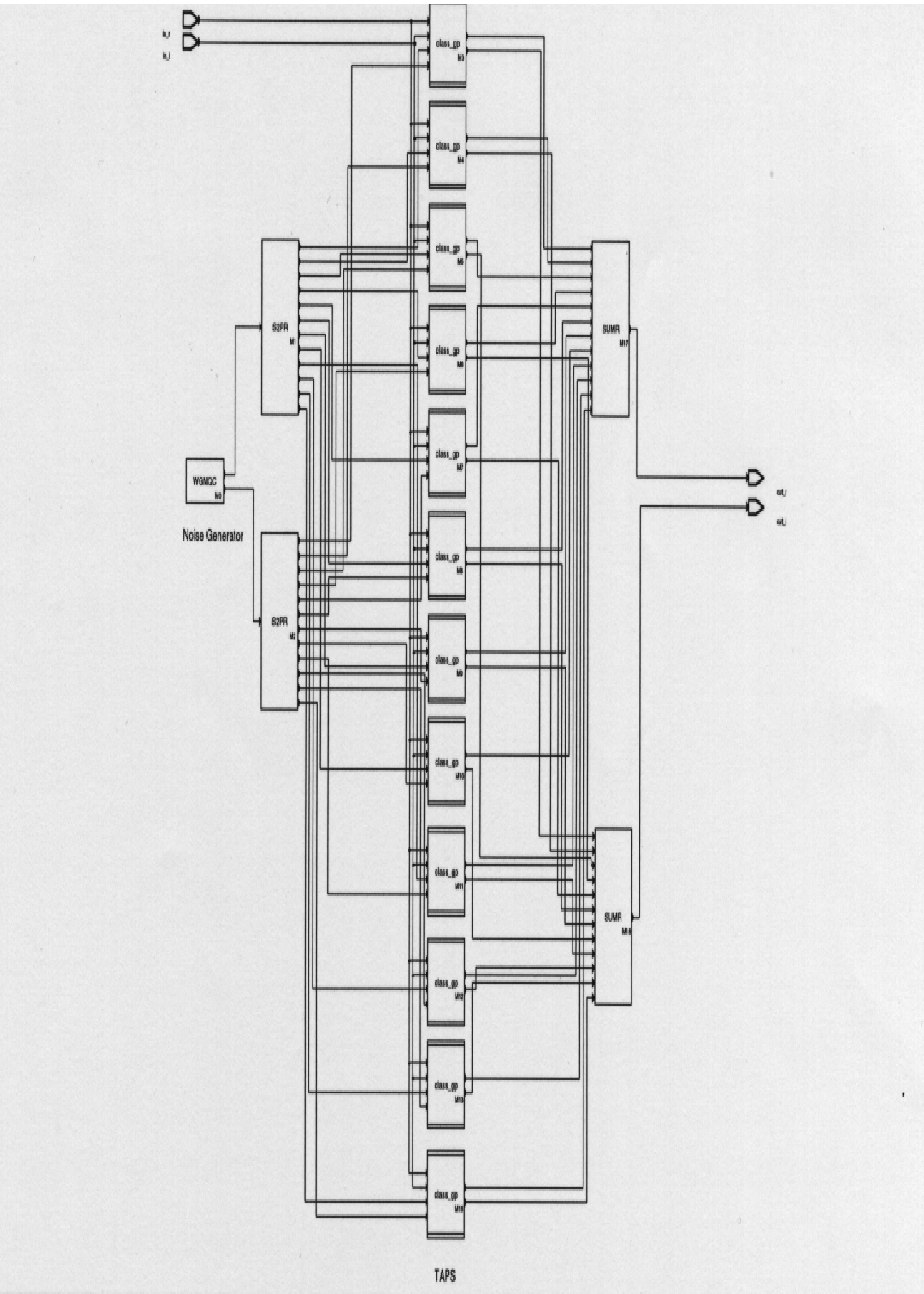


APPENDIX B.4.

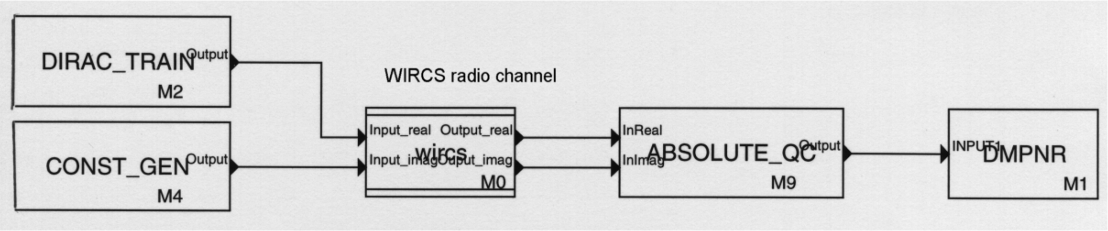
TEST ENVIRONMENT FOR STATISTICAL RADIO CHANNEL MODEL



12 TAPS model.



APPENDIX B.5
TEST ENVIRONMENT FOR WIRCS RADIO CHANNEL MODEL



APPENDIX C

SIMULATION PARTIAL REPORT EXAMPLE

```
Simulations performed with the following parameters:
      - 40 million samples
      - Mobile at a 3 m/s speed
      - Multipath consisting on 6 taps
      - Sampling rate: 1.2288 MHz
      - Carrier frequency: 2.154 GHz
      - Measured IR generated at 100 IRs/λ
Reports including duration and number of activation of each block
Simulation number 629 using measured data
Simulation number 630 using the statistical channel model
```

```
                   **********************
                   COSSAP report generator
                   **********************

 summary of simulation number 629
 elapsed time  0 00:11:22.577  CPU time  0 00:00:45.210

 configuration by unknown_user
               on unknown_installation
               running COSSAP release 1998.02
               on 12-oct-1998 07:25:08

 executed      by unknown_user
               on unknown_installation
               running COSSAP release 1998.02
               on unknown

 destination   is unknown_user
               on unknown_installation
               running COSSAP release 1998.02
               on 12-oct-1998 10:36:41

 model name                 instance name        activations
 ===========================================================
 MOD63:DMPNR                M5                         70797
 DSP:DIRAC_TRAIN            M2                         78432
 CHAN_LIB:TAP_FIR           M1                         70797
 CHAN_LIB:cir_source        M0                           833
 DSP:CONST_GEN              M4                         78126
 DSP:ABSOLUTE_QC            M3                         70797
```

```
                   **********************
                   COSSAP report generator
                   **********************

 summary of simulation number 630
 elapsed time  0 00:19:59.965  CPU time  0 00:06:27.700

 configuration by unknown_user
               on unknown_installation
               running COSSAP release 1998.02
               on 12-oct-1998 07:39:00

 executed      by unknown_user
               on unknown_installation
               running COSSAP release 1998.02
               on unknown
```

```
destination   is unknown_user
              on unknown_installation
              running COSSAP release 1998.02
              on 12-oct-1998 10:59:16
model name                 instance name        activations
===========================================================
DSP:ABSOLUTE_QC            M9                       1007666
DSP:CONST_GEN              M4                         78125
MOD63:DMPNR                M1                         78125
DSP:DIRAC_TRAIN            M2                         78432
MOD63:WGNQC                M0.M0                        909
MOD63:SRDR                 M0.M3.M11                 156250
MOD63:DELAYR               M0.M3.M6                  156250
MOD63:IIRC                 M0.M3.M1.M5                  909
MOD63:DELR                 M0.M3.M1.M12               77500
MOD63:MCVCSM               M0.M3.M1.POW                 909
MOD63:MCVMUL               M0.M3.M1.M9               233436
GSM:INTERPOL_FRAC          M0.M3.M1.M3                77500
GSM:INTERPOL_FRAC          M0.M3.M1.M2                77500
MOD63:SRDR                 M0.M4.M11                 156251
MOD63:DELAYR               M0.M4.M6                  156251
MOD63:IIRC                 M0.M4.M1.M5                  909
MOD63:DELR                 M0.M4.M1.M12               77500
MOD63:MCVCSM               M0.M4.M1.POW                 909
MOD63:MCVMUL               M0.M4.M1.M9               233436
GSM:INTERPOL_FRAC          M0.M4.M1.M3                77500
GSM:INTERPOL_FRAC          M0.M4.M1.M2                77500
MOD63:SRDR                 M0.M5.M11                 156251
MOD63:DELAYR               M0.M5.M6                  156251
MOD63:IIRC                 M0.M5.M1.M5                  909
MOD63:DELR                 M0.M5.M1.M12               77500
MOD63:MCVCSM               M0.M5.M1.POW                 909
MOD63:MCVMUL               M0.M5.M1.M9               233437
GSM:INTERPOL_FRAC          M0.M5.M1.M3                77500
GSM:INTERPOL_FRAC          M0.M5.M1.M2                77500
MOD63:SRDR                 M0.M11.M11                155638
MOD63:DELAYR               M0.M11.M6                 155638
MOD63:IIRC                 M0.M11.M1.M5                 909
MOD63:DELR                 M0.M11.M1.M12              77500
MOD63:MCVCSM               M0.M11.M1.POW                909
MOD63:MCVMUL               M0.M11.M1.M9              232826
GSM:INTERPOL_FRAC          M0.M11.M1.M3               77500
GSM:INTERPOL_FRAC          M0.M11.M1.M2               77500
MOD63:SRDR                 M0.M12.M11                155638
MOD63:DELAYR               M0.M12.M6                 155638
MOD63:IIRC                 M0.M12.M1.M5                 909
MOD63:DELR                 M0.M12.M1.M12              77500
MOD63:MCVCSM               M0.M12.M1.POW                909
MOD63:MCVMUL               M0.M12.M1.M9              232825
GSM:INTERPOL_FRAC          M0.M12.M1.M3               77500
GSM:INTERPOL_FRAC          M0.M12.M1.M2               77500
MOD63:SRDR                 M0.M13.M11                155025
MOD63:DELAYR               M0.M13.M6                 155025
MOD63:IIRC                 M0.M13.M1.M5                 909
MOD63:DELR                 M0.M13.M1.M12              77500
MOD63:MCVCSM               M0.M13.M1.POW                909
MOD63:MCVMUL               M0.M13.M1.M9              232212
GSM:INTERPOL_FRAC          M0.M13.M1.M3               77500
GSM:INTERPOL_FRAC          M0.M13.M1.M2               77500
DSP:ADD3_QC                M0.M6                     544716
DSP:ADD3_QC                M0.M14                    542883
MOD63:S2PR                 M0.M1                        909
MOD63:S2PR                 M0.M2                        909
DSP:ADD2_QC                M0.M15                   1007666
MOD63:FORKR                M0.FORKR_SIG_14            78125
MOD63:FORKR                M0.FORKR_SIG_15            78432
```