AALBORG UNIVERSITET

Estimation of the Number of Multipath Components in a Delay-Dispersive Environment for LTE OFDM Downlink

Vincenzo Malta

Aalborg University
Department of Electronic Systems

February 2010

"Dont worry about the future. Or worry, but know that worrying is as effective as trying to solve an algebra equation by chewing bubble gum..."

The Big Kahuna

AALBORG UNIVERSITET

Abstract

Aalborg University
Department of Electronic Systems

by Vincenzo Malta

Orthogonal Frequency Division Modulation (OFDM) is the modulation employed by the LTE downlink physical layer. Recent studies on pilot-aided channel estimation, typical of such modulation, show that assuming a parametric channel model can effectively improve the estimate accuracy. The algorithms based on this model perform the frequency response estimation of a multipath environment through the knowledge of the tap-delay parameters, i.e., the delays' values and their number. This thesis focuses on the estimation of the number of delays, required by the tap-delay estimators, e.g., ESPRIT. For this purpose the state-of-the-art MDL detection scheme is investigated over one of the proposed LTE channels and over a dynamic channel. Furthermore, it is compared with an approach based on a simple threshold operation. The study is carried out jointly with the preprocessing scheme named spatial smoothing. It's shown that over the static channel, MDL exhibits a certain robustness against the time observation reduction, while the *Threshold* algorithm suffers from a significant increase in the estimate's error. Tests over the dynamic channel show a general performances decrement. However MDL average error keeps constant within a wide window size range and it doesn't drastically get worse as the *Threshold* one. The state-of-the-art performances then, are not exceeded, but the noticed problems still leave this topic open.

Acknowledgements

First of all I would like to express my gratitude to Bernard Fleury and Velio Tralli. Thank you for advising me Aalborg University, this thesis and for allowing me to spend these last five months in the NavCom section. Then, against their will, thank you Carles and Niels, I hope it hasn't been too hard to supervise me. Thanks to my office mate Troels and to all the NavCom guys. Finally, thanks to my family, to Federica, to all of my old and new friends: it would have never been the same without you!

Vincenzo Malta
Aalborg, February 2010

Contents

List of Figures

List of Tables

Abbreviations

3GPP	**3**rd **G**eneration **P**artnership **P**roject
AIC	**A**kaike **I**information **C**riterion
BER	**B**it **E**rror **R**ate
CDMA	**C**ode-**D**ivision **M**ultiple **A**ccess
DFT	**D**iscrete **F**ourier **T**ransform
DPSK	**D**ifferential **P**hase-**S**hift **K**eying
EPA	**E**xtended **P**edestrian **A**
DOA	**D**irection **O**f **A**rrival
ESPRIT	**E**stimation (of) **S**ignal **P**arameters (via) **R**otational **I**nvariance
ETU	**E**xtended **T**ypical **U**rban
EVA	**E**xtended **V**ehicular **A**
FFT	**F**ast **F**ourier **T**ransform
GMSK	**G**aussian **M**inimum **S**hift **K**eying
GSM	**G**lobal **S**ystem (for) **M**obile (Communications)
ICI	**I**nter-**C**arrier **I**nterference
IDFT	**I**nverse **D**iscrete **F**ourier **T**ransform
IEEE	**I**nstitute (of) **E**lectrical (and) **E**lectronics **E**ngineers
IFFT	**I**nverse **F**ast **F**ourier **T**ransform
ISI	**I**nter-**S**ymbol **I**nterference
LMMSE	**L**inear **M**inimum **M**ean **S**quare **E**rror
LTE	**L**ong **T**erm **E**volution
LOS	**L**ine **O**f **S**ide
LS	**L**east **S**quares
MIMO	**M**ultiple **I**nput **M**ultiple **O**utput
MQAM	**M**ulti(-Level) **Q**uadrature **A**mplitude **M**odulation

MSE	**M**inimum **S**quare **E**rror
NLOS	**N**on **L**ine **O**f **S**ide
OFDM	**O**rthogonal **F**requency-**D**ivision **M**ultiplexing
OFDMA	**O**rthogonal **F**requency-**D**ivision **M**ultiple **A**ccess
PACE	**P**ilot-**A**ided **C**hannel **E**stimation
PSAM	**P**ilot **S**ymbol **A**ssisted **M**odulation
QAM	**Q**uadrature **A**mplitude **M**odulation
QPSK	**Q**uadrature **P**hase-**S**hift **K**eying
SBA	**S**equential **B**eamforming **A**lgorithm
SNR	**S**ignal-(to)-**N**oise **R**atio
SVD	**S**ingular **V**alue **D**ecomposition
TDMA	**T**ime-**D**ivision **M**ultiple **A**ccess
UMTS	**U**niversal **M**obile **C**ommunication **S**ystem
US	**U**ncorrelated **S**cattering
WiMAX	**W**orldwide **i**nteroperability (for) **M**icrowave **A**ccess
WSS	**W**ide **S**ense **S**tationary
WSSUS	**W**ide **S**ense **S**tationary **U**ncorrelated **S**cattering
ZF	**Z**ero **F**orcing

Dedicated to Aalborg...

Chapter 1

Introduction

The interest concerning the wireless communication systems, in the last twenty years has led to a real revolution in terms of speed, reliability and quality of service. This has been due to the improvement of the physical and the access layer, through the introduction of increasingly pushed modulation techniques, advanced coding and smart channel access protocols [1]. Talking just about mobile communications, the three standards that have grounded the respective generations are GSM, UMTS and LTE (respectively 2nd, 3rd and 3.9th), with a multitude of intermediate standards, placed between them. GSM, that is still the most diffused standard on the planet, introduced the digital through a simple *Gaussian Minimum Shift Key* (GMSK) modulation, with a *Time Division Multiple Access* (TDMA) technique to yield a multiple channel access. Then, the bigger innovation carried by UMTS is the access, that exploits the so-called *Code Division Multiple Access* (CDMA), which considerably increases the speed (from Kbps to Mbps), while the modulation levels are still four (QPSK). The real improvement anyway is going to be seen with the introduction of *Long Term Evolution* (LTE). The downlink transmission scheme supports up to a 64QAM and accesses the channel employing jointly both the time and the frequency domain, through the recently introduced *Orthogonal Frequency Division Multiple Access* (OFDMA), realized by an *Orthogonal Frequency Division Multiplexing* (OFDM) architecture. Although such technique has been proposed in the sixties, only with the recent technology, a useful implementation can be attained. For this reason the first standard that introduced it has been the IEEE 802.11a, just to modulate the data stream though. OFDMA indeed, appeared for the first time in WiMAX, but it's been overshadowed by LTE due to its better performances. Furthermore, the introduction of the MIMO technology, which make use of multiple antennas, has led to even higher data rates.

Although the standard specifications have already been frozen in [2], the 4th generation of mobile terminals, i.e., LTE-Advanced, is still subject to studies and although several issues occur, the topic of this thesis will cover just the channel estimation. It has been inspired by the previous works by Morten Lomholt Jakobsen [3] and Kim Laugesen [4] and aspires to be their continuation, in order to investigate an aspect regarding the channel estimation in OFDM systems that have not been studied in depth, i.e., the estimation of the number of delays in a multipath environment. They proposed the ESPRIT and the SBA algorithms as tap-delay estimators, but assuming the number of delays known, that is not the real case. It's why some of the actual number of delays detection schemes are going to be tested and compared to an empirical trivia algorithm, that has the same purpose. In addition, their behavior will be studied over a dynamic channel, self-developed in [3] and [4], that should better approximate the real one. This is because, despite its time-varying nature being evident, the actual state-of-the-art estimation algorithm assumes the channel is static. LTE is just the most recent system that motivates this study, whose results can be applied to any other future project with the same purposes and that will exploit the same transceiver structure. This first chapter gives an outline of the thesis, after an introduction about the multipath channel estimation for OFDM systems, which will be detailed in the next chapters.

1.1 Channel Estimation

In general, every wireless transmission system suffers the effect of the propagation medium under two main aspects: that of the channel distortion and of noise corruption. About the latter, there is nothing to do, it's just modeled as an additive random process and taken into account within the performances. However, the first can be somehow compensated in order to reconstruct the transmitted signal. A good transceiver's performance then, should depend only on the noise and not by the distortion induced by the channel. To briefly see this, an important relationship in the frequency domain that will be derived in chapter 2 is anticipated:

$$\boldsymbol{y} = \mathbf{H}\boldsymbol{x} + \boldsymbol{v}, \tag{1.1}$$

where $\boldsymbol{y}$ is the received signal, $\mathbf{H}$ represents the channel, $\boldsymbol{x}$ is the transmitted signal and $\boldsymbol{v}$ the noise. Now, roughly speaking, it's easy to see that dividing the received signal by $\mathbf{H}$, if it was known, the only factor that would keep corrupting the signal would be the noise. The purpose of the channel estimation, is to look for the estimate that best approximates $\mathbf{H}$, to yield the mentioned compensation, always referred as *equalization*. At a certain instant (it could change over time), it can be performed in both time and

frequency domain, because as is well known, the Fourier transform allows a switch from one domain to another. The actual state-of-the-art channel estimator for an OFDM system is the *Wiener Filter* in its robust setup. It directly reconstructs the frequency response over the whole bandwidth through an interpolation, aided simply by some frequencies observations. The meaning of observation will be clearer afterwards; for the moment it's enough to know that it's obtained by the transmission of data, know from both the transmitter and the receiver.

1.2 Multipath Mobile Environment

First of all, the channel that will be considered is wireless, thus as well known, it suffers the multipath effect, i.e., when more than one copy of the transmitted signal reaches the receiver with different delays and attenuations, as a result of the interactions with the surrounding environment. How these interactions occur is out of the topic of the thesis, but to understand that each type of environment has its own statistical multipath characterization it's intuitive enough. Now the most general case provides that the channel parameters are not fixed during the whole transmission time, even if sometimes a static approximation can be accepted. However one of the telecommunications' target is to improve the system performance also operating in a highly mobile environment, where by definition the channel is time-varying. This, in fact, will be the considered scenario, not compulsorily high speed, but absolutely not static. The equalization then, has to take it into account, because it could lead to a signal reconstruction error if the channel response changes within the estimation time. This will be simulated over both the static and the dynamic channel. How to modify the estimation time will be clear after introducing some estimation backgrounds in chapter 4. Now, static or dynamic apart, the multipath issues arise in any case and due to the robustness against them, the OFDM has been chosen.

1.3 OFDM Overview

The idea that stays behind the OFDM (qualitatively shown in figure 1.1), is essentially to use more than one carrier to modulate the symbols, that are obtained mapping the information bits according to any arbitrary multi-level modulation. Thus, the original bandwidth can be kept, with the difference that the original data rate would be split into several lower data rates, each one corresponding to a different carrier, or rather *subcarrier*. To do this the original symbols have to be serial to parallel converted to produce a number of data streams equal to the number of the subcarriers. After have

being modulated, they are summed together to produce the composite signal that is transmitted over the channel. Of course the dual operation has to be performed at the receiver. Anyhow, looking at the problem from the time point of view, a lower data rate corresponds to a larger symbol duration, that will be shown to be an important property within the multipath scenario. This is indeed, one of the main problems that limits the system speed.

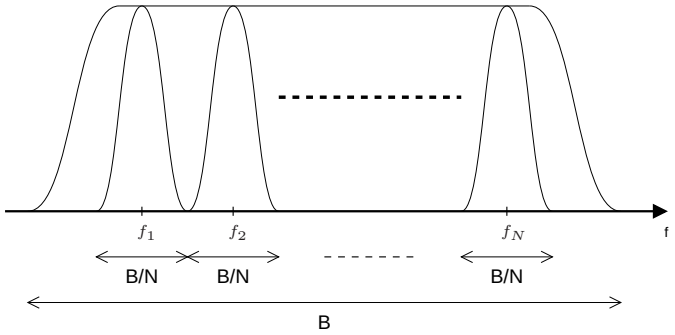

FIGURE 1.1: OFDM qualitative bandwidth splitting.

Looking at the figure, it could seem that the original bandwidth couldn't be kept, due to the absence of any kind of spectrum compacting, necessary to allow the subchannels separation by a bank of filters. Conversely, the different subchannels overlap, yielding a really compact spectrum. This is possible because of the modulation orthogonality property and in addition, in the real setup there is no need of any filter to separate them. It will be shown that to realize such idea, the digital setup makes use of an handy IDFT and DFT, respectively at the transmitter to produce the composite signal, and at the receiver to obtain the original data streams. The contribution of OFDM to the channel estimation, is that the above mentioned *known data* are transmitted just by symbols over different frequencies, i.e., modulating only certain subcarriers. These symbols are referred to as *pilots* and properly distributed in both time and frequency they can lead to an appropriate estimation.

1.4 Problem Statement

Thus far, the scenario, the purpose and the scheme to attain it have been introduced. The two main problems, are that the actual state-of-the-art algorithm assumes the channel static within the estimation time and that it requires the knowledge of some channel information like its statistic and the *Signal-to-Noise Ratio* (SNR). Although it's very robust to the mismatch, when a mismatch occurs there is an irreducible error floor that can't be avoided. Regarding the first problem, a dynamic channel instead of the

proposed static one, will lead to be considered. While about the second, a guideline has already been given by the previous works [3] [4] following a tap-delay estimation approach. However SNR and the number of taps have been assumed as well, by the proposed algorithms. The assumption about the first is not that unrealistic. In fact the receiver has to track the noise statistic in order to give feedback to the transmitter (talking about downlink), so it should be known. On the other hand, it's not real that the number of echoes is known before analyzing the received signal. The purpose of this research is to examine the existing schemes, well known in the channel estimation field to detect the number of delays, to test the here proposed *Threshold* algorithm.

The thesis contains eight chapters, grouped into three primary sections, aside from this introduction. Chapters 2, 3 and 4 contain the theory which is essential in introducing the problem. Chapters 5 and 6 present the algorithms, necessary to examine in order to improve the estimation. Finally chapters 7 and 8 report and conclude the research.

Chapter 2

OFDM System Model

It's well known that the channel transfer function, i.e., roughly speaking the frequency relationship between the received and the transmitted signal, can have a multitude of shapes, that in general lead to different attenuations for different frequencies. When it occurs, the channel is said to be *frequency-selective*. Thus at the receiver, this fact has to be taken into account and somehow compensated (*equalized*) to reconstruct the original signal, with the further problem of the noise corruption. The only way to do it is to estimate the channel response from the received signal, but this will be argument of chapter chapter 4. *Orthogonal Frequency Division Multiplexing* (OFDM) is a modulation technique based on the idea of splitting the channel into a defined amount N of narrowband and independent *subchannels* that are supposed to have a flat frequency response (figure 2.1), of course different for each subchannel.

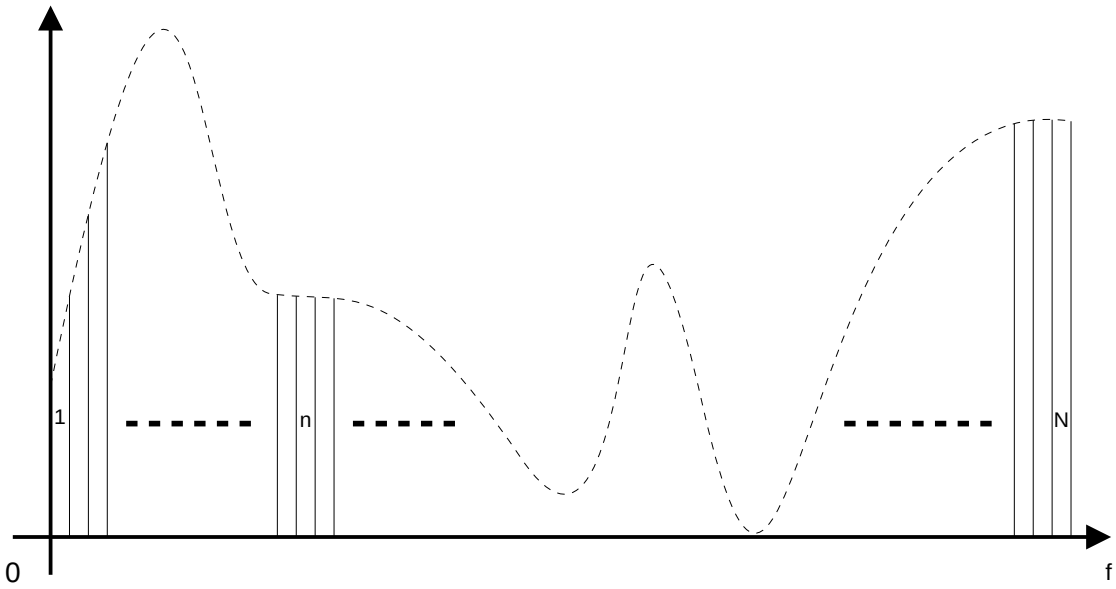

FIGURE 2.1: Channel transfer function for each subchannel.

Looking at the equalization problem from this point of view, such set of independent "flat" channels is easier to treat. By transmitting a narrowband signal, known from either the transmitter and the receiver, it would be possible to obtain the channel response at the signal frequency, simply observing the ratio between received and the known transmitted signal. Considering a wideband signal and channel contrariwise, the same treatment is not that immediate.
The frequency selectivity derives from that environment identified as *delay-dispersive* or *multipath*, i.e., when more than a copy of the transmitted signal, each with a different delay and attenuation factor, reaches the receiver, due to the reflections, whose behavior will be described in chapter 3. This fact highly limits high-data-rate transmission systems, and is the reason why OFDM is proper for such environments: due to the channel splitting, the signal is transmitted indeed over parallel low-data-rate subchannels. The bandwidth of each subchannel, and hence their number, depends on some parameters: the most significant one is the *delay spread*, that for the moment will be considered as a sort of indicator of the channel time distortion, motivated in chapter 3. Intuitively, due to the signal echoes, the pulse shape will suffer a spreading in time, interfering with the adjacent transmitted pulses. In this regard, the symbol duration (N times larger, after the S/P conversion) of the narrowband channels, i.e., the inverse of their bandwidth, must be larger than this spreading, in order to mitigate the *Inter-Symbol Interference* (ISI), a phenomenon that will be illustrated in chapter 3 as well. Furthermore, it will be shown how this issues can be removed using the OFDM scheme, by an artifice named *cyclic prefix*.

After introducing the working principle, the real implemented digital setup will be derived, in order to introduce a linear algebraic signal model. The last section then, provides the LTE downlink configuration, comprising the time domain frame structure, the pilot pattern and the OFDM parameters specification.

2.1 Orthogonal Frequency-Division Multiplexing

To understand how the system works it's useful to consider the scheme proposed in figure 2.2: information bits, mapped onto symbols, according to a certain digital modulation (e.g., DPSK, QPSK, MQAM), are Serial-to-Parallel converted into N data streams; N local oscillators (whose frequency is $f_n = nB/N$, where B is the total available bandwidth) are available and the respective frequency, i.e., *subcarrier*, is then modulated by the corresponding symbols. The composite signal, i.e., the sum of all modulated

subcarriers, is the signal that has to be transmitted over the channel:

$$s(t) = \sum_{i=-\infty}^{\infty} s_i(t) = \sum_{i=-\infty}^{\infty} \sum_{n=0}^{N-1} x_{i,n}\xi_n(t - iT_u), \tag{2.1}$$

where $x_{i,n}$ is the complex transmitted symbol, at time instant i on the nth subcarrier, T_u is the useful OFDM symbol duration (why useful will be clear afterwards), i.e., $T_u = NT_S$, where T_S is the original symbol duration and ξ_n is a rectangular frequency-shifted pulse:

$$\xi_n(t) = \begin{cases} \frac{1}{\sqrt{T_u}} e^{j2\pi n\Delta f t} & \text{for } 0 < t < T_u \\ 0 & \text{otherwise} \end{cases}, \tag{2.2}$$

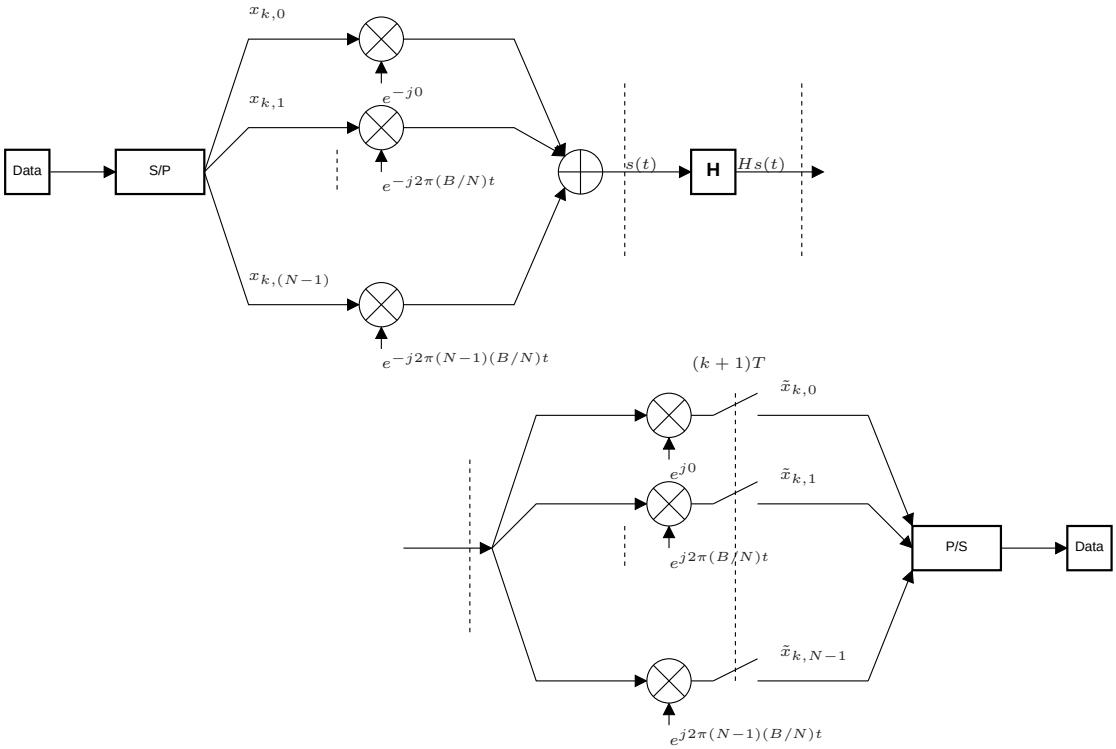

FIGURE 2.2: Analog OFDM system.

However this scheme has been introduced just like an explanation because for the LTE purpose (as for all actual purposes), when e.g., $N = 1200$ subcarriers are required, it's clear that a multi-oscillators design is not achievable.

2.1.1 Subchannels Spacing and Frequency Orthogonality

In order to increase the spectral efficiency, the subcarrier spacing is $\Delta f = B/N = 1/T_u$, even if there's a subchannel overlap, due to the rectangular shape of $\xi(t)$. It corresponds indeed, to a *sinc* function in the frequency domain that has spectral nulls

in correspondence of other subcarrier frequencies. Thus, since (2.3) holds, the receiver can separate the channels due to their orthogonality. The inner product orthogonality definition in fact:

$$\frac{1}{T_u}\int_{iT_u}^{(i+1)T_u} e^{j2\pi f_k t}e^{-j2\pi f_h t}dt = \begin{cases} 1 & \text{for } k=h \\ 0 & \text{otherwise} \end{cases}, \tag{2.3}$$

states that there's no interference between different subchannels assuming $f_0 = 0$, with $f_k = kB/N = k\Delta f$ denoting the kth subcarrier frequency. However, sometimes the frequency synchronization is not perfect and this yields a loss of orthogonality between the subcarriers producing the so-called *Inter-Carrier Interference* (ICI), that won't be treated here, though.

2.1.2 Modulation Scheme

The real transceiver structure depicted in figure 2.3, is different from the one treated in the introduction, due to the not possible hardware implementation.

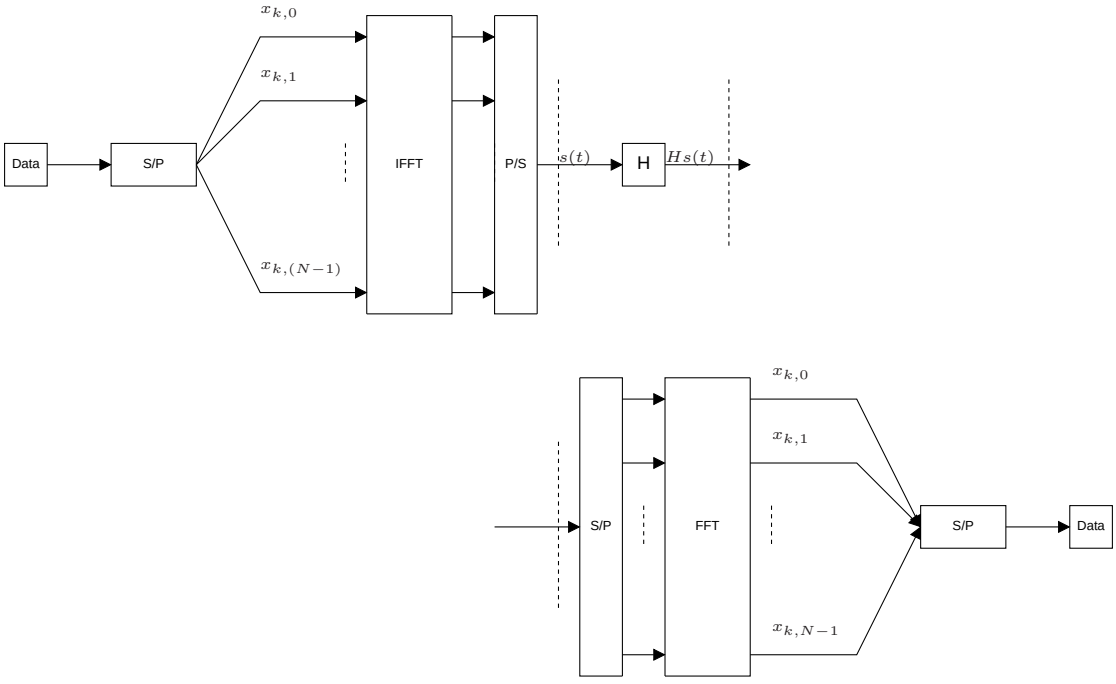

FIGURE 2.3: Digital OFDM system.

According to the definition of composite signal $s(t)$, a sampling can be performed at instances $t_k = kTs$. Passing over the time instant index i, it can be written as:

$$s_k = s(t_k) = \frac{1}{\sqrt{T_S}}\sum_{n=0}^{N-1} x_n e^{j2\pi n\frac{k}{N}}, \tag{2.4}$$

which is the *Inverse Discrete Fourier Transform* (IDFT) of the N parallel input symbols, at a certain time instant i. Thus the transmitter works performing an IDFT of the parallelized symbols (an IFFT in practical cases) at each time instant, then serializing them to build the composite signal $s(t)$, that's transmitted over the channel. Accordingly the receiver performs the inverse operations, i.e., the S/P, the FFT and the P/S conversion, to produce an estimate $\tilde{x}_n$.

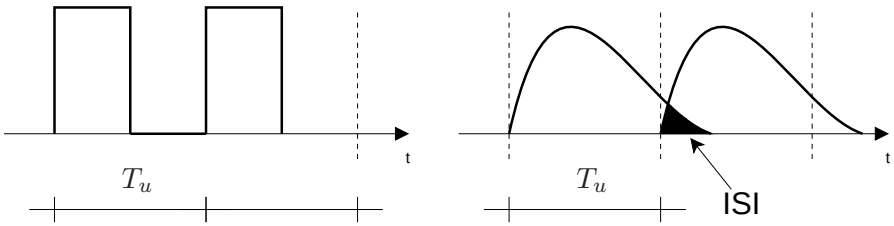

FIGURE 2.4: ISI qualitative description.

An issue that hasn't been treated yet, is the so-called ISI, which is a phenomenon that involves the received adjacent symbols. A qualitative behavior is anticipated in figure 2.4 and it consists in a pulse overlapping due to the spreading, induced by the multipath channel non-ideal impulse response, whose behavior will be described in detail in chapter 3. Intuitively it could be avoided by a guard interval at the beginning of each symbol, filled by non-signal samples, so as the tail of the preceding symbol doesn't overlap. Furthermore, it will be shown in the next subsection that these samples will be somehow related to the signal and useful for the channel equalization and to keep the subcarriers' orthogonality.

2.1.3 Cyclic Prefix

As mentioned, the number of samples of each symbol must be increased to introduce a guard interval whose duration must be longer than the impulse response of the channel, to avoid that the last part of the preceding symbol interferes with the current one, as it happens in figure 2.4. This length will be indicated as μ and the contained samples will be equal to the tail of the current symbol as illustrated in figure 2.5(a), where the time instant index has been passed over. These samples are added at the transmitter after the IFFT (figure 2.5(b)) and just discarded at the receiver before the FFT. Furthermore, besides the ISI issue, the cyclic prefix has two more objectives. The first is that it turns the linear convolution between the symbols of length $N + \mu$ and the channel impulse response, into a circular convolution of length N, considering the original symbols. It's well known that a multiplication of the *discrete Fourier coefficients* of two periodic sequences corresponds to a periodic convolution of the sequences [5]. Here the

original input $N \times 1$ sequences are not periodic, so the linear convolution cannot be applied. Therefore an artificial periodicity is introduced by the cyclic prefix yielding a multiplication in the frequency domain, between the signal and the channel frequency response. This artifice simplifies the equalization, i.e., to obtain the transmitted signal is immediate, by multiplying the Fourier transforms of received signal, by the inverse of the channel estimate on the correspondent subcarrier. This will be shown in section 2.2, after the introduction of the signal model. The second cyclic prefix's interesting property, is that it keeps the frequency orthogonality avoiding the ICI, that would be present due to the multipath. This can be seen just observing (2.3) considering the first case, i.e., when $k = h$ but with one of two pulses delayed by a certain delay τ:

$$\frac{1}{T_u} \int_{iT_u}^{(i+1)T_u} e^{j2\pi f_k t} e^{-j2\pi f_h (t-\tau)} dt. \tag{2.5}$$

Now, it's evident that the product is not equal to 1 anymore, because of the displacement and similarly in the case with $k \neq h$, the product wouldn't be equal to zero, giving rise to interference. However, if the condition $0 \leq \tau \leq \mu$ holds, the cyclic prefix provides the first missing symbol's samples within the integration interval, so as (2.3) still holds. The above explanation is not strict, because only the unmodulated subcarriers have been considered, but provides a qualitative justification about the ICI cancellation.

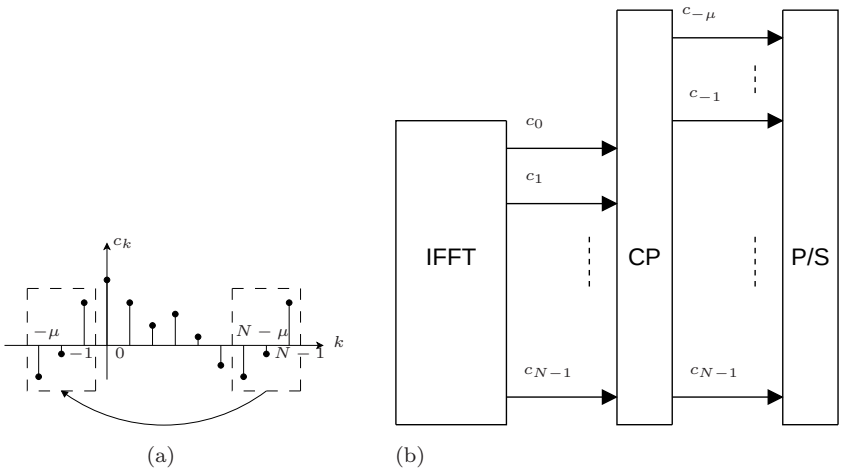

FIGURE 2.5: Cyclic prefix construction (a), and samples insertion (b).

2.2 Signal Model

The previous section served as an introduction to such a modulation technique: from that, the transceiver structure and the way that it operates in the time-frequency domain should be clear. For a more exhaustive analysis, a linear algebra representation is

preferable, to avoid long equations full of indexes, summations and exponentials. The notation suggested by the previous works [3] and [4] will be used as a reference.

First of all, the model is realized for the transmission of a single OFDM symbol. Thus, time instant indexes won't be written anymore. Furthermore some assumptions have to be made:

- there is neither ISI nor ICI, i.e., the channel impulse response is no longer than μ;
- the channel impulse response is time-invariant during the transmission of a single OFDM symbol, i.e., during the transmission of $N + \mu$ samples (quasi-stationary assumption);
- there is perfect synchronization in time and frequency between the transmitter and the receiver.

Every signal will be treated as a $N \times 1$ column vector, so for instance at a certain time instant, the input signal $\boldsymbol{x}$ will be the complex symbols vector of N elements:

$$\boldsymbol{x} = [x_0, x_1, \cdots, x_{N-1}]^{\mathsf{T}}, \tag{2.6}$$

and after the IDFT:

$$\boldsymbol{c} = [c_0, c_1, \cdots, c_{N-1}]^{\mathsf{T}}. \tag{2.7}$$

Then recalling the DFT definition:

$$\text{DFT}\{c[n]\} = x[i] = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} c[n] e^{-j2\pi n \frac{i}{N}}, \quad 0 \leq i \leq N-1, \tag{2.8}$$

by the introduction of the $N \times N$ Fourier matrix:

$$\mathbf{F} = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \dots & \omega^{(N-1)^2} \end{bmatrix}, \tag{2.9}$$

where $\omega \stackrel{\text{def}}{=} e^{-j2\pi \frac{1}{N}}$, such transformation can be written as:

$$\boldsymbol{x} = \mathbf{F}\boldsymbol{c}, \tag{2.10}$$

and it's inverse as:

$$\boldsymbol{c} = \mathbf{F}^{\mathsf{H}}\boldsymbol{x}, \tag{2.11}$$

F being a symmetric matrix and the IDFT the complex conjugated transpose of the DFT. After the above mentioned transformation, there is the cyclic prefix insertion, so the respective $(N+\mu)\times 1$ vector will be:

$$\boldsymbol{c}_{cp} = [c_{-\mu}, \cdots, c_{-1}, c_0, c_1, \cdots, c_{N-1}]^{\mathsf{T}}, \tag{2.12}$$

that is the signal that will be convolved with the channel impulse response. The channel is represented by a matrix, but an assumption has to be made: the channel impulse response samples $g[n]$, whose length is assumed equal to the cyclic prefix length μ, have to coincide with the receiver sampling time, which is not obvious. It defines a time grid indeed, whose spacing is the sample time; but the echoes' time of arrival is random and usually doesn't coincide with such grid. However, it has been shown [6] that is a reasonable assumption. The above matrix corresponding to $g[n]$, with $0 \le n \le \mu$ would be $(N+\mu)\times(N+\mu)$, to obtain an output of the same length of c_{cp}, by the definition of convolution:

$$d[n] = \sum_{m=0}^{\mu} c_{cp}[m]g[n-m], \quad -\mu \le n \le N-1, \tag{2.13}$$

but since the first μ samples are simply discarded, because they correspond to the cyclic prefix, (2.13) can be performed just for $0 \le n \le N-1$ as can be seen from the matrix:

$$\mathbf{G_{cp}} = \begin{bmatrix} g_\mu & \cdots & g_1 & g_0 & & & \mathrm{O} & \\ & g_\mu & \cdots & g_1 & g_0 & & & \\ \mathrm{O} & & \ddots & & & \ddots & & \mathrm{O} \\ & & & \ddots & & & \ddots & \\ & \mathrm{O} & & & g_\mu & \cdots & g_1 & g_0 \end{bmatrix}, \tag{2.14}$$

that is $N\times(N+\mu)$. Looking at the signal structure it's evident that there is some redundancy: the first μ elements of c_{cp}, are equal to the last by construction indeed, e.g., $x_{-\mu} = x_{N-\mu}$. For instance, the first row of the matrix accomplishes the following operation to yield d_0:

$$g_\mu c_{cp,-\mu} + g_{\mu-1}c_{cp,-\mu+1} + \cdots + g_0 c_{cp,0}. \tag{2.15}$$

That can be replaced by:

$$g_\mu c_{N-\mu} + g_{\mu-1}c_{N-\mu+1} + \cdots + g_0 c_0, \tag{2.16}$$

thus there is no need to use the vector from (2.12), it's enough the original signal without the cyclic prefix from (2.7), but (2.14) has to be reshaped to the following $N \times N$ matrix:

$$\mathbf{G} = \begin{bmatrix} g_0 & & & g_\mu & \cdots & g_1 \\ g_1 & g_0 & \mathrm{O} & & \ddots & \vdots \\ \vdots & & \ddots & & & g_\mu \\ g_\mu & & & g_0 & \mathrm{O} & \\ & \ddots & & & \ddots & \\ \mathrm{O} & & g_\mu & \cdots & g_1 & g_0 \end{bmatrix}, \tag{2.17}$$

from which the equivalence between (2.15) and (2.16) can be observed. Finally, to obtain the received signal $\boldsymbol{d}$, an adaptive white Gaussian noise $\boldsymbol{w}$ has to be added, so from the expanded form:

$$\begin{bmatrix} d_0 \\ d_1 \\ \vdots \\ d_{N-1} \end{bmatrix} = \begin{bmatrix} g_0 & & & g_\mu & \cdots & g_1 \\ g_1 & g_0 & \mathrm{O} & & \ddots & \vdots \\ \vdots & & \ddots & & & g_\mu \\ g_\mu & & & g_0 & \mathrm{O} & \\ & \ddots & & & \ddots & \\ \mathrm{O} & & g_\mu & \cdots & g_1 & g_0 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{N-1} \end{bmatrix} + \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_{N-1} \end{bmatrix}, \tag{2.18}$$

follows the compact form:

$$\boldsymbol{d} = \mathbf{G}\boldsymbol{c} + \boldsymbol{w}. \tag{2.19}$$

The next step is how to write the complex symbols vector $\boldsymbol{y}$, distorted from the channel, corrupted by the noise and that passes through a DFT block. Since $\mathbf{G}$ is a normal matrix, i.e., $\mathbf{G}\mathbf{G}^{\mathsf{H}} = \mathbf{G}^{\mathsf{H}}\mathbf{G}$, it has the following eigenvalue decomposition:

$$\mathbf{G} = \mathbf{F}^{\mathsf{H}}\mathbf{\Lambda}\mathbf{F}. \tag{2.20}$$

It has been shown [3] that the eigenvectors are the row vectors of $\mathbf{F}$ and the eigenvalues are arranged in the diagonal $N \times N$ matrix $\mathbf{\Lambda}$. Considering then that $\boldsymbol{v} = \mathbf{F}\boldsymbol{w}$, and that $\mathbf{F}\mathbf{F}^{\mathsf{H}} = \mathbf{F}^{\mathsf{H}}\mathbf{F} = \mathbf{I}$, to obtain the frequency response a Fourier transform is performed, multiplying by $\mathbf{F}$ both sides of (2.19):

$$\mathbf{F}\boldsymbol{d} = \mathbf{F}\mathbf{F}^{\mathsf{H}}\mathbf{\Lambda}\mathbf{F}\boldsymbol{c} + \mathbf{F}\boldsymbol{w}. \tag{2.21}$$

Since (2.10) holds and $\mathbf{G}$ can be written as pointed in (2.20):, the final expression for $\boldsymbol{y}$ is:

$$\boldsymbol{y} = \mathbf{\Lambda}\boldsymbol{x} + \boldsymbol{v}, \tag{2.22}$$

that states the channel frequency response, where each eigenvalue corresponds to each orthogonal subchannel attenuation, due to the correspondence with the respective eigenvector that is a row of the Fourier matrix (2.9). Such relationship in frequency holds because, looking at the transceiver structure, due to the IDFT/DFT blocks, both $\boldsymbol{y}$ and $\boldsymbol{x}$ samples, are representative of the frequency domain. Until now however, a discrete-time system has been considered, but in reality, the channel is analog and so the signals that pass over it; $x_{cp}(t)$ is indeed pulse shaped by $\psi(t - nT)$. At the receiver, to obtain $y(t)$ and then its sampled version $y[k]$, the received signal is convolved with the filter, matched with the pulse, $\psi_{mf}(t)$ and A/D converted. Finally, such system equation is given by:

$$\boldsymbol{y} = \mathbf{H}\boldsymbol{x} + \boldsymbol{v}, \tag{2.23}$$

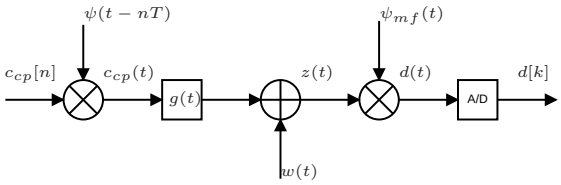

FIGURE 2.6: Pulse shaping.

where $\mathbf{H}$ is a diagonal matrix of eigenvalues like $\mathbf{\Lambda}$ from (2.22), that ensures the orthogonality between the subchannels, otherwise ICI would occur. The difference is that its eigenvalues take into account also the product between the frequency responses, of channel and the shaping pulse autocorrelation function.

Now that the signal model is described, finally it's interesting to see what is the advantage regarding the equalization in an OFDM system: (2.23) states that to obtain the specific received symbol y_n, it's enough to multiply the correspondent transmitted symbol x_n by the transfer function on the nth subcarrier, i.e., the nth diagonal element of $\mathbf{H}$ and add the noise:

$$y_n = H_{n,n}x_n + v_n, \quad 0 \leq n \leq N-1. \tag{2.24}$$

The purpose of the equalization, is to remove the effect of the channel, after its estimation, to obtain the transmitted symbol. Normally it's performed on the whole bandwidth, but since the OFDM subchannels are assumed to be independent, they can be treated separately, each with its flat transfer function. Thus, from (2.23), and considering that (2.24) holds, the estimate $\tilde{x_n}$ is obtained by:

$$\tilde{x_n} = \frac{H_{n,n}x_n}{\hat{H}_{n,n}} + \frac{v_n}{\hat{H}_{n,n}}, \quad 0 \leq n \leq N-1, \tag{2.25}$$

where $\hat{H}_{n,n}$ is the channel estimate indeed, that is the target on which the thesis is focused on and that will be investigated starting from chapter 4. Once that the signal has been equalized, the remaining unknown variable is the noise $\boldsymbol{v}$: the higher is the noise, the worse will the estimate (2.25) be. Hence, the performance of a good receiver, in terms of *Bit Error Rate* (BER), should depend only on the SNR and not on the transfer function, whose estimation should lead to a reconstruction of the transmitted symbols $\boldsymbol{x}$ corrupted by the noise. This is shown in (2.25) when $\hat{H}_{n,n} = H_{n,n}$. To end up, the system equation (2.23) can be rearranged as:

$$\boldsymbol{y} = \mathbf{X}\boldsymbol{h} + \boldsymbol{v}, \tag{2.26}$$

to follow the notation used in papers and works related to OFDM, where $\mathbf{X}$ is a $N \times N$ diagonal matrix containing the transmitted complex symbols and $\boldsymbol{h}$ is a $N \times 1$ column vector containing the diagonal elements of $\mathbf{H}$, i.e., the attenuation factors on each subchannel.

2.3 OFDM Downlink LTE Setup

This model has been introduced here as the LTE physical layer, so the basic parameters have to specified following the 3GPP technical specifications [7] and [2] with the aid of [1]. First of all there's an important difference among the formerly described OFDM system, that is the subcarriers frequency shift: they are distributed from $-N/2$ to $N/2-1$ with the *reference frequency* equal to 0, instead of from 0 to $N-1$. This means that the DFT summation indices have to be changed:

$$x[i] = \frac{1}{\sqrt{N}} \sum_{n=-\frac{N}{2}}^{\frac{N}{2}+1} c[n + \frac{N}{2}] e^{-j2\pi n \frac{i}{N}}, \quad 0 \leq i \leq N-1, \tag{2.27}$$

where if a variable change $m = n + \frac{N}{2}$ is made, it can be written as:

$$x[i] = \frac{1}{\sqrt{N}} \sum_{m=0}^{N-1} c[m] e^{-j2\pi m \frac{i}{N}} e^{j\pi i}, \quad 0 \leq i \leq N-1, \tag{2.28}$$

that is nothing else that a DFT of $x[m]$ multiplied by a phase rotation term $e^{j\pi i}$. Talking about signal model this is straightforward to implement by rotating the rows of the

Fourier matrix as follows:

$$\mathbf{F} = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & \omega^{-N/2} & \omega^{-2N/2} & \dots & \omega^{-(N-1)N/2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{-1} & \omega^{-2} & \dots & \omega^{-(N-1)} \\ 1 & 1 & 1 & \dots & 1 \\ 1 & \omega^{1} & \omega^{2} & \dots & \omega^{(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N/2-1} & \omega^{2(N/2-1)} & \dots & \omega^{(N-1)(N/2-1)} \end{bmatrix} \tag{2.29}$$

thus the same notation of section 2.2 holds. As it may be supposed from the previous explanations, the modulation access domain has both the time and the frequency domain. It can be seen as a time-frequency grid, where the time corresponds to the OFDM symbols while the frequency to the subcarriers, as shown in figure 2.7 (the pointed out resources pattern isn't casual, but it will be explained afterwards). For instance a receiver synchronized on a certain frequency f_n at a certain time t_i, will receive only the data corresponding to one of the squares of figure 2.7, identified by the frequency f_n and the time t_i. Of course this is not the practical case because the resources are assigned in blocks, but it should enlighten the concept.

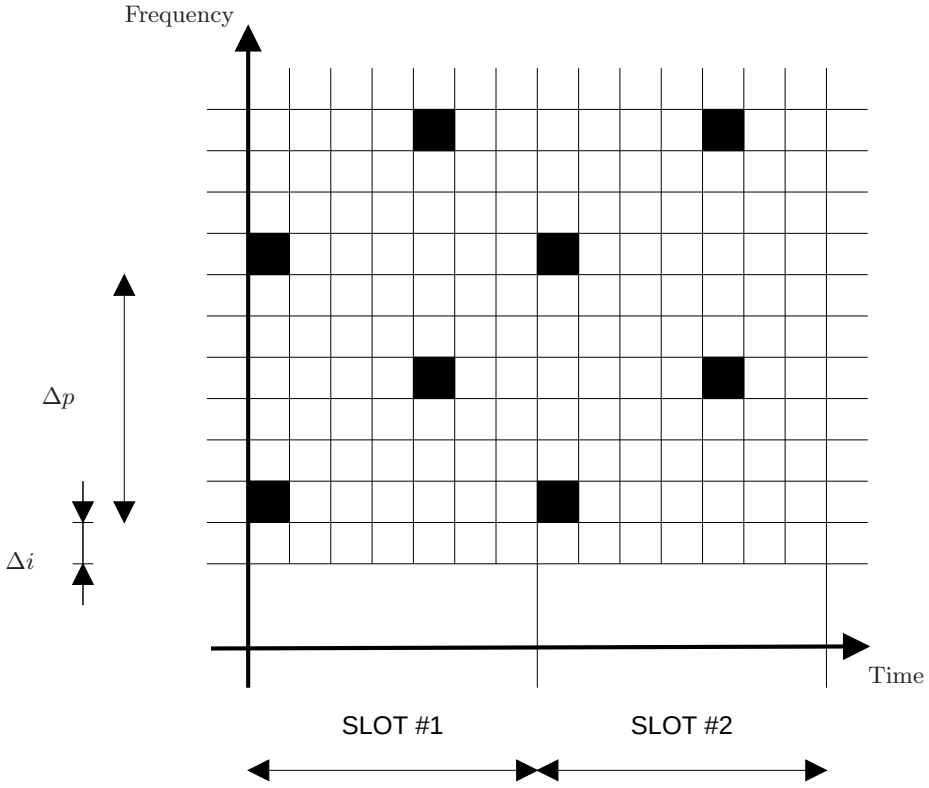

FIGURE 2.7: OFDM time-frequency grid.

In OFDM, the use of each element of the grid, i.e., a *resource element* identified by a

time-frequency index pair, is not necessarily the same: some of them called *pilots*, are used, to estimate the channel by known data. Over the other resources of course, there are the information bits mapped onto complex symbols.

The time-domain structure is organized in *radio frames* of two different types: *Type I*, that is the one that will be considered in this thesis and *Type II*, realized for the co-existence with the systems that use the current 3GPP different standard access. Type I frame structure is illustrated in figure 2.8 and it's organized in ten equally sized *subrfames*; then for the downlink transmission scheme each subframe is made up of two *slots*, where each of them contains seven OFDM symbols with their respective cyclic prefix. Finally for each frame there will be $7 \times 2 \times 10 = 140$ OFDM symbols.

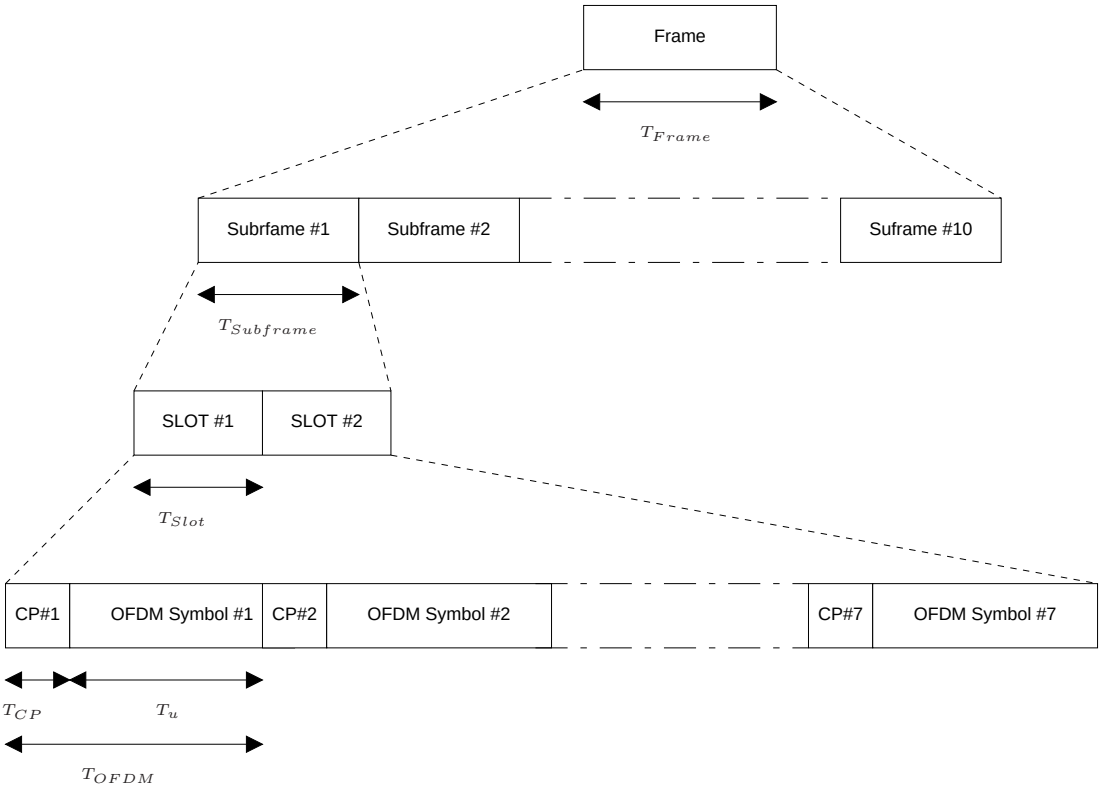

FIGURE 2.8: LTE frame structure.

2.3.1 Parameters Specifications

After a first survey to the physical layer structure it's necessary to specify some parameters relative to the OFDM modulation and to the time-domain structure, recalling some relation between them: T_S is the base time unit, i.e., the sampling time of the FFT-based transceiver with $N = 2048$, related to the original symbol duration T_u as $T_u = NT_S = N/B$, where B is the system bandwidth. It can also be derived by the

subcarriers spacing as $T_u = 1/\Delta f$ and added up with the cyclic prefix time T_{CP} corresponds to the overall OFDM symbol time $T_{OFDM} = T_{CP} + T_u \approx T_{Slot}/7$. Here, $\approx$ means that time slot is not divisible by seven, so the cyclic prefix of the first OFDM symbol has to be slightly longer. Here follows the list of such parameters:

$T_{Frame} = 10$ ms	$T_{OFDM} = 71.4\ \mu$ s	$\Delta f = 15$ kHz
$T_{Subframe} = 1$ ms	$T_u = 66.7\ \mu$ s	$N = 2048$
$T_{Slot} = 0.5$ ms	$T_{CP} = 4.7\ \mu$ s	$B = 30.72$ MHz
		$T_S = 32.55$ ns
		$N_u = 1200$

Table 2.1: LTE Downlink OFDM Parameters

Specifying that the cyclic prefix length is of 144 samples and noticing that not all the subcarriers are active even if the bandwidth is calculated on $N = 2048$. This is due to the shaping filter: physically a filter whose frequency response is a rectangular windows (it should be equal to 30.72 MHz) is not realizable, so to avoid the cut off of the border subcarriers they are kept turned off and when the tails of the composite signal superimpose due to the periodic repetition, there's no loss of information; such unused subcarriers are often referred as *guard bands*, exactly for their purpose. Furthermore, is also possible modify the effective bandwidth, that in the just mentioned case is $B = 18$ MHz, to fit different specifications [8] shown in table 2.2, by turning off more subcarriers and in any case with the center DC-subcarrier is always shut down.

Channel bandwidth [MHz]	1.4	3	5	10	15	20
Used subcarriers	72	180	300	600	900	1200

Table 2.2: LTE Bandwidth Configurations

2.3.2 Pilot Pattern

The pilot distribution showed in figure 2.7 is the one that is used in LTE, i.e., the pattern that comes from the 3GPP specifications [2] often referred as *reference signal sequence*. The pilot spacing in frequency in terms of number of subcarriers is $\Delta p = 6$ for each OFDM symbol that carries pilots and the initial indent is $\Delta_i = 1$, then a subcarriers shift exists, between the first and the second reference symbol in each slot, of half the pilot spacing, i.e., 3; the number of pilots per OFDM symbol is then 200, considering 1200 used subcarriers. Such positioning can be outlined by an indexes sequence to identify the subcarrier number (in the range from $-N_u/2$ to $N_u/2$), whose argument

corresponds to the pilot number:

$$p(m) = [(\eta + \Delta_i) \text{mod } 6] + 6(m-1) - \frac{N_u}{2} + 1, \quad 1 \leq m \leq M, \tag{2.30}$$

where:

$$\eta = \begin{cases} 0, & \text{at the first OFDM symbol in any slot} \\ 3, & \text{at the fifth OFDM symbol in any slot} \end{cases}. \tag{2.31}$$

For instance then, the fourth pilot on the first reference OFDM symbol will be at the -580th subcarrier, the fifth at the -574th and so on. These indexes will be useful afterwards to define all the signal and functions that require the pilot positioning within the N subcarriers.

This indexing yields a diamond pattern that seems to be the best choice for LTE purposes [2], considering in fact that for a good estimation should be taken into account the channel behavior both in time and in frequency: two extreme cases could be when the channel would be very selective in frequency/time-invariant and very time-variant/flat in frequency, and they could be treated respectively using all pilots in frequency once and with just one pilot but for each OFDM symbol. Of course these two cases are to emphasize the problem but they should clarify the meaning of such pattern.

In figure 2.9 it's depicted how this time-frequency access works in a MIMO environment, that is one of the innovation introduced by LTE to reach higher data rates. In this case with two antennas, the pilots corresponding to the other antenna's pilots have to be turned off (symbols in gray) in order to avoid interference.

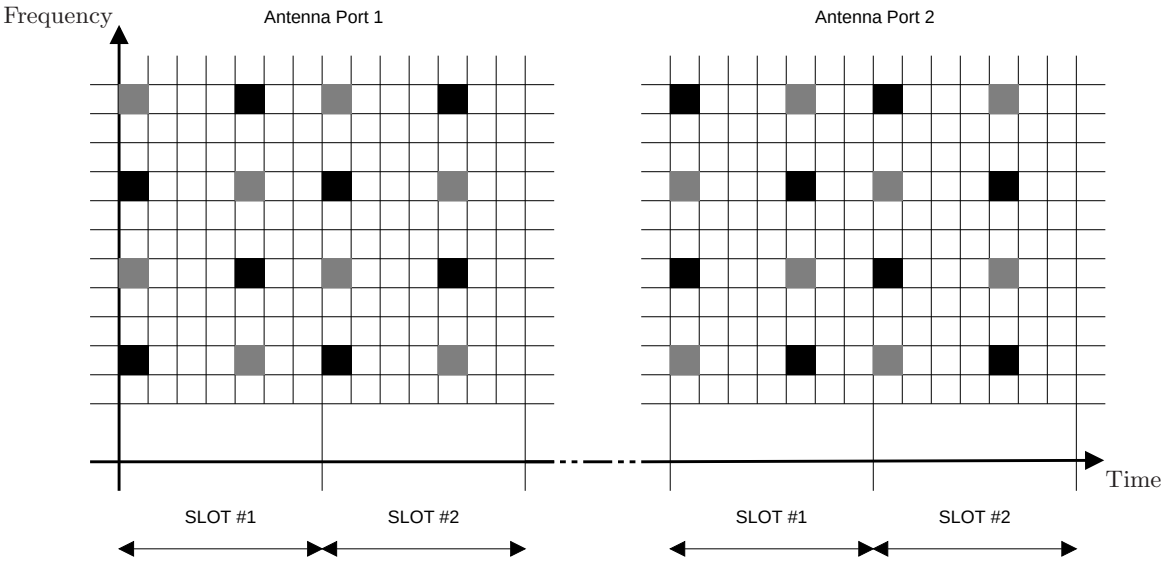

FIGURE 2.9: MIMO OFDM time-frequency grid.

In chapter 4, it will be shown the two main types of channel estimation techniques, both based on the pilot observations. The first interpolates their frequency responses over whole the bandwidth, while the second performs a tap-delay estimation through an eigenstructure applied to a sample covariance matrix. The scope of the thesis however, doesn't include the interpolation over time of the estimated frequency response. It's focused indeed, just on an individual OFDM symbol.

Chapter 3

Multipath Wireless Channel

Since the LTE radio access technology operates in the mobile communications, it's mandatory to describe the medium whereby the transmission occurs. LTE assumes some reference channel models [9], but it seems that they could be improved due to the poor pertinence with the real case, considering the typical mobile terminals environments. Behind every model indeed, there is a physical behavior that justifies it, but a general characterization of the multipath channel can be given. This through its impulse and frequency response, followed from significant physical and statistical properties, which will be different according to the scenario, e.g., pedestrian or vehicular. Generically, each transmission that doesn't occur in open space suffers the effect of multipath propagation due to the signal interactions with the obstacles (*scatterers*) between the fixed transmitter and the receiver, that talking about downlink, is usually moving. Furthermore, the scatterers could be moving as well, so in general the number, the attenuation and the delay of each multipath component are not easily predictable, and above all, time-varying. Here, two main types of channel families are outlined: the static and the dynamic. The first is easier to model but it's quite unreal; this is not difficult to see if a moving receiver is considered, due to the fact that it assumes fixed both the number of delays and their parameters. Conversely ,the second is more realistic but arises some problems when an estimation has to be preformed. The general (time-varying) model used to describe the lowpass *impulse response* (hence when not specified, the equivalent lowpass representation will be considered) is the following:

$$g(\tau, t) = \sum_{i=1}^{L(t)} \alpha_l(t)\delta(\tau - \tau_l(t)), \tag{3.1}$$

where the parameters variations, change differently in time according to the scenario. The just mentioned parameters are the number of delays $L(t)$, the complex attenuation factors $\alpha_l(t)$ and the propagation delays $\tau_l(t)$. The attenuation factors can be expressed

in terms of magnitude $a_l(t)$ and phase $\varphi_l(t) = 2\pi f_c \tau_l(t)$, as $\alpha_l(t) = a_l(t)e^{-j\varphi_l(t)}$ (where f_c is the carrier frequency). Now the model can assume them as unknown to proceed with a statistical characterization or rather can try to estimate them; this is the difference between a *nonparametric* and *parametric* channel model. This distinction has been shown [10] to yield different performances within the channel estimation, precisely in the *frequency response* reconstruction. The frequency response, often referred as *transfer function*, is another (time-varying) channel characterization, obtained simply by the Fourier transform of the impulse response:

$$h(f,t) = \int_{-\infty}^{\infty} g(\tau,t)e^{-j2\pi f\tau}d\tau, \tag{3.2}$$

but considering the discrete character attributed to the impulse response it becomes:

$$h(f,t) = \sum_{i=1}^{L(t)} \alpha_l(t)e^{-j2\pi f\tau_l(t)}. \tag{3.3}$$

This attribution is acceptable, even though seeing the impulse response as a train of pulses is quite an ideal case. This because the scatterers and then the interactions are not ideal at all, thus a continuous model would be better. However, this would complicate the treatment, while a sufficient accuracy is already achieved. Equation (3.3) together with (3.1), are the key to describe the correlation functions in both time and frequency, in order to define the statistical channel properties. Here follows a graphical example, illustrated to clarify the difference between channel models.

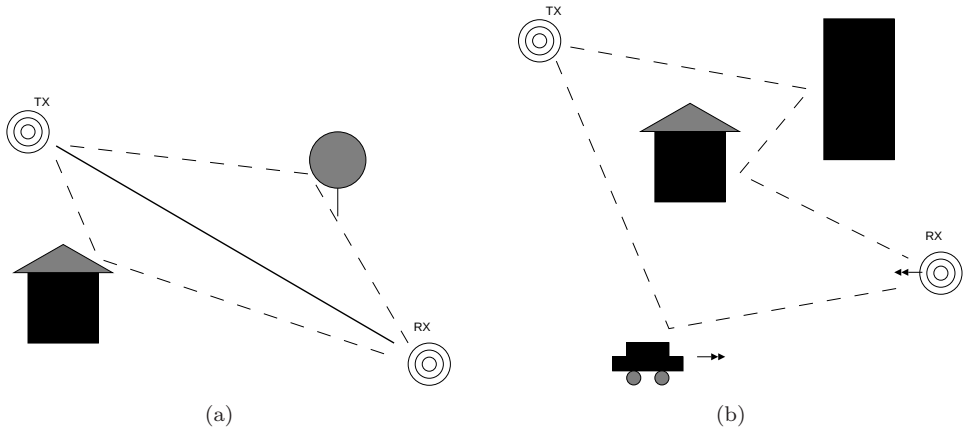

FIGURE 3.1: LOS static channel (a), and NLOS dynamic channel (b).

Figure 3.1 shows two different types of physical channels, i.e., a static (at least a good approximation can by static) *line of sight* (LOS) and a dynamic *Non Line Of Sight* (NLOS) channel. The first has the transmitter, the receiver and the scatterers fixed, and a dominant component, while the second only provides the transmitter fixed and the receiver and the scatterers moving, without any dominant component, which is a

more realistic mobile scenario. Consider for instance, a communication between a base station and a mobile phone whose user is walking along a busy street; it's evident that the multipath pattern is time-varying. The first model, is more a rough interpretation of the real channel behavior, because observing the circumstances where mobile communications take place, a fixed LOS component is quite rare, as well as the fixed scatterers and receiver. Despite that, the second model usually is not employed (LTE proposes three static delay profiles [9]), and it's why an investigation on its behavior is performed by simulations, compared with the first, whereto most of the estimation algorithms are tested.

3.1 Time Spread and Fading

Now if the transmission of a single short pulse $c(t)$ is considered, neglecting the noise, is straightforward to see by the convolution with the impulse response, that the received signal will be a train of pulses, each with different delay and complex amplitude, at a certain time t:

$$r(\tau, t) = \sum_{i=1}^{L(t)} \alpha_l(t) c(\tau - \tau_l(t)). \tag{3.4}$$

The delays are usually ordered in an increasing way so as, being $\tau_1(t) \leq \tau_2(t) \leq \cdots \leq \tau_L(t)$, the time-varying difference between the greatest and the smallest can be highlighted through the *maximum excess delay* $\tau_M(t) = \tau_L(t) - \tau_1(t)$, that is a rough but immediate to obtain channel spread indicator, used for instance to determine the spacing in frequency between the pilot subcarriers in an OFDM transmission. These multipath components can be resolved or not, depending on the receiver, but only under the particular condition that the time difference between adjacent delays, was much greater than the inverse of the signal bandwidth:

$$\tau_{l+1} - \tau_l \gg \frac{1}{B}, \tag{3.5}$$

thus a system with a large bandwidth will have a better resolution in the delay domain than a system with a smaller bandwidth. It's why such systems suffer differently the multipath effect, even though, if the multipath components are summed at the receiver, the resulting effect is an interference. Since the attenuations are complex, the sum can be either constructive or destructive, thus, if the receiver is moving, i.e., the impulse response is varying, the received signal amplitude will present positive and negative fluctuations over time. This phenomenon is called *fading* and it can be seen from a *large scale* and a *small scale*, but it's not interest of this thesis to examine it, despite in a real implementation it has to be taken into account.

Here, is better to see the problem from another point of view, the spreading in time induced by the multipath non-ideal impulse response. With the bandwidth increasing, the symbol time decreases, thus the problem is the interference between subsequent pulses, if the channel spreading is greater than the transmission time between them (the first echo of the second signal could arrive before the last one of the previous signal). This problem arises in each transmission system, where pulses are modulated by symbols carrying information at a certain rate, that is limited indeed, and it's called *Inter-Symbol Interference* (ISI): due to the delay echoes, the spread modulated pulses could overlap if the symbol rate is too high, i.e., if its inverse is small compared with the spreading factor (e.g., the maximum excess delay). Fortunately, even though the channel is always different, for similar scenarios similar characteristics are present, e.g., the number of delays, the maximum impulse response length, etc.; through this, is possible to remove ISI using for instance an OFDM modulation, choosing parameters like subcarriers bandwidth and cyclic prefix length, properly, as explained in chapter 2.

3.2 Statistical Characterization and Properties

After a first glance to the multipath behavior, it should be clear that the complex attenuation factors $\alpha_l(t)$, reported for clarity:

$$\alpha_l(t) = a_l(t)e^{-j2\pi f_c \tau_l(t)}, \tag{3.6}$$

associated with each delay can be modeled as a random processes in the variable t, due to the fact that the phase $\varphi_l(t) = 2\pi f_c \tau_l(t)$ change in a fast manner even though the movement is very small, so as it can't be predicted in a deterministic way. The time dependence comes of course from the fact that each movement at a certain speed, is associated with the respective value of the variable t, and it's why the properties of a pedestrian channel are different from those of a vehicular channel. Since the received signal $r(\tau, t)$ is obtained as the sum of independent and identically distributed random processes, from the *central limit theorem*, if the number of paths is large enough, the resultant random process, i.e., $r(\tau, t)$ can be approximated through a complex values Gaussian distribution. In addition considering a NLOS scenario, each signal process can be considered zero-mean, then the whole received signal as well; this type of channel is said *Rayleigh fading channel* because of its envelope distribution. Moreover is reasonable thinking that the random processes associated with different paths were uncorrelated and that the channel statistic depends only from the relative time between two observations and not from the specific observation times. These assumptions can be resumed with

the statement: *Wide Sense Stationary Uncorrelated Scattering* (WSSUS) and will lead to several important properties.

The term channel statistic include both the autocorrelation functions either in time and frequency; to obtain the first one is enough to correlate the impulse response, with its delayed version:

$$R_{gg}(\tau_1, \tau_2, \Delta t) = \frac{1}{2} E\left[g^*(\tau_1, t) g(\tau_2, t + \Delta t)\right]. \tag{3.7}$$

It shows how the WSS assumption has been considered, writing the equation as a function of the time lag Δt instead of two different time instants. The US assumption on the other hand can be taken into account by the δ function as:

$$R_{gg}(\tau_1, \tau_2, \Delta t) = R_{gg}(\tau_1, \Delta t)\delta(\tau_1 - \tau_2) = R_{gg}(\tau, \Delta t), \tag{3.8}$$

that means that the autocorrelation function R_{gg} is nonzero only if $\tau_1 = \tau_2$, i.e., each echo is correlated only with itself, it depends then, only from the delay and the time lag. An important function that will be useful to define the next important property, that derives from R_{gg} when $\Delta t = 0$, is the *power delay profile*, i.e., $R_{gg}(\tau, 0)$. It's simply defined as $R_{gg}(\tau)$ and state the channel output power corresponding to a certain delay (figure 3.2).

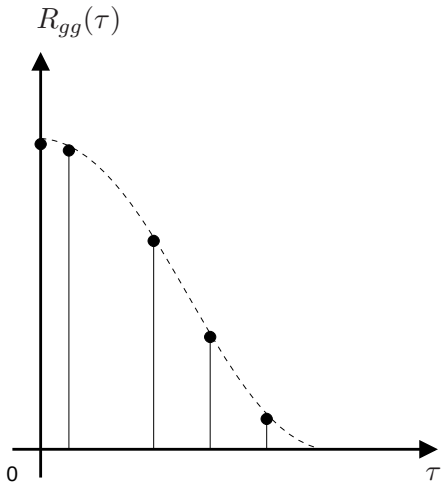

FIGURE 3.2: Power delay profile.

Its generic shape foresees a peak for $\tau = 0$, i.e., when the impulse response is correlated with a non-delayed version of itself, and a decay towards zero after a certain delay. This time after which the power delay profile is zero, or under a fixed threshold, is named *channel delay spread*, written as T_m and it gives a measurement of the channel time distortion. If a modulated pulse is transmitted with an initial shape, after the transmission it will have another shape depending on the impulse response, leading so, to spreading in time, that is the relevant problem. Then directly from by the delay

spread definition, arise the concept of *selectivity in frequency*: different frequencies will yield a different channel response if their displacement is greater than the *coherence bandwidth*. Its definition derives from the frequency correlation function (3.10) but it's simply the reciprocal of the delay spread:

$$B_c = \frac{1}{T_m}. \tag{3.9}$$

Hence, a signal with a bandwidth greater than the coherence bandwidth will be distorted by the channel, making its equalization difficult. It's intuitive how things would go better if the attenuation would be flat over the whole bandwidth. To define the analogous in the time domain, is useful to introduce the just mentioned correlation in frequency by the frequency response autocorrelation function R_{hh}, that can be obtained directly through the Fourier transform of the impulse response autocorrelation function $R_{gg}(\tau, \Delta t)$:

$$R_{hh}(\Delta f, \Delta t) = \int_{-\infty}^{\infty} R_{gg}(\tau, \Delta t) e^{-j2\pi\Delta f\tau} d\tau, \tag{3.10}$$

or alternatively by its definition, correlating the frequency response $h(f, t)$ with it's delayed version shifted in frequency:

$$R_{hh}(f_1, f_2, \Delta t) = \frac{1}{2} E\left[h^*(f_1, t) h(f_2, t + \Delta t)\right]. \tag{3.11}$$

Before introducing the next property, it's useful to recall a phenomenon that occurs when a signal impinges on a moving receiver, the *Doppler effect*. It consists in an apparent frequency shift caused by a phase variation induced by the motion, whose geometrical problem is illustrated in figure 3.3, i.e., a receiver that's moving from A to B at the speed v, while is receiving from S at the wavelength λ.

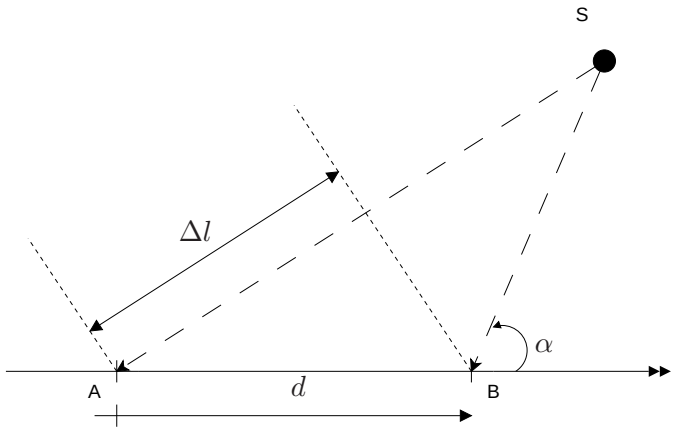

FIGURE 3.3: Doppler effect.

After some calculations, the Doppler frequency shift that results, is given by:

$$f_d = \frac{v}{\lambda} \cos \alpha, \tag{3.12}$$

that of course will be different for each multipath component, being the angles of arrival α different each other. Now it's interesting notice how the Doppler effect can be related together with the time-varying nature of the channel by the function $S_{hh}(\Delta f, \lambda)$, obtained Fourier transforming R_{hh}:

$$S_{hh}(\Delta f, \lambda) = \int_{-\infty}^{\infty} R_{hh}(\Delta f, \Delta t) e^{-j2\pi\lambda\Delta t} d\Delta t, \tag{3.13}$$

that for $\Delta f = 0$ is the so-called channel *Doppler power spectrum*, $S_{hh}(\lambda)$. Like it occurs for the delay spread in the power delay profile, there will be a band, over that such spectrum will decay to zero or under a fixed threshold. This band B_d, called *Doppler spread*, state that the received signal intensity is zero (or very low) after a certain shift in frequency due to the Doppler effect. The straightway parameter that derives from the Doppler effect is the *coherence time* T_c defined as its inverse:

$$T_c = \frac{1}{B_d}, \tag{3.14}$$

that assumes the dual meaning of the coherence bandwidth, i.e., if a signal arrive after a time interval, larger than T_c from a previous signal, they can be considered uncorrelated. By the definition of S_{hh}, that is a Fourier transform, can be deduced that more the channel is slowly time-varying, more the Doppler spread is small, then the coherence time is larger. For instance, if the channel was time-invarying, with a Doppler spread equal to zero, the coherence time would be infinite.

3.2.1 Delay Profiles

Talking about LTE, the 3GPP technical specification [9] suggests three kinds of channels with different power delay profiles, to characterize different environments. A different spreading factor occurs in different environments indeed, the parameters used to measure it are the delay spread and the maximum excess delay. They are significantly different for each different scenario, that is representative of a particular delay spread environment, i.e., low, medium and high. These are named respectively the *Extended Pedestrian A*, the *Extended Vehicular A* and the *Extended Typical Urban* and their parameters are reported in table 3.1:

Model	Number of channel taps	Delay spread [ns]	Maximum excess tap delay [ns]
EPA	7	45	410
EVA	9	357	2510
ETU	9	991	5000

TABLE 3.1: LTE Delay Profiles

These models were introduced in the specifications, combined with a set of various parameters, to simulate the performance of an LTE receiver in different environments.

3.3 OFDM Transmission Over Multipath Channel

In chapter 2 an OFDM system has been described, assuming as known the multipath channel theory that stays behind it. In this chapter a meaning is provided to the declared equations and parameters, like the impulse response $g(\tau, t)$ and the delay spread T_m. The last step is to relate the two chapters clarifying how an OFDM modulation sees a multipath channel, e.g., through its transfer function and why this scheme avoids some problems that has been presented as crucial in this environment. The first relationship is between the matrix (2.17):

$$\mathbf{G} = \begin{bmatrix} g_0 & & & g_\mu & \cdots & g_1 \\ g_1 & g_0 & \mathrm{O} & & \ddots & \vdots \\ \vdots & & \ddots & & & g_\mu \\ g_\mu & & & g_0 & \mathrm{O} & \\ & \ddots & & & \ddots & \\ \mathrm{O} & & g_\mu & \cdots & g_1 & g_0 \end{bmatrix}, \tag{3.15}$$

and the impulse response 3.1:

$$g(\tau, t) = \sum_{i=1}^{L(t)} \alpha_l(t)\delta(t - \tau_l(t)), \tag{3.16}$$

reported both in order to better relate them. The channel matrix assumes $g(\tau, t)$ discrete, i.e., $g[n]$ with $0 \leq n \leq \mu$, due to the sample alignement condition and time-invariant. Thus, removing the time dependence from $L(t)$ and assuming $L = \mu$, where μ is the cyclic prefix length, is evident how the matrix coefficients correspond to the impulse response, that results a train of pulses multiplied by complex attenuation factors; regarding its shape, it's because it fits the convolution between the signal and the impulse response. The purpose of the cyclic prefix now should be evident: for instance if the signal length, in number of samples is M and the channel impulse response one is L, their convolution

will have a maximum length of $M + L - 1$ [5]. It yields a spreading in time not greater than L, then any guard interval that has at least the same length L, added at the beginning of each symbol, will avoid the inter-symbol-interference. Follows then the OFDM transfer function $h(f, t)$, that can be decomposed in N sub-transfer functions, each one corresponding to the attenuation on each subchannel. Before showing it a consideration has to be made: since the model assumes the delays time-invariant and sample aligned another substitution has to be performed in (3.16), i.e., $\tau_l(t)$ becomes $\tau_l T_s$, where T_s is the system sampling time (see subsection 2.3.1), and τ_l is expressed in number of samples. The transfer function (3.3) becomes then:

$$h(f,t) = \sum_{i=1}^{L} \alpha_l(t) e^{-j2\pi f \tau_l T_s}, \tag{3.17}$$

which yields N components $h_n(t)$ by substituting $n\Delta f$ (see subsection 2.3.1) to f, in order to identify the nth subchannel, reminding that $\Delta f Ts = 1/N$:

$$h_n = \sum_{i=1}^{L} \alpha_l(t) e^{-j2\pi \frac{n}{N} \tau_l}. \tag{3.18}$$

In a pilot-assisted channel estimation the index n corresponds only to the pilots positions, because they are the only coefficients that have to be estimated by known data, to equalize the received signal, while in general $0 \leq n \leq N - 1$.

Finally the motivation of the pilot spacing in both frequency and time is given by [10], that relates the displacement to the channel correlation properties. Has been shown that the time distortion can be measured by the maximum excess delay or the delay spread, while the spectrum broadening by the Doppler spread. To define the pilot spacing in frequency D_f, the normalized maximum excess delay has been exploited:

$$D_f \leq \frac{NT_s}{\tau_M}, \tag{3.19}$$

that states the maximum separation in frequency between adjacent pilots. Again, the maximum separation in time D_t, in term of number of OFDM symbols is calculated on their duration and on the maximum Doppler frequency f_D:

$$D_t \leq \frac{1}{2 f_D T_{OFDM}}. \tag{3.20}$$

Of course they both have to be rounded to the smallest integer and by substituting the LTE specific values in the (3.19) and (3.20), an explanation to the pilots pattern introduced in subsection 2.3.2 is given: considering $N = N_u = 1200$ and $\tau_M = 5000ns$ from the ETU delay profile defined in [9], follows that $D_f \leq 7$ and with a maximum

Doppler frequency of 300 Hz, that $D_t \leq 2$, that almost fits with the proposed diamond pattern, since the time spacing is greater. It's of course a compromise between the channel estimation and a spectral and power efficiency.

Now that the channel of interest has been described in terms of impulse response, transfer functions and properties, useful to the preceding and the following arguments, a chapter on its estimation follows. This suits with the thesis purposes that wants to investigate how it could be possible to improve such estimate, by the knowledge of the channel parameters, that contrariwise are not considered by the curent state-of-the-art algorithm.

Chapter 4

Channel Estimation

After the introduction in the previous chapters, either on the multipath channel and the OFDM signal model, here follows the description of the state-of-the-art estimation algorithm, in order to emphasize its drawbacks. Then the candidate tap-delay estimation scheme ESPRIT, proposed to supply them is shown in the second section. An investigation on a better channel estimator in terms of performance/complexity and considering a dynamic channel instead of a static one, would be the target of the entire work, which this thesis wants to be a contribute. Two different approaches will be described, both based on the frequency domain (pilots) observation, but not considering the interpolation over time. The first tries to reconstruct directly the channel frequency response aided by the pilot observations. The second, performs a tap-delay estimation providing a parametric channel model (see [10]), but assuming the attenuation factors as random processes. Then a further algorithm, fed by such parameters, is necessary to estimate the transfer function. Both the techniques base their estimation on known data, from either the transmitter and the receiver, placing over all the subcarriers or just over some of them, which is the case used in practice to reduce the overhead, i.e., *Pilot Symbol Assisted Modulation* (PSAM). The case of all pilots anyway is often used to simplify the treatment and to adapt this afterwards. The first approach for instance, to produce the estimate also on the subchannels that don't carry pilots, has to perform an interpolation over the whole bandwidth, knowing only the pilots estimates (see later).
Regarding the frequency response estimation an *Linear Minimum Mean Square Error* (LMMSE) algorithm, rather known as *Wiener Filter* (and its robust approach), followed by a reduced complexity example, will be presented, while for the tap-delay estimation, the array signal processing theory will be introduced, to exploit the ESPRIT algorithm, that was initially introduced [11] for the *Direction Of Arrival* (DOA) estimation problem and that has a framework similar to that proposed for the delay estimation problem. Although this thesis is focused on the second approach, in detail on the number of delays

estimation, it's seemed proper to mention also the algorithm, that is the state-of-the-art channel estimator for the actual LTE applications.

4.1 Channel Frequency Response Estimation

This approach is introduced as first because its behavior is intuitably easier to understand: recalling well known relationship 2. between the received signal and the transfer function:

$$\boldsymbol{y} = \mathbf{X}\boldsymbol{h} + \boldsymbol{v}, \tag{4.1}$$

the most straightforward manner to estimate $\boldsymbol{h}$ would be to transmit known data $\mathbf{X}$, so receiving the signal $\boldsymbol{y}$, the only unknown variable would be the noise $\boldsymbol{v}$ that anyway has its statistic characterization. It's also evident that transmitting redundant data over all the subcarriers would be a waste of resources, even though only during certain OFDM symbols. Therefore it's reasonable thinking to transmit them only over some specific fixed subcarriers, that means a worst estimation but a better efficiency. Of course, missing some subcarriers information, they have to be reconstructed by an interpolation (figure 4.1) over whole the bandwidth, to recover the overall transfer function.

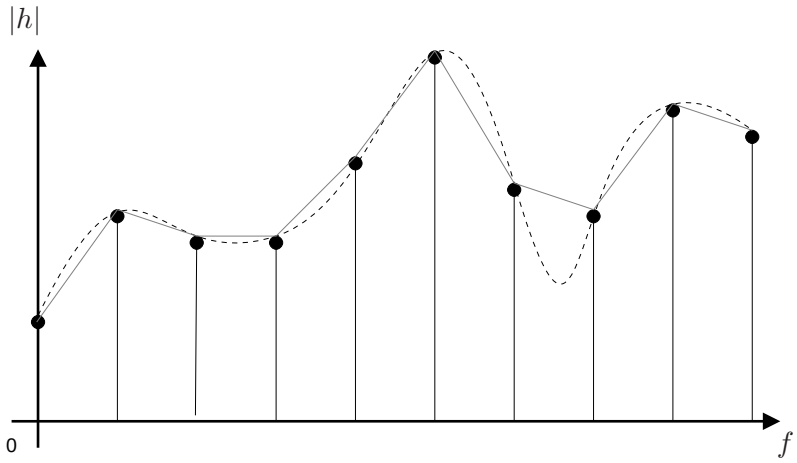

FIGURE 4.1: Interpolated channel transfer function.

Figure 4.1 shows the well known proceeding of a general interpolation: the dashed line is the real function and the dotted stems are the estimated value corresponding only to certain frequencies; a linear interpolation wouldn't give a good approximation, thus an LMMSE estimator is proposed as one of the state-of-the-art algorithms.

4.1.1 Pilot-Aided Channel Estimation

As explained in section 2.3, some resource elements carry known data to estimate the channel frequency response at certain subchannels during certain OFDM symbols. Although both the approaches exploit the pilots known data to estimate the channel and some of the vectors and matrices used to describe the proceeding will be the same, the estimation frameworks are different. The one named *Pilot-Aided Channel Estimation* (PACE) is presented into this section, followed by the description of one of the state-of-the-art algorithms that rely on it, i.e., the Wiener filter. First of all the subscript p is added to specify that the (4.1) is evaluated only at the subcarriers containing pilots:

$$\boldsymbol{y}_p = \mathbf{X}_p \boldsymbol{h}_p + \boldsymbol{v}_p. \tag{4.2}$$

The vector $\boldsymbol{y}_p$ is a $M \times 1$ (instead on $N \times 1$ like $\boldsymbol{y}$ is) containing only the received symbol at the pilot subcarriers. Accordingly the estimate is in terms of *Least Squares* (LS), often referred as *Zero Forcing* (ZF) estimate (because of the noise power that could be forced to zero increasing the pilots power), that is obtained multiplying $\boldsymbol{y}_p$ by the inverse of $\mathbf{X}_p$ to minimize the distance $||\boldsymbol{y}_p - \mathbf{X}_p \boldsymbol{h}_p||^2$, i.e., $(\boldsymbol{y}_p - \mathbf{X}_p \boldsymbol{h}_p)^{\mathsf{H}}(\boldsymbol{y}_p - \mathbf{X}_p \boldsymbol{h}_p)$. This won't be the final estimate here but only a starting point to apply the algorithm explained in the next section. After, decomposing the function as $\boldsymbol{h}_p = \mathbf{T}\boldsymbol{\alpha}$, where $\mathbf{T}$ is the $M \times L$ *steering matrix* (from the array signal processing terminology), with M and L that are respectively the number of pilots and the number of delays, whose estimation is treated in chapter 6, the least squares estimation of $\boldsymbol{h}_p$ can be written as:

$$\hat{\boldsymbol{h}}_p^{LS} = \mathbf{X}_p^{-1}\boldsymbol{y}_p = \boldsymbol{h}_p + \mathbf{X}_p^{-1}\boldsymbol{v}_p = \mathbf{T}\boldsymbol{\alpha} + \mathbf{X}_p^{-1}\boldsymbol{v}_p, \tag{4.3}$$

with:

$$\mathbf{T}(m, l) = e^{-j2\pi \frac{p(m)}{N} \tau_l} \tag{4.4}$$

and

$$\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \cdots, \alpha_L]^{\mathsf{T}}, \tag{4.5}$$

that is a vector of the propagation paths complex amplitudes. This means that the transfer function $\boldsymbol{h}_p$ is decomposed in a product of a matrix $\mathbf{T}$ related to the delays by a vector $\boldsymbol{\alpha}$ containing the complex amplitudes of the respective delay. Each column $\boldsymbol{t}(\tau_l)$ of $\mathbf{T}(\boldsymbol{\tau})$, is dependent from a different delay, so as the matrix can be written as $\mathbf{T} = [\boldsymbol{t}(\tau_1), \boldsymbol{t}(\tau_2), \cdots, \boldsymbol{t}(\tau_L)]$. Furthermore, another remark has to be made on the vector $\boldsymbol{\alpha}$: it won't be estimated but treated as a random process; the problem reduces than to the estimation of $\boldsymbol{\tau}$, i.e., the vector of delays. Considering from this point until the end

that all pilots hold unitary power, the pilot observations become:

$$\hat{\boldsymbol{h}}_p^{LS} = \mathbf{T}\boldsymbol{\alpha} + \boldsymbol{v}_p. \tag{4.6}$$

So, these will be used as initial estimates on the M subcarriers that are considered as independent channels, to estimate the remaining $N_u - M$ information channels. Thus, follows an interpolation algorithm and its robust approach [12] (the meaning will be clear after the next section).

4.1.2 The Wiener Filter

After the framework introduction it's good to detail the above mentioned technique, i.e., the Wiener filter, first of all recalling how an LMMSE estimator works [13]. Likewise the LS estimator, it's going to find the value of the estimation that minimize a certain metric, that in this case is the *Mean Square Error* (MSE), defined as $MSE[\hat{\boldsymbol{h}}_p] = E\left[(\boldsymbol{h}_p - \hat{\boldsymbol{h}}_p)^{\mathsf{H}}(\boldsymbol{h}_p - \hat{\boldsymbol{h}}_p)\right]$; so the interpolated channel transfer function follows from the derived equation:

$$\hat{\boldsymbol{h}}_{LMMSE} = \mathbf{R}_{h\hat{h}_{LS}}\mathbf{R}^{-1}_{\hat{h}_{LS}\hat{h}_{LS}}\hat{\boldsymbol{h}}_{LS}, \tag{4.7}$$

where $\mathbf{R}_{\hat{h}_{LS}\hat{h}_{LS}}$ is the autocovariance matrix of the channel pilot observations, i.e., the LS estimates of $\boldsymbol{h}_p$, and $\mathbf{R}_{h\hat{h}_{LS}}$ is the crosscovariance matrix between the channel pilot observations and the channel where an observation is not available, i.e., the remaining $N_u - M$ subcarriers. Now, recalling that $\hat{\boldsymbol{h}}_{LS} = \boldsymbol{h}_p + \boldsymbol{v}_p$, 4.7 can be written in the ordinary shape of a *Wiener filter*:

$$\hat{\boldsymbol{h}}_{WF} = \mathbf{R}_{hh_p}(\mathbf{R}_{h_ph_p} + \sigma_v^2\mathbf{I}_N)^{-1}\hat{\boldsymbol{h}}_{LS}, \tag{4.8}$$

where $\mathbf{I}_N$ is the $N \times N$ identity matrix and σ_v^2 the noise variance. Equation (4.8) represents then the best linear MSE approximation of the transfer function $\boldsymbol{h}$. The drawback of this approach could seem that in practice the two matrices $\mathbf{R}_{hh_p}$, $\mathbf{R}_{h_ph_p}$ and the noise variance σ_v^2 are not available, so they should be assumed somehow. Contrariwise, using the approach illustrated afterwards to design the estimator, will make it very *robust* either to variations in the channel statistic and to the SNR mismatch.

4.1.2.1 The Robust Approach

Now that the estimation has been derived, a choice regarding the unknown parameters, i.e., the channel statistic and the SNR, has to be made to perform it. A deep investigation on the mismatches have been made in [12] and integrated in [14], but only the

main results are reported. Starting from the SNR, it has been shown [14] that if its tracking is not affordable, a design for a high SNR is preferable, because intuitably, the estimation error can be confused with the noise at low SNRs, contrariwise is dominant for a high SNR. Regarding the channel statistic, given the covariance matrix definition (4.19) to calculate $\mathbf{R}_{hh} = E\left\{\boldsymbol{h}\boldsymbol{h}^{\mathsf{H}}\right\}$ and reminding (3.18), i.e., the attenuation on the nth subcarrier, omitting the time dependence due to the time-varying channel is given by:

$$h_n = \sum_{l=1}^{L} \alpha_l e^{-j2\pi \frac{n}{N}\tau_l}, \quad 0 \leq n \leq N_u - 1. \tag{4.9}$$

After expliciting the $\boldsymbol{h}$ correlation matrix expression from (4.9), assuming that τ_l are independent, the power delay profile, that usually is a decaying exponential, with its probability density function, has to be statistically characterized. It's shown in [12] that the delays' distribution that minimizes the error is:

$$f_{\tau_l}(\tau_l) = \begin{cases} \frac{1}{\tau_M} & \text{for } 0 \leq \tau_l \leq \tau_M \\ 0 & \text{otherwise} \end{cases}, \quad 1 \leq l \leq L \tag{4.10}$$

i.e., a continue uniform power delay profile distribution along the cyclic prefix, that's assumed to be longer as τ_M. Whereas considering a unitary total average power, the assumption that is made on the complex attenuation factors mean is:

$$E\left[|\alpha_l|^2\right] = \frac{1}{L}, \quad 1 \leq l \leq L. \tag{4.11}$$

In this way the Wiener filter can be applied to the LS estimate on the pilots by calculating the two matrices in (4.8), making use of the (4.9) and the two above assumptions as [14] suggests:

$$E\left[h_k h_j^*\right] = \frac{1 - e^{-j2\pi\tau_M \frac{(k-j)}{N}}}{j2\pi\tau_M \frac{k-j}{N}} \tag{4.12}$$

4.1.3 Singular Value Decomposition Low-Rank Estimator

Talking about LMMSE estimators, considering here only the frequency correlation (a second separate filter using the time correlation can be added), a lower complexity algorithm is derived, using the optimal low-rank approximation based on matrix singular value decomposition [14]. As [15] shows, a generic SVD factorization of a $M \times N$ matrix $\mathbf{B}$ is attained as:

$$\mathbf{B} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathsf{H}}, \tag{4.13}$$

where $\mathbf{U}$ and $\mathbf{V}$ are unitary matrices, i.e., $\mathbf{U}\mathbf{U}^{\mathsf{H}} = \mathbf{U}^{\mathsf{H}}\mathbf{U} = \mathbf{I}_M$ and $\mathbf{V}\mathbf{V}^{\mathsf{H}} = \mathbf{V}^{\mathsf{H}}\mathbf{V} = \mathbf{I}_N$, respectively $M \times M$ and $N \times N$ and Σ is a diagonal $M \times N$ matrix with the elements $\{\sigma_i\}$

on its diagonal called *singular values*, that are also the square roots of the eigenvalues of $\mathbf{B}\mathbf{B}^{\mathsf{H}}$ or $\mathbf{B}^{\mathsf{H}}\mathbf{B}$, arranged in non-increasing order after appropriate permutations. Now a special case occurs if the matrix $\mathbf{B}$ is Hermitian, i.e., when it's equal to its conjugated transposed, in symbols $\mathbf{B} = \mathbf{B}^{\mathsf{H}}$; its decomposition becomes then:

$$\mathbf{B} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{\mathsf{H}}, \tag{4.14}$$

that is also its eigenvalue decomposition, with the property of all real valued eigenvalues. From its definition it's clear that the covariance matrix is Hermitian (see subsection 4.2.1), so it has the latter decomposition $\mathbf{R}_{hh} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{\mathsf{H}}$, which eigenvalues $\lambda_1 \geq \lambda \geq \cdots \geq \lambda_N$, are ordered in a non-increasing way. The low complexity derives from the fact that the eigenvalues should become smaller after about $L+1$ values [14], where L is the cyclic prefix length, then the last smallest ones can be discarded. There's not an exact number of eigenvalues to keep or not, it's a tradeoff between complexity and MSE. The estimator rank will be named q, from which derive the terminology *rank-q estimator* and for the case with all pilots, its optimal expression is:

$$\hat{\boldsymbol{h}}_q = \mathbf{U}\mathbf{\Delta}_q\mathbf{U}^{\mathsf{H}}\hat{\boldsymbol{h}}^{LS}, \tag{4.15}$$

where $\hat{\boldsymbol{h}}^{LS}$ is the LS estimate at all the subcarriers and $\mathbf{\Delta}_q$ is a diagonal matrix which coefficients are related to the respective eigenvalues and to the SNR as:

$$\delta_k = \begin{cases} \frac{\lambda_k}{\lambda_k + \frac{\beta}{SNR}} & \text{for } 1 \leq k \leq q \\ 0 & \text{for } (q+1) \leq k \leq N \end{cases}. \tag{4.16}$$

The $\mathbf{\Delta}_q$ matrix then is obtained rearranging the SVD expression of the matrices $\mathbf{R}_{h\hat{h}^{LS}}\mathbf{R}_{\hat{h}^{LS}\hat{h}^{LS}}$ and keeping only the largest q singular values. These matrices are similar to that of (4.7), but in case of all pilots. Such coefficients decimation of course, gives rise to an error floor that limits the sensibility. Is shown that after $L+1$ taps it decays rapidly and that is equal to zero if no rank reduction is made. Finally from the estimation error expression $\boldsymbol{e}_q = \boldsymbol{h} - \hat{\boldsymbol{h}}_q$, an MSE lower can be obtained:

$$mse(q) = \frac{1}{N}\sum_{k=q+1}^{N}\mu_k, \tag{4.17}$$

that is the general expression in case of mismatch, i.e., when the designed channel correlation and noise are different from the real ones. When no mismatch occurs $\mu_k = \lambda_k$, contrariwise is the kth diagonal element of the matrix $\mathbf{U}^{\mathsf{H}}\mathbf{R}_{hh}\mathbf{U}$, where $\mathbf{R}_{hh}$ is the true channel correlation.

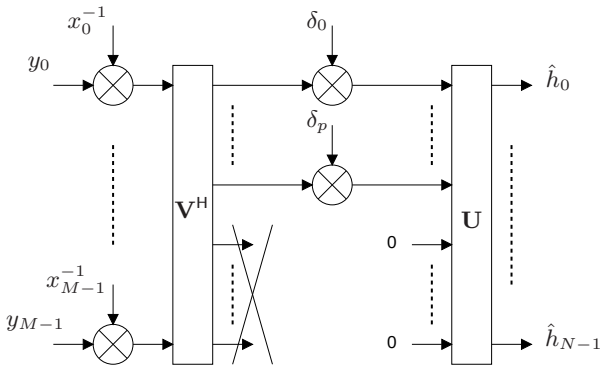

FIGURE 4.2: Rank-p PSAM estimator.

Similarly, the estimator can be used even for the PSAM case with M pilots instead that with all the N subcarriers (figure 4.2) modifying the (4.15):

$$\hat{\boldsymbol{h}}_q = \mathbf{U}\boldsymbol{\Delta}_q\mathbf{V}^{\mathsf{H}}\hat{\boldsymbol{h}}_p^{LS}, \tag{4.18}$$

noticing that the unitary matrices are no equal in this case. The estimator will have then less coefficients and the MSE will have a different expression but the working principle will be the same.

Such an estimator has been reported as an example strictly following the reference [14] and it wants to show one of the way to reduce the complexity of the Wiener Filter.

4.2 Tap-Delay Estimation

Before describing this second approach, an investigation on the array signal processing is advised due to the fact that the specific algorithm called ESPRIT, which wants to be used to estimate the multipath delays, is based on the *array manifold* and on the *signal subspace* concepts, which cause a geometrical solution to the signal parameter estimation problem. This technique in the beginning [11] was applied to the *direction of arrival* estimation problem, DOA, making use of an array of sensors with a specific geometry, but seeing the model is very similar to that one that has been used to organize the pilot observations, it's legitimate to use it to estimate the delays.

4.2.1 The Covariance Matrix

The key of the tap-delay estimation algorithms is the covariance matrix; its eigenvalue decomposition allow to identify the so-called signal subspace, that is the base of the ESPRIT and of the algorithms that have inspired it. An initial remark regarding the terminology has to be made to avoid misunderstanding. Often the autocovariance matrix

is called autocorrelation matrix or vice versa, because of their similar definitions, however there's a slight difference between them. Considering the $N \times 1$ signal vector $\boldsymbol{y}$, the covariance matrix is defined as:

$$\mathbf{R}_{yy} = E\left\{ \begin{bmatrix} y_0 - E[y_0] \\ y_1 - E[y_1] \\ \vdots \\ y_{N-1} - E[y_{N-1}] \end{bmatrix} \begin{bmatrix} y_0^* - E^*[y_0], & y_1^* - E^*[y_1], & \cdots, & y_{N-1}^* - E^*[y_{N-1}] \end{bmatrix} \right\}, \tag{4.19}$$

whereas the autocorrelation matrix (or the crosscorrelation, if it's between two different signals) is without subtracting the expectation vector. Anyhow, due to the signal definition, its mean is zero, i.e., $E[y_n] = 0$, for all n, so the two matrices can be considered the same and to follow the papers treatment will be used the latter name and the following vectorial notation:

$$\mathbf{R}_{yy} = E\left\{ \boldsymbol{y}\boldsymbol{y}^{\mathsf{H}} \right\}. \tag{4.20}$$

Now considering the channel LS estimation (4.6) introduced before is:

$$\hat{\boldsymbol{h}}_{LS} = \mathbf{T}\boldsymbol{\alpha} + \boldsymbol{v}, \tag{4.21}$$

and assuming that any component of $\boldsymbol{\alpha}$ is statistically independent from any component of $\boldsymbol{v}$, that means that the signal and the noise are uncorrelated, the theoretical covariance matrix can be written as:

$$\mathbf{R}_{\hat{h}_{LS}\hat{h}_{LS}} = \mathbf{T}\mathbf{A}\mathbf{T}^{\mathsf{H}} + \sigma_v^2 \mathbf{I}_M, \tag{4.22}$$

defining $E\left\{\boldsymbol{\alpha}\boldsymbol{\alpha}^{\mathsf{H}}\right\} = \mathbf{A}$, which is a full rank $L \times L$ diagonal matrix due to the uncorrelated scattering assumption, i.e., α_l are random uncorrelated processes and $E\left\{\boldsymbol{v}\boldsymbol{v}^{\mathsf{H}}\right\} = \sigma_v^2 \mathbf{I}_M$ which is the noise covariance matrix. Of course the theoretical matrix is not available, so some artifice has to be found to build it. Noticing that due to the assumption of pilots unitary power, the received signal $\boldsymbol{y}$ can be written similarly to the LS estimate $\hat{\boldsymbol{h}}_{LS}$ as:

$$\boldsymbol{y} = \mathbf{T}\boldsymbol{\alpha} + \boldsymbol{v} \tag{4.23}$$

and so writing $\mathbf{R}_{\hat{h}_{LS}\hat{h}_{LS}}$ or $\mathbf{R}_{yy}$ it's the same thing. To build the covariance matrix the first step is to store different time observation called *snapshots*, that are $M \times 1$ vectors $\boldsymbol{y}_k$ made by the pilot observations at the discrete-time instant k those coincide with the first and the fifth OFDM symbol in each slot for each frame. Grouping them columnwise, e.g., considering one frame, a snapshot matrix $\mathbf{Y}$, with $K = 40$ (one frame contains 20

slots with two reference symbols each) can be constructed:

$$\mathbf{Y} = \begin{bmatrix} | & | & & | \\ \boldsymbol{y}_1 & \boldsymbol{y}_2 & \cdots & \boldsymbol{y}_K \\ | & | & & | \end{bmatrix}, \tag{4.24}$$

so as, calling it *forward approach*, the covariance matrix estimate, adapting the definition (4.20) matrix to a limited observations case is the *sample covariance matrix*:

$$\hat{\mathbf{R}}_{yy} = \frac{1}{K}\mathbf{Y}\mathbf{Y}^{\mathsf{H}}. \tag{4.25}$$

Is straightforward to see that a better estimation is performed as K increase, because by definition (4.20), it should tend to infinity. Anyway, this increase the storage memory and the acquisition time as well, so a tradeoff has to be found. Since by definition, $\hat{\mathbf{R}}_{yy}$ is Hermitian and Toeplitz, i.e., conjugate symmetric and with equal elements along all diagonals, can be thought to enforce the (4.25) to have the same properties. In this regard it's shown [15] that the accuracy can be enhanced using a modified sample covariance matrix:

$$\hat{\mathbf{R}}_{yy}^{\mathsf{FB}} = \frac{1}{2}(\hat{\mathbf{R}}_{yy} + \mathbf{J}\hat{\mathbf{R}}_{yy}^{*}\mathbf{J}). \tag{4.26}$$

Such approach is named *forward-backward* and the resulting matrix is the equivalent centrosymmetric of $\hat{\mathbf{R}}_{yy}$, i.e., its elements are conjugate symmetric about both main diagonals, where $\mathbf{J}$ is the reversal $M \times M$ ones anti-diagonal matrix. In addition for the assumptions that have been made, the matrix $\mathbf{A}$ should be $L \times L$ full-rank, anyway the forward-backward approach has also a decorrelation effect that increases the rank just in case that the sources aren't perfectly uncorrelated as assumed. In any case, the most important property is the better estimation accuracy, regardless the signal correlation, that anyway it's not a drawback.

4.2.2 DOA Signal Model and Pilots Framework Analogies

Of course there is a set of analogies between the two fields, i.e., the DOA and the tap-delay estimation, so a glance to the first signal model helps to see that the structure that stays behind both the problems is the same. Considering a multitude of M sensors and L narrow-band signals coming from different directions, which have to be estimated, the measurements model is given by:

$$\boldsymbol{x}(t) = \mathbf{W}(\boldsymbol{\theta})\boldsymbol{s}(t) + \boldsymbol{n}(t), \tag{4.27}$$

where, $\boldsymbol{x}(t) = [x_1(t), \cdots, x_M(t)]^{\mathsf{T}}$ and $\boldsymbol{n}(t) = [n_1(t), \cdots, n_M(t)]^{\mathsf{T}}$ are $M \times 1$ vectors corresponding respectively to the output of the M sensors and to the noise, $\boldsymbol{s}(t) = [s_1(t), \cdots, s_L(t)]^{\mathsf{T}}$ is a $L \times 1$ vector that represent the signals coming from the L directions and $\mathbf{W}(\boldsymbol{\theta}) = [\boldsymbol{a}(\theta_1), \cdots, \boldsymbol{a}(\theta_L)]$ is a $M \times L$ steering matrix, whose L columns $\boldsymbol{a}(\theta_l)$ are for construction, $M \times 1$ independent vectors, each one corresponding to the array response from a certain direction:

$$\boldsymbol{a}(\theta_l) = \left[1, e^{-j\omega_0\tau(\theta_l)}, \cdots, e^{-j\omega_0(M-1)\tau(\theta_l)}\right]^{\mathsf{T}}, \quad 1 \leq l \leq L. \tag{4.28}$$

Very briefly the geometric problem is the following: in the absence of noise, the columns of $\mathbf{W}(\boldsymbol{\theta})$ span the so-called *signal subspace*, so knowing the directional array response termed *array manifold*, L independent signals, i.e., vectors, would be enough to determine the parameters, by the intersection of the array manifold and the signal subspace. However due to the fact that the measurements are noisy (the noise and the signal are though assumed uncorrelated, zero-mean processes), such subspace has to be estimated by finding a set of L independent vectors that spans it, for instance, examining the covariance matrix of the measurements. Neglecting the noise it has the eigenvalue decomposition:

$$\mathbf{R}_{xx} = \mathbf{W}\mathbf{S}\mathbf{W}^{\mathsf{H}}, \tag{4.29}$$

where, $\mathbf{S} = E\left\{\boldsymbol{s}\boldsymbol{s}^{\mathsf{H}}\right\}$ is the signal covariance matrix, that is diagonal if the signal are uncorrelated how in this case and $\mathbf{W}$ is just the same matrix of the (4.27) with the $\boldsymbol{\theta}$ dependence omitted. Since the rank of $\mathbf{W}\mathbf{S}\mathbf{W}^{\mathsf{H}}$ is L [15] it follows that it has the same number of nonzero eigenvalues, thus arrange them in a non-increasing order, $\tilde{\lambda}_1 \geq \tilde{\lambda}_2 \geq \cdots \geq \tilde{\lambda}_L$, will be the nonzero values, while $\tilde{\lambda}_{L+1} \geq \cdots \geq \tilde{\lambda}_N$ will be equal to zero, i.e., the last $M - L$. Considering now the noise in the (4.26), the (4.27) becomes:

$$\mathbf{R}_{xx} = \mathbf{W}\mathbf{S}\mathbf{W}^{\mathsf{H}} + \sigma_n^2\mathbf{I}_M, \tag{4.30}$$

that yields the noisy eigenvalues:

$$\lambda_m = \tilde{\lambda}_m + \sigma_n^2, \quad 1 \leq m \leq L \tag{4.31}$$

and the noise eigenvalues:

$$\lambda_m = \sigma_n^2, \quad L + 1 \leq m \leq M. \tag{4.32}$$

The first L will be then increased by the noise variance, while the last $M - L$ will be equal to the noise variance, suggesting a split in two subsets, i.e., the signal and the noise subset, with the respective associated eigenvectors. Now for the orthogonality property of the eigendecomposition, such set of linearly independent vectors mentioned

above that should span the signal subspace, is nothing more than the set of the L eigenvectors of $\mathbf{R}_{xx}$, corresponding to the L largest eigenvalues, which are the non-zero eigenvalues of the noise-free $M \times M$ rank L matrix $\mathbf{WSW}^{\mathsf{H}}$, plus the noise variance σ_n^2 [15]. On the other hand, the $M - L$ remaining eigenvalues are equal to σ_n^2 and the corresponding eigenvector, that are orthogonal to the eigenvectors associated with the L largest eigenvalues, span the so-called *noise subspace* the null space of $\mathbf{R}_{xx}$ as well. This property is particularly interesting because it can be exploited to estimate either the noise variance (for instance when $\boldsymbol{s}(t)$ is not present) and the number of delays.

The problem of the covariance matrix estimation has already been treated in the subsection 4.2.1, due to the fact that the real one is not available so the analogies with the pilots framework can be listed by a vectors-matrices comparison:

$$\begin{cases} \boldsymbol{x}(t) & \rightarrow & \hat{\boldsymbol{h}}_{LS}(k) \\ \mathbf{W}(\boldsymbol{\theta}) & \rightarrow & \mathbf{T}(\boldsymbol{\tau}(k)) \\ \boldsymbol{s}(t) & \rightarrow & \boldsymbol{\alpha}(k) \\ \boldsymbol{n}(t) & \rightarrow & \boldsymbol{v}(k) \end{cases} \qquad \begin{cases} t & \rightarrow & k \\ \boldsymbol{\theta} & \rightarrow & \boldsymbol{\tau}(k) \end{cases}. \tag{4.33}$$

It's shown that there is a correspondence between the time variables t and k, and between the unknown parameters $\boldsymbol{\theta}$ and $\boldsymbol{\tau}(k)$, that allows to write the $\mathbf{T}$ matrix from (4.6) as [3] do:

$$\mathbf{T}(\boldsymbol{\tau}(k)) = \begin{bmatrix} | & | & & | \\ \boldsymbol{t}(\tau_1(k)) & \boldsymbol{t}(\tau_2(k)) & \cdots & \boldsymbol{t}(\tau_L(k)) \\ | & | & & | \end{bmatrix}, \tag{4.34}$$

with the column vectors:

$$\boldsymbol{t}(\tau_l) = \left[e^{-j2\pi p(1)\frac{\tau_l}{N}}, e^{-j2\pi p(2)\frac{\tau_l}{N}}, \cdots, e^{-j2\pi p(M)\frac{\tau_l}{N}} \right]^{\mathsf{T}}, \quad \tau_l \geq 0. \tag{4.35}$$

Finally, the ESPRIT algorithm that was introduced in [11] is based on a particular sensors geometry that is a planar array composed by M doublets, whose elements are identical and separated by a known constant *displacement vector*. This, gives rise to a rotation operator between the signals impinging on the doublets elements, which is used to estimate the unknown parameters.

4.2.3 ESPRIT

After the underlying analogies between the two frameworks, the ESPRIT algorithm (estimation of signal parameters by rotational invariance techniques) can be introduced: first of all the signal and the noise subspaces have to be estimated, assuming known via previous estimation, the number of delays in order to select the L largest sample

covariance matrix eigenvalues, ($[\mathbf{U}_s, \mathbf{\Lambda}_s]$ and $[\mathbf{U}_n, \mathbf{\Lambda}_n]$ are defined as the pairs [eigenvectors,eigenvalues] of respectively signal and noise); then as suggests in [3], two $(M/2) \times L$ submatrices $\mathbf{T}_1(\boldsymbol{\tau})$ and $\mathbf{T}_2(\boldsymbol{\tau})$ are obtained by splitting the matrix $\mathbf{T}$:

$$\begin{aligned} \mathbf{T}_1(\boldsymbol{\tau}) &= \begin{bmatrix} \mathbf{I}_{M/2} & 0 \end{bmatrix} \mathbf{T} \\ \mathbf{T}_2(\boldsymbol{\tau}) &= \begin{bmatrix} 0 & \mathbf{I}_{M/2} \end{bmatrix} \mathbf{T}. \end{aligned} \tag{4.36}$$

The acronym explanation is given then by the relationship between these submatrices, i.e., the rotational operator $\mathbf{D}$:

$$\mathbf{T}_2(\boldsymbol{\tau}) = \mathbf{T}_1(\boldsymbol{\tau})\mathbf{D}, \tag{4.37}$$

where

$$\mathbf{D} = diag \left\{ e^{-j2\pi \frac{\Delta_p \frac{m}{2}+1}{N} \tau_1}, \cdots, e^{-j2\pi \frac{\Delta_p \frac{m}{2}+1}{N} \tau_L} \right\}. \tag{4.38}$$

The displacement factor is known by construction and the two submatrices are identical except for a shift due to such factor: that denotes that a displacement of $\Delta_p \frac{M}{2} + 1$ subcarriers is needed to rotate $\mathbf{T}_1(\boldsymbol{\tau})$ into $\mathbf{T}_2(\boldsymbol{\tau})$. After some manipulations following the signal subspace estimation, the next relationship is derived:

$$\mathbf{U}_s = \mathbf{TQ}, \tag{4.39}$$

where $\mathbf{Q} = \mathbf{A}\mathbf{T}^{\mathsf{H}}\mathbf{U}_s(\mathbf{\Lambda}_s - \sigma_v^2 \mathbf{I}_L)^{-1}$, that is used to define the $\mathbf{U}_s$ submatrices, $\mathbf{U}_{s,1} = \mathbf{T}_1\mathbf{Q}$ and $\mathbf{U}_{s,2} = \mathbf{T}_2\mathbf{Q}$ related by:

$$\mathbf{U}_{s,2} = \mathbf{U}_{s,1}\mathbf{Q}^{-1}\mathbf{D}\mathbf{Q}. \tag{4.40}$$

Then, defining the matrix $\mathbf{\Phi} = \mathbf{Q}^{-1}\mathbf{D}\mathbf{Q}$, the (4.40) can be written as:

$$\mathbf{U}_{s,2} = \mathbf{U}_{s,1}\mathbf{\Phi}, \tag{4.41}$$

that is the arrival point: the core of such algorithm is that the parameters matrix $\mathbf{D}$ and $\mathbf{\Phi}$ share the same eigenvalues, so seeing the second is known because obtained by the (4.41), multiplying $\mathbf{U}_{s,2}$ by the inverse of $\mathbf{U}_{s,1}$, its eigenvalues λ_l can be calculated, as well as can be calculated the delays exploiting the inverse relationship:

$$\hat{\tau}_l = \frac{arg(\hat{\lambda}_l^*)N}{2\pi(\frac{M}{2}+1)}, \quad 1 \leq l \leq L. \tag{4.42}$$

The matrices $\mathbf{U}_{s,1}$ and $\mathbf{U}_{s,2}$ are stated as known because they derive from the selection of the signal subspace based on the largest L eigenvalues.

Now that an example has been proposed for both the estimation approaches, two chapters follow, with a more experimental orientation: while the theory discussed until now is strongly consolidate, the target is to improve the channel estimation performance by the investigation on the *spatial smoothing* technique and on the *number of delay estimation* (they are assumed known until now) and their combination, that is still an open field.

Chapter 5

Spatial Smoothing

Chapter 4 discussed an eigenstructure technique, that applied to an array of frequencies observations, e.g., the pilot subcarriers, allows to define the signal and the noise subspaces. This splitting is the base of the tap-delay estimation by the ESPRIT algorithm [11], whose application has been proposed in [10] as initial delays estimation tool. The major problem of this approach is the accuracy of the covariance matrix estimate: the whole proceeding is based on its eigenvalue decomposition, that would yield a straightforward subspace splitting (see subsection 4.2.2), if such matrix was the theoretical one. The way to estimate it, often referred as *forward approach* has been proposed in (4.19) and (4.20), that are recasted:

$$\mathbf{Y} = \begin{bmatrix} | & | & & | \\ \boldsymbol{y}_1 & \boldsymbol{y}_2 & \cdots & \boldsymbol{y}_K \\ | & | & & | \end{bmatrix}, \qquad \hat{\mathbf{R}}_{yy} = \frac{1}{K}\mathbf{Y}\mathbf{Y}^{\mathsf{H}}. \tag{5.1}$$

In practice the problem is K: although if it's high it improves the matrix accuracy, as it increases, also the amount of storage memory and the observation time increase. This second issue is critical for the time-varying nature of the channel: some delays could change, die, some other could be born and in general it varies while the snapshots are collected. Then qualitatively speaking, if this storage time is too high, these variations could be significant and the covariance matrix would be obtained by snapshots, differently affected by the channel, that intuitively is not good.

Here a preprocessing scheme named *spatial smoothing*, is proposed to artificially supply more snapshots, i.e., to collect more time observation (K) without waiting any time to do it. To introduce it, it's behavior is initially shown graphically by figure 5.1 (the numbered black rounds can be considered as equally spaced pilot observations, selected

each time by a sliding window). Then some derivations follow, to see how it will be implemented in practice.

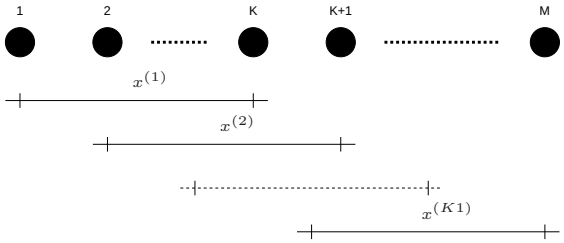

FIGURE 5.1: Array smoothing scheme.

The above smoothing scheme was proposed in [16], because the examined eigenstructure yields a correct estimate only if the sources are partially or totally uncorrelated, i.e., *noncoherent*. Now, this problem was the one that motivated the analysis of a technique to decorrelate the sources, i.e., to restore the rank of $\mathbf{S}$ in (4.29). However, in the considered setup this problem doesn't arise due to the US assumption, though, the spatial smoothing can also be employed, just to improve the estimation accuracy. This is what is necessary, since over a dynamic channel, it's preferable to collect less snapshots as possible, and this is counter the covariance estimation (5.1). The idea is to split the array of size M in a set of $K1$ overlapping smaller subarrays of size $M1$ by a sliding window, calculate the covariance matrices, that will be $M1 \times M1$ instead of $M \times M$ for each one and average them to obtain a smoothed matrix. In the next sections the smoothing expression and how to exploit the described theory to reshape the snapshots matrix will be shown, following the notation and the argument in [16], in order to improve the covariance matrix estimate, whose eigenvalue decomposition is supposed to give better results.

5.1 Smoothed Covariance Matrix

First of all, the received signal expression (4.23), followed by its covariance matrix decomposition ($\mathbf{R}_{yy} = \mathbf{T}\mathbf{A}\mathbf{T}^{\mathsf{H}} + \sigma_v^2\mathbf{I}_M$) properties, is recalled:

$$\boldsymbol{y} = \mathbf{T}\boldsymbol{\alpha} + \boldsymbol{v}, \tag{5.2}$$

- $\mathbf{A}$ is a full rank $L \times L$ diagonal matrix due to the US assumption
- $\mathbf{T}$ is a $M \times L$ matrix of linearly independent column vectors
- the smallest eigenvalues are $M - L$ and are equal to σ_v^2

- the eigenvectors associated with the smallest eigenvalues are orthogonal to the columns of $\mathbf{T}$, i.e., to the signal subspace

Thus the preprocessing scheme behavior can be illustrated. It's composed by two steps: a sliding window operation and an average. The first one decreases the size of the pilot observations, selecting by a window just $M1$ pilots each time and this selection is performed $M - M1 + 1$ times, i.e., $K1$. Then for each shift, the covariance matrix of the received signal is calculated. The sliding window operation is represented by the $L \times L$ matrix $\mathbf{C}^{(\Gamma)}$, with $\Gamma = 1, \cdots, K1$:

$$\mathbf{C} = diag\left\{e^{-j2\pi\frac{\Delta_p}{N}\tau_1}, \cdots, e^{-j2\pi\frac{\Delta_p}{N}\tau_L}\right\}, \tag{5.3}$$

applied to the submatrix $\mathbf{T}_{M1}$ by selecting the first $M1$ rows of $\mathbf{T}$ described in (4.34) and (4.35). After reducing to $M1$ the dimension of $\boldsymbol{\alpha}$ and $\boldsymbol{v}$ to obtain $\boldsymbol{\alpha}_{M1}$ and $\boldsymbol{v}_{M1}$, the received signal at the Γth window is:

$$y_{M1}^{(\Gamma)} = \mathbf{T}_{M1}\mathbf{C}^{(\Gamma-1)}\boldsymbol{\alpha}_{M1} + \boldsymbol{v}_{M1}. \tag{5.4}$$

To select for example the second window, it has to be $\Gamma = 2$, so as the product $\mathbf{T}_{M1}\mathbf{C}^{(1)}$ provided a matrix that is a $M1 \times L$ submatrix of $\mathbf{T}$, made by the rows $\{2, \cdots, M1 + 1\}$ and so on. Thus, by the covariance matrix definition (4.20), the Γth matrix is obtained by the expected value of the product between (5.4) and its Hermitian:

$$\mathbf{R}_{M1}^{(\Gamma)} = \mathbf{T}_{M1}\mathbf{C}^{(\Gamma-1)}\mathbf{A}_{M1}\mathbf{C}^{(\Gamma-1)^{\mathsf{H}}}\mathbf{T}_{M1}^{\mathsf{H}} + \sigma_v^2\mathbf{I}_{M1}. \tag{5.5}$$

The second and the last step just averages the $K1$ covariance matrices from (5.5) defining the $M1 \times M1$ *spatially smoothed matrix*. This is simply achieved by summing the matrices obtained from (5.5) and dividing the result by $K1$:

$$\bar{\mathbf{R}}_{yy} = \frac{1}{K1}\sum_{\Gamma=1}^{K1}\mathbf{R}_{M1}^{(\Gamma)}. \tag{5.6}$$

However, the improved accuracy of the covariance matrix has a drawback: decreasing the frequency domain dimension, decreases the ability to distinguish closely spaced signal sources [17], i.e., the *resolution*. Why the frequency dimension decreases will be shown graphically in the next section; for the moment the point that has to be clear is that a tradeoff between the time (the number of snapshots) and the frequency observations has to be found. Favoring the first will lead to a better accuracy; contrarily, a bigger resolution will be attained.

5.1.1 Application to the Snapshots Matrix

Considering the sample covariance matrix from (5.1), its theoretical definition is an expected value that foresees an infinite summation over k; thus, a manner to increase K, that is the column index that should tend to infinity, would better approximate it. The problems that arise from increasing it were just mentioned at the beginning of the chapter. Since the WSS assumption holds, i.e., the channel statistics depend only on the frequency shift and not on the absolute frequency, is reasonable to exploit the frequency observations to increase the time observation, but how to do it, is not that evident from the smoothed matrix expressions (5.5) and (5.6).

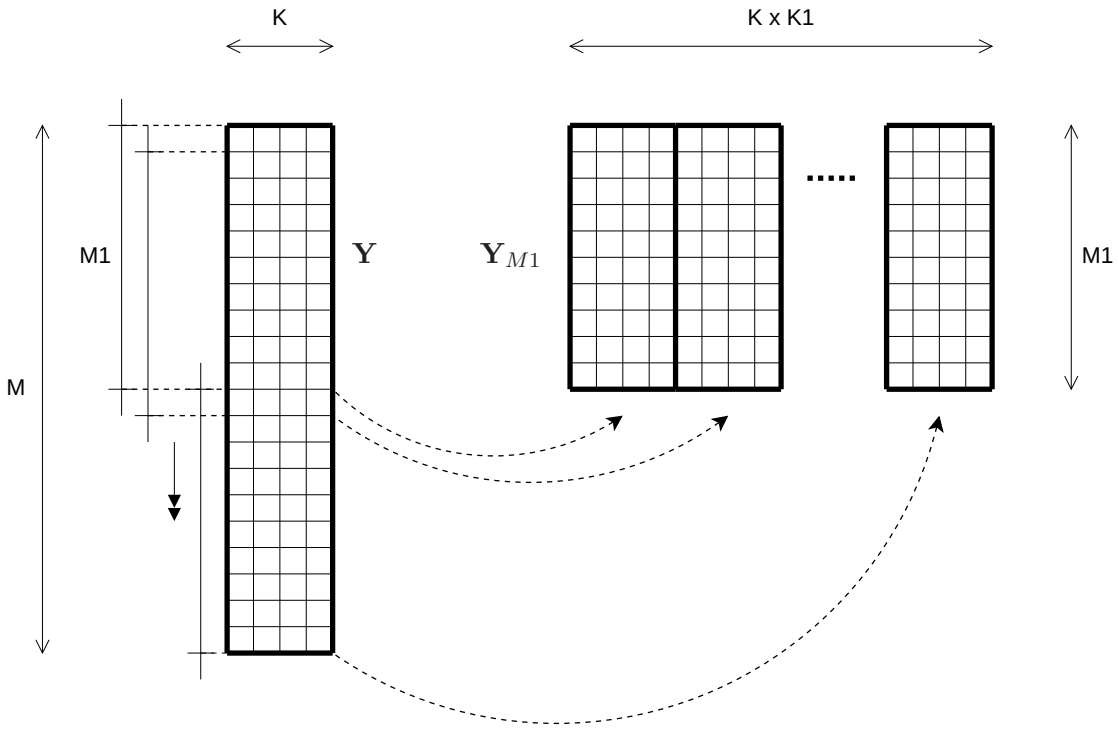

FIGURE 5.2: Snapshots matrix smoothing scheme.

In order to derive that, a set of covariance submatrices has to be defined, once again due to the fact that the theoretical matrices (5.5) are not available. The smoothing scheme that exploit such submatrices is shown in figure 5.2, resuming some parameters:

$$\begin{aligned} M&: \quad \text{number of pilots} \\ K&: \quad \text{number of snapshots} \\ M1&: \quad \text{window size} \\ K1&: \quad \text{number of windows} \\ L&: \quad \text{number of delays} \end{aligned} \tag{5.7}$$

Step by step, it just selects a submatrix from the snapshots matrix (5.1) and appends them row-wise to build a new snapshots matrix that has more snapshots and less frequency observation. This increases K without waiting any physical time to collect more snapshots as wanted. The matrix $\mathbf{Y}_{M1}$ is the reshaped snapshots matrix and it's composed by $K1$ submatrices $\mathbf{Y}_{M1}^{(\Gamma)}$, that are obtained from (5.4) by the analogy with (4.23) as:

$$\mathbf{Y}_{M1}^{(\Gamma)} = \begin{bmatrix} | & | & & | \\ \boldsymbol{y}_{M1_1}^{(\Gamma)} & \boldsymbol{y}_{M1_1}^{(\Gamma)} & \cdots & \boldsymbol{y}_{M1_K}^{(\Gamma)} \\ | & | & & | \end{bmatrix}. \tag{5.8}$$

In the same way as (5.1), using the submatrices (5.8), the estimated Γth sample covariance matrix is then:

$$\hat{\mathbf{R}}_{M1}^{(\Gamma)} = \frac{1}{K}\mathbf{Y}_{M1}^{(\Gamma)}\mathbf{Y}_{M1}^{(\Gamma)^{\mathsf{H}}}, \tag{5.9}$$

thus at this point, only the average has to be performed, like in (5.6) but using the matrices in (5.9):

$$\hat{\bar{\mathbf{R}}}_{yy} = \frac{1}{K \times K1}\sum_{\Gamma=1}^{K1}\mathbf{Y}_{M1}^{(\Gamma)}\mathbf{Y}_{M1}^{(\Gamma)^{\mathsf{H}}}. \tag{5.10}$$

However, $\hat{\bar{\mathbf{R}}}_{yy}$ obtained from the scheme in figure 5.2 could seem different if compared with (5.10); it's expression indeed, is:

$$\hat{\bar{\mathbf{R}}}_{yy} = \frac{1}{K \times K1}\mathbf{Y_{M1}}\mathbf{Y_{M1}^{\mathsf{H}}}. \tag{5.11}$$

The difference is that (5.10) derives Γ covariance matrices, multiplying them by their transposed and summing the result step by step. Contrariwise (5.11), first arranges in $\mathbf{Y_{M1}}$ all the submatrices and just perform a multiplication by its transposed; then the division by $K \times K1$ is present in both the expressions. An example with $K1 = 2$ is illustrated, to show that (5.10) and (5.11) lead to the same result. Figure 5.3(a) shows the calculation of the element $\hat{\bar{\mathbf{R}}}_{yy}(1,1)$ from (5.11) neglecting the term $1/2K$, while figure 5.3(b) of the elements $K\hat{\mathbf{R}}_{M1}^{(1)}(1,1)$ and $K\hat{\mathbf{R}}_{M1}^{(2)}(1,1)$ from (5.9) i.e., respectively:

$$\hat{\bar{\mathbf{R}}}_{yy}(1,1) = \sum_{k=1}^{2K} Y_{M1}(1,k)Y_{M1}^{\mathsf{H}}(k,1), \tag{5.12}$$

$$K\hat{\mathbf{R}}_{M1}^{(1)}(1,1) = \sum_{k=1}^{K} Y_{M1}^{(1)}(1,k)Y_{M1}^{(1)^{\mathsf{H}}}(k,1), \quad K\hat{\mathbf{R}}_{M1}^{(2)}(1,1) = \sum_{k=1}^{K} Y_{M1}^{(2)}(1,k)Y_{M1}^{(2)^{\mathsf{H}}}(k,1). \tag{5.13}$$

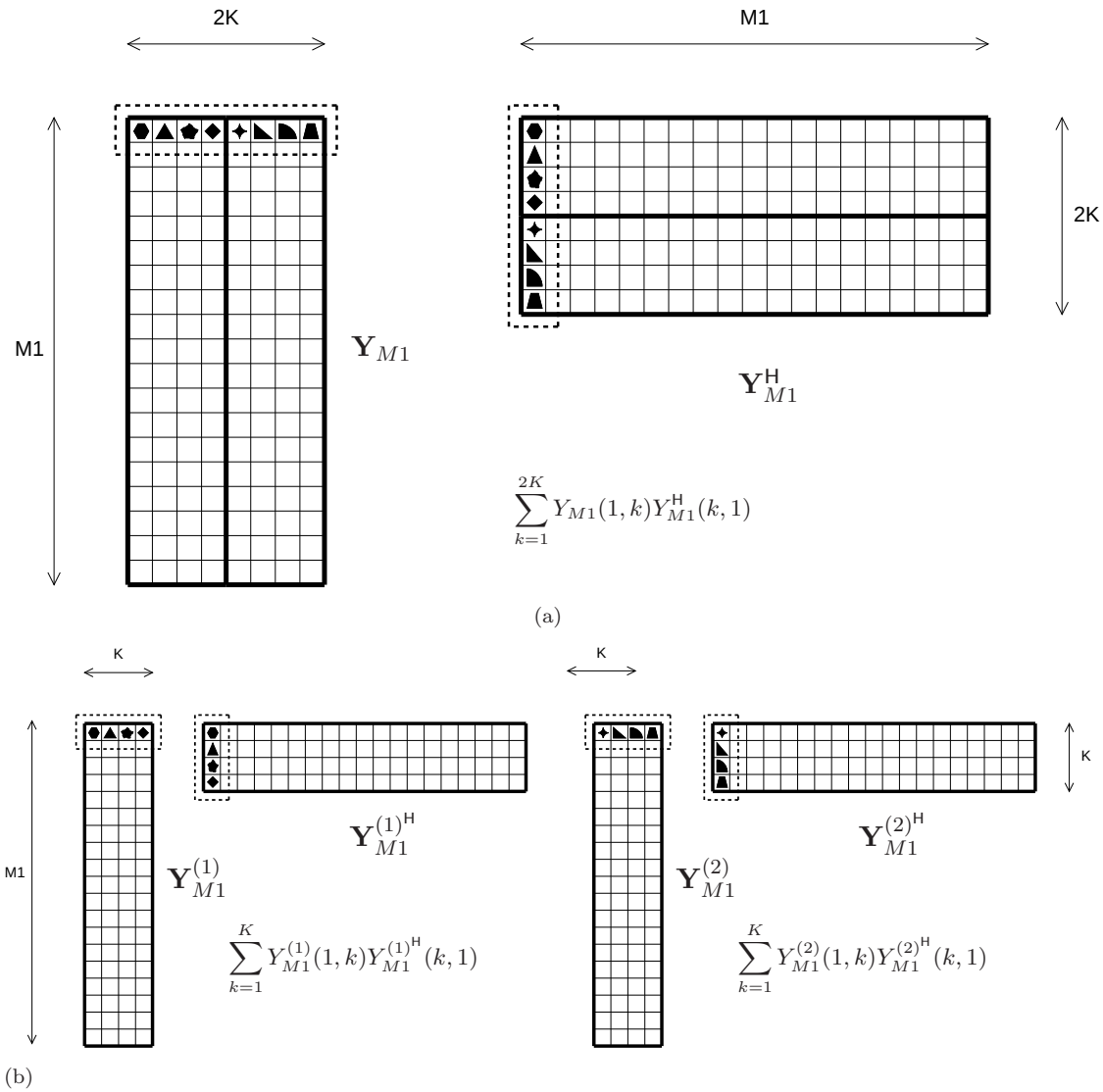

FIGURE 5.3: Spatial smoothing calculation draft.

However the summation (5.12) can be split in two summation respectively with $1 \leq k \leq K$ and $(K+1) \leq k \leq 2K$ and then rewritten as:

$$\hat{\hat{\mathbf{R}}}_{yy}(1,1) = \sum_{k=1}^{K} Y_{M1}(1,k)Y_{M1}^{\mathsf{H}}(k,1) + \sum_{k=K+1}^{2K} Y_{M1}(1,k)Y_{M1}^{\mathsf{H}}(k,1), \tag{5.14}$$

but the shapes within the matrices in figure 5.3 show that:

$$\sum_{k=1}^{K} Y_{M1}^{(1)}(1,k)Y_{M1}^{(1)\mathsf{H}}(k,1) = \sum_{k=1}^{K} Y_{M1}(1,k)Y_{M1}^{\mathsf{H}}(k,1) \tag{5.15}$$

and:

$$\sum_{k=1}^{K} Y_{M1}^{(2)}(1,k)Y_{M1}^{(2)\mathsf{H}}(k,1) = \sum_{k=K+1}^{2K} Y_{M1}(1,k)Y_{M1}^{\mathsf{H}}(k,1), \tag{5.16}$$

that concludes explanation. In addition, such scheme can be exploited also together with FB approach (4.26): in case (5.10) either directly on $\hat{\bar{\mathbf{R}}}_{yy}$ or on each $\hat{\mathbf{R}}_{M1}^{(\Gamma)}$, to perform the average afterwards, while in case (5.11), just on $\hat{\bar{\mathbf{R}}}_{yy}$. The latter will be the case used in the simulations. In any case the target to artificially increase the observations from K to $K \times K1$, reducing the frequency observation from M to $M1$ is achieved.

Anyhow, since $M1$ and $K1$ are related, the incrementing one involves the other's decreasing and that's why a compromise between them is mandatory. Although an analytic description to achieve the optimum window size is not known in literature yet, some simulations in [18] provide interesting results in terms of BER, function of the window size.

Chapter 6

Number of Delays Estimation

This chapter deals with an aspect of multipath delay estimation that has been omitted so far. Most of the known estimation algorithms based on the signal subspace research, don't contemplate how to estimate the number of delays, but assume it as known, i.e., derived from another algorithm like in [10]. From the noiseless covariance matrix eigenvalue decomposition (4.29), a number of L nonzero eigenvalues results, being the rank of $\mathbf{S}$ equal to L, that is the number of the different delays. The knowledge of this value L would lead to a perfect subspace splitting, in terms of dimensions, but it must be estimated as well, because a wrong assumption could bring to an estimation error. Here follows a motivation about the easier criterion that could be used in the simulations to estimate the number of delays, following the covariance matrix eigenvalue decomposition scheme. Then two well known *model order selection* tools, i.e., the AIC, *Akaike information criterion* and the MDL, *minimum description length*, will be illustrated, in order to compare their results with the adopted trivia method.

6.1 Subspaces Splitting by Eigenvalues Thresholding

First of all it's recalled that all the sizes parameters follow from the definitions of (5.7). Examining the eigenvalues structure (4.31) and (4.32), it could seem very easy to identify the two subsets: with a threshold operation, all the eigenvectors corresponding to the eigenvalues greater than σ_n^2 would span the signal subspace, while the remaining the noise subspace. However, as [19] recalls, all the smallest eigenvalues are not equal but just clustered around the noise variance σ_n^2, with a deviation that depends on the number of observations M and snapshots K and has been shown in [20] to be $O(1/\sqrt{K})$, if $K \gg M$. The problem is that such deviation has been derived on a nonsmoothed snapshots matrix, but that preprocessing scheme is going to be applied. Then by construction (figure 5.2),

the smoothed snapshots matrix has different properties from the nonsmoothed one, so it's not proper to adapt the above deviation according to the varying dimensions $K1$ and $M1$. Taking it into account, if the SNR is known, (in most of practical cases it is, because the receiver needs to periodically estimate it in order to report it back to the transmitter) it's straightforward to split the two subspaces by a threshold, whose selection will be discussed after the simulation results in section 7.2. Considering for instance the eigenvalues (4.31) and (4.32), a qualitative plot is shown in figure 6.1, to make the concept more clear. Furthermore, this splitting will be referred as *Threshold*, to identify it when compared with other algorithms.

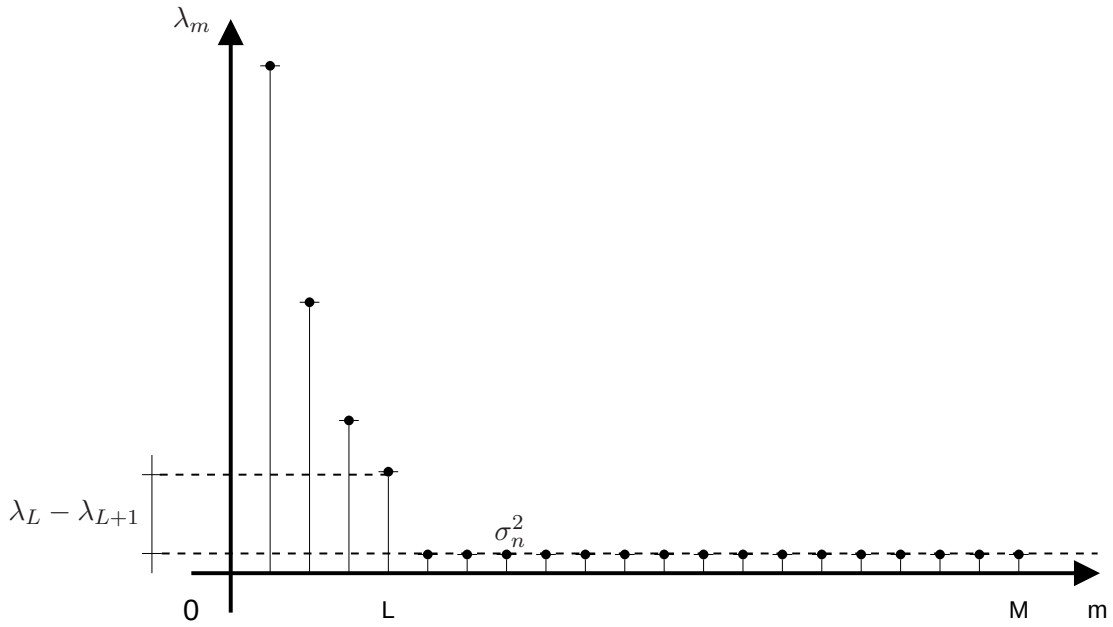

FIGURE 6.1: Theoretical eigenvalues qualitative plot.

As mentioned, after λ_L, the remaining $M-L$ values should be theoretically equal to the noise variance, thus it can be thought to exploit the gap $\lambda_L - \lambda_{L+1}$ showed in figure 6.1, to threshold the largest eigenvalues, in order to identify the signal subspace dimensions. However, the real plot doesn't foresee that the smallest values were all equal to σ_n^2, rather that they are different with probability one; but this doesn't forbid to use the *Threshold*, even though its value has to be adjusted. It cannot be equal to σ_n^2 indeed, because some of the noisy eigenvalues greater than σ_n^2 could be confused as signal, overestimating so, the signal subspace dimensions. On the other hand the threshold must be smaller than λ_L in order to don't underestimate them, and of course the greater is the distance, the easier it's to split the two subsets. Briefly, if the threshold is assumed to be equal to σ_{th}^2 the eigenvalues will suffer the splitting, described by the following inequalities system:

$$\begin{cases} \lambda_m \geq \sigma_{th}^2 & \text{for } 1 \leq m \leq L_{th} \\ \lambda_m < \sigma_{th}^2 & \text{for } L_{th}+1 \leq m \leq M \end{cases}. \tag{6.1}$$

Thus, counting the number of eigenvalues greater than σ^2_{th}, will yield the value L_{th}, that will represent the signal subspace dimension, i.e., the number of estimated delays.
In addition, if the noise statistic wasn't known, it's estimation could be performed by averaging a reasonable amount of the smallest eigenvalues, whose average should better approximate σ^2_n with a higher number of involved values. The best case would consist in transmitting a null signal and averaging all the M values though, but although this would lead to a better estimation, it implies some overhead. In the first case anyway, the number $M - \hat{L}$ of λ_m (4.32), that should be averaged, is unknown. Thus it can be supposed and increased by one margin to perform the average of the values afterwards, considering that a particular environment has its typical number of delays. However, the estimation of SNR is outside the scope of this work, and therefore is assumed to be known.

6.2 AIC and MDL Detection Schemes

Although the above *Threshold* is very simple and immediate to apply, it's interesting to consider the AIC and MDL criteria as number of delays estimation tools following the derivations in [21] [19]. The reason is that the SNR must be available, the threshold has to be selected and it's still a quite subjective operation. Contrariwise, looking at the problem from a statistical point of view, it becomes a question of *model selection* and precisely of *order selection*. Given the received data, i.e., the rank-l covariance matrix corrupted by the noise, there is a set of potential models that can describe the original noiseless data that generate it. Such model is nothing more than a parameters vector that includes the eigenvalues λ_l, the corresponding eigenvectors $\boldsymbol{u}_l$ and the noise variance σ^2_n:

$$\boldsymbol{\Theta}^{(l)} = \left[\lambda_1, \cdots, \lambda_l, \sigma^2_n, \boldsymbol{u}_1^{\mathsf{T}}, \cdots, \boldsymbol{u}_l^{\mathsf{T}}\right]^{\mathsf{T}}. \tag{6.2}$$

The subscript l is important because represents the order of the model, i.e., the rank of the covariance matrix (e.g., (5.11), in case of smoothing, that will be the case of interest), i.e., the number of delays. Since the covariance matrix is uniquely defined by the received data, the different models of (6.2) are a function of l: as it increases, more eigenvalues move to the signal subset. Intuitively, the models have to be compared by increasing l from 1 to M, thus modifying the subspaces dimension at each step. Although a maximum-likelihood approach would lead to an optimal solution, its complexity makes it unusable, thus a useful log-likelihood function has been proposed in [20]. It derives from the observations of the matrix (4.24), that are regarded as statistically independent complex Gaussian random vectors with zero mean, through their joint probability density function $f(\boldsymbol{y}_1, \cdots, \boldsymbol{y}_K|\boldsymbol{\Theta}^{(l)})$. Taking the logarithm and omitting the terms that

do not depend on the parameters vector, such function based on both the signal and the noise eigenvalues estimate is:

$$L(\hat{\boldsymbol{\Theta}}^{(l)}) = \log \left(\frac{\prod_{i=l+1}^{M} \hat{\lambda}_i^{\frac{1}{M-l}}}{\frac{1}{M-l} \sum_{i=l+1}^{M} \hat{\lambda}_i} \right)^{(M-l)K}, \tag{6.3}$$

noticing that term at the denominator is the noise estimation as proposed above, i.e., the average of the smallest eigenvalues that vary with l, and that $\hat{\lambda}_i$ are the estimation of (4.31), both from the sample covariance matrix (4.25). The proposed criteria adapts the likelihood function (6.3) through the a second term that takes into account the numbers of degrees of freedom of the space spanned by the vector $\boldsymbol{\Theta}^{(l)}$, considering that the eigenvectors are complex and orthonormal. Such term is introduced as a correction term and it's different in the two metrics because the problem has been approached from different points of view, as briefly cite [21]. Accordingly the two expressions follow:

$$\mathrm{AIC}(l) = -2 \log \left(\frac{\prod_{i=l+1}^{M} \hat{\lambda}_i^{\frac{1}{M-l}}}{\frac{1}{M-l} \sum_{i=l+1}^{M} \hat{\lambda}_i} \right)^{(M-l)K} + \quad 2b(M,l) \tag{6.4}$$

and

$$\mathrm{MDL}(l) = -\log \left(\frac{\prod_{i=l+1}^{M} \hat{\lambda}_i^{\frac{1}{M-l}}}{\frac{1}{M-l} \sum_{i=l+1}^{M} \hat{\lambda}_i} \right)^{(M-l)K} + \quad \frac{1}{2} b(M,l) \log K. \tag{6.5}$$

Both look for the value of l that minimizes them in the range $1 \leq l \leq M$ Thus, the estimated number of delays by the respective metrics follow:

$$\hat{L}_{AIC} = \arg \min_{l \in \{1, \cdots, M\}} AIC(l) \tag{6.6}$$

and

$$\hat{L}_{MDL} = \arg \min_{l \in \{1, \cdots, M\}} MDL(l). \tag{6.7}$$

Anyhow, if M is big the computational complexity could be relevant without any advantages; one can think then, to consider another range $1 \leq l \leq S$, with $S < M$, in order to reduce the complexity. This doesn't lead to any loss of information because

qualitatively speaking, the estimated number of delays should fluctuate around the typical number of relevant components L_{typ}, of a certain environment. Then if the margin between L_{typ} and S is reasonable, the number of estimated delays will be always under S, but avoiding $M-S$ trials. Finally, the above algorithms have been proposed for both the forward [21] and the forward-backward approach [19], that differ for the number of adjustable parameters, i.e., $b(M,l)$, because of their sample covariance matrix structure (see table 6.1).

$forward$	$b(M,l) = l(2M-l)$
$forward-backward$	$b(M,l) = \frac{1}{2}l(2M-l+1)$

TABLE 6.1: Free Adjustable Parameters Functions

Here all the number of delays estimation tools that will be used, have been shown; the final step of the thesis consists then in the combination of the spatial smoothing with the above procedures. The behavior will be examined through simulations on both the sample covariance matrix and its centrosymmetric version taking into account how the correction term $b(M,l)$ could vary. Finally all the results will be compared with the *Threshold* as well, to see if the complexity required by AIC and MDL is worth to use them as number of delays estimators.

Chapter 7

Simulations Results

This chapter summarize the results of the simulated *Threshold* and MDL algorithms, described in chapter 6, in order to compare their performances, that hopefully will yield the most suitable technique to estimate the number of delays. The AIC detection scheme has been discarded due to its poor capability in estimating the number of delays. The simulations have been performed using an OFDM transceiver simulator, that is a simplified reviewed version of that one implemented in [3], whose structure is detailed in appendix A. After introducing the error metrics that will be used to measure the performance, an heuristic justification to the threshold value will be given and after verifying which approach between the *forward* and the *forward-backward* is the best, a set of comparisons will be shown arranged in two main sections. The channel will be first considered static, to swap then to the dynamic channel, over which the most interesting static setups will be repeated. Then, a final comparison between the static and the dynamic channel is mandatory, since it wants to be shown how the algorithms introduced for the first, work for the latter, leading to more realistic results. Here, a loss in performance is expected because the snapshots are collected while the channel varies, thus it doesn't affect all of them in the same way. This goes again the fact that to supply a better covariance matrix accuracy, few snapshots are not enough. Briefly, the results that are going to be compared are:

- *forward* vs. *forward-backward* approaches (fixed channel, algorithm and K)
- *Threshold* vs. *MDL* algorithms (fixed channel, approach and K)
- $K = 1$ vs. $K = 4$ vs. $K = 40$ (fixed channel, approach, and algorithm)
- *Threshold* vs. *MDL* algorithms (fixed approach)

Where K is the number of stored snapshots which is proportional to the estimation time. The assigned values are not casual: $K = 40$ and $K = 4$ correspond respectively to the number of OFDM pilot symbols contained in a whole OFDM frame, i.e., in 10 ms, and in a subframe, i.e., in 1 ms, while $K = 1$ is just the current snapshot, for which no storage memory is required. Actually the second and the fourth points arouse more interest, due to the different complexity of the two algorithms. If they lead to similar performances indeed, the choice would point out the less complex. Furthermore the third will show another important capability that one would like an estimation algorithm to have, i.e., to yield a correct estimation, despite a small amount of collected snapshots. This is also important due to the fact that the channel changes over time and if the snapshots acquisition time is too high, the channel would affect these differently.

7.1 Metrics Definition

In order to evaluate the estimation performance, the two following metrics have been proposed: the *Error Rate* and the *Difference Error*. The first wants to show a qualitative estimation behavior, function of both the window size $M1$ and SNR, through a rate of incorrect detection. Then the second details such behavior, displaying the difference in terms of number of delays, between the estimate and the real value. Supposing that the estimated number of delays for a fixed window is $\hat{L}$ and defining the number of transmission mimics TX (see appendix A), the error parameter ψ_t, for $1 \leq t \leq TX$, can be defined as follows:

$$\psi_t = \begin{cases} 1 & \text{if } \hat{L} = L \\ 0 & \text{otherwise} \end{cases} . \tag{7.1}$$

Thus, the above metrics are stated:

$$E_r = 1 - \frac{1}{TX} \sum_{t=1}^{TX} \psi_t \tag{7.2}$$

and

$$D_e = \frac{1}{TX} \sum_{t=1}^{TX} \left(\hat{L} - L \right) . \tag{7.3}$$

While E_r doesn't present ambiguities, D_e must be taken with care when the respective *Error Rate* is not equal to zero: the difference without the absolute value indeed, keeps the information about the error sign, but being the result an average, the opposite effects of an overestimate and an underestimate could compensate. This has to be verified when the *Error Rate* is not low but the *Difference Error* is equal to zero. It's why the curves of such metric have to be shown together with a representation of an *Under-Estimation*

Rate Ue_r and an *Over-Estimation Rate* Oe_r, again function of $M1$ and SNR, in order to see when the algorithm is inclined to estimate more or less echoes, respect to their real number. Like for the *Error Rate*, an error parameter ν_t is defined as:

$$\nu_t = \begin{cases} 1 & \text{if } \hat{L} < L \\ 0 & \text{otherwise} \end{cases} \tag{7.4}$$

and immediately, the *Under-Estimation Rate* and the *over-Estimation Rate* follow:

$$Ue_r = \frac{1}{TX} \sum_{t=1}^{TX} \nu_t, \tag{7.5}$$

$$Ue_r = E_r - Ue_r. \tag{7.6}$$

In case of *Error Rate* equal or close to zero conversely, such representations can be avoided because if the detection never or rarely fails, the above compensation can't arise.

7.2 Threshold Justification

Before starting to compare several configurations, a *Threshold* selection based on simulation results is reported. Here a deep investigation on the optimum value hasn't been performed. One thinks that it was enough to start from a rougher value, to see if the working principle is able to produce acceptable results. Then in case of success, a fine study on the optimum can be carried out. From the preceding arguments should be clear that in theory the noise eigenvalues are all equal to the noise variance, i.e., σ^2, but the estimates are just clustered around it. Now, a *Threshold* value equal to σ^2 has already been discarded in section 6.1, so it seems reasonable to proceed by trials with a step of 0.5. Three results concerning the *Threshold* equal to $1.5\sigma^2$, $2\sigma^2$, and $2.5\sigma^2$ follow.

$2\sigma^2$ seems the best compromise between the *Error Rate* and the range over which it keeps low. For $2.5\sigma^2$ the "zero error" zone is slightly wider but only for high SNRs; when it starts to decrease $2\sigma^2$ yields absolutely better performance. $1.5\sigma^2$ denote the worst behavior: in any case the working range is narrower than the already mentioned thresholds and the lowest *Error Rate* peaks are not small enough compared to the others. Thus, safely discarding this last value, the *Threshold*, assumed to split the signal and the noise subspaces, is $2\sigma^2$. This will be used throughout the rest of the chapter.

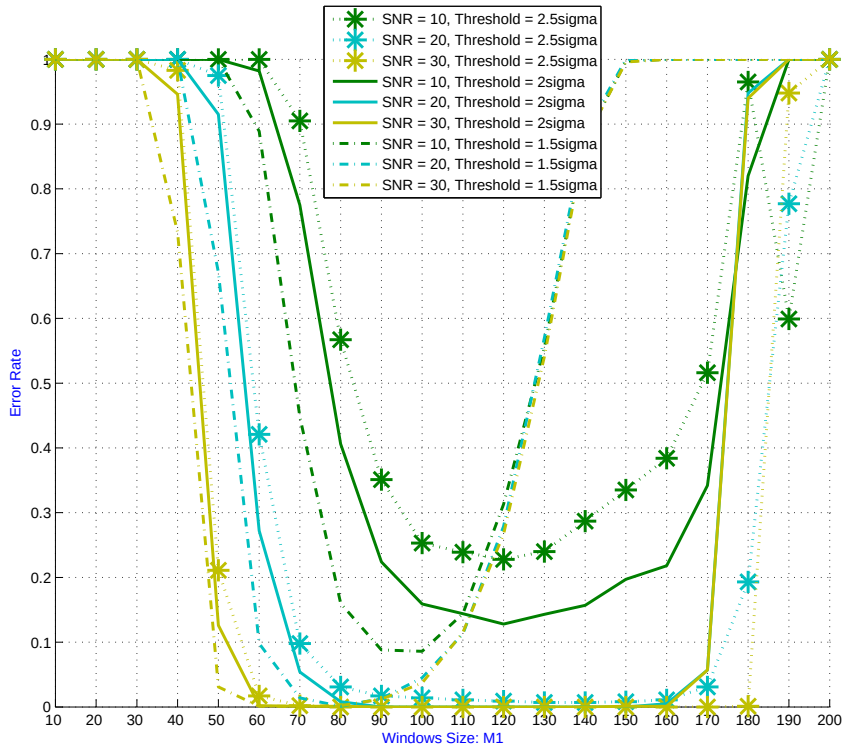

FIGURE 7.1: *Error Rate* plot using forward approach, with $K = 40$.

7.3 Static Channel

The following simulations have been attained initializing the channel delays before the OFDM transmission starts. Thus during the transmission, they stay fixed both in number and in value. The only parameters, that get updated every time that an OFDM symbol occurs, are the echoes complex amplitudes, by a function that simulates their behavior, assumed as random, developed by [3] and [4]. The first result is about the supposed estimation capability improvement of the *forward-backward* approach. Then the one that will yield a worse *Error Rate* will be discarded. Once the approach has been selected, the competing algorithms will be compared, then the analysis concerning the number of stored snapshots will follow.

7.3.1 Forward vs. Forward-Backward

Since the aim of this subsection is just to evaluate if the error of correct detection can be decreased employing the forward-backward approach, it's not of interest to deeply analyze its behaviors for a wide range of SNRs and snapshots number. It's why just the comparison between the two approaches, obtained by $SNR = 10dB$ and $SNR = 25dB$ will be reported, with $K = 40$.

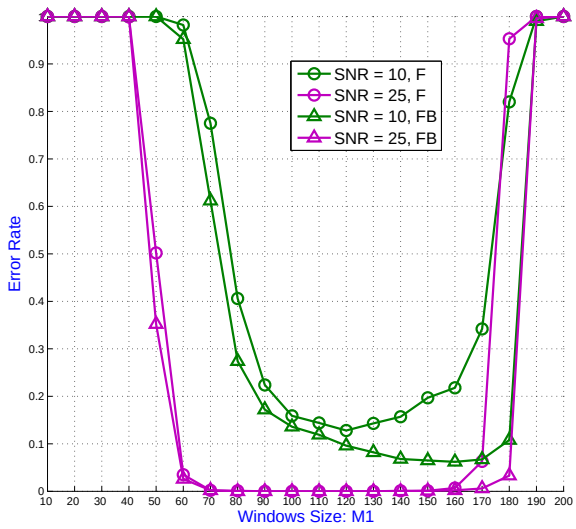

FIGURE 7.2: *Error Rate* plot using forward and forward-backward approach, with $K = 40$.

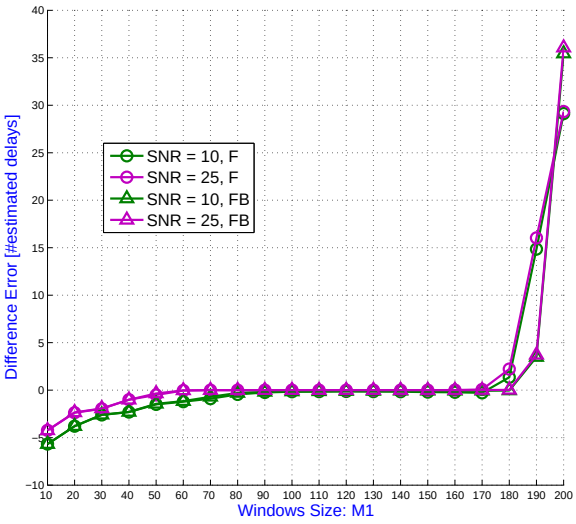

FIGURE 7.3: *Difference Error* plot using forward and forward-backward approach, with $K = 40$.

The *Error Rate* plotted in figure 7.2 shows that for both the SNRs and for all the window sizes, the forward-backward approach has better performances. However the *Difference Error* in figure 7.3, doesn't emphasize this difference since the curves are practically superimposed over whole the windows size range. This means that the average estimation errors are very similar, but that the rate of correct detection is better in the first case. Another consideration can be made about the SNR within a defined approach: this doesn't really affect the performance, since between $SNR = 10dB$ and $SNR = 25dB$ the average improvement is smaller than one delay almost for all the window sizes.

Hence, the forward approach will be discarded due to its worse performance. In order to allow a better comprehension, this variable won't be specified anymore. All the following results indeed, are obtained from the eigenvalue decomposition, forcing the covariance matrix to be centrosymmetric.

7.3.2 Threshold vs. MDL

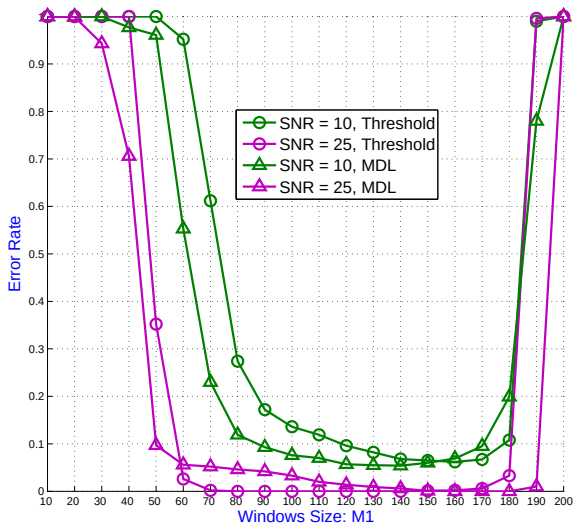

FIGURE 7.4: *Error Rate* plot using *Threshold* and MDL, with $K = 40$.

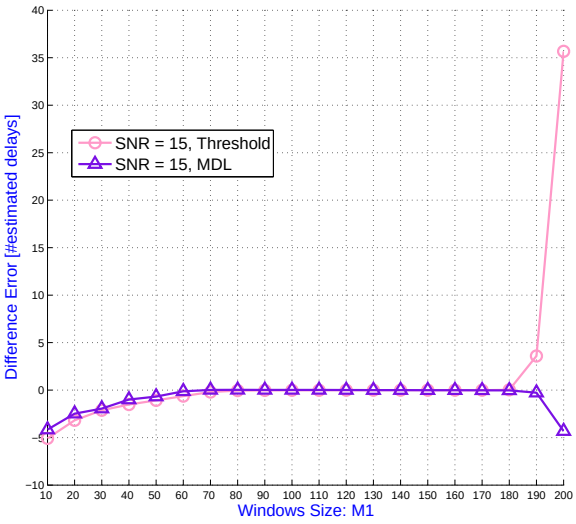

FIGURE 7.5: *Difference Error* plot using *Threshold* and MDL, with $K = 40$ and $SNR = 15dB$.

Here MDL is introduced and compared with the *Threshold*. The two aspects to be examined in order to determine if and when one of the candidates can be regarded as

better, are the behavior within the window size and the number of stored snapshots K. By plot 7.4 a different behavior between the two different SNRs can be observed: although the range over which an acceptable error is attained is wider for MDL for both SNRs, it doesn't present a "zero error" zone like the *Threshold* one for the higher one. However for $SNR = 10dB$ the *Error Rate* is lower for MDL. In plot 7.5 only one SNR value has been reported for clarity, because as in the previous subsection, it doesn't considerably affect the error. It shows that MDL presents a slightly smaller error at each windows, while a large improvement occurs in absence of smoothing.

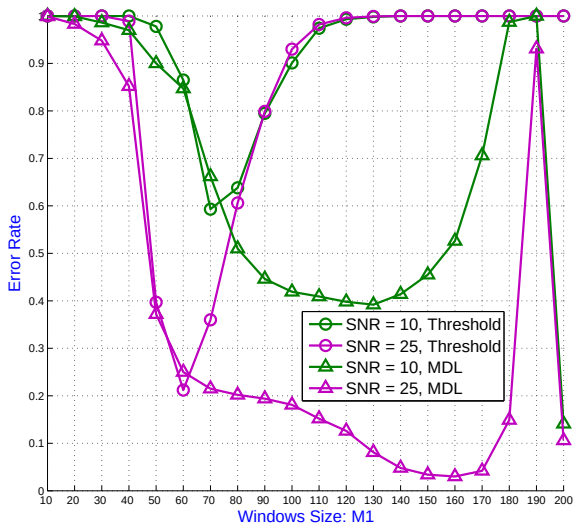

FIGURE 7.6: *Error Rate* plot using *Threshold* and MDL, with $K = 4$.

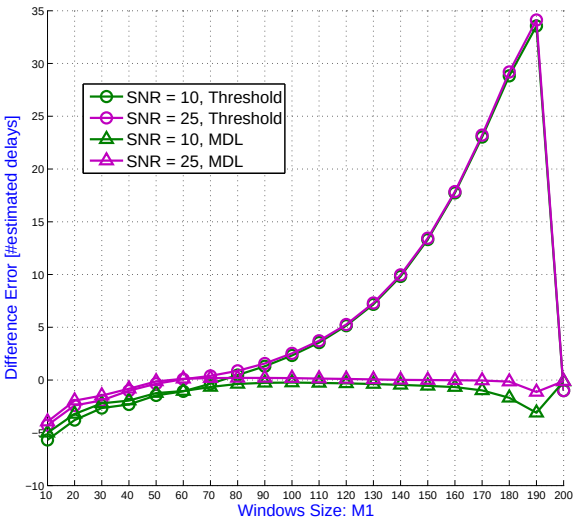

FIGURE 7.7: *Difference Error* plot using *Threshold* and MDL, with $K = 4$.

A performance reduction is noticed by figure 7.6, for both the *Threshold* and MDL, when $K = 4$, but it's the first one that suffers the most from the drastic decrement in

snapshots. The MDL error, on the other hand, is still acceptable.

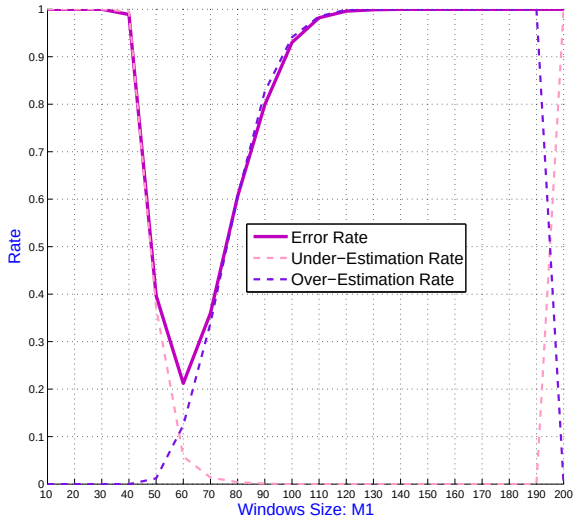

FIGURE 7.8: *Uestimation Rate*, *Oestimation Rate* and *Error Rate* plot using *Threshold*, with $K = 4$ and $SNR = 10dB$.

Plot 7.7 shows that the lowest *Error Rate*, corresponding to a window size equal to 60, yields an average error equal to zero. This anyway is also the result of compensation: figure 7.8 shows that for $M1 = 60$ the Under/Over estimation probability, though small, is almost the same and it helps to provide *Difference Error* $= 0$. Thus, this makes the *Threshold* unreliable with a small number of snapshots.

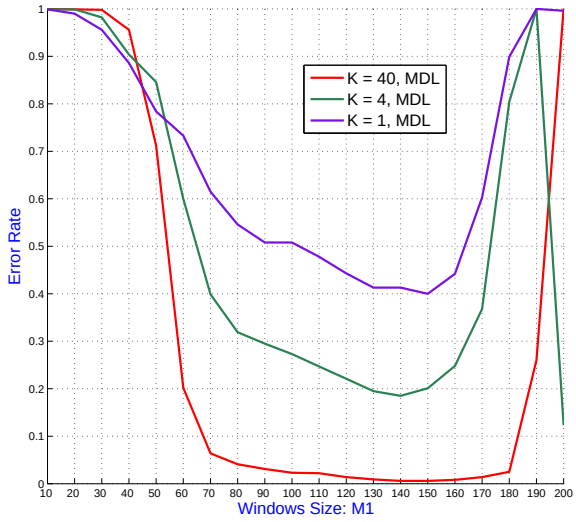

FIGURE 7.9: *Error Rate* plot using MDL and $SNR = 15dB$.

Now that a *Threshold* limit has been shown, there are no reason to report its results for $K = 1$. Since the covariance matrix approximation relies on the number of snapshots,

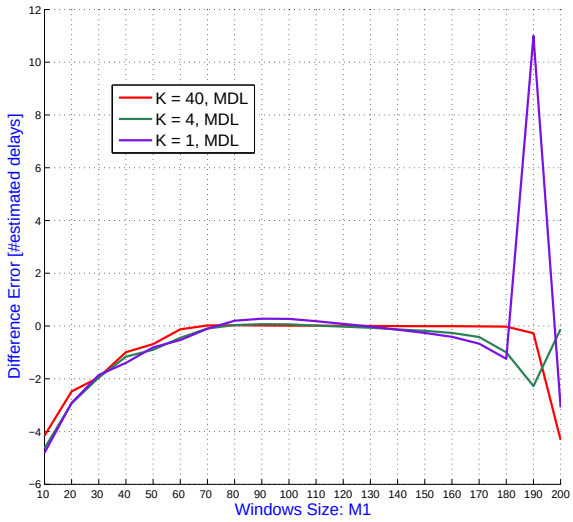

FIGURE 7.10: *Difference Error* plot using MDL and $SNR = 15dB$.

with a lower number not any kind of improving is expected. On the other hand, it's interesting to see how the things evolve for MDL. To see this, the results obtained using a different snapshots number, will be compared, i.e., $K = 40$, $K = 1$ and $K = 1$. Although the *Error Rate* denoted that when K decreases the performances gets worse, the *Difference Error* shows that it doesn't involve the magnitude of the error, in a critical way. This stays within one delay, in the window size range that presents the lowest *Error Rate*, indeed. Figure 7.11 shows that it's not a compensation effect, for $K = 1$: in correspondence of the lowest peak, i.e., $M1 = 150$ indeed, *Uestimation Rate* $\neq$ *Oestimation Rate*, that means that in this case the *Difference Error* average is reliable.

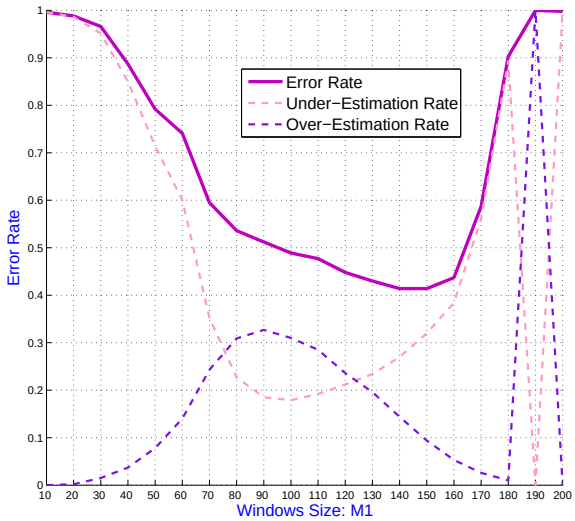

FIGURE 7.11: *Uestimation Rate*, *Oestimation Rate* and *Error Rate* plot using MDL, with $K = 1$ and $SNR = 15dB$.

The presented results, show that in the static channel the proposed *Threshold* can be employed just if a significant number of snapshots is collected. In fact for certain window sizes it yields even better performance than MDL for high SNR and moreover its complexity is definitely lower. Again, with $K = 4$ and $K = 1$, the error is not tolerable, while MDL doesn't present such degradation.

7.4 Dynamic Chanel

At this point, a time-varying channel, which implementation's details are explained in [3], is introduced. Beyond the channel initialization before starting the transmission like in the static case, for each OFDM symbol, an appropriate function checks if any new delay is born, if any old delay is dead and update their values. So as, according to the birth and the death delay rates, the channel will vary within the transmission. It's why the capability to perform a good estimation even with a small number of snapshots would be really valued.

7.4.1 Threshold vs. MDL

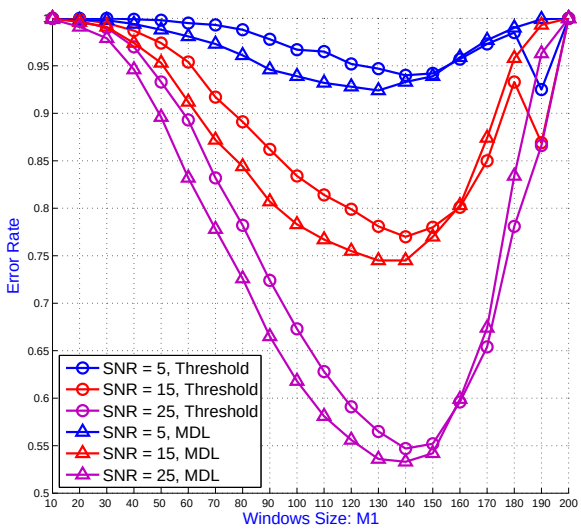

FIGURE 7.12: *Error Rate* plot using *Threshold* MDL, with $K = 40$.

Here the situation is clearly worse, since it yields an *Error Rate*, neither comparable to that obtained by MDL over the static channel with $K = 1$ (figure 7.15). At least plot 7.14 shows for $SNR = 15dB$ a compensation never occurs, being the *Oestimation Rate* close to zero almost everywhere, while the *Uestimation Rate* is very similar to the *Error Rate*, which means that such configuration tends to Underestimate the number

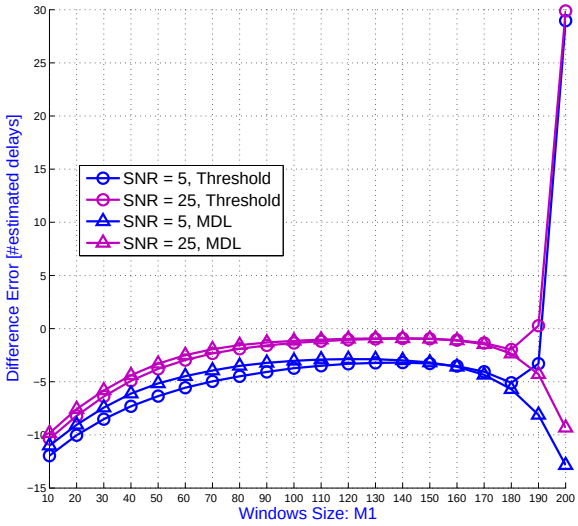

FIGURE 7.13: *Difference Error* plot using *Threshold* and MDL, with $K = 40$.

of delays. Like in the static setup, the SNR doesn't affect the average error too much, it's just larger, especially in those zones where the performance are worse, i.e., when the window is either big or small.

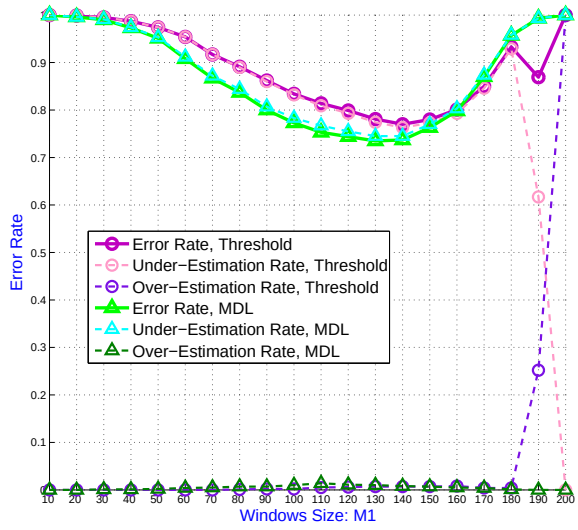

FIGURE 7.14: *Uestimation Rate*, *Oestimation Rate* and *Error Rate* plot using *Threshold* and MDL, with $K = 40$ and $SNR = 15dB$.

Although the above plots show that even with $K = 40$ both the algorithms yield inadequate results, a further degradation is shown setting $K = 4$. The *Error rate* curves (figure 7.16) shapes of the respective algorithms are really similar to those of plot 7.6, obtained for the static channel with $K = 4$. The problem is that they are shifted upward by a rate of about 0.6. The same holds for the *Difference Error* shapes, but not for the

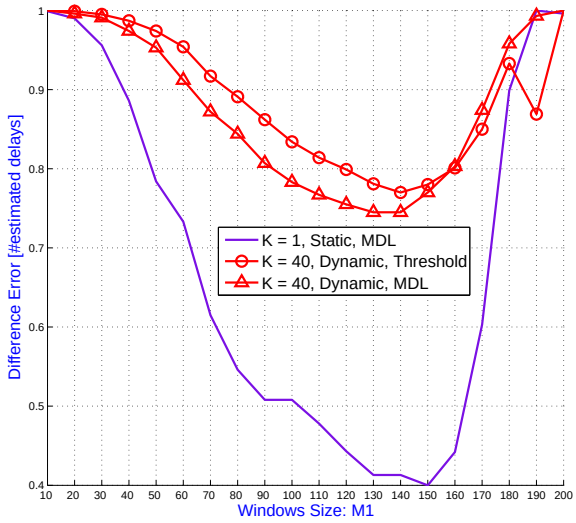

FIGURE 7.15: *Error Rate* plot using *Threshold* and MDL over dynamic channel, with $K = 40$ and MDL over static channel, with $K = 1$ and $SNR = 15dB$.

values: plot 7.17 shows that MDL keep underestimating the number of delays but by a constant factor of about 1 in the central window size range for $SNR = 25dB$, while the *Threshold* tends to overestimate increasingly with the window size until $M1 = 190$.

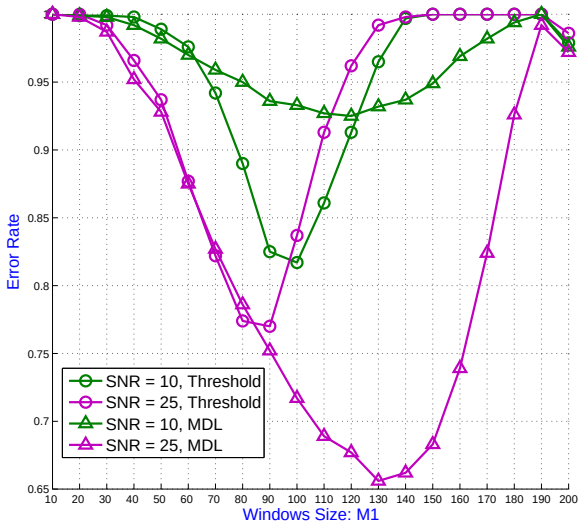

FIGURE 7.16: *Error Rate* plot using *Threshold* and MDL, with $K = 4$.

Looking at the last results with $K = 4$, those with $K = 1$ can be safely avoided, since the performance are quite mediocre, also with a large number of snapshots. As well as, the comparison between the static and the dynamic channel within the same K would be useless, since plot 7.15 shows that MDL with $K = 1$ over the static channel widely outclass both *Threshold* and MDL, with $K = 40$ over the dynamic channel.

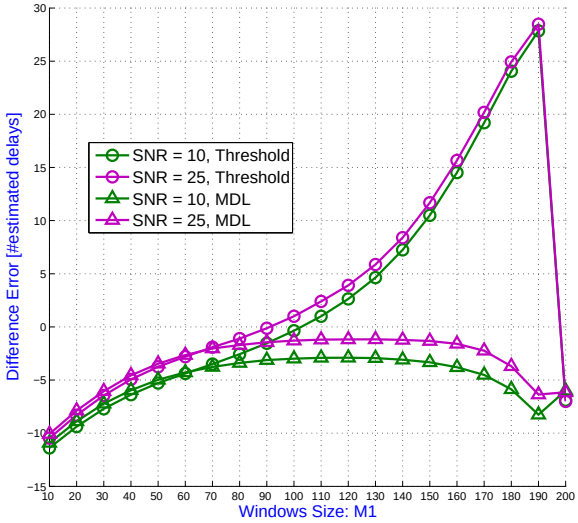

FIGURE 7.17: *Difference Error* plot using *Threshold* and MDL, with $K = 4$.

Chapter 8

Conclusions

This thesis focuses on improving of the existing channel estimation algorithms, based on a parametric channel model. It has been shown in [10] and [18] that a promising tap-delay channel estimator is the ESPRIT (or its reduced complexity version, i.e., the Unitary ESPRIT [22]), that requires the knowledge of the number of delays. For this purpose, the state-of-the-art MDL detection scheme has been investigated and compared with a trivial algorithm that simply selects the signal eigenvalues by a threshold operation. Furthermore, forward-backward averaging and spatial smoothing have been employed in order to improve the estimation accuracy. The study concerns both a static and a dynamic channel, over which the above algorithms have been compared, varying the number of time observations.

The results show that the forward-backward approach, if combined with spatial smoothing, really helps to improve the covariance matrix estimate, and that such gain holds for every window size.

An initial comparison over the static channel between MDL and the *Threshold*, using an entire OFDM frame of pilots observations, seems not justifying the use of the first as estimator, since its complexity only leads to a small performance improvement. However when the observations reduce to an OFDM subframe, the *Threshold* fails in robustness, while MDL keeps holding about the same estimation error. For the extreme case in which only one observation is exploited, the rate of correct detection decreases. However, the average error doesn't explode as for the *Threshold*.

On the other hand, over the dynamic channel, the *Error Rate* considerably degrades for both the algorithms, either with a large number of observations. They don't present great differences, except for the case without smoothing (the *Threshold* always highly overestimates). However, the average error is not low enough to immediately reject both

algorithms. This remark doesn't hold reducing the time observations, since also MDL, that worked over the static channel, starts to underestimate too much.

Regarding the preprocessing scheme, it has been shown that forcing the covariance matrix to be centrosymmetric, jointly with the spatial smoothing, really helps the estimation, even if the window size that leads to the best performances still has to be determined empirically. About the number of delays estimator, the *Threshold* can't be employed due to the limits revealed with a low number of time observations. Contrariwise, MDL exhibits a certain robustness, but over the dynamic setup, i.e., a channel that better approximates the real case, it still presents some problems.

8.1 Future Work

The problems noticed for both the algorithms lead to different suggestions regarding possible future studies. A fixed threshold is unusable due to the eigenvalues deviation, since it depends on the number of snapshots under certain conditions. Thus, to define a variable threshold, an investigation on its value related to the spatial smoothing and to the centrosymmetric forcing, could lead to an improved robustness. Differently, MDL is strongly consolidated but over the dynamic channel, it seems somehow biased, thus a study on a possible correction term could remove the error. Since the actual correction term is different according to the employed approach (forward or forward-backward), it could be related to the spatial smoothing parameters as well, condition that hasn't been taken into account. Finally, it could be interesting to test the implemented algorithms, in a complete estimation setup, in order to see the performance also in terms of BER.

Appendix A

Simulator Setup

This appendix wants to detail the simulator's implementation aspects, in order to provide a sort guide for future works, that is a reviewed version of the one used in [3]. The implementations that have been kept are the base OFDM transceiver structure and the channel initialization and update functions, while the whole part regarding the equalization has been removed in order to reduce the complexity and improve the simulations speed. The implemented updates are the *Threshold* and the MDL algorithms and the subsequent metrics, useful to measure the estimation error, described in chapter 7. The following sections describe respectively the used parameters and the simulator structure in terms of pseudo-code.

A.1 Parameters Specification

Assuming that all the parameters are tunable for future works, some of them are kept fixed due to the LTE setup and the multipath properties, e.g., the FFT/IFFT size and the maximum Doppler frequency, while others have been modified, e.g., the number of stored snapshots, in order to observe different estimation behaviors. To resume the parameters of table A.1, follows a list that group them in categories.

- **LTE Specifications:** all the parameters are obvious except M, CP and *LTE Type,* that are respectively the *number of pilots*, obtained from N_u and the *pilot spacing*, the *cyclic prefix* expressed in number of samples and LTE *Extended Vehicular A* delay profile.
- **Tunable:** K is the *number of snapshots* stored during the transmission, in order to obtain the sample covariance matrix; FB is a flag whose values 0 and 1 correspond respectively to the *forward* or the *forward-backward* approach.

LTE Specifications	Tunable	Technical	Multipath
$N = 2048$ $N_u = 1200$ $M = 200$ $CP = 144$ $f_c = 2 \times 10^9$ pilot spacing = 6 pilot indent = 1 LTE Type = 2 (EVA)	$K = [40 \quad 4 \quad 1]$ $FB = [0 \quad 1]$	$TX = 1000$ $S = 20$ $R = 100$	$f_D = 10$ $L_a = 15$ $lambdaS = 1$ $[a \quad d] = [10 \quad 88]$ $rho = 20$ $sd = 10$

TABLE A.1: Simulations Parameters Specification

- **Technical:** TX is the number of *transmission mimics*; S is the number of orders tried in the MDL detection scheme; R is the number of components, averaged to induce the random phase modulation in each channel echo.
- **Multipath:** f_D is the maximum Doppler frequency; L_a is the average number of delays; $lambdaS$ is the average lifespan of each tap in seconds; $[a \quad d]$ is the interval within the delays are generated; rho is the decay rate (the last two parameters have been chosen in order to fit the EVA LTE power delay profile, see [3] for more details); sd is the standard deviation of the Gaussian angle spread (except the first, all the list parameters are used only in the dynamic channel case, by the functions that initialize and update the delays).

A.2 Simulation Structure

The simulator reflects the LTE OFDM downlink scheme: assuming as parameters, the LTE specifications detailed in section 2.3, it randomly generates the information bits, modulates and transmits them over the channel, generates the noise according to the SNR and at the "receiver side" store the pilot observations to proceed then with the estimation. The set of pilots is placed according to (2.30), the parameters and the storage matrices are initialized, then the transmission starts, considering several SNRs. The outermost loop is exactly over different SNR values [5 10 15 20 25 30], the second one is over the transmission mimics, necessary due to the random nature of the signal, then two consecutive inner loops are run respectively over the OFDM symbols and the window size. Resuming, for each SNR, an OFDM transmission is performed, followed by the proper estimation of the number of delays, using different window sizes to smooth the obtained covariance matrix, which eigenvalue decomposition feds the

estimation algorithms. Then before swapping to the next SNR, the metrics used to evaluate the performance are calculated. The above setup is the same for both the static and the dynamic channel, with the only difference that in the latter case, the delays' number and value are updated at each pilot OFDM symbol, by the proper function developed in [3]. Here follows the scheme that describes the simulator's main steps (the static and the dynamic differences will be emphasized):

- parameters definition
- pilots placing
- **T** matrix building (*static, dynamic*)

SNR [5 10 15 20 25 30]

Mimics [1 : TX]

- channel initialization

OFDM Transmission [1 : $OFDMsymbols$]

- pilot OFDM symbol check
- random bit generation
- 4PSK modulation, $\mathbf{X}$
- noise generation, $\boldsymbol{v}$
- channel update (*dynamic*)
- **T** matrix building (*dynamic*)
- delays' complex amplitude update, $\boldsymbol{\alpha}$
- transfer function calculation, $\boldsymbol{h} = \mathbf{T}\boldsymbol{\alpha}$
- received symbol calculation, $\boldsymbol{y} = \mathbf{X}\boldsymbol{h} + \boldsymbol{v}$
- pilots LS estimate, $\hat{\boldsymbol{h}}_p^{LS} = \mathbf{X}_p^{-1}\boldsymbol{y}_p$
- snapshots matrix update

- sample covariance matrix calculation, $\hat{\mathbf{R}}_{yy} = \frac{1}{K}\mathbf{Y}\mathbf{Y}^{\mathsf{H}}$

Window size [10 20 $\cdots$ M]

- covariance matrix smoothing, $\hat{\bar{\mathbf{R}}}_{yy} = \frac{1}{K \times K1}\mathbf{Y_{M1}}\mathbf{Y_{M1}^{H}}$
- centrosymmetric enforcing (if FB = 1)
- eigenvalue decomposition
- *MDL* estimation
- *Threshold* estimation
- partial error metrics update

- error metrics calculation

A.3 Error Matrices

In order to access the already obtained data and the future ones, an outline on the errors storage matrices follows. The same 3D structure is the same for each matrix and it's depicted in figure A.1, though.

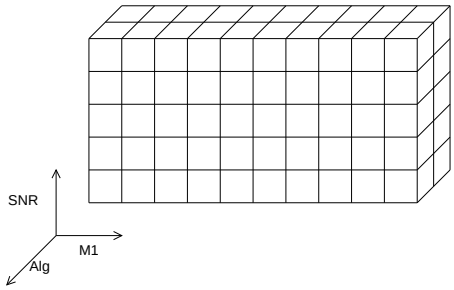

FIGURE A.1: Error matrices' structure.

Each row represents a different SNR, each column a different window size while the third dimension contains the different implemented algorithm. Thus if the matrix is read row-wise the behavior will be function of the window size and each curve will correspond to a different SNR; the opposite, if it's read column-wise.

Bibliography

[1] E. Dahlman, S. Parkvall, J. Skold, and P. Beming, *3G Evolution: HSPA and LTE for Mobile Broadband*. Academic Press, 2007.

[2] "Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation (release 8)," Tech. Rep. TS 36.211, V8.8.0, 3rd Generation Partnership Project, Sep 2009.

[3] M. L. Jakobsen, "Modeling and Estimation of Wireless Multipath Channels - An Application within Pilot-assisted Channel Estimation for Downlink OFDM," Master's thesis, Aalborg University.

[4] K. Laugesen, "Tap-Delay Estimation for the Next-Generation of Mobile Terminals," Master's thesis, Aalborg University.

[5] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing*. Prentice Hall, 1999.

[6] J.-J. van de Beek, O. Edfors, M. Sandell, S. Wilson, and P. Borjesson, "On Channel Estimation in OFDM Systems," *In Preceedings of IEEE Vehicular Technology Conference*, vol. 2, Jul 1995.

[7] "Evolved Universal Terrestrial Radio Access (E-UTRA); Base Station (BS) Radio Transmission and Reception (release 8)," Tech. Rep. TS 36.104, V8.4.0, 3rd Generation Partnership Project, Dec 2008.

[8] "Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) radio transmission and reception (release 8)," Tech. Rep. TS 36.101, V8.7.0, 3rd Generation Partnership Project, Sep 2009.

[9] "Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) radio transmission and reception (release 8)," Tech. Rep. TS 36.803, V1.1.0, 3rd Generation Partnership Project, Apr 2008.

[10] B. Yang, K. B. Letaief, R. S. Cheng, and Z. Cao, "Channel Estimation for OFDM Transmission in Multipath Fading Channels Based on Parametric Channel Modeling," *IEEE Transactions on Communications*, vol. 49, Mar 2001.

[11] R. Roy and T. Kailath, "ESPRIT-Estimation of Signal Parameters Via Rotational Invariance Techniques," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, Jul 1989.

[12] Y. Li, L. J. Cimini, Jr., and N. R. Sollenbeg, "Robust Channel Estimation for OFDM Systems with Rapid Dispersive Fading Channels," *IEEE Transactions on Communications*, vol. 46, Jul 1998.

[13] L. L. Scharf, *Statistical Signal Processing.* Addison-Wesley, 1991.

[14] O. Edfors, M. Sandell, J.-J. van de Beek, S. K. Wilson, and P. O. Borjesson, "OFDM Channel Estimation by Singular Value Decomposition," *IEEE Transactions on Communications*, vol. 46, Jul 1998.

[15] P. Stoica and R. Moses, *Spectral Analysis of Signals.* Prentice Hall, 2005.

[16] T.-J. Shan, M. Wax, and T. Kailath, "On Spatial Smoothing for Direction-of-Arrival Estimation of Coherent Signals," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 33, Aug 1985.

[17] H. Krim and M. Viberg, "Two Decades of Array Signal Processing Research: the Parametric Approach," *IEEE Signal Processing Magazine*, vol. 13, Jul 1996.

[18] M. L. Jackobsen, K. Laugesen, C. Navarro, G. E. Kirkelund, C. Rom, and B. H. Fleury, "Parametric Modeling and Pilot-Aided Estimation od the Wireless Multipath Channel in OFDM Systems." IEEE International Conference on Communications, 2010.

[19] G. Xu, R. H. Roy, III, and T. Kailath, "Detection of Number of Sources Via Exploitation of Centro-Symmetry Property," *Transactions on Signal Processing, IEEE*, vol. 42, Jan 1994.

[20] T. W. Anderson, "Asymptotic Theory for Principal Component Analysis," *The Annals of Mathematical Statistics*, vol. 34, no. 1, 1963.

[21] M. Wax and T. Kailath, "Detection of Signals by Information Theoretic Criteria," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 33, Apr 1985.

[22] N. J. Haardt, M., "Unitary ESPRIT: How to Obtain Increased Estimation Accuracy with a Reduced Computational Burden," *IEEE Transactions on Signal Processing*, vol. 43, May 1995.