Escola d'Enginyeria de Telecomunicació i Aeroespacial de Castelldefels
UNIVERSITAT POLITÈCNICA DE CATALUNYA

# MASTER THESIS

**TITLE: Reconfigurable OFDM Tx/Rx for multi-purpose physical layers over fast Heterogeneous Systems**

**MASTER DEGREE: Telecommunications engineering**

**AUTHOR: Adam Andreu Cresswell Baldellou**

**DIRECTOR: Joan Bas and Ciprian Gavrincea.**

**SUPERVISOR: Pere L. Gilabert**

**DATE: May 23rd 2013**

**Title:** Reconfigurable OFDM Tx/Rx for multi-purpose physical layers over fast Heterogeneous Systems

**Author:** Adam Cresswell Baldellou

**Director:** Joan Bas and Ciprian Gavrincea.

**Supervisor:** Pere L. Gilabert

**Date:** May 23rd 2013

## Overview

The main goals of this project are the study from a mathematical and practical point of views of different correlated Rayleigh fading generator, linear channel generator estimators, and strategies for implementing them in a reconfigurable way by means of Simulink and FPGA. Thus it is not necessary to remap again the FPGA for different configurations of the channel estimation.

The techniques of correlated Rayleigh fading that have been analysed under this project are the next: the sum of sinusoids, the Smith technique and the Bealieu-Young method. The first one consists on a sum of exponentials whereas the other two algorithms are based on IDFT's decompositions. From all these methods we show that the Bealieu-Young's channel generation requires the lowest complexity for the same accuracy. Consequently, this method of channel estimation is the technique that has been implemented into the FPGA.

Regarding to the channel estimation process two different approaches has been considered. The first one consists on spreading the pilots uniformly along the frames, whereas the other one places the pilots at the beginning of the frames. Next, in order to estimate the channel two strategies have been analysed: the averaging channel estimator and the linear interpolation technique. The averaging strategy can be considered as a low pass filter. So, it offers the best results when there channel is flat and the noise is high. In this case, the averaging filter is able to remove the high frequencies that introduce the noise signal. On the contrary, the linear interpolator gives the best performance when the noise level is small. Moreover, it allows a degree of mobility, which limits the distance between the pilots. In both cases we have computed the theoretical error power of the channel estimators. The practical results from Matlab and Simulink perfectly match with the theoretical ones.

In a co-simulation environment the FPGA works with Matlab. The results show that the speed of the simulation is limited by the wire that connects the FPGA and Matlab. So, it is crucial for a hardware accelerator to try to integrate the full system into the FPGA and if it is not possible to use high-speed links between the FPGA and Matlab.

**Título:** Reconfigurable OFDM Tx/Rx for multi-purpose physical layers over fast Heterogeneous Systems

**Autor:** Adam Cresswell Baldellou

**Director:** Joan Bas and Ciprian Gavrincea.

**Supervisor:** Pere L. Gilabert

**Data:** 23 de Mayo de 2013

## Resumen

Los objetivos principales en este proyecto son los análisis matemáticos y prácticos de diferentes generadores de variables Rayleigh, estimadores de canal y estrategias para su implementación de una manera reconfigurable a través de Simulink y FPGA.

La diferentes técnicas para generar variables Rayleigh usadas en este proyecto son las siguientes: la suma de sinusoides, la técnica de Smith y el método de Bealieu-Young. El primero se base en la suma de exponenciales complejas mientras los otros dos se basan en la IDFT. De todos estos métodos vemos que el método de Bealieu-Young tiene la mínima complejidad para la misma precisión que los otros métodos. En consecuencia ha sido el escogido para ser implementado en la FPGA

Cuando hablamos de estimadores de canal se han considerado dos enfoques diferentes. El primero consiste en introducir los pilotos de manera uniforme a lo largo de la trama, mientras el otro introduce los pilotos al principio de la trama. A la hora de estimar la trama dos estrategias han sido analizadas: el estimador de canal promedio y el estimador de canal por interpolación lineal. La técnica de promediado se puede considerar como un filtro paso bajo. En este caso el estimador por promediado es capaz de eliminar la frecuencias altas que son introducidas por el ruido. Por el contrario el estimador por interpolación lineal, ofrece los mejores resultados cuando el ruido es bajo. En ambos casos hemos calculado la potencia de error en el proceso de estimación. Los resultados prácticos y teóricos obtenidos a través de Matlab y Simulink concuerdan a la perfección.

En un escenario co-simulado la FPGA trabaja con Matlab. Los resultados nos dicen que el sistema se ve limitado por el cable que conecta ambos sistemas. Por lo que es crucial para aprovechar al máximo la potencia de la FPGA que el sistema si integre por completo dentro de la FPGA y si no es posible usar conexiones de alta velocidad.

# INDEX

# CHAPTER 1. INTRODUCTION

## 1.1 Motivation

The motivation of this project is twofold: a theoretical understanding of a communication channel and its practical implementation in a Field Programmable Gate Array (FPGA). In the former case, there is a review of the main parameters that characterize a communication channel, techniques for channel generation and strategies for channel estimation at the receiver. Moreover, analytical expressions for the average error power for different strategies of pilot positioning are given and compared with the ones obtained via simulation. In the second case, the practical implementation of the channel estimation has been carried out via System Generator, which allows us to connect its domain with the Matlab one. This fact makes that the debugging process of the programs that run on a FPGA be speeded up. In addition, it is also possible to run in the FPGA parts of Matlab code that imply a high computational burden. That it is the case of channel estimation process when the number of subcarriers in an OFDM system is high. That it is called hardware acceleration and it is thought that in the future will increase their use if communication systems augment their complexities (i.e. multicore-systems, adaptive coding and modulation schemes, etc.).

## 1.2 Objectives

The first objective of the project is the study and analyses of different ways of implementing a correlated Rayleigh fading multipath channel. Next, after doing the theoretical studies on channel estimation, the most efficient strategy of channel estimation and channel generation in the sense of lower computational load and error power in the channel estimation process is selected.
The second objective of the project consists on analysing real-time channel estimators using FPGA´s and Simulink implementations. In particular it will evaluate the processing time and the error in the accuracy of the channel estimations when the distance between pilots changes, different values of Doppler frequency and types of channel estimators. Moreover, it is possible to observe that practical results match with the theoretical ones. These practical results are obtained by means of the interaction of Matlab and FPGA domains, which it is a plus in the learning process of telecommunications students.

## 1.3 Composition of the Document

The document consists of 6 chapters. In the first chapter, a communication system is explained making emphasis in multipath channels, the second one different, different methods of creating a Rayleigh correlation Channel are

analysed. In chapter three, different types of channels estimators and pilot positioning are studied and compared. In the next chapter the linear interpolation channel estimator is implemented in Matlab and Simulink to check its behaviour from the practical point of view. Finally, the chapters of connecting the System Generator into the FPGA and the conclusions come.

## 1.4 Communication System

A basic telecommunication system is composed by the following stages: transmitter, channel and receiver (Figure 1.1). The transmitter takes the information s(t) and converts it into a signal x(t). Then, this signal is sent through the communication channel to the receiver where the original information is reconstructed as much as possible, ŝ(t). In particular if we denote by h(t) the impulse response of the channel, the received data, denoted by r(t), will be:

$$\boldsymbol{r(t) = x(t) * h(t)} \tag{1.1}$$

Due to the channel impairments the information sent and received will not be the same. In order to minimize this issue, and make the information more robust and reliable against the channel impairments some components are added to the transmitter and receiver (Figure.1.1).


**Figure 1.1** Communication system

An example of a basic transmitter-receiver communication system in which the signal is protected by a coding scheme is shown in Figure 1.2.

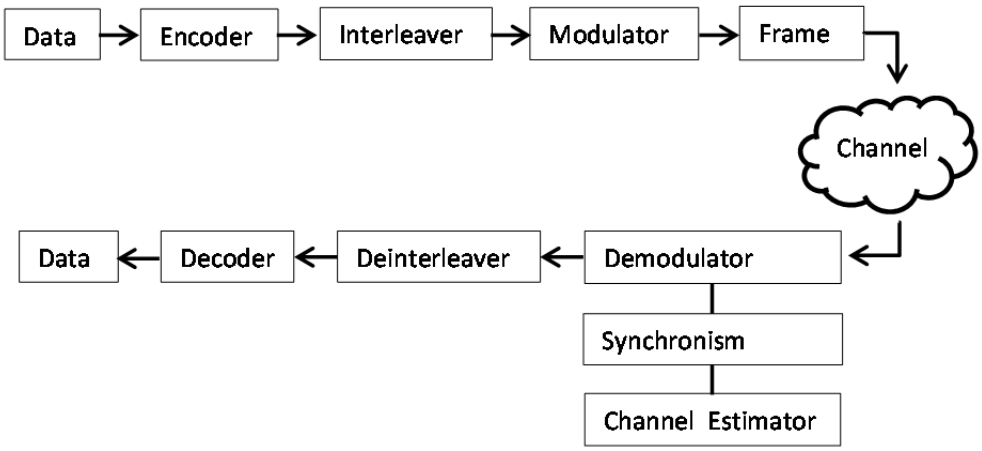


**Figure 1.2** Detailed communication system

Encoding consists in including redundancy in the signal to transmit in order to increase the error correction capability of the receiver. In communication channels the errors typically occur in bursts rather than independently. For that reason an interleaver is used since its main goal is to change the positions of the bits to spread the errors along the codewords. Otherwise the number of errors in the obtained codeword could be too large for the error correction capability of the code.

After encoding and interleaving the coded information the modulator comes. The modulation is the process in which one or more properties of a periodic waveform, called the carrier signal are varied. The three key parameters of a periodic waveform that can be modified are: the amplitude, phase and frequency. The frame generator is a cyclically repeated block that consists of a fixed number of data symbols, zeros and pilots. A pilot is a signal, transmitted over a communications system for supervisory, control, estimation, continuity, synchronization or reference purposes.

Inside the frame there is a sequence of bits or symbols that makes it possible for the receiver to detect the beginning and end of the frame. This is known as synchronism. Deinterleaver and decoder are the inverse processes for the interleaver and encoder respectively.

Finally, in order to increase the amount of information that it is transmitted the signal can be multiplexed to transmit in frequency, time or coding. If the signal is transmitted in different subcarriers at the same time instant, it´s called Frequency Division Multiplexing (FDM). On the contrary in Time Division Multiplexing (TDM) the signal is sent in different time instants with the same frequency. Finally in Coding Division Multiplexing (CDM), the signal is protected with a spreading code and sent to the received in the same time instant and all frequencies (Figure 1.3).

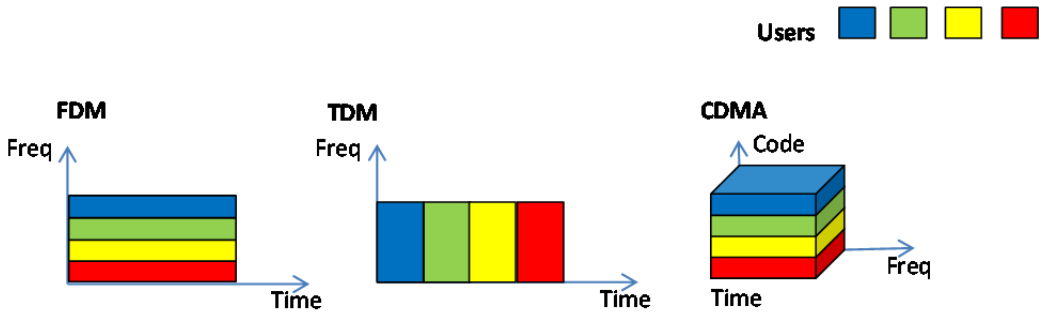


**Figure 1.3** Multiplexing coding schemes

Orthogonal FDM (OFDM) is a technique inside FDM that takes advantage of the orthogonality between subcarriers to overlap them in an efficient way. The maximum of a subcarrier coincides with the zero of the next one (Figure 1.4).

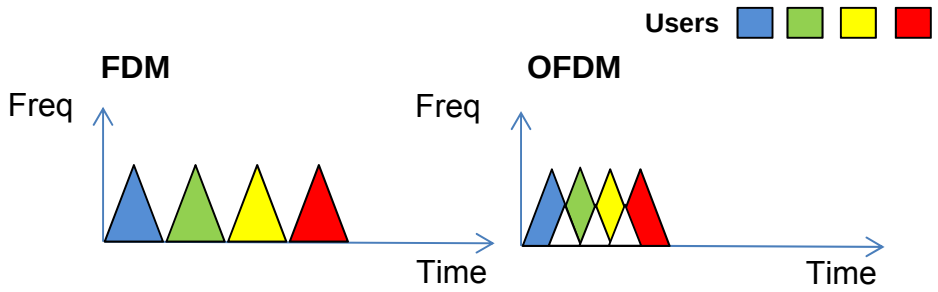


**Figure 1.4** OFDM system

### 1.4.1. Multipath Channel

In a multipath environment, the receiver sees the superposition of multiple copies of the transmitted signal, each travelling in a different path, and having their own attenuation *(α)* and delay *(τ)*.

$$\boldsymbol{r(t)} = \sum_{\boldsymbol{m}}^{\boldsymbol{N}} \boldsymbol{\alpha_m}\, \boldsymbol{x(t-\tau_m)} + \boldsymbol{n(t)} \qquad (1.2)$$

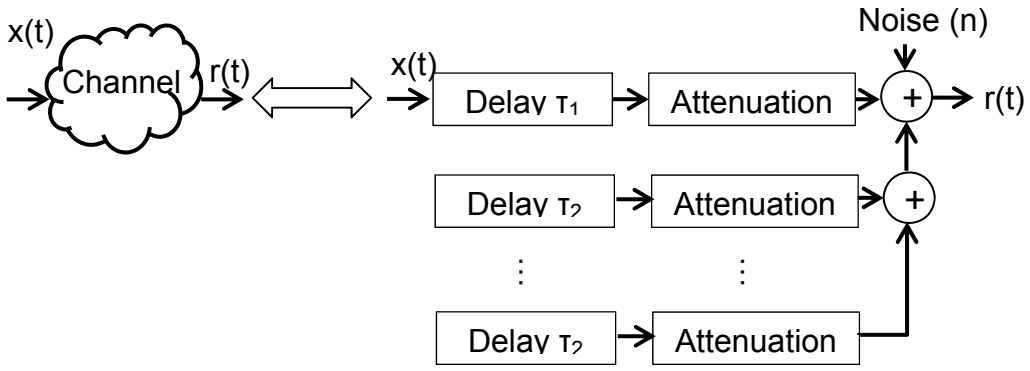


**Figure 1.5** Equivalent scheme of the channel

When two or more versions of the attenuated transmitted signal arrive at the receiver and they are added destructively they cause fading. These multiple versions of the transmitted signal result from the multiple paths present in the channel or from the rapid dynamic changes of the channel.

There are two types of fading: fast-fading and slow-fading. Multi-path fading, so-called fast fading, comes out from the rapid changes of the channel impulse response within the symbol duration. Slow fading also called shadowing, is the result of a slow variation in the channel impulse response within the symbol duration.

A multipath channel is characterized by many important parameters. Among these: Delay spread and coherence bandwidth which help to describe the time-dispersive nature of the channel. On the other hand Doppler spread and coherence bandwidth describe the time-varying nature of the channel.

The Power delay profile also known as multipath intensity profile or delay power spectrum is the average output signal power of the channel as a function of the Excess delay. There are certain parameters that can be determined from the power delay profile. These are the mean Excess delay ($\tau_{mean}$) and the root mean squared ($rms$) delay spread ($\sigma_\tau$). Excess delay is the relative delay of the i-th multipath signal component, compared to the first arriving component and is given by $\tau_{i,}$ whereas delay spread is the deviation in the Excess delay. Their expressions are:

$$\tau_{mean} = \frac{\sum_i P(\tau_i)\tau_i}{\sum_i P(\tau_i)} \tag{1.3}$$

$$\sigma_\tau = \sqrt{mean[(\tau)^2] - \tau_{mean}^2} \tag{1.4}$$

Where

$$mean[(\tau)^2] = \frac{\sum_i P(\tau_i){\tau_i}^2}{\sum_i P(\tau_i)} \tag{1.5}$$

Delay spread is the range of values of excess time delay $\sigma$ over which the Power delay profile is essentially nonzero. The frequency band of the spectral components that conform the transmitted signal when pass through the channel with equal gain and phase is known as coherence bandwidth ($BW_{coh}$), with a frequency correlation of approximately 90%, the $BW_{coh}$ can be approached as

$$\mathrm{BW_{coh}} \approx \frac{1}{50\sigma_\tau} \tag{1.6}$$

Doppler shift is known as the apparent change in frequency of the transmitted signal due to motion, which is given by

$$f_{ds} = \frac{v}{\lambda}\cos\theta \tag{1.7}$$

Where $v$ is the velocity, $\lambda$ is the carrier wavelength, and $\theta$ is the spatial angle between the direction of motion of the mobile and the direction of arrival of the wave. The absolute value of the largest difference in Doppler shifts between different signal components contributing to a single fading channel tap is known as the Doppler spread. If the transmitted signal frequency, $f_c$ is the resultant Doppler spectrum and it is encompassed in the range between $(f_c - f_{d,max})$ and $(f_c + f_{d,max})$, being $f_{d,max}$the maximum Doppler frequency shift. On the other hand the Coherence Time ($T_{coh}$) is the period of time in which the channel impulse response remains invariant [1].

$$T_{coh} \approx \frac{1}{f_{d,max}} \tag{1.8}$$

For the signal processing applications and analyses, the mobile propagation fading channels are modelled statistically. One of the channel models is the Rayleigh fading, which it is used to represent the fading statistics in multipath propagation for channels with No Line Of Sight (NLOS) (See Figure 1.6). This model is useful in scenarios where the signal may be considered scattered between the transmitter and receiver. In this scenario there is no single signal path that dominates and a statistical approach is required to the analysis of the overall nature of the radio communications channel. A Rayleigh distribution can be obtained as the radial component of the sum of two uncorrelated Gaussian random variables.

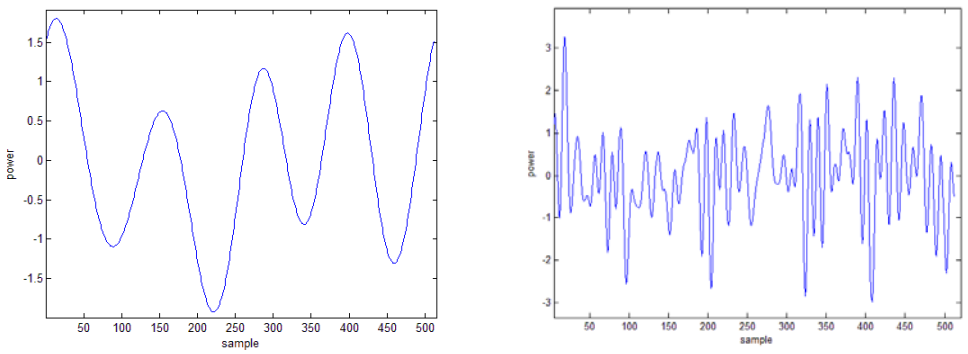


**Figure 1.6** Rayleigh fading for a normalized Doppler frequency of 0.01 Hz (left) and 0.1 Hz (right)

The normalised autocorrelation function of a Rayleigh faded channel when the mobile is moving at a constant velocity is the zero-th Bessel function of the first kind.

$$r[d] = J_0(2\pi\, f_d\, |d|) \tag{1.9}$$

where $d$ is the sample lag and $f_d$the maximum Doppler shift .

At Figure 1.7 we can see how fast the autocorrelation varies depending on the Normalized Doppler frequency.

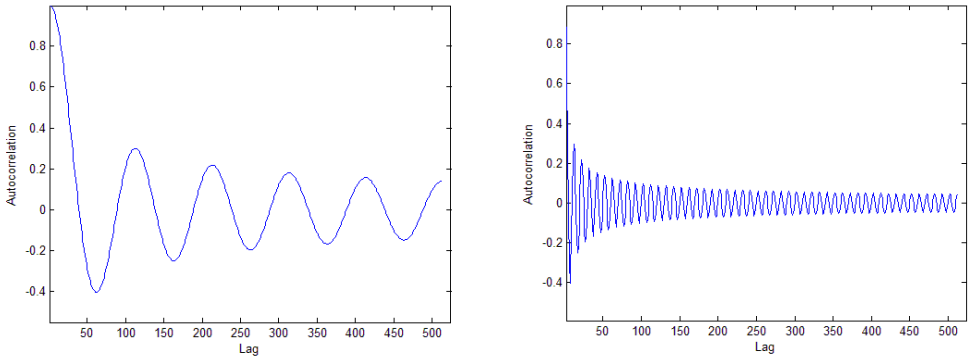


**Figure 1.7** Autocorrelation for a normalized Doppler frequency of 0.01 Hz (left) and 0.1 Hz (right)

Another interesting parameter for understanding the channel's behaviour is the Doppler spectral density. This parameter is obtained by means of the Fourier transform of the time-autocorrelation function of the channel. For Rayleigh fading with a vertical receive antenna with equal sensitivity in all directions is equal to [2]

$$S(\nu) = \frac{1}{\pi f_d \sqrt{1 - \left(\frac{\nu}{f_d}\right)^2}} \tag{1.10}$$

Where $\boldsymbol{\nu}$ is the frequency shift relative to the carrier. This equation is valid only for absolute values of the frequencies, $\boldsymbol{\nu}$, lower than the Doppler frequency, denoted as $\boldsymbol{f_d}$; since the Doppler spectrum outside this range is zero. Figure 1.8 shows the Doppler spectral density when the Doppler frequency is equal to 50Hz. Note the characteristic U-shape of this type of spectrum.

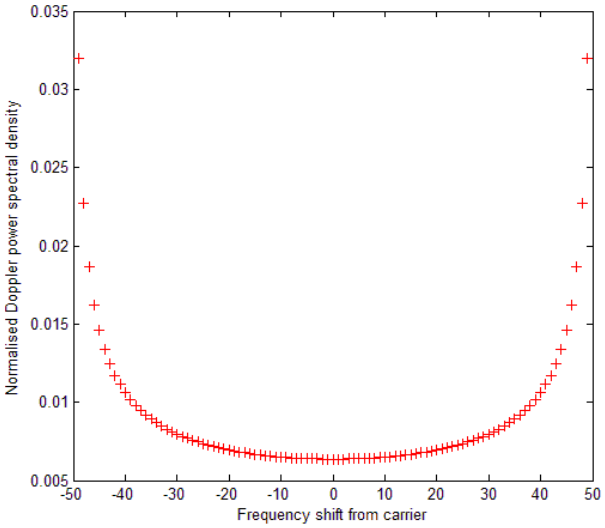


**Figure 1.8** Normalized Doppler Power Spectral Density for a Doppler frequency of 50 Hz

# CHAPTER 2. CHANNEL GENERATION TECHNIQUES

In this project we are going to describe how it is possible to generate correlated Rayleigh fading channels for wireless communication systems. Three methods for that purpose will be explained: The sum of sinusoids, Smith's technique and the Bealieu-Young strategy. The first method is based on simple sums of exponentials whereas the last two ones resort to the Inverse Discrete Fourier Transform (IDFT) to generate the channel coefficient.

## 2.1 Methods for Correlated Rayleigh Fading Generation.

### 2.1.1. Sum of sinusoids technique.

The main idea of this algorithm is to add sinusoids that have been generated using random Gaussian variables to obtain a signal that will have the properties of a correlated Rayleigh Fading. The channel for the in phase and quadrature components of a signal with $\boldsymbol{M}$ rays can be expressed as [3];

$$h_I(NT_s) = \sum_{m=1}^{M} cos\left\{\underbrace{B\ cos[(2m-1)D+\theta]}_{C_m} + \alpha_m\right\} \quad (2.1)$$

$$h_Q(NT_s) = \sum_{m=1}^{M} sin\{C_m + \beta_m\} \quad (2.2)$$

Being the constants $B$ and $D$ the expressions $B = 2\pi\, f_d\, NT_s$ and $D = \pi/4M$ respectively. In (2.2) $\alpha_m$and $\beta_m$are the phases uniformly distributed over $[0,2\pi]$ and, $\theta$ is the phase uniformly distributed over $[0, (2\pi/4M)]$. There $T_s$ corresponds to the sampling period and $N$ represents the number of samples. The total expression is given by:

$$h(NT_s) = A\big(h(NT_s) + jh(NT_s)\big) \quad (2.3)$$

Where $A = \frac{1}{\sqrt{M}}$

Considering that *B* and *A* are constant values and can be pre-calculated in tables the load of the algorithm is computed in Table 2.1.

**Table 2.1** Load of the Sum of Sinusoids method.

| **Operation type** | **Number Operations** |
|---|---|
| Generation of random variables | $(2M+1)N$ |
| Multiplications | $(2+2M)N$ |
| Cosine or sinusoid | $3MN$ |
| Additions or subtraction | $(2(M-1)+4M)N$ |
| Complex Additions | $1N$ |

If our signal was formed by infinite rays the theoretical power spectrum of either the real or imaginary would be the same has the one of a correlated Rayleigh Fading.

$$S(\nu) = \frac{1}{\pi f_d \sqrt{1 - \left(\frac{\nu}{f_d}\right)^2}} \tag{2.4}$$

Where $\nu$ is the frequency shift relative to the carrier frequency. This equation is valid only for values of frequencies, $\nu$, lower than the Doppler frequency, $f_d$; since the Doppler spectrum is zero outside this range.

### 2.1.2. Method of Smith.

The method of Smith [2] has been widely utilized in simulations of wireless systems. In this method, the IDFT operation is applied to complex sequences of independent, normally distributed random numbers, each sequences weighted by appropriate filter coefficients.

In order to obtain a perfect Rayleigh fading the spectrum of both the filter of the Method of Smith and Rayleigh fading should be the same. In order to have a perfect match between these Doppler spectrum is necessary to simulate a large number of samples. However, sometimes is not possible to simulate a large ensemble of data since we have constrains on time or complexity. For this reason, the Smith algorithm makes two big changes respect to the Sum of Sinusoids method. First it will sample the continuos-time spectrum at frequencies $\mathrm{k}\emptyset_s/\mathrm{N}$ with $\mathrm{k} = 0,1,2 \ldots \mathrm{N}-1$ to avoid the finite-time effects. then special treatment is given to the frequency (k=0), which is made zero, and to the point $k_m$ that's the point at, or just below the maximum Doppler frequency.

$$k_m = \lfloor f_m \, N \rfloor \tag{2.5}$$

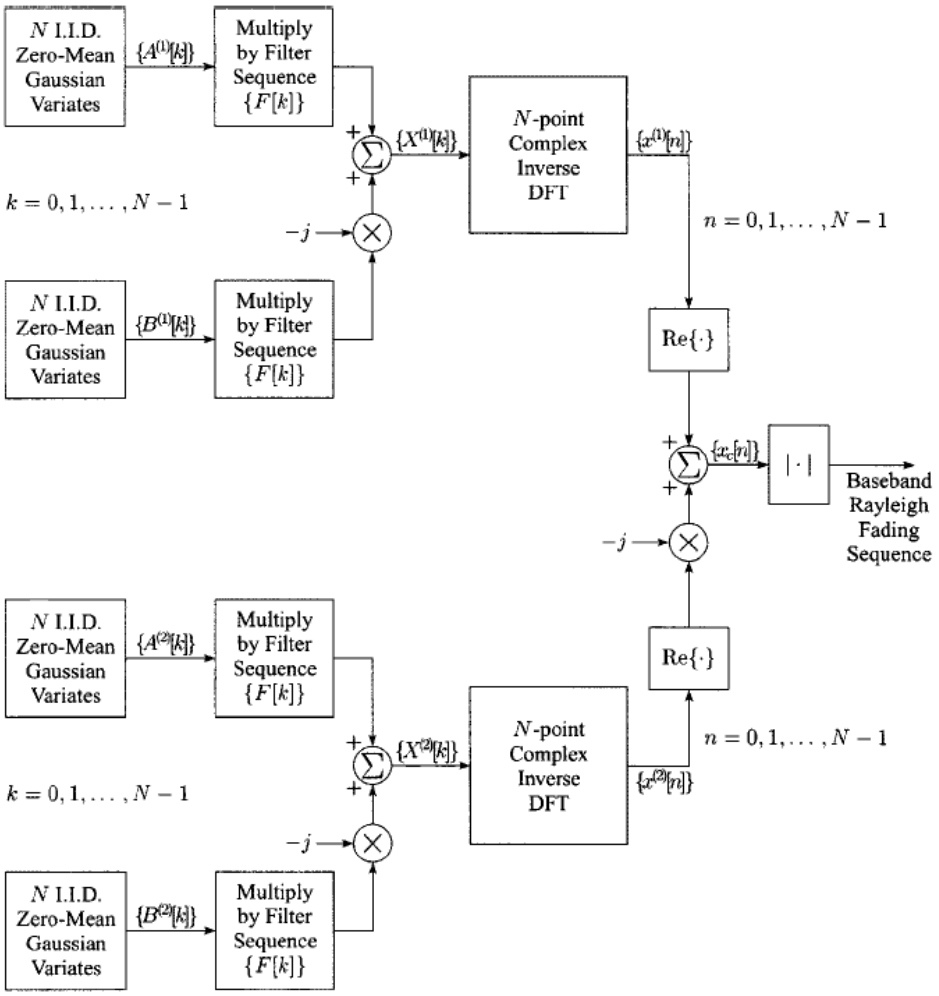


**Figure 2.1** Block diagram [2] of the algorithm of Smith to generate correlated Rayleigh variables.

The coefficient at $k_m$ is chosen such that the area under an interpolation of the spectrum coefficients is equal to the area under the continuous-time spectrum curve. After the modifications, the filter can be expressed as,

$$F_s[k] = \begin{cases} 0, & k = 0 \\ \sqrt{\dfrac{1}{\sqrt{1-\left(\dfrac{k}{B}\right)^2}}}, & k = 1,2,\ldots,k_m - 1 \\ A & k = k_m \\ 0, & elsewhere \end{cases} \tag{2.6}$$

With the parameters *A* and *B* being equal to:

$$A = \sqrt{k_m \left[\frac{\pi}{2} - arctan\left(\frac{k_m - 1}{\sqrt{2k_m - 1}}\right)\right]} \tag{2.7}$$

$$B = N\, f_m \tag{2.8}$$

Where $f_m$ is the Doppler frequency normalised by the sampling frequency $f_m = \frac{f_d}{f_s}$ .Thus the computational load for generating the Smith's filter, taking into account that *A* and *B* are constant values that can be pre-calculated in tables is found on Table 2.2

**Table 2.2** Load of the filter for Smith's method.

| **Operation type** | **Number Operations** |
|---|---|
| Square root | $2\,\mathrm{k_m}$ |
| Multiplications or division | $3\mathrm{k_m}$ |
| Addition or substraction | $1\mathrm{k_m}$ |

The structure of the Smith's algorithm according to Figure 2.1 is the following; Initially, we generate four Gaussian random variables that are multiplied by the Smith's filter sequence $F_s[k]$, then they are combined to form two complex sequences, one of the will be the real part and the other the complex one. Each of these sequences will pass through its own IDFT. Next the imaginary part of each of the sequences is discarded and finally combined to form the final complex signal.

The filter sequence is non-zero in $k_m$ points. Therefore it will not be necessary to create *N* random Gaussian variables.

Computing the IFFT directly is to slow. For that reason the Inverse Discrete Fast Fourier Transform (IDFT) algorithm can be used. It´s a much faster way of obtaining the IDFT but with the drawback that *N* needs to be a power of two. Zero padding, which consists in adding zeros until the desired value, can be used to obtain the closest power of two.

The load of the block diagram taking into account the top assumptions is found in Table 2.3;

**Table 2.3** Load of the Smith's method.

| **Operation type** | **Number Operations** |
|---|---|
| Generation random variables | 4 $k_m$ |
| Multiplications | $4k_m$+ $3N_{FFT}$ |
| Complex IFFT | 2 $N_{FFT}$ |
| Complex Additions | 3 $N_{FFT}$ |
| Absolute value | 1 $N_{FFT}$ |

### 2.1.3. Method of Belieu-Young

This modification of Smith method minimizes the number of IDFT´s used to compute the channel coefficients. If the filter used in the Method of Smith is given by $F_S[k]$, then the modified filter is defined as

$$F_s[k] = \begin{cases} F_S[0], & k = 0 \\ \frac{1}{\sqrt{2}} F_S[k], & k = 1,2,\dots,\frac{N}{2} - 1 \\ F_S[k] & k = \frac{N}{2} \\ \frac{1}{\sqrt{2}} F_S[N-k], & k = \frac{N}{2} + 1,\dots,N-1 \end{cases} \tag{2.9}$$

The differences between this filter and the one in the Method of Smith are: a constant value, that can be pre-calculated in a table, and that the filter is symmetric on $N/2$. These differences do not introduce additional computational load respect to the Method of Smith. The block diagram has the form

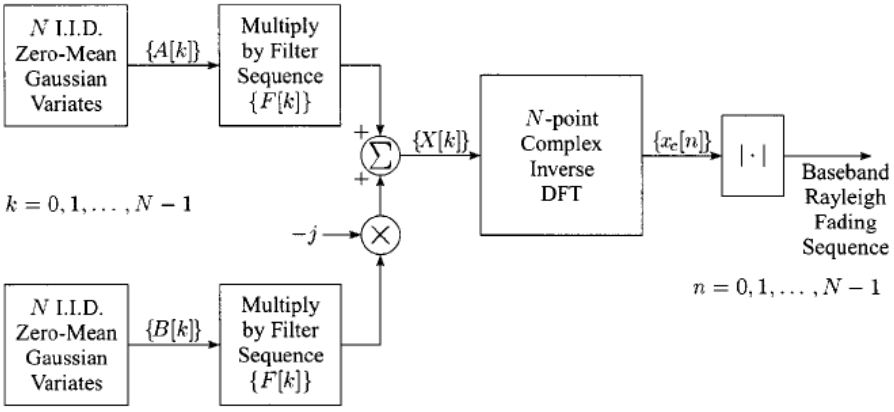


**Figure 2.2** Block diagram [2] of Bealieu-Young correlated Rayleigh fading generation.

Two Gaussian random variables are generated, multiplied by the filter sequence $F[k]$, then combined to form the imaginary and real part of the final sequence that will pass through the IDFT.

Taking into account that an IFFT is used and that the values of the filter that are not non-zero are $2k_m$ .The load of this method is given in Table 2.4:

**Table 2.4** Load of the Bealiu-Young method.

| **Operation type** | **Number Operations** |
|---|---|
| Generation random variables | (2)2$k_m$ |
| Multiplications | (2)2$k_m$+1$N_{FFT}$ |
| Complex IFFT | 1$\mathrm{N_{FFT}}$ |
| Complex Additions | 1$\mathrm{N_{FFT}}$ |
| Absolute value | 1$\mathrm{N_{FFT}}$ |

The use of this modified filter and block diagram will produce a complex Gaussian sequence with identical autocorrelation to Smith´s original routine and the required independence between the real and imaginary parts.

## 2.2 Results

The three methods are implemented using Matlab and analysed in terms of time consumption and auto-correlation. In order to study the results in an accurate way it has been run a large number of iterations. The results compare the processing time and the error power in the estimation of the autocorrelation function.

### 2.2.1. Autocorrelation

The normalized autocorrelation function of a correlated Rayleigh Fading is the Bessel function of order zero which it is denoted as,

$$r[d] = J_0(2\pi\, f_m\, |d|) \tag{2.10}$$

where $d$ is the sample lag.

The results have been obtained using a signal of $2^{13}$ samples, a Doppler frequency of 64 Hz and $2^{17}$ iterations. The correlation of the three methods compared to the ideal behaviour, the Bessel function of order zero, has been calculated in terms of power Error. The closest to this function the better the algorithm is. Two different types of sum of sinusoids are analysed, one formed by 10 rays and the other by 50 rays.

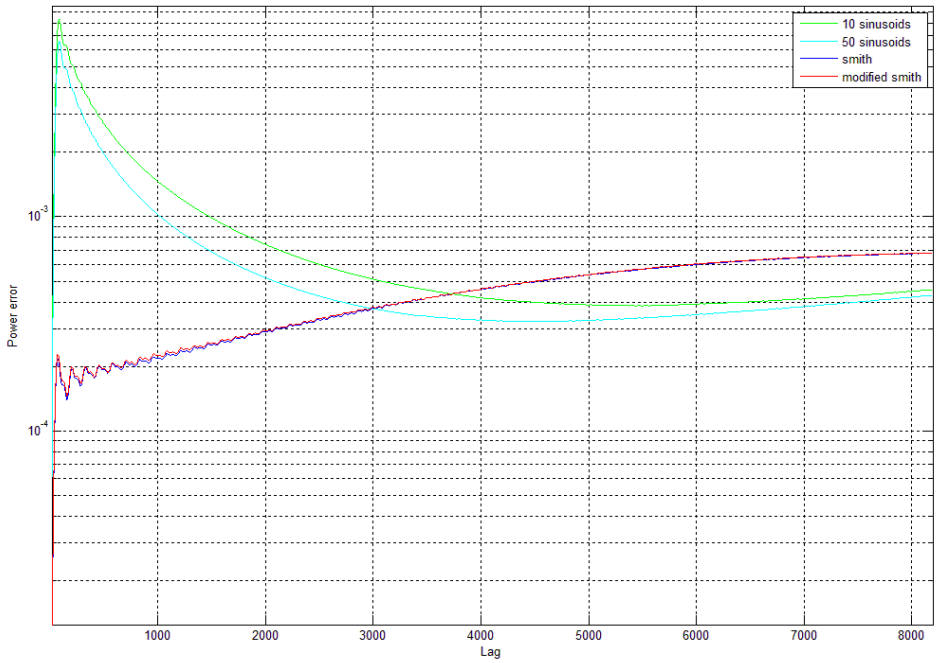


**Figure 2.3** Power error of the sum of sinusoids and IDFT methods in comparison to the Bessel function of order zero.

In Figure 2.3. we can observe that there is no difference between the two IDFT methods, the error power increases with the number of lags. The error power obtained for the sum of sinusoids method formed by 10 and 50 rays is different. The method that uses 50 rays is better. The error increases really fast and then there is a point where it starts to decrease then it stabilizes to finally start to grow again.

The smallest power error is found at low lags, the best simulation of the channel is at these points. The methods that give us the smallest power error are the IDFT methods.

Part of the error is due that the theoretical value of the autocorrelation, the Bessel function of order zero is obtained through the autocorrelation of samples of infinite values and in the practice this is not possible. That´s why an analysis done with a larger number of samples and iterations offers us better results.

### 2.2.2. Processing time

The processing time is very important, the faster our algorithm the better it is. The three different methods have been implemented in Matlab R2012b and run over a computer with the following characteristics (Table 2.5)

**Table 2.5** Computer characteristics

| | |
|---|---|
| Windows edition | Windows 7 Professional |
| Processor | Intel® Core™2 Quad CPU at 2,66 GHz |
| RAM | 2 GB |
| System type | 64-bit Operating System |

Whereas the theoretical computational load of each of the methods is (Table 2.6):

**Table 2.6** Summary of the load of the all studied methods for correlated Rayleigh generation.

| Method | Sum of Sinusoids | Method of Smith | Method of Belieu-Young |
|---|---|---|---|
| **Generation of random variables** | $(2M+1)N$ | $4k_m$ | $4k_m$ |
| **Multiplications or divisions** | $(2+2M)N$ | $7k_m + 3N_{FFT}$ | $7k_m + 1N_{FFT}$ |
| **Cosine or sinusoid** | $3MN$ | | |
| **Square root** | | $2k_m$ | $2k_m$ |
| **Complex IFFT** | | $2N_{FFT}$ | $1N_{FFT}$ |
| **Additions or subtractions** | $(2(M-1)+4M)N$ | $1k_m$ | $1k_m$ |
| **Complex Additions or subtractions** | $1N$ | $3N_{FFT}$ | $1N_{FFT}$ |
| **Absolute value** | | $1N_{FFT}$ | $1N_{FFT}$ |

The computational load of the filter in the method of Belieu-Young and Smith are the same, the differences are found in the implementation of the block diagram. In the sum of sinusoids almost all of the terms depend on the number

of rays, more rays mean a higher computational load.

In Table 2.7 the practical processing time for one iteration has been calculated for samples that go from $2^{11}$ to $2^{16}$ .

**Table 2.7** Processing times in milliseconds depending of the number of samples for the smith and modified smith algorithm.

| **Method\N** | $\mathbf{2^{11}}$ | $\mathbf{2^{12}}$ | $\mathbf{2^{13}}$ | $\mathbf{2^{14}}$ | $\mathbf{2^{15}}$ | $\mathbf{2^{16}}$ |
|---|---|---|---|---|---|---|
| **Method of Belieu-Young** | 7,9 | 8,4 | 10 | 13,1 | 18,7 | 28,8 |
| **Method of Smith** | 11 | 13 | 18 | 37 | 72 | 118 |
| **10 sinusoids** | 19,4 | 22,3 | 25,9 | 33,9 | 44,1 | 74,3 |
| **50 sinusoids** | 32,5 | 46 | 64,1 | 84,8 | 146,9 | 288,1 |

From the results of Table 2.7 we conclude that the method of Belieu-Young is the one that takes less time to be calculated, followed by the Smith version, and finally the sum of sinusoids is the most time consuming one.

### 2.2.3. Conclusions

The smallest error power in the Rayleigh generation methods based on IDFT is obtained when the lags are small. From the computational point of view, the method with the lowest burden requirements is the method of Belieu-Young. This strategy provides the best estimation of the autocorrelation with the lowest burden requirements. So, this method will be the strategy of correlated Rayleigh fading generation that will be implemented in Simulink and FPGA. Next, after selecting the channel generating method we will evaluate different strategies of channel estimation in the next chapter.

# CHAPTER 3. THEORETICAL AND PRACTICAL ANALYSIS OF DIFFERENT CHANNEL ESTIMATORS

Orthogonal Frequency Division Multiplexing (OFDM) has been applied widely in different communication standards due to its high bandwidth efficiency and its robustness to multi-path delay (e.g. DVB-T, LTE,WIMAX,etc).

A channel estimator improves the system performance in terms of bit error rate by means of the addition of known values in the OFDM frame called pilots that help the receiver to predict the channel behaviour. Channel estimation strategies depend on the pilot positioning, the number of pilots to use, and the criteria used to estimate the channel. In this project we evaluate the performance of the channel estimation for a different number of pilots, position of the pilots, and strategies of channel estimation.

The OFDM transmission scheme makes it easy to assign pilots in both time and frequency domains. There are two major types of pilot arrangement. The first kind of pilot arrangement is denoted as block-type pilot arrangement. The pilot signal is assigned to a particular OFDM symbol, which is sent periodically in time domains. This type of pilot arrangement is especially suitable for slow-fading radio channels. The second kind of pilot arrangement is denoted as comb-type pilot arrangement. The pilot signals are uniformly distributed within each OFDM symbol. It´s been introduced to satisfy the need for estimators even when the channel changes in one OFDM symbol, it provides better robustness to fast-fading channels. So, in this case the pilot spacing must be much smaller than the coherence bandwidth of the channel.

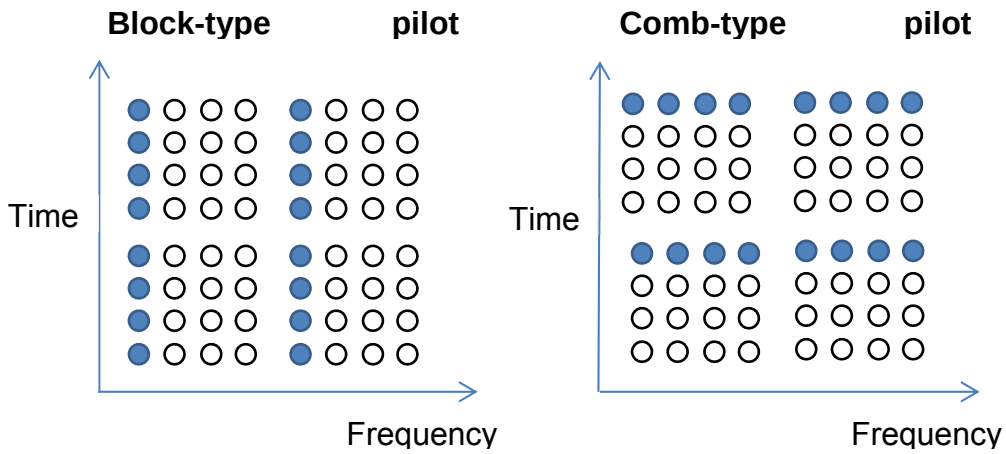


**Figure 3.1** Different strategies for pilot arrangement, block-type (left) and comb-type (right)

The noise relation ratio (SNR) is;

$$SNR = \frac{E\{|S_k|^2\}}{E\{|n|^2\}} = \frac{E_s}{N_0} = \frac{E_b}{\sigma^2} = \frac{1}{\sigma^2} \tag{3.1}$$

where $\sigma^2$ is the noise power This noise is a random variable that is multiplied by a constant value of $\sqrt{\sigma^2}$ making the stand deviation of the noise $\sigma^2$.

In this chapter will first analyze two different ways of estimating the value of the pilots, then three different ways of calculating the data values are evaluated for particular values of Doppler frequency, distance between pilots and noise power.

## 3.1 Pilot estimation methods.

If the guard interval is longer than the length of the channel impulse response then there is no inter-symbol interference between OFDM symbols. So, if we assume that there is no intersymbol interference, and that the received signal is the k-th subcarrier of the n-th OFDM symbol we have;

$$y_{k,n} = h_{k,n} x_{k,n} + w_{k,n} \tag{3.2}$$

Where $y$ is the received signal, $h$ represents the channel, $x$ is the transmitted data and $w$ identifies the Additive White Gaussian Noise (AWGN). In a matrix notation, the subcarrier of the n-th OFDM symbol is;

$$Y_n = X_n \cdot H_n + W_n \tag{3.3}$$

$$\begin{bmatrix} y_{1,n} \\ \vdots \\ y_{N,n} \end{bmatrix} = \begin{bmatrix} x_{1,n} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & x_{N,n} \end{bmatrix} \begin{bmatrix} h_{1,n} \\ \vdots \\ h_{N,n} \end{bmatrix} + \begin{bmatrix} w_{1,n} \\ \vdots \\ w_{N,n} \end{bmatrix} \tag{3.4}$$

The estimated channel can be obtained using different methods.

### 3.1.1. Least Square (LS) Estimation.

The main aim of the Least Square (LS) method is to minimize the distance between the data and the signal model.

$$J\left(H_{LS,n}\right) = \sum_{n=0}^{N-1} (h_n - \widehat{H}_{LS,n})^2 \tag{3.5}$$

Being the estimation of the channel equal to:

$$\widehat{H}_{LS,n} = X_P^{-1} Y_n = \left[\frac{y_{1,n}}{x_{1,n}}, \dots, \frac{y_{N,n}}{x_{N,n}}\right] \tag{3.6}$$

### 3.1.2. Minimum mean square error (MMSE) Estimation.

Minimum Mean Square Error (MMSE) estimator is an estimation method which minimizes the Mean Square Error (MSE) considering channel and noise statistics.

$$\widehat{H}_{MMSE,n} = (R_H + \sigma_w^2 I)^{-1} Y_n \widehat{H}_{LS} \tag{3.7}$$

Where $\sigma_w^2$ is the noise variance, $I$ is the identity matrix and $R_H$ is:

$$R_H(\Delta t, \Delta f) = R_t(\Delta t) R_f(\Delta f) \tag{3.8}$$

$R_t(\Delta t)$ and $R_f(\Delta f)$ represent the correlation function of the channel given a temporal variation equal to $\Delta t$ and a separation in frequency of $\Delta f$. The LS estimate is susceptible to AWGN because it does not take it in account in its calculations. The MMSE estimate is better because it does take it into account, but its complexity is much higher .LS is going to be good for systems with small AWGN and the MMSE for systems with large AWGN. In our calculations will consider the LS estimator [3],[4].

## 3.2 Data estimation methods

Once the value of the pilots is calculated the estimation of the data can be obtained. The error power between the estimated and real channel is calculated as

$$E\left\{e_{h_{nD_p+k}}\right\} = E\left\{\left|\hat{h}_{nD_p+k} - h_{nD_p+k}\right|^2\right\} \tag{3.9}$$

Where in (3.9) the value of $E\{.\}$ represents the expected operator, $\hat{h}_{nD_p+k}$ the estimated channel and $h_{nD_p+k}$ the real channel for the subcarrier $nDp + k$. The estimated channel in the n-th pilot is given by

$$\hat{h}_{nD_p} = \left(h_{nD_p} + n_{nD_p}\right) \tag{3.10}$$

$$\hat{h}_{(n+1)D_p} = \left(h_{(n+1)D_p} + n_{(n+1)D_p}\right) \tag{3.11}$$

Where $n_{nD_p}$ is the noise of the in the n-th pilot. We have considered that the pilots can be distributed in two forms along the frames: uniformly distributed, like the linear interpolator and the average channel estimator, or at their beginning.

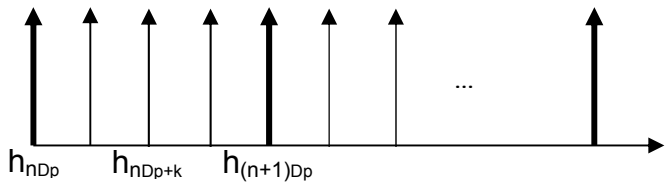


**Figure 3.2** Scheme where the pilots are uniformly distributed along the frame.

Where in Figure 3.2, $k$ is the separation of the position of the data to the pilot, $D_p$ represents the separation between pilots and $\hat{h}_{nD_p}$ is the estimated channel of the n-th pilot.

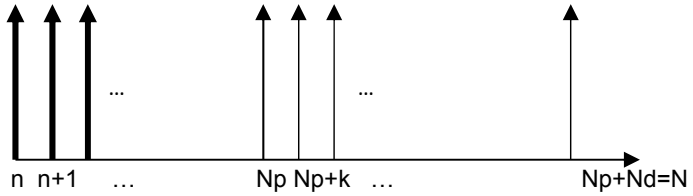


**Figure 3.3** Scheme where the pilots are at the beginning of the frame.

where in Figure 3.3, $n$ is the n-th pilot, $Np$ represents the number of pilots and $Nd$ symbolizes the number of data.

### 3.2.1. Linear Interpolation estimator.

This estimator consists in calculating the linear interpolation between two consecutive pilots [5].

The estimated channel in the k-th position is calculated as

$$\hat{h}_{nD_p+k} = \hat{h}_{nD_p} + \left(\hat{h}_{(n+1)D_p} - \hat{h}_{nD_p}\right)\frac{k}{D_p} \tag{3.12}$$

The error power of the channel estimation between two pilots, without considering the error of the pilots can be written as

$$\begin{aligned} J(D_p,\sigma,R) = Re\left(\frac{{D_p}^2-1}{3D_p}\right)R(D_p) - 2Re\left\{\sum_{k=1}^{D_p-1}\left(\frac{D_p-k}{D_p}\right)R(k) + \left(\frac{k}{D_p}\right)R(D_p-k)\right\} \\ + (D_p-1)R(0) + (R(0)+\sigma^2)\,\frac{(D_p-1)(2D_p-1)}{3D_p} \end{aligned} \tag{3.13}$$

Where $R$ is the autocorrelation

### 3.2.2. . Average Channel Estimation

It has the same disposition of pilots as the linear interpolation. The estimated channel values are obtained multiplying the pilots by a variable, such as

$$\hat{h}_{nDp+k} = \alpha_1 \hat{h}_{nDp} + \alpha_2 \hat{h}_{(n+1)Dp} \tag{3.14}$$

These variables are obtained through the following equations

$$E\{\hat{h}_{nDp+k}\ \hat{h}_{nDp}{}^{*}\} \rightarrow R_h(\mathrm{k}) = \alpha_1 R_h(0) + \alpha_2 R_h(\mathrm{D_p}) \tag{3.15}$$

$$E\{\hat{h}_{nDp+k}\ \hat{h}_{(n+1)Dp}{}^{*}\} \rightarrow R_h(\mathrm{k} - \mathrm{D_p}) = \alpha_1 R_h(-\mathrm{D_p}) + \alpha_2 R_h(0) \tag{3.16}$$

The error power of the channel estimation between two pilots, without considering the error of the pilots can be written as

$$\begin{aligned} J(\alpha_1, \alpha_2, \sigma, D_p, \mathrm{R}) = {} & (|\alpha_1|^2 + |\alpha_2|^2)(\mathrm{R}(0) + \sigma^2)(D_p - 1) + 2\mathrm{Re}\{\alpha_1 {\alpha_2}^* R(-D_p)\}(D_p - 1) \\ & - 2\mathrm{Re} \sum_{\mathrm{k}=1}^{D_p - 1} \mathrm{R}(\mathrm{k}) + \mathrm{R}(0)(D_p - 1) \end{aligned} \tag{3.17}$$

There are two considerations that can be assumed to find $\alpha_1$ and $\alpha_2$, the first one is

$$\alpha_1 + \alpha_2 = 1 \tag{3.18}$$

then $\alpha_1$ and $\alpha_2$ are written as

$$\alpha_1 = \frac{R_h(\mathrm{k}) - R_h(\mathrm{D_p})}{R_h(0) - R_h(\mathrm{D_p})} \tag{3.19}$$

$$\alpha_2 = \frac{R_h(0) - R_h(\mathrm{k})}{R_h(0) - R_h(\mathrm{D_p})} \tag{3.20}$$

And the second consideration is when

$$\alpha_1 + \alpha_2 \neq 1 \tag{3.21}$$

then $\alpha_1$ and $\alpha_2$ are written as

$$\alpha_1 = \frac{R_h(\mathrm{k})R_h(0) - R_h(\mathrm{k}-\mathrm{D_p})R_h(\mathrm{D_p})}{|R_h(0)|^2 - |R_h(\mathrm{D_p})|^2} \tag{3.22}$$

$$\alpha_2 = \frac{R_h(0)R_h(\mathrm{k}-\mathrm{D_p}) - R_h(-\mathrm{D_p})R_h(\mathrm{k})}{|R_h(0)|^2 - |R_h(\mathrm{D_p})|^2} \tag{3.23}$$

Once the mathematical expressions are obtained for both the estimators the behaviour for different, Doppler frequencies, pilot distance and SNR can be analysed.

First the error floor, the system with no noise is calculated

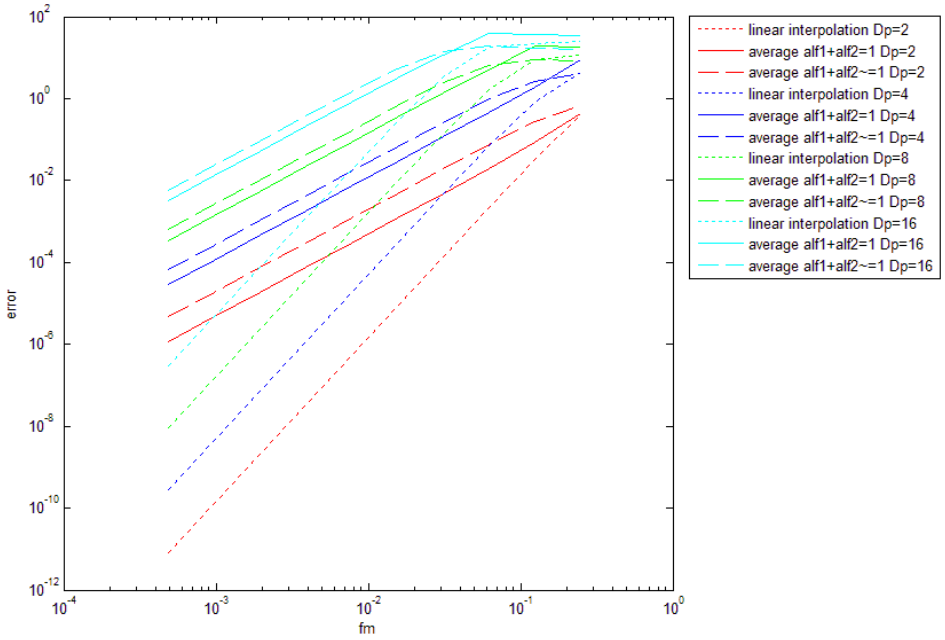


**Figure 3.4** Error power for average vs linear interpolation channel estimator with no noise.

In Figure 3.4 we observe how the error power increases with the Normalized Doppler frequency and the number of pilots. The average estimators are worse that the linear channel estimator for all Doppler frequencies, this last one is able to follow better the variations of the channel. Between both channel estimators the strategy of averaging the channel estimates is the one that gives better results when the sum of the weighters of the pilots is the unity.

When noise is added to the system we obtain Fig. 3.4.

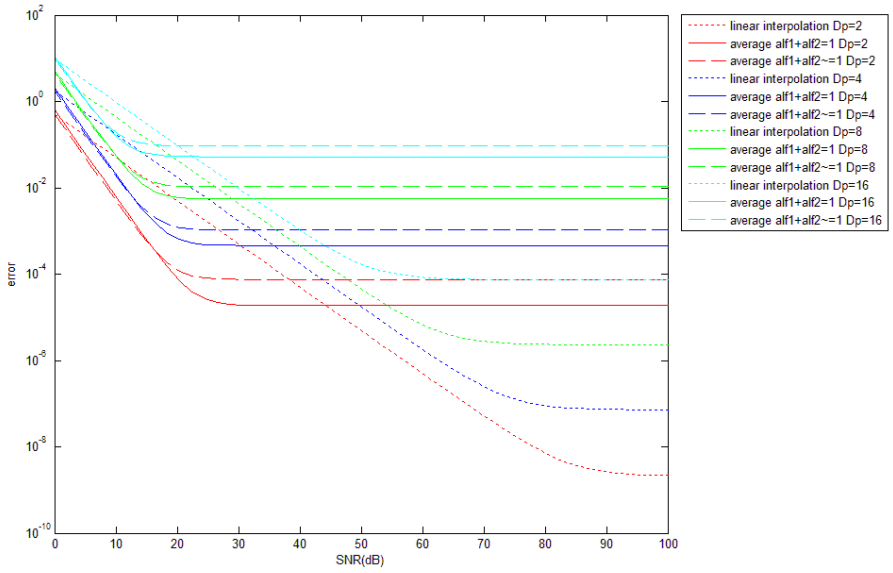


**Figure 3.5** Error power for average vs linear interpolation channel estimator for a normalized Doppler frequency of 4/2048.

The linear interpolation channel estimator has high error power for low SNR, making this method more susceptible to noise .When the noise is small it gives better results than the average channel estimator. The reason is that at low SNR the noise mask the channel value. If we average the channel samples and we assume that the channel has a slow variation then the noise effect is removed from the averaged channel. On the contrary, at high SNR, the noise effect is small and consequently the interpolator is able to track the changes in the channel much better than the averaging estimation. In fact the averaging filter can be considered as a low pass-filter. So, at low SNR it removes the noise effect, since introduces the high frequencies in the channel estimation process. However, at high SNR the averaging filter removes the high frequency bands of the channel, and therefore it offers a worse quality than the linear interpolator.

### 3.2.3. Pilots at the beginning of the frames

There are two ways of estimating the channel when putting the pilots at the beginning. The first one consists in using one of the pilots to calculate the channel and the second where the average of all the pilots is used. For the first case, the pilot selected is the pilot that is found in the position $N_p/2$ , such as

$$\hat{h}_k = \hat{h}_{\left\lfloor \frac{Np}{2} \right\rfloor} \tag{3.24}$$

The error power of the entire frame without considering the error of the pilots can be written as

$$J(N_d, N_p, \sigma, \mathrm{R}) = N_d(2R(0) + \sigma^2) - 2Re \sum_{k=N_p+1}^{N} R\left(k - \left\lceil \frac{N_p}{2} \right\rceil\right) \tag{3.25}$$

The second case where the average of all the pilots is considered

$$\hat{h}_k = \frac{1}{N_p} \sum_{n=1}^{N_p} \hat{h}_n \tag{3.26}$$

The error power of the entire frame without considering the error of the pilots can be written as

$$\begin{aligned} J(Nd, N_p, \sigma, \mathrm{R}) = \frac{N_d}{{N_p}^2} \left( N_p(R(0) + \sigma^2) + \sum_{n=1}^{N_p-1} 2(N_p - n)R(n) \right) \\ - \frac{2}{N_p} Re \sum_{k=1}^{N_d} \sum_{n=1}^{N_p} R(k-n) + N_d \left(R(0)\right) \end{aligned} \tag{3.27}$$

The error power for both the methods considering the error can be found in the Annex.

When the pilots are positioned at the beginning of the frames the performance of the channel estimator depends on the frame length. In order to note this effect we have compared the error power in the channel estimator for two different frame lengths. The results are obtained for the same proportion of pilots as the previous system, the parameter to describe this relation is $N/Np$ that means frame length divided by number of pilots, more pilots smaller the division and the other way round.

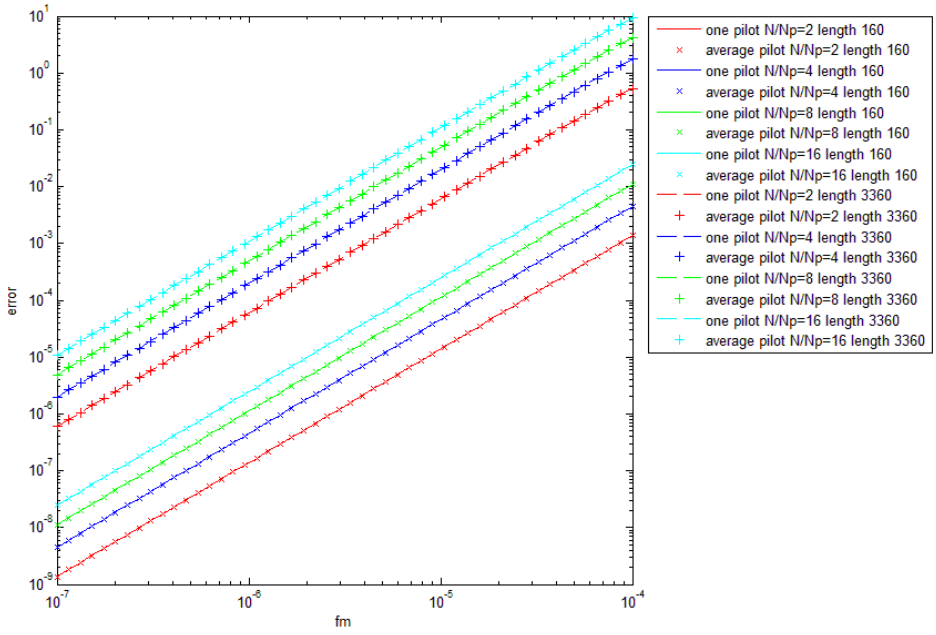


**Figure 3.6** Error power for pilots in front channel estimator for different lengths and no noise.

Figure 3.6 shows that the larger the Doppler frequency and the ratio $N/Np$ are, the worse the error power is. If the channel has any mobility then, the positioning of the pilots of the frame is not able to capture the mobility of the channel. As a result the error power in the channel estimation increases. So, this method is good for channels with a low Doppler frequency.

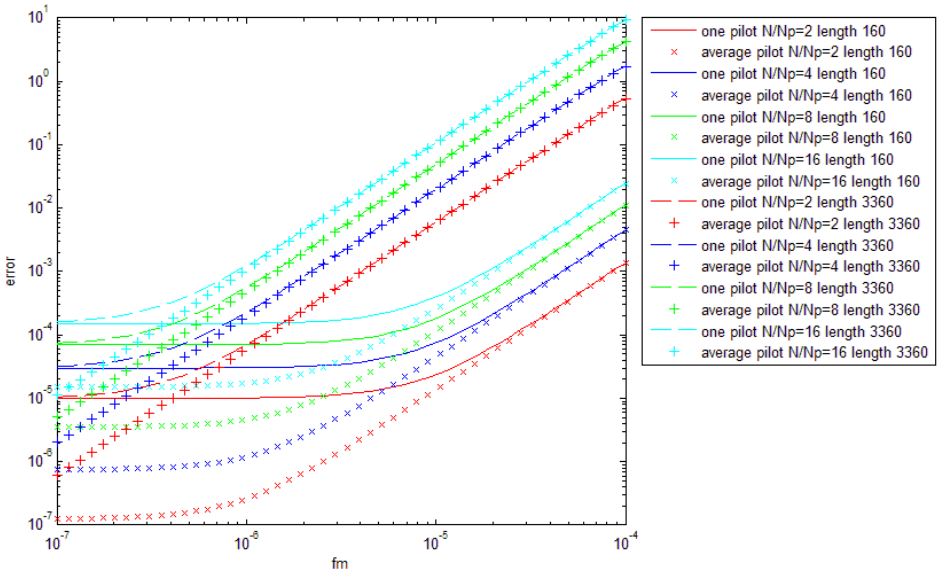


**Figure 3.7** Error power for pilots in front channel estimator for different lengths and SNR=50dB.

Now that there is noise, a difference between just using one pilot and the average of all is observed, the noise introduced has average zero, this means

that doing the average of all the pilots is better because the noise starts to compensate and introduces less error than when only one pilot is used.

We have considered that the pilots can be distributed in two forms along the frames: uniformly distributed or at their beginning. In the first case the channel has a medium – is good for high mobility (i.e. vehicular channel) whereas in the second one, the system is better for lower variation (i.e. indoor channel). The first case also has a higher computational load, while the second case it is smaller.

# CHAPTER 4. REAL SYSTEM IMPLEMENTATION WITH SIMULINK AND MATLAB

In this chapter a reconfigurable linear channel estimator has been implemented using Simulink and Matlab. Simulink is a data flow graphical programming language tool for modeling, simulating and analyzing multidomain dynamic systems. This allows us to visualize how the system works from a time perspective. First the implementation and results of Simulink are obtained then they are sent to Matlab, where they are represented in figures.

## 4.1 Simulink Implementation

The system consists of four main blocks such as

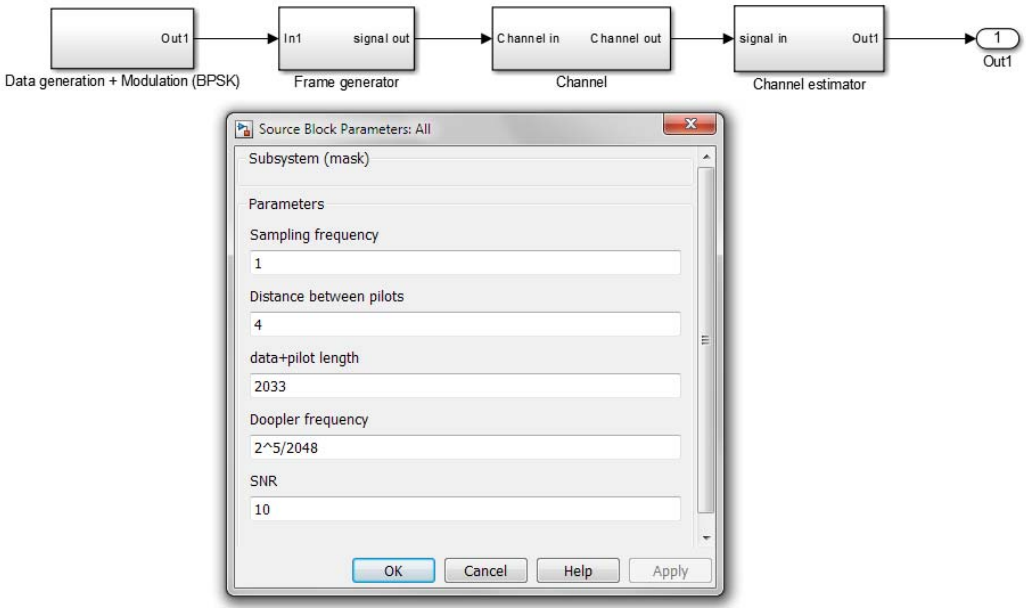


**Figure 4.1** Discretized channel estimation.

In the first block a series of random binary variables are generated and BPSK modulated. Next in the frame generator the pilots and zeros are added. The frame length is going to be the same as the channel length, which will be a power of two since the FFT and IFFT algorithms are optimized for a number of subcarriers equal to a power of two.

In the channel block the filter used to obtain the channel parameters is created, then used to model the random variables that form the signal that will be introduced in the IFFT module.

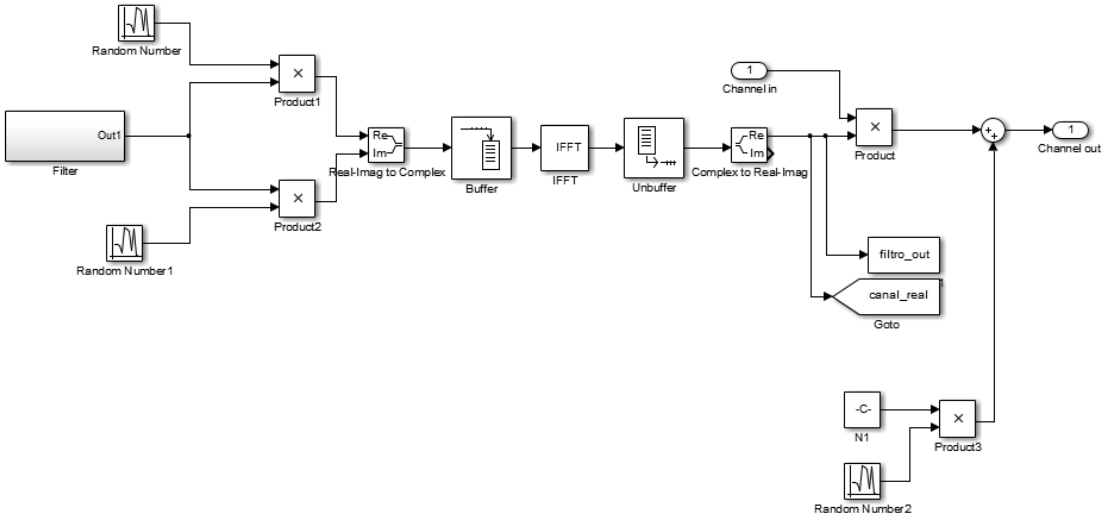


**Figure 4.2** Channel creation Block Diagram In Simulink.

The only delay added to this system is the one introduced by the IFFT calculation. This operation needs all the values of the frame at the same time, this is why a buffer that will accumulate the values to the desired length is added. After passing through the IFFT the values are unbuffered one by one, then multiplied by the values obtained by the frame generator and finally the noise is added.

After adding the channel estimation block, it is computed the error power of the channel estimation process.

## 4.2 Simulink Results

The first Simulink results are obtained for a channel with no noise and different values of pilot distance and normalized Doppler frequency.

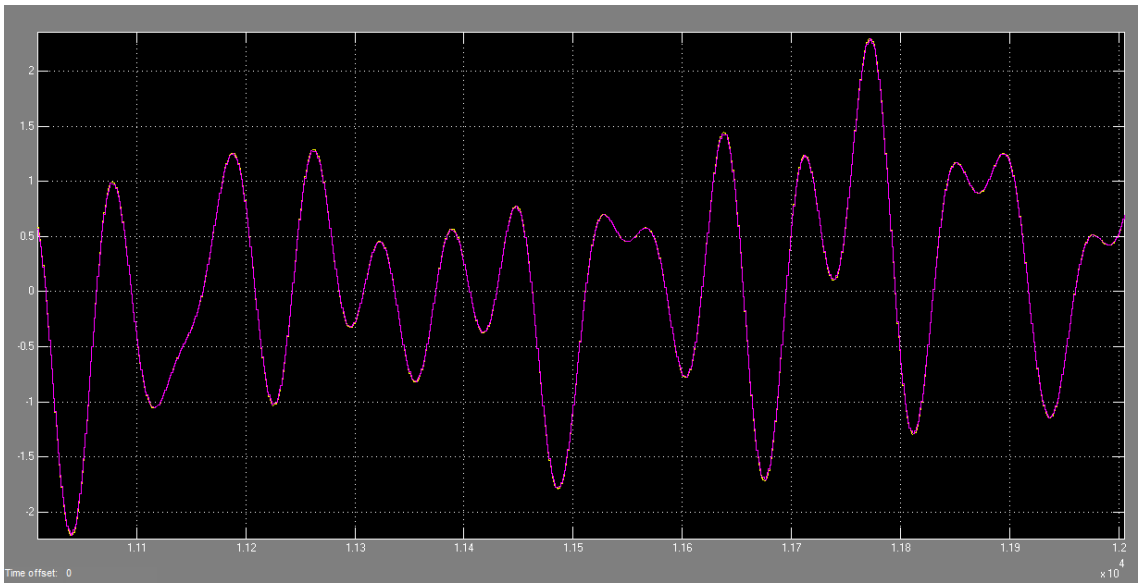


**Figure 4.3** Normalized Doppler frequency of 32/2048 and Dp=4 with no noise.

Figure 4.3. shows the channel estimation when the number of subcarriers is 2048, whereas the Doppler frequency is 32/2048 and the distance between pilots is equal to four. From this figure, it is possible to observe that the channel estimation is quite close to the actual channel value.

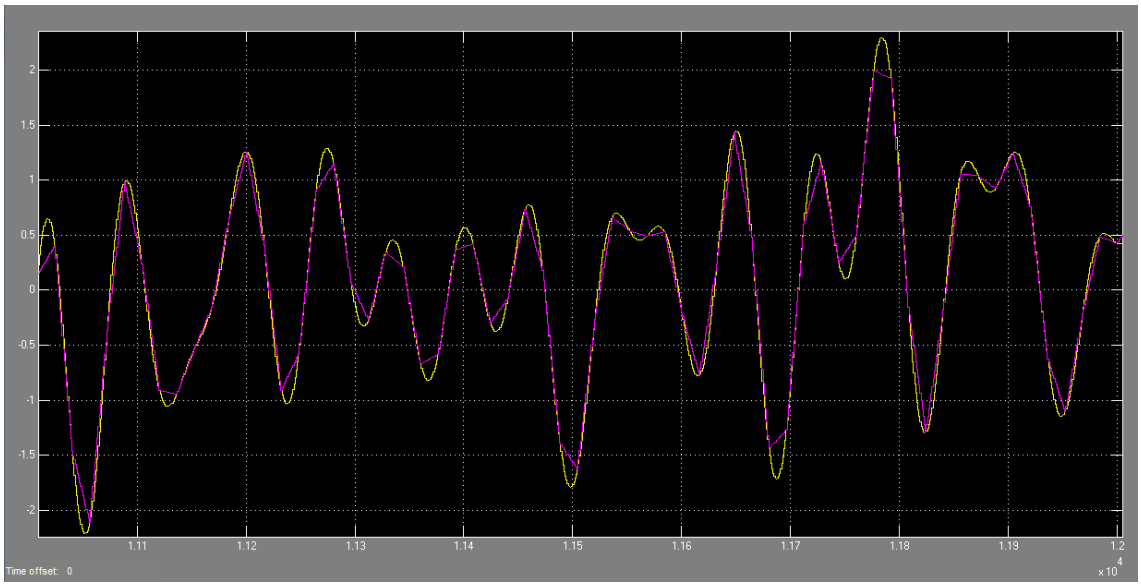


**Figure 4.4** Normalized Doppler frequency of 32/2048 and Dp=16 and no noise.

On the contrary in Figure 4.4 we increase the separation of the channel pilots to sixteenth subcarriers and keep constant the frequency Doppler. So, note as the increase of the separation between channel pilots increase the errors in the channel estimation. However, the channel estimator is able to track the changes in the channel.

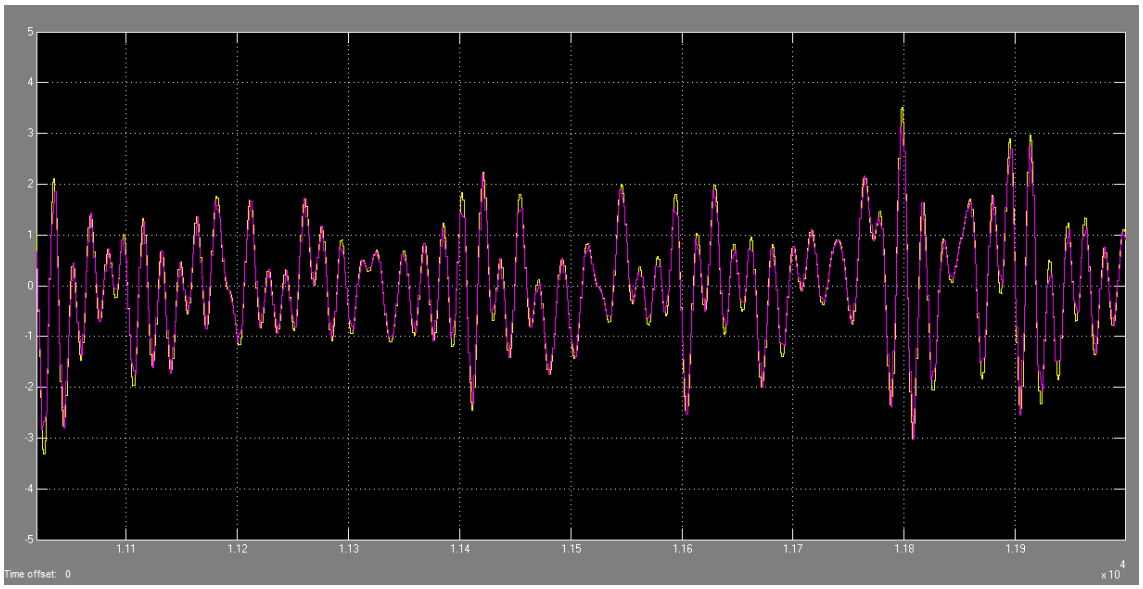


**Figure 4.5** Normalized Doppler frequency of 128/2048 and Dp=4 and no noise

In Figure 4.3 is not able to distinguish any difference between the channel estimation and its actual value. However, in Figure 4.5 the Doppler frequency is 128/2048 and now it is possible to observe that the channel estimate is less accuracy than for the case of Figure 4.3. However, the channel estimator is still valid for this channel estimation.

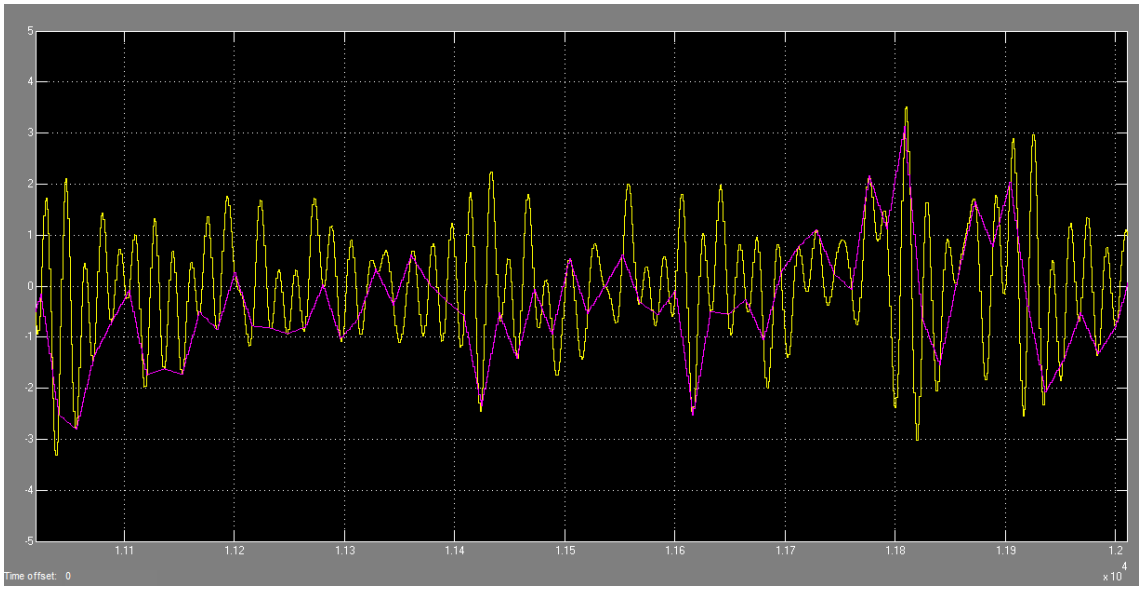


**Figure 4.6** Normalized Doppler frequency of 128/2048 and Dp=16 and no noise.

When the Doppler frequency is large and the distance between pilots also increases, the channel estimation cannot cope with the mobility of the channel. (seeFig.4.6). As a result the channel estimation cannot predict the channel since they tend to be uncorrelated. If noise is added to the channel, the result in the estimation is

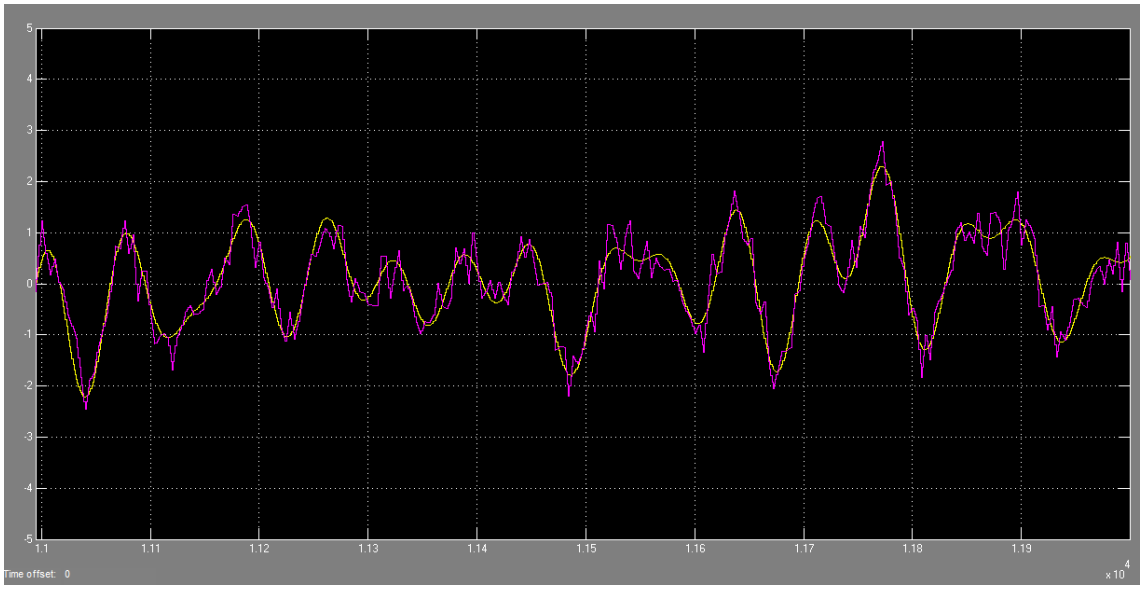


**Figure 4.7** Normalized Doppler frequency of 32/2048 and Dp=4 and SNR=10dB.

In Figure 4.7 the SNR is of 10 dB and the estimated channel can follow the real channel but with more errors than when there was no noise, bigger the SNR the better the channel estimation.

## 4.3 Matlab results

The real system is also implemented in Matlab and the results are the same as in Simulink, but instead of just analyzing one Normalized Doppler, pilot distance or noise at a time, our Matlab code was capable of doing a sweeping between various values. Such as

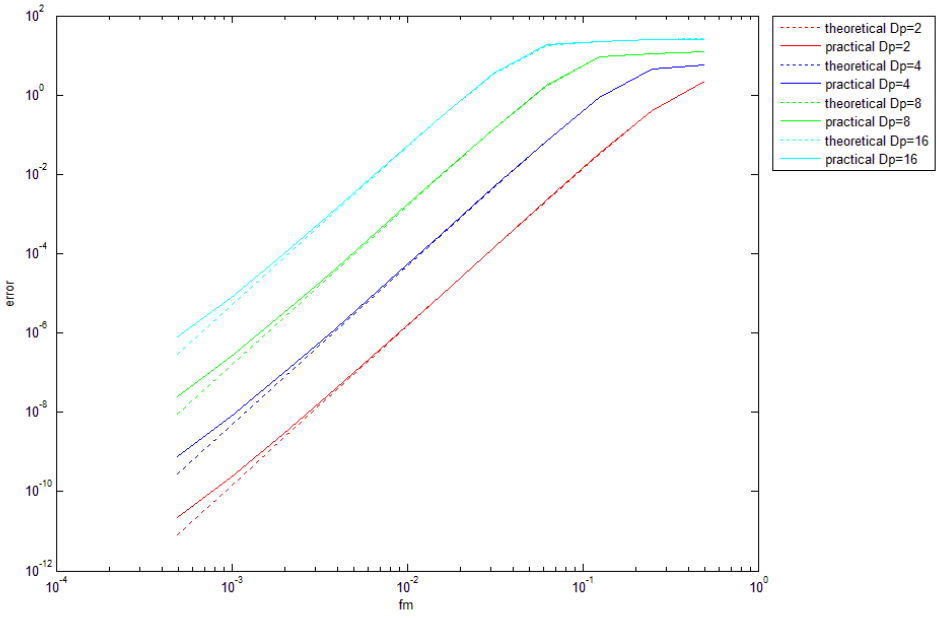


**Figure 4.8** Error power vs. normalized Doppler frequency.

Figure 4.8 shows the error power for the theoretical and practical channel estimation when the separation between pilots, denoted as Dp, is 2,4,8 and 16, the normalized Doppler frequency spans from $10^{-4}$ to 1. The bigger the Doppler frequency the larger the variation in the channel is and consequently more pilots are needed.

When noise is added to the system we obtain the results of Figure 4.9

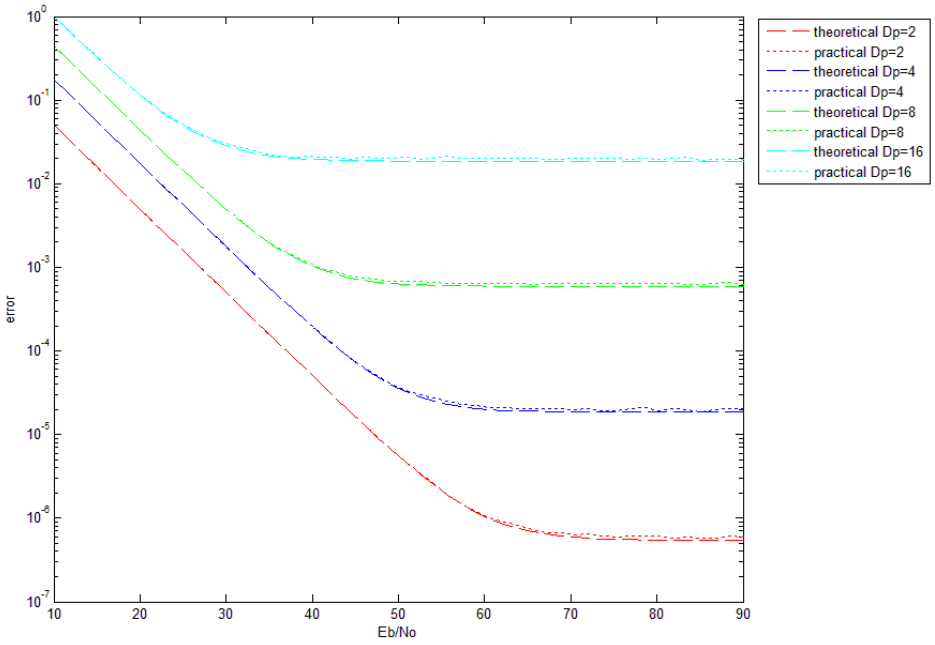


**Figure 4.9** Error power vs. Eb/No (dB) for 250 iterations and a normalized Doppler frequency of 16/2048

In Figure 4.9 we observe that at larger SNR values the better the channel estimation is. However, there is an error floor which represents the systematic error of estimating the non-linear function of the channel by the linear one of the least squares channel estimation that we have used.

At smaller Eb/N0's the difference between the errors of the different pilot distances is smaller. This fact is due that the noise only changes one term of the equation (3.13) when the noise is small the effect of this term is negligible whereas at low Eb/N0's this term has a considerable impact on the accuracy of the channel estimation.

Finally we have to consider that to obtain the theoretical expressions we have assumed that the expected values of the correlation and the noise power are:

$$E\left\{ h_{nD_p}\, {h_{nD_p+k}}^{*} \right\} = R(k) \tag{4.1}$$

$$E\left\{n_{nD_p} {n_{nD_p}}^*\right\} = \sigma^2 \tag{4.2}$$

So, it is necessary to average a large number of samples to obtain close expressions to theoretical magnitudes of the error power. Otherwise, we are doing an error due to a reduced number of samples.

# CHAPTER 5. SYSTEM GENERATOR

In this chapter the linear interpolator channel estimator is going to be implemented in a FPGA using System generator.

System generator is a way of programing Xilinx FPGA where you can automatically generate HDL (Hardware description language) code for FPGAs from Simulink models. You create models of algorithms for implementation in FPGAs using high-level components from Xilinx-specific block sets.

The FPGA used is a VIRTEX-5 FPGA ML507 with chip XC5VFX70T that has the following characteristics

**Table 5.1** Some VIRTEX-5 FPGA ML507 specifications

| | |
|---|---|
| Chip | XC5VFX70T |
| Linear Flash | 256 MB |
| DDR2 SODIMM | 256 MB |
| Watch Frequency | 100Hz |
| ZBT SRAM | 1MB |
| JTAG programming interface | |
| External clocking (2 differential pairs) | |
| RS-232 (Male) – serial port | |
| GPIO DIP switch (8), LEDs (8), and push buttons (5) | |

When programing in System Generator we have to take into account that some parameters are asked and need to be studied to be sure they meet the characteristics we are looking for,

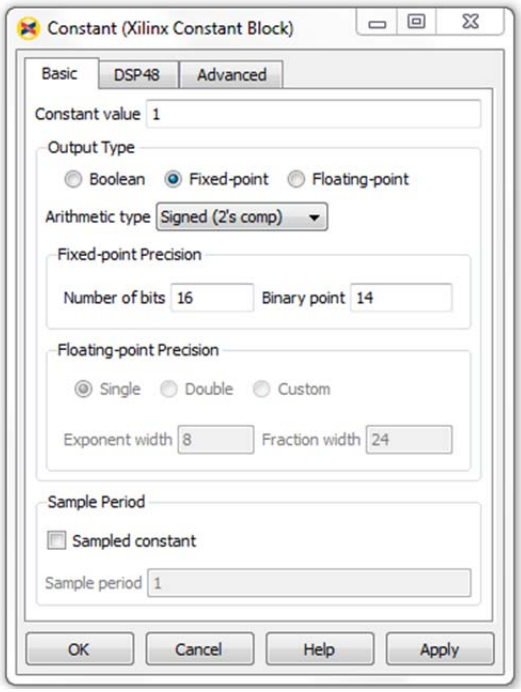


**Figure 5.1** Example of configuration in a System generator block

Some of these shown in Figure 5.1 are the number of bits, where is our binary point, if there is one, if the signal is a Boolean, Fixed-point or floating-point etc... The number of bits is the most important parameter, the bigger the number of bits the slower the system but less quantification error.

First some of the most significant blocks are explained. Then the modifications needed to implement the system in a Co-simulation environment and results are discussed.

## 5.1 Implementation

The system is composed by the frame generator, where the pilots and zeros are added, by the Channel generator, where the Rayleigh variables are obtained and finally the channel estimator where the estimated channel is calculated using the linear interpolator channel estimator.

The full system has been created in system generator except the filter used to obtain the values of the Rayleigh channel. The system is created to get these values from Simulink once and keep them in a memory inside system generator. The values of the filter are independent on the pilot distance and only change with the length of the frame and Doppler frequency.

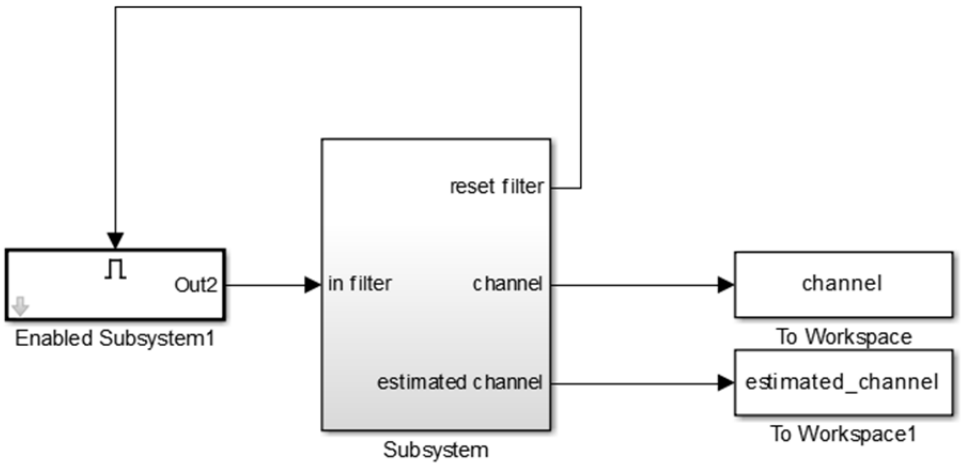


**Figure 5.2** Full system in System Generator.

In Figure 5.2 , the Subsystem block is where the entire System Generator components are found, there are three outputs, two are the values used to calculate the error power of the estimated channel and the other one enables the Subsystem containing the calculations for the filter coefficients.

Some components in System Generator add a delay to the system, which have to be taken into account when implementing the system. The block that adds more delay is the FFT block, because it has to wait until all the values of the length of the FFT have arrived and then do the calculations.

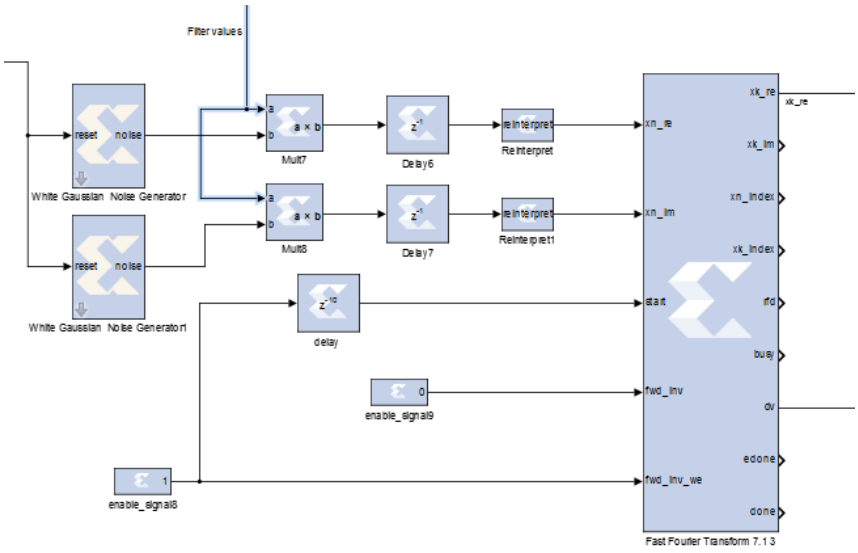


**Figure 5.3** Channel implementation blocks

To obtain the IFFT from the Fast Fourier Transform block the pin *fwd_inv* has to be put to cero, and the *fwd_inv_we* pin needs to be one. The delay in front of the *start* pin is added to compensate the delay that the Gaussian Noise

Generators that are used to generate the random Gaussian variables, needs to start given valid output values.

In the FFT block the pin *xn_re* and *xn_Im* are the real and imaginary component of the input data. In the FPGA the imaginary and real values are treated independently. The input data has to be a single precision floating-point, to force this situation the reinterpret block is added.

Between waiting for all the signals to arrive and process them the FFT block has a big delay. To make sure that all the data send to the receptor passes through correct channel coefficients, all transmission system and frame generation blocks are activated by the pin *dv* (data valid). This pin is found in the output of the FFT block and indicates when the data at the output is valid.

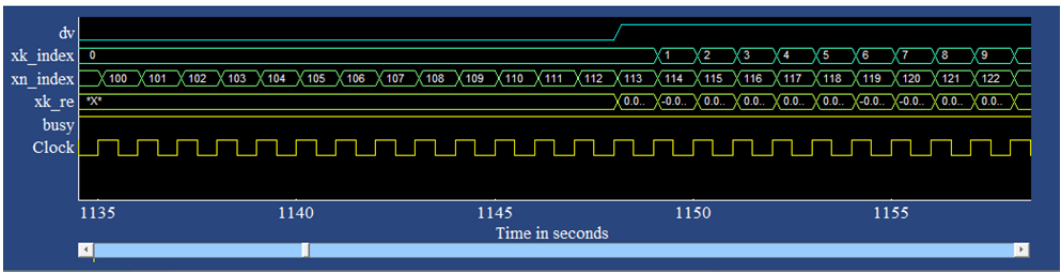


**Figure 5.4** Fast Fourier Transform output pins

In Figure 5.4 we can observe some of the output pins of the Fast Fourier Transform, *xn_index* indicates the index of the input signal, this is used to make sure that the values that enter the system are situated in the correct position inside the FFT block, in our case this was not true and in Figure 5.3 before the reinterpret block a delay of one was needed.

To generate the frame with the pilots, data and ceros a switch is used. Depending on the value of the pin *sel*, one or another entrance is selected to go through to the output.

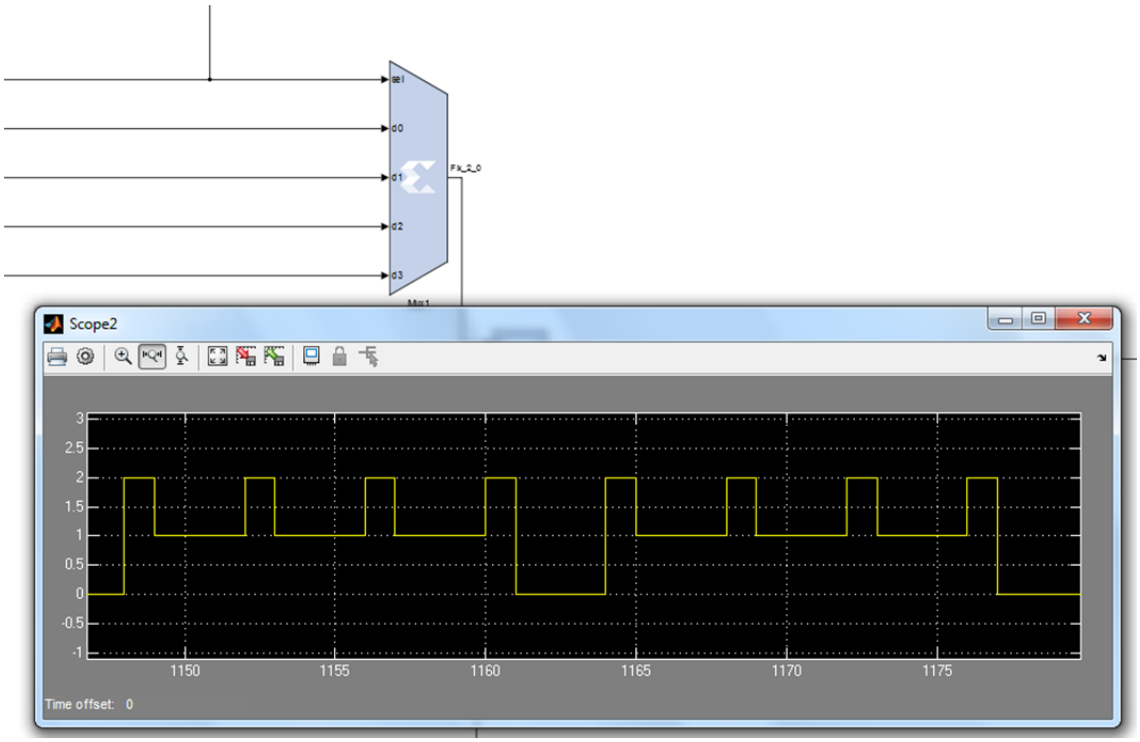


**Figure 5.5** Example for a frame length of 13 samples, channel length of 16 samples and pilot distance of 4.

In Figure 5.5 when the signal is two, the switch will select the pilots, when it is one, the data is selected and when it is zero the output is the zero padding, used to match the frame length to the channel length.

The division operation in FPGA´s is a very complex operation that consumes a lot of resources and it is very inefficient. So, in order to the estimate the channel we will avoid them as much as possible. If particular, if the expression of the linear interpolator for obtaining the channel estimates is:

$$\hat{h}_{nD_p+k} = \hat{h}_{nD_p} + \left(\hat{h}_{(n+1)D_p} - \hat{h}_{nD_p}\right)\frac{k}{D_p} \tag{5.1}$$

To estimate the channel with the linear interpolation channel estimator a division operation is needed see formula (5.1), the values of $D_p$ analysed are 2, 4, 8, and 16, that are multiples of $2\hat{}n$ . Instead of using the division block the property of shifting is used, to shift to the right one position is like dividing by two, shifting two positions is like dividing by four, so on [6].

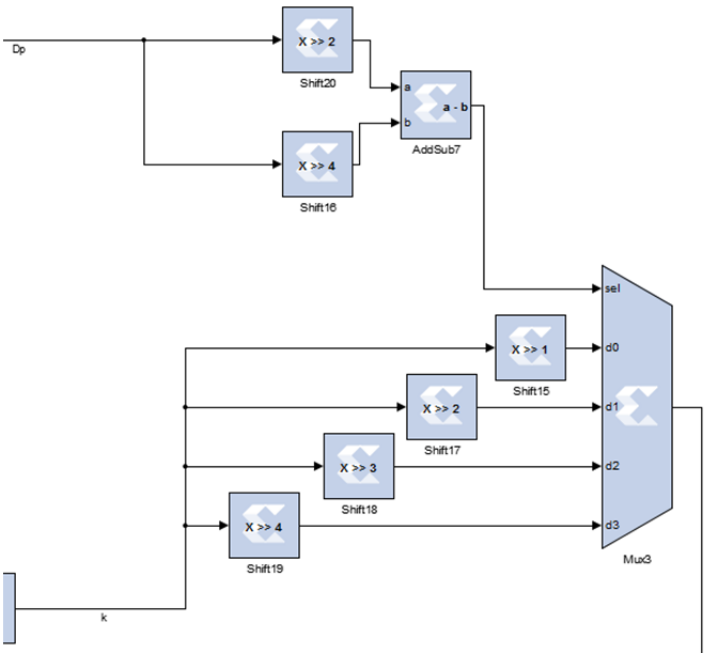


**Figure 5.6** Division by shifting depending on the Dp value.

## 5.2 Co-Simulation with FPGA

Co-simulation is needed when the FPGA has to exchange values with Matlab, in our case there are four values to exchange, two input signal, the pilot distance and the coefficients of the filter, and two output signal the channel and estimated channel. These last ones are needed to compute the error power.

The FPGA works at a very high speed, this speed is so high that the computer is not able of registering all the output values. In order to assure a correct interchange of data between the FPGA and Matlab is necessary to introduce some changes in the system.

One of the biggest changes is related on how the values of the filter are obtained by System Generator from Simulink and how a FIFO is needed to make sure we do not lose values of the channel and channel estimator.

Instead of putting these changes directly to the final version, a smaller system is implemented. This system focuses in how the values of the filter that are created in Simulink are passed to the FPGA and how the results can be visualized in the computer.

In co-simulation environments the system is divided in two, the blocks that are compiled and put inside the FPGA, and the blocks used to interact with the FPGA [7]. Figure 5.7 shows the blocks that are going to be compiled and inserted inside the FPGA.

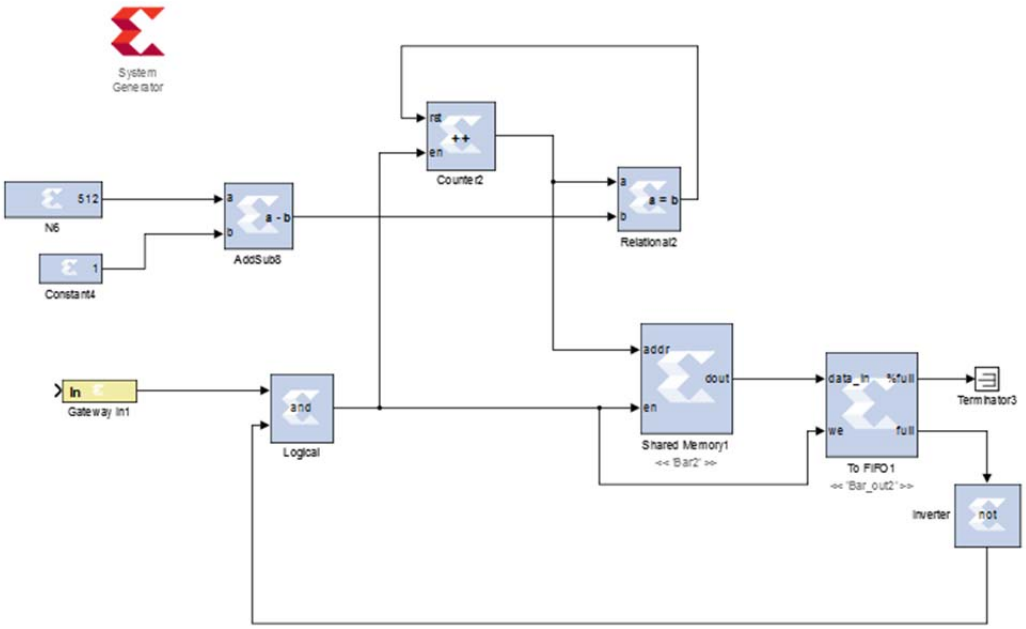


**Figure 5.7** Input Blocks

The system in Figure 5.7 has a counter that reads the values found in the Shared Memory *`Bar2´*. In our case every 512 samples the counter resets and starts to count from the begging. This means that the shared memory output values are repeated every 512 samples.

There is a control signal that comes from the 'Gateway In1', this signal is used to enable the Shared Memory, FIFO and the counter. This signal is sent to make sure that the FPGA does not start to read values of the *`Bar2´* memory before they have been calculated and written by the channel filter created in Simulink.

The output values are not sent directly to Simulink, they are sent to a FIFO where they are saved to a memory called *`Bar_out2´*, this is done to keep the values coherent when they have to be read by Simulink.

To make sure that the values at the output are always coherent a feedback from the pin *full* in the FIFO is needed. This pin gives has a high value when the memory in the FIFO is full, when this happens the enable port is put to zero in the counter, shared memory and no more data is written to the FIFO. If this feedback is not taken into account the output signal would look like the one shown in Figure 5.8.

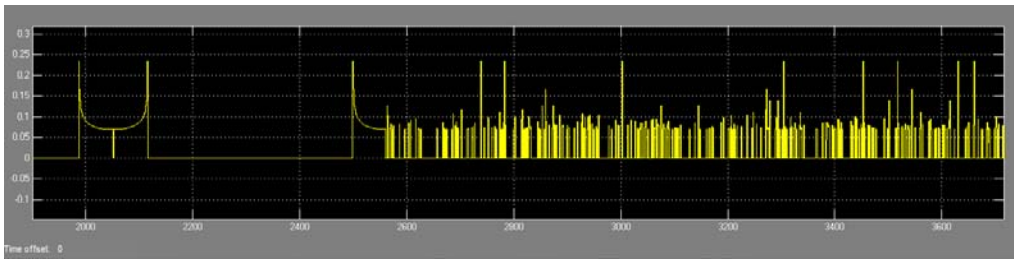

**Figure 5.8** Filter output if no feedback is connected to the full port of the FIFO.

At Figure 5.8 the FIFO is able to save the first values but when the memory is finish the values saved make no sense, this is because the FPGA delivered the output values faster than the computer could read them.

After compiling the system of Figure 5.7, System Generator returns a block like the one on Figure 5.9.


**Figure 5.9** Co-simulation System Block

This block is put into the output system where the Memory containing the channel variables is written into. And it is also where the FIFO containing the output values is read. The other Blocks of the system are used to indicate the addresses where the values of the filter are written to

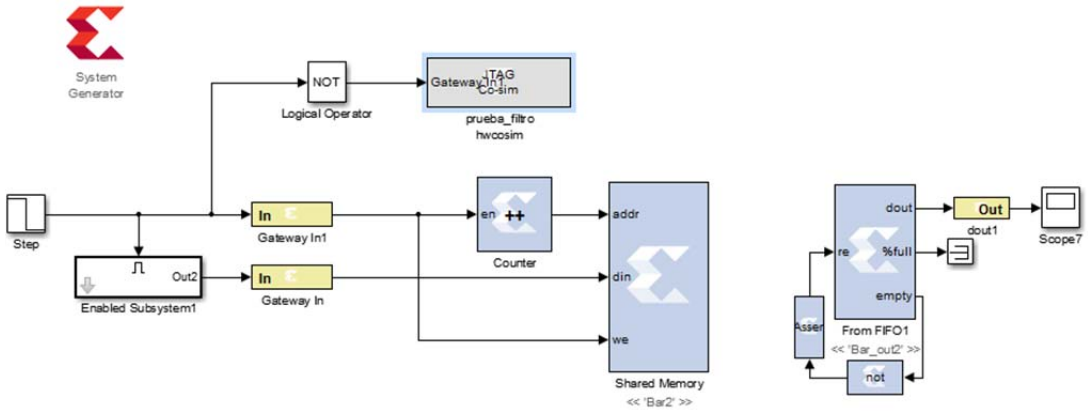


**Figure 5.10** Output Block

## 5.3 Final system

The changes needed for a co-simulation are added to the final system.

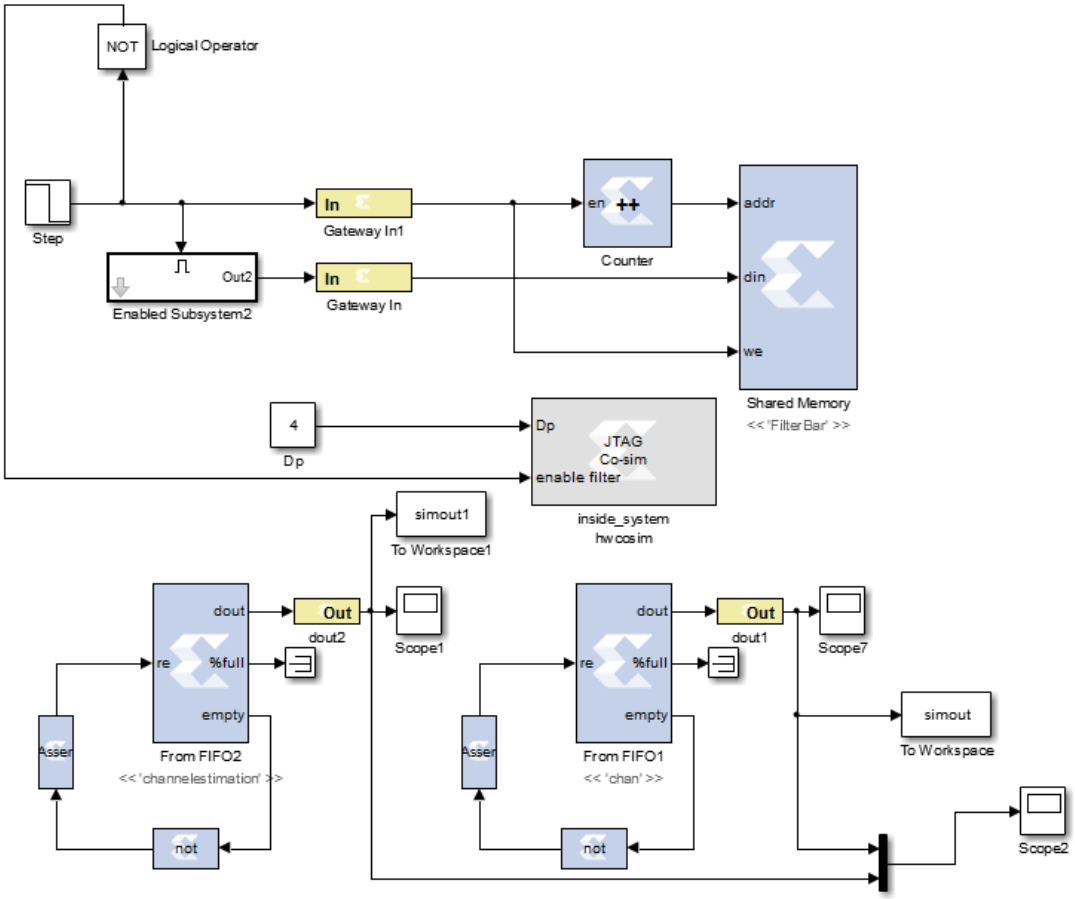


**Figure 5.11** Final scheme

The JTAG Co-sim block contains all our channel estimator system. The Shared Memoty `*FilterBar*´ is where the parameters of the filter are saved, and FIFO2 and FIFO1 contain the results of the channel estimation and the real channel respectively.

Our system the Doppler frequency and the number of bits can be changed without the need to compile the system again. So, the FPGA is reconfigurable without remapping it again, which saves a lot of simulation time.

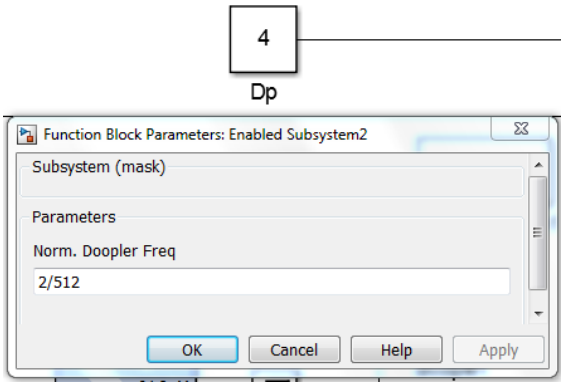


**Figure 5.12** Reconfigurable system parameters

To make sure that the system is going to work properly with no errors, the maximum frequency of the system has to be checked,

```
Timing Summary:
---------------
Speed Grade: -1

   Minimum period: 20.876ns (Maximum Frequency: 47.902MHz)
   Minimum input arrival time before clock: 3.982ns
   Maximum output required time after clock: 17.198ns
   Maximum combinational path delay: 17.076ns
```

**Figure 5.13** Maximum frequency for the system with FIFOS of 4K locations

```
Timing Summary:
---------------
Speed Grade: -1

   Minimum period: 0.525ns (Maximum Frequency: 1904.762MHz)
   Minimum input arrival time before clock: 4.694ns
   Maximum output required time after clock: 1.280ns
   Maximum combinational path delay: 3.380ns
```

**Figure 5.14** Maximum frequency for the system with FIFOS of 1K locations

The time that the longest path on our system takes to be solved is the minimum period, with this value we obtain the maximum frequency our system can work. The ML507 FPGA has a velocity of 100 MHz, this frequency is too big for a system with FIFOS of 4k locations and it can work for a FIFO of 1k location. The FIFO location affects to the length of the path because this memory is divided into block of memory, more blocks of memory mean more connections between blocks and this affects the length of the paths.

The resources used in our FPGA are described in the Device utilization summary.

```
Device utilization summary:
---------------------------

Selected Device : 5vfx70tff1136-1


Slice Logic Utilization:
 Number of Slice Registers:            5006  out of  44800    11%
 Number of Slice LUTs:                 3981  out of  44800     8%
    Number used as Logic:              2827  out of  44800     6%
    Number used as Memory:             1154  out of  13120     8%
       Number used as RAM:               64
       Number used as SRL:             1090

Slice Logic Distribution:
 Number of LUT Flip Flop pairs used:   8987
   Number with an unused Flip Flop:    3981  out of   8987    44%
   Number with an unused LUT:          5006  out of   8987    55%
   Number of fully used LUT-FF pairs:     0  out of   8987     0%
   Number of unique control sets:        76

IO Utilization:
 Number of IOs:                         109
 Number of bonded IOBs:                   0  out of    640     0%

Specific Feature Utilization:
 Number of Block RAM/FIFO:                6  out of    148     4%
    Number using Block RAM only:          6
 Number of DSP48Es:                      24  out of    128    18%
```

**Figure 5.15** Device utilization summary

The device utilization summary is very important, because it tells us the memory and resources the system implemented is consuming, our system should be the more efficient as possible.

When comparing the results of error power obtained in System Generator and Simulink for the lineal channel estimator, some differences are observed. The results are calculated for a different number of pilot distance and Normalized Doppler frequency.

**Table 5.2** Results obtained in Simulink

| Simulink | $Dp$=2 | $Dp$=4 | $Dp$=16 |
|---|---|---|---|
| $Fm$=2/512 | 6.1382e-8 | 1.0995e-6 | 2.636e-4 |
| $Fm$=8/512 | 1.0663e-5 | 1.8032e-4 | 0.0423 |
| $Fm$=16/512 | 1.414e-4 | 0.0024 | 0.4395 |

**Table 5.3** Results obtained in System Generator

| System Generator | $Dp$=2 | $Dp$=4 | $Dp$=16 |
|---|---|---|---|
| $Fm$=2/512 | 5.8973e-8 | 2.0953e-6 | 0.0022 |
| $Fm$=8/512 | 1.18144e-5 | 4.1288e-4 | 0.3740 |
| $Fm$=16/512 | 2.033e-4 | 0.0057 | 1.6788 |

The biggest difference between Table 5.3 and Table 5.2 is found when $Dp$=16 this is due that this value is the sum of $(Dp-1)$ data point, each with its own

error. Smaller $Dp$ smaller the sum of error of data points and less difference there is.

The difference is due to the quantification error, in Simulink the system is working at double precision, this means 64 bits, in our system each of the block works with a different number of bits, for example the values of the filter are 16 bits, the random variables 12 bits and the division between $k/Dp$ only 8 bits. The less bits used the less time needed for the FPGA to compute but more quantification error.

When talking about time, no comparison between the systems has been done because the approach of using co-simulation and obtaining the values for calculating the error power in Matlab has made our system so low that it takes less time without the FPGA. This does not mean that the FPGA is slow, this means that the connection between our computer and the FPGA is blocking all the speed. In our case the 12 MHz.

```
Type = 0x0005.
 Cable Type = 3, Revision = 0.
 Setting cable speed to 12 MHz.
Cable connection established.
Firmware version = 2401.
File version of C:/Xilinx/14.3/ISE_DS/ISE/data/xusb_xp2.hex = 2401.
Firmware hex file version = 2401.
PLD file version = 200Dh.
 PLD version = 200Dh.
Type = 0x0005.
ESN option: 00001322BFFE01.
```

**Figure 5.16** Velocity of the connection between the computer and the FPGA

To solve this issue the FPGA should be connected by means of an Ethernet link. Thus we would have given us a faster transmission between both of them. However, this FPGA does not support this option. The other option to solve this inconvenient, may be the introduction of the full system into the FPGA. In this way, there is no time-loss due to the interchange of data between Matlab and the FPGA.

If this last method was implemented the system should have been configured to do the average of the error power every pilot distance, it is very important to take into account the number of bits of the system, the average has to be done in a way it never exceeds them. In this case to observe different parts of the system an output port can be connected but instead of going to Matlab this port can send information to different pins on the FPGA (each pin corresponding to a bit), can also sent information to a led, useful for control signals. An oscilloscope can also be connected, if these pins are connected to a Digital to analog converter first. All this pins can be configured in the .ucf (user configuration file) of the board.

# CHAPTER 6. CONCLUSION

After the evaluation of the three methods for generating correlated Rayleigh fading generation, the sum of sines, the Smith's strategy and the Bealieu-Young technique we conclude that the Bealieu-Young method provides the lowest complexity for the same accuracy.

If the channel has a slow fading the best way of channel positioning is at the beginning of the frames, since the channel is quite constant and has the lowest complexity. However, if the system has almost no noise it is enough to take only one pilot to estimate the channel. On the contrary, if there is a lot of noise the best approach is to get the average of the channel estimations.

If the variation of the channel is fast fading, then the pilots have to be positioned in an uniform way along the frames. If there is noise the best method is the average channel estimator and when the noise is small the best one is the linear interpolator channel estimator.

The larger the number of pilots, the better the estimation is. However, it also means that when we use more pilots, it is required to do more computations, and the throughput of the system is reduced since we send a lower number of payload information. Also we have to take into account that the pilot estimation process is only a part of a real communication system, and that the final decision has to be made with all the other components taken into account.

The implementation of the channel generation and estimation has been done in Matlab and Simulink. Next, their results have been compared with the theoretical ones, concluding that they are almost identical if the number of iterations in the channel estimation and generation are used.

In a co-simulation environment the FPGA works with Matlab. The results show that this is not the best method, because the system is been limited by the velocity of the connection between both the components. The best way to take advantage of the power of the FPGA is to make the full system inside and just obtain a final result. Although co-simulation is not the best way to obtain data steadily it can be useful to introduce data inside the system.

# REFERENCES

[1] Rahman Q.M. and Ibnkahla M. *Signal Processing for future Mobile Communications Systems*: Callenges and Perspectives. 2005

[2] David J. Young and Norman C. Beaulieu, ”*The Generation of Correlated Rayleigh Random Variates by Inverse Discrete Fourier Transform”,* IEEE Transactions on communications, vol. 48, no. 7, pp.14, July 2000.

[3] Chengshan Xiao, Yahong R. Zheng, and Norman C. Beaulieu, “*Statistical Simulation Models for Rayleigh and Rician Fading”*

[4] Devroye N. “*Least Squares Estimation*”, Estimation: chapter 8, pp 10 Spring 2010.

[5] Prasanta Kumar Pradhan, Oliver Fausty, Sarat Kumar Patra and Beng Koon Chuay, *“Channel Estimation Algorithms for OFDM Systems”,* Dept. of Electrons & Comm. Engg. National Institute of Technology.

[6] DSP Desigh Flow. Lab Manual. Xilinx University Program. Creating a 12 x 8 MAC Using VHDL.

[7] System Generator for DSP User Guide. Chapter 3. Using Hardware Co-Simulation. Xilnix.

,

# ANNEX

**Mathematical analysis of the linear interpolation channel estimator.**
The main parameters of this interpolator are:

$k$ separation of the position of the data to the pilot
$D_p$ separation between pilots
$\hat{h}_{nD_p}$ estimated channel of the n-th pilot
$n_{nD_p}$ noise in the n-th pilot

The estimated channel in two consecutive pilots is

$$\hat{h}_{nD_p} = \left(h_{nD_p} + n_{nD_p}\right)$$
$$\hat{h}_{(n+1)D_p} = \left(h_{(n+1)D_p} + n_{(n+1)D_p}\right)$$

If we use a linear interpolator, the channel estimation at the k-th subcarrier is:

$$\hat{h}_{nD_p+k} = \hat{h}_{nD_p} + \left(\hat{h}_{(n+1)D_p} - \hat{h}_{nD_p}\right)\frac{k}{D_p}$$

Being the error power equal to:

$$E\left\{\left|e_{h_{nD_p+k}}\right|^2\right\} = E\left\{\left|\hat{h}_{nD_p+k} - h_{nD_p+k}\right|^2\right\}$$
$$= E\left\{\left(\hat{h}_{nD_p+k} - h_{nD_p+k}\right)\left(\hat{h}_{nD_p+k} - h_{nD_p+k}\right)^*\right\}$$
$$= E\left\{\left(\hat{h}_{nD_p+k}\,\hat{h}_{nD_p+k}{}^* - \hat{h}_{nD_p+k}\,h_{nD_p+k}{}^* - h_{nD_p+k}\,\hat{h}_{nD_p+k}{}^*\right.\right.$$
$$\left.\left. + h_{nD_p+k}\,h_{nD_p+k}{}^*\right)\right\}$$

The equation is separated in four steps called $I_1$,$I_2$,$I_3$ and $I_4$

$$I_1 = E\left\{\hat{h}_{nD_p+k}\,\hat{h}_{nD_p+k}{}^*\right\}$$

$$= E\left\{\left(\hat{h}_{nD_p} + \left(\hat{h}_{(n+1)D_p} - \hat{h}_{nD_p}\right)\frac{k}{D_p}\right)\left(\hat{h}_{nD_p}{}^* + \left(\hat{h}_{(n+1)D_p} - \hat{h}_{nD_p}\right)^*\frac{k}{D_p}\right)\right\}$$

$$= \left(\frac{D_p k - k^2}{{D_p}^2}\right)2Re\,E\left\{\hat{h}_{nD_p}\,\hat{h}_{(n+1)D_p}{}^*\right\} + \left(1 - \frac{2k}{D_p} + \frac{k^2}{{D_p}^2}\right)E\left\{\hat{h}_{nD_p}{}^*\hat{h}_{nD_p}\right\}$$
$$+ E\left\{\hat{h}_{(n+1)D_p}\hat{h}_{(n+1)D_p}{}^*\right\}\frac{k^2}{{D_p}^2}$$

$$= \left(\frac{k^2}{{D_p}^2}\right)(R(0)+\sigma^2) + 2Re\left(\frac{D_p k - k^2}{{D_p}^2}\right) R(D_p) + \left(1 - \frac{2k}{D_p} + \frac{k^2}{{D_p}^2}\right)(R(0)+\sigma^2)$$

$$= 2Re\left(1-\frac{k}{D_p}\right)\frac{k}{D_p} R(D_p) + \left(\left(1-\frac{k}{D_p}\right)^2 + \frac{k^2}{{D_p}^2}\right)(R(0)+\sigma^2)$$

Whereas the second summatory, $I_2$, is :

$$I_2 = E\left\{-\hat{h}_{nD_p+k}\, {h_{nD_p+k}}^*\right\}$$
$$= E\left\{-\left(\hat{h}_{nD_p} + \left(\hat{h}_{(n+1)D_p} - \hat{h}_{nD_p}\right)\frac{k}{D_p}\right) {h_{nD_p+k}}^*\right\}$$
$$= E\left\{-\hat{h}_{nD_p}\ {h_{nD_p+k}}^* - \left(\hat{h}_{(n+1)D_p} - \hat{h}_{nD_p}\right)\frac{k}{D_p}\ {h_{nD_p+k}}^*\right\}$$
$$= E\left\{-\hat{h}_{nD_p}\ {h_{nD_p+k}}^* - {h_{nD_p+k}}^* \hat{h}_{(n+1)D_p}\frac{k}{D_p} + {h_{nD_p+k}}^* \hat{h}_{nD_p}\frac{k}{D_p}\right\}$$
$$= -R(k)\left(1-\frac{k}{D_p}\right) - R(D_p-k)\frac{k}{D_p}$$

After that the third component to analyze, $I_3$, can be decomposed as :

$$I_3 = E\left\{-h_{nD_p+k}\, {\hat{h}_{nD_p+k}}^*\right\} = E\left\{-h_{nD_p+k}\left(\hat{h}_{nD_p} + \left(\hat{h}_{(n+1)D_p} - \hat{h}_{nD_p}\right)\frac{k}{D_p}\right)^*\right\}$$

$$= -h_{nD_p+k}\, {\hat{h}_{nD_p}}^* - \frac{k}{D_p} h_{nD_p+k}\, {\hat{h}_{(n+1)D_p}}^* + \frac{k}{D_p} h_{nD_p+k}\, {\hat{h}_{nD_p}}^*$$
$$= -R(k)^*\left(1-\frac{k}{D_p}\right) - R(D_p-k)^*\frac{k}{D_p}$$

Whereas that the last summitry, $I_4$, corresponds to the autocorrelation of the channel:

$$I_4 = E\left\{h_{nD_p+k}\, {h_{nD_p+k}}^*\right\} = R(0)$$

Finally, if we group the four components, we obtain:

$$\sum_{k=1}^{D_p-1} I_k = \sum_{k=1}^{D_p-1} 2Re\left(1-\frac{k}{D_p}\right)\frac{k}{D_p} R(D_p) + \left(\left(1-\frac{k}{D_p}\right)^2 + \frac{k^2}{{D_p}^2}\right)(R(0)+\sigma^2)$$
$$- R(k)\left(1-\frac{k}{D_p}\right) - R(D_p-k)\frac{k}{D_p} - R(k)^*\left(1-\frac{k}{D_p}\right) - R(D_p-k)^*\frac{k}{D_p}$$
$$+ R(0)$$

Which can be simplified to:

$$= Re\left(\frac{{D_p}^2-1}{3D_p}\right)R(D_p) - 2Re\left\{\sum_{k=1}^{D_p-1}\left(\frac{D_p-k}{D_p}\right)R(k) + \left(\frac{k}{D_p}\right)R(D_p-k)\right\} + (D_p-1)R(0) + (R(0)+\sigma^2)\,\frac{(D_p-1)(2D_p-1)}{3D_p}$$

If we consider the error in the pilots, then the total error will be:

$$Re\left(\frac{{D_p}^2-1}{3D_p}\right)R(D_p) - 2Re\left\{\sum_{k=1}^{D_p-1}\left(\frac{D_p-k}{D_p}\right)R(k) + \left(\frac{k}{D_p}\right)R(D_p-k)\right\} + (D_p-1)R(0) + (R(0)+\sigma^2)\,\frac{(D_p-1)(2D_p-1)}{3D_p} + \sigma^2$$

**Mathematical analysis for the error power that commits the average channel estimator**

The average channel estimator has the next expression

$$\hat{h}_{nDp+k} = \alpha_1\hat{h}_{nDp} + \alpha_2\hat{h}_{(n+1)Dp}$$

Being the error power equal to

$$E\left\{\left|e_{h_{nDp+1}}\right|^2\right\} = \left(h_{nDp+k} - \left(\alpha_1\hat{h}_{nDp} + \alpha_2\hat{h}_{(n+1)Dp}\right)\right)\left(h_{nDp+k} - \left(\alpha_1\hat{h}_{nDp} + \alpha_2\hat{h}_{(n+1)Dp}\right)\right)^*$$

The system can be separated in the following sums:

$$I_1 = E\left\{\left(\alpha_1\hat{h}_{nDp} + \alpha_2\hat{h}_{(n+1)Dp}\right)\left(\alpha_1\hat{h}_{nDp} + \alpha_2\hat{h}_{(n+1)Dp}\right)^*\right\}$$

$$I_2 = E\left\{-\left(\alpha_1\hat{h}_{nDp} + \alpha_2\hat{h}_{(n+1)Dp}\right)\left({h_{nDp+k}}^*\right)\right\}$$

$$I_3 = {I_2}^*$$

$$I_4 = E\left\{h_{nDp+k}\,{h_{nDp+k}}^*\right\}$$

Which have the next expressions

$$I_1 = E\left\{\left(\alpha_1\hat{h}_{nDp} + \alpha_2\hat{h}_{(n+1)Dp}\right)\left(\alpha_1\hat{h}_{nDp} + \alpha_2\hat{h}_{(n+1)Dp}\right)^*\right\}$$

$$= \mathrm{E}\left\{(|\alpha_1|^2 + |\alpha_2|^2)(\mathrm{R}(0)+\sigma^2) + 2\mathrm{Re}\left\{\alpha_1{\alpha_2}^*R(-D_p)\right\}\right\}$$

$$I_2 = -\left(\alpha_1\hat{h}_{nDp} + \alpha_2\hat{h}_{(n+1)Dp}\right)\left({h_{nDp+k}}^*\right) = -\mathrm{R(k)}$$

$$I_3 = -\mathrm{R(k)}^*$$

$$I_4 = \mathrm{R(0)}$$

Finally, if we sum all of them , we obtain

$$(|\alpha_1|^2 + |\alpha_2|^2)(\mathrm{R(0)} + \sigma^2) \sum_{\mathrm{k=1}}^{D_p-1} 1 + 2\mathrm{Re}\{\alpha_1 \alpha_2{}^* R(-D_p)\} \sum_{\mathrm{k=1}}^{D_p-1} 1 - 2\mathrm{Re} \sum_{\mathrm{k=1}}^{D_p-1} \mathrm{R(k)} + \mathrm{R(0)} \sum_{\mathrm{k=1}}^{D_p-1} 1$$

$$= (|\alpha_1|^2 + |\alpha_2|^2)(\mathrm{R(0)} + \sigma^2)(D_p - 1) + 2\mathrm{Re}\{\alpha_1 \alpha_2{}^* R(-D_p)\}(D_p - 1) - 2\mathrm{Re} \sum_{\mathrm{k=1}}^{D_p-1} \mathrm{R(k)} + \mathrm{R(0)}(D_p - 1)$$

Next, if we consider that the sum of the weighting factors is equal to one,:

$$\alpha_1 + \alpha_2 = 1$$
$$\alpha_2 = 1 - \alpha_1$$

The expression of the correlation is equal to:
$R_h(\mathrm{k}) = \alpha_1 R_h(0) + \alpha_2 R_h(\mathrm{D_p}) = \alpha_1 R_h(0) + (1 - \alpha_1) R_h(\mathrm{D_p})$

Whereas the expression of the weighting factors in terms of the correclation functions is:

$$\alpha_1 = \frac{R_h(\mathrm{k}) - R_h(\mathrm{D_p})}{R_h(0) - R_h(\mathrm{D_p})}$$
$$\alpha_2 = \frac{R_h(0) - R_h(\mathrm{k})}{R_h(0) - R_h(\mathrm{D_p})}$$

If the sum of the weighting factors is not equal to the unity:

$$\alpha_1 + \alpha_2 \neq 1$$

Then the correlation function is equal to:

$$R_h(\mathrm{k}) = \alpha_1 R_h(0) + \alpha_2 R_h(\mathrm{D_p})$$
$$R_h(\mathrm{k} - \mathrm{D_p}) = \alpha_1 R_h(-\mathrm{D_p}) + \alpha_2 R_h(0)$$

Which can be expressed in matrix form as follows:

$$\begin{bmatrix} R_h(0) & R_h(\mathrm{D_p}) \\ R_h(-\mathrm{D_p}) & R_h(0) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} R_h(\mathrm{k}) \\ R_h(\mathrm{k} - \mathrm{D_p}) \end{bmatrix}$$

Resulting the next expressions for the weighting factors:

$$\alpha_1 = \frac{\begin{pmatrix} R_h(\mathrm{k}) & R_h(\mathrm{D_p}) \\ R_h(\mathrm{k}-\mathrm{D_p}) & R_h(0) \end{pmatrix}}{D} = \frac{R_h(\mathrm{k})R_h(0) - R_h(\mathrm{k}-\mathrm{D_p})R_h(\mathrm{D_p})}{|R_h(0)|^2 - |R_h(\mathrm{D_p})|^2}$$

$$\alpha_2 = \frac{\begin{pmatrix} R_h(0) & R_h(\mathrm{k}) \\ R_h(-\mathrm{D_p}) & R_h(\mathrm{k}-\mathrm{D_p}) \end{pmatrix}}{D} = \frac{R_h(0)R_h(\mathrm{k}-\mathrm{D_p}) - R_h(-\mathrm{D_p})R_h(\mathrm{k})}{|R_h(0)|^2 - |R_h(\mathrm{D_p})|^2}$$

**Mathematical analysis of the error power when the pilots are placed at the beginning of the frames and it is considered one pilot value.**

The first analysis is done when the channel estimator is estimated with only one pilot value.

$$\hat{h}_k = \hat{h}_{\left\lceil \frac{Np}{2} \right\rceil}$$
$$\hat{h}_{\frac{Np}{2}} = h_{\left\lceil \frac{Np}{2} \right\rceil} + n_{\left\lceil \frac{Np}{2} \right\rceil}$$

The error power of the data subcarriers when it is assigned the channel at the average position of the pilots is

$e_{h_k} = \hat{h}_k - h_k$
$E\left\{\left|e_{h_k}\right|^2\right\} = E\{(\hat{h}_k - h_k)(\hat{h}_k - h_k)^*\} = E\{\hat{h}_k\hat{h}_k{}^* - \hat{h}_k h_k{}^* - h_k\hat{h}_k{}^* + h_k h_k{}^*\} =$
$E\left\{\hat{h}_{\left\lceil \frac{Np}{2} \right\rceil}\hat{h}_{\left\lceil \frac{Np}{2} \right\rceil}{}^* - \hat{h}_{\left\lceil \frac{Np}{2} \right\rceil}h_k{}^* - \hat{h}_{\left\lceil \frac{Np}{2} \right\rceil}{}^* h_k + h_k h_k{}^*\right\} = E\left\{\hat{h}_{\left\lceil \frac{Np}{2} \right\rceil}\hat{h}_{\left\lceil \frac{Np}{2} \right\rceil}{}^*\right\} - E\left\{\hat{h}_{\left\lceil \frac{Np}{2} \right\rceil}h_k{}^*\right\} -$
$E\left\{\hat{h}_{\left\lceil \frac{Np}{2} \right\rceil}{}^* h_k\right\} + E\{h_k h_k{}^*\} = (R(0) + \sigma^2) - 2\,Re\,E\left\{\hat{h}_{\left\lceil \frac{Np}{2} \right\rceil}h_k{}^*\right\} + R(0) = 2R(0) + \sigma^2 -$
$2\,Re\,R(k - \left\lceil \frac{Np}{2} \right\rceil)$

Now the sum from Np+1 to N is done

$$\sum_{n=Np+1}^{N} (2R(0) + \sigma^2) - \sum_{k=Np+1}^{N} 2\,Re\,R\left(k - \left\lceil \frac{Np}{2} \right\rceil\right)$$
$$= Nd(2R(0) + \sigma^2) - 2Re \sum_{k=Np+1}^{N} R(k - \left\lceil \frac{Np}{2} \right\rceil)$$

The sum from every power error values is done except to the pilot at position $\left\lceil \frac{Np}{2} \right\rceil$ due that the power error in ths position is only caused by the noise
The error for the first half of the pilots is:

$$\sum_{n=1}^{\left\lceil\frac{Np}{2}\right\rceil-1}(2R(0)+\sigma^2)-\sum_{n=1}^{\left\lceil\frac{Np}{2}\right\rceil-1}2\,Re\,R(k-\left\lceil\frac{Np}{2}\right\rceil)$$

$$=\left(\left\lceil\frac{Np}{2}\right\rceil-1\right)(2R(0)+\sigma^2)-2Re\sum_{n=1}^{\left\lceil\frac{Np}{2}\right\rceil-1}R(k-\left\lceil\frac{Np}{2}\right\rceil)$$

And the error power for the second half of the pilots is:

$$\sum_{n=\left\lceil\frac{Np}{2}\right\rceil+1}^{Np}(2R(0)+\sigma^2)-\sum_{n=\left\lceil\frac{Np}{2}\right\rceil+1}^{Np}2\,Re\,R(k-\left\lceil\frac{Np}{2}\right\rceil)$$

$$=\left\lceil\frac{Np}{2}-1\right\rceil(2R(0)+\sigma^2)-2Re\sum_{n=\left\lceil\frac{Np}{2}\right\rceil+1}^{Np}R\left(k-\left\lceil\frac{Np}{2}\right\rceil\right)$$

The total error considering pilots and data is:

$$Nd(2R(0)+\sigma^2)-2Re\sum_{k=Np+1}^{N}R(k-\left\lceil\frac{Np}{2}\right\rceil)+\left(\left\lceil\frac{Np}{2}\right\rceil-1\right)(2R(0)+\sigma^2)$$

$$-2Re\sum_{n=1}^{\left\lceil\frac{Np}{2}\right\rceil-1}R(k-\left\lceil\frac{Np}{2}\right\rceil)+\left\lceil\frac{Np}{2}-1\right\rceil(2R(0)+\sigma^2)$$

$$-2Re\sum_{n=\left\lceil\frac{Np}{2}\right\rceil+1}^{Np}R\left(k-\left\lceil\frac{Np}{2}\right\rceil\right)+\sigma^2$$

**Mathematical analysis of the error power when the pilots are placed at the beginning of the frames and it is considered the average channel of the pilots.**

This analysis is done when the average of all the pilots is used to estimate the channel.

$$\hat{h}_k=\frac{1}{Np}\sum_{n=1}^{Np}\hat{h}_n$$

In this case the error power is

$$e_{h_k}=\hat{h}_k-h_k$$

$$E\left\{\left|e_{h_k}\right|^2\right\}=E\{(\hat{h}_k-h_k)(\hat{h}_k-h_k)^*\}=E\{\hat{h}_k\hat{h}_k{}^*-\hat{h}_k h_k{}^*-h_k\hat{h}_k{}^*+h_k h_k{}^*\}$$

$$= \frac{1}{Np^2} E\left\{\sum_{n=1}^{Np} \hat{h}_n \left(\sum_{n=1}^{Np} \hat{h}_n\right)^*\right\} - 2ReE\left\{\left(\frac{1}{Np}\sum_{n=1}^{Np} \hat{h}_n\right)^* h_k\right\} + R(0)$$

$$= \left(\frac{1}{Np^2}\left(Np(R(0)+\sigma^2) + \sum_{n=1}^{Np-1} 2(Np-n)R(n)\right)\right) - \frac{2}{Np} Re \sum_{n=1}^{Np} R(k-n) + R(0)$$

Adding from the subcarriers k=Np+1 to N , we obtain.

$$= \left(\frac{1}{Np^2}\left(Np(R(0)+\sigma^2) + \sum_{n=1}^{Np-1} 2(Np-n)R(n)\right)\right) Nd$$

$$- \frac{2}{Np} Re \sum_{k=Np+1}^{N} \sum_{n=1}^{Np} R(k-n) + Nd\, R(0)$$

Whereas the error in the pilots will be:

$$\sum_{k=1}^{Np}\left(\left(\frac{1}{Np^2}\left(Np(R(0)+\sigma^2) + \sum_{n=1}^{Np} 2(Np-1)R(n)\right)\right)\right.$$

$$\left. - \frac{2}{Np} Re \sum_{n=1}^{Np} R(k-n) \; + R(0)\right)$$

Which can be grouped as:$R(0)$

$$= \frac{1}{Np}\left(Np(R(0)+\sigma^2) + \sum_{n=1}^{Np} 2(Np-1)R(n)\right)$$

$$- \frac{2}{Np} Re \sum_{k=1}^{Np}\sum_{n=1}^{Np} R(k-n) + Np\left(R(0)\right)$$

And resulting the next error:

$$\frac{Nd}{Np^2}\left(Np(R(0)+\sigma^2)+\sum_{n=1}^{Np-1}2(Np-n)R(n)\right)$$
$$-\frac{2}{Np}Re\sum_{k=1}^{Nd}\sum_{n=1}^{Np}R(k-n)+Nd\left(R(0)\right)$$
$$+\frac{1}{Np}\left(Np(R(0)+\sigma^2)+\sum_{n=1}^{Np}2(Np-1)R(n)\right)$$
$$-\frac{2}{Np}Re\sum_{k=1}^{Np}\sum_{n=1}^{Np}R(k-n)+Np\left(R(0)\right)$$

# ACKNOWLEDGEMENTS

This work could not have been done without Dr. Joan Bas constant support, patience, knowledge and guidance. I am indebted to Dr. Ciprian Gavrincea for teaching me in the intricate world of the FPGA. I am also grateful to Bea Miranda, Maria Lema and Mathew Churchman for always being there so supportive.