



UNIVERSIDAD  
POLITECNICA  
DE VALENCIA



**Máster Universitario**  
en Tecnologías, Sistemas y  
Redes de Comunicaciones

# **Synchronization Algorithms for 60 GHz Communication Standards**

*Autor:* Pablo Olivas González

*Director TU Braunschweig:* Tomas Kürner

*Tutor TU Braunschweig:* Marcos Liso Nicolás

*Tutor UPV:* Narcís Cardona Marcet

*Fecha de comienzo:* 15/03/2010

*Lugar de trabajo:* Institute for Communications Technology,  
Braunschweig Technical University (Germany)





**Technische  
Universität  
Braunschweig**

Diploma Thesis  
Synchronization Algorithms  
for 60 GHz Communication Standards

Pablo Olivas González  
August 2010

Prof. Dr.-Ing. Tomas Kürner  
Tutors: Dipl.-Ing. Marcos Liso Nicolás

Department of Mobile Radio Systems  
Institute for Communications Technology  
Braunschweig Technical University



## Abstract

The objective of every digital transmission system is to provide the receiver exact copies of the information generated by the source. Synchronization is then a critical function which has to be guaranteed to avoid problematic effects that lead to the degradation of the received signal and an incorrect detection of the originally transmitted symbols. Some of these harmful effects include incorrect sampling and demodulation of the received signal or inter-symbol and inter-carrier interference, among others.

In the EU project QStream, digital signal transmission in the 60 GHz frequency band has been addressed. As a part of this project, a demonstrator is being developed to prove that communications at such frequencies is possible. In 60 GHz systems, like in any other digital communications system, it is necessary to achieve both frequency and timing synchronization. For this reason, the implementation of synchronization recovery algorithms is essential.

This diploma thesis analyzes the difficulties that can appear when implementing a transmission system based on a 60 GHz standard. Specifically, this thesis will particularize its studies to the IEEE 802.15.3c-2009 standard, since it is very easy to adapt the results to other standards.

IEEE 802.15.3c is a revision to the IEEE 802.15.3-2003 standard for high rate Wireless Personal Area Networks (WPAN) which defines alternative physical and MAC layers operating in the millimeter wave, and can reach rates of up to 5 Gb/s.

The IEEE 802.15.3c standard specifies different physical layer transmission modes [1]:

- a) Single Carrier PHY (SC PHY).
- b) High Speed Interface PHY (HSI PHY).
- c) Audio/Visual PHY (AV PHY).

This document thesis is focused on the study of the synchronization issues for the SC PHY and HSI PHY modes of the IEEE 802.15.3c standard. For the single carrier based SC PHY mode, a number of algorithms that compensate the effect of the symbol timing offset have been tested. On the other hand for the OFDM based HSI PHY mode several algorithms have been implemented to achieve both timing and carrier synchronization recovery.

In summary, the aim of this diploma thesis is to carry out a literature study about the different algorithms that can be employed to achieve synchronization in both Single Carrier (SC) and Orthogonal Frequency Division Multiplexing (OFDM) systems, showing the different characteristics of each algorithm as well as the advantages and disadvantages they can offer. After that, several algorithms will be selected and implemented using the Simulink environment, and they will be fully tested and compared according to the configuration parameters of the IEEE 802.15.3c standard.

## Contents

<b>Abstract</b>	i
<b>Contents</b>	iii
<b>1. Basic Synchronization Fundamentals</b>	1
1.1 Synchronization in SC systems	2
1.2 Synchronization in OFDM systems	6
<b>2. Synchronization Algorithms</b>	13
2.1 Synchronization Algorithms for SC Systems	13
2.1.1 Feedforward Algorithm for Timing Recovery	13
2.1.2 Feedback Algorithm for Timing Recovery	15
2.2 Synchronization Algorithms for OFDM Systems	20
2.2.1 Data-Aided Synchronization Algorithms	21
2.2.2 Non-Data-Aided Synchronization Algorithms	29
<b>3. Simulations and Results</b>	31
3.1 Simulations and Results for the SC PHY Mode	31
3.2 Simulations and Results for the HSI PHY Mode	37
<b>4. Conclusions</b>	45
<b>5. Future work</b>	47
<b>List of Figures</b>	49

<b>List of Tables</b>	50
<b>List of Acronyms</b>	51
<b>Bibliography</b>	53
<b>Declaration</b>	57

## 1. Basic Synchronization Fundamentals.

This diploma thesis challenges the possible synchronization difficulties that can appear when implementing a transmission system based on the IEEE 802.15.3c-2009 standard, which defines two types of transmission modes: Single Carrier (SC) and Orthogonal Frequency Division Multiplexing (OFDM). In this chapter the basics of SC and OFDM are explained, along with the degradation caused by synchronization errors in both systems.

### 1.1 Synchronization in SC systems

Single Carrier modulation is the classic choice which has been employed for decades to implement all kinds of free-space transmission systems. The basic scheme of a single carrier based system is depicted on *Figure 1*:

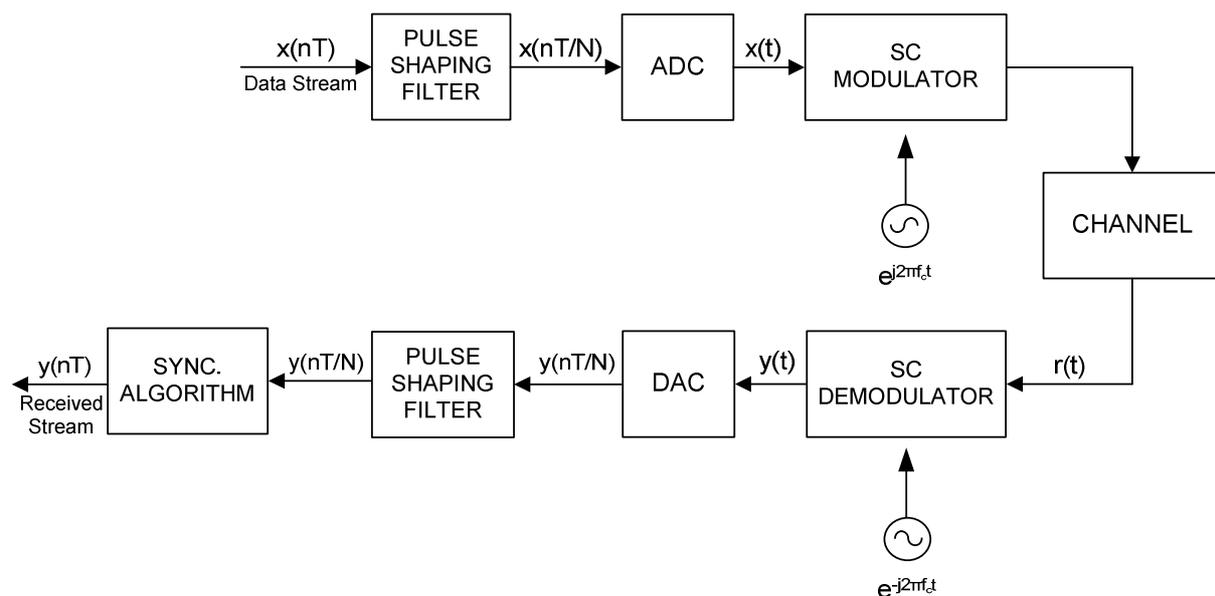


Figure 1: Basic Single Carrier system scheme.

The data source symbols containing the source information are generated and shaped to originate the baseband pulses. This pulse shaping filter upsamples the signal to provide  $N$  samples/symbol to the Analog to Digital Converter (ADC), which transforms the digital signal into an analog signal. Afterwards the analog signal modulates a carrier frequency and is transmitted through the channel. In the receiver side, the signal is demodulated and converted into the digital domain by the Digital to Analog Converter (DAC) to get the original baseband pulses. These pulses are filtered by a matched pulse shaping filter and the synchronization

algorithm finally selects the correct value of the symbol by interpolating the  $N$  received samples.

The low-pass representation of the received signal after the matched filter assuming perfect synchronization can be written as:

$$y(t) = \sum_i c_i h(t - iT) + n(t)$$

where  $\{c_i\}$  corresponds to the originally transmitted symbols,  $h(t)$  corresponds to the impulse response of the whole transmission system, and  $n(t)$  corresponds to a white Gaussian process with two-sided spectral density  $N_0/2$ .

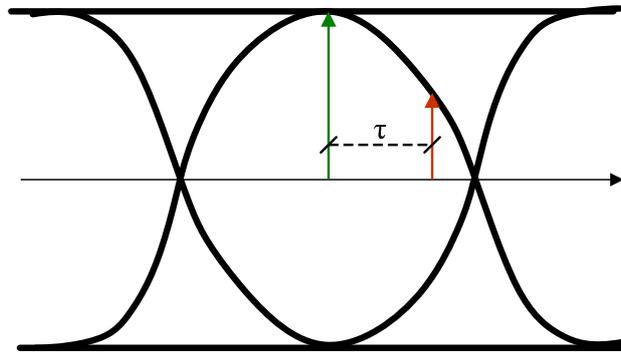
### 1.1.1 Synchronization errors in SC

In real SC systems synchronization is far from the ideal case. The transmission is usually affected by both timing and frequency deviation errors, which can be introduced by the time propagation of the signal from the transmitter to the receiver, by the effect of the channel or by imperfections at the transmitter or receiver components (sampling clock, local oscillator, etc). Taking these deviation errors into account, the received signal can be rewritten as:

$$y(t) = \sum_i c_i e^{j\epsilon} h(t - iT - \tau T) + n(t)$$

where  $\tau$  refers to the symbol timing offset and  $\epsilon$  corresponds to the frequency deviation. In this document, only the symbol timing synchronization will be analyzed, assuming always perfect carrier synchronization for the SC system ( $\epsilon = 0$ ).

The effect of the symbol timing offset (or jitter) can be easily understood by observing *Figure 2*:



*Figure 2: Optimum sample point in the eye diagram*

The figure shows the eye diagram formed by the shaped pulses of a received signal. As shown, the optimum sampling point would be the point with the highest energy. However, a symbol timing deviation ( $\tau$ ), makes the receiver to sample signals in any other point, which can lead the detector to decide that a different symbol was transmitted. The timing recovery algorithms provided in this diploma thesis pretend to estimate this offset ( $\hat{\tau}$ ) so that the correct value of the symbols can be recovered.

In the SC PHY mode, the frame format specified in the standard easily allow integer timing synchronization, i.e. to avoid timing offsets that exactly correspond to a multiple of the sample time. Integer timing synchronization is therefore considered as part of the frame synchronization and will not be studied in this thesis, which will focus on the fractional timing synchronization recovery.

### **1.1.2 SC PHY mode characteristics for IEEE 802.15.3c**

In this section, the main characteristics of the IEEE 802.15.3c standard that are related with symbol timing synchronization in single carrier mode will be described. There are many aspects of the physical layer that are fully specified in the standard like the symbol spreading or the Forward Error Correction (FEC) techniques, that won't be taken into account, since this document will focus on the performance of the synchronization recovery algorithms.

### 1.1.2.1 Constellation Mapping

There are five modulation schemes that are specified for the SC PHY mode. In this diploma thesis four of them will be tested:  $\pi/2$  BPSK,  $\pi/2$  QPSK,  $\pi/2$  8-PSK and  $\pi/2$  16-QAM. The symbols will be mapped according to these constellations:

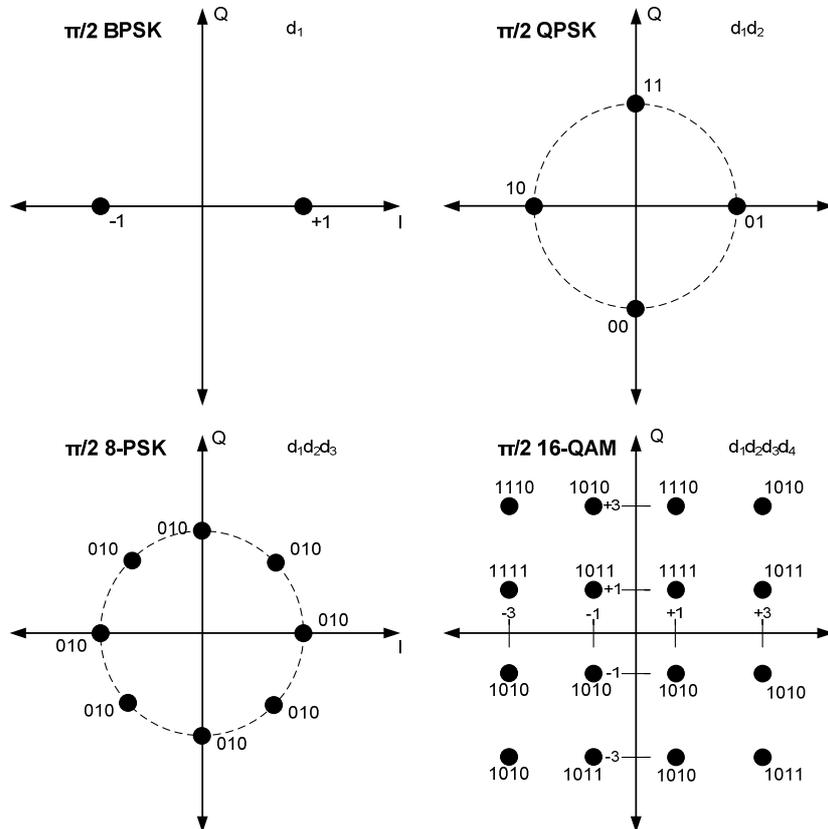


Figure 3: Constellations specified for the PHY SC mode by IEEE 802.15.3c standard

The  $\pi/2$  indicator implies a  $\pi/2$  phase shift counter clock-wise, according to:

$$z_n = j^n \times d_n \quad n = 1, 2, \dots, N$$

where  $j = \sqrt{-1}$ ,  $d_n$  corresponds to the original symbols, and  $z_n$  refers to the rotated symbols.

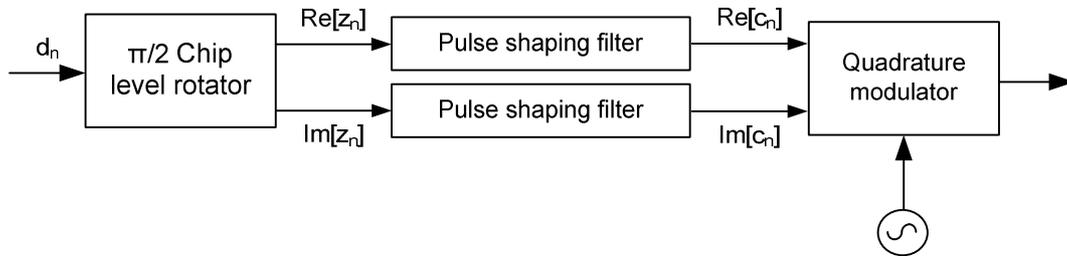
### 1.1.2.2 Pulse shaping filters

The standard doesn't specify the pulse shaping filter, so the classical raised-cosine-filter can be used. This filter is defined as [2]:

$$G(f) = \begin{cases} \sqrt{T} & |f| \leq \frac{1-\alpha}{2T} \\ \sqrt{T} \cos\left[\frac{\pi}{4\alpha}(|2fT| - 1 + \alpha)\right] & \frac{1-\alpha}{2T} \leq |f| \leq \frac{1+\alpha}{2T} \\ 0 & \text{otherwise} \end{cases}$$

where  $\alpha$  is the roll-off factor, i.e. the excess bandwidth of the filter.

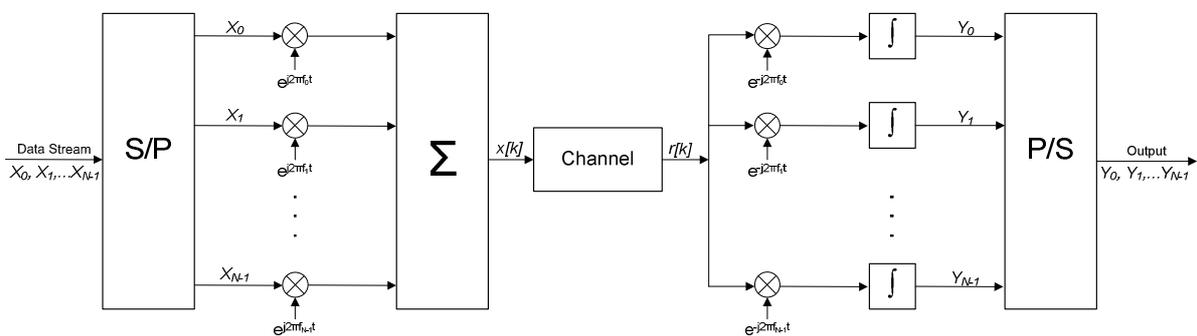
The scheme of the modulation including the  $\pi/2$  rotator and the pulse shaping filter is depicted in *Figure 4*:



*Figure 4: Scheme of the modulation including the  $\pi/2$  rotator*

## 1.2 Synchronization in OFDM systems

The HSI PHY mode of the IEEE 802.15.3c standard is based in the Orthogonal Frequency Division Multiplexing (OFDM) modulation. This multicarrier modulation is very common in wireless communication nowadays, due to its robustness against frequency selectivity in multipath channels. A great number of current wireless systems like DAB, DVB, WiFi, WiMAX or LTE make use of the OFDM. The basic scheme of an OFDM modulation system is depicted in *Figure 5* [4].



*Figure 5: Basic OFDM system scheme.*

The bandwidth is divided into several orthogonal subchannels, each one of them corresponds to a different subcarrier. This orthogonally between subchannels makes possible for the receiver to recover the originally sent information.

Every symbol of the data stream modulates a different subcarrier frequency, and is sent through the transmission channel together with the rest of the modulated data symbols. In the received, every symbol is demodulated by the same subcarrier.

In this way, the data information is spread all along time between the different subcarriers, allowing the system to reduce the number of consecutive data symbols lost due to frequency selective fading, giving the possibility of recovering the original information when using Forward Error Correction (FEC) techniques. In *Figure 6*, the distribution of the OFDM subcarriers along time can be seen.

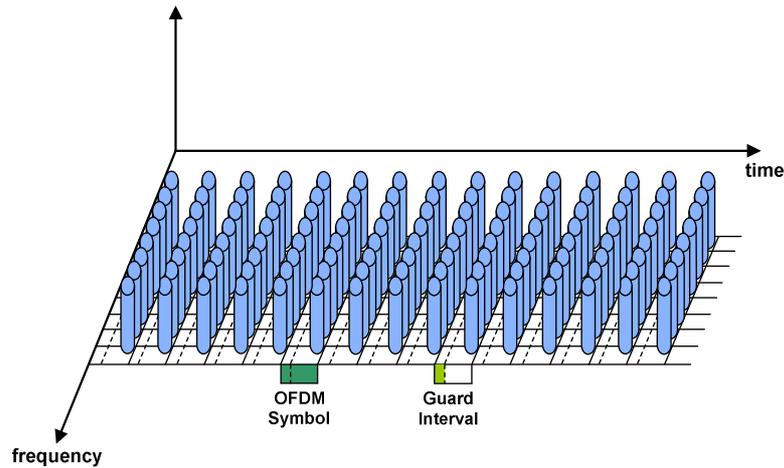


Figure 6: Distribution of the OFDM subcarriers along time

As seen, a Guard Interval (GI) is left between the data symbols along time to reduce the Inter-Symbol Interference (ISI). To completely avoid the ISI and highly reduce multipath interference, this GI is filled with a copy of the last part of the data symbol, which is known as the Cyclic Prefix (CP).

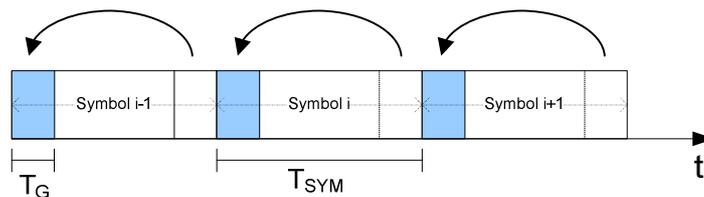


Figure 7: Cyclic Prefix extension of the OFDM symbol.

### 1.2.1 OFDM System Model

According to *Figure 5*, an OFDM symbol is expressed like [3]:

$$x(t) = \left[ \sum_{n=0}^{N-1} X_n e^{j2\pi f_n t} \right] \times \text{rect}\left(\frac{t}{T}\right)$$

where  $f_n$  is the subcarrier frequency (a multiple of the subcarrier spacing  $f_{sub}$ ),  $N$  is the total number of subcarriers, and  $X_n$  represents the original data symbols. Every OFDM symbol is transmitted during an interval  $[0, T]$ . To assure the orthogonality between subcarriers, the inter-subcarrier spacing must be proportionally inverse to the symbol duration ( $f_{sub} = 1/T$ ).

The discrete implementation of the OFDM system can be written as:

$$x[k] = \frac{1}{\sqrt{N}} x\left(k \frac{T}{N}\right) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} X_n e^{\frac{j2\pi kn}{N}} \quad k = 0, 1, 2, \dots, N-1$$

where  $k$  is the subcarrier index and also the time index. After adding the CP,  $x[k]$  correspond to  $\{x[N-G], \dots, x[0], \dots, x[N-1]\}$ , where  $G$  is the length of the guard interval.

It is clear from the previous formula that these same operations can be easily performed by using an Inverse Fast Fourier Transform (IFFT) algorithm in the digital domain, avoiding the use of several oscillators.

A practical scheme of the complete OFDM system model is finally showed in *Figure 8*:

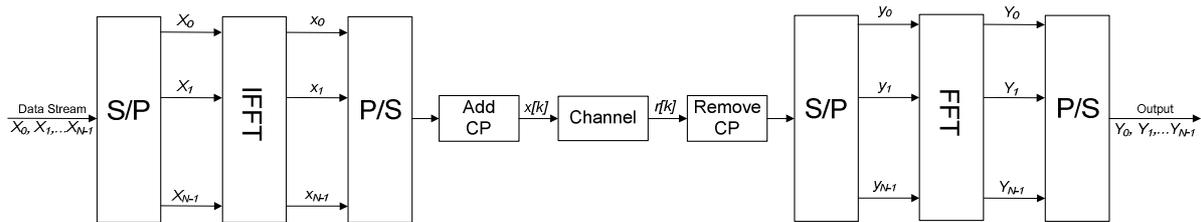


Figure 8: Practical OFDM system scheme.

As seen, the OFDM symbol is formed by performing the IFFT of the data stream and by adding the Cyclic Prefix. After the channel, the CP is removed and the FFT is performed to recover the data symbols.

### 1.2.2 Synchronization errors in OFDM

Synchronization in OFDM systems is much more sensitive to frequency and integer symbol timing errors than in SC systems due to the  $N$  times longer duration of an OFDM symbol and the intercarrier interference produced by the loss of orthogonality between the subcarrier frequencies [5].

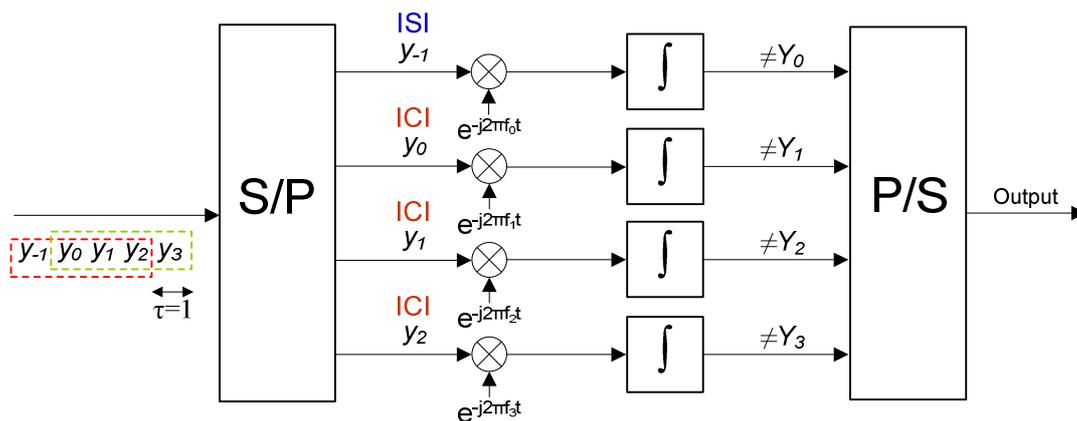
The received signal including the synchronization errors can be represented as:

$$r[k] = y[k - \tau] e^{\frac{j2\pi k\epsilon}{N}} + n[k] \quad k = 0, 1, 2, \dots, N-1$$

where  $\tau$  represents the integer symbol timing offset,  $\varepsilon$  represents the frequency deviation and  $n[k]$  corresponds to a white Gaussian process. Only the integer timing offset is considered because it is assumed that the fractional timing offset has been compensated after the matched filter in a similar way that it is done in the SC systems.

Not dealing with integer symbol timing errors can lead to inter-symbol interference (ISI), since the last data symbols of the current OFDM symbol will have an effect on the next OFDM symbol; and inter-carrier interference (ICI), because a delay on time translates into a inter carrier interference after the FFT.

To understand how timing offset generates ISI and ICI, see the example in *Figure 9*, where an OFDM system that employs only 4 subcarriers is depicted.



*Figure 9: Timing offset in OFDM produce both ISI and ICI*

The incoming stream should be  $y_0 y_1 y_2 y_3$ , but due to a timing offset of 1 symbol ( $\tau=1$ ), the incoming stream is  $y_{-1} y_0 y_1 y_2$ . That means that the first oscillator will demodulate a symbol from the previous OFDM symbol ( $y_{-1}$ ), which corresponds to ISI; and the rest of the oscillators will demodulate every symbol with a wrong subcarrier, which corresponds to ICI. Therefore, the output stream will not be  $Y_0 Y_1 Y_2 Y_3$  as it was expected.

Concerning the frequency synchronization errors, they lead to ICI, losing the orthogonality between the subcarrier frequencies. A time-variant phase deviation is introduced by the factor  $e^{\frac{j2\pi k\varepsilon}{N}}$  of the previous formula, which implies a rotation of the points of the constellation symbols at the receiver (see *Figure 10*).

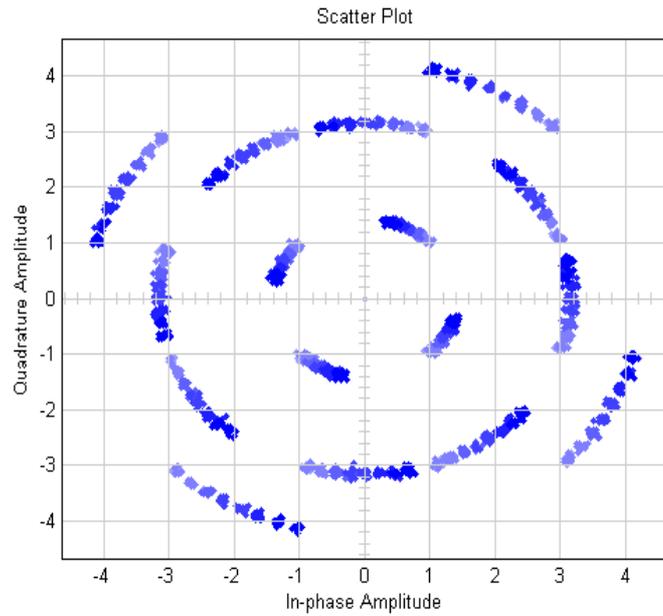


Figure 10: Rotation of the points of the constellation due to the frequency deviation

This time variation of the phase offset follows the next curve with a period  $N_{FFT}/\varepsilon$  that fluctuates between  $-\pi$  and  $\pi$ .

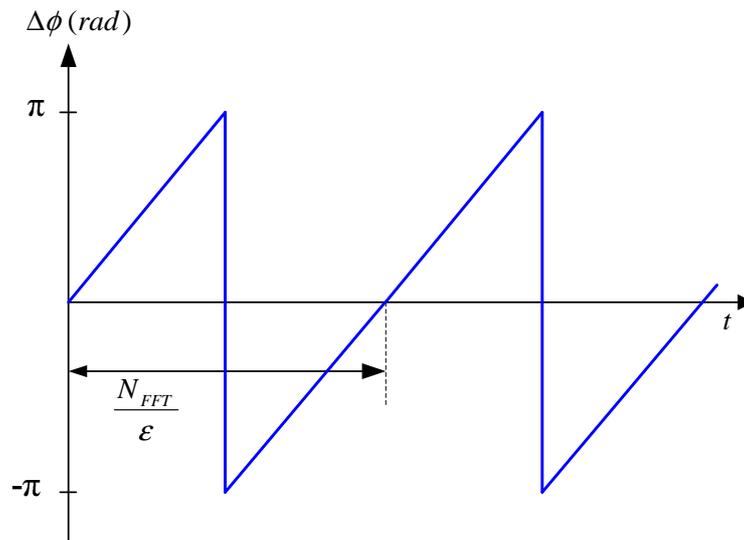


Figure 11: Time-variant phase error introduced in the OFDM symbol

### 1.2.3 SC HSI mode characteristics for IEEE 802.15.3c

In this chapter, the main characteristics of the IEEE 802.15.3c standard that are related with symbol timing and carrier synchronizations for the OFDM mode will be described. As it happened in the SC PHY mode, the standard specifies many aspects of the HSI PHY physical

layer that will not be considered in this thesis to focus on the performance of the synchronization recovery algorithms.

As described in the chapter 1.2.1, it is necessary to achieve not only fractional but also integer symbol timing synchronization before the OFDM demodulator to avoid ISI and ICI. The fractional symbol timing synchronization can be achieved in the same way as in the SC mode, so in this thesis we will only focus on the integer symbol timing synchronization and the carrier frequency synchronization.

### 1.2.3.1 Timing Related Parameters

The OFDM specified timing-related parameters which are in the interest of this study are listed in the next table:

Parameters	Description	Value	Formula
$f_s$	Reference sampling rate	2640 MHz	
$T_C$	Sample duration	$\sim 0.38$ ns	
$N_{sc}$	Number of subcarriers/FFT size	512	$1/f_s$
$N_{dsc}$	Number of data subcarriers	336	
$N_P$	Number of pilot subcarriers	16	
$N_G$	Number of guard subcarriers	141	
$N_{DC}$	Number of DC subcarriers	3	
$N_R$	Number of reserved subcarriers	16	
$N_U$	Number of used subcarriers	352	$N_{dsc} + N_P$
$N_{GI}$	Guard Interval length in samples	64	
	Subcarrier frequency spacing	5.15625 MHz	$f_s/N_{sc}$
$BW$	Nominal used bandwidth	1815 MHz	$N_U \times \Delta f_{sc}$
$T_{FFT}$	IFFT and FFT period	$\sim 193.94$ ns	$1/\Delta f_{sc}$
$T_{GI}$	Guard Interval duration	$\sim 24.24$ ns	$N_{GI} \times T_C$
$T_S$	OFDM Symbol duration	$\sim 218.18$ ns	$T_{FFT} + T_{GI}$
$F_S$	OFDM Symbol rate	$\sim 4.583$ MHz	$1/T_S$
$N_{CPS}$	Number of samples per OFDM symbol	576	$N_{sc} + N_{GI}$

Table 1: Timing Related Parameters

### 1.2.3.2 Subcarrier frequency allocation

The mapping of the different subcarriers within an OFDM symbol is summarized in the next table:

Subcarriers type	Number of subcarriers	Logical subcarriers indexes
Null subcarriers	141	[-256:-186] $\cup$ [186:255]
DC subcarriers	3	-1, 0, 1
Pilot subcarriers	16	[-166:22:-12] $\cup$ [12:22:166]
Guard subcarriers	16	[-185:-178] $\cup$ [178:185]
Data subcarriers	336	All others

Table 2: Subcarrier frequency allocation

### 1.2.3.3 Pilot Subcarriers

In every OFDM symbols 16 pilot subcarriers are included to allow for coherent detection, channel estimation and frequency synchronization. These pilot signals shall be placed into logical frequency subcarriers [-166:22:-12]  $\cup$  [12:22:166]. The value of the m-th pilot subcarrier of every OFDM symbol shall be defined as:

$$p_m = \begin{cases} (1+j)/\sqrt{2} & m=0,3,5,7,9,13,15 \\ (1-j)/\sqrt{2} & m=1,2,4,6,8,10,11,12,14 \end{cases}$$

### 1.2.3.4 Constellation Mapping

There are three modulation schemes that are specified for the HSI PHY mode. In this diploma thesis two of them will be tested: QPSK and 16-QAM. The symbols will be mapped according to these constellations:

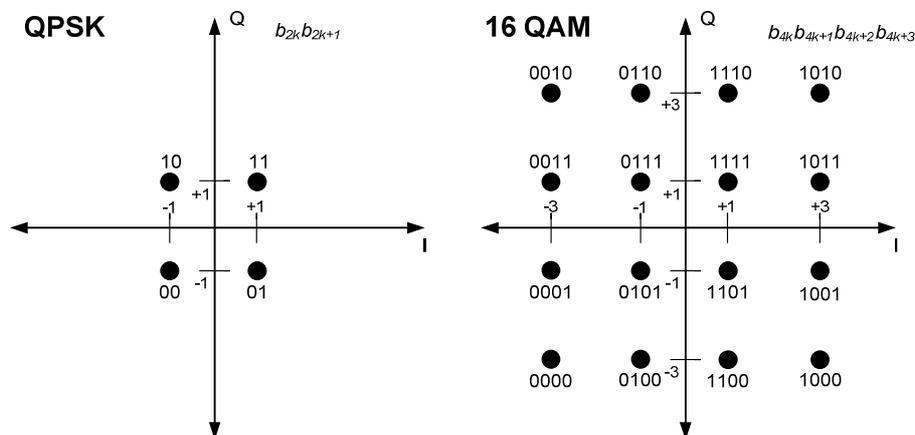


Figure 12: Constellations specified for the PHY HSI mode by IEEE 802.15.3c standard

None of them include a symbol rotation like it happened in the SC PHY mode.

## 2. Synchronization Algorithms

The literature comprises a vast list of algorithms to perform symbol timing recovery or carrier frequency recovery for either SC or OFDM modulations. In this chapter some of the synchronization techniques which present better performance will be explained, choosing several of these techniques to be implemented, simulated and analyzed in chapter 3.

As it was mentioned before, this diploma thesis focuses on fractional symbol timing recovery for SC systems and integer symbol timing recovery and carrier recovery for OFDM systems.

### 2.1 Synchronization Algorithms for SC Systems

All kinds of synchronization algorithms for symbol timing recovery are based on estimating the timing offset  $\tau$  (timing measurement) and applying this estimation to the sampling process (timing correction) [2].

Timing offset recovery algorithms can be classified in feedforward and feedback algorithms. Based on this classification, the following algorithms will be analyzed in this section:

- **Feedforward Algorithms for Timing Recovery:**
  - Square Timing Recovery
- **Feedback Algorithms for Timing Recovery:**
  - Gardner Timing Recovery
  - Modified Gardner Timing Recovery
  - Early-Gate Timing Recovery

#### 2.1.1 Feedforward Algorithm for Timing Recovery

The following scheme corresponds to a Feedforward algorithm for timing recovery:

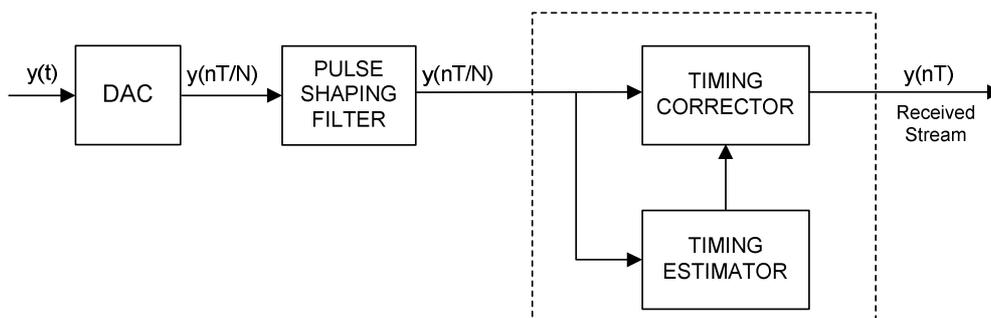


Figure 13: Feedforward Algorithm for Timing Recovery structure

As seen, from the  $N$  samples/symbol available after the pulse shaping filter, the timing estimator algorithm calculates the timing offset. Then, the timing corrector uses this estimate to obtain the correct value of the current symbol by interpolating between the  $N$  samples. This interpolation is illustrated in *Figure 14*.

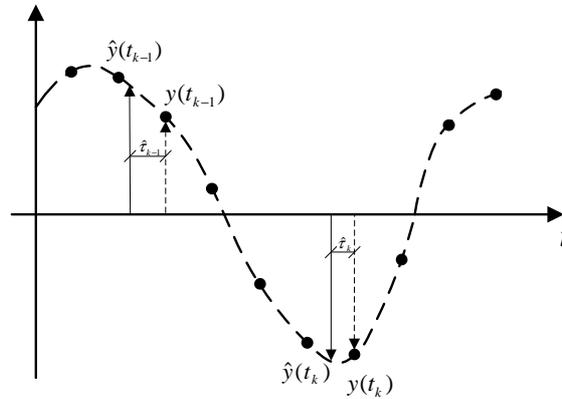


Figure 14: Interpolation of the  $N$  samples of the symbol to obtain the correct value

In the example, a 2PAM waveform signal upsampled at 4 samples / symbol is showed, where the dots represent the samples of the digital signal and the dashed line represents the analog equivalent signal. The timing corrector interpolates the received samples to get the correct value  $\hat{y}(t_k)$ . The interpolation instant depends on the estimated timing offset  $\hat{\tau}_k$  has been calculated by the timing estimator algorithm.

### 2.1.1.1 Square Timing Recovery

The Square Timing Recovery Algorithm [6,7] is one of the simplest symbol timing synchronization algorithms, but quite effective when the timing offset is stable enough. It divides the received signal by groups of  $L$  symbols where the delay can be considered constant. In every group of symbols, a single value of the timing offset  $\hat{\tau}$  is estimated.

The received signal  $y[k]$  has been filtered by the pulse shaping filter in the transmitter and the receiver, so there should be a main spectral component at  $1/T$  that is easily detectable when the deviation is constant. Since the received signal is sampled at a higher rate of  $N/T$ , it will be easier to detect the spectral component at  $1/T$ .

To perform this spectral detection, the algorithm calculates the FFT of the squared received signal at the symbol rate  $1/T$ . The angle of this coefficient gives then a very good estimate of

the symbol delay. The complete formula and scheme used by the algorithm to estimate the timing offset are:

$$\hat{\tau} = -\frac{1}{2\pi} \text{angle} \left( \sum_{i=kLN}^{(k+1)LN-1} |y[i]|^2 e^{-j2\pi i/N} \right)$$

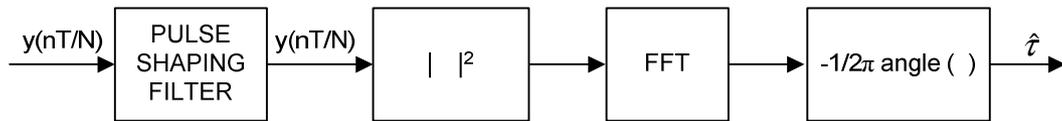


Figure 15: Square Timing Recovery Algorithm scheme

Its main drawback is that is not such an effective algorithm with time-variant offsets

### 2.1.2 Feedback Algorithm for Timing Recovery

The algorithm for timing recovery is depicted in the following figure:

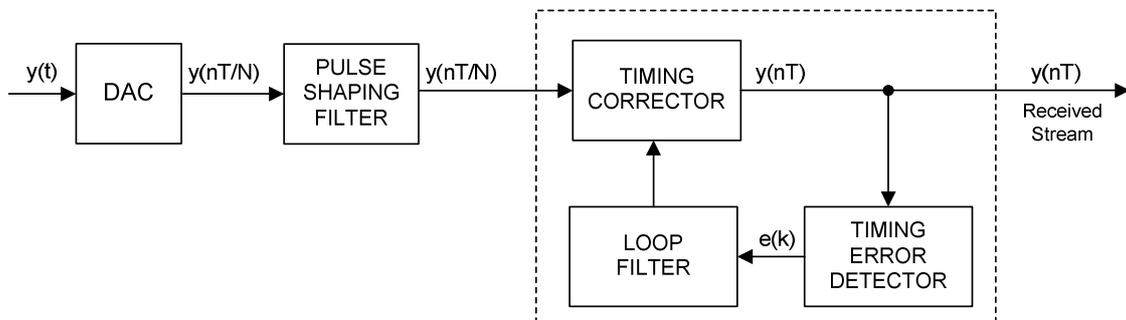


Figure 16: Feedback Algorithm for Timing Recovery structure

In this case, the timing offset estimation ( $e[k]$ ) of the current symbol, provided by the Timing Error Detector (TED), is smoothed by a loop filter and used by the timing corrector to obtain the correct value of the next symbol by interpolating the received signal.

The interpolator used depends on the implementer's choice, being the linear and the Farrow interpolators the most common ones. The loop filter can also be freely chosen but is usually specified by the designer of the algorithm.

In all the three algorithms presented in this section, the loop filter consists of a simple FIR filter of one tap. The overall timing offset for the next symbol is then updated from the estimation of the current timing offset like:

$$\hat{t}_{k+1} = \hat{t}_k + \gamma e[k]$$

and the interpolation instant (sampling instant) for the next symbol can be calculated as:

$$t_{k+1} = kT + \hat{t}_{k+1}$$

where  $T$  is the time period between symbols,  $k$  is the time index (symbol index) and  $\gamma$  is the step size (gain of the loop filter). The example of the *Figure 14* for Feedforward algorithms illustrates also Feedback algorithms. The only difference, as it has been just explained, is that  $\hat{t}_{k+1}$  is calculated instead of  $\hat{t}_k$  by the Feedback algorithms, and that a loop-filter is used.

The convergence of the algorithm to a stable timing offset is faster when  $\gamma$  is greater, but the remaining timing offset is higher. On the other hand, the convergence of the algorithm is slower and the timing offset is smaller when  $\gamma$  has a smaller value [8].

### 2.1.2.1 Gardner Timing Recovery

The Gardner Timing Recovery Algorithm [9] is probably the best known symbol timing synchronization algorithm. It has been extensively employed as a result of its simplicity and its independency of the carrier phase.

In every iteration, the Timing Error Detector calculates the new timing deviation estimation ( $e[k]$ ) employing the current ( $y[t_k]$ ) and the previous ( $y[t_{k-1}]$ ) corrected symbol values, and an intermediate interpolated value between both of them ( $y[t_{k-1/2}]$ ). The formula to obtain the estimation  $e[k]$  can be written as:

$$e[k] = \Re\{[y[t_{k-1}] - y[t_k]]y^*[t_{k-1/2}]\}$$

where  $t_{k-1/2} = (t_{k-1} + t_k)/2$ ,  $\Re\{a\}$  represents the real part of  $a$ , and  $a^*$  represents the complex conjugate of  $a$ . Since it is a fractional symbol timing offset estimator, the value of  $e[k]$  must

be always in the interval  $[0, T]$ , normalizing  $e[k]$  to fit into this interval by a ‘modulus after division’ operation when necessary.

The following picture showing a 2PAM waveform signal can be considered to explain how the algorithm works [10]:

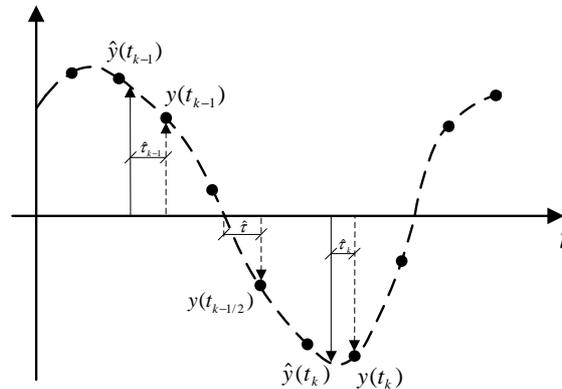


Figure 17: Operation principle of the Gardner algorithm

It is assumed that  $\hat{y}(t_{k-1})$  and  $\hat{y}(t_k)$  are the correct values of the symbols, but a  $\hat{\tau}$  seconds timing delay provides  $y(t_{k-1})$  and  $y(t_k)$  instead. In this situation, the slope of the intermediate symbol  $y(t_{k-1/2})$  will have a different value from zero that will be useful to calculate the timing offset.

Despite its many advantages, the Gardner algorithm has the inconvenient of having too much self-noise, i.e. too much timing offset present on the estimation generated by the algorithm itself.

For example, when using multilevel signals like M-QAM, the intermediate samples don't have to be zero even when there is no delay, which incurs in a higher timing offset. A similar problem can be found when working with narrowband signals, which forces to use a roll-off factor of the raised-cosine filters between 0.4 and 1.

### 2.1.2.2 Modified Gardner Timing Recovery

The high levels of self-noise of the Gardner recovery algorithm are a drawback that has been fully analyzed in the literature. Several modifications of the Gardner algorithm have been proposed (examples in [11, 12]) based on prefiltering or compensation techniques. Since one of the objectives of this diploma thesis is to perform good synchronization in M-QAM



### 2.1.2.3 Early-Gate Timing Recovery

The Early-Gate Timing Recovery algorithm [14] is very similar to the Gardner algorithm. Its philosophy is based on the same principle, but considering the value of the current symbol ( $y[t_k]$ ), and the half-way values between the last and the current symbols ( $y[t_{k-1/2}]$ ), and between the next and the current symbols ( $y[t_{k+1/2}]$ ).

The formula for the timing offset estimation calculation is:

$$e[k] = \Re\{[y[t_{k+1/2}] - y[t_{k-1/2}]]y^*[t_k]\}$$

Theoretically, its performance is better than the Gardner algorithm for M-QAM modulations, but in the rest of the cases, its performance is worse for high SNR values due to its greater levels of self-noise. For low SNR values it works better for higher roll-off factors of the raised-cosine filter.

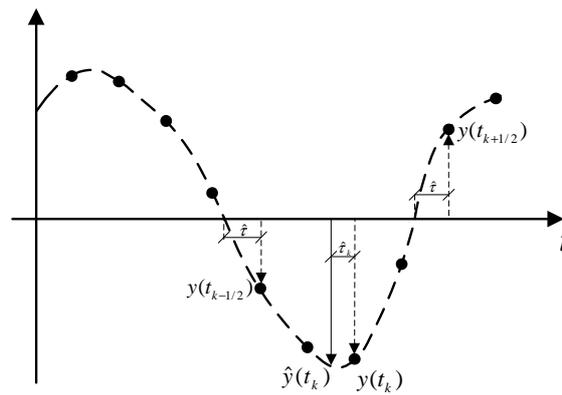


Figure 19: Operation principle of the Early-Gate algorithm

## 2.2 Synchronization Algorithms for OFDM Systems

The great popularity of the OFDM based systems has motivated a great number of synchronization algorithms for both symbol timing and frequency recovery. Different techniques have been proposed to deal with synchronization problems, and many of them can face timing and frequency synchronization together. In this chapter, a study of the literature has been performed. Examples of well-known algorithms of each type of technique are explained before some of them are directly implemented and analyzed in chapter 3.

It is possible to distinguish between Data-Aided and Non-Data-Aided algorithms, although there are many other ways to classify the OFDM synchronization algorithms. The algorithms explained in this section are summarized hereinafter. The name of the first of the authors who proposed the algorithms has been used to identify them:

- **Data-Aided Synchronization Algorithms**
  - Nogami Algorithm
    - Preamble-Aided and Pilot-Aided Algorithm.
  - Classen Algorithm
    - Pilot-Aided Algorithm.
  - Schmidh Algorithm
    - Preamble-Aided Algorithm.
  - Awoyesila Algorithm
    - Preamble-Aided Algorithm.
- **Non-Data-Aided Synchronization Algorithms**
  - Van De Beek Algorithm

### 2.2.1 Data-Aided Synchronization Algorithms

Data-Aided algorithms are identified for using known transmitted data to perform synchronization in the receiver. Most typical cases of these algorithms are Preamble-Aided and Pilot-Aided techniques. In the first case, a known transmitted data preamble is periodically employed, while in the second case, some pilots are sent together with the information data.

#### 2.2.1.1 Nogami Algorithm

The Nogami algorithm, proposed in [15] by Hiroshi Nogami and Toshio Nagashima, is one of the first successful preamble-aided synchronization algorithms. It makes use of the following transmitted OFDM frame structure to achieve carrier frequency deviation synchronization:

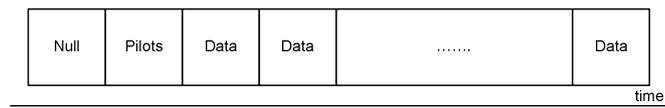


Figure 20: Nogami's algorithm frame structure

Every time slot corresponds to a complete OFDM symbol. As seen, a first symbol formed only by null symbols ( $X_n=0, \forall n$ ) is transmitted so that the drop in the received power can be used to detect the start of the beginning of the frame. The second slot, which combines null symbols with some pilot symbols distributed over the whole OFDM symbol, will be the preamble used to achieve synchronization at the receiver. After this symbol, only data will be transmitted, and the receiver will use for each of these symbols the synchronization information obtained from the second slot, until a new null and pilot OFDM symbols are transmitted. The structure of the Nogami's algorithm is depicted in Figure 21:

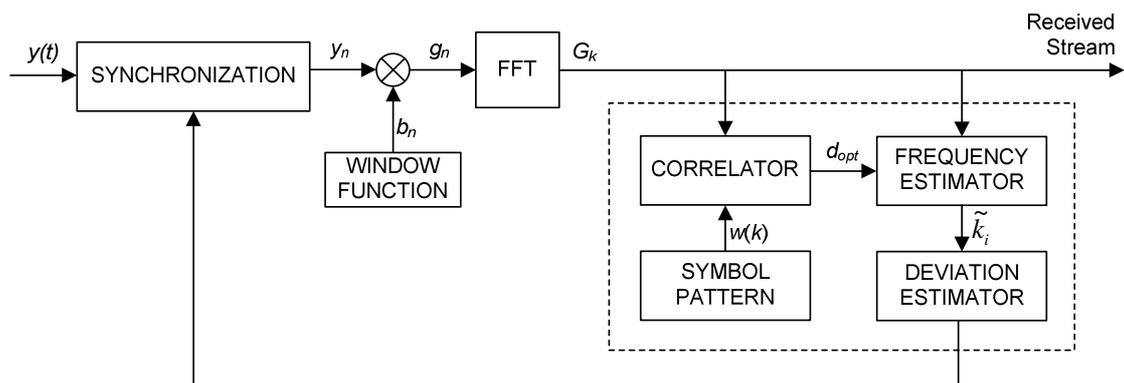


Figure 21: Nogami's algorithm scheme

After the CP removal, the received signal is multiplied by the Hanning window function defined by:

$$b_k = 1 - \cos\left(\frac{2\pi k}{N}\right) \quad k = 0, 1, 2, \dots, N-1$$

The result ( $g_n = b_n y_n$ ) is then applied to the FFT of the OFDM demodulator to obtain  $G_k$ . The properties of the Hanning function increase the precision of the estimation, highlighting clear local peaks in  $|G_k|$  in those subcarriers where the pilots are sent:

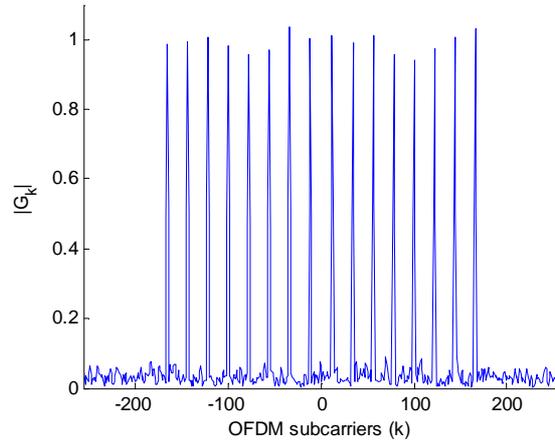


Figure 22: Peaks of  $|G_k|$  at the pilot frequencies

A first rough estimation of  $\varepsilon$  named  $d_{opt}$  is obtained from the frequency shift that achieves the maximum value of the correlation between the values of  $|G_k|$  and the transmitted frequency pattern:

$$d_{opt} = \arg \max_d \left( \sum_k w(k-d) |G_k| \right)$$

where  $d$  is a shift of the window pattern ( $w$ ), which is expressed by:

$$w = \begin{cases} 1 & X_k \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

The estimation  $k'_i$  of every subcarrier frequency is obtained as:

$$\tilde{k}_i = k_i + d_{opt} + \frac{1 - 2\gamma_i}{1 + \gamma_i}, \quad \text{where } \gamma_i = \frac{|G_{k_i + d_{opt} - 1}|}{|G_{k_i + d_{opt}}|}$$

or

$$\tilde{k}_i = k_i + d_{opt} - \frac{1 - 2\beta_i}{1 + \beta_i}, \quad \text{where } \beta_i = \frac{|G_{k_i + d_{opt} + 1}|}{|G_{k_i + d_{opt}}|}$$

From these estimations, a value of the frequency deviations for every subcarrier frequency is derived as:

$$\varepsilon_i = \frac{k_{i+1}\tilde{k}_i - k_i\tilde{k}_{i+1}}{\tilde{k}_{i+1} - \tilde{k}_i}$$

Finally, the average of these values provides a global estimation of the frequency synchronization deviations.

$$\hat{\varepsilon} = E[\varepsilon_i]$$

### 2.2.1.2 Schmidl Algorithm

The Schmidl algorithm, proposed in [16] by Timothy M. Schmidl and Donald C. Cox, is a preamble-aided synchronization algorithm which employs a preamble structure that was quite innovative when it was first released in 1997. It has been the base for many current algorithms, since it can be used for either continual or discontinuous transmission. It uses the following frame structure:

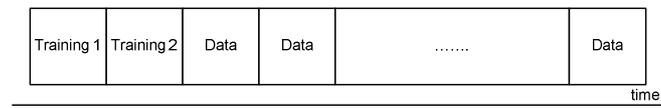


Figure 23: Schmidl's algorithm frame structure

Two different training OFDM symbols are used. The first training symbol is formed by mapping a pseudo-noise (PN) sequence at the even frequencies of the IFFT, inserting null symbol at the odd frequencies. Such distribution provides an OFDM symbol which has two identical halves in time, which will be used to achieve fractional carrier frequency and integer symbol timing synchronization. The PN sequence may be chosen from the external points of the constellation to increase the energy.

The second training symbol is formed by a PN sequence on the even frequencies to help determine the integer frequency deviation and a different PN sequence on the odd frequencies to measure these subchannels if needed.

The structure of the Schmidl's algorithm is depicted in Figure 24:

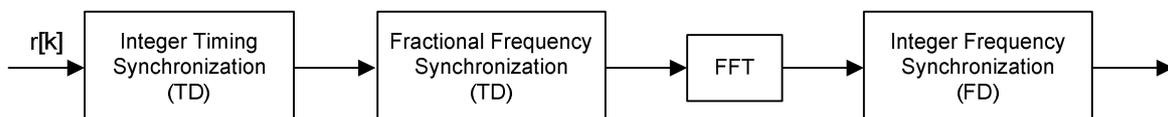


Figure 24: Schmidl's algorithm scheme

Both timing synchronization and fractional frequency deviation synchronization are achieved by employing the first training symbol in the time domain (before the FFT), while the integer deviation synchronization is achieved in the frequency domain (after the FFT).

The autocorrelation of the first training symbol with two identical halves is used to achieve time synchronization. The autocorrelation is computed in a window that slides along time as:

$$P(d) = \sum_{k=0}^{N/2-1} r^*(d+k)r(d+k+N/2)$$

where  $d$  is the shift index. On the other hand, the energy of the second half of the symbol is:

$$R(d) = \sum_{k=0}^{N/2-1} |r(d+k+N/2)|^2$$

Schmidl and Cox define a timing metric as:

$$M(d) = \frac{|P(d)|^2}{(R(d))^2}$$

An example of this timing metric is shown in the following:

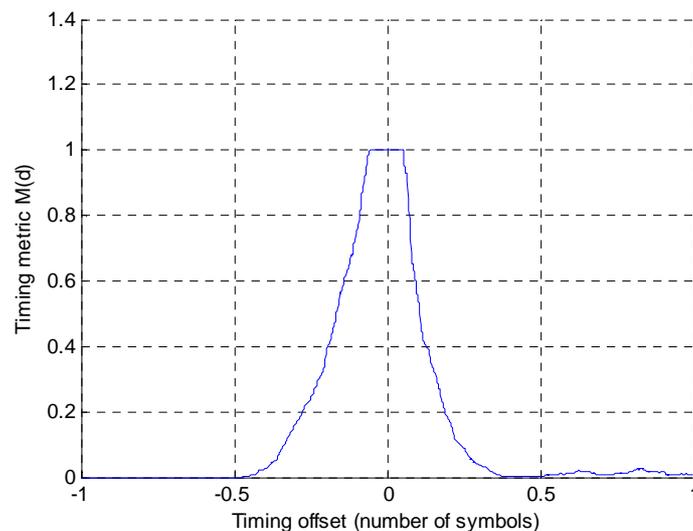


Figure 25: Timing metric  $M(d)$

As shown, there is a flat area of the curve of the length of the CP. The maximum of this flat area is considered as the timing offset estimate:

$$\hat{\tau} = \arg \max_d \{M(d)\}$$

On the other hand, due to the frequency deviation, the two identical halves of the first training symbol will have a phase difference between them of

$$\phi = \pi T \varepsilon$$

which can be estimated from the autocorrelation value  $P(\tau)$ :

$$\hat{\phi} = \text{angle}(P(d))$$

The fractional frequency deviation is then computed as:

$$\hat{\varepsilon}_f = \frac{\hat{\phi}}{\pi T}$$

The second training symbol can be used to achieve the integer frequency deviation, correcting first the fractional frequency deviation to avoid inter carrier interference (ICI). Afterwards, the number of even positions of the second training symbol can be calculated:

$$B(g) = \frac{\left| \sum_{i \in W} s_1^*(i + 2g) v^*(i) s_2(i + 2g) \right|^2}{2 \left( \sum_{i \in W} |s_2(i)|^2 \right)^2}$$

where  $g$  covers the range of possible frequency deviations,  $s_1$  and  $s_2$  are the FFT of the two corrected training symbols and  $v_k$  is the PN sequence mapped on the even frequencies of the second transmit symbol.

Finally, the integer frequency deviation is calculated as:

$$\hat{\varepsilon}_i = \frac{2}{T} \arg \max_g \{B(g)\}$$

### 2.2.1.3 Awoyesila Algorithm

The Awoyesila algorithm, proposed in [20] by Adegbenga B. Awoyesila, Christos Kasparis and Barry G. Evans, is one of the most efficient preamble-aided synchronization algorithm that are based in the Schmidl algorithm. It combines low complexity with fast convergence, and only employs one symbol training:

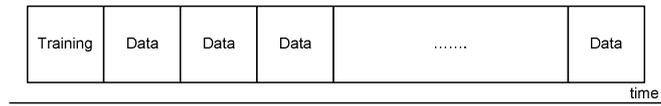


Figure 26: Awoyesila's algorithm frame structure

This training symbol has the same structure as the first of the two symbol training used by Schmidl and Cox: PN sequence mapped in the even subcarriers and null symbols mapped in the odd subcarriers.

Although the algorithm scheme includes new blocks of path timing to face multipath channels, they will not be explained in this section, since the only channel which will be tested in this diploma thesis is the AWGN channel. In that case, the implemented part of the Awoyesila's algorithm employs the same structure as the Schmidl's algorithm (see *Figure 24*).

The correlation of the received training symbol is performed in the same way as in the Schmidl's algorithm:

$$P(d) = \sum_{k=0}^{N/2-1} r^*(d+k)r(d+k+N/2)$$

but the defined timing metric differs from the other algorithm:

$$M_c(d) = \frac{1}{G+1} \sum_{k=0}^G |P(d-k)|^2$$

where G is the length of the guard interval (GI).

An example of this timing metric is shown in *Figure 27*:

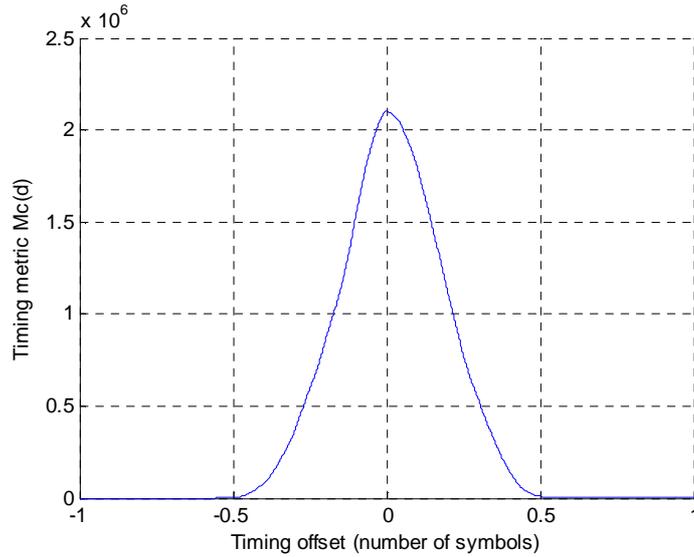


Figure 27: Timing metric  $M_c(d)$

A more clear maximum than in the Schmidl's algorithm can be distinguished. Therefore, the integer timing offset and the fractional frequency deviation are obtained as:

$$\hat{\tau} = \arg \max_d \{M_c(d)\}$$

$$\hat{\epsilon}_f = \frac{1}{\pi} \text{angle}(P(\hat{\tau}))$$

and the received signal can be partially corrected as:

$$r_{cor}[k] = r[k]e^{-j2\pi\hat{\epsilon}_f / N}$$

The fractional frequency deviation has been compensated, and  $r_{cor}[k]$  can only include the effect of an integer frequency deviation.

For the AWNG channel case, where there is a single arriving path, the Awoyesila's algorithm can be simplified, and the integer frequency deviation can be calculated as:

$$U(\tau, k) = r_{cor}[d+k]S^*[k]; \quad k = 0, \dots, N-1$$

$$I(\tau, i) = |FFT\{U(\tau, k)\}|; \quad k = 0, \dots, N-1$$

$$\hat{\epsilon}_i = \arg \max_i \{I(\tau, i)\}$$

where  $S[k]$  is the training symbol with the two identical halves in the time domain (after IFFT).

### 2.2.1.4 Classen Algorithm

The Classen algorithm, proposed in [22] by Ferdinand Classen and Heinrich Meyr, is a known reference as pilot-aided frequency synchronization algorithm. Instead of sending dedicated training symbols, it sends pilot-symbol on a few known subcarriers of every OFDM symbol.

It differentiates between two operation modes: the acquisition mode, where the large frequency deviations are challenged; and the tracking mode, which constantly compensates the small frequency deviations. Therefore, the first stage performs coarse synchronization, leaving the fine synchronization to the second stage. The structure of the Classen's algorithm is depicted in *Figure 28*:

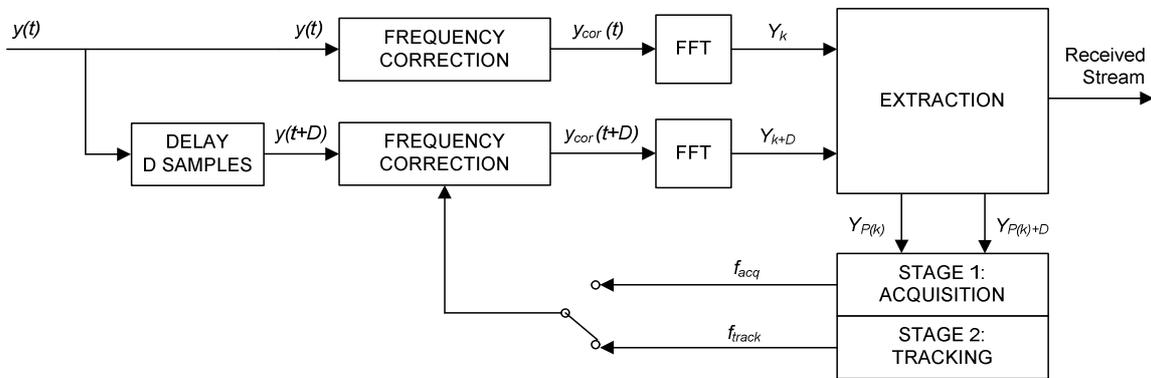


Figure 28: Classen's algorithm scheme

Every new received OFDM symbol, together with a  $D$  seconds delayed version of itself, is corrected in the receiver using the previous estimation of the frequency deviation. The FFT is applied to both OFDM symbols (the original and the delayed one), and the values which correspond to every pilot subcarrier ( $Y_{P(k)}$ ) are extracted and employed to get a new frequency deviation estimation.

In the acquisition stage, an initial coarse frequency deviation estimate  $f_{acq}$  is obtained:

$$f_{acq} = \frac{1}{2\pi} \arg \max_{f_{trial}} \left\{ \sum_{k=0}^{P-1} \left[ (Y_{P(k)+D} Y_{P(k)}^*) (c_1^* c_0) \right] \right\}$$

where  $Y_{P(k)}$  and  $Y_{P(k)+D}$  have been frequency-corrected with different  $f_{trial}$  chosen by the implementer of the algorithm,  $P(k)$  is a vector containing the index of the pilot subcarriers,  $c_0$  and  $c_1$  are the values of the pilot symbols and  $D$  is the number of symbols that can be placed between two pilot symbols.

In the tracking stage, it is assumed that the remaining deviation is inferior to the half of the subcarrier spacing, so it can consider a similar situation to a single carrier case, and apply a similar recovery algorithm. The fine frequency deviation estimate  $f_{\text{track}}$  is then achieved like:

$$f_{\text{track}} = \frac{1}{2\pi D} \text{angle} \left\{ \sum_{k=1}^{P-1} \left[ (Y_{P(k)+D} Y_{P(k)}^*) (c_1^* c_0) \right] \right\}$$

where  $Y_{P(k)}$  and  $Y_{P(k)+D}$  have been frequency-corrected with the  $f_{\text{acq}}$  obtained in the first stage.

### 2.2.2 Non-Data-Aided Synchronization Algorithms

Non-Data-Aided algorithms avoid the insertion of special known information in the transmitter to achieve synchronization in the receiver. In this section, an efficient example of this kind of algorithms is explained.

#### 2.2.2.1 Van De Beek Algorithm

The Van De Beek algorithm, proposed in [23] by Jan-Jaap van de Beek, Magnus Sandell, and Per Ola Börjesson, is a joint maximum likelihood (ML) estimator of the integer symbol timing and fractional frequency deviation synchronization. As a Non-Data-Aided algorithm, it doesn't need any transmitted extra data to achieve synchronization, assuming that the data symbols already contain enough information to do it. The main inconvenience of this algorithm, is that the frequency deviation must be in the range  $|\varepsilon| < 1/2$ , so it cannot perform integer frequency deviation synchronization.

The algorithm considers an observation interval of  $2N+G$  samples which include a complete OFDM symbol and two half OFDM symbol which correspond to the previous and the next symbols.  $N$  refers to the number of subcarriers of the OFDM symbol, and  $G$  refers to the length of the Guard Interval.

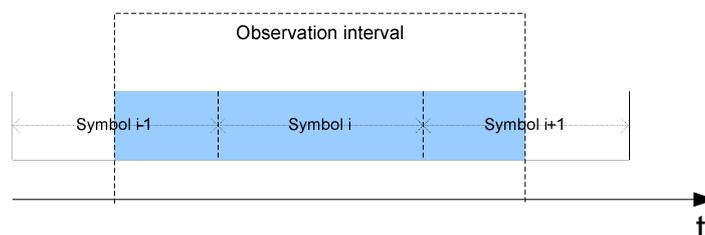


Figure 29: Observation interval of  $2N+G$  samples

The logarithm of the probability density function  $f(r | \tau, \varepsilon)$  of the observed samples is known as the log-likelihood function.

$$\Lambda(\tau, \varepsilon) = \log f(r | \tau, \varepsilon)$$

The maximization of the log-likelihood function, provide the desired values.

$$\max_{\tau, \varepsilon} \Lambda(\tau, \varepsilon)$$

Assuming that the channel is Gaussian, and after several approximations, the timing and frequency deviations can be estimated by:

$$\hat{\tau} = \arg \max_{\tau} \{|\gamma(\tau)| - \rho \Phi(\tau)\}$$

$$\hat{\varepsilon} = -\frac{1}{2\pi} \text{angle}\{\gamma(\hat{\tau})\} + n$$

where  $n$  is the integer frequency deviation. We need to assume then that  $n=0$  (no integer frequency deviation error) to achieve the synchronization. For this reason, the frequency deviation cannot be greater than half of the subcarrier spacing ( $|\varepsilon| < 1/2$ ), as it was said before. The factor  $\gamma(m)$  is an autocorrelation of the received signal, which will get its maximum near the beginning of the OFDM symbol thanks to the repetition that the CP insertion involves. On the other hand  $\Phi(m)$  is an energy term, independent of the frequency deviation.

$$\gamma(m) = \sum_{k=m}^{m+L-1} r[k] r^*[k+N]$$

$$\Phi(m) = \frac{1}{2} \sum_{k=m}^{m+L-1} |r[k]|^2 + |r[k+N]|^2$$

The weighting factor  $\rho$  depends on the received SNR:

$$\rho = \left| \frac{E\{r[k]r^*[k+N]\}}{\sqrt{E\{|r[k]|^2\}E\{|r[k+N]|^2\}}} \right| = \frac{\sigma_s^2}{\sigma_s^2 + \sigma_n^2} = \frac{SNR}{SNR + 1}$$

### 3. Simulations and Results

After the literature study detailed in the previous chapter, some of the explained synchronization algorithms were implemented and tested for an AWGN channel following the configuration parameters specified in the IEEE 802.15.3c standard. All the simulations were performed in a Simulink environment, and the results obtained are showed in this chapter.

#### 3.1 Simulations and Results for the SC PHY Mode

##### 3.1.2 Simulations

The following figure shows a capture of the schematic used to implement the SC based system and every synchronization recovery algorithm.

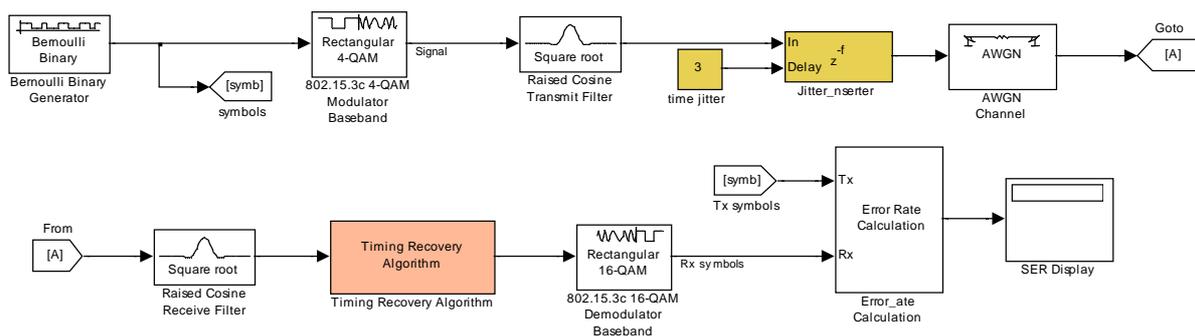


Figure 30: Implementation of a SC system with a generic timing recovery algorithm

The rotated constellations defined by the standard were used (see 1.1.2.1.). The pulse-raised cosine filters were configured with a roll-off factor of  $\alpha=0.75$ , a group delay of 3 symbols and an up-sampling factor of  $N=8$ .

Two types of timing offset were used: a constant timing offset on time and a timing offset that slightly varies over time following a random distribución between the timing offset values of  $3/16$  and  $9/16$ .

The main metric used to evaluate the performance of every algorithm was the Symbol Error Rate (SER) instead of the typical Bit Error Rate (BER) since it is more interesting to analyze

the amount of complete symbols that were correctly detected as a result of using every synchronization algorithm.

On the other hand, to analyze the errors on the deviation estimation, the Mean Square Error (MSE) of these errors were used. The Timing Estimation Error was defined as the estimation error in the timing offset estimation:

$$\tau_e = \tau - \hat{\tau}$$

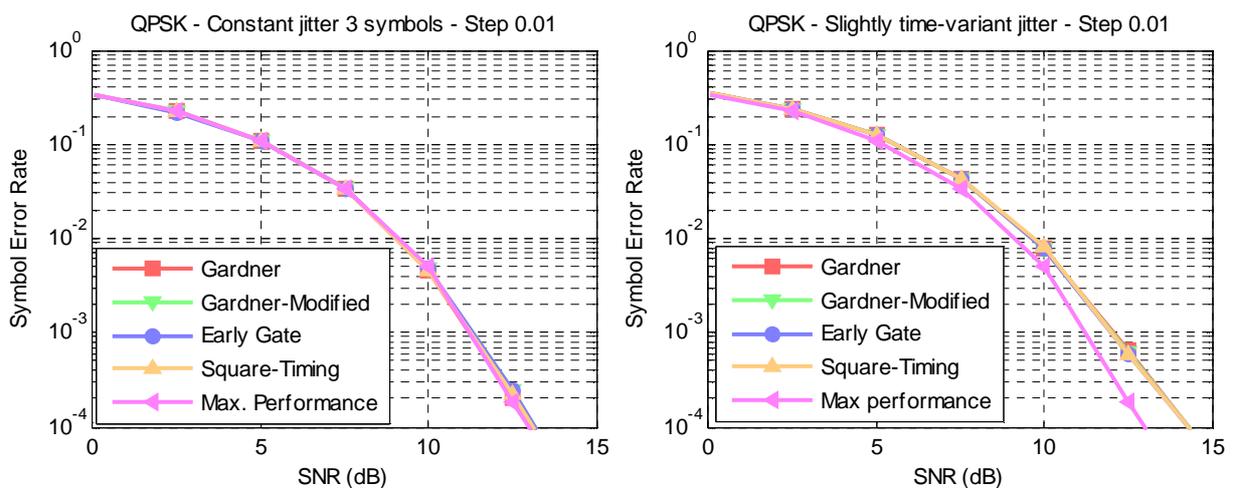
where  $\tau$  the real value of the timing offset,  $\hat{\tau}$  is the value of the timing offset estimated by the algorithm and  $\tau_e$  is the Timing Estimation Error. The MSE Timing Estimation Error was then obtained as:

$$MSE = (\tau - \hat{\tau})^2 = \tau_e^2$$

### 3.1.2 Results

#### 3.1.2.1 Influence of the SNR and the modulation

The performance comparison of the proposed timing recovery algorithms for different values of the SNR, using a QPSK modulation and a step value of the loop filter of 0.01, is showed in *Figure 31*:



*Figure 31: Comparison of the different timing recovery algorithms for different SNR*

As seen, all the performance of all the proposed algorithms is almost the same. In the case of the constant timing offset, the algorithms exactly match the curves that provides the maximum performance, i.e. the curve obtained without considering any synchronization aspect. When using a slightly time-variant offset, the algorithms are not so close to the maximum performance curve for high SNR, but they are still very close nevertheless.

These curves indicate that all the algorithms provide the same performance with the employed timing offsets, reaching almost a perfect synchronization. It seems clear that all the algorithms can handle a constant and a slightly time-variant offset distribution. In fact, comparing the MSE Timing Estimation Error, it can be seen the estimated deviation of the square-timing algorithm is the most accurate, which proves that the slightly time-variant offset employed is stable enough from the square-timing algorithm's point of view.

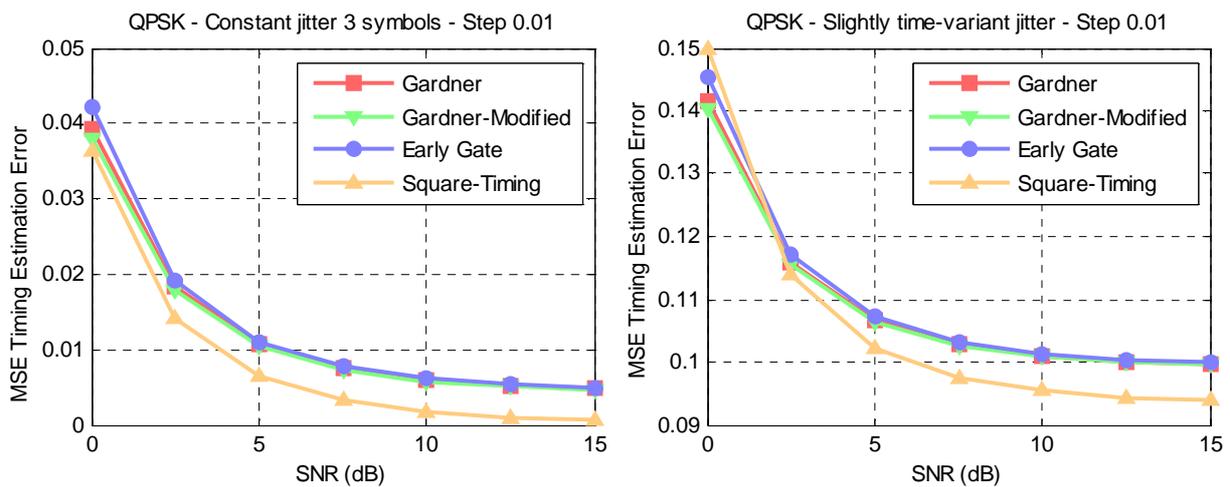


Figure 32: Comparison of the MSE timing estimation error of the different timing recovery algorithms

Nevertheless, provided that all the algorithms perform the same way for the used timing offset the comparison of the performance between the algorithms and the different modulations of the standard when introducing the slightly time-variant offset is depicted in *Figure 33*. Only the Gardner's algorithm curves together with the maximum performance curves are showed to avoid confusion.

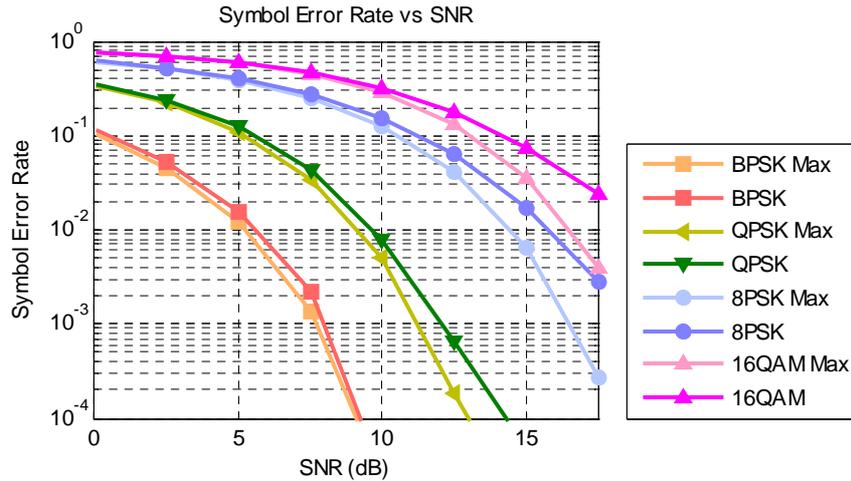


Figure 33: Comparison of the different timing recovery algorithms for different modulations

It can be noted that when the modulator has a greater number of levels, the curves that represent the algorithms are farther from the maximum performance curves, which show the degradation of the performance of the algorithms for multilevel signals.

### 3.1.2.1 Influence of the constant timing offset and the step gain

In this section, the SNR needed to achieve a SER of  $5 \cdot 10^{-3}$  is fixed to study the performance of the algorithms when changing the value of the constant timing offset and the step gain of the loop filter. For this SER, the Probability Density Function (PDF) of the Timing Estimation Error of the Gardner's algorithm is:

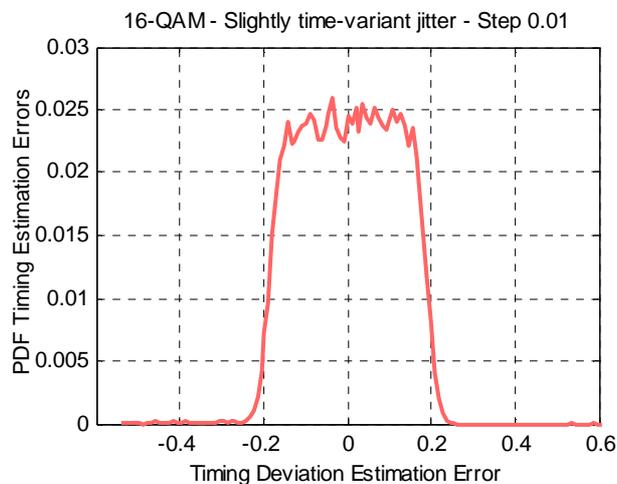
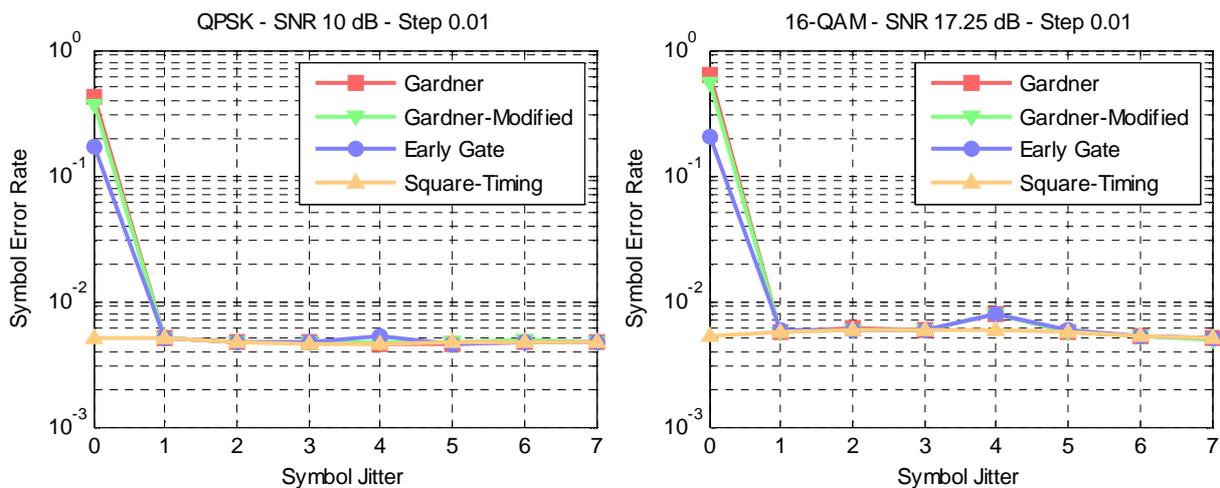


Figure 34: Probability Density Function of the timing offset estimation errors

As can be seen, most part of the timing offset erroneous estimations provide values below the 20 % of the period of the symbol. For other algorithms and other modulations, the range is very similar.

A SER value of  $5 \cdot 10^{-3}$  corresponds to a SNR of 10 dB for QPSK and 17.25 dB for 16-QAM (see *Figure 33*). Once these values are fixed, the performance of the algorithms when changing the constant timing offset value is depicted in the following picture.



*Figure 35: Performance of the timing algorithms when changing the constant timing offset value*

As seen, all the algorithms have a very similar behaviour for any constant timing offset, except for when there is no timing offset introduced. In this case, the Square-timing algorithm has still the same performance, but the feedback algorithms show very poor results because of the great self-noise that have this kind of algorithms.

Finally, the influence of the step gain of the loop filter of these feedback algorithms is showed in *Figure 36*. The best results for QPSK and 16-QAM are achieved when using a loop gain step value of 0.01. The worst value is achieved when using a step of 0, which corresponds to not using the timing offset estimation. Such a value of the step gain reduces the self-noise of the algorithms, but increases the convergence time of the same.

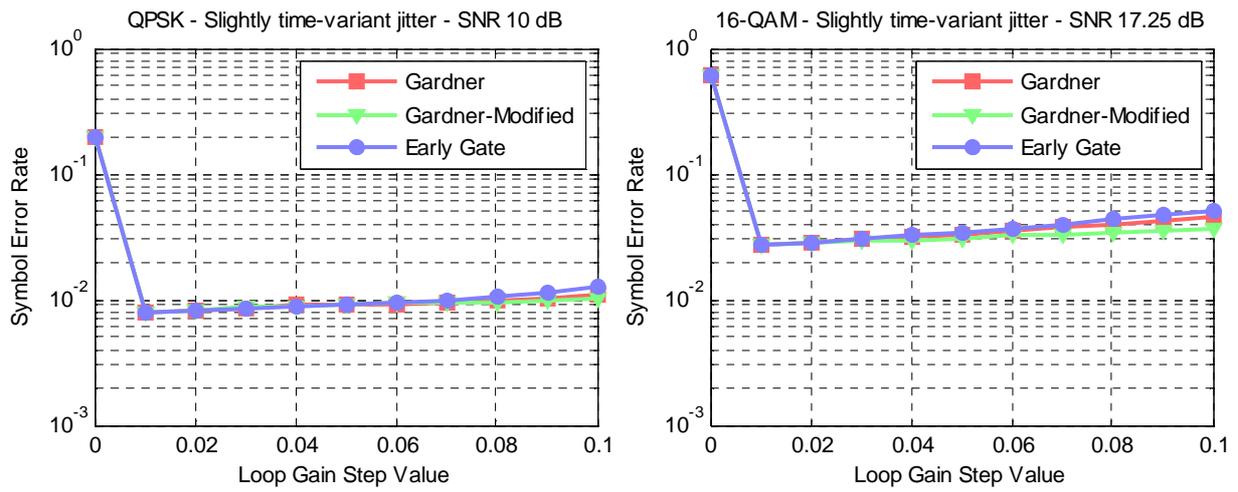


Figure 36: Comparison of the performance of the timing algorithms when changing the loop gain step value

### 3.2 Simulations and Results for the HSI PHY Mode

In this section the results obtained when evaluating the Awoyesila and Van de Beek synchronization algorithms for OFDM will be analyzed.

#### 3.2.1 Simulations

The following figure shows a capture of the schematic used to implement the OFDM based system and every synchronization recovery algorithm.

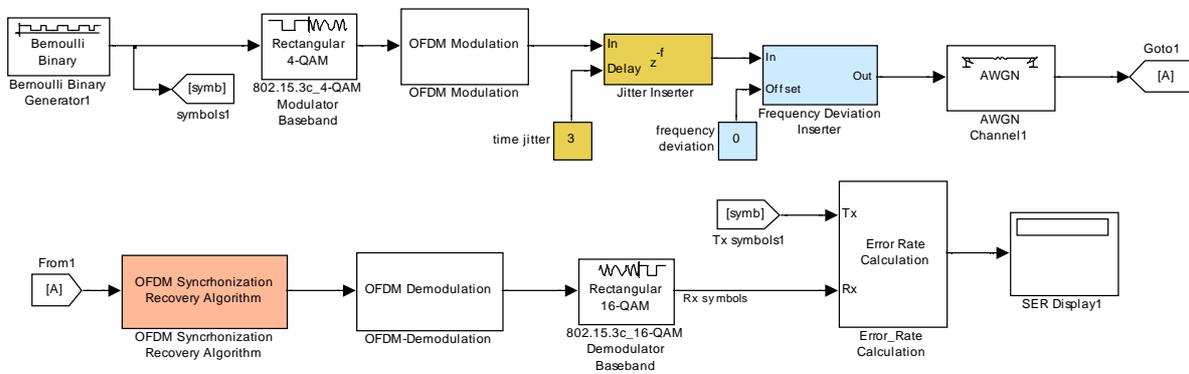


Figure 37: Implementation of a OFDM system with a generic timing recovery algorithm

The main changes introduced consist of including the OFDM modulator and demodulator, and the frequency deviation inserter. Only constant symbol timing offset will be introduced, and the same metrics to evaluate performance will be analyzed.

Both the SER and the MSE Timing Estimation Error metrics will be used, together with a new metric: the MSE Frequency Estimation Error. It has been defined as the estimation error in the frequency deviation estimation:

$$\varepsilon_e = \varepsilon - \hat{\varepsilon}$$

where  $\varepsilon$  the real value of the frequency deviation,  $\hat{\varepsilon}$  is the value of the frequency deviation estimated by the algorithm and  $\varepsilon_e$  is the Frequency Estimation Error. The MSE Frequency Estimation Error is then obtained as:

$$MSE = (\varepsilon - \hat{\varepsilon})^2 = \varepsilon_e^2$$

### 3.2.2 Results

#### 3.2.2.1 Characterization of the algorithms without deviations introduced

When using a synchronization recovery algorithm, it is important to notice that there is always a penalization introduced by the same algorithm which is called self-noise of the algorithm. That means that if no timing offset and no frequency deviation are introduced at the transmitter, there will be some synchronization errors at the receiver. The reason is that the estimation of the deviation, which is zero in this case, is not perfect. Obviously, in real systems there are always synchronization errors, so the gain of using these algorithms will justify its use in these situations.

However, it is necessary to characterize the performance of the algorithm under these conditions, so these are the curves obtained with the Awoyesila and the Van de Beek algorithm when no errors are introduced:

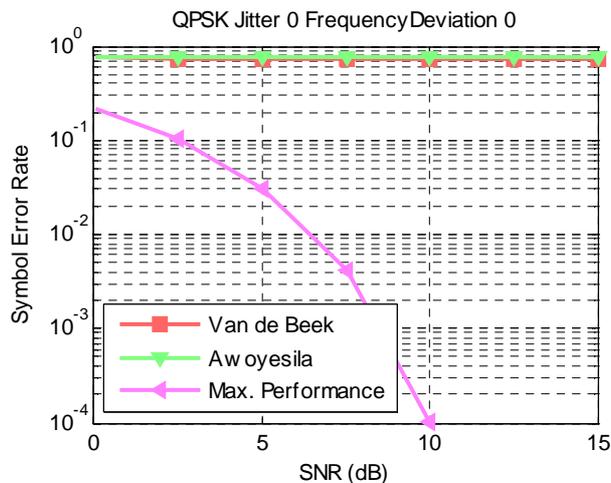


Figure 38: Performance of the algorithms when no deviations are introduced

The results show what seems to be a very bad performance of the synchronization algorithms. Both algorithms achieve more than the 80 % of the erroneous received symbols and are very far of the maximum performance curve.

The reason that explains this behaviour is that after the frequency deviation recovery, there is always a remaining frequency deviation due to errors in the estimation process. These errors are common in every synchronization algorithm and do not mean they are bad algorithms. In

fact, their estimation of the correct frequency and time offset is quite accurate, as can be seen in the following figure:

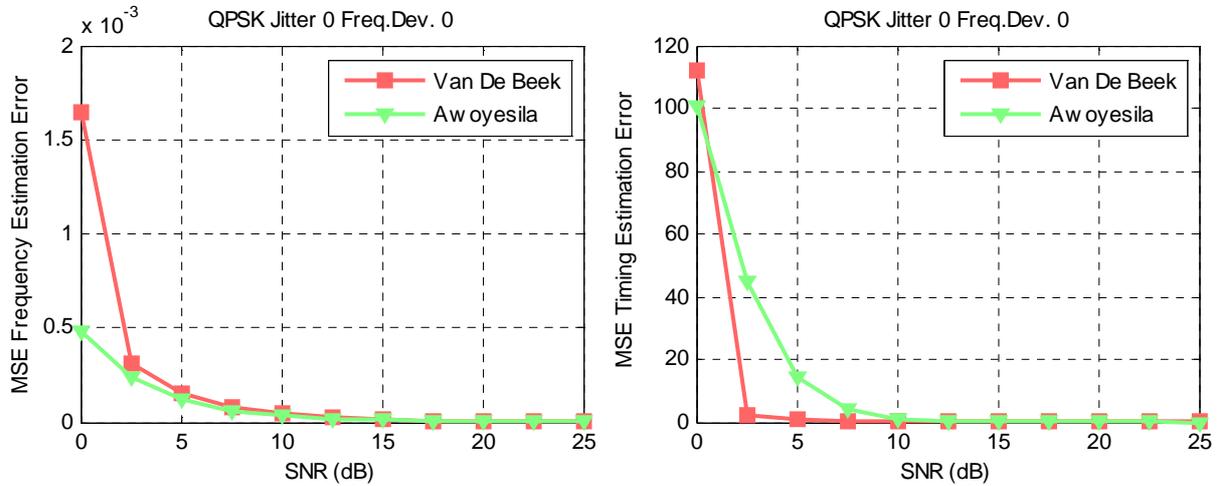


Figure 39: MSE of the estimation errors when no errors are introduced in the transmitter

However, as explained in section 1.2.2, the frequency deviation in OFDM adds a time variant phase error which rotates the constellation according to the factor  $e^{\frac{j2\pi k\epsilon}{N}}$ . Due to the frequency deviation estimation errors, there is still a residual frequency deviation that rotates the constellation according to the factor  $e^{\frac{j2\pi k\epsilon_e}{N}}$ . This phase error will always exist as soon as there is any remaining frequency deviation, so it is necessary to compensate this rotation.

The solution adopted to compensate it, is to use the pilot subcarriers specified by the standard. Since the phase of the received symbols is known, as well as the phase that the pilots should have, we can easily estimate this phase deviation and correct the whole constellation. This solution can only be adopted because only a small residual frequency deviation ( $\epsilon_e$ ) is left due to the errors in the frequency deviation estimation. If the frequency deviation had not been reduced before by the frequency estimation synchronization algorithm, it would have been impossible to recover the original symbols only with the subcarrier pilots, since such a great frequency deviation introduces such a high ICI that completely damages the whole OFDM symbol.

However after applying these changes, there is still a bad performance:

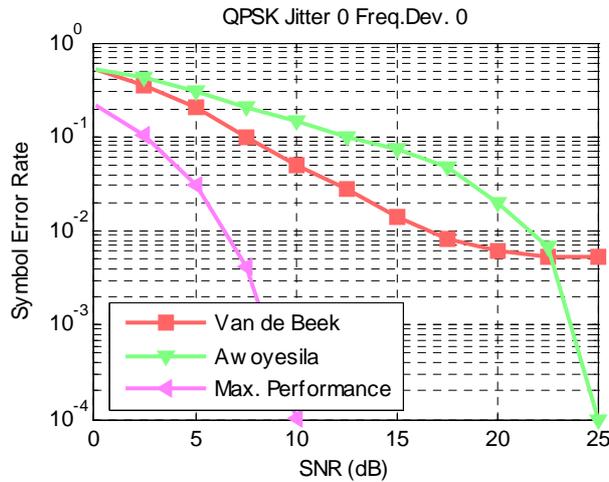


Figure 40: Performance of the algorithms after applying the rotation compensation

The new reason that explains this bad performance is related to errors in the timing offset estimation process. The algorithms usually provide the correct integer timing offset value, but sometimes they also provide an erroneous value, which translated to the FFT affects the whole OFDM symbol.

To reduce the number of errors, the second solution proposed is to realize an average of the estimated timing offset over time. By doing this, the average smoothes the estimation along time, avoiding the instantaneous errors on the timing offset estimation process to corrupt so many symbols. The curves obtained by applying this second solution are depicted in the following figure.

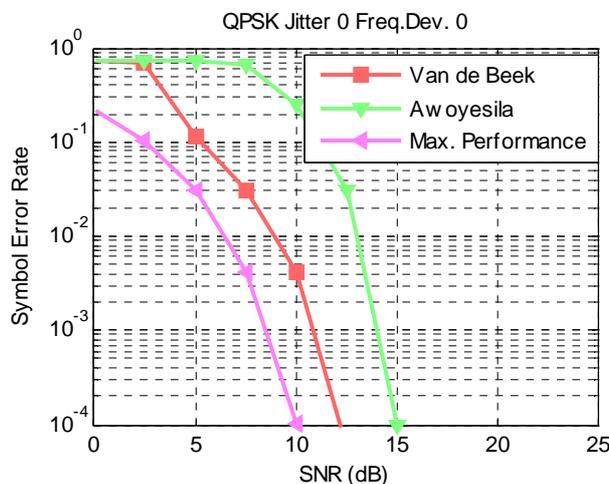


Figure 41: Performance of the algorithms after applying the average of the timing offset estimation

This is the best performance that has been achieved without inserting any timing offset or frequency deviation. It is important to remember that in real systems there is always added protection, like FEC codes, scrambling or spreading of the information, etc. That would help to completely recover the original data. But in this thesis, they do not have been implemented to focus on the performance of the synchronization process.

Finally, the performance of the algorithms when using the 16-QAM modulation together with the two proposed solutions is showed:

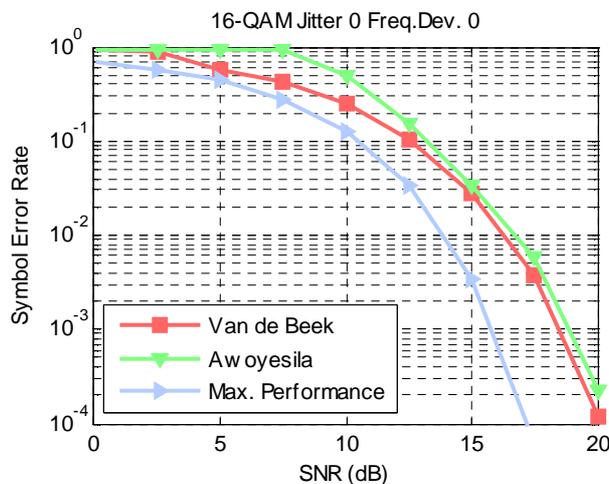


Figure 42: Performance with a 16-QAM modulation after applying the two proposed solutions

As seen, the performance of the two synchronization algorithms is much close to the maximum performance when using a 16-QAM modulation.

### 3.2.2.1 Performance of the OFDM synchronization algorithms

After characterizing the behavior of the synchronization algorithms and understanding how to get the best performance from them, they can now be tested with real synchronization errors. The figure shows the results obtained after inserting a timing offset value of 3 symbols and a frequency deviation of the 25% of the subcarrier spacing ( $0.25\Delta f_{sc}$ ).

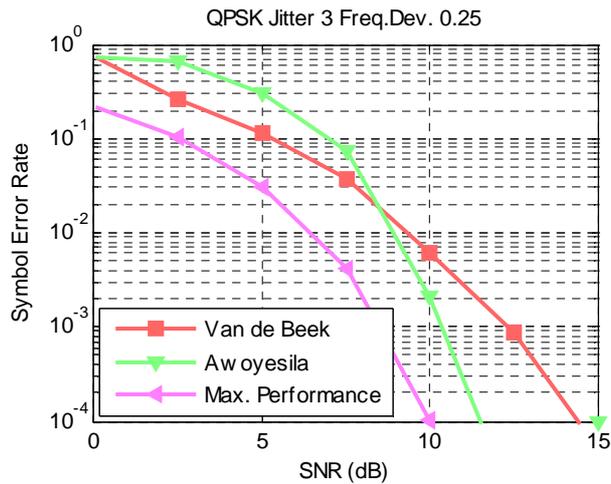


Figure 43: Performance when inserting a jitter of 3 symbols and a freq. deviation of  $0.25Af_{sc}$

It can be seen that the Aw oyesila algorithm provides better results for high SNR rates than the Van de Beek's algorithm, which provides better results for low SNR rates. The reason that explains this is that the Aw oyesila algorithm just obtains one deviation estimation when it receives the training symbol, using the same estimations for all the rest of the OFDM data symbols until a new training symbol is received. This means that if the estimates are good enough (high SNR), a lot of OFDM symbols will be properly corrected and if the estimates are bad (low SNR), a lot of OFDM symbols will be erroneously corrected. On the other hand, the Van de Beek algorithm estimates one deviation value for every new OFDM symbol, so a good or bad estimation will affect only that only OFDM symbol.

In the following figure, the timing and frequency deviation MSE of the two synchronization symbols can be seen:

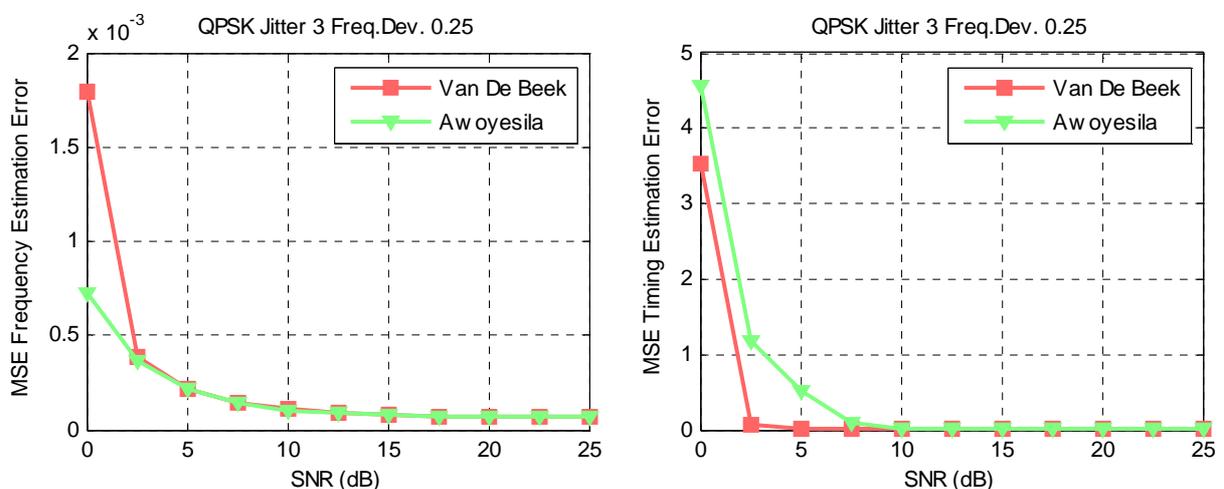


Figure 44: MSE of the estimation errors when inserting a jitter of 3 symbols and a freq. deviation of  $0.25Af_{sc}$

As seen both synchronization algorithms are quite accurate. There is a residual deviation left in both timing and frequency recovery. The frequency deviation MSE for high SNR is approximately  $6.45 \cdot 10^{-5}$  while the timing offset MSE for high SNR is approximately  $9 \cdot 10^{-3}$ .

Finally, to analyze the distribution of the errors, a SNR of 10 dB has been fixed, which implies a SER of  $6.3 \cdot 10^{-3}$  for the Van de Beek algorithm and  $2.1 \cdot 10^{-3}$  for the Awoyesila algorithm.

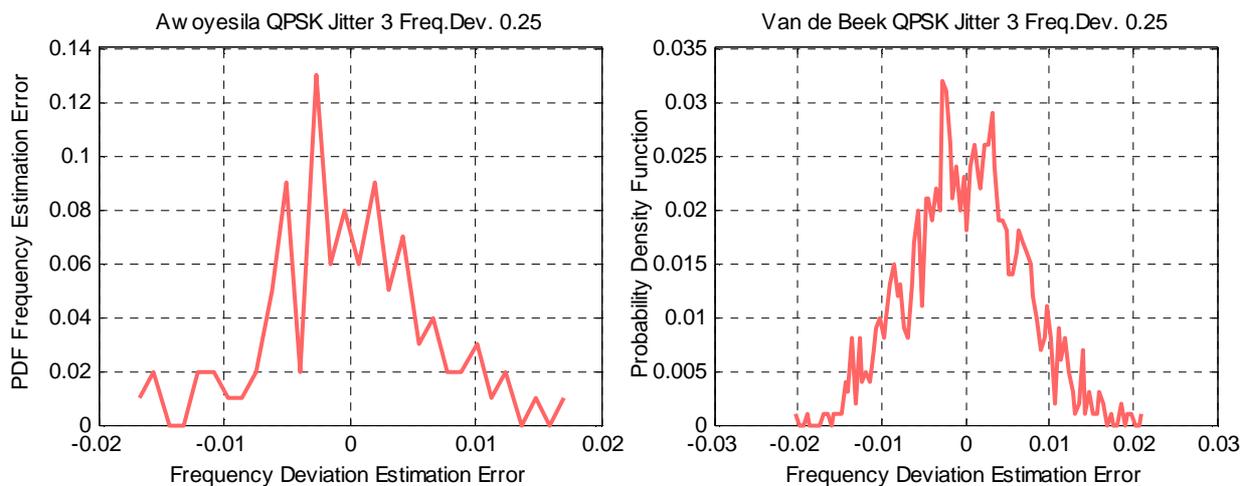


Figure 45: PDF of the frequency deviation estimation errors for 10 dB of SNR

The Probability Density Function of the frequency deviation errors shows that both algorithms show a similar precision, providing values below the 2% of the subcarrier spacing. The differences in the figures are related to the lower number of estimations that provides the Awoyesila algorithm in contrast to the Van de Beek algorithm.



## 4. Conclusions

In this diploma thesis, different options to achieve synchronization in 60 GHz systems have been studied, implemented and analyzed for Single Carrier and OFDM systems. The implementation was particularized for the IEEE 802.15.3c standard, using a Simulink environment and an AWGN channel.

For SC systems, the implemented algorithms showed similar performance under a constant timing offset and a slightly variant introduced timing offset. Also, all the algorithms were very close to the maximum performance, which proves that the timing offsets introduced were stable enough for them, so they could all challenge synchronization in those conditions.

For the OFDM system, it was necessary to characterize first the performance of the algorithms in the absence of introduced deviations, in order to minimize their self-noise. The bad performance of the algorithms related with an unavoidable time-variant phase offset produced by the remaining frequency deviation was corrected by using the pilot subcarriers defined in the standard. The other penalizing factor, the remaining timing offset after synchronization, was reduced by averaging the estimates along time. The best possible performance was then achieved.

The Van de Beek algorithm showed better performance for low SNR, with the drawback of being only able to compensate a limited fractional frequency deviation of one half of the subcarrier spacing. On the other hand, the Awoyesila algorithm showed better performance for high rates of SNR, being able to perform integer and fractional frequency deviation estimation, but with the drawback of the waste of resources that implies the need of sending a dedicated OFDM training symbol to perform synchronization.



## 5. Future work

Synchronization in digital communications systems is such a huge subject of study that it is always offering new possibilities. For that reason, many new tasks can be performed by using this thesis as a start point.

First of all, all the implemented algorithms and even the basic transmission systems can be deeply analyzed and even improved. For example, the different configurations of the pulse-shaped filters could lead to a different performance of the algorithms that would be interesting to analyze. Also, there are many algorithms in the literature which could be implemented and tested.

Secondly, the channel employed for all the simulations was the AWGN channel, which is a good starting point to analyze the basic performance of the algorithms. But it would be necessary to test the algorithms on the real transmission channels that affect every desired application of the 60 GHz transmissions systems.

Furthermore, the frame synchronization of both SC and OFDM systems could be implemented, since it is another aspect of the synchronization. This could permit to integrate the synchronization algorithms into a complete transmission system which included all the features of the standard, like Forward Error Correction (FEC), spreading and scrambling of the information, etc. The overall performance of the synchronization algorithms in a complete transmission system would be then analyzed.

In addition, this thesis has focused on testing the proposed synchronization methods for a concrete 60 GHz standard: the IEEE 802.15.3c standard. However, the task of adapting these algorithms to another 60 GHz standard like the IEEE 802.11ad standard is very simple. Even more, it would also be a straightforward task to adapt these algorithms to many other standards and transmission systems.

Finally, all this studies could be complemented by testing the performance of the algorithms on real hardware. The Software Defined Radio (SDR) solutions could be the employed hardware to realize these tests.



## List of Figures

Figure 1: Basic Single Carrier system scheme.....	1
Figure 2: Optimum sample point in the eye diagram.....	3
Figure 3: Constellations specified for the PHY SC mode by IEEE 802.15.3c standard.....	4
Figure 4: Scheme of the modulation including the $\pi/2$ rotator.....	5
Figure 5: Basic OFDM system scheme.....	6
Figure 6: Distribution of the OFDM subcarriers along time .....	7
Figure 7: Cyclic Prefix extension of the OFDM symbol. ....	7
Figure 8: Practical OFDM system scheme. ....	8
Figure 9: Timing offset in OFDM produce both ISI and ICI .....	9
Figure 10: Rotation of the points of the constellation due to the frequency deviation.....	10
Figure 11: Time-variant phase error introduced in the OFDM symbol.....	10
Figure 12: Constellations specified for the PHY HSI mode by IEEE 802.15.3c standard .....	12
Figure 13: Feedforward Algorithm for Timing Recovery structure.....	13
Figure 14: Interpolation of the N samples of the symbol to obtain the correct value.....	14
Figure 15: Square Timing Recovery Algorithm scheme .....	15
Figure 16: Feedback Algorithm for Timing Recovery structure .....	15
Figure 17: Operation principle of the Gardner algorithm .....	17
Figure 18: Operation principle of the Modified Gardner algorithm .....	18
Figure 19: Operation principle of the Early-Gate algorithm.....	19
Figure 20: Nogami's algorithm frame structure.....	21
Figure 21: Nogami's algorithm scheme .....	21
Figure 22: Peaks of $ G_k $ at the pilot frequencies.....	22
Figure 23: Schmidl's algorithm frame structure.....	23
Figure 24: Schmidl's algorithm scheme.....	23
Figure 25: Timing metric $M(d)$ .....	24
Figure 26: Awoyesila's algorithm frame structure .....	26
Figure 27: Timing metric $M_c(d)$ .....	27
Figure 28: Classen's algorithm scheme .....	28
Figure 29: Observation interval of $2N+G$ samples.....	29
Figure 30: Implementation of a SC system with a generic timing recovery algorithm .....	31
Figure 31: Comparison of the different timing recovery algorithms for different SNR .....	32
Figure 32: Comparison of the MSE timing estimation error of the different timing recovery algorithms .....	33
Figure 33: Comparison of the different timing recovery algorithms for different modulations.....	34
Figure 34: Probability Density Function of the timing offset estimation errors .....	34
Figure 35: Performance of the timing algorithms when changing the constant timing offset value .....	35
Figure 36: Comparison of the performance of the timing algorithms when changing the loop gain step value..	36

---

<i>Figure 37: Implementation of a OFDM system with a generic timing recovery algorithm .....</i>	<i>37</i>
<i>Figure 38: Performance of the algorithms when no deviations are introduced.....</i>	<i>38</i>
<i>Figure 39: MSE of the estimation errors when no errors are introduced in the transmitter.....</i>	<i>39</i>
<i>Figure 40: Performance of the algorithms after applying the rotation compensation .....</i>	<i>40</i>
<i>Figure 41: Performance of the algorithms after applying the average of the timing offset estimation.....</i>	<i>40</i>
<i>Figure 42: Performance with a 16-QAM modulation after applying the two proposed solutions .....</i>	<i>41</i>
<i>Figure 43: Performance when inserting a jitter of 3 symbols and a freq.deviation of <math>0.25\Delta f_{sc}</math> .....</i>	<i>42</i>
<i>Figure 44: MSE of the estimation errors when inserting a jitter of 3 symbols and a freq. deviation of <math>0.25\Delta f_{sc}</math> ..</i>	<i>42</i>
<i>Figure 45: PDF of the frequency deviation estimation errors for 10 dB of SNR.....</i>	<i>43</i>

## List of Tables

<i>Table 1: Timing Related Parameters .....</i>	<i>11</i>
<i>Table 2: Subcarrier frequency allocation.....</i>	<i>12</i>

## List of Acronyms

<b>ADC</b>	<i>Analog to Digital Converter</i>
<b>AWGN</b>	<i>Additive White Gaussian Noise</i>
<b>BER</b>	<i>Bit Error Rate</i>
<b>BPSK</b>	<i>Binary Phase Shift Keying</i>
<b>CP</b>	<i>Cyclic Prefix</i>
<b>DAB</b>	<i>Digital Audio Broadcasting</i>
<b>DAC</b>	<i>Digital to Analog Converter</i>
<b>DVB</b>	<i>Digital Video Broadcasting</i>
<b>FEC</b>	<i>Forward Error Correction</i>
<b>FFT</b>	<i>Fast Fourier Transform</i>
<b>GI</b>	<i>Guard Interval</i>
<b>ICI</b>	<i>Inter-Carrier Interference</i>
<b>IFFT</b>	<i>Inverse Fast Fourier Transform</i>
<b>ISI</b>	<i>Inter-Symbol Interference</i>
<b>LTE</b>	<i>Long Term Evolution</i>
<b>ML</b>	<i>Maximum Likelihood</i>
<b>MSE</b>	<i>Mean Square Error</i>
<b>OFDM</b>	<i>Orthogonal Frequency Division Multiplexing</i>
<b>PAM</b>	<i>Pulse Amplitude Modulator</i>
<b>PDF</b>	<i>Probability Density Function</i>
<b>PN</b>	<i>Pseudo-Noise</i>
<b>QAM</b>	<i>Quadrature Amplitude Modulator</i>
<b>QPSK</b>	<i>Quadrature Phase Shift Keying</i>
<b>SC</b>	<i>Single Carrier</i>
<b>SER</b>	<i>Symbol Error Rate</i>
<b>SNR</b>	<i>Signal To Noise Ratio</i>
<b>TED</b>	<i>Timing Error Detector</i>
<b>WiMAX</b>	<i>Worldwide Interoperability for Microwave Access</i>



## Bibliography

- [1] IEEE Standard for Information technology. “Part 15.3: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for High Rate Wireless Personal Area Networks (WPANs). Amendment 2: Millimeter-wave-based Alternative Physical Layer Extension”. IEEE Std 802.15.3c – 2009.
  
- [2] Umberto Mengali, Aldo N. D’Andrea, “Synchronization techniques for digital receivers”, 1997.
  
- [3] H. Meyr, M. Moeneclaey, and S. A. Fechtel, “Digital Communication Receivers”, New York: Wiley, 1998.
  
- [4] Ting-Jung Liang, “Synchronization and channel estimation for designing OFDM-WLAN systems”, Aachen : Shaker, 2008.
  
- [5] Thieny Pollet, Mark Van Blade and Marc Moeneclaey, “BER sensitivity of OFDM systems to carrier frequency deviation and Wiener phase noise”, IEEE Transactions On Communications, Vol. 43, No. 2/3/4, February / March / April 1995
  
- [6] Martin Oerder, “Digital Filter and Square Timing Recovery”, IEEE Transactions On Communications, Vol. 36, No. 5, May 1988.
  
- [7] K. Schmidt and A. Wittneben, “Systematic Complexity Reduction for Digital Square Timing”, 1992.
  
- [8] Zhang Hang and Markku Renfors, “A new symbol Synchronizer with reduced timing jitter for QAM systems”, 1992.
  
- [9] F.M. Gardner, “A BPSK-QPSK Timing-Error detector for sampled data receivers”, IEEE Transactions on Communications, Vol. 32, pp.423-429, May 1986.

- 
- [10] Weimin Leng, Yu Zhang, Zhixing Yang, "A Modified Gardner Detector for Multilevel PAM-QAM System", IEEE Transactions on Communications, Vol. 32, pp.423-429, May 1986.
- [11] Dongmin Lim, "A Modified Gardner Detector for Symbol Timing Recovery of M-PSK Signals", IEEE Transactions on Communications, Vol. 52 No 10, Oct 2004.
- [12] A. N. D'Andrea and M. Luise, "Design and analysis of a jitter-free clock recovery scheme for QAM systems," IEEE Transactions on Communications, Vol. 41, pp. 1296–1299, Sept. 1993
- [13] K.H.Mueller and M. Muller, "Timing recovery in digital synchronous data receivers", IEEE Transactions on Communications, vol. COM-24, pp.516-530, May, 1976.
- [14] B. Sklar, "Timing Digital Communications: Fundamentals and Applications", Englewood Cliffs, N.J., Prentice-Hall, 1988.
- [15] H. Nogami and T. Nagashima, "A frequency and timing period acquisition technique for OFDM systems", Personal, Indoor and Mobile Radio Communications (PIRMC), pp. 1010–1015, Sept. 27–29, 1995.
- [16] T. Schmidl and D. C. Cox, "Robust frequency and timing synchronization for OFDM" IEEE Transactions On Communications, Vol. 45, No. 12, pp. 1613 – 1621, Dec. 1997.
- [17] P. Moose, "A technique for orthogonal frequency division multiplexing frequency offset correction", IEEE Transactions on Communications, Vol. 42, pp. 2908–2914, Oct. 1994.
- [18] M. Morelli and U. Mengali, "An improved frequency offset estimator for OFDM systems", IEEE Commun. Lett., Vol. 3, No. 3, pp. 239-241, Mar. 1999.
- [19] G. Ren, Y. Chang, H. Zhang, and H. Zhang, "Synchronization methods based on a new constant envelope preamble for OFDM systems", IEEE Transactions Broadcasting, Vol. 51, no. 1, pp.139-143, Mar. 2005.

- 
- [20] A Adegbenga, B. Awoseyila, Christos Kasparis and Barry G. Evans, "Robust Time-domain Timing and Frequency Synchronization", IEEE Transactions on Consumer Electronics, Vol. 55, No. 2, May 2009
- [21] A. B. Awoseyila, C. Kasparis, and B. G. Evans, "Improved preamble-aided timing estimation for OFDM systems", IEEE Commun. Lett., Vol. 12, no. 11, pp. 825-827, Nov. 2008.
- [22] Ferdinand Classen and Heinrich Meyr, "Frequency Synchronization Algorithms for OFDM Systems suitable for Communication over Frequency Selective Fading Channels", 1994.
- [23] Jan-Jaap van de Beek, Magnus Sandell, and Per Ola B"orjesson, "ML Estimation of Time and Frequency Offset in OFDM Systems", IEEE Transactions on Signal Processing, Vol. 45, No. 7, July 1997.
- [24] Michael Speth, Stefan A. Fechtel, Gunnar Fock and Heinrich Meyr, "Optimum Receiver Design for Wireless Broad-Band Systems Using OFDM - Part I", IEEE Transactions On Communications, Vol. 47, No. 11, November 1999
- [25] Michael Speth, Stefan A. Fechtel, Gunnar Fock and Heinrich Meyr, "Optimum Receiver Design for Wireless Broad-Band Systems Using OFDM - Part II: A Case Study", IEEE Transactions On Communications, Vol. 49, No. 4, April 2001



## Declaration

I certify that the work presented here is, to the best of my knowledge and belief, original and the result of my own investigations, except as acknowledged. Any uses made within it of the works of any other author are properly acknowledged at their point of use.

Pablo Olivas González

