

Design and Implementation of Synchronization and AGC for OFDM-based WLAN Receivers

Víctor P. Gil Jiménez, Student Member, IEEE, M. Julia Fernández-Getino García, Member, IEEE, Francisco J. González Serrano, Member, IEEE and Ana García Armada, Member, IEEE

Abstract —*An efficient implementation of several tasks at the receiver becomes crucial in OFDM-based high-speed WLAN systems, such as automatic gain control, time and frequency synchronization and offset tracking. This paper deals with fixed point constraints and accuracy requirements for implementation of those algorithms. Also, a complete set of thresholds for the practical implementation of time and frequency synchronization sub-blocks is obtained. Moreover, a technique to mitigate the remaining frequency offset after coarse acquisition is proposed, yielding a good trade-off between performance and complexity. Finally, we propose the implementation of a simple and effective automatic gain control procedure.*¹

Index Terms — OFDM, Time and Frequency Synchronization, AGC, Offset Tracking, WLAN.

I. INTRODUCTION

WLAN (Wireless Local Area Network) access has recently been deployed and worldwide extended. Several standards such as IEEE 802.11b [1] (providing up to 11 Mbps) or IEEE 802.11a/g [2] [3] (increasing speed up to 54 Mbps) are transforming the way we experience personal communications. There are several other standards coming up, e.g. IEEE 802.16 [4] or IEEE 802.15G3 [5] that will increase capacity and range of coverage for wireless access.

Most of these standards are OFDM-based (Orthogonal Frequency Division Multiplexing) because of the good behavior of this technique in multi-path radio channels. However, several aspects have to be taken into account when an OFDM system is implemented. One of the most critical issues is synchronization at the receiver side. Due to OFDM properties, it has to be synchronized not only in time (acquisition) but also in frequency. In the literature there are several contributions that provide time acquisition [6], frequency estimation [7] [8] or both [9] [10]. In those papers the theoretical problem is analyzed and simulation results are provided without considering implementation constraints. In this paper the complete design from system level to

implementation that has been carried out is explained. The design has been done targeting the IEEE 802.11a standard [2] but it can be extended to other packet-based OFDM systems. The algorithms used have been taken from those previous works but they have been adapted to the physical implementation and technical problems related to trade-off design have been solved.

The implementation of a simple and multi-stop flexible Automatic Gain Control is described; it yields a good behavior not only in terms of accuracy but also on complexity and convergence time.

A complete set of thresholds has been obtained in order to efficiently implement time and frequency synchronization algorithms. Also the effect of the remaining frequency offset has been analyzed and a technique for its mitigation based on using a smart update of values is proposed, what provides a good trade-off between complexity and performance. For offset tracking an extension of [7] is proposed.

On the other hand, fixed point, efficiency and accuracy on different operations across the transmitter or receiver chain have to be taken into account when an algorithm is implemented. From this point of view, in order to improve accuracy either on operations due to fixed point devices or on the algorithm implemented for calculation of angle rotation - called CORDIC (COordination Rotation DIGital Computer) -, an iterative process to update values has been designed yielding a significant improvement in terms of accuracy.

The paper is organized as follows. In Section II a brief description of high-level system and signal model is given. Then, different receiver blocks are explained through their respective sections such as Automatic Gain Control (section III), time and frequency synchronization block (section IV) and offset tracking block (section V). In section VI performance of the proposed design is shown. Finally some conclusions are drawn.

II. SYSTEM DESCRIPTION

In this section, a brief description of the OFDM system is given and the general OFDM signal model and the IEEE 802.11a standard [2] are described in order to understand the algorithms and solutions proposed.

In Table 1 the list of symbols and definitions used across the paper is presented for reference purposes.

¹ This work has been partially funded by Spanish government with project TIC 2002-03498 (ORISE), Telefonica I+D by the contract n° 25756, and the Chamber of Madrid Community and European Social Fund by a grant to the first author.

Víctor P. Gil Jiménez et al. are with the department of Signal Theory and Communications of the University Carlos III of Madrid. Leganés (Madrid). 28911. Spain (e-mail: {vjimenez, mjulia, fran, agarcia }@tsc.uc3m.es).

Symbol	Description
$y(t)$	Intermediate frequency signal
$x(t)$	OFDM complex envelope
T_g	Guard interval
T	Symbol period
R	Instantaneous receiver power estimation
r	Time-domain received signal
Q	Auto-correlation
N_U	Number of symbols to average frequency estimation
N	Number of sub-carriers
M	Correlation metric
L	Window length for correlations
H	Channel frequency response
$g(t)$	Rectangular pulse shape
BW	Signal Bandwidth
σ_w	AWGN variance
Δf	$\Delta f = BW/N$. Frequency spacing
α	Power estimation weight
$\hat{\sigma}$	Average power estimation
Δf	Carrier Frequency spacing
$\hat{\phi}$	Estimation of the angle
$\hat{\epsilon}_r$	Remaining frequency offset
$\hat{\epsilon}_t$	Tracked frequency offset
f_e	Carrier frequency offset
ϵ	Normalized frequency offset

TABLE I. Symbols and Definitions

A. System Description

The block diagram of an OFDM system is shown in Fig. 1. At the transmitter, binary stream b is converted from serial to parallel and modulated by using the adequate complex mapping. Then pilots are inserted and the IFFT is performed in order to obtain the time domain signal, which is converted back to serial, and finally the Cyclic Prefix (CP) is included. The signal travels across the channel that corrupts it (attenuation, rotation, distortion) and then it arrives to the receiver where the signal has to be synchronized in time and frequency and the remaining processes have to be carried out inversely. By using scattered pilots at certain symbol positions the channel can be tracked and, with this additional information, demodulation can be improved by mitigating offset estimation errors. This paper is focused on the blocks highlighted in gray in Fig 1.

B. Signal Model

The complex envelope signal model (base band signal) will be used in the following sections except for the Automatic Gain Control (AGC) part where the Intermediate Frequency

(IF) signal will be adopted. The sampling frequency for base band signal is 20 MHz whereas it is 80 MHz for IF. The complex envelope for the OFDM signal is given by:

$$x(t) = \frac{1}{\sqrt{N}} \sum_{s=1}^{N_s} \left[\sum_{k=0}^{N-1} X_{s,k} e^{j2\pi k \Delta f t} \right] g(t - sT') \quad (1)$$

where $X_{s,k}$ is the modulated data sent in the k -th sub-carrier at the s -th OFDM symbol of the total of N_s symbols (note that it is a packet-based system and therefore the number of OFDM symbols is finite), $g(t)$ is the rectangular pulse shape function, Δf is the frequency spacing ($\Delta f = BW/N$), $T' = T + T_g$ is the symbol duration where T is the effective symbol time, T_g is the Cyclic Prefix duration and N is the number of sub-carriers.

After experiencing channel effects, the discrete-time signal at the receiver in time domain prior to FFT can be written as:

$$r_{s,n} = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X_{s,k} H_{s,k} e^{j2\pi n \frac{k+\epsilon}{N}} + w_{s,n} \quad (2)$$

where $H_{s,k}$ is the channel's frequency response at k -th sub-carrier for the s -th OFDM symbol, w represents the Additive White Gaussian Noise (AWGN) process, n represents time index and ϵ is the carrier frequency offset (f_e) normalized to Δf ($\epsilon = f_e/\Delta f$), due to both channel effects and misalignments between transmitter and receiver.

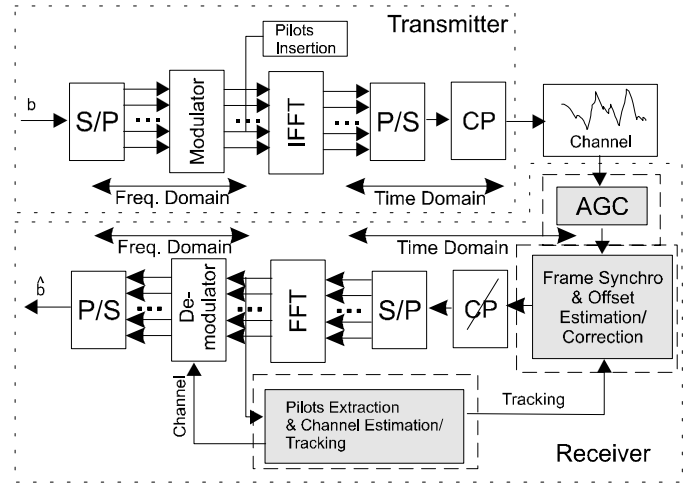


Fig 1. Block diagram of an OFDM System

C. IEEE 802.11a

The IEEE 802.11a is a packet-based OFDM standard for high-speed WLAN in the 5 GHz band. Each packet contains a header (shown in Fig. 2) for detection, channel estimation and synchronization purposes. This preamble is composed of five OFDM symbols known at both sides of the communication link. First two OFDM symbols are called Short Training

Symbols (STS) and they are intended for signal detection, AGC stabilization, diversity (if several antennas were used), coarse acquisition and frequency synchronization purposes. Those STS are formed by the repetition of ten short symbols of 16 samples each; these samples have very good correlation properties. There are in the literature several algorithms for time and frequency synchronization based on the correlation of known sequences and the repetition of symbols [7][9]; in this paper, an application of these techniques will be used for coarse and fine synchronization. On the other hand, the second part of the preamble (see Fig. 2) is composed of two identical symbols and a cyclic prefix twice longer than the CP of the other symbols. This structure is called Long Training Symbols (LTS) and it is designed to be used for channel estimation and fine frequency offset correction. The design proposed in this paper uses these LTS for channel estimation, coarse and fine frequency estimation. Last symbol in the header is a special symbol called SIGNAL which carries information about the length of the packet as well as the modulation type of the encoded data.

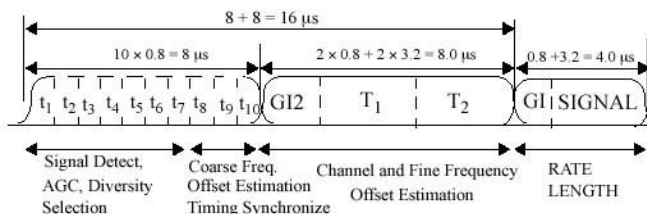


Fig 2. IEEE 802.11a header, defined in [2]

D. System Modes

During the reception of a packet, the system is moving among different states or modes. In order to synthesize this procedure a brief description of them is shown in this section.

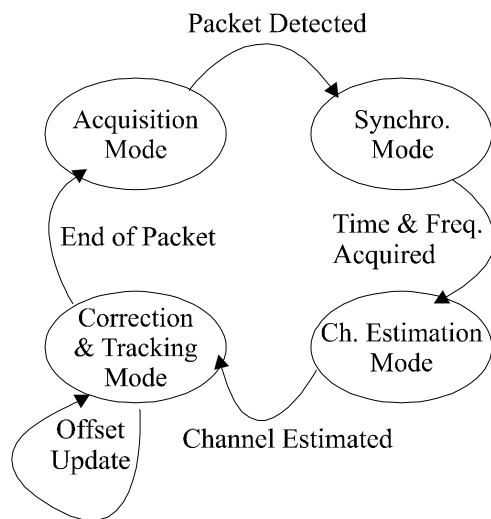


Fig 3. System Modes

In Fig. 3 the four modes and events that make transitions from one mode to another are depicted. Initially the system is in the *Acquisition Mode*. In this mode, the AGC measures the

average received power, and produces the minimum attenuation; when the received power exceeds a given threshold, the AGC produces a signal of “*detected packet*” and it starts to adjust the optimum gain. After *Acquisition Mode*, the system enters in the *Synchronization Mode*, where the time and frequency synchronization is accomplished. After that the system enter in the *Channel Estimation Mode* in which the channel is estimated and stored to be corrected for the whole packet. Finally the system is in the *Correction and Tracking mode* where the OFDM symbols are being received and corrected by using the stored estimated channel and also in this mode the offset is tracked in order to improve results. Once the last OFDM symbol is received, the system moves back to the first mode, i.e. *Acquisition Mode* waiting for a new packet to arrive.

III. AUTOMATIC GAIN CONTROL

In this section we will describe the Automatic Gain Control. First part is dedicated to give a very brief description of the hardware devices interconnected to the AGC (see Fig. 4). In the second part the detailed description of the AGC algorithm will be shown.

A. Situation of AGC in the Receiver Chain

The AGC design strongly depends on the devices it is connected to. For our proposal, we are assuming that the AGC is fed by an Analog to Digital Converter (ADC) and the signal from the AGC is applied to a variable attenuator. Once the signal is attenuated depending on the AGC result, it travels to the I-Q demodulator. Usually, ADCs provide an overrange indicator flag when an input range that exceeds the full-scale input range of the converter has arrived. This flag will be used to increase the speed of convergence of the gain control.

Usually, receivers are designed to provide adaptive sensitivity to incoming signal strength by employing variable gain (or attenuation) at the lower IF frequencies, where it is easier to manipulate the signal of interest. Typically, the AGC drives a variable gain amplifier. In this work, however, the AGC drives an IF digitally controlled attenuator in order to maintain a constant input to ADC. This can be better seen in Fig. 4.

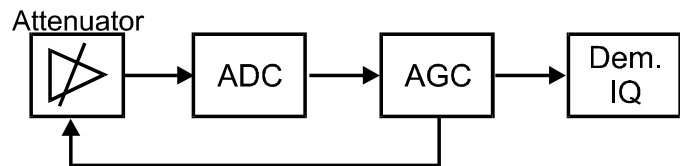


Fig 4. Situation of the AGC Block

B. Description of AGC

The AGC must track the power variations, caused by changes in channel attenuation, during the transmission of every packet. It should be noticed that in multipath fading

channels, dynamical range of the received power may be very large (60 – 80 dB). Another important issue is the high Peak-to-Average power Ratio (PAR) of the OFDM signal.

It can be inferred that the receiver AGC is an important issue that must exhibit fast initial convergence and adaptive capabilities so as to maintain an acceptable link quality. Furthermore, its algorithmic complexity must be sufficiently low to enable its implementation in low-power digital circuits at the transceiver.

The Automatic Gain Control algorithm can be divided into three parts:

- 1) Average Power estimation
- 2) Packet detection
- 3) Attenuation computation.

1) Average Power Estimation

The average power is continuously estimated by using the following recursive equation:

$$\hat{\sigma}(p) = \alpha \cdot y^2(p) + (1 - \alpha)\hat{\sigma}(p-1) \quad (3)$$

where $y(p)$ is the Intermediate Frequency received signal at the p -th sample after the ADC and α takes into account how much memory the estimator has: the larger α is the less memory the estimator has. In our design α has been fixed to 0.125 for two reasons: First, the estimation must be stable and avoid noise and therefore small values of α are better; due to the latency of the analog devices as the attenuator the impact in signal level is not shown immediately but after several samples. Second reason is the simplicity, since 0.125 implies dividing by 8 which can be done easily by shifting 3 bits to the right. However, the algorithm does not need the estimation of the mean power but only an energy reference. Therefore the absolute value has been used instead of the exponentiation, i.e. exchange y^2 by $||\cdot||$ in (3). In this way the algorithm is simpler and faster.

2) Packet Detection

The simplest approach to detect a packet, while spending the minimum amount of signal processing, is to sense for an increase in the ambient power of the environment. A power detector can be implemented in the analog or digital domain. Here a threshold ($Th1$) is set and the digital processing is triggered when power is seen above this threshold.

3) Attenuation computation

The optimal attenuation is computed as follows. Before arrival of any packet, the AGC remains in a reset state. In the reset state, all the counters are set to zero and the digitally controlled attenuation is set to the minimum value (0 dB,

disregarding the insertion losses). After detecting a packet, an asynchronous up counter drives the digital attenuator. Since the power estimation requires a few samples to produce a stable output, the counter's clock speed should be lower than that of the ADC. In our work, the counter's clock is obtained by dividing 4 times the ADC clock (that runs at 80 MHz). In order to speed up the initial convergence of the algorithm, the counter's clock signal is generated after a binary OR of the down-sampled ADC's clock and the ADC's over-range bit. It should be noticed that when this bit is high, the input signal level exceeds the range of operation of the ADC, and, for this reason, it should be attenuated.

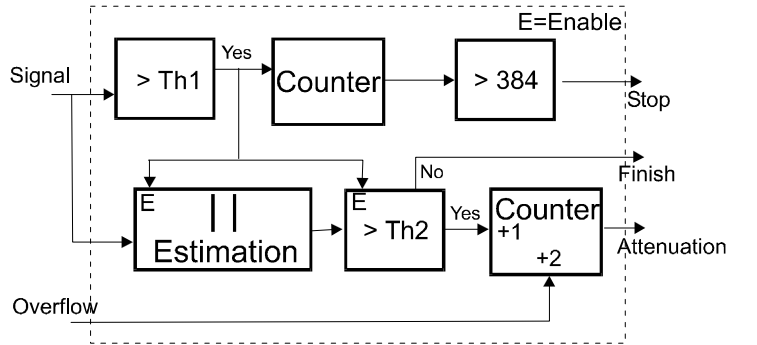


Fig 5. AGC Scheme

There are two stopping criteria for the up counter that drives the attenuator. One is when the estimated power $\hat{\sigma}$ has reached the desired level ($Th2$ in Fig. 5); and the other one is when the time index p (we assume that the arrival of a packet is detected at time 0) is larger than 384 (the number of samples at 80 MHz of the first 6 STS). Stopping the algorithm before the first 6 STS have finished allows the remaining algorithms to be able to work properly even when the packet has arrived very damaged. However the algorithm continues working until it reaches the correct value; nonetheless with the first 384 samples it gives an accurate approach to the correct value. In this way this scheme is not only fast but also robust.

IV. TIME AND FREQUENCY SYNCHRONIZATION

It is well known that time and frequency synchronization in OFDM is a critical task that severely impacts on the performance of the system [11][12]. In this section, these problems will be studied and solutions will be proposed.

A. Time Synchronization

Once the received signal levels have been adapted and after I-Q demodulation has been carried out, the packet has to be synchronized in time (*Acquisition Mode*) in order to determine where the beginning instant of the packet and therefore each OFDM symbol is. As pointed out before, STS can be used for this purpose. Based on algorithm in [9], if the first half part of the training symbol is identical in time-domain to the second part except for the phase shift caused by the carrier frequency offset, we are able to determine not only the first sample of the

packet but also a good estimation of the frequency offset by using the following metric:

$$M(n) = \frac{|Q(n)|^2}{(R(n))^2} \quad (4)$$

where n denotes time-index². The numerator in (4) is evaluated as:

$$Q(n) = \sum_{m=0}^{L-1} (r_{n+m}^* r_{n+m+L}) \quad (5)$$

where r_n is the time-domain n -th received sample³, $(\cdot)^*$ denotes complex conjugate and L is the number of samples in the first half part of the symbol.

On the other hand, the denominator represents the received energy of the second part of the symbol and it is calculated using:

$$R(n) = \sum_{m=0}^{L-1} |r_{n+m+L}|^2 \quad (6)$$

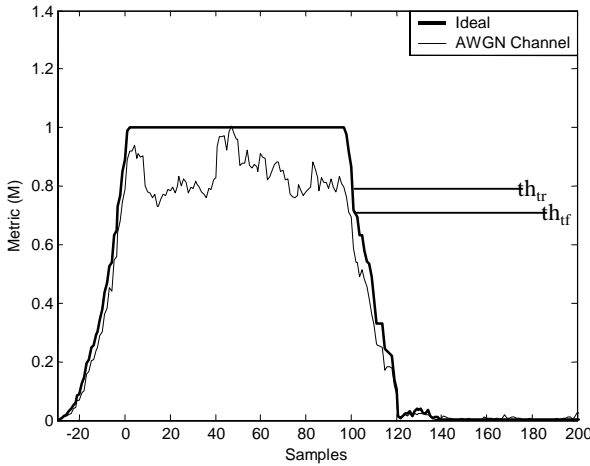


Fig 6. Metric Example by using STS

1) Characteristics of the Metric and values of L

In the IEEE 802.11a standard [2] STS are formed by ten identical short symbols of 16 samples each. In this way, L can be fixed to 16 or 32 depending on how long the correlation window is desired. Let us discuss about different values of L . If L is set to 16, STS seems to be divided into five different symbols. However, the larger the window, the more stable the estimation is due to the fact that the correlation is using more

² It should be noted that here n represents time-index after the signal has been IQ de-modulated at 20 MHz. In the previous section p denoted time-index before crossing the IQ-demodulator at 80 MHz.

³ The first sub-index in the received signal r_n has been removed in the remaining of the paper for the sake of simplicity since STS are just first and second symbols.

samples. A larger value of L as 32 can be used since STS are long enough to handle it (note that in this situation STS would be only divided into two different symbols). In order to keep a window of 64 samples ($L = 32$) the AGC has to be stable by only using first 96 samples⁴ of the STS, i.e. six first short symbols. Our AGC design is designed to be stopped when desired (as mentioned in III), and it is fast enough to attain convergence in 96 samples. From this point of view it can be combined with the long window acquisition, but our simulations have shown better performance when $L = 16$ was chosen.

Due to the STS correlation characteristics the appearance of the metric is as shown in Fig. 6. The metric can be interpreted as a normalized autocorrelation of the training sequence in order to avoid the power fluctuation caused by channel impairments. Ideally, when the metric reaches its maximum value, i.e. one, synchronization is acquired as it can be seen in Fig. 6. This figure shows the metric both in an ideal situation (without noise or channel effect) and with Additive White Gaussian Channel (AWGN) when a certain number of samples of noise (in this case 30) has been inserted at the beginning of the packet simulating an unsynchronized scenario. It can be observed that the metric follows a curve starting with a ramp that finishes in a flat region and at the end the metric falls down. For the ideal case, the synchronization is very simple: only waiting until the metric in (4) reaches one. On the other hand, for the AWGN scenario it is slightly different but still quite easy to be implemented: to wait until the metric reaches a certain threshold. Thus, the threshold value becomes an important issue and the performance will highly depend on it. If the threshold is too high the probability of missing a packet will increase and if the threshold is too low the false alarm probability or premature synchronization will go up too. The optimum value of the threshold has been studied by simulations, yielding as a conclusion that it depends on the channel, i.e. for AWGN channels a value about 0.6 is the best choice but however for multi-path channels as HiperLAN 2 A or B [13] optimum value is around 0.8 (see Fig. 6 and Table 2 for more details).

2) Implementation Issues

Up to here this applies if and only if the whole STS are received. If some part is missing, which is the common situation because the AGC block needs some samples to become stabilized, the first ramp disappears and then it is not possible to use the threshold in the way mentioned before. Nevertheless the threshold can still be taken in the descent slope. The problem with this solution is that the last part of the curve is very noisy due to the accumulation of impairments (as shown in Fig. 6). The way to proceed is as follows: once the AGC becomes stabilized, the metric in (4) is computed (note that before the AGC becomes stable no samples arrive to the

⁴ 96 samples at 20 MHz are equivalent to 384 samples at 80 MHz.

synchronization module). Now the system waits until the metric begins to fall down but in order to avoid noise effects it is necessary that a minimum value is reached by the metric, i.e. another threshold must guarantee that the signal is not too much distorted. For this reason two different thresholds are needed, one to avoid distortion effects and the other to estimate the beginning of the packet. Again, optimum values for thresholds have been obtained via simulation, yielding values of 0.75 for both thresholds (see Fig. 6 and Table 2).

It should be noted that if the synchronization algorithm estimates the beginning of the packet too early (no more than a few samples) the system will use data from the cyclic prefix which have valid information maybe corrupted by Inter Symbol Interference (ISI) and therefore the only effect will be a small degradation due to ISI (small at the end of CP if present) and a rotation in the output of the FFT. That rotation will be estimated by the channel estimation block as part of the channel effect. Therefore, a relative early synchronization can be allowed. However, if the synchronization block estimates the beginning too late the system will use data from next symbol and therefore data will be corrupted by information from contiguous symbol causing ISI. In order to mitigate this problem, a possibility for OFDM systems is the use of the so called pre-guards, i.e. when the symbol arrives instead of demodulating by starting the FFT processing after the CP, the FFT window is taken a few samples in advance using last samples in the CP (see Fig. 7 for more details). In this way, if the synchronization is not perfect data can still be recovered. However those pre-guards have to be short (no more than four samples) because they take data from the CP and it may be distorted by ISI. The benefits of using pre-guards will be seen in section VI.

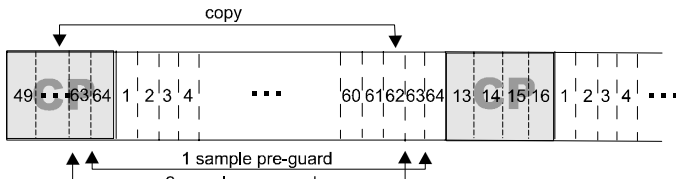


Fig 7. Pre-guards example

B. Frequency Synchronization

Another important issue is frequency synchronization to estimate and compensate possible misalignments between local oscillators or impairments caused by time-varying nature of the channel. Also, in [12] the effect of the residual frequency offset is analyzed. Once the packet has been acquired the system has an estimate of time reference. But frequency offset correction is still needed for good performance [12]. In [9] it is shown that having two equal halves of an OFDM symbol the differences between both parts provide information about the carrier frequency offset and this can be estimated with the same correlation as in (4) in the following way:

$$\hat{\Delta}_f = \hat{\phi} / (\pi T) \quad (7)$$

where T is the symbol period and $\hat{\phi}$ is an estimation of the angle between the two halves, that can be calculated from (5) as:

$$\hat{\phi} = \angle Q(d) \quad (8)$$

1) Estimation

Again STS can be used for these purposes. However, inside the packet header the LTS are also included, which have a similar structure to the STS, and that can be used for frequency offset estimation too. Thus, a first approach with STS⁵ can be done and later a fine acquisition with LTS is possible after coarse offset has been removed. Our simulations show that remaining carrier offset after using the STS is quite small $((\phi - \hat{\phi}) / \Delta f \approx 5 \cdot 10^{-3})$ but it is large enough to degrade the performance of the system, so that frequency offset has to be re-estimated.

As pointed out, a good solution may be the use of LTS for this second tracking step. Nevertheless, it should be noted that eventually these algorithms have to be implemented in a real system, and based on our experience, there exist several constraints in operations: complex rotation or angle estimation. Moreover, if they are implemented by using the well-known CORDIC algorithm [14], the accuracy of this algorithm is not good enough for small angles [15]. For this reason it is a better solution the use of only one-step estimation. Instead of using STS for this estimation, LTS have been selected.

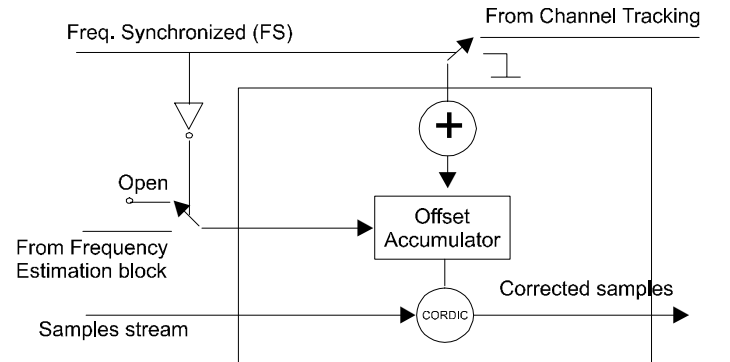


Fig 8. Frequency Offset Correction Block

Since LTS are two symbols of 64 samples each, the correlation window is of 128 samples ($L = 64$). In this case, two new thresholds are needed: first one is employed to check if the metric reaches the minimum (a value of 0.3 has been obtained by using simulations) and the other one to determine the optimum sample to estimate the carrier offset. Our simulations show that thresholds are different for time and frequency synchronization. Thresholds can be implemented by

⁵ It should be noted that STS may be very damaged because of AGC acquisition and usually they are not complete.

using registers, which are fast and not too expensive in resources.

2) Correction

After carrier offset has been estimated, it has to be corrected with the block shown in Fig. 8. This block allows two states driven by the signal *Frequency Synchronized (FS)*. When the system is in *Synchronization Mode*, the Offset Accumulator is fed by the estimation done in the frequency estimation block and there is no update to the accumulator. Once the system is in *Correction and Tracking Mode* the Offset Accumulator stores the first estimation and it is updated with the new values obtained from the channel tracking process. The correction is carried out by using the CORDIC algorithm. In this way, values to be corrected by the CORDIC module are large enough even when the system is in the tracking mode because correction is carried out with the whole carrier offset but not only with the remaining offset (see Fig. 8).

Threshold	Value
Time Synchronization raising (th_{tr})	0.8
Time Synchronization falling (th_{tr})	0.75
Frequency Synchronization raising (th_{fr})	0.3
Frequency Synchronization falling (th_{fr})	0.75

TABLE II. Summary of thresholds for time and frequency synchronization

V. FREQUENCY OFFSET TRACKING

After the channel has been estimated the receiver passes to next status: *Correction and Tracking Mode* (see Fig. 3 for more details).

IEEE 802.11a standard provides four pilot tones scattered across every OFDM symbol [2], that can be used for frequency offset tracking. Before explaining the technique to track this offset, it is important to show why this step is required.

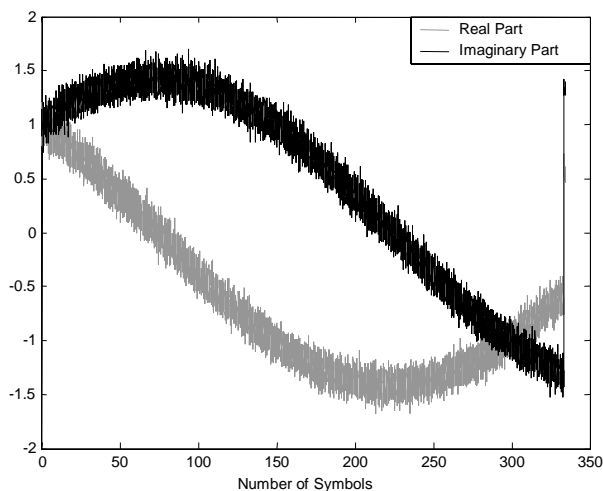


Fig 9. Effect of the residual frequency offset

Fig. 9 shows the effect of the remaining frequency offset on the demodulated real and imaginary parts of the signal. In this figure, a packet of more than 300 OFDM symbols has been transmitted and analyzed. In order to show the effect of frequency offset the data (payload) were composed of ones (“1”) QPSK-modulated into the OFDM sub-carriers. The packet is perfectly synchronized after passing through an HiperLAN 2B channel [17] with a reasonably high SNR (20 dB), and the only effects present in Fig. 9 are both errors in channel estimation (Mean Square Error = 10^{-2}) and residual frequency offset (around $\epsilon = 5e-4$). It can be observed that the effect due to the remaining offset is a rotation of the data. This rotation is not too relevant if the packet is shorter than 20 OFDM symbols, yielding a slight performance degradation. Nevertheless, for packets longer than 20 OFDM symbols, rotation is such that errors made by the slicer will overcome the hits. This effect has to be taken into account and corrected via channel tracking.

A. Estimation

Once the need of tracking the channel has been shown we will present a way to perform it; a similar approach as in [9] is given in [7]. The difference is that correlation in [7] is performed in frequency domain instead of time domain. Equation (4) therefore can be adapted to frequency domain and it can be used for remaining frequency offset estimation, applied only to those scattered pilots as follows:

$$O_s = \sum_{k=1}^{N_p} (\beta \cdot P_{s,k}^* \cdot P_{s+1,k}) \quad (12)$$

where $P_{s,k}$ is the k -th pilot at the s -th OFDM symbol, N_p is the number of pilots (in this case $N_p = 4$) and $\beta = 1$ when both pilots have the same sign and $\beta = -1$ when signs are opposite in order to correlate always the same pilots values and to have constructive contributions for the estimation. The use of β is due to the fact that IEEE 802.11a standard [2] establishes a pseudo-random sequence to choose when the pilot sequence is positive and when is negative, what has to be taken into account for the implementation. The remaining offset is then estimated as:

$$\hat{\epsilon}_r(s) = \frac{1}{2\pi} \angle O_s \quad (13)$$

B. Averaging and Correction

However, since there are only four pilot tones available and this is not enough to achieve a good estimation, an average over N_U OFDM symbols is performed in order to improve

estimation:

$$\hat{\epsilon}_t = \frac{1}{N_U} \sum_{s=1}^{N_U} \hat{\epsilon}_r(s) \quad (14)$$

By using this average, results similar as if the number of pilots used were larger than four are obtained (assuming that the coherence time of the channel is larger than the duration of the N_U OFDM symbols). The more symbols are used for averaging, the more accurate the estimation is, but also more memory is required and mismatch effects are more severe. There is a trade-off between accuracy and complexity. Several values of N_U have been used and a value of 10 has been chosen for three reasons. First, different values for N_U have been simulated and good enough results have been obtained with this value. Second, since four pilots are used for every symbol, by averaging over 10 symbols is equivalent in practice to using 40 samples, that are almost the same number of samples as in the preamble. Third, as it can be observed in Fig. 9 for packets longer than 20 OFDM symbols, the effect of the offset is considerable, so this number of 20 can be used as an upper limit of maximum number of OFDM symbols that can be averaged (10 is half of this value). The complexity of storing 10 offsets to average them is not too significant. In Fig. 10 the same packet as in Fig. 9 is shown but with a channel tracking step. It can be observed how rotation is almost avoided, only remaining effect is due to channel estimation error...

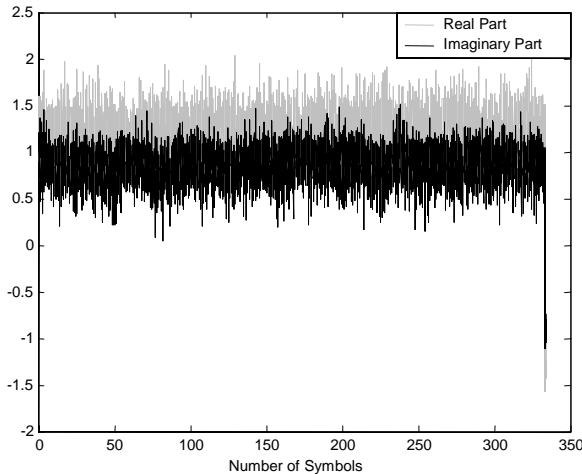


Fig 10. Demodulated data with a frequency offset tracking step

VI. PERFORMANCE EVALUATION

In this section, performance curves for the algorithms described in previous sections will be presented to show how requirements have been fulfilled.

First of all, in Fig. 11 histogram of time synchronization error (measured in number of samples) is shown. This figure has been obtained for different Signal to Noise Ratios (SNR) ranging from 15 to 25 dB, and for different frequency offsets ($\epsilon = f_e / \Delta f$ from 0 to 0.5, averaged in the results) in an HiperLAN 2 B channel [17]. Both thresholds (th_{tr} and th_{tr})

have been fixed to those in Table 2. It can be seen that time synchronization algorithm works quite well in the whole range.

There exists high probability of perfect synchronization, (mean around 0.4) and the probability of having only one sample of error is also high.

As pointed out before, the system can handle a small error in time synchronization by using pre-guards. In our design with only one sample of pre-guard⁶ the averaged probability of synchronization will be of 90 % for the whole range.

In section IV it was shown that it is better to synchronize in advance than after time. In Fig. 11 it can be seen that if synchronization is not perfect there is a high probability that algorithm estimates in advance the beginning of the packet (negative values).

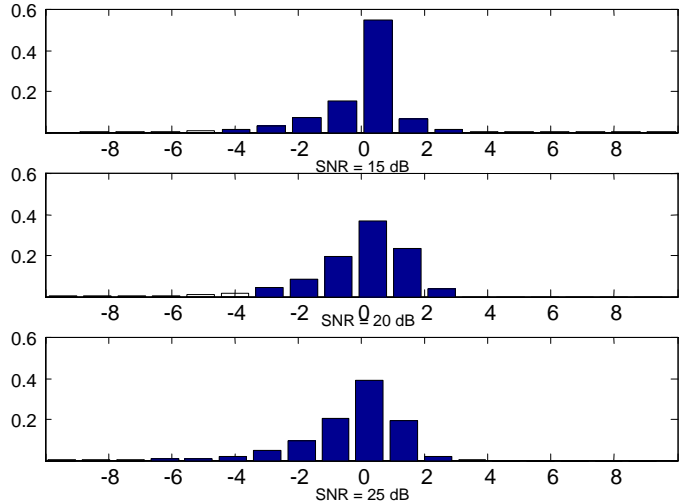


Fig 11. Averaged Time Synchronization Error

In Fig. 12 the benefits of tracking the channel can be checked. The figure has been obtained considering packets of different lengths, namely, short (30 OFDM Symbols), medium (100 OFDM symbols) and long (320 OFDM symbols) QPSK-modulated and without coding, i.e. the Bit Error Rate (BER) shown in Fig. 12 is raw BER.

As shown in section V for short packets (up to 20 OFDM symbols) the effect of the remaining offset on the performance can be afforded. However for longer packets the rotation introduced by this remaining offset is large enough to move the constellation outside the decision bounds of the demodulator and therefore BER increases dramatically. It can be seen in Fig. 12 that short packets, when the tracking algorithm is not applied, suffer a small degradation. However for longer packets the effect is too high and the system can not be used anymore. On the other hand, when the channel is tracked, performance is such as if no offset would exist independently on how long the packet is. Error floor in Fig. 12 is due to the HiperLAN 2 B channel. As outlined before, the complexity introduced by this tracking algorithm is not very high and benefits are large enough to justify its use.

⁶ By using only one sample the degradation can be neglected because the system uses the last sample from the CP which usually is not corrupted by ISI.

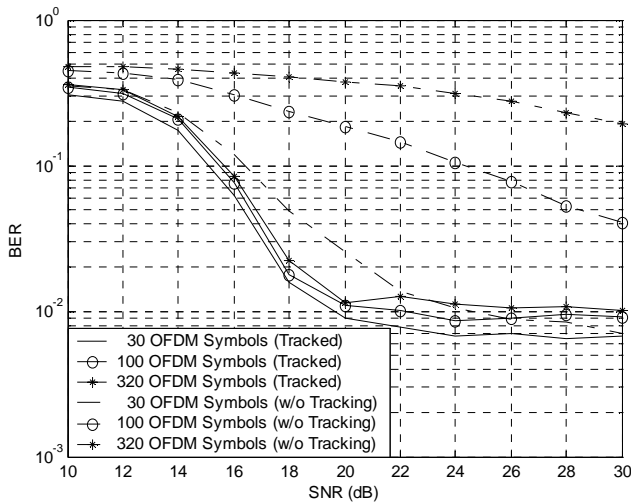


Fig 12. Raw BER for HiperLAN 2B channel with and without channel tracking

VII. CONCLUSIONS

In this paper the AGC, Synchronization and Tracking offset design for an IEEE 802.11a receiver has been shown. Different implementation problems such as AGC fast-convergence and accuracy problems across the receiver chain have been solved. A set of thresholds for synchronization blocks have been analyzed and designed. The complexity introduced has a negligible impact into the physical design. Besides, an approach to track the offset has been proposed and analyzed yielding significant improvements. All these algorithms have been implemented and successfully tested and results obtained by using these solutions have accomplished technical requirements.

REFERENCES

- [1] "WLAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: High-Speed Physical Layer extension in the 2.4 GHz band". IEEE 802.11b, 1999.
- [2] "WLAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: High-Speed Physical Layer in the 5 GHz Band". IEEE 802.11a, 1999.
- [3] "WLAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Further Higher Data Rate Extension in the 2.4 GHz Band". IEEE 802.11g, 2003.
- [4] "Part 16: Air Interface for Fixed Broadband Wireless Access Systems. Amendment 2: Medium Access Control Modifications and Additional Physical Layer Specifications for 2-11 GHz". IEEE 802.16a, 2003
- [5] "Multiband OFDM physical Layer Proposal for IEEE 802.15 task Group 3a". IEEE 802.15.3a, 2003.
- [6] B. Yang, K. B. Letaief, R. S. Cheng and Z. Cao. "Timing Recovery for OFDM Transmission". IEEE Journal on Selected Areas in Communications. Vol 18, N° 11, Nov. 2000.
- [7] P. H. Moose. "A technique for orthogonal frequency division multiplexing frequency offset correction". IEEE Transactions on Communications, Vol. 42, N°10, pp: 2908-2914, Oct. 1996.
- [8] J. Li, G. Liu and G. B. Giannakis. "Carrier Frequency Estimation for OFDM-based WLANs". IEEE Signal Processing Letters. Vol 8, N° 3, Mar 2001.
- [9] T. M. Schmidl, D. C. Cox. "Robust Frequency and Timing Synchronization for OFDM". IEEE Transactions on Communications, Vol. 45, N°12, pp: 1613-1621. Dec. 1997.

- [10] H. Minn, V. K. Bhargava and K. B. Letaief. "A Robust Timing and Frequency Synchronization for OFDM Systems". IEEE Trans. On Wireless Communications. Vol 2, No 4, Jul. 2003.
- [11] T. Pollet, M. Van Bladel and M. Moeneclaey. "BER sensitive of OFDM systems to carrier frequency offset and Wiener phase noise". IEEE Transaction on Communications. Vol 43, N° 2/3/4, pp:191-193, Feb./Mar./Apr. 1995.
- [12] X. Wang, T.T. Tjhung, Y. Wu and B. Caron. "SER Performance Evaluation and Optimization of OFDM System with Residual Frequency and Timing Offset from Imperfect Synchronization". IEEE Transaction on Broadcasting. Vol. 49, N° 2, pp: 170-176, Jun. 2003.
- [13] J. Medbo and P. Schramm, 'Channel Models for HIPERLAN/2 in Different Indoor Scenarios', ETSI EP BRAN 3ERI085B, 30 Mar. 1998.
- [14] Ray Andracka, FPGA '98. Proceedings of the 1998 ACM/SIGDA sixth international symposium on Field programmable gate arrays, Feb. 22-24, 1998, Monterey, CA, pp 191-200.
- [15] K. Kota and J.R. Cavallaro. "Numerical accuracy and hardware tradeoffs for CORDIC arithmetic for special-purpose processors". IEEE Transactions on Computers. Vol. 42 (7), Pp.: 769-779, Jul 1993.
- [16] Baoguo Yang, Zhigang Cao and Khaled Ben Letaief. "Analysis of Low-Complexity Windowed DFT-Based MMSE Channel Estimator for OFDM Systems". IEEE Trans. On Communications. Vol 49, N° 11, pp: 1977-1987, Nov 2001.
- [17] "Channel models for HiperLAN 2 in Different Indoor scenarios". Technical report, ETSI – BRAN, 1998.



Víctor P. Gil Jiménez (S'99) received the B. Eng. in Telecommunications with honors from University of Alcalá in 1998 and the M. Eng. in Telecommunications from the University Carlos III of Madrid in 2001. He is working towards the PhD. Currently, he is with the Department of Signal Theory and Communications of University Carlos III of Madrid as a researcher. He was at the Spanish Antarctica Base I in 1999 as Communications Staff. He visited the University of Leeds in 2003. His research interests include multicarrier communications and signal processing for wireless systems.



M. Julia Fernández-Getino García (S'99, AM'02, M'03) received the M. Eng. and Ph.D. degrees in telecommunication engineering, both from the Polytechnic University of Madrid, Spain, in 1996 and 2001, respectively. Currently, she is with the Department of Signal Theory and Communications of Carlos III University of Madrid, Spain, as an Assistant Professor since 2001. From 1996 to 2001, she held a research position at the Department of Signals, Systems and Radiocommunications of Polytechnic University of Madrid. She was on leave during 1998 at Bell Laboratories, Murray Hill, NJ, and visited Lund University, Sweden, during two periods in 1999 and 2000, and Politecnico di Torino, Italy, during 2003. Her research interests include multicarrier communications, coding and signal processing for wireless systems. In 1998 and 2003, she respectively received the best 'Master Thesis' and 'Ph.D. Thesis' awards from the Professional Association of Telecommunication Engineers of Spain, and in 1999 and 2000, she was respectively awarded the 'Student Paper Award' and 'Certificate of Appreciation' at the IEEE International Conferences PIMRC'99 and VTC'00. In 2004, she was distinguished with the 'Ph.D. Excellence Award' from the Polytechnic University of Madrid.



Francisco J. González Serrano (M'94) received the Ingeniero de Telecomunicación and Doctor Ingeniero de Telecomunicación degrees from the Universidad de Vigo, Spain, in 1992 and 1997 respectively. He is now an Associate Professor at the Departamento de Teoría de la Señal y Comunicaciones, Universidad Carlos III de Madrid, Spain. His research interests include digital signal processing, wireless communications and blind source separation techniques.



Ana García Armada (S'95 AM'98, M'00) received the Telecommunication Engineer degree from the Polytechnic University of Madrid (Spain) in July 1994 and Ph.D. in Electrical Engineering from the Polytechnic University of Madrid (Spain), February 1998. She is currently working as an Associate Professor at the University Carlos III de Madrid, Spain where she has occupied several management positions. She has participated in several national and international research projects, most of them related to Orthogonal Frequency Division Multiplexing (OFDM). She is co-author of three books on wireless communications. She has 10 published papers in international journals and more than 30 papers in international conferences. She has contributed to international organisms such as ITU and ETSI. She has performed research stays in ESA-ESTEC, Kansas University and Stanford University. Her research interests are simulation of communication systems and multicarrier modulations.