Document downloaded from:

http://hdl.handle.net/10251/46598

This paper must be cited as:

Canet Subiela, MJ.; Almenar Terre, V.; Flores Asenjo, SJ.; Valls Coquillat, J. (2012). Low Complexity Time Synchronization Algorithm for OFDM Systems with Repetitive Preambles. Journal of Signal Processing Systems. 68(3):287-301. doi:10.1007/s11265-011-0618-6.

The final publication is available at

http://dx.doi.org/10.1007/s11265-011-0618-6

Copyright Springer Verlag (Germany)

Low complexity time synchronization algorithm for OFDM systems with repetitive preambles

Authors:

M. J. Canet, V. Almenar, S. Flores and J. Valls

Instituto de Telecomunicaciones y Aplicaciones Multimedia.

Universitat Politècnica de València.

C/ Paranimf, 1

46730 Gandia, SPAIN

Corresponding author: M.J. Canet

Phone Number: +34 962849300 (ext. 43573)

Fax: +34 962849313

macasu@eln.upv.es

Abstract:

In this paper, a new time synchronization algorithm for OFDM systems with repetitive preamble is proposed. This algorithm makes use of coarse and fine time estimation; the fine time estimation is performed using a cross-correlation similar to previous proposals in the literature, whereas the coarse time estimation is made using a new metric and an iterative search of the last sample of the repetitive preamble. A complete analysis of the new metric is included, as well as a wide performance comparison, for multipath channel and carrier frequency offset, with the main time synchronization algorithms found in the literature. Finally, the complexity of the VLSI implementation of this proposal is discussed.

Keywords: *Synchronization, preamble, OFDM, WLAN, IEEE 802.11a/g, VLSI implementation*

1 Introduction

Orthogonal Frequency Division Multiplexing (OFDM) is a multicarrier modulation technique that has been adopted in many digital communication standards. Each OFDM symbol is composed of *N* samples; these are generated performing an *N*-point inverse Fast Fourier transform (IFFT) on *N* complex data symbols. Before transmission, the OFDM symbol is cyclically extended (this extension is called cyclic prefix, CP). If the CP is at least as long as the maximum delay spread of the channel, the inter symbol interference (ISI) caused by time dispersive channels is avoided. OFDM systems are very sensitive to errors in time and frequency synchronization. Multipath wireless channels blur the boundary between consecutive OFDM symbols, so time synchronization consists in estimating where the fast Fourier transform (FFT) window must begin to cover only samples belonging to the same OFDM symbol so as to avoid ISI and inter-carrier interference.

In this paper we present a new time synchronization algorithm for preamble-based OFDM systems whose preamble has a repetitive structure in the time domain; examples of this are the WLAN standard IEEE 802.11 a/g [1] [2], the WMAN standard IEEE 802.16-2004 [3], or the future standard for wireless access in vehicular environments IEEE 802.11p, which is under development as an amendment to the IEEE 802.11. For example, in Fig. 1 the structure of the IEEE 802.11 a/g preamble is shown: it consists of 10 identical short symbols (SS) of 16 samples. This repetitive part is usually completed with several OFDM symbols designed for channel estimation.

A great number of methods for OFDM time synchronization have been proposed in the literature. Some of them exploit the periodic structure of the cyclic prefix in OFDM symbols such as [4]-[7]. They were originally designed for general OFDM systems and can be adapted to be used in preamble-based OFDM systems; nevertheless, a better performance can be obtained by using algorithms that take advantage of the known preamble structure. Other algorithms make use of repeated preambles [8]-[11] similar to the IEEE 802.11a/g preamble, so they can be easily adapted to this standard. Some methods employ training symbols specifically designed to improve the performance of the proposed algorithm, such as [12]-[15]; these solutions cannot be used in the synchronization of existing systems with a defined preamble. And finally, some time synchronization techniques have been specifically designed for the IEEE 802.11a/g standard, such as [16]-[23].

Among the algorithms that make use of a preamble, two main techniques can be distinguished. The first technique is based on the Maximum-Likelihood principle and the preamble structure. For example, in [18] the Generalized Akaike Information Criterion (GAIC) is used to jointly estimate the channel and establish timing synchronization. Although the achieved performance is good, it has an extremely high complexity. To solve this problem, in [17] another ML algorithm was presented by Yik-Chung *et al.*, which has similar performance but much lower complexity, due to the smaller observation length. This one was chosen for comparison with the algorithm proposed in this work. The second main technique for time synchronization consists in correlating the received signal with a delayed version of itself [8], [11], with the known transmitted preamble [9]; or a combination of both auto-correlation and cross-correlation [10], [16], [19], [22].

Among the algorithms that can be adapted to a standard preamble, Shi & Serpedin [11] achieves the best performance, so it was selected for comparison. It presents a robust frame and time synchronizer, which finds the peak of a certain correlation metric that is achieved by invoking maximum likelihood principles. It calculates several auto-correlations using only the repetitive short symbols of the preamble. Among the algorithms based on correlation specially designed for IEEE 802.11a/g, only the algorithms that perform both coarse and fine time estimation were analyzed. Manusani *et al*. [21] and Troya *et al*. [19] proposed an auto-correlation based on Schmidl & Cox [8] for coarse time estimation using the short symbols and another algorithm for fine time estimation using the symbols for channel estimation. Manusani *et al*. [21] performs a fine time estimation exploiting the conjugate property of the symbols for channel estimation, whereas Troya *et al*. [19] makes use of a simplified cross-correlation with the first 32 complex samples of

the symbols for channel estimation (carrier frequency offset (CFO) is estimated and compensated before doing this cross-correlation). The algorithms proposed by Chang & Kelly [16], Zhou & Saito [22] and Kim & Park [23] calculate a cross-correlation between the received signal and the transmitted short symbols for fine time synchronization. For coarse time estimation [16] performs a maximum-likelihood auto-correlation, [22] an auto-correlation based on [8], and [23] uses the same cross-correlation for coarse and fine time estimation. For all these three algorithms, time synchronization only employs the short symbols of the preamble and finishes before the symbols for channel estimation (LS).

The algorithm proposed in this paper also calculates a cross-correlation between the received signal and the transmitted short symbols for fine time synchronization such as Chang & Kelly [16], Zhou & Saito [22] and Kim & Park [23], but a new metric is proposed for coarse time estimation. As will be demonstrated later, the proposed algorithm achieves good performance for low signal to noise ratio (SNR) and has a low-cost VLSI implementation that would facilitate its integration in a multi-standard processor for OFDM communications [24].

This paper is organized as follows: Section 2 presents the system model used in the rest of the paper. Section 3 describes the proposed synchronization algorithm, whereas in Section 4, an analysis of the new time metric is given. Section 5 is devoted to the performance evaluation by means of simulations. Section 6 deals with the VLSI implementation. Finally, conclusions are given in Section 7.

2 System Model

The samples of an OFDM symbol generated by an IFFT in the transmitter are given by:

$$x(n) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X_k \, e^{j2\pi kn/N} \quad , \quad n = 1, 2, \ldots, N-1, \qquad (1)$$

where N is the number of subcarriers, X_k represents the data symbol transmitted on the k-th subcarrier and $x(n)$ are the OFDM symbol samples after the IFFT. As commented above, the transmitted symbol is periodically extended (cyclic prefix) appending, at the beginning, the last G samples of the symbol: $[x(N-G), \ldots, x(N-1), x(0), \ldots x(N-1)]$. At the receiver, the received signal from a multipath channel after being sampled can be represented as:

$$r(n) = s(n) \, e^{j2\pi \lambda n/N} + w(n) \,, \qquad (2)$$

where λ is the carrier frequency offset (CFO) normalized to the subcarrier spacing $1 / (N T_s)$, $w(n)$ represents the zero-mean complex additive white Gaussian noise (AWGN) and $s(n) = \sum_{l=0}^{L_h} h(l) \, x(n-l)$, where $h(l)$ is the impulse response of the multipath radio channel with L_h paths.

Fig. 1 presents an example of repetitive preamble (this structure will also be used by the future standard IEEE 802.11p for wireless access in vehicular environments) designed for time synchronization and channel estimation. It consists of ten identical short symbols (SS) of 16 samples and two identical long symbols (LS) of 64 samples preceded by a guard interval (GI) composed of 32 samples from last samples of the LS. The problem of time synchronization is to find a known sample as a reference. For example, once signal detection and automatic gain control (AGC) are completed at sample n_i (see Fig. 1), time synchronization can begin: this must find the starting-point of GI (n_{GI}); so, channel estimation can be correctly achieved using the long symbols and also a reference for the first OFDM symbol is obtained.

3 Proposed Algorithm

The purpose of the time synchronization algorithm is to find where the first sample of GI (n_{GI}) is, using the repetitive preamble or short symbols. To accomplish this objective, the proposed algorithm is divided in two parts: fine and coarse time synchronization. The fine time estimation is

based on a cross-correlation, as some of the algorithms found in the literature [16], [22], [23]. On the other hand, the coarse time estimation is performed using a new metric. The proposed algorithm makes use of both coarse and fine time synchronization subsystems to search iteratively the first sample of GI.

3.1 Fine Time Synchronization

Fine time synchronization tries to find the transition between two short symbols (sample n_0 in Fig. 1), once AGC and signal detection have been accomplished. This task is achieved using a cross-correlation between the received signal $r(n)$, beginning at sample n_i, and the repetitive segment of the preamble as a pilot data $SS(k)$:

$$n_0 = \arg \max_{0 \le n \le L-1} \left(\sum_{m=0}^{N_{reg}} \left| \sum_{k=0}^{L-1} r(n + Lm + k) \cdot SS^*(k) \right|^2 \right). \qquad (3)$$

In IEEE 802.11a/g the length of a SS is 16 samples, therefore, in this case parameter L would be set to 16. As a result of (3), a peak every L samples is obtained, whose position indicates the starting point of each SS. Taking the average of N_{reg} blocks reduces any shift of the peak position due to channel noise. Finally, the position of the first sample of GI is calculated as: $n_{GI} = L \cdot N_{reg} + n_0$.

A simplification was introduced in this algorithm to reduce its computational complexity: we quantized pilot samples from SS's to values {-1, 1} for the real and the imaginary parts, except those that are originally zero, which were maintained, giving $SS_q = \{1-j, -1-j, -1+j, 1+j, 1+j, -1+j, -1-j, 1-j, 1+j, -1+j, -1-j, -j, -1-j, -1+j, 1+j\}$ for the IEEE 802.11a/g preamble. This simplification avoided the use of multipliers and the cross-correlation could be efficiently implemented as a wired complex filter, as described in Section 6.

3.2 Coarse Time Synchronization

The new metric for coarse time estimation is:

$$P(n) = \sum_{k=0}^{L-1} \left| r(n+k) - \frac{1}{M} \sum_{m=1}^{M} r(n + k - L \cdot m) \right|^2. \qquad (4)$$

This metric makes use of the periodicity of the preamble to calculate the squared difference between the present SS and a mean of M previous SS's. Therefore, in the ideal case with neither Gaussian noise nor carrier frequency offset, the metric $P(n)$ equals zero while samples from SS's are processed and grows abruptly when the first term in (4) includes samples from GI. Fig. 2 shows this behaviour for the IEEE 802.11a/g preamble with $L = 16$ and $M = 4$. Before setting a threshold to check if the abrupt change happens, it is necessary to normalize the new metric by the estimated local input signal power $R(n) = \sum_{k=-L}^{L-1} |x(n+k)|^2$ to avoid fluctuations due to changes in the received signal strength:

$$Q(n) = \frac{P(n)}{R(n)}. \qquad (5)$$

This metric is degraded in real scenarios where noise, multipath and carrier frequency offset are present. Section 4 deals with how these impairments affect the performance of the algorithm.

3.3 Iterative Search of n_{GI}

Once the OFDM signal has been detected, the synchronization algorithm begins (at sample n_i) the next iterative procedure to find where the transition between the last SS and the GI occurs:

1. Metric $Q(n)$ is calculated from n_i and during the length of the repetitive preamble (128 samples for the IEEE 802.11a/g preamble) to include the abrupt change that occurs at n_{GI}.
2. A coarse estimation of the beginning of GI is obtained (called n_{coarse} in Fig. 1):

$$n_{coarse} = \min\left\{ \arg_n \left[Q(n) > Thr \right] \right\}, \qquad (6)$$

   where Thr is a threshold.
3. A counter q is set to 1.
4. The received signal is cross-correlated to obtain n_0 using (3) with quantified SS_q and $N_{reg} = q - 1$.
5. If $n_0 + q \cdot L > n_{coarse}$, then $n_{GI} = n_0 + (q-1) \cdot L$; if not, then $q = q + 1$ and go to step 4.

Step 5) checks where the last correlation peak (n_0+L, n_0+2L, n_0+3L,…) before n_{coarse} is, since this will be the first sample of GI (n_{GI}). At each iteration, the number of terms (parameter N_{reg}) in the average made in (3) is incremented before n_0 is re-calculated, so the estimation of n_0 becomes more and more accurate.

4 Analysis of the New Timing Metric

As Fig. 2 shows, the proposed metric has low values during the periodic part of the preamble and suffers an abrupt change towards its maximum value when this part finishes. So, the selected threshold must be high enough to ensure that there are no false alarms (FA) when the periodic part of the preamble is being processed. On the other hand, when the repetitive section of the preamble has finished, the selected threshold must be low enough to minimize the number of detection failures (DF). Therefore, it is necessary to study the behaviour of the proposed metric $Q(n)$ to set the threshold to be employed in (6).

We are interested in the statistics of (5); as this is a ratio of two estimators, we can make use of the results in [25], where the mean and variance of a ratio of estimators are approximated by means of the first-order terms of a Taylor series expansion:

$$E\{Q(n)\} = \frac{E\{P(n)\}}{E\{R(n)\}}, \qquad (7)$$

$$\mathrm{var}\{Q(n)\} = \frac{\mathrm{var}\{P(n)\} + (E\{Q(n)\})^2 \,\mathrm{var}\{R(n)\} - 2E\{Q(n)\}\mathrm{cov}\{P(n), R(n)\}}{(E\{R(n)\})^2}. \qquad (8)$$

The mean and variance must be calculated during the repetitive part of the preamble to determine the probability of false alarm, and during the beginning of the GI to determine the probability of detection failure. The following subsections are devoted to analyse the performance of the proposed metric and to discuss the selection of the threshold.

4.1 Timing Metric during the Repetitive Part of the Preamble

We make use of R_{SS} and P_{SS} to denote the value of the estimated power and the un-normalized metric, respectively, during the short symbols. As shown in Appendix A, both mean and variance are constant during the repetitive preamble and are given by:

$$E\{R_{SS}\} = 2\varepsilon_{SS} + 2L\,\sigma_w^2, \quad (9)$$

$$\mathrm{var}\{R_{SS}\} = 4\varepsilon_{SS}\,\sigma_w^2 + 2L\,\sigma_w^4, \qquad (10)$$

$$E\{P_{SS}\} = L\frac{M+1}{M}\sigma_w^2, \qquad (11)$$

$$\mathrm{var}\{P_{SS}\} = L\left(\frac{M+1}{M}\right)^2\sigma_w^4, \qquad (12)$$

$$\mathrm{cov}\{P_{SS}, R_{SS}\} = L\sigma_w^4, \qquad (13)$$

where σ_w^2 is the power of the additive white Gaussian noise present in the channel, L is the length of each short symbol, M is the number of SS's used in (4), and ε_{SS} is the measured energy of a L-length period from the received preamble. This energy is constant during the repetitive preamble since it is measured during $2L$ samples and the period is L samples length; therefore, we can also express the energy of a SS as $\varepsilon_{SS} = L\sigma_{SS}^2$, where σ_{SS}^2 is the power of the received preamble.

If we define the signal to noise ratio as $\rho = \sigma_{SS}^2 / \sigma_w^2$, then (7) and (8) become:

$$E\{Q_{SS}\} = \frac{(M+1)/M}{2(\rho+1)}, \qquad (14)$$

$$\mathrm{var}\{Q_{SS}\} = \left(\frac{(M+1)/M}{2(\rho+1)\sqrt{L}}\right)^2\left(1+\frac{2\rho+1}{2(\rho+1)^2}-\frac{2M}{(M+1)(\rho+1)}\right). \qquad (15)$$

Fig. 3 shows how our theoretical approach fits the actual values of both estimators in AWGN channel for the IEEE 802.11a/g preamble with $L = 16$ and $M = 4$. It must be noted that these results are also valid for multipath channels because the periodicity of the preamble is maintained when it is convolved with the channel impulse response.

As commented before, the threshold value must be chosen to avoid false alarms during the repetitive preamble. Given that the numerator of metric Q can be characterized as a central chi-squared random variable of $2L$ degrees scaled by the signal power, we can follow a similar reasoning to [9] and [13], and approximate metric Q by a Gaussian random variable with mean (14) and variance (15). In this case, the false alarm probability is given by:

$$\Pr(Q_{SS} > Thr) \approx \frac{1}{2}erfc\left(\frac{Thr - E\{Q_{SS}\}}{\sqrt{2\,\mathrm{var}\{Q_{SS}\}}}\right), \qquad (16)$$

where $erfc(x)$ is the complementary error function. As an example, at the bottom of Fig. 4 the continuous line with circles represents the mean of Q_{SS} for the IEEE 802.11a/g preamble at different values of SNR, the dashed line with circles also shows a standard deviation of 3.1 from the mean (this is equivalent to a probability of false alarm of 1×10^{-3}).

4.2 Timing Metric outside the Repetitive Part of the Preamble

In this section we evaluate the behaviour of the metric (5) when the repetitive preamble has finished and the metric is calculated using samples from the beginning of the GI (Q_{GI}): it has an abrupt change as is shown in Fig. 2, which is searched using the iterative procedure presented in Section 3.3. As (6) looks for the first sample higher than the threshold, we can focus on what happens at the first peak of the metric since, if this peak is lower than the threshold, the synchronization algorithm would give a detection failure. We saw in Fig. 2 that the first peak is reached at the first sample of GI $n = n_{GI}$; so, the behaviour of the metric in this point is analysed next.

In the numerator of the metric, calculated in (4), the signal component does not cancel out anymore. Now we have $u(n) = r_{GI}(n) - \left(\sum_{m=1}^{M} r_{SS}(n - L{\cdot}m)\right)\Big/M$, as the samples inside the sum

come from the short symbols this gives: $u(n) = r_{GI}(n) - r_{SS}(n)$. The noise component has the same form as before: $w'(n) = w(n) - \left(\sum_{m=1}^{M} w(n - L{\cdot}m)\right)\Big/ M$. With these signals, in Appendix B it is shown that:

$$E\{P_{GI}\} = \varepsilon_z + L\left(\frac{M+1}{M}\right)\sigma_w^2, \qquad (17)$$

$$\mathrm{var}\{P_{GI}\} = 2\varepsilon_z\left(\frac{M+1}{M}\right)\sigma_w^2 + L\left(\frac{M+1}{M}\right)^2 \sigma_w^4, \qquad (18)$$

$$E\{R_{GI}\} = \varepsilon_{SS} + \varepsilon_{GI} + 2L\,\sigma_w^2, \qquad (19)$$

$$\mathrm{var}\{R_{GI}\} = 2\,(\varepsilon_{SS} + \varepsilon_{GI})\;\sigma_w^2 + 2L\,\sigma_w^4, \qquad (20)$$

$$\mathrm{cov}\{P_{GI}, R_{GI}\} = L\sigma_w^4 + 2\sigma_w^2(\varepsilon_{GI} - C), \qquad (21)$$

where R_{GI} denotes the value of the estimated power and P_{GI} denotes the un-normalized metric at sample n_{GI}, $\varepsilon_z = \varepsilon_{GI} + \varepsilon_{SS} - 2C$, ε_{SS} was defined previously, ε_{GI} is the energy of the first L samples from the GI and $C = \sum_{k=0}^{L-1} \mathrm{Re}\{s_{GI}(k)\, s_{SS}^*(k)\}$ comes from the cross terms between GI and SS. Again, Fig. 3 presents the comparison between the theoretical estimators and those obtained by simulation, which proves that our analytical approach is valid.

With these results, the probability of detection failure $\Pr(Q_{GI} < Thr)$ can be approximated in a similar way to the probability of false alarm. However, in real scenarios, only an exact solution can be obtained for the AWGN channel, because the calculation of C needs the knowledge of the channel impulse response that is not available until time and frequency synchronization have finished. Fig. 4 shows the results obtained with our analytical approach: in solid line with squares the mean of the estimator at the first sample of the GI for the AWGN channel (a standard deviation of 3.1 from the mean is shown using the dashed line with squares, this is equivalent to a probability of detection failure of $1{\times}10^{-3}$). This figure also shows the mean of the estimator for two multipath channels: BRAN channel models A and C [26]; these models are widely employed to measure the performance of WLAN systems in indoor scenarios. Channel model A represents a typical office environment with a root mean square (rms) delay spread of 50 ns, whereas channel model C represents an indoor open space environment with an rms delay spread of 150 ns. It can be seen in both cases that the mean of the estimator decreases due to a higher correlation between the received GI and SS caused by the multipath channel; this correlation increases with the rms delay spread of the channel. Therefore, for a given threshold the probability of detection failure estimated for AWGN channels is lower than the one obtained in a multipath channel.

4.3 Threshold Selection

The results obtained above can be employed to choose the threshold to be used in (6). If we set a threshold of 0.4 for SNR higher than 3 dB, we can see in Fig. 4 that this value is far enough from $E\{Q_{SS}\}$ and $E\{Q_{GI}\}$ to assure that both, the probability of false alarm and detection failure are lower than $1{\times}10^{-3}$ for the AWGN channel. For multipath channels the probability of false alarm is maintained, whereas the probability of detection failure worsens, although it would be still close to $1{\times}10^{-3}$ since the chosen threshold is lower than the limit estimated for the AWGN channel. So, a threshold of 0.4 can be a practical solution to simplify the implementation of the synchronization algorithm. For a lower SNR, the threshold would be chosen in a compromise between the probability of false alarm and the probability of detection failure. For example, in Section 5, where the evaluation of this algorithm is presented, we employed the threshold that makes both probabilities of false alarm and detection failure equal.

4.4 Analysis of the Carrier Frequency Offset Effects

In this section we evaluate how the presence of a residual CFO affects the time metric presented above. The signal model that includes the CFO was defined as $r(n) = s(n)\, e^{j2\pi\lambda n/N} + w(n)$. Looking at (9) and (19) we see that the mean of $R(n) = \sum_{k=-L}^{L-1} \left| x(n+k) \right|^2$ depends on the energy of the preamble and on the noise. As the CFO only affects the signal term and $\left| s(n)\, e^{j2\pi\lambda n/N} \right|^2 = \left| s(n) \right|^2 \left| e^{j2\pi\lambda n/N} \right|^2 = \left| s(n) \right|^2$, we can deduce that (9) and (19) are still valid when CFO is present.

In Annex C it is shown how the CFO must be taken into account when the mean of (4) is evaluated. For the repetitive preamble the mean is given by:

$$E\{P_{SS}\} = \gamma(\lambda, L, M)\ \varepsilon_{SS} + L\frac{M+1}{M}\sigma_w^2, \qquad (22)$$

where $\gamma(\omega, L, M)$ is the CFO factor :

$$\gamma(\lambda, L, M) = \left| 1 - \frac{1}{M}\frac{\sin(M\omega L/2)}{\sin(\omega L/2)} e^{-j\omega L\left(1+\frac{M}{2}\right)} \right|^2, \qquad (23)$$

and $\omega = 2\pi\lambda/N$. If the CFO is 0 kHz then $\gamma(\omega, L, M) = 0$ and (22) equals (11). Now, the mean and the variance of the time metric are:

$$E\{Q_{SS}\} = \frac{\gamma\rho + (M+1)/M}{2(\rho+1)}, \qquad (24)$$

$$\mathrm{var}\{Q_{SS}\} = \left(\frac{\gamma\rho + (M+1)/M}{2(\rho+1)\sqrt{L}} \right)^2 \left(1 + \frac{2\rho+1}{2(\rho+1)^2} - \frac{2M}{(M+1)(\rho+1)} \right). \qquad (25)$$

These results mean that the CFO causes a growth in the value of the time metric during the repetitive part of the preamble, and then the probability of false alarms also grows. Fig. 5 presents how the mean of time metric and its 3.1 standard deviation changes with a CFO equal to 0.1, 0.5 and 0.8 of the subcarrier spacing (the highest value in IEEE 802.11a/g is a CFO of 0.73).

On the other hand, when we evaluate the metric at the first sample of the guard interval the results given in Section 4.2 change due to the energy of the signal term that is given by (see Appendix C):

$$\varepsilon_{\hat{z}} = \varepsilon_{GI} + \kappa^2(\omega, M, L)\,\varepsilon_{SS} - 2\kappa(\omega, M, L)\,C, \qquad (26)$$

where

$$\kappa(\omega, M, L) = \frac{1}{M}\frac{\sin(M\omega L/2)}{\sin(\omega L/2)}\, e^{-j\omega L\left(1+\frac{M}{2}\right)}, \qquad (27)$$

is another factor that takes into account the CFO. Then, the mean of (4) is:

$$E\{P_{GI}\} = \varepsilon_{\hat{z}} + L\left(\frac{M+1}{M}\right)\sigma_w^2. \qquad (28)$$

The next question is to evaluate the effect of these results on the threshold proposed in the previous section. As the performance of the time metric is degraded in the presence of high CFO, this must be estimated and compensated from the short symbols before applying this algorithm. In any OFDM system it is necessary to estimate the CFO and to compensate it before channel estimation and data detection. There are many solutions in the literature to perform this task; a well-known solution is the one proposed in [8] based on the autocorrelation of the preamble. We evaluated the performance of this method using the repetitive part of the IEEE 802.11a/g standard

preamble and we obtained a residual CFO lower than 30 kHz (with a subcarrier spacing of 312.5 kHz the residual CFO is lower than 0.1 of the subcarrier spacing) when an average length of 64 samples was employed. This residual CFO can be reduced if more samples are used in the average. When we introduce this residual CFO in (24) and (25) we observe that $E\{Q_{SS}\}$ hardly changes and the threshold of 0.4 still gives a probability of false alarm lower than 1×10^{-3} for SNR higher than 3 dB. The probability of detection failure $\Pr(Q_{GI} < Thr)$ does not worsen with the presence of a CFO lower than 0.1 of the subcarrier spacing, since factor $\kappa(\omega, M, L)$ is nearly 1 at this CFO, as can be seen in Fig. 6. Therefore, we can consider that the threshold selection we gave in Section 4.3 is still valid.

5 Simulation Results

For performance comparison between our proposal and several synchronization algorithms, we employed the preamble and the physical layer parameters of the WLAN standard IEEE 802.11a/g. Next conditions were used: transmission of 10^4 test frames; BRAN multipath channel models A and C [26]; additive white Gaussian noise in a range of SNR values from 0 to 20 dB; and a CFO of 0.73 of the subcarrier spacing. It was assumed that AGC and signal detection are completed during the third SS as in [16] (n_i is randomly set following a uniform distribution in the range 32 to 47). Nevertheless, our simulations showed that the proposed algorithm only needs four SS's to work correctly, that is, AGC and signal detection can last until the sixth SS. For all the algorithms used in the comparison, their thresholds were obtained by simulation: the threshold that gave the best performance for each SNR was employed in each case; whereas for our algorithm we employed the method described in Section 4.3.

The effect of time synchronization errors on the performance was studied by using a physical layer simulator of the IEEE 802.11a/g standard. We performed several simulations which showed us that, when the estimated n_{GI} deviated between −4 and 0 samples from the ideal initial sample of the GI, the measured bit error rate of the OFDM receiver worked without performance loss using a constellation order of 64-QAM in channels with an rms delay spread up to 100 ns, or using a constellation order of 16-QAM in channels with an rms delay spread up to 150 ns. Therefore, in this paper a frame is considered to be correctly synchronized if the estimated initial sample falls inside this window of 5 samples.

The algorithms selected for comparison were: Chang & Kelly [16], Zhou & Saito [22], Kim & Park [23], Manusani *et al.* [21], Yik-Chung *et al.* [17], Troya *et al.* [19] and Shi & Serpedin [11]. As has been said before, among these algorithms only Shi & Serpedin needs to be adapted to the IEEE 802.11a/g standard. Then, its timing metric Q_n was obtained by calculating four auto-correlations with an average of 32 samples and different delays:

$$Q_n = \frac{F_n}{\sum_{i=1}^{5} \left|\mathbf{R}_{i,n}\right|^2}, \qquad (29)$$

where **¡Error! Marcador no definido.** $F_n = \left|F_{1,n}\right| + \left|F_{2,n}\right| + \left|F_{3,n}\right| + \left|F_{4,n}\right|$, $F_{1,n} = \mathbf{R}_{1,n}^H \mathbf{R}_{2,n} + \mathbf{R}_{2,n}^H \mathbf{R}_{3,n} + \mathbf{R}_{3,n}^H \mathbf{R}_{4,n} + \mathbf{R}_{4,n}^H \mathbf{R}_{5,n}$, $F_{2,n} = \mathbf{R}_{1,n}^H \mathbf{R}_{3,n} + \mathbf{R}_{2,n}^H \mathbf{R}_{4,n} + \mathbf{R}_{3,n}^H \mathbf{R}_{5,n}$, $F_{3,n} = \mathbf{R}_{1,n}^H \mathbf{R}_{4,n} + \mathbf{R}_{2,n}^H \mathbf{R}_{5,n}$, $F_{4,n} = \mathbf{R}_{1,n}^H \mathbf{R}_{5,n}$ and $\mathbf{R}_{i,n} = \left[x_{n+(i-1)\cdot 32}, ..., x_{n+i\cdot 32-1}\right]^T$ are sub-vectors of the received signal, with $i = 1,...,5$.

Among the algorithms chosen for comparison, Zhou & Saito, Troya *et al.* and the proposed algorithms include CFO estimation and compensation in order to achieve time synchronization. On the other hand, CFO has no effect in the rest of the algorithms. Fig. 7 and Fig. 8 show correct time synchronization probability for channel A and C, respectively, and a CFO of 0.73 of the subcarrier spacing. For SNR lower than 6 dB the proposed algorithm clearly outperforms the rest of the algorithms for both channels A and C. For low SNR (from 0 to 4 dB) and channel A, Shi & Serpedin's algorithm achieves quite good results: its success rate is only 4-5% below the proposed algorithm. Nevertheless, this algorithm does not work properly for channels with higher dispersion

such as channel C: its success rate is 10% below the proposed algorithm for SNR higher than 8dB. This is owing to the fact that Shi & Serpedin's algorithm only makes use of auto-correlations, which are less precise than cross-correlations. Chang & Kelly's algorithm achieves a correct time synchronization probability similar to the proposed algorithm, for SNR higher than 6dB and both channels A and C, but the results achieved for low SNR are really poor. This is due to the symbol synchronization stage. We observed that when the algorithm begins to work (instant n_i) at samples in the middle of a SS, the results are close to the 100% of success rate (see [16]), but when it begins to work at samples near the transition between two consecutive SS's a problem appears. In that case, the estimated GI starting point is often detected with an offset of ±16 samples, giving a synchronization failure as a result. This offset is due to the rounding to the lowest integer operation proposed in [16]. As commented above, for the evaluation of the algorithms n_i was randomly set in the range 32-47 (the third SS); this is the reason why it gives a lower performance than the one claimed in [16], whose results correspond to an n_i in the middle of a SS. The poor results given by Troya *et al.* algorithm for SNR higher than 8 dB (error probability higher than 4% at 8 dB SNR and channel A) are caused by the simplified cross-correlation based on XNOR 1-bit complex multipliers. The frame synchronization stage of the ML method (Yik-Chung *et al.* algorithm) does not give a good estimation of the channel order at low SNR. As this estimation is used by the symbol synchronization stage, the algorithm performance is penalized causing a high probability of detection error.

To sum up, the proposed time synchronization algorithm, which combines coarse synchronization with iterative fine synchronization, has better performance than the state-of-the-art for low SNR and multipath channel. Next, the reason of this is explained: the algorithm proposed for fine time synchronization based on a cross-correlation gets a peak every 16 samples, whose position indicates the starting point of each SS. The number of peaks depends on when the AGC finishes. To set a threshold and detect the last peak is quite difficult, because the magnitude of the peaks strongly depends on the SNR and on the multipath channel. The proposed coarse time estimation algorithm helps to find the last peak of the cross-correlation. Additionally, the time estimation given by the fine time synchronization is more and more accurate thanks to the iterative process. In fact, our simulations showed that when the fine time synchronization is used alone (we applied a similar iterative procedure to find the last peak), the correct time synchronization probability is drastically reduced: from 99.1% to 35.1% for 5 dB SNR and channel A; and from 99.9% to 74.8% for 15 dB SNR. In conclusion, the good performance of the proposed method is due to the combination of both fine and coarse time synchronization algorithms..

6 VLSI Implementation

In this section the hardware implementation of the proposed time synchronization algorithm is presented and compared to Shi & Serpedin's algorithm implementation, whose performance approaches the proposed algorithm. Table I summarizes the hardware cost of both algorithm, including the number of real multipliers, real adders and delay lines. Next, we explain how these results are obtained.

Fig. 9 shows the implementation of the fine time synchronization algorithm proposed in this paper. The cross-correlation in (3) is implemented as a wired complex filter: the figure shows the first five coefficients: $1+j$, $-1+j$, $-1-j$, $-j$ and $-1-j$; the signal input is $r(n) = a + jb$, so the first multiplier is replaced by the addition $-(b-a) + j(a+b)$, the second multiplier is replaced by $-(a+b) - j(b-a)$, and so on. In this way no multipliers are required and hardware cost is drastically reduced. Only 32 real adders and 30 real delay lines are used, whereas the traditional implementation as a complex filter of 16 coefficients requires 16 complex multipliers (4 real multipliers and 2 real adders each one), 15 complex adders (2 real adders each one) and 15 complex delay lines (2 real delay lines each one).

Once cross-correlation is calculated, the squared modulus is obtained and then the average in (3) is done. Finally, a circuit for calculating the position of the cross-correlation maximum is presented: actual and previous averaged samples are compared. If actual sample is greater, it is saved in register D1. After 16 cycles, the maximum value can be read from register D1 and the time offset

estimation can be read from register D2. The rest of the algorithm also has low hardware cost: the squared modulus is calculated with 2 real multipliers and 1 real adder; the average only requires 1 real adder and 16 real delay lines; and, finally, the maximum detection block is implemented with 2 real delay lines, a comparison circuit (combinational logic) and a counter (4 registers and combinational logic). The total resources required by the proposed fine time algorithm are: 2 real multipliers, 34 real adders and 46 real delay lines.

The implementation of the proposed coarse time synchronization algorithm can be seen in Fig. 10. On the one hand, the metric $P(n)$ in (4) needs 4 complex additions, 128 complex delay lines, 1 squared modulus operation (2 real products and 1 real addition) and 1 real moving average with a cost of 2 real additions and 16 real delay lines. On the other hand, the computational cost of $R(n)$ adds one squared modulus and one real moving average. The division in (5) is a costly operation [28] that can be avoided by multiplying $R(n)$ by the threshold with a cost of 1 real product. The hardware cost of this coarse time synchronization algorithm is: 5 real multipliers, 14 real adders, 160 delay lines and a comparison circuit (combinational logic).

To sum up, the complete time synchronization algorithm requires: 7 real multipliers, 48 real adders and 206 delay lines. It must be noted that although the new coarse time synchronization only works properly if the CFO has been removed from the repetitive preamble, in any OFDM system the CFO must be estimated for channel estimation and data demodulation, so the CFO estimation is not taken into account in the hardware cost of the time synchronization.

According to [11], Shi & Serpedin algorithm needs 12 complex products and four modulus operation. Each modulus operation can be calculated with a COordinate Rotation DIgital Computer (CORDIC) in vectoring mode, which can be implemented with (Nbits + 2)·3 additions [29], where Nbits is the number of bit precision needed at the output of the CORDIC. We have checked that 10 bits precision at the CORDIC output is enough to maintain the floating point performance, so each CORDIC operation would need 36 additions (4·36 = 144 additions are required). Nevertheless, we propose an implementation (Fig. 11) with only 4 complex products and no modulus operations (they are replaced by squared modulus operations: 2 real products and 1 real adder each one), these changes does not modify the performance of the algorithm. In this scheme there are 4 auto-correlations: the received signal is auto-correlated with an average of 32 samples and delays of 32, 64, 96 and 128 samples to obtain $F_{1,n}$, $F_{2,n}$, $F_{3,n}$ and $F_{4,n}$, respectively. The calculation of each auto-correlation needs a delay line of 32 complex samples, a complex conjugate operation, 1 complex multiplier and a moving average of length 32. The moving average can be implemented with a delay line of 32 complex samples, a delay line of 1 complex sample and 2 complex additions. Additionally, after the auto-correlation, 6 delay lines of 32 complex samples, 6 complex additions, 4 squared modulus operations (2 real products and 1 real adder each one) and 3 real additions are required to obtain F_n. In short, the algorithm requires 27 real multipliers, 48 real adders and 1032 real delay lines. These requirements suppose 285.7 % more multipliers and 400.9% more delay lines than the VLSI implementation of the proposed algorithm.

7 Conclusions

In this work a new time estimation algorithm for OFDM preamble-based systems is proposed. It is based on a cross-correlation with the transmitted repetitive part of the preamble for fine time synchronization and a new metric for coarse time estimation. Both algorithms finish before the part of the preamble dedicated to channel estimation, so the complete algorithm presents a low latency.

The new metric was analysed to evaluate its behaviour against noise, carrier frequency offset and multipath channels; these results were employed to set a criterion for choosing a threshold that guaranteed a low probability of error detection and of false alarm. The performance of the synchronization algorithm was evaluated for the IEEE 802.11a/g standard and compared to the performance of several algorithms found in the literature. Simulation results showed that, among

the studied algorithms, our proposal has the highest correct synchronization probability for low SNR and multipath channel. Finally, a low cost VLSI implementation of the proposed algorithm was presented: it only requires 7 real multipliers and 48 real adders. To summarize, the main advantages of the proposed algorithm are good performance for low SNR, low complexity, easy VLSI implementation and low latency.

Appendix A. Statitical Properties of the Timing Metric at the Repetitive Part of the Preamble

Each received sample is composed of an OFDM signal and a noise component $r(n) = s_{SS}(n) + w(n)$, where $s_{SS}(n)$ are samples of the received preamble during the repetitive part (short symbols) and $w(n)$ are samples of complex additive white Gaussian noise with variance σ_w^2 and zero mean. Using the common approach of calculating the mathematic expectation of $R_{SS}(n)$, the input signal power has a mean:

$$E\{R_{SS}(n)\} = \sum_{k=-L}^{L-1} \left| s_{SS}(n+k) \right|^2 + E\left\{ \sum_{k=-L}^{L-1} \left| w(n+k) \right|^2 \right\} = \varepsilon_s(n) + 2L\sigma_w^2,$$

where $\varepsilon_s(n)$ is the estimated local energy of the received preamble. As the short symbols are periodic with L, we can write: $\varepsilon_s(n) = 2\varepsilon_{SS}$, where ε_{SS} is the energy of a period of the received preamble. So, during the repetitive part of the preamble the mean of R is a constant due to the periodicity.

Another method to obtain this result is to take into account that $r(n)$ is formed by noise (characterized as a complex Gaussian random variable) added to the received preamble samples (a deterministic signal). When the sum of the squared modulus of $r(n)$ is obtained, we get a non-central chi-squared random variable represented by symbol χ_p^2, where p represents the degrees of freedom (number of real random Gaussian variables, in this case $p = 2L$). A central chi-squared random variable G obtained as $G = \sum_{i=1}^{p} (V_i/\sigma_v)^2$, where V_i are real random Gaussian variables with variance σ_v^2 and zero mean, has a mean of $p\sigma_v^2$ and a variance of $2p\sigma_v^4$ [27]. When the real Gaussian variables have mean μ_i, a non-central chi-squared random variable is generated with mean $(p+\lambda)\sigma_v^2$ and variance $2(p+2\lambda)\sigma_v^4$, with $\lambda = \sum_{i=1}^{p} (\mu_i/\sigma_v)^2$. In our case, the real and imaginary components of the Gaussian noise have variance $\sigma_w^2/2$ and their mean is the value of the real or imaginary part of the received preamble: $\mu_i = s_R(i)$ or $\mu_i = s_I(i)$. Then, if we calculate parameter λ we obtain:

$$\lambda = \sum_{i=-L}^{L-1} \left(\frac{s_R^2(i)}{\sigma_w^2/2} + \frac{s_I^2(i)}{\sigma_w^2/2} \right) = \frac{2\varepsilon_{SS}}{\sigma_w^2/2}.$$

So, the mean of $R(n)$ can be obtained as:

$$E\{R_{SS}\} = (p+\lambda)\frac{\sigma_w^2}{2} = \left(2\cdot(2L) + \frac{\varepsilon_s(n)}{\sigma_w^2/2} \right)\frac{\sigma_w^2}{2} = 2\varepsilon_{SS} + 2L\sigma_w^2.$$

By this method the variance can be easily obtained as:

$$\mathrm{var}\{R_{SS}\} = 4\varepsilon_{SS}\,\sigma_w^2 + 2L\sigma_w^4.$$

Next, we study the numerator when it is applied to samples from the short symbols, we call this $P_{SS}(n)$. In this case, M must be chosen in such a way that assures that: $s(n) = s_{SS}(n-L) = s_{SS}(n-2L) = \cdots = s_{SS}(n-M\cdot L)$, then the signal component inside the squared modulus cancels out and this leaves only noise terms: $w'(n) = w(n) - \left(\sum_{m=2}^{M+1} w(n - L\cdot m) \right) / M$, where $w'(n)$ is a white Gaussian random variable with variance $\sigma_{w'}^2 = (M+1)\sigma_w^2/M$. Using the same reasoning as before, the numerator is the sum of a squared modulus of Gaussian random variables and the obtained signal is a central chi-squared random variable with mean and variance:

$$E\{P_{SS}\} = 2L\frac{\sigma_{w'}^2}{2} = L\frac{M+1}{M}\sigma_w^2,$$

$$\text{var}\{P_{SS}\} = 2(2L)\left(\frac{\sigma_{w'}^2}{2}\right)^2 = L\left(\frac{M+1}{M}\right)^2\sigma_w^4.$$

Finally, we need to calculate the covariance between P_{SS} and R_{SS}. The covariance is:

$$\text{cov}\{P_{SS}, R_{SS}\} = E\{P_{SS}\ R_{SS}\} - E\{P_{SS}\}\,E\{R_{SS}\}.$$

After some mathematical manipulations, the first term is given by:

$$E\{P_{SS}\ R_{SS}\} = \varepsilon_{SS}L\sigma_w^2 + (2L^2 + L)\sigma_w^4 + \varepsilon_{SS}\ L\frac{\sigma_w^2}{M} + 2L^2\frac{\sigma_w^4}{M}.$$

Using the mean of *R* and *P* calculated previously we obtain:

$$\text{cov}\{P_{SS}, R_{SS}\} = L\sigma_w^4.$$

Appendix B. Statistical Properties of the Timing Metric after the Repetitive Part of the Preamble

At the first sample of the GI the subtraction in (4) can be expressed as $u(n) = r_{GI}(n) - r_{SS}(n)$. We can decompose it in noise and a combination of samples from the preamble: $u(n) = z(n) + w'(n)$, where the noise is as in Appendix A: $w'(n) = w(n) - \left(\sum_{m=1}^{M} w(n - L{\cdot}m)\right)\Big/M$; and the other term is: $z(n) = s_{GI}(n) - s_{SS}(n)$, where $s_{SS}(n)$ is the received repetitive preamble and $s_{GI}(n)$ is the received guard interval. To obtain $E\{P_{GI}\}$, that is, the mean of the numerator of the time metric at $n = n_{GI}$, we can use the same reasoning as in appendix A, when we estimated $E\{R_{SS}\}$, since signal $u(n)$ is composed of a sum of a deterministic signal plus a white Gaussian noise with variance $\sigma_{w'}^2 = \sigma_w^2(M + 1/M)$. Therefore we obtain:

$$E\{P_{GI}\} = \varepsilon_z + L\left(\frac{M+1}{M}\right)\sigma_w^2,$$

$$\mathrm{var}\{P_{GI}\} = 2\varepsilon_z\left(\frac{M+1}{M}\right)\sigma_w^2 + L\left(\frac{M+1}{M}\right)^2 \sigma_w^4,$$

where ε_z is:

$$\begin{aligned}\varepsilon_z &= \sum_{n=0}^{L-1}|z(n)|^2 = \sum_{n=0}^{L-1}|s_{GI}(n) - s_{SS}(n)|^2 \\ &= \sum_{n=0}^{L-1}|s_{GI}(n)|^2 + \sum_{n=0}^{L-1}|s_{SS}(n)|^2 - 2\sum_{n=0}^{L-1}\mathrm{Re}\{s_{GI}(n)\, s_{SS}^*(n)\} \\ &= \varepsilon_{GI} + \varepsilon_{SS} - 2C.\end{aligned}$$

And as the mean of $R(n)$ includes part of SS and part of GI becomes:

$$E\{R_{GI}\} = \varepsilon_{SS} + \varepsilon_{GI} + 2L\,\sigma_w^2,$$

and the variance:

$$\mathrm{var}\{R_{GI}\} = 2(\varepsilon_{SS} + \varepsilon_{GI})\,\sigma_w^2 + 2L\,\sigma_w^4.$$

The covariance between P_{GI} and R_{GI} can be obtained in a similar way to the one calculated in Appendix A giving:

$$\mathrm{cov}\{P_{GI}, R_{GI}\} = L\sigma_w^4 + 2\sigma_w^2(\varepsilon_{GI} - C).$$

Appendix C. Effect of the Carrier Frequency Offset in the Time Metric

The signal model that includes the CFO was defined as: $r(n) = s(n)\, e^{j\omega n} + w(n)$. As the CFO affects the signal term, during the repetitive part of the preamble the mean of (4) can be written as:

$$E\{P_{SS}(n)\} = \sum_{k=0}^{L-1} \left| s_{SS}(n+k)\, e^{j\omega(n+k)} - \frac{1}{M}\sum_{m=1}^{M} s_{SS}\left(n+k-L\cdot m\right) e^{j\omega(n+k-Lm)} \right|^2 + L\frac{M+1}{M}\sigma_w^2,$$

where the signal term does not cancel out as before. Now, the mean can be rewritten, due to the periodicity of $s_{SS}(n)$, as:

$$E\{P_{SS}(n)\} = \sum_{k=0}^{L-1} \left| s_{SS}(n+k)\, e^{j\omega(n+k)} \left(1 - \frac{1}{M}\sum_{m=1}^{M} e^{-j\omega Lm}\right) \right|^2 + L\frac{M+1}{M}\sigma_w^2.$$

The sum of complex exponentials can be expressed as:

$$\sum_{m=1}^{M} e^{-j\omega Lm} = \frac{\sin(M\omega L/2)}{\sin(\omega L/2)}\, e^{-j\omega L\left(1+\frac{M}{2}\right)};$$

then, the signal term is given by:

$$\sum_{k=0}^{L-1} \left| s_{SS}(n+k)\, e^{j\omega(n+k)} \left(1 - \frac{1}{M}\frac{\sin(M\omega L/2)}{\sin(\omega L/2)}\, e^{-j\omega L\left(1+\frac{M}{2}\right)}\right) \right|^2 =$$

$$= \sum_{k=0}^{L-1} \left| s_{SS}(n+k) \right|^2 \left| e^{j\omega(n+k)} \right|^2 \left| 1 - \frac{1}{M}\frac{\sin(M\omega L/2)}{\sin(\omega L/2)}\, e^{-j\omega L\left(1+\frac{M}{2}\right)} \right|^2 =$$

$$= \gamma(\omega, M, L)\ \varepsilon_{SS}\ ,$$

where ε_{SS} is the energy of a period from the received preamble and $\gamma(\omega, L, M)$ is a CFO factor that is given by

$$\gamma(\lambda, L, M) = \left| 1 - \frac{1}{M}\frac{\sin(M\omega L/2)}{\sin(\omega L/2)}\, e^{-j\omega L\left(1+\frac{M}{2}\right)} \right|^2.$$

Finally, the mean of the estimator is:

$$E\{P_{SS}\} = \gamma(\lambda, L, M)\ \varepsilon_{SS} + L\frac{M+1}{M}\sigma_w^2.$$

To evaluate the effect of the CFO on the mean of (4) at n_{GI}, we must calculate how the CFO affects the signal term:

$$\hat{z}(n) = s_{GI}(n)\, e^{j\omega n} - \frac{1}{M}\sum_{k=1}^{M} s_{SS}(n-kL)\, e^{j\omega(n-kL)}\ .$$

Again we consider that all the terms from the repetitive preamble $s_{SS}(n-kL)$ are equal, then

$$\hat{z}(n) = s_{GI}(n)\, e^{j\omega n} - s_{SS}(n)\, e^{j\omega n}\, \frac{1}{M}\sum_{k=1}^{M} e^{-j\omega kL}\ ,$$

where the sum of complex exponentials is:

$$\kappa(\omega, M, L) = \frac{1}{M}\sum_{m=1}^{M} e^{-j\omega Lk} = \frac{1}{M}\frac{\sin(M\omega L/2)}{\sin(\omega L/2)}\, e^{-j\omega L\left(1+\frac{M}{2}\right)}.$$

If we calculate the energy of the signal term:

$$\begin{aligned}
\varepsilon_{\hat{z}} &= \sum_{n=0}^{L-1}\left|\hat{z}(n)\right|^2 = \sum_{n=0}^{L-1}\left|s_{GI}(n)e^{j\omega n} - s_{SS}(n)e^{j\omega n}\kappa(\omega, M, L)\right|^2 \\
&= \sum_{n=0}^{L-1}\left|s_{GI}(n)e^{j\omega n}\right|^2 + \kappa^2(\omega, M, L)\sum_{n=0}^{L-1}\left|s_{SS}(n)e^{j\omega n}\right|^2 - 2\kappa(\omega, M, L)\sum_{n=0}^{L-1}\mathrm{Re}\left\{s_{GI}(n) - s_{SS}^{*}(n)\right\} \\
&= \varepsilon_{GI} + \kappa^2(\omega, M, L)\,\varepsilon_{SS} - 2\kappa(\omega, M, L)C.
\end{aligned}$$

Then, the mean of (4) at n_{GI} is:

$$E\{P_{GI}\} = \varepsilon_{\hat{z}} + L\left(\frac{M+1}{M}\right)\sigma_w^2.$$

Acknowledgements

This work was supported by the Spanish Ministerio de Educación y Ciencia under grants TEC2006-14204-C02-01 and TEC2008-06787.

References

1. IEEE 802.11a standard (1999). Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: high-speed physical layer in the 5 GHz band.
2. IEEE 802.11g standard (2003). Wireless LAN specifications: Further Higher Data Rate Extension in the 2.4 GHz Band.
3. IEEE 802.16-2004 (2004). Standard for local and metropolitan area networks, part 16: Air interface for fixed broadband wireless access systems.
4. Lee, D., & Cheun, K. Coarse symbol synchronization algorithms for OFDM systems in multipath channels. (2002). IEEE Communications Letters, 6 (10), 446-448.
5. Park, B., Cheon, H., Ko, E., Kang, C., & Hong, D. (2004). A blind OFDM synchronization algorithm based on cyclic correlation. IEEE Signal Processing Letters, 11 (2), 83-85.
6. Beek, J.J., Sandell M., & Börjesson, P.O. (1997). ML estimation of time and frequency offset in OFDM system. IEEE Trans. on Signal Processing, 45 (7), 1800-1805.
7. Ma, S., Pan, X., Yang, G., & Ng, T. (2009). Blind Symbol Synchronization Based on Cyclic Prefix for OFDM Systems. IEEE Trans. on Vehicular Technology, 58 (4), 1746-1751.
8. Schmidl, T., & Cox, D. (1997). Robust Frequency and Timing Synchronization for OFDM. IEEE Trans. on Communications, 45 (12), 1613–1621.
9. Coulson, A. J. (2001). Maximum likelihood synchronization for OFDM using a pilot symbol: Algorithms. IEEE J. Sel. Areas Communications, 19 (12), 2495-2503.
10. Tufvesson, F., Edfors, O., & M. Faulker (1999). Time and Frequency Synchronization for OFDM using PN-Sequence Preambles. Proc. Vehicular Technology Conference (VTC), 4, 2203-2207.
11. Kai Shi & Serpedin, E. (2004). Coarse Frame and Carrier Synchronization of OFDM Systems: A New Metric and Comparison. IEEE Trans. on Wireless Communications, 3 (4), 1271-1284.
12. Minn, H., Zeng, M., & Bhargava, V. K. (2000). On timing offset estimation for OFDM Systems., IEEE Communications Letters, vol. 4, no. 7, pp. 242-244, July 2000.
13. Minn, H., Bhargava, V. K., & Letaief, K. B. (2003). A robust timing and frequency synchronization for OFDM systems. IEEE Trans. on Wireless Communications, 2 (4), 822-839.
14. Minn, H., Bhargava, V. K., & Letaief, K. B. (2006). A combined timing and frequency synchronization and channel estimation for OFDM. IEEE Trans. on Communications, 54 (3), 416-422.
15. Park, B., Cheon, H., Ko, E., Kang, C., & Hong, D. (2003). A novel timing estimation method for OFDM systems. IEEE Communications Letters, 7 (5), 239-241.
16. Chang, S., & Kelley, B. (2003). Time synchronization for OFDM-based WLAN systems. Electronics Letters, 39 (13), 1024-1026.
17. Wu, Y., Yip, K., Ng, T. & Serpedin, E. (2005). Maximum-Likelihood symbol synchronization for IEEE 802.11a WLANs in unknown frequency-selective fading channels. IEEE Trans. on Wireless Communications, 4 (6), 2751-2763.
18. Larsson, E.G., Liu, G., Li, J., & Giannakis, G. B. (2001). Joint symbol timing and channel estimation for OFDM based WLANs. IEEE Communications Letters, 5 (8), 325-327.
19. Troya, A., Maharatna, K. , Krstic, M. , Grass, E., Jagdhold, U., & Kraemer, R. (2007). Efficient Inner Receiver Design for OFDM-based WLAN Systems: Algorithm and Architecture. IEEE Trans. on Wireless Communications, 6 (4), 1374-1385.
20. Yang, J., & Cheun, K. (2006). Improved Symbol Timing Synchronization in IEEE 802.11a/g Wireless LAN Systems in Multipath Channels. Int. Conf. on Consumer Electronics. doi: 10.1109/ICCE.2006.1598425.
21. Manusani, S. K., Hshetrimayum, R. S., & Bhattacharjee, R. (2006). Robust Time and Frequency Synchronization in OFDM based 802.11a WLAN Systems. Annual India Conference. doi: 10.1109/INDCON.2006.302775.

22. Zhou, L., & Saito, M. (2004). A New Symbol Timing Synchronization for OFDM Based WLANs under Multipath Fading Channels. 15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications. doi: 10.1109/PIMRC.2004.1373890.
23. Kim, T., & Park, S.-C. (2007). A New Symbol Timing and Frequency Synchronization Design for OFDM-Based WLAN Systems. 9th Conference on Advanced Communication Technology. doi:10.1109/ICACT.2007.358691.
24. Baek, J.H., Kim, S.D. & Sunwoo, M.H. (2008). SPOCS: Application Specific Signal Processor for OFDM Communication Systems. Journal of Signal Processing Systems, 53 (3), 383–397.
25. Van Kempen, G., & van Vliet, L. (2000). Mean and Variance of Ratio Estimators Used in Fluorescence Ratio Imaging. Cytometry, 39 (4), 300–305.
26. J. Melbo, J., & Schramm, P. (1998). Channel models for HIPERLAN/2 in different indoor scenarios. 3ERI085B, HIPERLAN/2 ETSI/BRAN contribution.
27. Abramowitz, M., & Stegun, I.A. (1972). Handbook of Mathematical Functions. Dover.
28. López-Martínez, F. J., del Castillo-Sánchez, E., Entrambasaguas, J. T., & Martos-Naya, E. (2010). Iterative-Gradient Based Complex Divider FPGA Core with Dynamic Configurability of Accuracy and Throughput. Journal of Signal Processing Systems. doi: 10.1007/s11265-010-0464-y.
29. Angarita, F., Canet, M.J., Sansaloni, T., Perez-Pascual, A., & Valls, J. (2008). Efficient mapping of CORDIC Algorithm for OFDM-based WLAN. Journal of Signal Processing Systems, 52 (2), 181-191.

Figure Captions

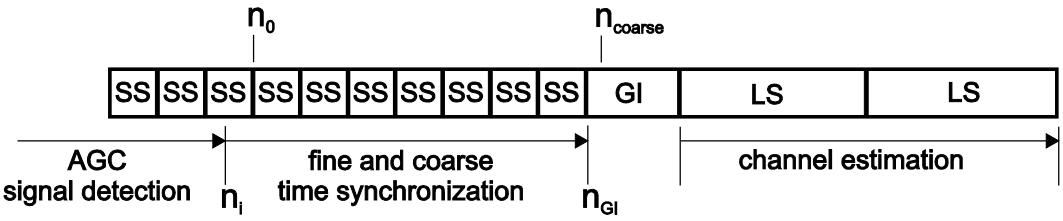

Fig. 1 IEEE 802.11a/g preamble

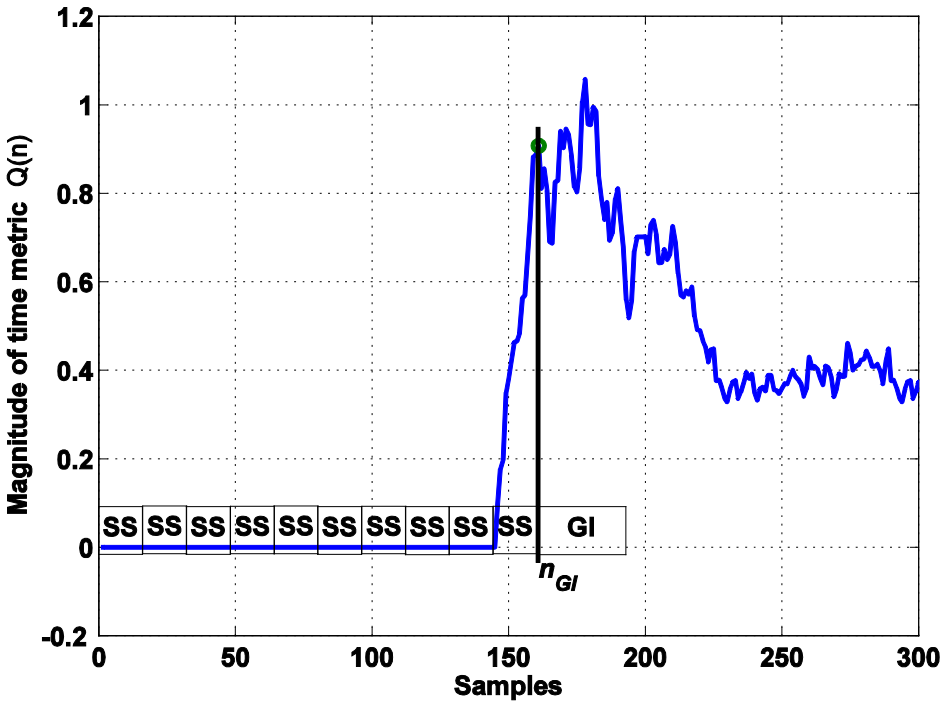

Fig. 2 Magnitude of the coarse time metric $Q(n)$ during the repetitive part of the preamble and at the beginning of the guard interval for the IEEE 802.11a/g preamble

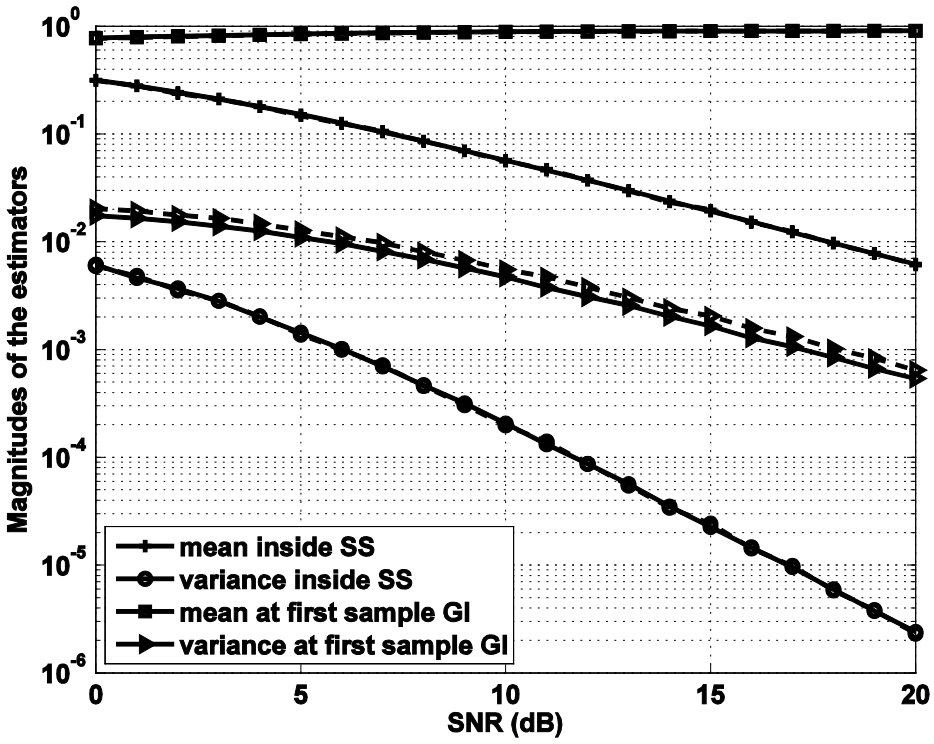

Fig. 3 Mean and variance of the new timing metric inside SS and at first sample of GI for AWGN channel. Continuous line: theory, discontinuous line: simulation

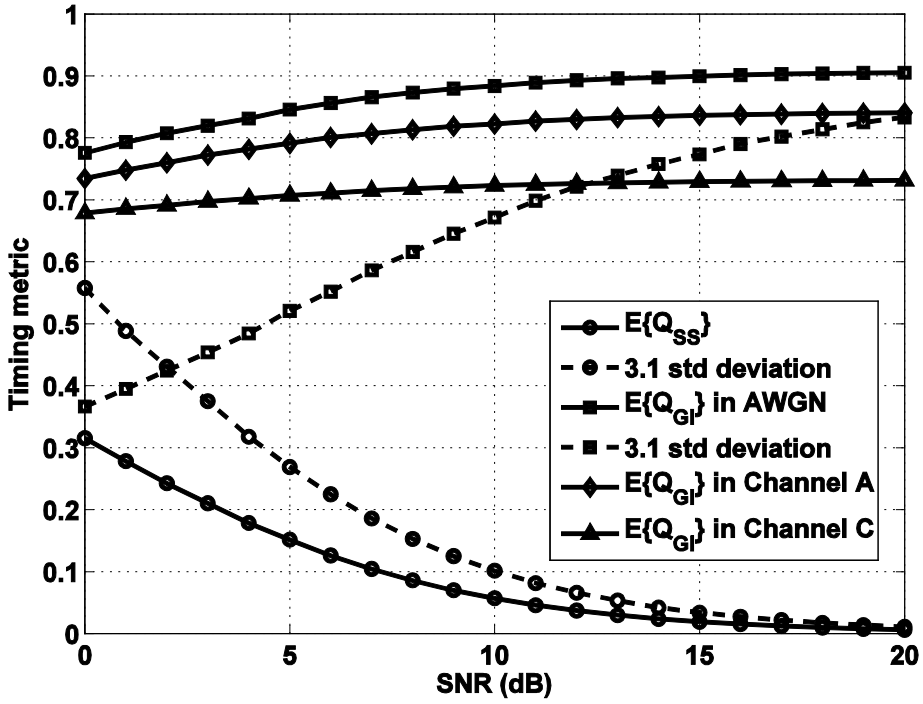

Fig. 4 Expected value of timing metric for the IEEE 802.11a/g preamble. Continuous lines: mean, dashed lines: 3.1 standard deviations from mean

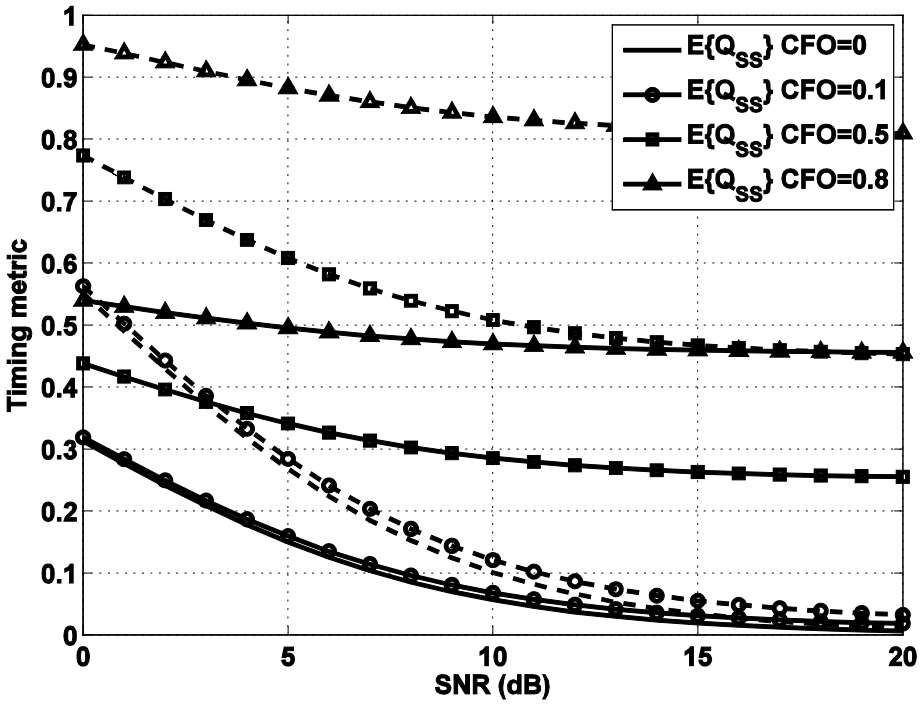

Fig. 5 Mean of the time metric (solid lines) and its 3.1 standard deviation (dashed lines) for the repetitive part of the preamble when CFO is present

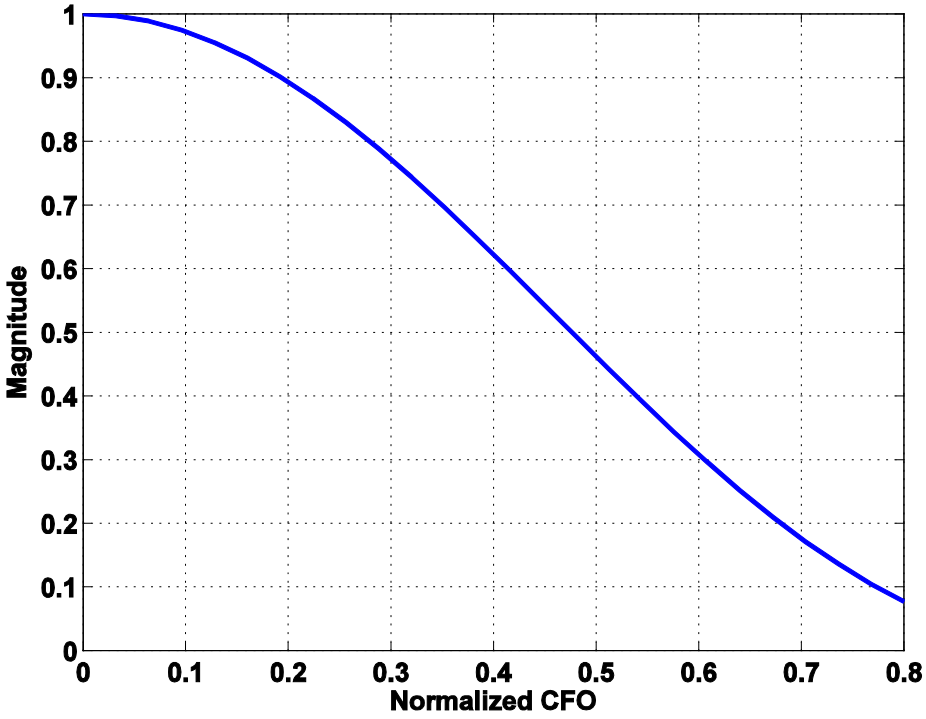

Fig. 6 Magnitude of $\kappa(\omega, M, L)$ against the CFO normalized by the subcarrier spacing for the IEEE 802.11a/g standard

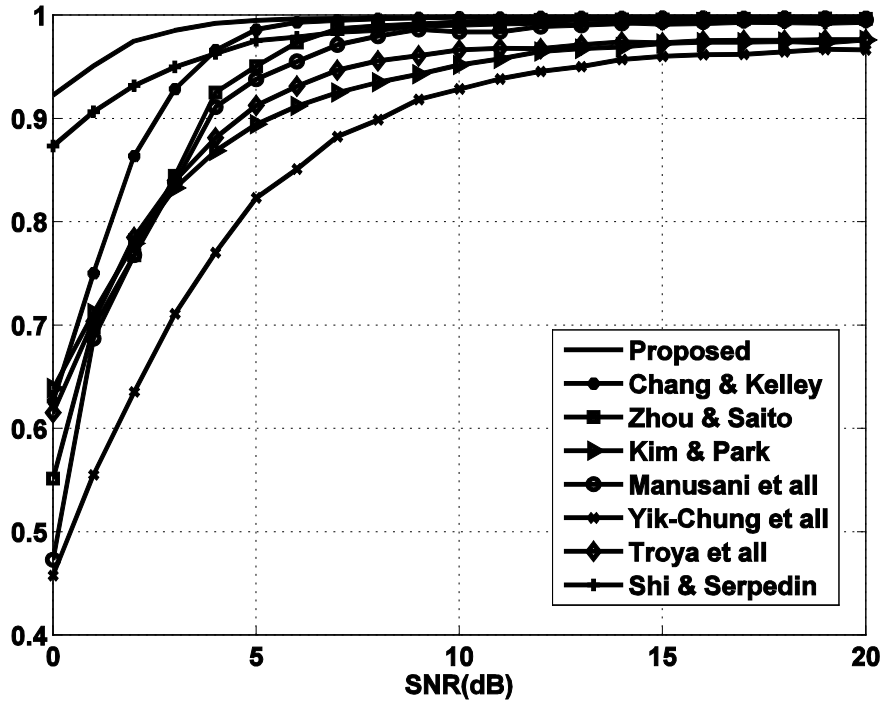

Fig. 7 Correct Time Synchronization Probability for all the evaluated schemes. Channel A. CFO 232kHz

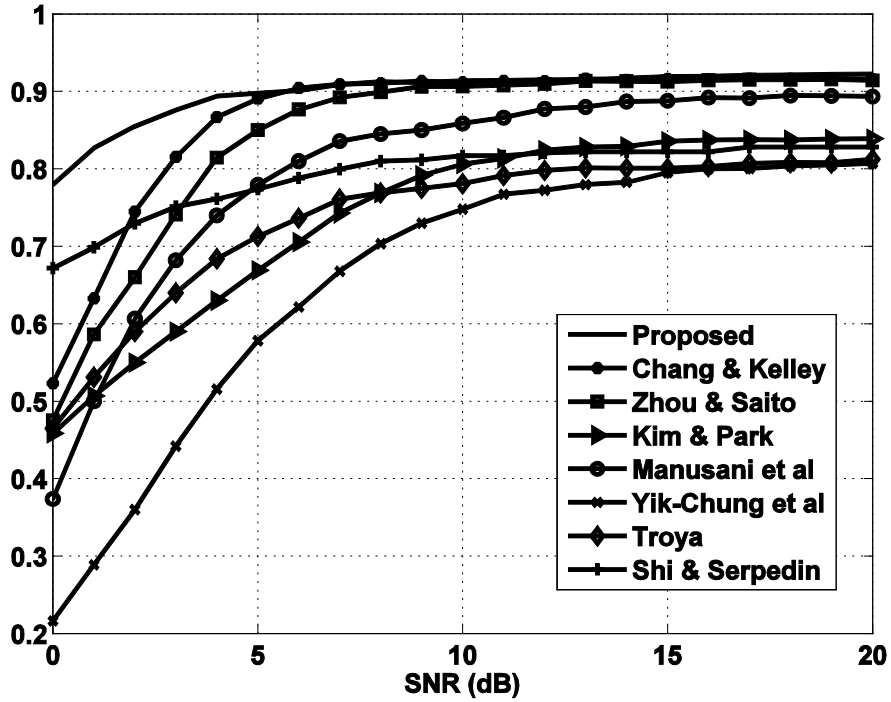

Fig. 8 Correct Time Synchronization Probability for all the evaluated schemes. Channel C. CFO 232kHz

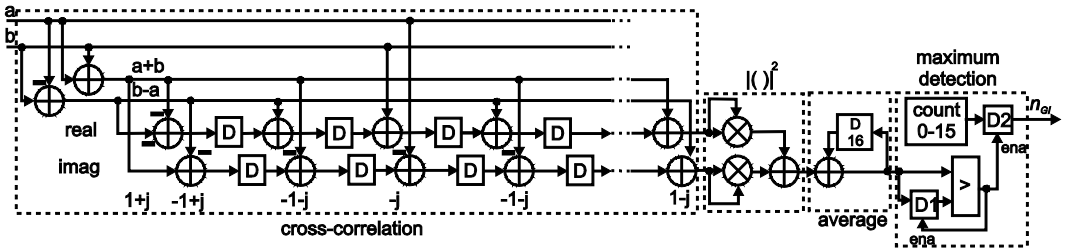

Fig. 9 VLSI implementation for the proposed fine time synchronization algorithm

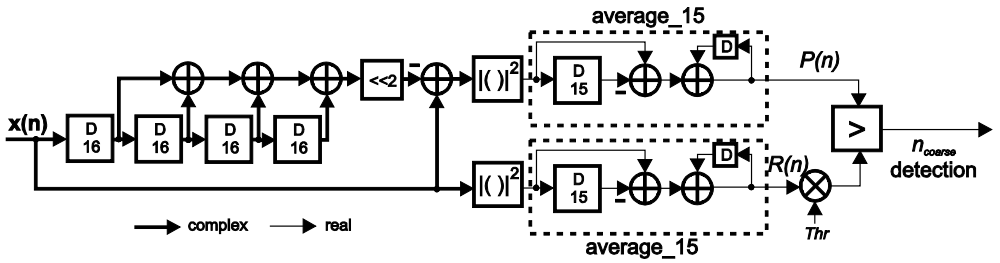

Fig. 10 VLSI implementation for the proposed coarse time synchronization algorithm

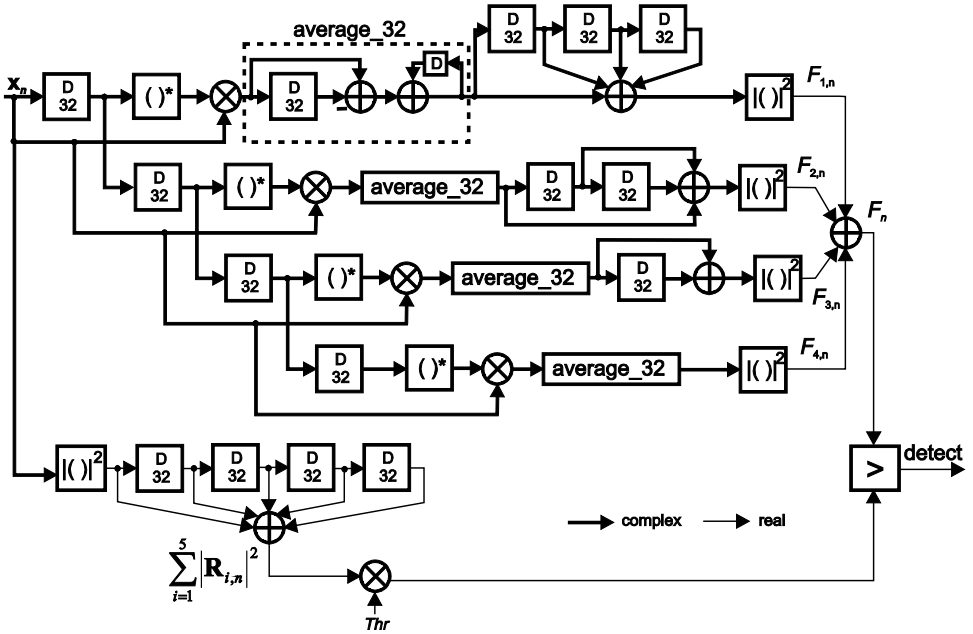

Fig. 11 VLSI implementation of Shi & Serpedin's algorithm

Table I. Hardware cost comparison

	Real multipliers	Real adders	Delay lines
Proposed algorithm	5	14	160
Shi&Sepedin	27	48	1023