Robust Signal Detection and Timing Synchronization Algorithms for OFDM Based Wireless Systems

Subhadeep Ghosh¹, Gordhan Singh², R.V. Raja Kumar³
¹,²,³Department of Electronics & Electrical Communication Engineering
Indian Institute of Technology, Kharagpur, India
Email: {subhadeepg_81@yahoo.co.in, gordhanrathore@yahoo.co.in, rkumar@ece.iitkgp.ernet.in}

Abstract

In this paper, we propose robust and efficient signal detection and symbol timing synchronization schemes applicable to preamble based wireless systems such as WLAN and WMAN-OFDM. The signal detection algorithm is based on detecting the increase in energy at the beginning of the preamble. For symbol timing synchronization, instead of using conventional normalized timing metric, a new algorithm utilizing both the autocorrelation and cross-correlation information is presented. The algorithm is primarily aimed at OFDM systems but can be extended to any broadband wireless access system. The simulations were carried out for both WLAN and WMAN standards.

Index terms - Orthogonal Frequency Division Multiplexing (OFDM), Correlation, training symbol, synchronization, wireless receiver.

1. Introduction

In any wireless system, which employs burst transmission, quick and efficient detection of packet arrival is necessary to start further operations like synchronization, carrier recovery, channel equalization etc. Symbol timing synchronization is important as it provides a timing reference for the rest of the burst. In OFDM systems symbol timing synchronization means declaring the end of the preamble to get an estimate of where the data symbol starts, so that further processing, like Cyclic Prefix (CP) removal and FFT can be carried on.

Several methods for preamble synchronization in both single carrier and multicarrier systems have been proposed over the years. For the OFDM systems the Schmid-Cox algorithm [1] is one of the most widely referred algorithm. It uses two (or more) repetitions of the same sequence in the preamble and performs the autocorrelation of the received sequence over two repetitions lengths. That algorithm then searches for the maximum value in the ACF characteristic. The timing reference flag is then decoded as the mean of the two sample numbers where ACF is 90% of the maximum. The metric they chose was normalized with energy and so the inherent variations in OFDM symbol energy have some adverse effect on the performance. Moreover the timing metric plateau inherent in this method causes larger variance of the timing estimate.

Nogami & Nagashima [5] suggested transmission of null symbols where nothing is transmitted for one symbol duration so that the drop in received energy can be detected to find the beginning of the frame. This extra overhead of a null symbol is avoided in our method. Moreover in burst transmission it is impossible to differentiate between a null symbol and the idle period between bursts. The correlation of the CP as suggested by Van de Beek [6] is futile, as it still requires detecting the beginning of the frame to obtain the location of CP. Moreover, as CP is usually affected by ISI, the result of estimation depends on channel estimation prior to synchronization. Classen [7] proposed using a longer GI and use of ISI free part of the GI for symbol timing synchronization. But it fails in some channel conditions.

In this work, the packet detection is derived from the fact that when a desired packet arrives, even at a low SNR, the energy of the received signal will be higher than that obtained prior to that. This increase in received energy is utilized for signal detection.

For the symbol timing synchronization scheme, we make use of the repetitive structure of the preamble by performing auto-correlation (ACF) of the preamble and cross-correlation (CCF) between the received signal and a previously stored pattern at the receiver. The correlation is performed such that the ACF has gradual rise at the start of preamble and a gradual fall at its end, while the CCF has sharp peaks at periodic intervals. We will detect the last peak of the CCF occurring at the end of the preamble with help of the falling edge of ACF there.

The organization of the paper is as follows: Section 2 gives the theoretical background to the OFDM system being considered. The preamble structure of WLAN and WMAN used for simulation are discussed in Section 3. In Section 4 development of proposed algorithms for signal detection and symbol timing synchronization along with their novelties are discussed. Performance results are given in Section 5. Finally, the conclusions are provided in Section 6.

2. System Description

OFDM is a special form of Multicarrier modulation and particularly suitable for mitigating the multipath effect during transmission over dispersive wireless channels. With the help of Guard Interval (GI) between OFDM symbols wireless systems can overcome limitations in data rate due to large multipath delay spreads. OFDM symbols are cyclically extended and the CP provides the GI between adjacent OFDM symbols. Due to this CP OFDM symbols are less sensitive to timing errors.
2.1 Transmitter

The samples of the \( l \)-th transmitted baseband OFDM symbols can be expressed as

\[
s_i(k) = \sum_{n=-N_{\text{SYM}}}^{N_{\text{SYM}} - 1} d_{l,n} * e^{j2\pi nf_i(k) + j2\pi n\Delta f} \left( \frac{k}{N_{\text{FFT}}} - \frac{2N_{\text{FFT}}}{N_{\text{FFT}}} \right)
\]

\[\text{for} \quad -N_{\text{CP}} < k < N_{\text{FFT}} - 1\] (1)

The OFDM signal is generated at baseband by taking IFFT of the constellation points of a suitable modulation scheme like QAM and QPSK where \( d_{l,n} \) is the constellation point mapped to \( k \)-th subcarrier of the \( l \)-th OFDM symbol, \( \Delta f \) is the subcarrier spacing, \( T_s \) is the sampling period, \( N_{\text{SYM}} \) defines the number of actual data samples, samples for Cyclic Prefix, number of FFT points and points in complete OFDM symbol respectively.

The digital baseband signal is converted to analog form and up-converted to a suitable RF and transmitter through the channel.

2.2 Channel

The channel model considered here is IEEE recommended modified SUI (Stanford University Interim) model for Fixed Broadband Wireless Access network. It can be modeled as an FIR filter, whose impulse response can be expressed as,

\[
h(k) = \sum_{i=0}^{P-1} a_i(k) \delta(k - \tau_i)
\]

With fixed tap delay \( \tau_i \) and time varying amplitude \( a_i(k) = \alpha p_i(k) \) in an \( P \) path model, where \( \alpha \) is the fixed attenuation factor according to the SUI model and \( p_i(k) \) is the Rayleigh noise factor. All the \( p_i \) are independent, though they are caused by the same Doppler frequency \( \nu \).

2.3 Receiver

The received RF signal is down converted directly to baseband, and upon digitization, can be expressed as

\[
r_i(k) = e^{j\phi} * e^{N_{\text{FFT}} * \sum_{i=0}^{P-1} a_i(k) \delta(k - \tau_i)} + n(k)
\]

There exists both carrier-phase offsets and symbol timing offset which are represented as the exponential term in the above expression. Here \( \phi \) is the arbitrary carrier phase factor, \( \nu \) is the carrier frequency offset normalized by the subcarrier spacing; \( n(k) \) is the complex Gaussian noise which zero mean and variance \( \sigma^2 \).

3. Preamble Structure

In any burst transmission based wireless communication system each frame is preceded by a preamble. The preamble is utilized for various tasks like AGC, signal detection, timing and frequency synchronization, ranging etc.

We used WLAN and WMAN preamble structures for simulating the proposed algorithm that are discussed below. The IEEE 802.11a standard for wireless LANs specifies identical preamble structure for both Uplink and Downlink. It consists of ten short training sequences and two long training sequences along with a GI between them. Each short training symbol is of 0.8\( \mu \)s duration and with 20Mps sampling rate has 16 samples. The long training sequences are of 3.2\( \mu \)s duration with 64 samples each. Thus the total preamble duration is 16\( \mu \)s including a GI of 1.6\( \mu \)s. Out of the ten short training sequences the first 7 may be reserved for signal detection AGC and diversity detection, whereas the rest may be kept for coarse frequency offset estimation and timing synchronization. The long training sequence is used for channel and fine frequency offset estimation.

<table>
<thead>
<tr>
<th>16X128+8 samples</th>
<th>16X2X32+8 samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>GI</td>
<td>T</td>
</tr>
</tbody>
</table>

Fig. 1: Preamble structure of the IEEE 802.11a WLAN

IEEE 802.16a (OFDM) utilizes different preamble structures for uplink and downlink. In the uplink the preamble consists of two repetitions of 128 samples preceded by a CP of length same as that of the CP in the traffic mode. This preamble is referred to as the short preamble. The first preamble in the downlink consists of a cyclic prefix followed by four repetitions of 64 samples and another CP followed by two repetitions of 128 samples. This preamble is referred to as the long preamble.

Fig. 2: UL Preamble structure of IEEE 802.16a WMAN

Fig. 3: DL Preamble structure of IEEE 802.16a WMAN
By default, the data sub-frame shall start with the long preamble but users may negotiate to use the short preamble. The choice of CP is user dependent.

4. The Proposed Methods

4.1 Signal Detection

Signal detection is necessary for starting preamble processing. The receiver normally remains in idle mode, to save power, until the arrival of a packet is declared. The proposed signal detection algorithm is based on relative increase in energy when a packet arrives. The energy calculation is continually performed over a sample window of $M$ samples and the ratio is calculated between energy values at the present sampling instant and $M$ samples earlier. When this ratio goes over a certain threshold, the arrival of a burst is declared.

Energy calculation is done as:

$$\text{Energy}(d) = \sum_{i=0}^{M-1} |r(i+d)|^2$$  \hspace{1cm} (4)

where $r(\cdot)$ is the received baseband signal, and $M$ is repetition window length.

And the ratio used for signal detection is calculated as:

$$\text{Signal-Detection-Ratio}(d) = \frac{\text{Energy}(d)}{\text{Energy}(d-M)}$$  \hspace{1cm} (5)

The Signal-Detection-Ratio characteristic is shown in Fig. 4.

4.2 Synchronization

Synchronization provides the timing reference for carrying out further operations like Cyclic Prefix removal and FFT at the receiver. The purpose can be served by detecting the end (last sample) of the preamble.

In a preamble consisting of repeated training sequences, correlation can be utilized to detect the last sample of the preamble. The signal detection algorithm discussed earlier will give a rough estimate of the start of the preamble. After that the ACF and CCF are calculated for each sample over the repetition window.

The auto-correlation is computed as

$$R_p(d) = \sum_{i=0}^{M-1} r^*(i+d) \times r(i+d+M)$$  \hspace{1cm} (6)

The cross-correlation function is performed over the received signal $r(\cdot)$ and the previously stored sequence $s(\cdot)$ at the receiver in a reversed manner as

$$R_{RS}(d) = \sum_{i=0}^{M-1} r^*(d-i) \times s(M-i)$$  \hspace{1cm} (7)

Thus we get a gradual rise, at the start of the preamble, and fall, at the end of it, in the autocorrelation characteristics. The algorithm is based on detecting the falling edge of the autocorrelation characteristics at the end of the preamble to define a window and then searching for the last peak in the cross-correlation characteristics in that window.

The synchronization scheme is based on correlation of the received preamble over a certain number of repetitions of the training sequence. The number of repetitions over which the correlation operation is performed decides the memory requirement for hardware implementation. If more of them are taken into account, the ACF as well as the CCF reach a higher peak value and will provide greater noise immunity. But averaging over a larger window requires more memory. Hence an optimum averaging window size has to be determined for each specific application.

Depending upon the number of repetitions used in the ACF and CCF calculations, equation (6) can be modified as,

$$A(d) = \sum_{k=0}^{L-1} \sum_{i=0}^{M-1} r^*(i+d+kM) \times r(i+d+(k+1)M)$$  \hspace{1cm} (8)

where, the fundamental autocorrelation $A(\cdot)$ is averaged over $L$ repetitions.

Cross-correlation $C(\cdot)$ can be averaged over $R$ repetitions to have suitable peaks:

$$C(d) = \sum_{k=0}^{R} R_{RS}(d+kM)$$  \hspace{1cm} (9)

where, $R$ is defined as $0 \leq R \leq L$

The Energy, ACF, and CCF characteristics calculated by the above given equations under noiseless condition are shown in Fig.5. Here an arbitrary preamble with 5 repetitive training sequences is chosen. The ACF is averaged over the entire preamble duration, whereas the CCF and energy are averaged over a single repetition. In the noiseless case, the onset of the falling edge of the ACF
and the last peak of the CCF match, but with AWGN and Rayleigh noise, as shown in Fig. 6, ACF will not be stable while the CCF will continue to have a peak right at the end of the preamble.

The most crucial part of the algorithm is detection of the falling edge of the ACF characteristics. The ACF characteristics rise up to the end of the preamble then fall steadily. We can detect this falling edge by observing the slope of the curve by continually searching for two sets of points where each set has two points on the ACF, which are separated by $Y$ samples. The two sets themselves are separated by $X$ samples. The difference between the two points of a set is calculated. The decision on the slope of the edge is taken by comparing the differences with energy-dependent thresholds for each set as:

\[
|A(i)| - |A(i + Y)| \geq \text{Th}_1 \quad (10) \\
|A(i + X)| - |A(i + X + Y)| \geq \text{Th}_2 \quad (11)
\]

To accommodate energy fluctuations, the threshold is no exactly the energy but scaled by $\beta_1$ and $\beta_2$ ($\beta_1, \beta_2 \leq 1$) for two sets of difference matching.

\[\text{Th}_1 = \beta_1 \ast \text{Energy}(i), \quad \text{Th}_2 = \beta_2 \ast \text{Energy}(i + X) \quad (12)\]

If the above two conditions in equations (10) and (11) are satisfied then the CCF peak is searched in a window. The end of the preamble is denoted by

$Flag = j$,

where $C(j)=\max(|C(i-shift+w)|)$, for $0 \leq w < W$,

$shift$ is the number of samples behind the $i$th one where the window of size $W$ starts.

To make the algorithm more robust, energy dependent threshold may be set for the CCF to avoid false detection in case the ACF has a steady falling edge at places other than the end of the preamble.

There are two issues to consider while deciding the values of $X$ and $Y$. First, if $X$ and $Y$ are large, the four points’ duration may exceed the ACF falling edge duration. Whereas smaller values of $Y$ (smaller than $M$) will either require calculation of energy over a smaller window that leads to more fluctuations in it, or scale down the calculated energy reducing noise immunity. On the other hand, a small $X$ means the two sets of points are close together making them redundant. Hence an optimum value of separation between points has to be decided.

Some considerations have to be taken into account while deciding the window size around the falling edge of the ACF over which the CCF peak is searched. If the window size is greater than the repetition length then two peaks will occur in the window and additional steps are required to choose the correct one. If it is very small then there is certain probability of not getting any CCF peak in that window. In our algorithm we have chosen the window to be slightly less than the repetition length.

4.3 Novelties in the algorithms

There are numerous other signal detection and synchronization schemes, all lacking in some aspect or the other. Our scheme for Signal Detection, unlike other proposed algorithms, does not look for the received signal itself to cross a threshold to declare the arrival of packet, rather it continually measures the ratio of the received energy to that a particular period of time ($M$ samples) earlier. This improves detection performance as it mitigates the effect of Rayleigh noise by excluding the need of any absolute threshold for the energy. The scheme proposed in this paper is unique in not normalizing the ACF by the energy and hence removing the restriction of identical window size for calculation of ACF and energy. It also prevents the fluctuations in energy due to AWGN from affecting the ACF. Furthermore, as the decision threshold is dependent on received energy, the miss detection probability due to larger Rayleigh multiplicative factor at lower SNRs is reduced. It is due to the fact that when an absolute threshold is set for a normalized metric, sometimes it may so happen that due to large Rayleigh effect the energy is higher but with low SNR the ACF may not be sufficient to push the metric above the threshold. Finally, the shifting of decision preference to CCF makes the algorithm more robust as the CCF is inherently less noise-prone than the ACF. It has another added benefit in the fact that the ACF characteristics has some dependence on the sequence chosen as the training
sequence as mentioned by [2], but the CCF has no such shortcoming, thus making the algorithm even more robust by making the performance independent of the training sequence.

5. Simulation Results

The performance evaluation of the algorithm is done by carrying out simulation studies for both WLAN and WMAN with parameters as specified in the respective standards.

The SU1 4 Channel Model used for the simulation specifies three taps where relative strengths of the tap with respect to first one are 0dB, -5dB and -10dB. The table below gives the parameters used in simulation.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>WLAN (IEEE 802.11a)</th>
<th>WMAN (IEEE 802.16a-OFDM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N_{FFT} )</td>
<td>64</td>
<td>256</td>
</tr>
<tr>
<td>( N_{USED} )</td>
<td>52</td>
<td>200</td>
</tr>
<tr>
<td>( N_{CP} )</td>
<td>16</td>
<td>64</td>
</tr>
<tr>
<td>BW</td>
<td>20 MHz</td>
<td>24 MHz (MMDS)</td>
</tr>
<tr>
<td>( F_s )</td>
<td>20 Msps</td>
<td>28 Msps</td>
</tr>
<tr>
<td>( \Delta f = T_s )</td>
<td>0.3125 MHz</td>
<td>62.5KHz</td>
</tr>
<tr>
<td>( T_{SYM} = T_{CP} + T_{FFT} )</td>
<td>4( \mu )s</td>
<td>20( \mu )s, 4( \mu )s+16( \mu )s</td>
</tr>
</tbody>
</table>

In our scheme, any flat plateau in the ACF was avoided by averaging over a smaller number of repetitions. Here, only last 3 of the 10 repetitions in 802.11a and all 5 repetitions in the short training sequences (including the CP) in the downlink short preamble of 802.16a are used.

Figures 7 and 8 illustrate the detection performance in the form of miss detection probability of the proposed methods in the case of the IEEE 802.11a and the IEEE 802.16a. The results show that proposed algorithms perform satisfactorily even at low SNRs. From Fig. 7, it can be shown that the miss detection probability is less than 1% even for an SNR that is as low as 3.5 dB for WLAN. In the case of the WMAN it is even less than 1 dB since the available window size is large. From the simulation curves it is quite clear that the algorithm mitigates the effect of Rayleigh noise to a great extent. Attention was also given to achieve highest efficiency at any SNR value.

6. Conclusions

In this work, a method has been proposed for efficient and robust symbol timing synchronization and signal detection in OFDM based wireless systems employing preambles. The algorithm is based on autocorrelation and cross-correlation among the repetitive structure of the preamble. Qualitative comparison with other proposed algorithms are given. All the considerations and intricacies that have to be taken into account while implementing this algorithm for a specific application are discussed.

References