Implementation of an IEEE802.11a synchronizer

Mattias Olsson
Division of Electronic Systems
Department of Electrical Engineering
Linköping University
Email: matol@isy.liu.se

***Abstract*— This paper applies some of the work that has been conducted regarding timing synchronization and carrier frequency offset estimation and correction in OFDM to the IEEE802.11a standard for wireless LAN. The algorithms are implemented as a floating-point model in Matlab using Simulink and as a synthesizable fixed-point model (FPGA) using Xilinx System Generator.**

I. Introduction

The popularity of the multicarrier modulation technique called Orthogonal Frequency Division Multiplexing (OFDM) has increased since its inclusion in standards for wireless LAN such as IEEE802.11a ([1]) and HiperLan/2. Advantages of OFDM are that it is bandwidth efficient and that it is rather insensitive to frequency selective fading and timing offset. One disadvantage though, is that OFDM is sensitive to carrier frequency offset (CFO), which will be seen in section 3.

This paper will focus on the effects of non-synchronized signals in IEEE802.11a and on how synchronization can be achieved and implemented.

II. An IEEE802.11a System Model

IEEE802.11a uses the previously introduced modulation method OFDM to transmit data. An OFDM symbol can be constructed as follows. First, the data to be transmitted is mapped to a complex value $X(k)$ in the frequency domain, representing a certain phase and amplitude. The complex values that can be chosen are organized in a rectangle with equal distance between them. This constellation is usually referred to as M-QAM (Quadrature Amplitude Modulation) where M is the number of points to choose from. In IEEE802.11a 52 subcarriers out of 64 are used for data and the remaining subcarriers are equal to zero. Four subcarriers of these 52 are used as pilot tones and can not be used to transmit arbitrary data. Second, the IDFT is calculated, normally using an IFFT algorithm, to get a complex time domain OFDM symbol

$$x(n) = \text{IFFT}\{X(k)\} = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X(k) e^{-\frac{j2\pi kn}{N}} \qquad (1)$$

where N is equal to 64 in IEEE802.11a.

To make OFDM more robust against multipath and timing offset, each symbol is extended with a guard interval or cyclic prefix (CP). Since, in the FFT-case, the multiplication-convolution relation is only valid when the convolution is a so called circular convolution the CP is constructed as

$$[x(-N_g)\; x(-N_g+1)\; ...\; x(N-2)\; x(N-1)] \qquad (2)$$

where N_g is equal to 16 in IEEE802.11a. In IEEE802.11a, the resulting time-domain signal is then multiplied by a window to reduce the problems with out-of-band frequencies.

Finally, the time-domain signals are D/A-converted, mixed with a carrier, filtered and transmitted through the air. In IEEE802.11a the sample rate is 20 Msamples/s.

In the receiver, the opposite operations are performed using A/D-conversion and DFT calculation. Since QAM uses coherent detection the receiver has to estimate the phase to be able to successfully recover the information that was sent. Coherent detection means that all the signal alternatives are known in the receiver. It also has to compensate for distortion caused by the channel.

Since IEEE802.11a transmits packets in a bursty manner synchronization is vital. To help the receiver accomplish synchronized reception a known data sequence, the preamble, is transmitted at the beginning of each packet, see Fig. 1. The first

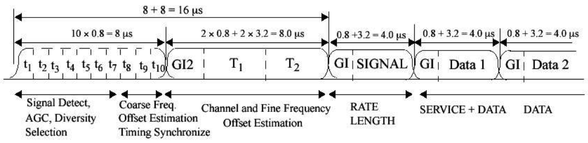

Fig. 1. *The IEEE802.11a preamble.*

half of the preamble consists of ten identical short symbols. Each short symbol consists of 16 samples. The second half of the preamble consists of two identical long symbols, each 64 samples long, preceded by a 32-sample CP. The symbols are designed so that the correlation between two subsequent samples is minimal. The preamble can, for example, be used for packet detection, frequency offset estimation and channel estimation.

In the IEEE802.11a system model described above, interleaving, scrambling and forward-error correction (FEC), etc., has been omitted since they do not affect the problem of synchronization.

III. The Effects of Non-Synchronization

In this section we will study what happens if the received signal is not synchronized in time and frequency. Sampling frequency offset and phase tracking will not be treated here.

A. Timing offset

The timing offset can be defined as the difference between the estimated timing instant and the correct timing instant.

If the timing offset τ_0 is within the CP it can be seen as if each complex OFDM symbol is multiplied by $e^{j\frac{2\pi k\tau_0}{NT_s}}$ where k is the subchannel. The multiplication causes a subchannel dependent rotation in the complex plane with the largest rotation on the edges of the frequency band [2]. See Fig. 2 for a picture of how the QPSK constellation is affected. As long as the timing offset is small enough this rotation can be estimated and corrected in the channel estimator [3] or in the phase tracker.

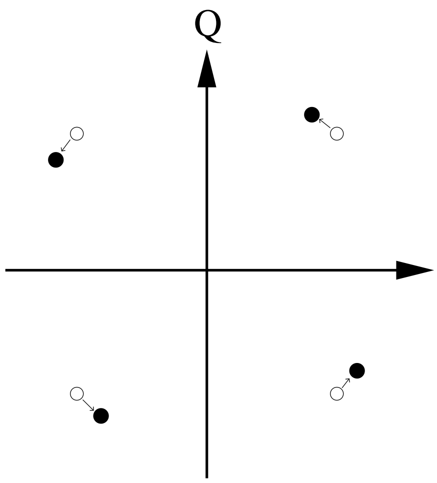

Fig. 2. *Phase rotation for QPSK.*

If the timing offset is larger than the CP minus the delay spread the subcarriers will lose their orthogonality. This is because the DFT is no longer calculated using the samples within a symbol or the CP, instead it is overlapping with either earlier or later symbols.

B. Carrier frequency offset

In the receiver the signal is shifted down to the baseband by mixing and low pass filtering. Because of differences between the oscillators in the transmitter and receiver, Doppler shift, etc., each path in the received signal is affected by a Carrier Frequency Offset (CFO) Δf. The received signal, affected by the CFO Δf, a multipath channel with L_p paths each with the attenuation factor $\alpha_l(t)$ and the white Gaussian noise $w(t)$, can be written as

$$r(t) = \sum_{l=0}^{L_p-1} \alpha_l(t)x(t-T_l(t))e^{j2\pi\Delta f t} + w(t). \qquad (3)$$

To see how the time and frequency offsets affect the received signal take (3), use a sample period of T_s seconds and assume that the channel is time invariant. Let $H_{i,k} = \sum_{l=0}^{L_p-1} \alpha_{i,l}e^{-j\frac{2\pi kT_l}{NT_s}}$ and (3) can be written as

$$r_{i,n} = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} H_{i,k}X_{i,k}e^{j\frac{2\pi n(k+\Delta f NT_s)}{N}} + w_n. \qquad (4)$$

where i is the symbol number. From (4) it can be seen that the effect of the CFO Δf is as if each sample n is multiplied by $e^{j2\pi n\Delta f T_s}$, causing a time varying rotation in the complex plane. The direction of the rotation depends on the sign of the CFO Δf. If the CFO is not completely corrected a slow rotation with time will occur. Since a coherent system has to calculate the phase continuously, for example by using the known pilot sub carriers, the rotation can be estimated and compensated for if it is small [4]. Unfortunately, the remaining CFO will also cause errors that can not be completely compensated for by simply rotating the phase in the frequency domain.

If the symbol number i is neglected the DFT of $r_{i,n}$ calculated at the receiver is equal to

$$R_k = \sum_{n=0}^{N-1} r_n e^{\frac{-j2\pi kn}{N}} = X_k H_{i,k}\Big(\frac{\sin(\pi\epsilon)}{N\sin(\frac{\pi\epsilon}{N})}\Big)e^{\frac{j\pi\epsilon(N-1)}{N}} + S_k + W_k, \qquad (5)$$

where the term S_k, which represents the inter-carrier interference (ICI), can be written as

$$S_k = \sum_{\substack{l=0\\ l\neq k}}^{N-1} X_l H_k\Big(\frac{\sin(\pi\epsilon)}{N\sin(\frac{\pi(l-k+\epsilon)}{N})}\Big)e^{\frac{j\pi(\epsilon(N-1)-l+k)}{N}}. \qquad (6)$$

$X_{i,k}$ is the desired signal and W_k is the additive white noise. It can be seen as if the frequency axis is shifted and the sampling of the spectrum is no longer performed at the center of the sub carriers, see Fig. 3. As can be seen from (5) this results in a decreased amplitude and rotated phase, but most important: it results in ICI.

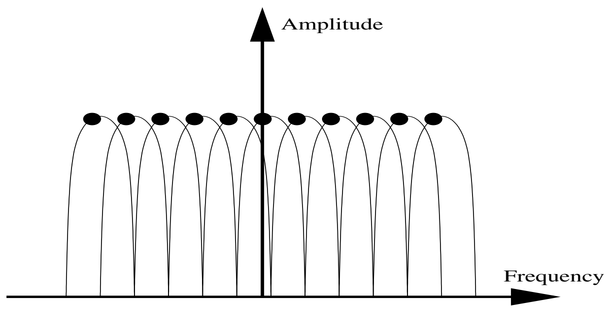

Fig. 3. *The OFDM spectrum with a frequency offset.*

An approximation of the effective SNR with a CFO can be calculated, see [5].

IV. CFO ESTIMATION

The algorithms described in the literature can roughly be divided into two groups, algorithms that are executed in the frequency domain and algorithms that are executed in the time domain.

One of the first articles regarding synchronization in the frequency domain is [6]. In [7] that algorithm is applied to IEEE802.11a.

In [8] time domain ML-estimation in a non frequency selective environment is treated. Channel information is included in the estimate in [9] and [10].

A. Estimation

In this paper we will consider an algorithm that is executed in the time domain. It is based on the algorithm presented in [8], but it is adopted to the IEEE802.11a situation. First, calculate the following correlation

$$\gamma = \sum_{k=0}^{N_g-1} r(k)r^*(k+N), \tag{7}$$

where N_g is the number of samples in the CP. An estimate of the CFO is then found by calculating

$$\hat{\epsilon} = -\frac{1}{2\pi}\arg\gamma \tag{8}$$

Since the CFO estimator depends on the angle of (7) it will be periodic and therefore the upper limit on the CFO that can be estimated is

$$|\Delta f| \leq \frac{1}{NT_s} = \Delta f_{\text{max}} \tag{9}$$

where N is the delay between the correlated samples and T_s is the sampling period. If the CFO is greater than Δf_{max} it will be impossible to determine the part of the CFO that consists of an integer number times the distance between the subcarriers.

The preamble is designed to aid the CFO estimation. Each short training symbol can be seen as a CP to the other short symbols, making it easy to calculate an average. The same is the case for the long training symbols.

For the short training symbols in the IEEE802.11a preamble the maximum detectable frequency error is 625 kHz and for the long symbols it is 156.25 kHz. The maximum allowed CFO according to the standard is 212 kHz which is within the range of the short symbol, but not within the range of the long symbol. Thus, as soon as the coarse estimate is calculated it can be used to start correcting for the CFO, before the long symbols are received.

The algorithm described above does not include information about the channel in the estimation process. When the packet starts we do not have a channel estimate and hence we cannot use it.

B. Tracking

In reality there will always exist a small residual frequency offset (RFO) even after the CFO has been corrected. This RFO will cause a rotating phase that can be modeled as [11]

$$\phi_m^{\text{RFO}}(n) = \phi_{m-1}^{\text{RFO}}(N-1) + 2\pi\Delta f(N_g+n), 0 \leq n \leq N-1, \tag{10}$$

with $\phi_0^{\text{RFO}}(n) = 2\pi\Delta f(N_g+n)$. The rotating phase can be used to track the CFO by measuring the phase rotation between two OFDM symbols.

C. Correction

The most natural way to compensate for CFO would be to simply feed back the estimated CFO to the oscillator in the receiver. The compensation can, however, be done digitally by multiplying each incoming sample by $e^{-j2\pi n\Delta f T_s}$. Note the minus sign which counteracts the rotation caused by the CFO.

V. Timing estimation

The timing estimation is usually divided into packet detection and some kind of fine timing estimation. In [4] an algorithm for packet detection is presented. It works as follows. Calculate the correlation for the short training symbols and the received energy continuously. Divide these two with each other and trigger when the result grows towards one. Ideally the energy and correlation will be equal when the short training symbols are received. Precautions have to be taken to prevent division with small energy value, otherwise error amplification will occur. In a practical implemementation the division can be avoided by using the following comparison instead:

$$\text{correlation} > \text{threshold} \cdot \text{energy} \tag{11}$$

A fine time estimate can later be found by correlating the incoming samples with the known long symbols and then try to find the maximizing instant. A rough estimate of when this maximum can be expected can be found using the timing from the packet detection. It can be shown that the complexity of this correlation can be drastically reduced by only correlating the sign bits of the known samples and the received samples. In an ideal situation in IEEE802.11a the correlation of the quantized long symbol gives 126 as the result. Knowing the maximum value allows us to choose a reasonable threshold level.

Due to the CP it is not critical if the timing estimate is a little too early as long as the multipath delay is small. On the other hand, if the timing estimate is too late, ICI will occur. In a well designed system the CP will be long enough to take care of the normal multipath delay and more. This margin can be used by systematically shifting the symbol timing point inside the CP with a few samples. In [4] a 4–6 sample shift is presented as a rule of thumb for an IEEE802.11a system.

VI. Implementation

In a packet-based system like IEEE802.11a it is important that the time- and frequency synchronization are performed quickly and within the preamble. This forces us to favor algorithms with short delay and that are easily implemented.

In Fig. 4 the structure of our CFO estimator and corrector is presented. At the start of the packet the incoming samples pass through the rotator unchanged and into the correlator. The output from the correlator is then transfered into the angle calculator which computes the argument of the complex signal. The mean angle is then calculated and finally a CFO estimate is found. This first estimate is then used to decrease the CFO before the new estimate is calculated using the long training symbols.

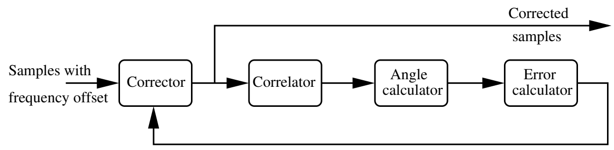

Fig. 4. *The structure of the frequency estimator and corrector.*

The synchronizer has been implemented as a floating point model using Matlab and Simulink to verify the operation. The CFO estimator and corrector was then implemented as a synthesizable (FPGA) fixed-point model using Xilinx System Generator. The simulations show acceptable performance in an AWGN (Additive White Gaussian Noise) channel. At an SNR of 25 dB the mean remaining CFO after the correction was 0.2% of the distance between two subcarriers. In a multipath environment an expected performance degradation was observed.

A. Corrector

The straightforward way to implement a CFO corrector is to use a complex multiplicator, a LUT (Look-Up-Table) and an accumumulator. The size of the LUT can be adapted to the desired resolution. An alternative way is to implement a vector rotator using the CORDIC algorithm. In [12] an architecture for implementing an angle rotator using the radix-4 CORDIC algorithm is presented.

B. Correlator

The most central part of the CFO estimator is the correlator. The structure of the correlator is shown in fig. 5.

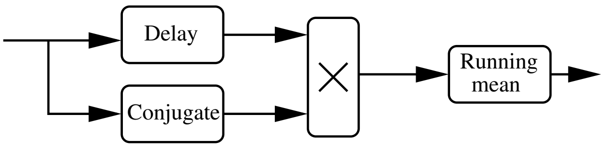

Fig. 5. *The structure of the correlator.*

The correlator delay is different for the short and long training symbols. To save space and implementation cost the correlator can be adopted to the different delays during run time. This can be done by using multiplexers and by designing the control signals so that zeroes are shifted in to clear the shift registers, see fig. 6.

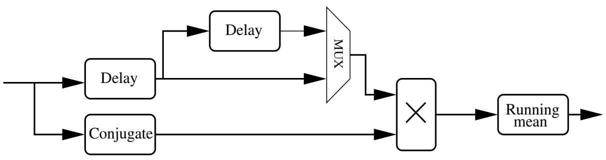

Fig. 6. *The structure of the adaptive correlator.*

The running mean device calculates the mean value continuously for a certain number of past samples. To reduce the number of additions that has to be performed during each sample the calculations are performed iteratively, see fig. 7.

C. Angle calculator

The argument of the complex correlation is computed by dividing the complex part with the real part and inserting the result into an angle calculator. The angle calculator can either be implemented as a LUT or as a Cordic processor.

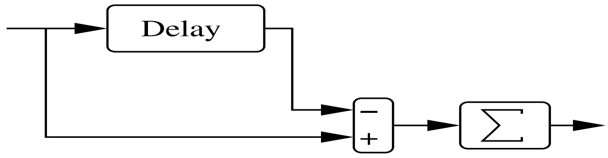

Fig. 7. *The structure of the running mean calculator.*

The angles from the angle calculator contain noise that can be decreased by averaging. This is done in a running mean calculator over 32 samples.

D. Power consideration

The synchronizer, as it has been proposed, requires many delay elements. To reduce the number of power consuming transitions the delay elements can be replaced with a dual-port memory and two pointers working in a circular fashion.

VII. Conclusions

The frequency offset estimation algorithm that is studied in this paper does not take the characteristics of the channel into account. One future goal could be to merge an algorithm that performs CFO estimation in the frequency domain with a channel estimation algorithm. The new algorithm could then be applied to the long training symbols in the preamble.

Ongoing work is to further divide a complete IEEE802.11a synchronizer into sub blocks that can be implemented in an ASIC or FPGA.

References

[1] *IEEE802.11a-1999 Part 11: Wireless Lan Medium Access Control(MAC) and Physical Layer(PHY) specifications*, IEEE.

[2] B. Y. Prasetyo, F. Said, and A. H. Aghvami, “Fast burst synchronisation technique for OFDM-WLAN systems,” *IEE Proc. Commun.*, vol. 147, no. 5, Oct 2000.

[3] M. Nilsson, “Timing and frequency synchronization in OFDM systems,” Master’s thesis, Linköpings Universitet, 1997.

[4] J. Heiskala and J. Terry, *OFDM Wireless LANs: A Theoretical and Practical Guide*. SAMS, 2002.

[5] K. Sathananthan and C. Tellambura, “Probability of error calculation of OFDM systems with frequency offset,” *IEEE Transactions on communications*, vol. 49, no. 11, November 2001.

[6] P. H. Moose, “A technique for orthogonal frequency division multiplexing frequency offset correction,” *IEEE Trans. on Commun.*, vol. 42, Oct 1994.

[7] C. J. You and H. Horng, “Optimum frame and frequency synchronization for OFDM systems,” in *Int. Conf. on Consumer Electronics*. IEEE, 2001, pp. 226–227.

[8] J.-J. van de Beek, M. Sandell, and P. Börjesson, “ML estimation of time and frequency offset in OFDM systems,” *IEEE Trans. on Signal Proc.*, vol. 45, no. 7, pp. 1800–1805, July 1997.

[9] Y.-S. Choi, P. J. Voltz, and F. A. Cassara, “ML estimation of carrier frequency offset for multicarrier signals in rayleigh fading channels,” *IEEE Trans. on Vehicular Technology*, vol. 50, no. 2, pp. 644–655, March 2001.

[10] J. Li, G. Liu, and G. B. Ginnakis, “Carrier frequency offset estimation for OFDM-based WLANs,” *IEEE Signal Processing Letters*, vol. 8, no. 3, March 2001.

[11] K. Nikitopoulos and A. Polydoros, “Compensation schemes for phase noise and residual frequency offset in OFDM systems,” in *Global Telecommunications Conference*. IEEE, 2001.

[12] H. Zhang, Z. Wang, and S. S. Chandra, “Implementation of Frequency Offset Correction Using CORDIC Algorithm for 5 GHz WLAN Applications,” in *The 8th International Conference on Communication Systems*, vol. 2. IEEE, 2002, pp. 983–987.