Low-Overhead, Low-Complexity [Burst] Synchronization for OFDM

Timothy M. Schmidl and Donald C. Cox
Stanford University, Stanford, CA 94305

Abstract — A rapid synchronization method is presented for an OFDM system using either a continuous transmission or a burst operation over a time-varying, fading channel. It will acquire the signal and provide channel estimation upon the receipt of just one training sequence of two symbols in the presence of unknown symbol and frame timing, large carrier and sampling frequency offsets, and very low SNR’s, while maintaining low latency and low complexity. It can then track the signal with the same algorithms.

1. Introduction

In an orthogonal frequency division multiplexing (OFDM) system, synchronization of the carrier frequency at the receiver must be performed very accurately, or there will be a loss of orthogonality between the subsymbols. This paper describes a method to acquire synchronization for either a continuous stream of data as in digital TV or for a burst of data as in a wireless local area network (WLAN). In both cases the receiver must continuously scan for incoming data, and rapid acquisition is needed. The ratio of the number of overhead bits for synchronization to the number of message bits must be kept to a minimum, and low-complexity algorithms are needed.

Acquisition is achieved in three separate steps through the use of a two-symbol training sequence. These symbols may be placed in the center of the burst for a WLAN application to decrease the effects of sampling frequency offset and the time variation of the channel, and they will be placed at the start of the frame for a digital TV signal. The use of a repetition was first applied to OFDM by Moose [1] and Classen and Meyr [2]. This paper introduces a low-complexity method to acquire synchronization and greatly extends the acquisition range for the carrier frequency offset. First the symbol/frame timing is found by searching for a symbol in which the first half is identical to the second half. Then the carrier frequency offset is partially corrected, and a correlation with a second symbol is performed to find the carrier frequency offset. Finally, in an optional step, the sampling frequency offset is estimated.

An example of a digital TV signal will be used to show the operation of the synchronization algorithms. The parameters are taken from the European specification for digital TV [3]. There are 6785 subcarriers (-3392 to 3392) in a 7.57 MHz bandwidth for a symbol length of 896 μs, and there is an additional guard interval of 28 μs. It is sampled at a rate of 8192 samples/symbol (which is 0.14 MSPS), and there are 96 symbols in a frame.

2. OFDM Principles

The OFDM signal is generated at baseband by taking the IFFT of QAM subsymbols \( c_k = a_k + j b_k \). An OFDM symbol has a useful period \( T \) and preceding each symbol is a cyclic prefix of length \( T_p \), which is longer than the channel impulse response so that there will be no intersymbol interference [ISI]. The frequencies of the complex exponentials are \( f_k = k/T \), and the useful part for 2N+1 subcarriers is given by

\[
u(t) = \sum_{k=-N}^{N} c_k \exp(j2\pi f_k t), \quad 0 \leq t \leq T\]

After quadrature modulation, upconversion to radio frequency (RF), passing through the channel, downconversion to an intermediate frequency (IF), and quadrature demodulation, a carrier frequency offset of \( \Delta f \) causes a phase rotation of \( 2\pi \Delta f T \). If uncorrected this causes both a rotation of the constellation and a spread of the constellation points similar to additive white Gaussian noise (AWGN). A symbol-timing error will have little effect as long as all the samples taken are within the length of the cyclically-extended OFDM symbol. A sampling frequency offset will result in the first sampling point changing by \( \Delta f \) from symbol to symbol, which will cause a rotation of each subsymbol by \( 2\pi k\Delta f / T \), where \( k \) is the frequency number of the subsymbol. The phase rotations tend to cancel out for the positive and negative frequencies, so that even a large sampling offset will have little effect on the symbol/frame timing recovery or the carrier frequency offset estimation.

3. Estimation of Symbol Timing

3.1. Symbol Timing Estimation Algorithm

The symbol timing recovery relies on searching for a training symbol with two identical halves, which will remain identical after passing through the channel, except that there will be a phase difference between them caused by the carrier frequency offset. The two halves of the training symbol are made identical by transmitting a pseudo-noise (PN) sequence on the even frequencies, while zeros are used on the odd frequencies. This means that at each even frequency one of the points of a QPSK constellation is transmitted. In order to maintain an approximately constant signal power for each symbol the frequency components of this training symbol are multiplied by \( \sqrt{2} \) at the transmitter, or the four points of the QPSK constellation are selected from a larger constellation, such as 64-QAM, so that points with higher energy can be used. Transmitted data will not be mistaken as the start of the frame since any actual data must contain odd frequencies. The second training symbol contains a PN sequence on the odd frequencies to measure these subchannels, and another PN sequence on the even frequencies to help determine frequency offset. Table I illustrates the use of PN sequences in the training sequence for an OFDM signal with 8 subcarriers (-4 to 4 excluding DC) with the points chosen from a subset of a 64-QAM constellation.

Complex samples \( r_m \) are taken by mixing the received signal down to IF, splitting the signal into two branches, multiplying by both \( \cos(2\pi f_{IF}t) \) and \( \sin(2\pi f_{IF}t) \), and low-pass filtering and sampling to get baseband in-phase and quadrature components. Consider the first training symbol where the first half is identical to the second half, except for a phase shift caused by the carrier frequency offset. If the conjugate of a sample from the first half is multiplied by the corresponding sample from the second half \( T/2 \) seconds later, the effect of the channel should cancel, and the result will have a phase of approximately \( \phi = \pi T \Delta f \). At the start of the frame, all these products will add coherently.
Let there be \( L \) complex samples in one-half of the first training symbol, and let the sum of the pairs of products be

\[
P(d) = \sum_{m=0}^{L-1} (r^*_{d+m} r_{d+m+L}),
\]

which can be implemented with the iterative formula

\[
P(d + 1) = P(d) + (r^*_{d+L} r_{d+2L}) - (r^*_{d-1} r_{d+L-1}).
\]

The received power for the second half-symbol is defined by

\[
R(d) = \sum_{m=0}^{L-1} |r^*_{d+m+L}|^2,
\]

which can also be calculated iteratively. \( R(d) \) may be used as part of an automatic gain control (AGC) loop. The best estimated timing is the index \( d \) which maximizes

\[
M(d) = \frac{|P(d)|^2}{R(d)}.
\]

### 3.2. Properties of Symbol Timing Estimate

Let each complex sample \( r_m = s_m + n_m \) be made up of a signal and a noise component. Let the variance of the real and imaginary components be:

\[
\begin{align*}
E[Re\{s_m^2\}] &= \sigma_s^2, \\
E[Im\{s_m^2\}] &= \sigma_s^2, \\
E[Re\{n_m^2\}] &= \sigma_n^2, \\
E[Im\{n_m^2\}] &= \sigma_n^2,
\end{align*}
\]

so that the SNR is \( \sigma_s^2/\sigma_n^2 \). To find the mean and variance of the estimator at the best symbol timing, first look at \( P(d) \) which can be written as

\[
P(d) = \sum_{m=0}^{L-1} \left( s^*_{d+m} s_{d+m+L} + s^*_{d+m} n_{d+m+L} + n^*_{d+m} s_{d+m+L} + n^*_{d+m} n_{d+m+L} \right)
\]

At the correct symbol timing this can be broken into parts that are in-phase and quadrature to the \( s^*_{d+m} s_{d+m+L} \) product which has phase \( \phi \). When the magnitude is taken the quadrature part will be small compared to the in-phase part and can be neglected, so

\[
|P(d_{\text{opt}})| \approx e^{-j\phi} \sum_{m=0}^{L-1} s^*_{d+m} s_{d+m+L} + \sum_{m=0}^{L-1} \left( s^*_{d+m} n_{d+m+L} + n^*_{d+m} s_{d+m+L} \right)
\]

where \( \text{Phase}_\phi \cdot \) means the component in the \( \phi \) direction.

Define the square root of \( M(d) \) to be \( q(d) = |P(d)|/R(d) \).

\[
\mu_q = E[q(d_{\text{opt}})] = \frac{\sigma_s^2}{\sigma_s^2 + \sigma_n^2},
\]

\[
\sigma_q^2 = \text{Var}[q(d_{\text{opt}})] = \frac{(1 + \mu_q^2)\sigma_s^2 \sigma_n^2 + (1 + 2\mu_q)\sigma_n^4}{2L(\sigma_s^2 + \sigma_n^2)^2}.
\]

Since \( M(d_{\text{opt}}) \) is \( (q(d_{\text{opt}}))^2 \),

\[
M(d_{\text{opt}}) = (\mu_q + n(0, \sigma_q^2))^2 = \mu_q^2 + 2\mu_q n(0, \sigma_q^2) + (n(0, \sigma_q^2))^2
\]

The expected value and variance are:

\[
\mu_M = E[M(d_{\text{opt}})] = \mu_q^2 + \sigma_q^2
\]

\[
\approx \frac{\sigma_q^2}{(\sigma_s^2 + \sigma_n^2)^2}
\]

\[
\approx 1 \text{ at high SNR.}
\]

\[
\text{Var}[M(d_{\text{opt}})] = \frac{4\mu_q^2 \sigma_q^2 + 2\sigma_q^4}{L(\sigma_s^2 + \sigma_n^2)^4}
\]

\[
\approx \frac{4}{L \cdot \text{SNR}} \text{ at high SNR.}
\]

The value of \( M(d_{\text{opt}}) \) also can give an estimate of the SNR which is

\[
\text{SNR} = \frac{\sqrt{M(d_{\text{opt}})}}{1 - \sqrt{M(d_{\text{opt}})}}.
\]

At lower values of SNR, \( \text{SNR} \) is approximately Gaussian with

\[
E[\text{SNR}] = \frac{\sigma_s^2}{\sigma_n^2} = \text{SNR}
\]

\[
\text{Var}[\text{SNR}] = \frac{\sigma_s^2}{(1 - \mu_q^2)^2}.
\]

This estimator works well for the SNR below 20 dB. Above this level, \( M(d_{\text{opt}}) \) is so close to 1 that an accurate estimate of the SNR can not be determined, but only that the SNR is high. For example, if \( M(d_{\text{opt}}) = 0.9 \), then \( \text{SNR} = 12.7 \) dB. This can be used to set a threshold so that very weak signals will not be decoded, or it can be used in WLAN to feed back to the transmitter to indicate what data rate will be supported so that an appropriate constellation size can be chosen. A lookup table can be implemented based on \( M(d_{\text{opt}}) \), so that no square roots or divisions need to be performed.

Even if there is a fading channel, all the signal energy will go into the signal component term except when the length of the channel impulse response becomes so large that it is longer than the cyclic prefix. At this point the energy located at longer delays becomes interference and would be added to the noise terms.

Let \( \alpha \) denote the distance from \( d \) to the best timing point as a fraction of the \( L \) samples in the repeated half-symbol so that \( \alpha = 0 \) corresponds to the best timing \( (d = d_{\text{opt}}) \).
At a position outside the first training symbol ($\alpha > 1$), $(d_{outside})$ has a chi-square distribution with two degrees of freedom and is not dependent on the SNR with

$$E[M(d_{\alpha})] = \frac{(1 - \alpha)^2 \sigma_i^2}{\sigma_i^2 + \sigma_o^2}$$

$$\approx (1 - \alpha)^2 \text{ at high SNR.} \quad (17)$$

Figure 1 shows a plot of the expected value of timing metric $M(d)$ versus SNR at both the best timing instance $d$ a point outside the training symbol for the digital TV sample ($L = 4096$ samples in half of a symbol). The dashed lines indicate three standard deviations from each curve, figure 2 shows a plot of the timing metric for a channel impulse response of length 32, a carrier frequency set of 12.4 subchannel bandwidths, and an SNR of 20 dB at the peak of the timing metric the phase of $P(d)$ is 97 degrees (not shown in figure), which corresponds to 999 subchannel bandwidths. This phase will be used in estimation of carrier frequency offset.

**Figure 1: Expected value of timing metric with $L = 4096$. Dashed lines indicate three standard deviations.**

3.3. Complexity of Symbol Timing Estimator

In order to provide fast acquisition, the samples must be continually written into a buffer, which must contain samples for at least two symbols (about 17,000 complex samples for digital TV). To determine whether $M(d)$ has risen above a threshold value, check for the condition,

$$|P(d_{\text{opt}})| > \text{threshold} \cdot |R(d)|^2 \quad (20)$$

For each sample that is processed, there are 10 real multiplications and 11 real additions. If only one out of 100 samples is processed, the metric should still be able to correctly determine when a training sequence has been received and give a rough estimate of the best timing point. Then the expected value of $M(d)$ at the best timing point is 0.98 with a standard deviation of 0.031, and at a position outside the first training symbol the expected value is 0.024 with a standard deviation of 0.024. This requires a total of about two million operations per second while waiting for the training sequence to arrive. The operation count is about evenly divided between multiplications and additions. Once the training sequence arrives, all the acquisition functions should be performed during the length of the frame so that data can be demodulated during the next frame.

With a rough estimate of the best symbol timing point, $|P(d_{\text{opt}})|^2$ can be calculated for all values of $d$ near this timing point by using samples stored in the buffer in order to get good estimates of $d_{\text{opt}}$ and the phase of $P(d_{\text{opt}})$. In this small range $(R(d))^2$ will not vary much, so it can be ignored. It will take 4096 complex multiplies and adds to get one value of $P(d)$ and several thousand more operations to iteratively calculate $P(d)$ and find the maximum for a total of about 40,000 operations. Then the phase of $P(d_{\text{opt}})$ is found by using a lookup table for the arctangent function and interpolating to get the necessary precision.

4. Estimation of Carrier Frequency Offset

4.1. Carrier Frequency Offset Estimation Algorithm

The main difference between the two halves of the first training symbol will be a phase difference of

$$\phi = \pi T \Delta f, \quad (21)$$

which can be estimated by

$$\hat{\phi} = \text{angle}(P(d)) \quad (22)$$

near the best timing point. If $|\hat{\phi}|$ can be guaranteed to be less than $\pi$, then the frequency offset estimate is

$$\hat{\Delta f} = \hat{\phi}/(\pi T), \quad (23)$$

and the even PN frequencies on the second training symbol would not be needed. This is the estimate that will be used during the tracking mode. Otherwise, the actual frequency offset would be

$$\phi = \frac{\pi T + 2z}{T}, \quad (24)$$

where $z$ is an integer. By partially correcting the frequency offset, adjacent carrier interference (ACI) can be avoided, and then the remaining offset of $2z/T$ can be found. After the two training symbols are frequency corrected by $\phi/(\pi T)$
(by multiplying the samples by \(\exp(-j2\phi/T)\)), let their FFT’s be \(x_1,k\) and \(x_2,k\), and let the differentially-modulated PN sequence on the even frequencies of the second training symbol be \(v_k\) (as illustrated in Table I). The PN sequence \(v_k\) will appear at the output except it will be shifted by 2\(g\) positions because of the uncompensated frequency shift of \(2z/T\). Let \(X\) be the set of indices for the even frequency components, \(X = \{-W, -W + 2, \ldots, -4, -2, 2, 4, \ldots, W - 2, W\}\). The number of even positions shifted can be calculated by finding \(g\) to maximize

\[
B(g) = \left| \frac{\sum_{k \in X} x_1,k + x_2,k + 2g}{2 \left( \sum_{k \in X} |x_2,k|^2 \right)} \right|^2
\]

with integer \(g\) spanning the range of possible frequency offsets and \(W\) being the number of even frequencies with the PN sequence. Then the frequency offset estimate would be

\[
\hat{\Delta f} = \left[ \phi / (\pi T) \right] + (2g/T).
\]

In the case of digital TV, a frequency offset of over 700 subchannel bandwidths (780 kHz) can be corrected if \(B(g)\) is calculated for \(g\) in the range -350 to 350. This should be a far larger acquisition range than should ever be needed.

4.2. Properties of Carrier Frequency Offset Estimate

Since the carrier frequency offset estimate is made up of the sum of the initial estimate and an even integer, the variance of the initial estimate, \(\sigma^2/\pi\), will be the variance of the final estimate if the integer \(g\) is equal to \(g_{\text{correct}}\). At high SNR,

\[
\text{Var}[\phi/\pi] = \frac{1}{\sigma^2 \cdot L \cdot \text{SNR}}.
\]

For \(L = 4096\) and SNR = 20 dB, the variance is \(2.5 \times 10^{-7}\), and the standard deviation is \(5 \times 10^{-4}\) subchannel bandwidths, which is 0.6 Hz.

The expected values and variances of the estimator \(B(g)\) can be calculated to determine if \(g_{\text{correct}}\) can be found reliably. At the correct frequency offset, all the signal products \(s_{1,h+\text{correct}} v_k s_{2,h+\text{correct}}\) have the same phase and

\[
\mu_B = E[B(\text{correct})] = \frac{\sigma^2}{(\sigma^2_2 + \sigma^2_\phi)} \approx 1 \text{ at high SNR.}
\]

\[
\text{Var}[B(\text{correct})] = \frac{4[(2 + 4\mu_B)\sigma^2_2 + (1 + 4\mu_B)\sigma^2_\phi]}{W(\sigma^2_2 + \sigma^2_\phi)^4} \approx \frac{4}{W \cdot \text{SNR}} \text{ at high SNR.}
\]

At an incorrect frequency offset the signal products no longer add in phase, and \(B(\text{incorrect})\) has a chi-square distribution with two degrees of freedom with

\[
E[B(\text{incorrect})] = \frac{1}{2W} \left( 1 + \frac{\sigma^2_2}{\sigma^2_2 + \sigma^2_\phi} \right) < \frac{1}{W}
\]

\[
\text{Var}[B(\text{incorrect})] = \frac{1}{4W^2} \left( 1 + \frac{3\sigma^4 + 2\sigma^4_2}{(\sigma^2_2 + \sigma^2_\phi)^2} \right) < \frac{1}{W^2}.
\]

4.3. Complexity of Carrier Frequency Offset Estimator

First the samples need to be frequency corrected by multiplying by complex exponentials, \(\exp(-j2\phi/T)\). The exponential is obtained from a lookup table which chooses a point on the unit circle in the complex plane given a phase angle. The term \(2\phi/8192\) can be computed to a high precision and stored in memory. Then for each sample, there is one addition to determine the argument to the exponential, one lookup (using the first several bits of the argument) to determine the value of the exponential, and one complex multiplication to perform the frequency correction. This could be about 150,000 operations for the 16384 samples to be corrected. Then these samples can be sent to the FFT chip to perform the two FFT’s. Since the FFT chip will have to process one symbol every 924\(\mu\)s, it will take about 2 ms to process the two FFT’s. Then \(B(g)\) needs to be calculated over the acquisition range. The denominator is the same value for all \(g\), so it can be calculated once or ignored completely. Multiplication by \(v_k\) is very simple, so

Figure 3 shows a plot of the expected value of frequency offset metric \(B(g)\) versus SNR for both a correct and incorrect frequency offset. The dashed lines indicate three standard deviations from each curve. At the peak of the timing metric (Figure 1) the initial frequency offset estimate was 0.3999 subchannel bandwidths. After correcting by this estimate, Figure 4 shows the frequency offset metric \(B(g)\) has a maximum at \(g = 6\), so the final frequency offset estimate is \(2g + \phi/\pi = 12.3999\) subchannel bandwidths.
calculation of $B(g)$ for one value of $g$ requires about $4W$ real multiplications and $4W$ real additions. Since the separation between the metrics for the correct and incorrect offsets is very large, only a fraction of the frequency values need be used. If one out of every 100 even frequencies is used, then the metric $B(g_{\text{correct}})$ will have an expected value of 0.99 and a standard deviation of 0.03, while $B(g_{\text{incorrect}})$ will have an expected value of 0.03 and a standard deviation of 0.03. If 21 values of $g$ are tested, carrier frequency offsets up to 22 kHz can be corrected. The number of operations is then about 6,000.

5. Estimation of Sampling Frequency Offset

5.1. Sampling Frequency Offset Estimation Algorithm

Finding the sampling frequency offset is a secondary consideration since it has a small effect in degrading the bit error rate (BER). A sampling clock offset causing the first sample to be shifted by $\Delta f$ causes a phase rotation of $2\pi k \Delta f / T$, in the $k$th subcarrier. This can be seen as a slow rotation of the constellation points, with the constellation points at higher frequencies rotating faster. The maximum rotation between adjacent subcarriers is directly proportional to the number of subcarriers, so an application which must handle a long channel impulse response (such as digital TV) will need long symbols (and thus more subcarriers) and will have more of a problem with a sampling frequency offset. For the digital TV example with the highest frequency of 3392 and a guard interval that is 1/32 of the useful symbol period, if the relatively inexpensive oscillators at the receiver have a frequency offset of at most a factor of $10^{-5}$, then the maximum phase rotation will be $360(3392)(10^{-5})(33/32) = 12.59$ degrees for subcarriers -3392 and 3392. This will result in a higher BER, especially with a 64-QAM constellation. At subcarrier number 1, the maximum rotation is only 0.0037 degrees. If a differential phase-shift keying (DPSK) modulation format is used along with differential detection, it is possible to ignore the sampling frequency offset and just make the BER slightly worse since the phase shifts do not accumulate. Also, if a small number of subcarriers are used, then the phase rotation, even over a large number of symbols, will not be enough to warrant the correction of a sampling frequency offset. For example, with a WLAN with a short channel impulse response, there might be 53 subcarriers (-26 to 26) with a maximum phase rotation of $360(26)(10^{-5})(33/32) = 0.096$ degrees. From the training sequence to either end of the burst (±47 symbols), the cumulative rotation is only 4.5 degrees at subcarriers -26 and 26. For digital TV, a sampling frequency offset adjustment will probably be needed since the phase shifts can be substantial within one frame.

Let $y_{1,k}$ and $y_{2,k}$ be the $k$th frequency components of the first and second training symbols, respectively, after finding symbol/frame timing, correcting the carrier frequency offset, taking the FFT’s of the two training symbols, and removing the differential modulation on the second symbol, which means multiplying by $y_{1,k}^*$. This last multiplication is really just changing some sign bits and swapping some real and imaginary parts. After the differential modulation is canceled, the only differences between the two sets of frequency components should be caused by the sampling frequency offset. The phase difference between each pair of frequency components can be found by taking the difference of frequency component indices and the phase differences are zero when weighted by the squares of the magnitudes of the subchannels. Note that since only the slope is being calculated, it will not be affected much by any residual frequency offset or phase noise from the local oscillator.

5.2. Variance of Sampling Frequency Offset Estimate

Each frequency component can be written as the sum of a signal and a noise term, so $y_{1,k} = \sqrt{2}s_k + n_{1,k}$ and $y_{2,k} = s_k \exp(jk\theta) + n_{2,k}$. First, the variance for a flat channel will be calculated where the weightings for the subchannel amplitudes do not need to be used. Using the small angle approximation for the arc tangent function,

$$\theta = \sum_{k \in X} \frac{k^2}{\sum_{k \in X} k^2}$$

This is equivalent to the linear minimum mean squared error (MMSE) estimator if the expected values of both the frequency component indices and the phase differences are zero when weighted by the squares of the magnitudes of the subchannels. Note that since only the slope is being calculated, it will not be affected much by any residual frequency offset or phase noise from the local oscillator.

The variance of $\theta$ can be calculated as a function of the SNR and $W$, which is the number of even frequencies. Assuming a high SNR,

$$\text{Var} \{\theta\} \approx \frac{3\sigma_n^2 + \sigma_n^4}{(2\sigma_t^2)^2} \left( \frac{\sum_{k \in X} k^2}{\sum_{k \in X} k^2} \right)^2$$

$$\approx \frac{0.75}{SNR} \left( \frac{3}{W(W+1)(W+2)} \right)$$

$$\approx \frac{2.25}{SNR W^3}.$$  

For example, with an SNR of 20 dB and $W=3392$, the variance is $5.8 \times 10^{-13}$, and one standard deviation is $7.6 \times 10^{-7}$. This is about 85 times smaller than the maximum possible value of $\theta$, which is $2 \times \pi 10^{-5} (33/32) = 6.5 \times 10^{-5}$. In order to reduce the variance of this estimate further, the training symbols can be placed farther apart. The variance is reduced by a factor of $C^2$ if the training symbols are placed $C$ times farther apart, provided that the channel does not change very much between the symbols. In the tracking mode, the sampling frequency offset can be calculated very accurately using training symbols from consecutive frames as long as the channel is slowly time-varying.
In the above estimate, the higher frequency subchannels have a larger influence than the lower frequency subchannels. Thus with a fading channel, if the higher frequency subchannels have more than the average SNR, than the estimate can be better than that for a flat channel with the same average SNR. Similarly, if the higher subchannels are worse, than it will perform more poorly. A particular fading channel will do as well as a flat channel with an SNR equal to the weighted average (by k2) of the SNR's of its subchannels.

5.3. Complexity of Sampling Frequency Offset Estimator
The maximum expected phase rotation at the highest frequency (number 3392) is about 12.59 degrees. Since all the angles are small, there are two simplifications that can be made. First there should be no phase wrap-around, so there is no need to unwrap the phase angles. Second, the small-angle approximation for the arctangent function can be used, atan(x) \approx x. Let y denote the real part and \( \text{im} \) denote the imaginary part of y. Then using the approximation for small angles

\[
\text{angle}(a + jb)(c + jd) = \frac{1}{2} \left( \text{atan}\left(\frac{a \cdot c + b \cdot d}{a \cdot d - b \cdot c}\right) \right)
\]

the less computationally complex estimate of \( \theta \) is:

\[
\theta = \frac{\sum_{k \in X} (y_{2k+1}^* y_{2k+2} + y_{2k+2}^* y_{2k+1})}{\sum_{k \in X} (y_{2k+2}^* y_{2k+3} + y_{2k+3}^* y_{2k+2})} k^2
\]

which requires 12 real multiplies, 6 real additions, and 1 real division for each of the W terms in the estimate. The lower frequency components have very little effect on the estimate of the sampling frequency offset, so they can be safely ignored if they are not needed to help unwrap the phase. For example, frequency 1 has 1/10,000 times the influence of frequency 100 if both subchannels have the same attenuation. If only the highest half of the frequencies are used [-3392, -3390, ..., -1700, -1698, 1700, 1702, 3390, 3392], this cuts the number of computations by a factor of 2, but only increases the variance of the estimate by 7%. Using only the higher half of the frequencies, there are about 60,000 operations. The total of all of the processing for acquisition of the signal can be about 250,000 operations. This will allow all the necessary computations to be completed within one 88 ms frame if they can be performed at a rate of about 3 million operations per second.

If the data format is a continuous stream of data such as in digital TV, then all the offsets can be computed in one frame, and then the IF and sampling oscillators can be adjusted so that all the additional frames can be decoded. If a burst of data is to be decoded, as with a wireless LAN, then the samples from the entire burst must be buffered while synchronization is performed.

6. Tracking Mode Operation
Most of this paper discusses acquisition of the OFDM signal, but the same estimators can be used in a tracking mode. The metric \( |P(d)| \) from equation (2) can be calculated in the neighborhood of the first training symbol to monitor the starting position of the frame. The frequency offset can be calculated from the phase of \( P(d) \) by using equation (22). The sampling frequency offset can be calculated from the FFT's of the first training symbols from consecutive frames by using equation (32) with \( v_k = 1 \) and without correcting for the residual frequency offset.

7. Comparison with Other Techniques
There have been many papers about synchronization for OFDM. Many of them assume the correct timing or have a limited acquisition range for the carrier frequency offset such as [1]. In [2], the acquisition range is only expanded by doing an exhaustive search over the entire range using fractional steps, which requires a tremendous amount of computation. Nogami and Nagashima’s algorithm [4] has a large acquisition range for the carrier frequency offset, but the symbol timing is only roughly estimated by searching for a Null symbol with zero power, and a limited acquisition range for the carrier frequency offset. For example, the variance of their frequency offset estimator over a fading channel with 822 subcarriers and an SNR of 20 dB is \( 3 \times 10^{-4} \). Using the same number of subcarriers, the variance for the method described in this paper is \( 4 \times 10^{-8} \), which is about 75 times smaller. Nogami and Nagashima’s algorithm does not include channel estimation and requires two training symbols - a Null symbol for symbol timing and a training symbol to calculate frequency offset. For the algorithm in this paper, if channel estimation is not needed, then the PN sequence on the odd frequencies would not be needed, so the overhead would be at most 1.5 symbols for a reduction in overhead of 25%. This could be reduced further by transmitting actual data on the even frequencies of the first training symbol.

8. Conclusion
A method has been presented for the rapid and robust synchronization of OFDM signals. Acquisition is obtained upon the receipt of just one training sequence, and the computational complexity and overhead is low. By averaging over all the subchannels, it works well in frequency selective fading channels. This makes it ideal for digital TV so that synchronization can be achieved within a maximum of 2 frames in the worst case - up to one frame to wait for the training symbol to arrive, and the duration of one frame to process the training symbol. This would require a maximum of 0.17 seconds for acquisition of the signal. This method also gives very accurate estimates of symbol timing and carrier frequency offset, and provides a very wide acquisition range for the carrier frequency offset. It also provides an estimate of the SNR and the probability of false locks or missing the training symbols is very low. The same algorithms with some simplifications can be used to track the signal. For a wireless LAN, such a fast and low-overhead synchronization process is necessary because there will be only one training sequence transmitted in each burst that is available for synchronization.

References