TIMING AND FREQUENCY SYNCHRONIZATION

IN ULTRA HIGH SPEED OFDM LINKS

A THESIS

submitted by

MALLA DINESH
(EE09B021)

for the award of the degree

of

BACHELOR OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY MADRAS

MAY 2013

THESIS CERTIFICATE

This is to certify that the thesis titled **TIMING SYNCHRONIZATION AND FREQUENCY ESTIMATION IN OFDM SYSTEMS**, submitted by **Malla Dinesh (EE09B021)**, to the Indian Institute of Technology Madras, Chennai for the award of the degree of **Bachelor of Technology**, is a bona fide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. K. Giridhar
Research Guide
Professor
Dept. of Electrical Engineering
IIT-Madras, 600 036

Date: 28th May 2013

Dr. Sheetal Kalyani
Research Guide
Assistant Professor
Dept. of Electrical Engineering
IIT-Madras, 600 036

Date: 28th May 2013

Place: Chennai

ACKNOWLEDGEMENTS

This thesis is the end of my journey in obtaining B.Tech degree. I have not travelled alone in this path. Along the course, there has been constant support and encouragement from numerous people including my professors, my friends and my well-wishers too. At the end of my thesis, I feel it right to convey my gratitude to all those who have contributed to my success and made this one unforgettable experience.

At this moment of accomplishment, first and foremost I offer my sincerest gratitude to my project guides Prof. K Giridhar and Prof. Sheetal Kalyani for their unequivocal support and invaluable advice, for which I am extremely grateful. I am especially thankful for the patience they have shown, unaltered by the bad results that I have achieved during the initial phases of my project. Only through them, I have developed a real zeal towards this field of research and been motivated constantly at all times to push myself into achieving better things.

I am indebted to many of my friends and classmates for making my stay in IIT a stimulating and a fun filled experience. I wish to thank my best friends Ankit, Sidharth and Vaibhav for their undistinguished helping nature and their moral support. My special appreciation goes to Sunaina and Bande for their friendship and the most random discussions that we always had. I would like to extend a huge, warm thanks to my roommates Karthik and Kass for their help and support, not to forget their invaluable teachings about life and politics. Also my friends, Kolan and Chitra deserve a special mention here for their constant fun-taunting about my project, which challenged me to work hard into producing better results.

Last but not least, I would like to pay high regards to my Dad, Mom and Sister for their sincere encouragement and inspiration they have given all through my life. I owe everything to them. A special acknowledgment to all the people who, knowingly or unknowingly helped me in the successful completion of this project.

ABSTRACT

KEYWORDS: Doppler Estimation, Timing Synchronization, Frequency estimation, OFDM systems

Owing to the growing demand for the high-speed wireless communication, there is a need to provide high-quality and efficient transmission schemes to overcome various channel impairments appearing over the transmission link. One of the main problems in providing such mechanisms is Doppler shift. It plays a crucial role in reducing the spectral efficiency of the system, raising a need to determine its value with high accuracy. In order to estimate this frequency shift, first the timing recovery is required. So these two problems of timing synchronization and frequency estimation go hand-in-hand.

Over the years, numerous algorithms have been designed to determine this Doppler shift with considerate amount of success. But with the increasing speed of the vehicle and the worsening channel conditions, these mechanisms succumb to failure. The primary objective of this project is to determine these limitations of certain algorithms and to suggest necessary improvements to be made, so as to push them to work better at harder conditions too.

The first chapter deals with the introduction to CP correlation method and Schmidl-Cox algorithm for timing synchronization, their performance and their limitations. In second chapter, the improvisations that can be made to the Schmidl-Cox algorithm are discussed in detail and the resulted better performance is shown with the tabulations. The final chapter deals with the existing algorithms for frequency estimation and limitations. Finally, the necessary steps that can be taken to improve the performance of this estimation are suggested.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

OFDM	Orthogonal Frequency Division Multiplexing
SNR	Signal to Noise Ratio
AWGN	Additive White Gaussian Noise
FFT	Fast Fourier Transform
ISI	Inter Symbol interference
BER	Bit Error Rate
CP	Cyclic Prefix
PDP	Power Delay Profile
dB	Decibel
QPSK	Quadrature Phase Shift Keying

NOTATIONS

P	Schmidl-cox Correlation value
PP	Schmidl-Cox Correlation value using multiple repetitions
S	Double Schmidl-cox Correlation value
T	Double Schmidl-Cox Correlation value using multiple preambles
Δf	Frequency Shift
α	Phase of Correlation
$\overline{\Delta f}$	Estimated Fractional Frequency Offset

CHAPTER 0

INTRODUCTION

Owing to the rapid growth in the demand for high-speed wireless-communication, or in specific high-speed satellite communication, it is essential to provide high-quality and efficient transmission schemes to overcome various channel impairments appearing over the transmission link. One of the main problems in providing such mechanisms is Doppler shift. It plays a crucial role in reducing the spectral efficiency of the system, raising a need to determine its value with high accuracy.

Over the years, numerous algorithms have been designed to determine this Doppler shift with considerate amount of success. But with the increasing speed of the vehicle and the worsening channel conditions, these mechanisms succumb to failure. The primary objective of this project is to determine the limitations of certain algorithms and to suggest any improvements to be made so as to push them to work better at harder conditions too.

In any communication system, where there is a burst transmission, it is of utmost importance to maintain timing synchronization, since it serves as the timing reference to the data symbols that follow. In OFDM system this means figuring out the end of the preamble, which in turn gives an estimate of where the data starts. The frequency shift problem is entangled with this timing synchronization, without which the estimation of its shift is not possible. So the frequency estimation and timing synchronization algorithms go hand-in-hand. Several efficient and accurate timing recovery algorithms have been proposed to obtain "coarse" and "fine" timing synchronizations. But the severe effects of the time slip enforces these techniques to be performed regularly at several instances, in order to correct the error being accumulated gradually over time. The preambles reduce bandwidth efficiency considerably, so the algorithms are to be made robust and efficient so as to optimize the number of preambles used.

Time Offset:

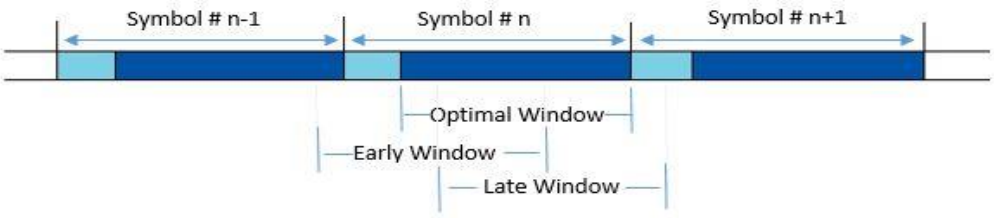
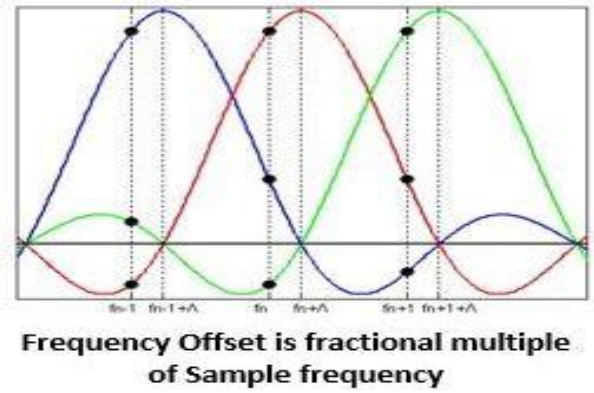


Figure 0.1: Effects of Timing Offset

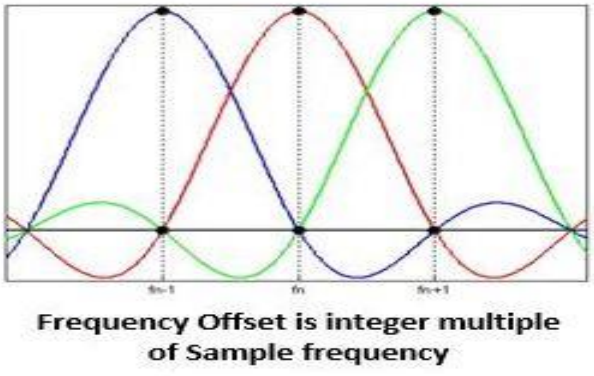
The primary objective is to determine where to start the FFT-window:

1) The window need not coincide exactly with the signal position. Due to the FFT cyclic-properties, it can start from any sample within the cyclic-prefix region. This would result in a cyclically-shifted signal.
2) If the window spans samples from two different symbols, it would result in an Inter-Symbol-Interference (ISI)

Frequency Offset:



There is a loss of orthogonality between the sub-carriers resulting in the severe degradation of the BER.



Although the sub-carriers are still orthogonal to each other, their frequency spectrum of the received data shows that they are in wrong positions, i.e., shifted cyclically.

Figure 0.2: Effects of Frequency Offset

With respect to the timing synchronization algorithms, the analysis was kicked off with the implementation of Cyclic-Prefix (CP) Correlation method. Despite being extremely simple, it was not preferred owing to various limitations, the primary being insufficient correlation window length. On the other hand, the Schmidl-Cox algorithm proved to be quite successful even at channel noises of around 6dB SNR and upto 0.01 relative frequency shift. But the satellite systems that we are aiming at can have worse Doppler shifts as high as 0.7 (relative) and noise levels upto 0dB SNR. The Schmidl-Cox algorithm alone is not capable of handling such harsh conditions. But then there was a scope to improvise upon at several fronts.

The number of signal repetitions in a preamble was primarily increased to four, which paved way for a serious development of the algorithm. The correlations were no longer limited to the adjacent lobes (signal), all possible combinations have been included into the correlation factor. This has showed clear signs of improvement in the efficiency of the system.

But the main problem with respect to this Schmidl-Cox algorithm was the lack of a more specific designation for the timing synchronization in the correlation graph. Determining the end point of the plateau becomes an illusionary task, once the noise factor came into picture. Also, at high levels of noise, the correlations are not capable of carrying out adequate averaging of noise. On careful examination of the structure of the plateau, it can be realized that the plateau itself can act as a pilot with multiple repetitions. So, a second correlation was carried out on the Schmidl-Cox correlation graph, with window length of half the size of plateau. This scheme has clearly produced astounding results making the system robust and not to falter even at -3dB SNR AWGN noise.

From the list of the timings recovered, it is easy to observe that error is symmetrical over the actual timing, i.e., it is positive or negative with equal probability. So, using multiple preambles across the time domain and adding their individual correlation graphs would definitely improve the performance.

After the above additional improvement, the timing can be recovered very accurately, at 99% efficient system performance, the error of the timing synchronization is just ±10 samples. Also the performance doesn't degrade at all even in the presence of a channel. So even considering a 40-tap PDP channel at 0 dB SNR AWGN noise, the timing can still be recovered with 99% accuracy.

In the frequency estimation algorithm, there are two steps. One, to determine the fractional frequency offset and the other, integer offset. There is a standard algorithm by Schmidl-Cox to evaluate each of these offset components using two preamble symbols. The scheme used to find integer offset is very efficient, it works perfectly even at noise of -6dB SNR AWGN noise without a slightest hint of error. But the fractional part has significant error at high noise levels. This can be attributed to improper averaging out of noise in the correlation made. So, similar to the mechanism in the Timing Synchronization case, all possible correlations are considered and their average is taken. Once this correction was made, it can be observed that the error is both positive and negative with equal probability. So considering multiple preambles would definitely better the performance. At the end, this improvised mechanism is very efficient. At 0dB SNR AWGN channel, the error is less than 0.1%.

CHAPTER 1

TIMING SYNCHRONISATION ALGORITHMS

This chapter deals with certain standard algorithms which have proved to be quite successful. First the cyclic prefix correlation method has been introduced and its advantages and disadvantages have been discussed. Finally, the famous Schmidl-Cox algorithm has been introduced.

1.1 Introducing algorithms:

In general, the repetitive structure of the preamble is exploited by using correlation methods. Here, the symbol is either auto-correlated with itself or it can be cross-correlated with the previously saved pattern, for example the symbol on the transmitter side.

Synchronization in OFDM systems can be done using:

1) Methods based on redundancy exploiting the inherent characteristics (viz., cyclic extensions)
   Example: Cyclic-Prefix (CP) Correlation method
2) Methods based on pilot sequences
   Example: Schmidl-Cox method

Cyclic Prefix: One of the main advantages of the cyclic prefix is its ability to protect the OFDM symbol from the multipath delay spread. It acts as a guard interval between subsequent symbols preventing the Inter-Symbol-Interference (ISI). Here, the length of the guard interval must be greater than the maximum delay spread introduced by the channel.

It is just the cyclic extension of the waveform into the guard interval, i.e., part of OFDM symbol at the end is copied as the cyclic prefix.

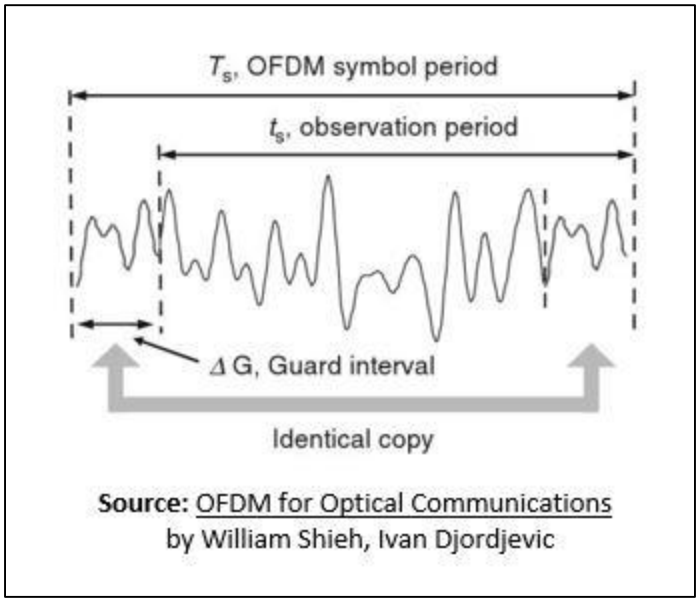


Figure 1.1: A sample figure of an OFDM symbol with a cyclic prefix

1.2 Cyclic-Prefix (CP) Correlation:

The symbol boundary can be estimated based on the correlation which utilizes the repetitive structure of an OFDM symbol. This exploits the fact that the Cyclic-Prefix is identical to the last part of the symbol. Under this scheme, the correlation is carried out between the set of received symbol samples arrived during a time interval of length equal to that of cyclic prefix and another set delayed by 'N' (length of the frame minus length of the cyclic-prefix). The correlation reaches its expected maximum when the sets considered would be the cyclic prefix and its identical counterpart at the end of the symbol, enabling to estimate the boundaries precisely in the absence of noise. '*x*' is the input OFDM signal.

$$Correlation\ Value: \qquad P(d) = \sum_{i=0}^{Ncp-1} x(d-i)x(d-i-N) \qquad (1.1)$$

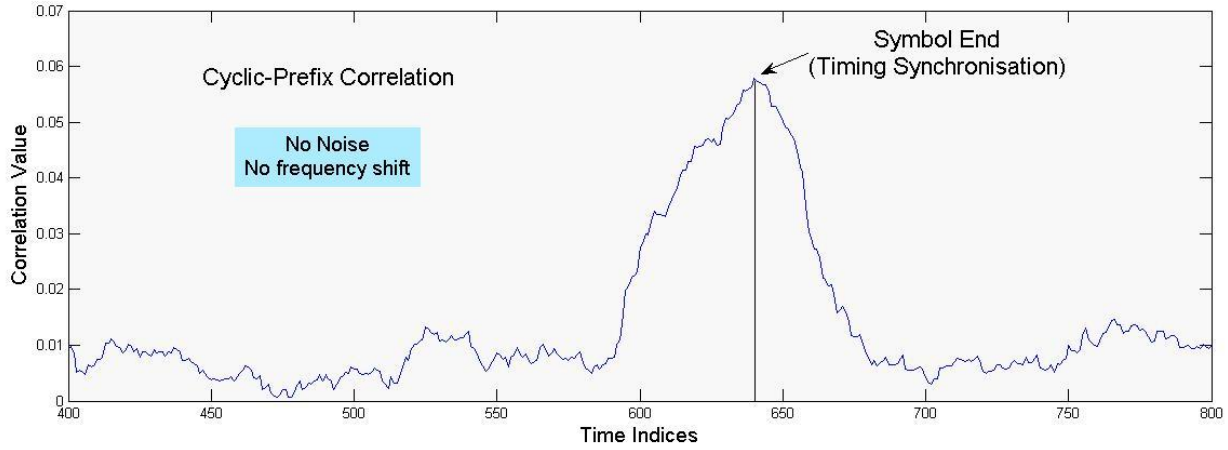


Figure 1.2: Graph depicting a typical CP correlation

It is easy to detremine the peak in the absence of any external factors like Noise, Frequency Shift, and Channel Fading. But once these factors are accounted for, the peak is not tweaked properly and several other global peaks rise up due to noise, so we have to settle down for a rough approximation by making sure that the end of the symbol is still estimated to fall in the same symbol. So, effectively we would read a cyclically-shifted version of the transmitted signal at the receiver which, later, through several cyclic FFT properties, can be studied.

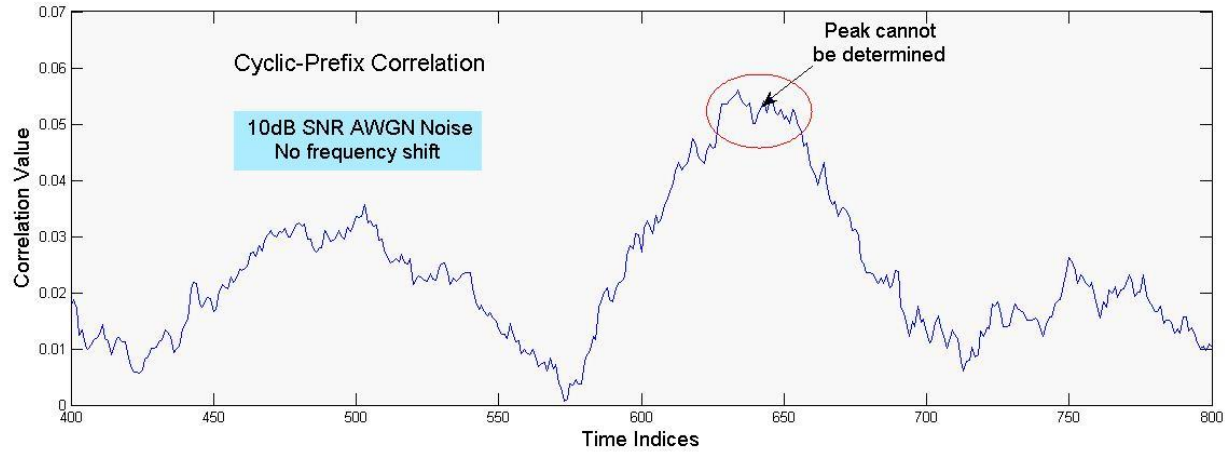


Figure 1.3: Graph depicting CP correlation with a system with high noise

90% estimate: In this scheme, first the peak of the curve is to be detected and then the points to the left and to the right of the peak with amplitude more than 90% of the peak amplitude. The average of these two positions is to be noted as our timing synchronization. In general, adding an offset of around 'Ncp/2' would increase the probability of this timing recovered falling in the center of the cyclic-prefix region.

Frequency Shift ($\Delta f/fs$)	@ 6dB (Efficiency)	@ 3dB (Efficiency)
0.01	81	69
0.05	79	72
0.1	78	74
0.3	82	70
0.4	80	72
0.5	81	71
0.6	83	70
0.7	79	70
0.8	80	72
0.9	81	73

Table 1.1: Efficiency of CP correlation scheme at different SNR values

The main advantages of this estimator are its simplicity and the absence of the pilots. This would facilitate the algorithm to be applied on symbols more frequently to counteract the effects of time-slip without any actual requirement of pilot symbols.

But this scheme is not robust, it succumbs in the case of high SNR. This owes mainly to the fact that the correlation is carried out over a very small interval (CP length), so the noise hasn't been averaged out properly, resulting in huge errors. But the increase in CP length would result in the degradation of the bandwidth efficiency, so it is not recommended either. So owing to these limitations, it is observed that there is a need for pilot symbols with repetitive patterns over long intervals, through which the correlations can be carried out with better neutralizing of the noise.

1.3 Schmidl-Cox Algorithm:

This scheme uses pilot sequences at regular intervals of time to identify the start of the frame and then to correct it in case of any slip. In each preamble, two (or more) repetitions of the same samples are transmitted, so these parts still remain identical on the receiver side, except for a phase being added because of the frequency shift. So correlation of the signal with a 'N/rep' shifted version of itself, would give a maximum when the correlation window is across the two repetitions.

$$OFDM\ symbol\ representation: \qquad s(n) = \frac{1}{N}\sum_{k=0}^{N-1} x(k)e^{j2\pi k\frac{n}{N}} \qquad (1.2)$$

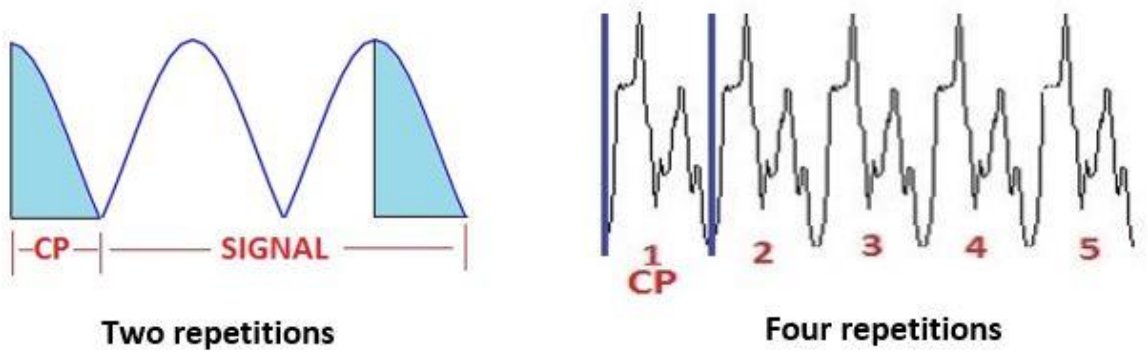


Figure 1.4: OFDM symbol preamble with repetitions

These repetitions are achieved by transmitting a pseudo-random (PN) sequence with required number of zeroes (viz., 3 zeroes for four repetitions) between two consecutive symbols. These PN symbols can be chosen randomly from a QPSK constellation, which would enable uniform avg. power over the frequency spectrum.

For a 256-point FFT OFDM system, we preferred a preamble with four repetitions. Here the correlation is carried out across samples which are spaced 'N/4' distance apart over '3N/4' samples. So, when the conjugate of the symbol is multiplied with "N/4' delayed version of itself, maximum is achieved as a plateau when the correlation interval spans the lobes {1,2,3,4} until its end when it is on {2,3,4,5}.

$$Correlation: \qquad P(d) = \sum_{n=0}^{\frac{3*N}{4}-1} r_{d-n} r^*_{d-n-\frac{N}{4}} \qquad (1.3)$$

As per the definition, plateau occurs at the position of the last lobe in the symbol, i.e., it spans a length of 'N/4' before and until the start of the CP of symbol next to preamble.

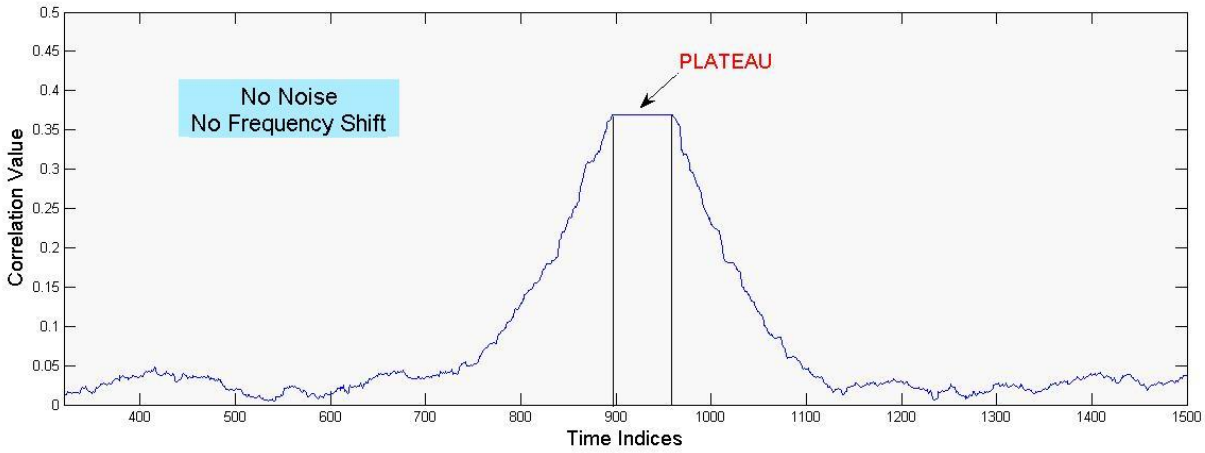


Figure 1.5: Correlation graph of Schmidl-Cox algorithm applied on an ideal system with no noise and no frequency shift

But a fine plateau can be observed only in the case of a system with no noise. In the case of frequency shift, although there is phase difference between the samples correlated, its absolute value will still remain the same, keeping the shape of plateau intact.

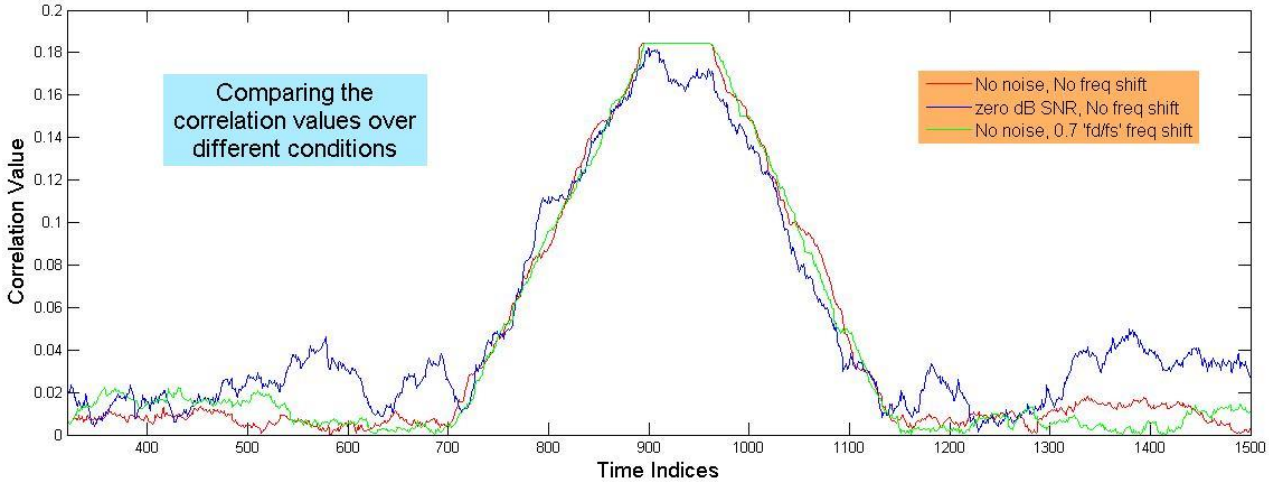


Figure 1.6: Correlation graph of Schmidl-Cox algorithm comparing at various conditions

As can be seen in the figure, it gets more difficult to determine the peak in the graph, making it impossible to find the edge of the plateau. Here 90% estimate procedure can still be carried out to find the points that atleast lie on the plateau. These simulations have produced results in a system of 0dB SNR AWGN noise which are only 70% correct.

Frequency Shift ($\Delta f/fs$)	@ 3dB (Efficiency)	@ 0dB (Efficiency)
0.01	82	72
0.05	80	69
0.1	83	76
0.3	81	74
0.4	81	72
0.5	81	70
0.6	80	71
0.7	83	72
0.8	81	75
0.9	80	71

Table 1.2: Efficiency of Schmidl-cox algorithm at several Frequency Shifts

Noise	12 dB	9 dB	6 dB	3 dB	0 dB
Efficiency	98	91	87	80	71

Table 1.3: Efficiency of Schmidl-cox algorithm at various SNR values of AWGN noise

CHAPTER 2

IMPROVISATIONS ON SCHMIDL-COX ALGORITHM

2.1. Multiple Correlations:

It can be observed that the correlation need not just be confined to the samples that are just 'N/4' distance away. Even the samples 'N/2' and '3N/4' distance apart are identical. So, correlation is extended accordingly.

$$PP(d) = \left| \sum_{n=0}^{\frac{3N}{4}-1} r_{d-n} r^*_{d-n-N/4} \right| + \left| \sum_{n=0}^{\frac{2N}{4}-1} r_{d-n} r^*_{d-n-N/2} \right| + \left| \sum_{n=0}^{\frac{N}{4}-1} r_{d-n} r^*_{d-n-3N/4} \right| \quad (2.1)$$

The absolute values of the correlations that are executed for samples spaced 'N/4', 'N/2', '3N/4' apart are added, and this cumulative value projects an improved performance than the earlier one. This improvement is attributed to more averaging out of the noise that is done by increasing the correlation window, because of which noise effects have been nullified to some extent.

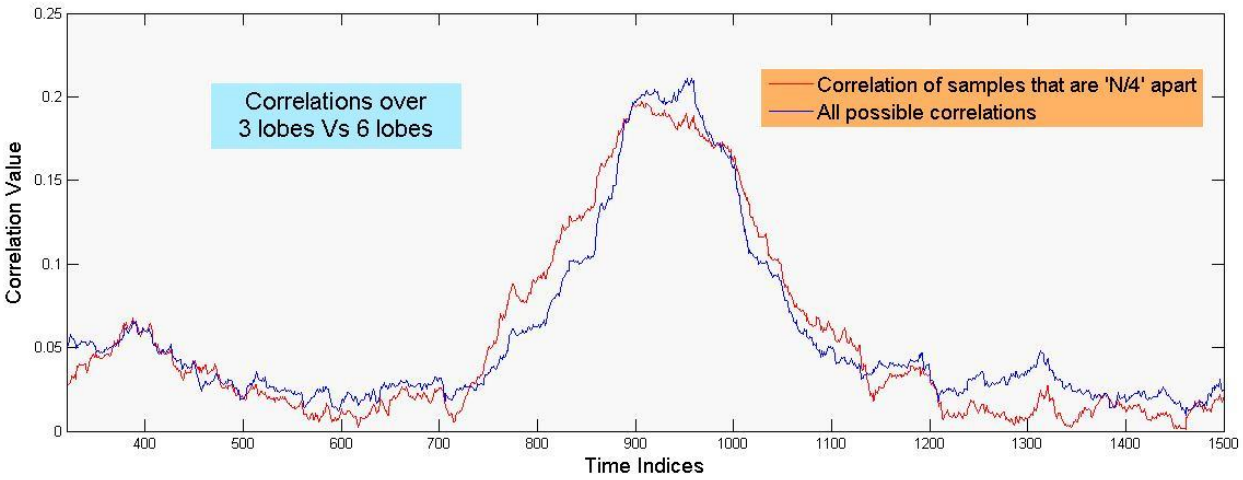


Figure 2.1: Comparison of new scheme vs old scheme

As depicted in the graph, in the new correlation curve, the shape of plateau is more imprinted which makes it all the more easy to pinpoint some point on the plateau, thus increasing the efficiency of the algorithm. Our standard method of 90% algorithm can be applied here.

Frequency Shift ($\Delta f/fs$)	@ 6dB (Efficiency)	@ 3dB (Efficiency)
0.01	92	89
0.05	93	88
0.1	92	89
0.3	92	90
0.4	92	90
0.5	91	89
0.6	89	88
0.7	91	87
0.8	91	89
0.9	90	88

Table 2.1: Schmidl-Cox algorithm with the multiple correlations

But this would provide efficient results only in the absence of delay spread. The very purpose of including cyclic-prefix is to compensate for this delay-spread, in the presence of which the starting portion of the CP is corrupted. To be more specific, the delay caused when the symbol is passed through the channel would extend into the cyclic prefix of the next symbol. This would effectively reduce the length of the plateau, since the starting few samples in a symbol will no longer be identical to that of the actual signal samples.

So if the timing recovered lies in this starting portion of the CP, certain erroneous samples will be collected in every symbol, which in turn degrades the system performance. So in the 90% estimate method, after taking the average of the symbols to the right and left of the peak, it is to be shifted to the right by 'Ndelay/2', i.e., care is taken for the recovered timing to fall on the center of the plateau, which is now of length '(Ncp-Ndelay)'.

Although, one interesting fact to note is that the system behavior is hardly altered when a PDP channel is introduced. The efficiency remains the same, including the mean and variance of the error (offset of the timing from the actual).

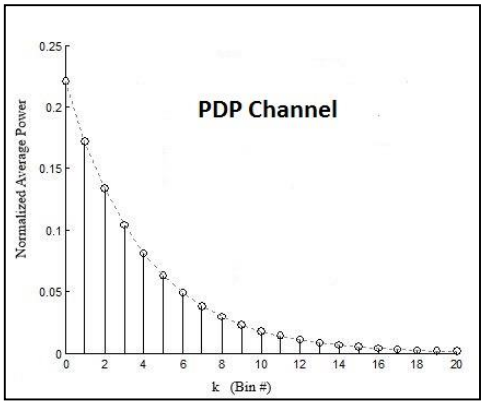


Figure 2.2: Example of a PDP sequence

2.2 Double Schmidl-Cox Algorithm:

Although, the above method is preferred to get the timing sync to fall in the cyclic prefix, like discussed, due to the constraints of channel, it is not an ideal method. Since the goal is not pinpointed (any point on the plateau) and the noise being not averaged out properly, there is a need for more robust method through which the edge of the symbol is more visible and easy to figure out.

On observing the plateau carefully, one can see that it is similar to the preamble with several repetitions, i.e., all the samples are equal (under ideal conditions). Since the length of the plateau is already small, it can be considered as a symbol with just two identical halves.

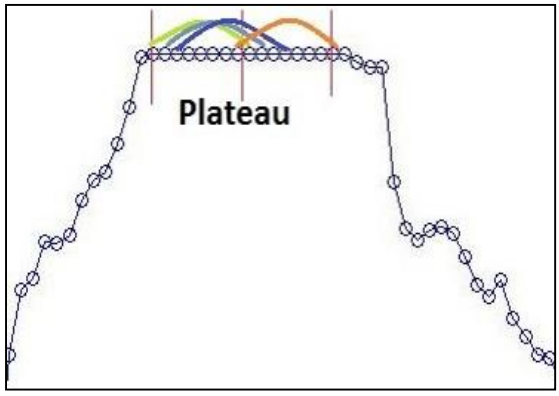


Figure 2.3: Second correlation on the plateau

So upon auto-correlation of these correlated values with 'Ncp/2' delayed version of itself, this would give a unique peak, when half the plateau is multiplied by conjugate

$$Double\ Schmidl-Cox\ Correlation: \qquad S(d) = \sum_{n=0}^{\frac{Ncp}{2}-1} P_{d-n} P^*_{d-n-\frac{Ncp}{2}} \quad (2.1)$$

of the other half of the plateau, i.e., at the end of the symbol. A triangle shaped graph is formed, in which finding the peak must be fairly easier than in the case of a plateau where the peak might not even lie in this region if the noise added to the samples is strong enough.

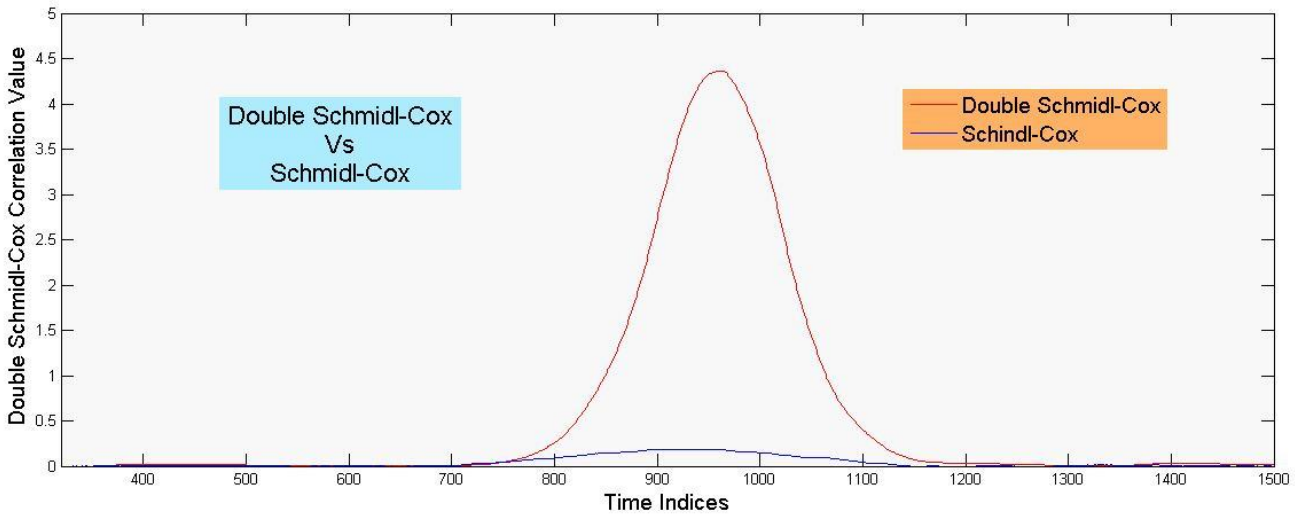
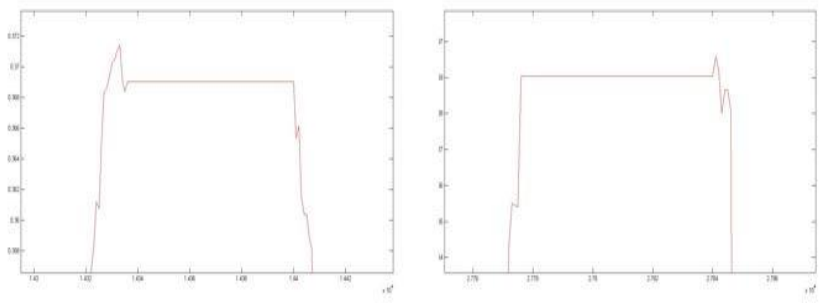


Figure 2.4: Comparing the Double Schmidl-Cox to normal Schmidl-Cox method

The advantage of this second correlation is clearly visible from the graph. The peak can be easily found out, but this cannot be directly detected as the timing, because, due to the noise, this peak can be offset from the actual timing by positive or negative value.



In the Schmidl-Cox correlation, like discussed, there can be a peak before or after the plateau depending on the error on the samples there respectively. So after a second correlation, this will shift the peak of the graph accordingly. A more robust method is needed. First the peak of the graph must be found out and let's say its amplitude is 'max'. Now, the sample to the left of the peak which has an amplitude (x*max) and a sample to the right of the peak with amplitude (y*max) are to be marked.

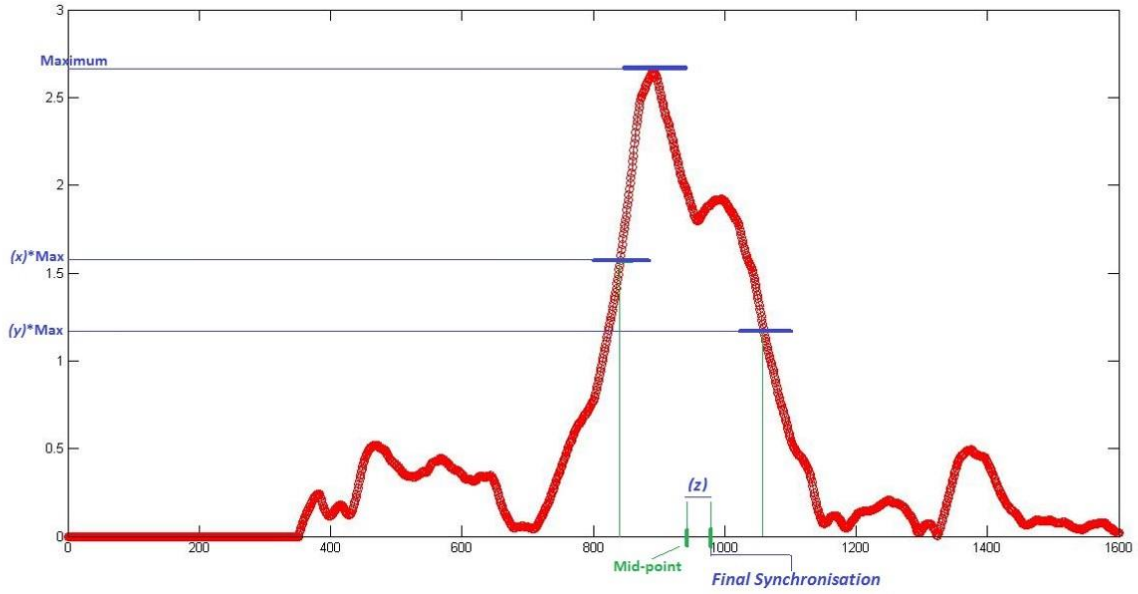


Figure 2.5: Mechanism to find the timing synchronization from the correlation graph

The average of these two points should ideally give end of the symbol timing. But the error can be positive or negative, which would lead in detecting the timing either before the end of the symbol which is still reasonable or into the next symbol which is not preferred. So we shift our final value to the left by some distance parameter 'z'.

Simulations have been carried out using several values for parameters (x,y). The ideal values have turned out to be (0.25,0.15) where the system has shown some exceptional efficiency. At 3dB SNR AWGN noise, the system is 99% efficient, i.e., the timing recovered lies in the CP range, irrespective of any amount of frequency shift.

Frequency Shift ($\Delta f/fs$)	@ 3dB (Efficiency)	@ 0dB (Efficiency)
0.01	99	90
0.05	100	91
0.1	98	87
0.3	100	92
0.4	100	90
0.5	100	92
0.6	99	92
0.7	100	92
0.8	100	91
0.9	98	90

Table 2.2: Tabulations of Double Schmidl-Cox scheme

2.3 Multiple preambles:

We need to get better results than this, because this scheme falters when a channel is introduced. So effectively the range of timings recovered must be reduced. It is evident from the results that the error is positive or negative with almost equal probability. So there is a scope for more averaging out by considering multiple preambles spread out across the time domain. The idea is to shift the correlation curve by the required amount (distance between two preambles), so as to overlap these triangle shaped correlation forms of each of the preamble on to each other and then apply the above technique on this cumulative curve to find the timing.

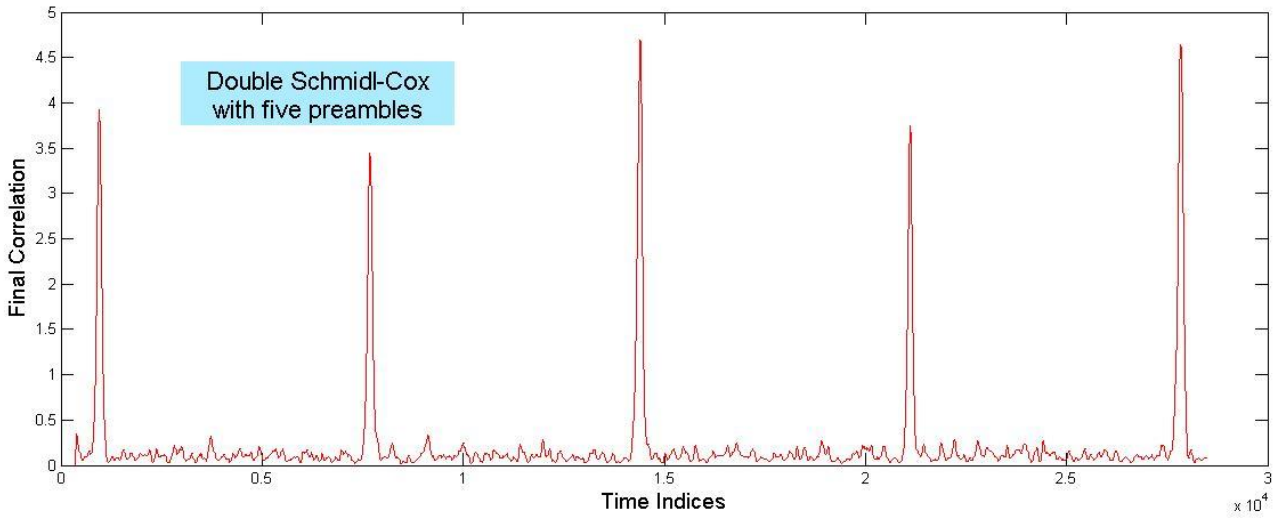


Figure 2.6: Correlation graph of Double Schmidl-Cox algorithm with five preambles each spaced 10 symbols apart

But the problem of the time slip arises here. Care must be taken that over the period, in which all these preambles are received, the time-slip must be negligible. This would raise a concern over the number of preambles used and the space between every two consecutive preambles. Owing to all these conditions, the simulations have been carried out using five preambles, each spaced 10 symbols apart.

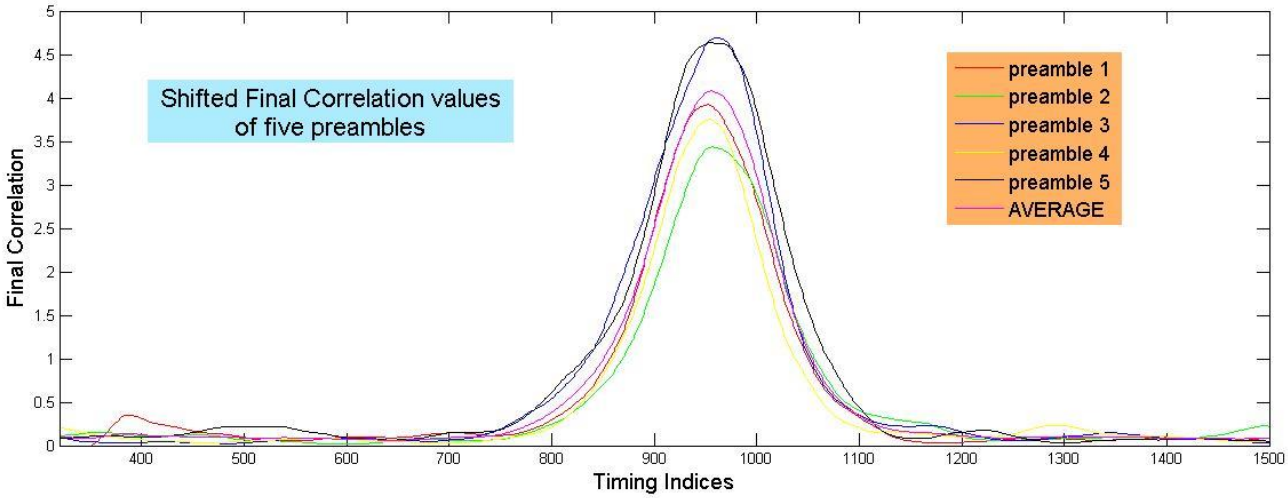


Figure 2.7: Comparison of Correlation graphs for the five preambles and the average correlation

This scheme would produce results that make the system 100% efficient at even -3dB SNR. But with the increasing noise, the timings syncs are more spread out making the system lesser efficient when a channel is introduced. At 0dB SNR, the range of timing indices is just 24 for a 256-point FFT. So, effectively even if a channel with 40-tap is introduced, the system still behaves the same with 100% efficiency.

Frequency		Correct Hits (out of 1000)	Indices Range (CP: 896-960)	Length of this range	Length at 99% efficiency	max n-tap channel possible
0.01		1000	955-985	31	28	36
0.05		1000	956-984	29	28	36
0.07		1000	955-984	30	28	36
0.1		1000	955-984	30	28	36
0.2		1000	954-986	33	31	33
0.3		1000	954-985	32	30	34
0.4		1000	953-984	32	30	34
0.5		1000	954-984	31	30	34
0.6		1000	953-984	32	30	34
0.7		999	955-984	30	30	34
0.8		999	954-987	34	30	34
0.9		1000	953-985	33	31	33

Table2.3: Tabulations of Double Schmidl-Cox algorithm on a system with 0dB SNR noise

This range increases to over 38-40 at -3dB SNR. So this is the maximum noise that can be tolerated and still managing to identify the timing recovery with more than 95% efficiency. When the noise is further increased, i.e., at -6dB, it is 81% efficient and at -9dB, it is close to 68% efficient.

Frequency		Correct Hits (out of 1000)	Indices Range (CP: 896-960)	Length of this range	Length at 99% efficiency	max n-tap channel possible
0.01		1000	947-985	39	36	28
0.05		1000	946-985	40	38	26
0.07		1000	941-983	43	40	24
0.1		1000	943-985	43	39	25
0.2		1000	944-985	42	41	23
0.3		1000	944-985	42	40	24
0.4		1000	943-983	41	39	25
0.5		1000	944-985	42	40	24
0.6		1000	942-986	44	43	21
0.7		999	945-984	40	38	26
0.8		999	944-985	42	40	24
0.9		1000	944-986	43	41	23

Table 2.4: Tabulations of Double Schmidl-Cox algorithm on a system with -3dB SNR noise

CHAPTER 3

FREQUENCY ESTIMATION ALGORITHM

3.1 Schmidl-Cox frequency estimation algorithm:

Once the timing is realized, the frequency shift in the system must be estimated. The preamble with multiple repetitions again comes into use for this purpose.

When the lobes/signals that are identical are passed and collected at the receiver, they remain identical except that there will be a phase shift. So the phase of the correlation values on the plateau must give this phase difference. If the conjugate of a sample from the one quarter (half) is multiplied by the sample in another quarter (half), the product will have a phase that is directly proportional to the frequency shift.

$$Phase\ Difference \qquad \emptyset = 2\pi\Delta f \frac{T}{4} \qquad (3.1)$$

This phase difference will be of the form $(2z\pi+\alpha)$, where 'α' is the fractional part and 'n' the integer part. 'α' can be determined directly i.e., it is just the argument of '∅' which follows the condition $-\pi<\emptyset<\pi$.

The correlation can be carried out across three repetitions, so as to average out more noise and to better estimate the fractional part. Let 'm' be the timing recovery.

$$P = \sum_{n=0}^{\frac{3*N}{4}-1} r_{m-n} r^*_{m-n-N/4} \qquad (3.2)$$

$$\alpha = arg(P) \qquad (3.3)$$

The symbol received can be corrected with the fractional part of frequency offset, where

$$\overline{\Delta f} = \frac{2\alpha}{\pi T} \qquad (3.4)$$

The received samples are thus to be multiplied by $e^{-j2\pi\overline{\Delta f}t}$ for the correction. Now the integer part can be estimated using two preambles. The first preamble has four repetitions, i.e., it has a PN sequence of QPSK symbols at every fourth sample in the frequency domain. The second symbol has a PN sequence of QPSK symbols at all samples in frequency domain, but with half the amplitude, so that both the preamble symbols have the same average power.

Let 'x_1' be the PN sequence of '4p+1' sample frequencies of first preamble
Let 'x_2' be the PN sequence of '4p+1' sample frequencies of second preamble

Now let 'v' be the relative PN sequence, where

$$v(k) = \frac{x_1(k)}{x_2(k)} \qquad (3.5)$$

Integer part of phase difference implies that the sub-carriers are still orthogonal to each other but the frequency positions itself are shifted. So, at the receiver end, the frequency spectrum shows cyclically versions of the original PN sequences. This shift is to be measured.

Let 'y_1' be the '4p+1' sample frequencies FFT of the corrected first preamble
Let 'y_2' be the '4p+1' sample frequencies FFT of the corrected second preamble

The sequence 'v' appears at the output but shifted by '$4g$' positions because of the uncompensated frequency shift. So the expression

$$B(g) = \sum_{k=0}^{N/4} y_1(k+4g)v^*(k){y_2}^*(k+4g) \qquad (3.6)$$

This attains the peak value when the value 'g' coincides with the actual integer shift of the system.

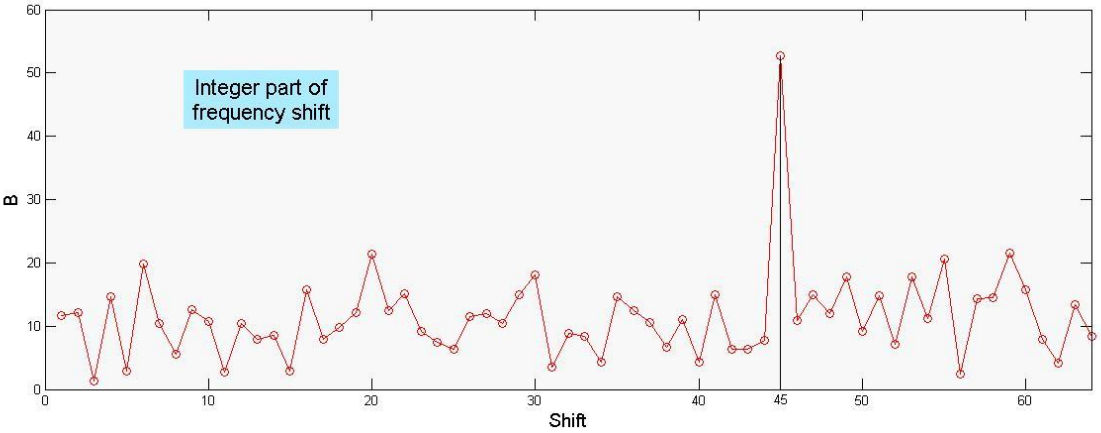


Figure 3.1: Graph depicting the integer part of relative frequency shift

The total frequency shift is just the sum of the fractional and integer part.

$$frequency\ Shift \qquad \Delta f = \overline{\Delta f} + \frac{4g}{T} \qquad (3.7)$$

This method is very efficient, the integer part gives correct results always even at negative SNR values of noise. But the fractional part has significant error because of the noise not being averaged out properly. For a 256-point FFT, maximum fractional offset is 4/256=0.015625.

Relative Fractional		Error (%)	Error (%)
		0 dB	**-3 dB**
0.1		9.07	21.87
0.2		9.43	20.76
0.3		9.57	21.43
0.4		9.41	20.65
0.5		9.36	20.60
0.6		9.76	20.24
0.7		9.62	21.61
0.8		9.60	21.01
0.9		9.34	22.17

Table 3.1: Results of Schmidl-cox Frequency Estimation algorithm carried out in a system with an AWGN noise of 0dB and -3dB SNR

3.2 Improvisation using multiple repetitions:

To increase the efficiency, the multiple repetitions in the preamble can be exploited. The other correlations are calculated and their phases are taken into consideration.

$$P_1 = \sum_{n=0}^{\frac{2N}{4}-1} r_{m-n} r^*{}_{m-n-N/2} \tag{3.8}$$

$$P_2 = \sum_{n=0}^{\frac{N}{4}-1} r_{m-n} r^*{}_{m-n-\frac{3N}{4}} \tag{3.9}$$

The phase difference can be corrected to

$$\alpha = \begin{cases} \arg(P) + \dfrac{\arg(P_1)}{2} + \dfrac{\arg(P_2)}{3}; & 0 \leqslant \arg(P) \leqslant \dfrac{\pi}{3} \\ \arg(P) + \dfrac{\arg(P_1)}{2} + \dfrac{\arg(P_2) \pm 2\pi}{3}; & \pm\dfrac{\pi}{3} \leqslant \arg(P) \leqslant \pm\dfrac{\pi}{2} \\ \arg(P) + \dfrac{\arg(P_1) \pm 2\pi}{2} + \dfrac{\arg(P_2) \pm 2\pi}{3}; & \pm\dfrac{\pi}{2} \leqslant \arg(P) \leqslant \pm\dfrac{2\pi}{3} \\ \arg(P) + \dfrac{\arg(P_1) \pm 2\pi}{2} + \dfrac{\arg(P_2) \pm 4\pi}{3}; & \pm\dfrac{2\pi}{3} \leqslant \arg(P) \leqslant \pm\pi \end{cases} \tag{3.10}$$

Relative Fractional		Error (%)	Error (%)
		0 dB	**-3 dB**
0.1		1.06	5.21
0.2		1.76	4.76
0.3		1.51	5.10
0.4		2.07	5.41
0.5		1.94	4.92
0.6		1.91	5.04
0.7		2.01	4.84
0.8		1.78	4.61
0.9		1.84	4.73

Table 3.2: Results of improvised Schmidl-Cox Frequency estimation algorithm

3.3 Multiple Preambles:

By observation it can be found that, the error is both positive and negative with equal probability. The system can be made better by using multiple preambles and just averaging out the fractional parts. This helps in nullifying the noise to the maximum extent and the error is close to zero even at 0dB SNR noise.

Relative Fractional		Error (%)	Error (%)
		0dB SNR	**-3dB SNR**
0.1		0.04	0.3
0.2		0.02	0.1
0.3		0.03	0.4
0.4		0.04	0.4
0.5		0.01	0.7
0.6		0.03	0.5
0.7		0.04	0.6
0.8		0.04	0.4
0.9		0.04	0.4

Table 3.3: Results of improvised Schmidl-Cox algorithm along with using five preambles each spaced 20 symbols apart

These are astounding results and this completes the whole topic of determining the effects of timing and frequency offsets, finding the timing synchronization and evaluating the frequency shift eventually.

CHAPTER 4

RESULTS AND CONCLUSION

4.1 Timing Synchronization algorithm:

The analysis has been started with implementation of the existing algorithms and finding their limitations with respect to the worsening of the channel conditions like high noise or high frequency shift. CP correlation method has been dismissed off owing to its unrepairable drawbacks (period of convolution being very less, i.e., CP length).

So Schmidl-Cox scheme has been executed and it was found to be around 80% efficient at 6dB SNR AWGN noise. The number of repetitions in the standard Schmidl-Cox algorithm were just two, but our simulations have been carried out using a preamble with four repetitions. This change has triggered a scope for improvement in the existing mechanism. Instead of getting confined to convoluting a signal with the conjugate of its adjacent lobe, all possible combinations can be considered, i.e., sum of the absolute values of each of these correlations is taken as the final correlation value to decide upon the timing synchronization.

But the very notion of determining the end point on the plateau turns meaningless with the inclusion of noise. So on the dire need to determine something more specific, it was noted that the plateau itself can act as a preamble with multiple repetitions. A second correlation was done on the Schmidl-Cox correlation curve with window length of half the preamble length. This was a clear hit. A little more careful examination has showed that the error is positive or negative with equal probability, so multiple preambles are taken and their respective correlation curves are added to get a curve, which was studied through certain mechanisms. This final improvised algorithm has shown 100% efficiency.

Frequency Shift ($\Delta f/fs$)	CP Correlation	Schmidl-cox	Multiple Correlations	Double Schmidl-Cox	Final algorithm
0.01	63	72	82	90	100
0.05	62	69	81	91	100
0.1	61	76	80	87	100
0.3	64	74	83	92	100
0.4	63	72	81	90	100
0.5	63	70	82	92	100
0.6	61	71	80	92	100
0.7	62	72	82	92	100
0.8	59	75	77	91	100
0.9	58	71	79	90	100

Table 4.1: Comparing the efficiencies of the algorithms based on each improvement

It can be clearly seen from the graph about how the performance is improved by making each correction in the algorithm followed.

4.2 Frequency Estimation Algorithm:

There is a standard algorithm by Schmidl-Cox for frequency estimation. The frequency shift essentially has two parts: Fractional offset and Integer offset. The integer offset estimation mechanism is very efficient. Even the slightest error in this integer offset will affect the performance badly. So we cannot afford to have any error here and the algorithm itself works fine, without any need for improvising, even at -6dB SNR AWGN noise. But the fractional offset has significant error at high noise levels. To overcome this, we steered to the previous improvements made in the timing synchronization algorithms. The multiple correlations have been considered, and mean of all these values is taken. An observation made was that the error here is equi-probably positive or negative. So, taking multiple preambles and averaging out the frequency estimated from all of these will produce better results. The results are totally compatible with the expected shift, with a maximum error of just 0.1% at 0 dB SNR.

Relative Fractional Frequency Shift (=$\overline{\Delta f}$/max $\overline{\Delta f}$)		Schmidl-Cox	Multiple Correlations	Multiple Preambles
0.1		9.07	1.06	0.04
0.2		9.43	1.76	0.02
0.3		9.57	1.51	0.03
0.4		9.41	2.07	0.04
0.5		9.36	1.94	0.01
0.6		9.76	1.91	0.03
0.7		9.62	2.01	0.04
0.8		9.60	1.78	0.04
0.9		9.34	1.84	0.04

Table 4.2: Comparing the Error Percentage in each of the algorithms with a channel noise of 0 dB SNR

Relative Fractional Frequency Shift (=$\overline{\Delta f}$/max $\overline{\Delta f}$)		Schmidl-Cox	Multiple Correlations	Multiple Preambles
0.1		21.87	5.21	0.3
0.2		20.76	4.76	0.1
0.3		21.43	5.10	0.4
0.4		20.65	5.41	0.4
0.5		20.60	4.92	0.7
0.6		20.24	5.04	0.5
0.7		21.61	4.84	0.6
0.8		21.01	4.61	0.4
0.9		22.17	4.73	0.4

Table 4.3: Comparing the Error Percentage in each of the algorithms with a channel noise of -3 dB SNR

APPENDIX

A.1 Duplicating Signal Structure:

```
clear all
clc
close all
N=256; % OFDM system of 256-point FFT
Ncp=N/4; % Cyclic Prefix length is one-fourth of the window length
Ns=N+Ncp; % Sampling frequency
Nrep=5; % number of time-frame repititons considered
Nf=20; % number of symbols between two such preambles
Ni=(4+(Nrep-1)*Nf+Nrep)*Ns;

SNR=[-2 0 2 4 6 8 10 12 14];
doppler=[0.01 0.05 0.07 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9];

count=zeros(12,3000); % counting the timing syncs

for dplr=1:12
   for p=1:500

      % QPSK symbols used

      QAM=[0.707+0.707i -0.707+0.707i 0.707-0.707i -0.707-0.707i];
      buf=QAM(randint(N/4,Nrep,4)+1);

      % preambles used
      a=zeros(N,Nrep);
      index=1;
      for n=1:4:N
         a(n,:)=buf(index,:);
         index=index+1;
      end

      a(N/2-1,:)=0;
      a(N/2,:)=0;
      a(N/2+1,:)=0;
```

```
a(N/2+2,:)=0;

% random signals
b=ifft(a);

random=0.5*QAM(randint(N,((Nrep-1)*Nf+4),4)+1);

random(N/2-1,:)=0;
random(N/2,:)=0;
random(N/2+1,:)=0;
random(N/2+2,:)=0;

rand=ifft(random);

sample=zeros(N,((Nrep-1)*Nf+4)+Nrep);

sample(:,1:2)=rand(:,1:2);
sample(:,end-1:end)=rand(:,end-1:end);
index1=3;
index2=3;
for index3=1:(Nrep-1)
   sample(:,index1)=b(:,index3);
   index1=index1+1;
   for j=1:Nf
      sample(:,index1)=rand(:,index2);
      index1=index1+1;
      index2=index2+1;
   end
end
sample(:,index1)=b(:,Nrep);


%adding Cyclic-Prefixes
signal=[sample(N-Ncp+1:N,:);sample];
signal1=reshape(signal,1,size(signal,1)*size(signal,2));

% channel impulse response
channel=zeros(1,Ncp);
for i=0:((Ncp/2)-1)
   channel(i+1)=exp(-i);
end

% signal passing through channel
signal2=convn(signal1,channel,'full');
```

```
% signal affected by frequency shift
for index4=1:size(signal,1)*size(signal,2)
    signal3(index4)=signal1(index4)*(exp(1i*2*pi*doppler(dplr)*index4));
end

% adding AWGN noise
sig=awgn(signal3,0,'measured');
```

A.2 Timing Synchronization algorithm

```
P=zeros(1,Ni);
P1=zeros(1,Ni);
P2=zeros(1,Ni);
P3=zeros(1,Ni);
S=zeros(1,Ni);
T=zeros(1,Ni);

% Schmidl-Cox correlation
for i=Ns:Ni
    % correlation of adjacent lobes
    for j=0:(0.75*N-1)
        P1(i)=P1(i)+conj(sig(i-j))*sig(i-j-N/4);
    end
    % correlation of alternate lobes
    for j=0:(0.5*N-1)
        P2(i)=P2(i)+conj(sig(i-j))*sig(i-j-N/2);
    end
    % correlation of lobes spaced two lengths apart
    for j=0:(0.25*N-1)
        P3(i)=P3(i)+conj(sig(i-j))*sig(i-j-0.75*N);
    end

    % cumulative correlation
    P(i)=abs(P1(i))+abs(P2(i))+abs(P3(i));
end

% Double Schmidl-Cox correlation
for i=Ncp:Ni
    for k=0:((Ncp/2)-1)
        S(i)=S(i)+conj(P(i-k))*P(i-k-Ncp/2);
    end
```

```
end

% adding multiple preambles
T=abs(S);
for i=1:(Nrep-1)
   T=T+abs(circshift(S,[0 -(((Nf+1)*Ns)*i)]));
end

U=abs(T(1:5*Ns));

% determining the timing recovery
maxim=max(U);
[l1 m1]=find(U==maxim);
V=U(1:m1(end));
W=U(m1(end):end);
[l2 m2]=find(abs(V-maxim*0.25)<=0.3);
[l3 m3]=find(abs(W-maxim*0.15)<=0.3);

m= ceil(0.5*(m2(end)+m3(1)+m1(end)));


count(dplr,m)=count(dplr,m)+1;
```

A.3 Frequency Estimation algorithm:

```
P1=0;
P2=0;
P3=0;

% timing correction
m=m-32;

% multiple correlations
for j=0:(0.75*N-1)
      P1=P1+(sig(m-j))*conj(sig(m-j-N/4));
end
for j=0:(0.5*N-1)
      P2=P2+conj(sig(m-j))*sig(m-j-N/2);
end
for j=0:(0.25*N-1)
      P3=P3+conj(sig(m-j))*sig(m-j-0.75*N);
end
```

```
        % FFT of transmitted preambles
        x1=fft(signal1(2*Ns+Ncp+1:3*Ns));
        x2=fft(signal1(3*Ns+Ncp+1:4*Ns));
        y=x1.*conj(x2);

        % fractional frequency offset
        f1=2*angle(P1)/(pi*N);

        % correcting signal with the frequency offset
        for index8=1:Ni
            sig1(index8)=sig(index8)*exp(-1i*2*pi*f1*index8);
        end

        % FFT of received signal
        xx1=fft(sig1(m-N+1:m));
        xx2=fft(sig1(m+Ncp+1:m+Ns));
        yy=xx1.*conj(xx2);

        % Determining integer frequency offset
        num=zeros(1,N);
        B=zeros(1,N);

        for g=0:N/4-1
            for index9=1:4:N
            num(g)=num(g)+conj(y(index9))*(yy(rem(index9+4*g,N)));
            end
            B(g)=abs(num(g));
        end

        [int1 int]=find(B==max(B));

        % Final Frequency Shift estimated
        freq(dplr,p)=(4*int/N)+f1;

    end
end
```

REFERENCES

[1] Robust Frequency and Timing Synchronization on OFDM
(Timothy M. Schmidl and Donald C. Cox)

[2] Iterative Symbol Offset Correction Algorithm for Coherently Modulated OFDM Systems in Wireless Communication
(V.S. Abhayawardhana, I.J. Wassell)

[3] Low-Overhead, Low-Complexity [Burst] Synchronization for OFDM
(Timothy M. Schmidl and Donald C. Cox)