國 立 交 通 大 學

電子工程學系 電子研究所碩士班

碩 士 論 文

無線都會網路之下行取樣頻率同步設計

Design of Sampling Clock Synchronization for WMAN Downlink

研究生：林運翔

指導教授：周世傑 博士

中 華 民 國 九 十 九 年 三 月

無線都會網路之下行取樣頻率同步設計

Design of Sampling Clock Synchronization for WMAN Downlink

研究生：林運翔　　　　　　　Student：Yuan-Xiang Lin

指導教授：周世傑 博士　　　　Advisor：Dr. Shyh-Jye Jou

國立交通大學
電子工程學系 電子研究所碩士班
碩士論文

A Thesis
Submitted to Department of Electronics Engineering &
Institute of Electronics
College of Electrical and Computer Engineering
National Chiao Tung University
in Partial Fulfillment of the Requirements
for the Degree of Master of Science
In
Electronics Engineering
January 2010
Hsinchu, Taiwan, Republic of China

中華民國九十九年三月

摘要

論文名稱：無線都會網路之下行取樣頻率同步設計　　　　頁數：54 頁

校所組別：國立交通大學 電子工程研究所 系統組

研究生：林運翔 指導教授：周世傑 博士

論文提要內容：

IEEE 802.16e 為高速無線傳輸規格，其所採用的通訊技術為正交分頻多工（OFDM）與多輸入多輸出(MIMO)。正交分頻多工是一種多載波的調變方式，將資料分散到各個不同的頻率且互相正交的次載波。多輸入多輸出是使用多天線傳送與接收的技術。在本論文裡，我們專注於 802.16e 取樣頻率漂移，並使用與 C 語言建立系統模擬平台。此平台包含載波同步，時序同步與頻域等化器。在正交分頻多工和多輸入多輸出情況之下，我們提出了幾種模式來說明取樣頻率飄移的現象，並提出 3 個架構來補償取樣頻率飄移。最後，此論文提出有效率使用記憶體之取樣頻率飄移補償之架構。此架構可以減少記憶體使用量達 85%，並能將頻率飄移在 20 ppm 以下的情況完全補償回來，而在 40 ppm 時所增加的誤差會小於 0.01 位元錯誤率。相對於未補償的情況在 20 ppm 與 40 ppm 時分別是 0.15 位元錯誤與 0.36 位元錯誤率，本論文提出之有效率使用記憶體之取樣頻率飄移補償架構大幅改進系統錯誤率。

Abstract

IEEE 802.16e is a recently proposed standard for high speed wireless transmission. Orthogonal Frequency Division Multiplexing (OFDM) and Multiple-input Multiple-output (MIMO) technical is adopted in 802.16e. OFDM is a kind of multi-carrier modulation and data are divided into several differential and orthogonal subcarriers. MIMO is a kind of transmission way with multiple transmitting antennas and multiple receiving antennas. In this thesis, we focus on sampling clock offset (SCO) of 802.16e and build a simulation platform using C language. This platform contains carrier frequency synchronization, timing synchronization and frequency domain channel estimation, and we simulate SCO effect on this platform. We propose two kinds of model to illustrate SCO effects proposes three architectures to compensate SCO on OFDM and MIMO system. Finally, we propose a memory-efficiency architecture for sampling clock offset compensation. This architecture reduces the usage of memory by 85 %. This architecture can perfectly compensate SCO effect when SCO value is below 20 ppm, and make the BER less than 0.01 when SCO value is 40 ppm. However, without compensation, under 20 and 40 ppm SCO, the BER is 0.15 and 0.36 respectively. Thus, the proposed memory-efficiency architecture for sampling clock offset compensation improves the BER of the system significantly.

誌　謝

首先要感謝 周世傑老師的細心指導，使我在研究所的生涯中得到很多。在研究上的小細節，也因為老師的指導知道很多應該注意的地方。另外要謝謝我的家人在生活上和精神上的支持和鼓勵。最後謝謝實驗室的學長姊和同學的教導。

運翔

謹誌於 新竹

2010 年 3 月

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Overview of Wireless Systems and 802.16e

Wireless communication systems are prevailing everywhere recently and their function extends from basic voice services to advanced picture, video, and data service. Thus, the requirement of high speed wireless transmission is growing quickly.

IEEE 802.16 Working Group is for Broadband Wireless Access (BWA) established in 1998, which is responsible to the development of IEEE 802.16 standard. The original 802.16 standard was completed in December 2001 [1]. It was based on the fixed light-of-sight (LOS) operation in the 10 GHz-66 GHz range. In the physical layer (PHY), only single carrier modulation is supported. IEEE 802.16a [1] standard is an amendment to 802.16 in 2003. It provides the capacities of point-to-multipoint connection and non-line-of-sight (NLOS) transmission in the 2GHz-11GHz frequency band. Three modulation schemes of PHY are supported: SC, Orthogonal Frequency Division Multiplexing (OFDM), and Orthogonal Frequency Division Multiple Access (OFDMA).

IEEE 802.16-2004, also known as IEEE 802.16d, is only for fixed broadband wireless and has upgraded to all prior versions in June 2004 [1]. An industry consortium, the Worldwide Interoperability for Microwave Access (WiMAX) adopted IEEE 802.16-2004 as the first solution of fixed applications. IEEE 802.16e-2005 [2] amends 802.16-2004 to add mobility support. It enables mobile speed up to 120km/h, but also be backward compatible to support the fixed mode in 802.16-2004. Operation in mobile mode is limited to the licensed bands between 2GHz-6GHz, on the other hand, operation in fixed mode is limited to 2GHz-11GHz. IEEE 802.16e-2005 is usually referred to as mobile WiMAX. In this thesis, we will focus on this standard.

OFDM attracts much attention in that it can combat frequency-selective fading channels, which is a severe challenge especially in wideband system. Another popular way to improve transmission ability is using multiple antennas, which are now widely termed as the multiple inputs and multiple outputs (MIMO) system. Space-time block codes (STBC) [6] is one of the types which MIMO systems may be implemented to provide spatial diversity and thus improve power efficiency.

In OFDM and MIMO transmission systems, there are several steps to reconstruct signals in a receiver. In the digital domain, the first step will be synchronization, which means to extract the signal timing information so that a receiver can get the correct signal. The second step will be channel estimation and compensation to recover the distortion from the channel between a transmitter and a receiver. The third step will be de-mapping which is used to recognize the point in constellation. In synchronization, sampling clock offset (SCO) caused by the different sampling rate between a transmitter and a receiver, has a significant effect on the bit error rate (BER).

802.16e supports MIMO, STBC and OFDM and these features make the system increase the data rate and more spatial diversity in channels. However, on the other hand, the system is more complicated, and is more difficult to analyze the received data in digital domain. In wireless mobile systems, the challenge will be that the channel in a varying environment will fade, making SCO very difficult to estimate.

This thesis is mainly to demonstrate the synchronization of OFDM and MIMO systems, and analyzes SCO effect in a MIMO OFDM system. Then, a new scheme to compensate SCO in 802.16e system will be proposed.

1.2 Motivation of SCO Compensation

The inaccuracy in oscillator may make the sampling frequency higher or lower in receiver than in the transmitter. Moreover, the specification of 802.16e is for high mobility. For a high mobile system, Doppler effect, which means the relation between a moving receiver and a moving transmitter will plays an important role. Doppler effects have the following consequence: not only the change of frequency but also the fade of multi-path channel. Moreover, both of these effects will change the sampling rate that we expected and introduce SCO. Therefore, SCO shall be carefully analyzed and find a way to reduce the effect by reasonable hardware architecture with less complicated design.

The effect of SCO is coupled with other effects such as channel noise, phase noise, etc. For the convenience of analysis, we separate the SCO effect into 2 parts – sampling timing difference in a symbol (STDS), which causes inter-channel interference (ICI) and fractional symbol timing offset (FSTO), which causes phase rotation. Both of these effects will be illustrated in Chapter 2. The way to compensate SCO can be either using analog circuit or using digital circuit, and Chapter 3 will discuss the proposed scheme in detail. In our design, we choose the compensation in digital circuit to reduce the cost of an accurate voltage control oscillator (VCO).

In this thesis, we propose two kinds of SCO estimation and compensation architecture: one is compensated in time domain, and the other one is compensated in frequency domain. The ideal time domain compensation can eliminate ICI effect and phase rotation, whereas the ideal frequency domain compensation can only eliminate the effect of phase rotation. The ICI effect and phase rotation effect caused by SCO will be introduced in Chapter 2. In Chapter 3, we will find that the effect of phase rotation is much more important than the effect of ICI, so the effects of the two compensation ways are almost the same. However, the hardware cost of time domain compensation is much more than that of frequency domain compensation. These two ways of compensation will be discussed detailed in Chapter 4. Further more, the way to compensate in frequency domain can be simplified to reduce the usage of memory. Thus, the hardware to compensate in frequency domain can be reduced to a very small architecture. All algorithms in frequency domain compensation need to compute absolute value and trigonometric functions. We use flexible architecture like Coordinate Rotation Digital Computer (CORDIC) to make hardware more reusable. A CORDIC-based rotator reduces the complex multiplier (four multipliers and two adders) by a conversational rotator. However, the realization in CORDIC may makes the loss of precision due to finite word length effect, and it can be compensated by a method that will be introduced in Chapter 4.

1.3 Thesis Organization

The rest of this thesis is organized as follows. In Chapter 2, the principles of OFDM will be introduced. The MIMO system in 802.16e and SCO effects will be introduced in Chapter2. In Chapter 3, SCO effects will be well analyzed and the way of estimating and compensating SCO will be illustrated. The architecture and hardware design is discussed in Chapter 4. Finally, Chapter 5 is our conclusions.

Chapter 2

Principles of MIMO OFDM and SCO Effects

2.1 Principles of OFDM

2.1.1 OFDM Technology Overview

OFDM used in 802.16e will be briefly introduced in this section. OFDM is based on the idea of frequency division multiplexing (FDM). The concept of using parallel data transmission was published in the mid 60s [8]. In FDM, the total frequency bandwidth is divided into *N* non-overlapping sub-channels which are modulated with a separate symbol. In order to prevent from the adjacent channel interference, frequency spacing is allocated between sub-channels. However, this is inefficient to use the spectrum. In OFDM, the total frequency bandwidth is divided into N overlapping sub-channels which are mutual orthogonal. The orthogonality allows simultaneous transmission of a lot of subcarriers without interference from each other. The sub-channels in FDM and OFDM are shown in Fig. 2.1.

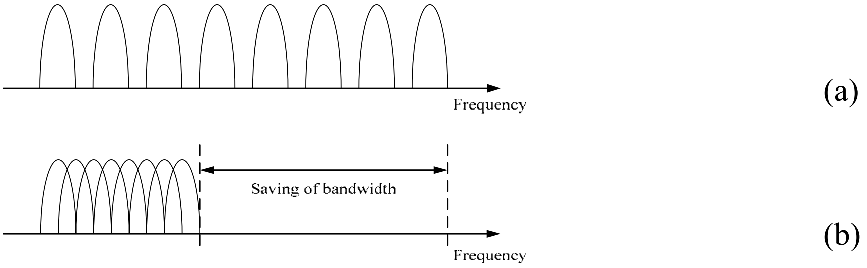


Fig. 2.1 Sub-channels and bandwidth in (a) FDM (b)OFDM

2.1.2 OFDM Specifications in 802.16e

In 802.16e standard, lots of different frequency sinusoidal wave with finite length will be stacked up in time domain as shown in Fig. 2.2. On the other hand, the wave in frequency domain will be stacked up with lots of Sinc wave, since the length is finite making the corresponding wave in frequency domain become a Sinc function. DC gain will be zero as shown in Fig. 2.3, because there is no DC component in time domain. In order to keep the orthogonality in frequency domain, the frequency of higher frequency sinusoidal wave will be a multiple of that of the lowest sinusoidal wave.

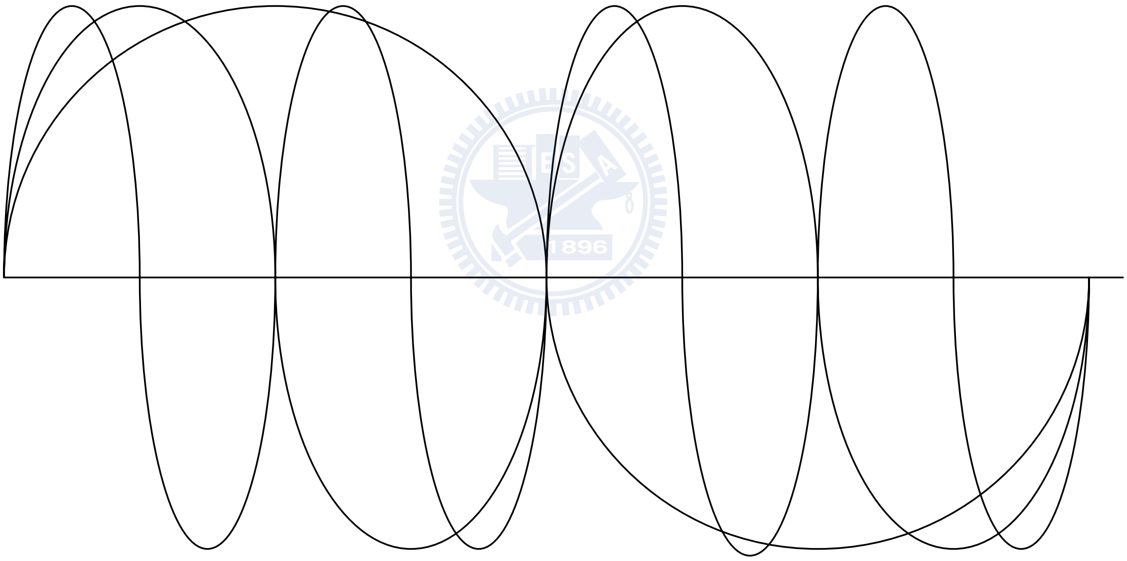


Fig. 2.2 Time domain representation of OFDM

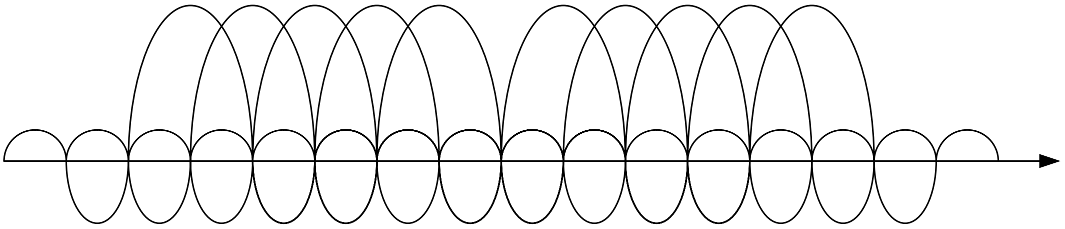


Fig. 2.3 Frequency domain representation of OFDM

In 802.16e standard [6], there are several processing units specified in the PHY layer of OFDMA. They include one-dimension domain units and two-dimension domain units because of data allocated on both time and frequency. An essential OFDMA symbol is based on the format of OFDM symbol and the most basic unit is "subcarrier". The "subcarrier" is the one-dimension unit and consists of three types as shown in Fig. 2.4 and Fig. 2.5:

1. Data subcarriers: for carrying data.
2. Pilot subcarriers: for the purpose of channel estimation, channel tracking and synchronization.
3. Null subcarriers: no power is allocated to the DC subcarrier and guard subcarriers. The frequency of DC subcarrier is equal to RF, and this will induce the local oscillator re-radiation effect such that this subcarrier is not suitable to carry data. The reason for guard band is to reduce the interference between adjacent channels.

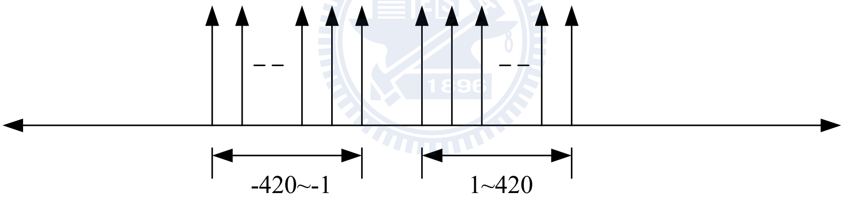


Fig. 2.4. Data subcarriers in 802.16e

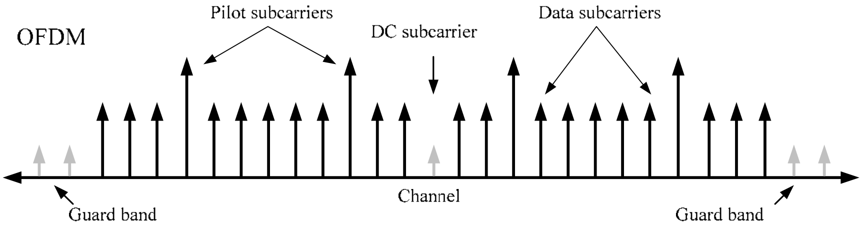


Fig. 2.5 Subcarriers in 802.16e

A set of subcarriers in frequency domain are grouped as a "subchannel". Two-dimension units from large number of subcarrier to small number of subcarrier are "frame", "sub-frame", "zone", "segment", "burst", "slot" and "cluster". Brief definitions are described as follows respectively and the relationship between these units is shown in Fig. 2.6:

1. Frame: It is an essential packet format of transmitted data sequence.
2. Sub-frame: It is a component to make up a frame and is identified as downlink and uplink.
3. Zone: A zone is the region of contiguous OFDMA symbols with the same data allocation method. It is allowed to have different zones in a sub-frame.
4. Segment: It is a subdivision of the set of subchannels for certain particular allocation zone. The content of the Medium Access Control (MAC) layer is the same in a segment.
5. Burst: It is a region which includes the contiguous subchannel and OFDMA symbol to transmit the broadcast or unique data for corresponding users.
6. Slot: It is the minimum possible data allocation unit and described in both time and subchannel dimension. It contains 48 data subcarriers for all subchannelization schemes, but their arrangement is different in different schemes
7. Cluster: It contains 14 adjacent subcarriers over 2 contiguous symbols with 4 pilot subcarriers in partial usage of subcarriers (PUSC) permutation scheme.

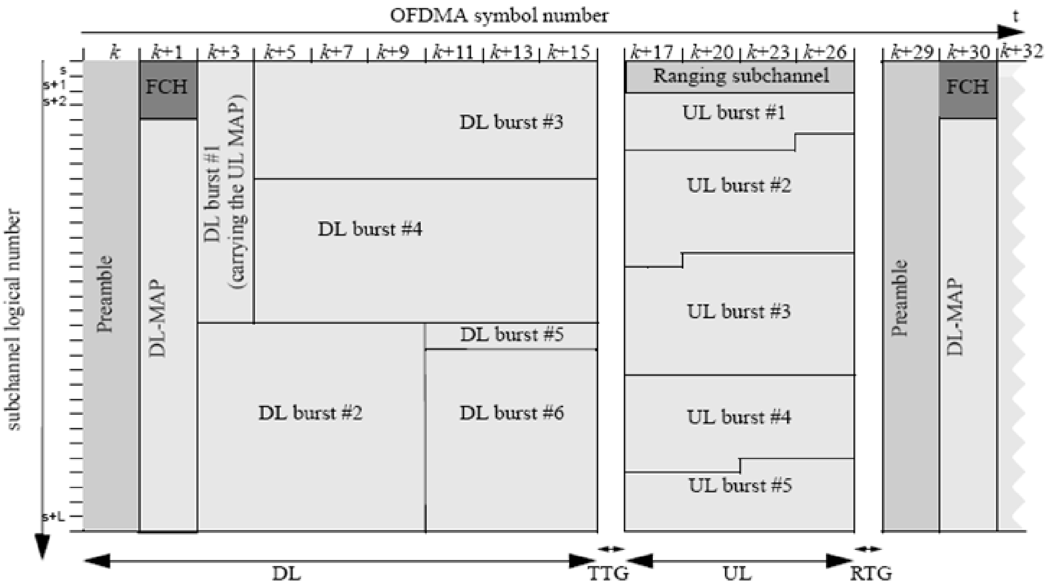


Fig. 2.6 Frame structure[2]

Table 2.1 shows the adopted specification of 802.16e. We choose the FFT size 1024 and QPSK mode for our system. Thus we follow the specification of 802.16e with FFT size 1024. In the specification, there are 1024 subcarriers in a symbol. In 1024 subcarriers, there are 720 subcarriers for data, 120 for pilots, and 184 for guard band. The guard time for a symbol is 11.4 us, and useful symbol time is 91.4 us. Thus symbol duration is about 102.9 us. Each subcarrier requires 10.94 kHz bandwidth, and total system requires 10MHz bandwidth.

Table 2.1 Adopted 802.16e system specifications

<table>
<tr><td colspan="2">Parameters</td><td>Deriving formulas</td><td>Values</td></tr>
<tr><td colspan="3">FFT size (N_{FFT})</td><td>1024</td></tr>
<tr><td colspan="3">System channel bandwidth (B)</td><td>10 MHz</td></tr>
<tr><td colspan="2">Sampling factor (n)</td><td>n=28/25 if B is a multiple of 1.25、1.5、2、2.75 MHz
n=8/7 for the other cases</td><td>28/25</td></tr>
<tr><td colspan="2">Sampling frequency (F_s)</td><td>floor(n·BW/8000)×8000</td><td>11.2 MHz</td></tr>
<tr><td colspan="2">Subcarrier spacing (Δf)</td><td>F_s/ N_{FFT}</td><td>10.94 kHz</td></tr>
<tr><td colspan="2">Useful symbol time (T_b)</td><td>1/Δf</td><td>91.4 us</td></tr>
<tr><td colspan="2">Guard time (T_g)</td><td>G·T_b</td><td>11.4 us</td></tr>
<tr><td colspan="2">OFDMA symbol duration (T_s)</td><td>T_b+T_g</td><td>102.9 us</td></tr>
<tr><td colspan="3">Frame duration (T_F)</td><td>5 ms</td></tr>
<tr><td colspan="2">Number of OFDMA symbols (N)</td><td>floor(T_F/T_s)</td><td>48</td></tr>
<tr><td rowspan="5">DL PUSC</td><td colspan="2">Number of null subcarriers (N_n)</td><td>184</td></tr>
<tr><td>Number of clusters (N_c)</td><td>(N_{FFT}-N_n)/14</td><td>60</td></tr>
<tr><td>Number of subchannels (N_{sc})</td><td>N_c/2</td><td>30</td></tr>
<tr><td>Number of pilot subcarriers (N_p)</td><td>N_c×2</td><td>120</td></tr>
<tr><td>Number of data subcarriers (N_d)</td><td>N_c×12</td><td>720</td></tr>
<tr><td colspan="3">Modulation mode</td><td>QPSK</td></tr>
<tr><td colspan="3">Raw data rate</td><td>13.6 Mbps</td></tr>
<tr><td colspan="3">Coding rate</td><td>3/4 CC</td></tr>
<tr><td colspan="2">Peak data rate *</td><td>N_d×2×3/4×(N-2)×1/ T_F</td><td>9.93 Mbps</td></tr>
</table>

* Assuming 46 data OFDMA symbol in a frame

2.2 Principles of MIMO

2.2.1 Concepts of MIMO

The technology of MIMO is supported in WMAN, which provides several benefits as described below:

1. Increase the system reliability.
2. Increase the achievable data rate and enhance system capacity.
3. Increase the coverage area.
4. Decrease the required transmitting power.

For different application purposes, different MIMO schemes are adopted such as beamforming, spatial multiplexing and space-time code since these schemes usually conflict with one another. For example, increasing the data rate will decrease the reliability or increase the transmitted power.

1. Beamforming:

When multiple antennas are used in the closed-loop mode, the transmitter will know the channel state information. Thus, the interference caused from directional noise signals can be reduced by the technology of adaptive antenna systems, beamforming. A beamformer is similar to the spatial filter, which adjusts the strength of the transmitted and received signals based on direction as shown in Fig. 2.7. Two principles of beamforming, direction of arrival (DOA) based and eigenvalue beamforming based are generally used.

$$y(k)=\sum_{i=1}^{M} w_i^{*} x_i(k) = w^H x(k)$$

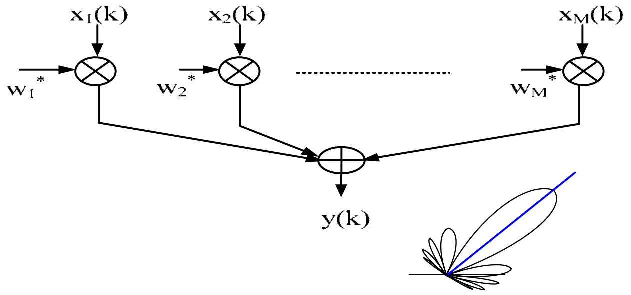


Fig. 2.7 Concepts of MIMO

2. Spatial multiplexing:

Spatial multiplexing is a technique to increase the throughput rate by using multiple antennas at both ends. The transmitted data stream is divided into N_t independent sub-streams. Multiple sub-streams are parallel transmitted through multiple antennas. If these data sub-streams can be separated successfully in the receiver, the data rate and system capacity will be higher than single antenna system.

3. Transmit diversity:

Transmit spatial diversity means that the transmitted signals can pass through different transmit antenna to overcome the deep fading channel. The transmit diversity is attractive for subscriber stations, because the cost of multiple antennas is on the transmitter. The space time block code (STBC), a transmit diversity technique, proposed by Alamouti in 1998 [6] is supported in WMAN. The case of 2*1 antenna system is described in Section 2.2.2 and is realized in our system.

2.2.2 STBC for MIMO

In the case of 2*1 antenna system, S_1 and S_2 are two consecutive symbols, and S_1^* and S_2^* mean complex conjugate of S_1 and S_2. h_1 is the channel response of antenna 0 to Rx, and h_2 is the channel response of antenna 1 to Rx. The receiver will receive data h_1S_1+ h_2S_2 for the first time slot, and -$h_1S_2^*$+ h_2 S_1^* for the second slot. The relation between transmitter and receiver is shown in Fig 2.8. The matrix in Fig. 2.8 is orthogonal in nature.

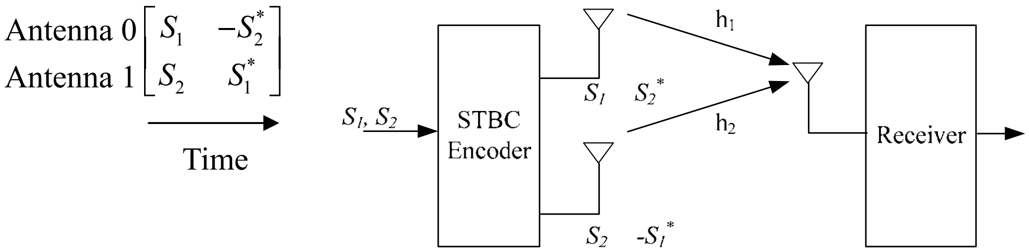


Fig. 2.8 Transmit diversity

2.2.3 Symbol Representation of STBC and Effects in 802.16e

In 802.16e, pilots play an important role for detection, such as symbol boundary, channel estimation, and SCO. However, the pilot representation in 802.16e has predefined rules, and by using STBC, the consequence will be more complicated.

The pilots in subcarriers are shown in Fig. 2.9, and the location of the pilots repeats every four symbols. In each symbol, the pilot location repeats every fourteen subcarriers, and different antennas and time slots make the location of the pilots different in four consecutive symbols as shown in Fig. 2.10. For example, antenna 0 transmits pilots in the (9+14*i*) subcarriers in the first symbol time; in the (5+14*i*) subcarriers in the second symbol time; in the (13+14*i*) subcarriers in the third symbol time; in the (1+14*i*) subcarriers in the forth symbol time. Antenna 1 transmits pilots in the (5+14*i*) subcarriers in the first symbol time; in the (9+14*i*) subcarriers in the second symbol time; in the (1+14*i*) subcarriers in the third symbol time; in the (13+14*i*) subcarriers in the forth symbol time. The receiver will receive signals which are the sum of antenna 0 and antenna 1. Therefore, the pilots in the receiver will be in the (5+14*i*) and the (9+14*i*) subcarriers in both of the first and the second symbol time; in the (1+14*i*) and the (13+14*i*) subcarriers in both of the third and the forth symbol time. This arrangement makes an important decision in the estimation for SCO, which will be introduced in Section 2.3.6 and Section 3.3.

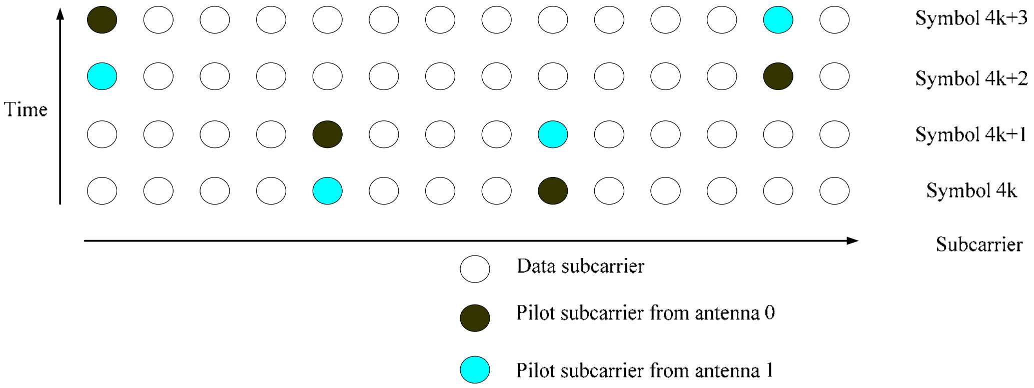


Fig. 2.9 Pilots in MIMO 802.16e

	Symbol 4*k*	Symbol 4*k*+1	Symbol 4*k*+2	Symbol 4*k*+3
antenna 0 (*h0*)	S_0 pilot: 5+14*i*	$-S_1^*$ pilot: 9+14*i*	S_2 pilot: 1+14*i*	$-S_3^*$ pilot: 13+14*i*
antenna 1 (*h1*)	S_1 pilot: 9+14*i*	S_0^* pilot: 5+14*i*	S_3 pilot: 13+14*i*	S_2^* pilot: 1+14*i*

rx_signal	S_0+S_1	$-S_1^*$+S_0^*	S_2+S_3	$-S_3^*$+S_2^*

Fig. 2.10 Format of STBC in 802.16e

2.3 Sampling Clock Offset (SCO) Effects

2.3.1 Synchronization in 802.16e

The synchronization in 802.16e mainly compensates the effect of symbol timing offset (STO) and carrier frequency offset (CFO) [5][6]. The following algorithm of estimation and compensation of SCO will use the concept of compensation of STO. The effect of symbol timing offset is introduced in Section 2.3.4.

2.3.2 Introduction of SCO

The effects of SCO is caused by the difference of transmitter sampling rate and receiver sampling rate, as shown in Fig. 2.11. This means the sampled data will be different between the transmitter and the receiver. In a system, with large SCO, there will be lots of errors in the received data. The way to estimate SCO and compensate SCO will become an important issue.

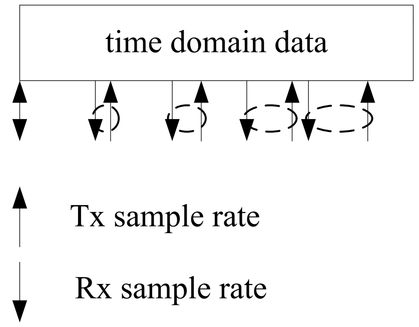


Fig. 2.11 The effect of different sample rate

2.3.3 SCO Effects on ICI and Phase Rotation in Frequency Domain

In fact, the effects of SCO are complicated to describe in frequency domain. For the convenience of analysis, we divide the SCO effects into two parts as shown in Fig. 2.12:

1. Fractional symbol timing offset (FSTO) cause phase rotation in frequency domain.
2. Sampling timing difference in a symbol (STDS) cause inter-channel interference (ICI) in frequency domain.

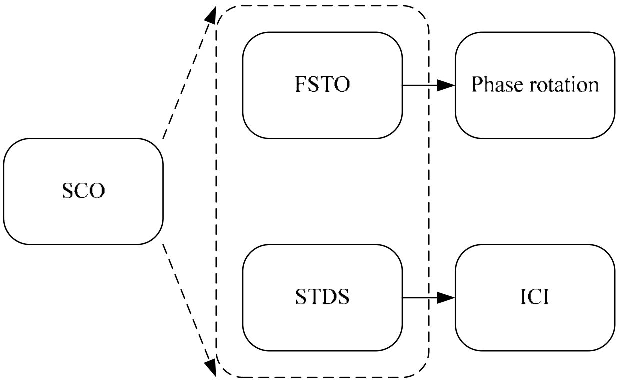


Fig. 2.12 Separations of SCO effects

In Fig. 2.13, we separate a general SCO effect in time domain as shown in Fig. 2.13(d), into the timing offset of the first sample time (Fig. 2.13(b)) and the difference of sampling rate while the first sampling time is the same (Fig. 2.13(c)) between Tx and Rx. The timing offset of the first sample in time domain for each symbol will cause FSTO. Since SCO will cause the difference of the sampling point in time domain, the timing offset of the first symbol and the second symbol will be different. This difference can be accumulated by the accumulation of SCO. The difference of sampling rate while the first sampling is the same will cause ICI.

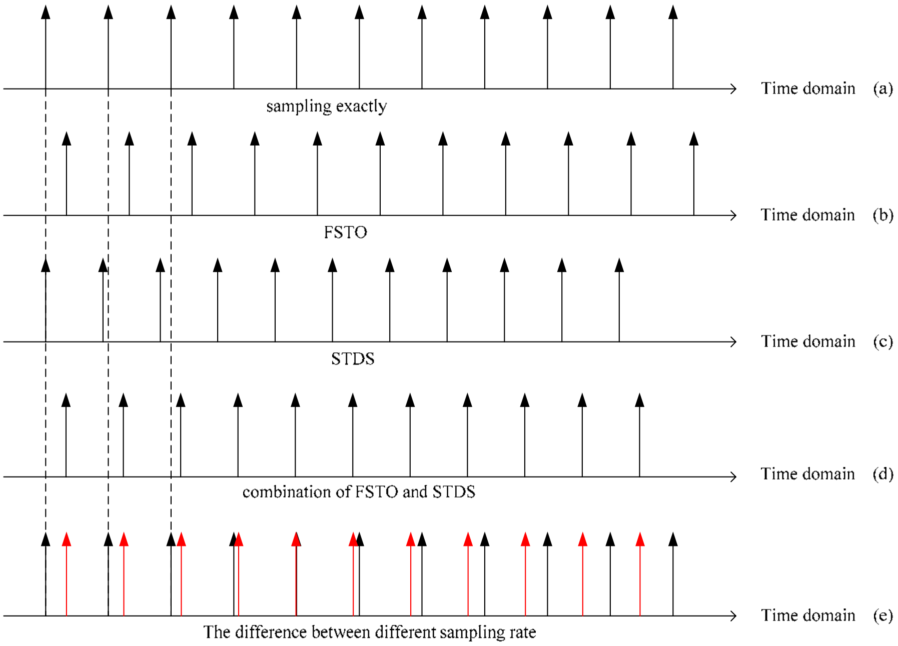


Fig. 2.13 FSTO and STDS illustration

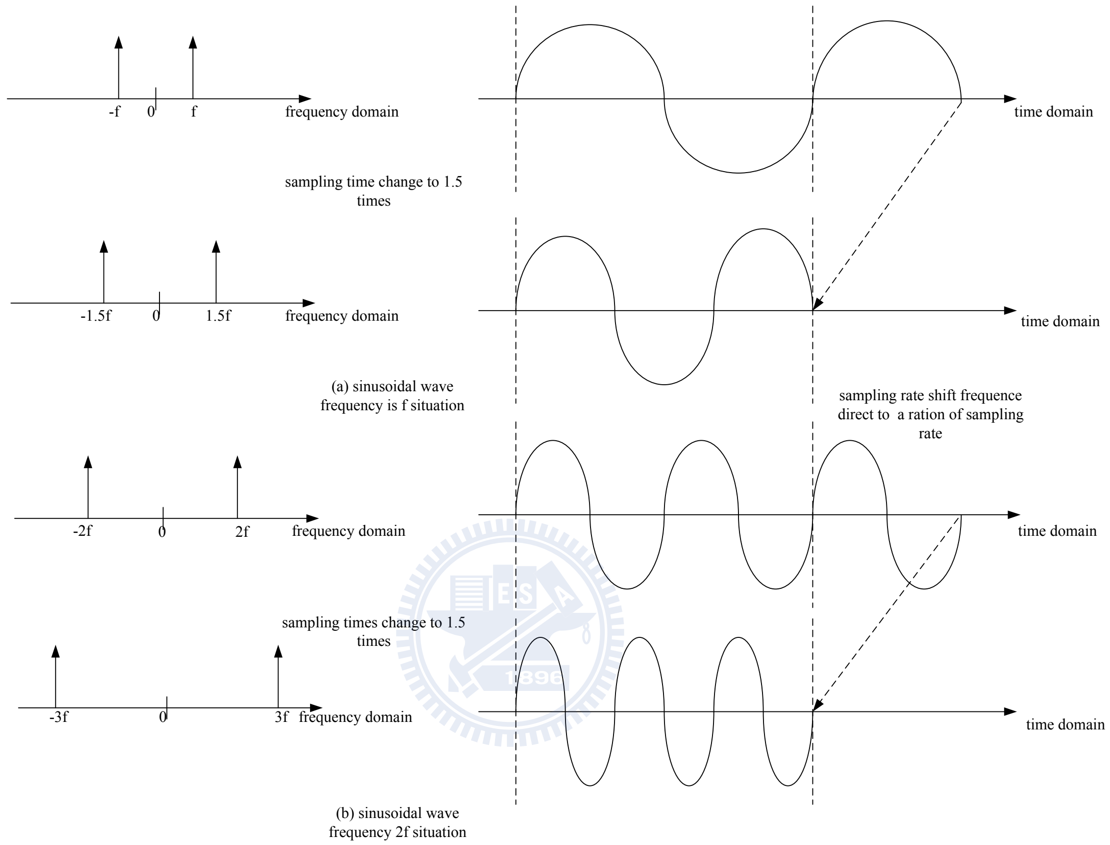
In Eqn. (2.1), x_1 is the first symbol from Tx, x_2 is the second symbol from Tx, x_3 is the third symbol from Tx, x_1' is the first symbol after sampling in Rx , x_2' is the first symbol after sampling in Rx , x_3' is the first symbol after sampling in Rx , n is the sample index in time domain, ε is FSTO, δ is SCO, and N is the number of subcarriers with guard interval, Eqn. (2.2) shows that the accumulation of δ will become the change of ε.

$$
\begin{aligned}
&x_1(n) = x(nT) \\
&x_2(n) = x((N+n)T) \\
&x_3(n) = x((2N+n)T) \\
&x_1'(n) = x'(nT) = x(\varepsilon_1 T + n(1+\delta)T) = (\varepsilon_1 T + n\delta\, T + nT) \\
&x_2'(n) = x'((N+n)T) = x(NT + \varepsilon_2 T + n(1+\delta)T) = x(\varepsilon_1 t + N(1+\delta)T + n(1+\delta)T) \\
&= x(\varepsilon_1 t + (N+n)\delta\, T + (N+n)T) \\
&x_3'(n) = x'((N*2+n)T) = x(2NT + \varepsilon_3 T + n(1+\delta)T) = x(\varepsilon_1 t + 2N(1+\delta)T + n(1+\delta)T) \\
&= x(\varepsilon_1 t + (2N+n)\delta\, T + (2N+n)T) \\
&\text{where} \quad \varepsilon_2 = \varepsilon_1 + N\delta \quad \text{and} \quad \varepsilon_3 = \varepsilon_1 + 2N\delta
\end{aligned}
\tag{2.1}
$$

2.3.4 The Illustration of ICI Caused by SCO Effect

ICI occurs when the first sample is correct but other samples are incorrect. This means ICI effect only affected by STDS δ as shown in Fig. 2.13(c).

In order to figure out this phenomenon and effect, we use a simple example as shown in Fig. 2.14 to illustrate how the sinusoidal wave behaves when sampling frequency change. In fact, OFDM system is the sum of different frequency and phase sinusoidal wave as shown in Fig. 2.2. The example shown in Fig. 2.14 can be easily extended to multiple carrier case as that in OFDM system. In Fig. 2.14, we can see that, if a multiple of sampling frequency change in continuous time domain, it can be thought as the boundary is changed so that there are more or less waves of the origin signal in the region. That is, the original sinusoidal wave becomes faster or slower in the region, and all different frequency sinusoidal wave will become faster or slower by the factor of change of sampling rate, since the change of waves follows this rate. So it will become a change of location change in frequency domain, and the change is the same as the change of sampling frequency. This means we can think it as the expansion or shrink by the multiplication in frequency domain as shown in Fig. 2.15.



-f
0
f
frequency domain
time domain
sampling time change to 1.5 times
-1.5f
0
1.5f
frequency domain
time domain
(a) sinusoidal wave frequency is f situation
sampling rate shift frequence direct to a ration of sampling rate
-2f
0
2f
frequency domain
time domain
sampling times change to 1.5 times
-3f
0
3f
frequency domain
time domain
(b) sinusoidal wave frequency 2f situation

Fig. 2.14 Frequency change illustration

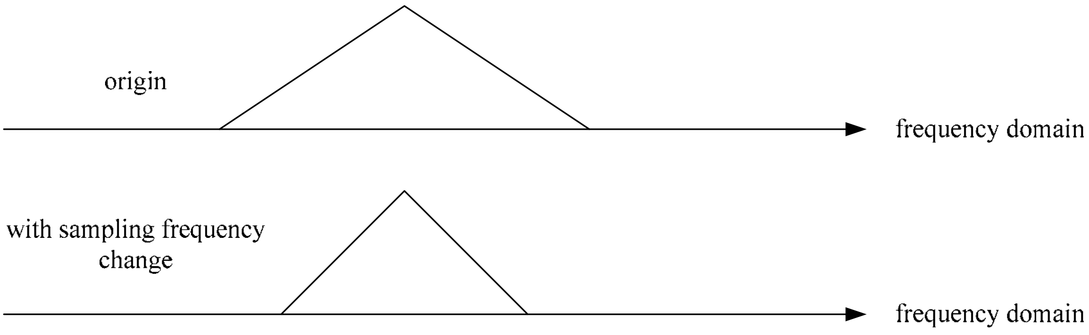


Fig. 2.15 Expansion or shrink in frequency domain

In our system the data are in the discrete domain. In order to figure out the relationship between digital domain and analog domain, the relations between Fourier

Transform and Fast Fourier Transform shall be mentioned. The relationship is shown in Fig. 2.16 [7].

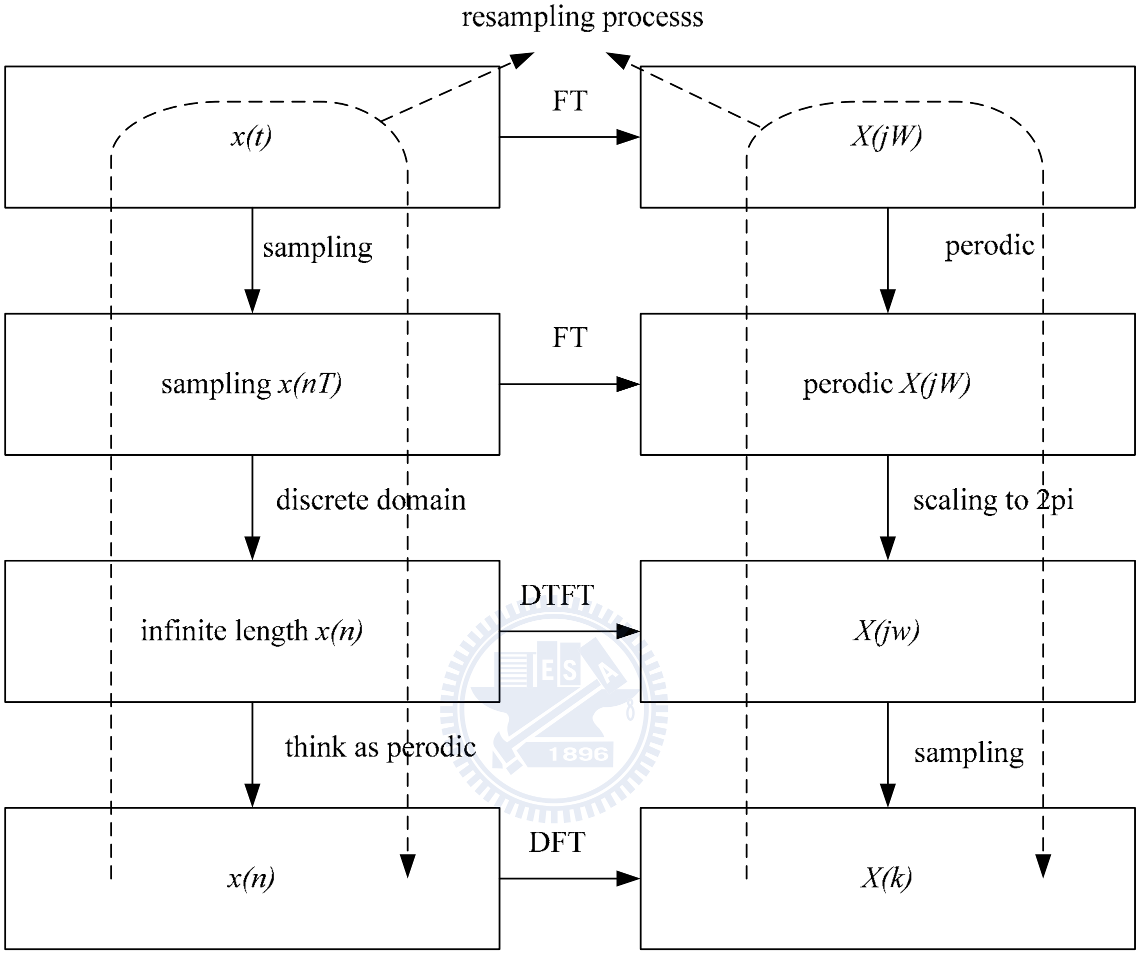


Fig. 2.16 Time and frequency domain relationship

In baseband, we represent the signal in *x(n)* or *X(k*). In conventional digital signal process, we can transform from time domain signal *x(n)* to frequency domain signal *X(k)* by Discrete Fourier Transform (DFT) and transform from frequency domain signal *X(k)* to frequency domain signal *x(n)* by inverse Discrete Fourier Transform (IDFT). If we need to resample a signal in discrete frequency domain, the basic idea to resample is to transform signal to time domain *x(n)*, reconstruct it to analog domain *x(t)*, resample it to digital time domain *x'(n)*, and then transform it to frequency domain *X'(k)*. However, the computation will become very large, if we need to simplify and show the process just in frequency domain, we just follow the processes in frequency domain corresponding to those in time domain. In time domain, copying multiple finite length data to infinite length data corresponds to using Sinc function to reconstruct in each tone of digital data in frequency domain. Making digital infinite length data becoming analog data in time domain corresponds to making the data in frequency domain periodically.

From the relationship discussed above, we know to explain the signal behavior in frequency domain and it just need 3 steps:

1. Use Sinc function to reconstruct continuous signal.
2. Make it periodically in frequency domain to expand from finite length to infinite length.
3. Resample it by the new sampling rate for the wanted length.

Fig. 2.17 shows a general condition of resampling in frequency domain.

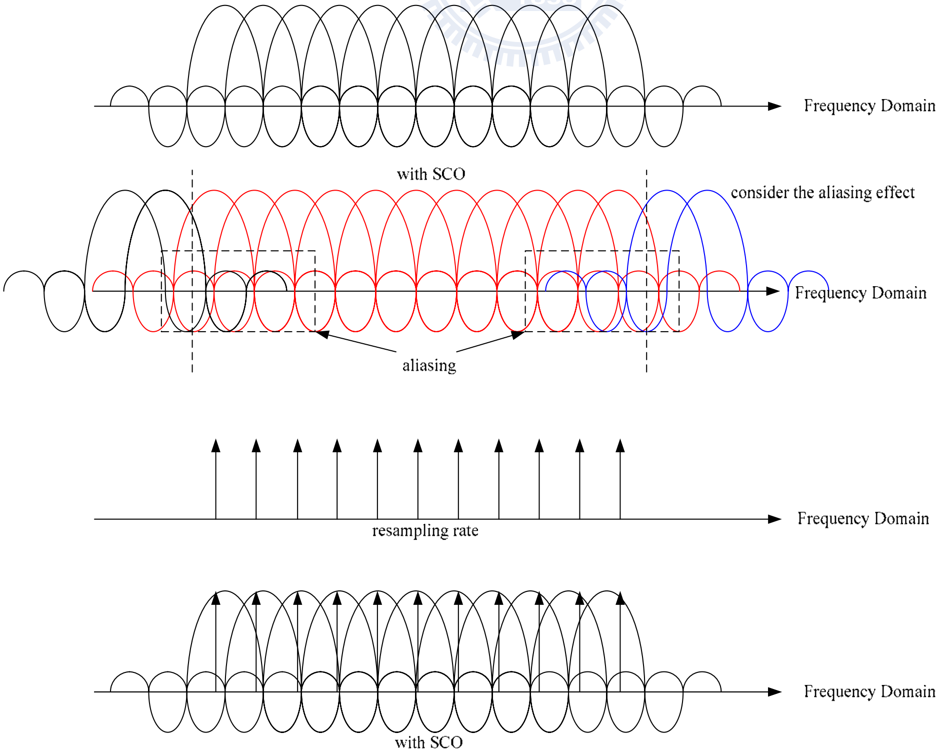


Fig. 2.17 ICI effect illustration

From Fig. 2.17, we know there will be aliasing on the higher frequency. However, in 802.16e the guard bands on higher frequency are all 0. So this effect can be neglected. So the effect is that the variant of sampling rate breaks the orthogonality in frequency domain. This causes ICI effect.

2.3.5 The Relationship between SCO and Symbol Boundary

In order to introduce the symbol boundary synchronization algorithm, the effect of symbol timing offset is derived in advance. STO means the estimated symbol boundary does not locate on the accurate location. It consists of two possible cases, earlier or later than the actual boundary index. If the estimated boundary is earlier than the ideal index but not locates at the channel response region, it means we choose the data in cyclic prefix (CP), which is the copy of final symbol data (Fig. 2.18); it induces the constellation of signals to rotate in the frequency domain, but can be compensated by channel estimation. On the other hand, the later case means we choose the data in the next symbol, and that means the irrelevant data is introduced, so the data is irrecoverable. In order to describe the earlier case, the received signal with this timing offset in mathematics is derived in Eqn. (2.2), where k is the subcarriers index, n is the sample index in time domain, and N_{sc} is the number of subcarriers in an OFDM symbol. If symbol timing offset is ε, then we have $\hat{X}(k) = X(k)\ \ e^{j2\pi k\varepsilon/Nsc}$ which means the phase rotation is proportional to subcarrier index k.

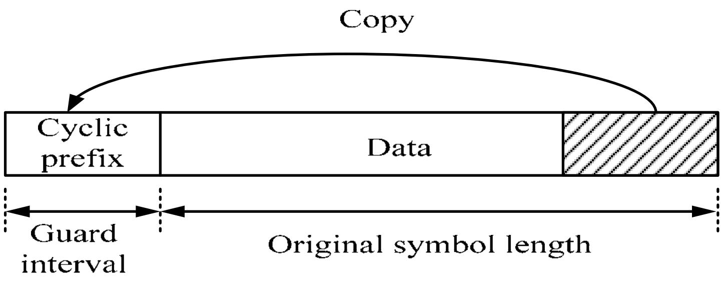


Fig. 2.18 OFDM symbol with cyclic prefix

$$
\begin{aligned}
\hat{X}(k) &= \sum_{n=0}^{N_{sc}-1} x(n-\varepsilon)e^{j2\pi k\frac{n}{N_{sc}}} \\
&= \sum_{n=0}^{\varepsilon-1} x(n-\varepsilon)e^{j2\pi k\frac{n}{N_{sc}}} + \sum_{n=\varepsilon}^{N_{sc}-1} x(n-\varepsilon)e^{j2\pi k\frac{n}{N_{sc}}} \\
&= \sum_{n=0}^{\varepsilon-1} x(n+N_{sc}-\varepsilon)e^{j2\pi k\frac{n+N_{sc}-\varepsilon}{N_{sc}}} e^{j2\pi k\frac{\varepsilon}{N_{sc}}} + \sum_{n=\varepsilon}^{N_{sc}-1} x(n-\varepsilon)e^{j2\pi k\frac{n-\varepsilon}{N_{sc}}} e^{j2\pi k\frac{\varepsilon}{N_{sc}}} \\
&= \sum_{m'=N_{sc}-\varepsilon}^{N_{sc}} x(m')e^{j2\pi k\frac{m'}{N_{sc}}} e^{j2\pi k\frac{\varepsilon}{N_{sc}}} + \sum_{m''=0}^{N_{sc}-\varepsilon-1} x(m'')e^{j2\pi k\frac{m''}{N_{sc}}} e^{j2\pi k\frac{\varepsilon}{N_{sc}}} \\
&= \sum_{m=0}^{N_{sc}} x(m)e^{j2\pi k\frac{m}{N_{sc}}} e^{j2\pi k\frac{\varepsilon}{N_{sc}}}
\end{aligned}
$$

$$\hat{X}(k) = X(k)e^{j2\pi k\frac{\varepsilon}{N_{sc}}} \qquad (2.2)$$

Another way to illustrate this phenomenon is by graph. In Fig. 2.19, we can see if the symbol boundary shifts the start phase of waves will be changed with a ratio of the wave frequency. This is a key point that the phase will rotate in frequency domain with a ratio of the frequency as shown in Fig. 2.20. That is when the frequency of sinusoidal wave higher the same length of boundary offset will insert more waves in that length. So the phase change will be larger.

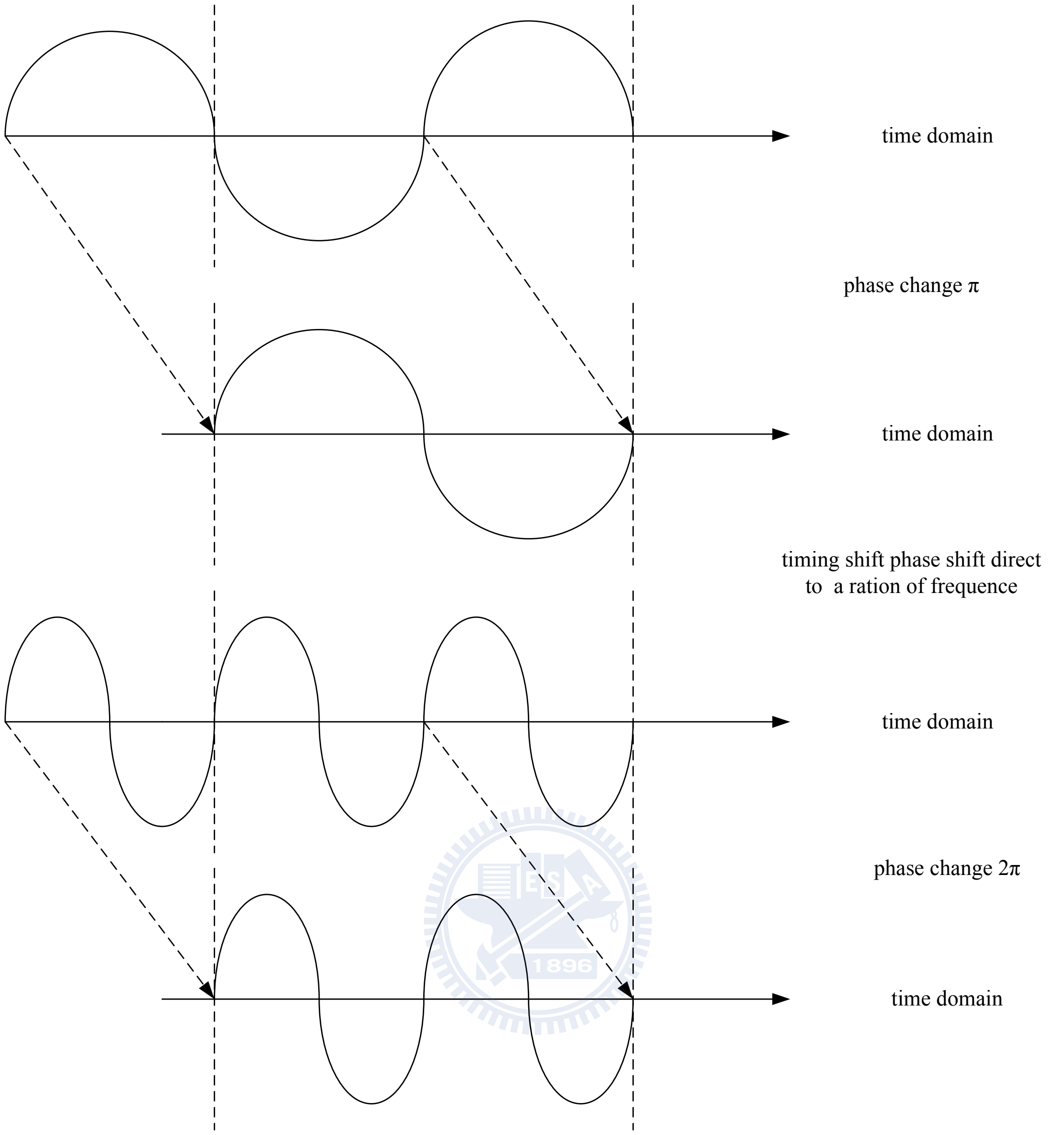


Fig. 2.19 Phase and symbol boundary shift

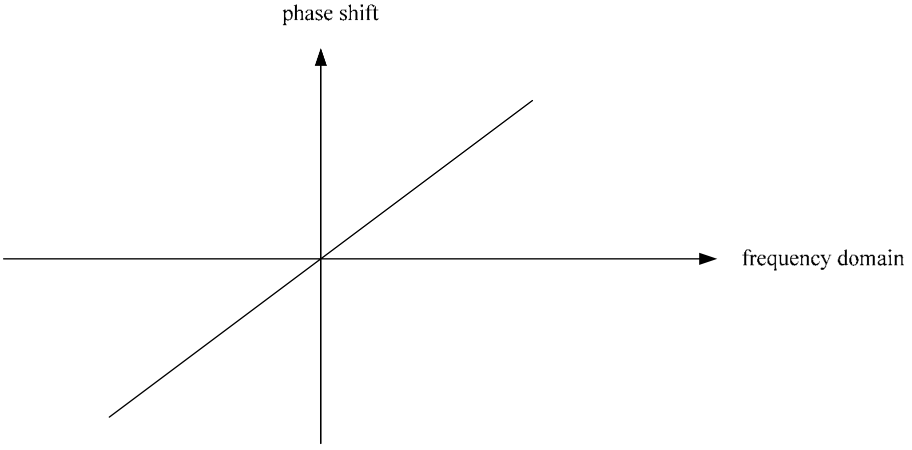


Fig. 2.20 Phase rotation

2.3.6 Phase Rotation Caused by SCO

For SCO, the FSTO part, which can be accumulated due to SCO, means the difference of boundary that shown Fig. 2.21. The main difference between FSTO and ISTO is the boundary offset is a fractional or an integer of sample rate The ISTO effect will makes the received data rotate in constellation, and the FSTO will make the phase rotate.

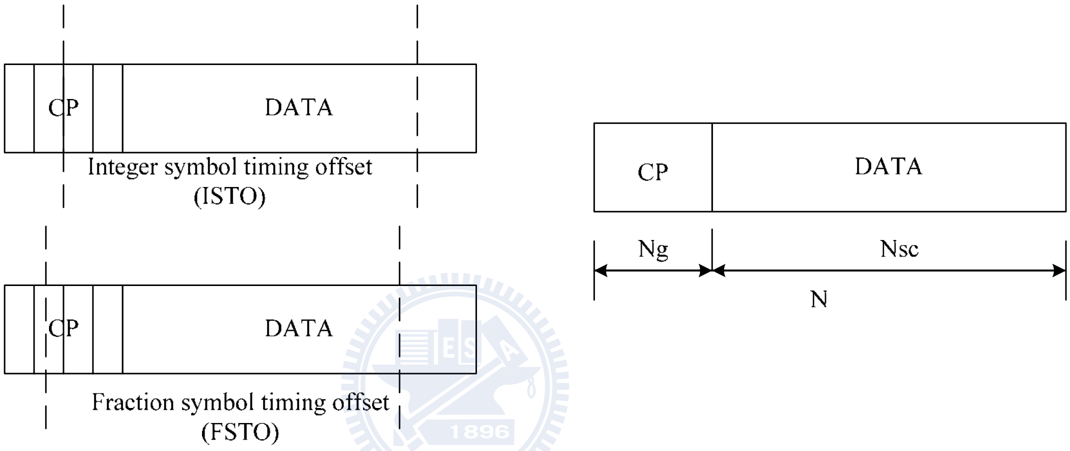


Fig. 2.21 Symbol timing relationship

This conclusion makes an important role on detecting SCO. Since FSTO will change by the accumulation of SCO. SCO value can be generated by the difference of phase rotation in two consecutive symbols in the system. The best way to detect the difference may be from the fixed value data such as pilots. In next section (Section 2.3.6), the challenge using pilots in 802.16e MIMO system will be discussed.

2.3.7 Challenge on a MIMO System

In 802.16e, pilot locations repeat every four consecutive symbols. In four consecutive symbols, pilot location differs from the next symbol as shown in Fig. 2.22. For example, pilot in antenna 0 arise in the (9+14*i*) subcarriers for first symbol, but arise in the (5+14*i*) subcarriers for the second symbol, in the (13+14*i*) subcarriers for third symbol, in the (1+14*i*) subcarriers for fourth symbol. On the other hand, pilots in antenna 1 arise in other different locations.

Though the pilot regulation of 2*1 antennas STBC in 802.16e as shown in Fig. 2.9, it seems that we can compare two consecutive symbols easily, for the pilots are in the same subcarrier location but from different antennas. However, in fact, the channel delay and channel response from two antennas are different as shown in Fig. 2.22. And this may cause different phase rotation in frequency domain as shown in Fig. 2.23., so that the phase rotation of two consecutive symbol are not with the same base phase rotation, and the difference of the phase of two symbols become meaningless. For the reason we decide to compare the phase every four symbols. However, the channel would vary; if we assume the channel does not change in four symbols, the phase rotation with channel response will be discussed in Section 3.1.

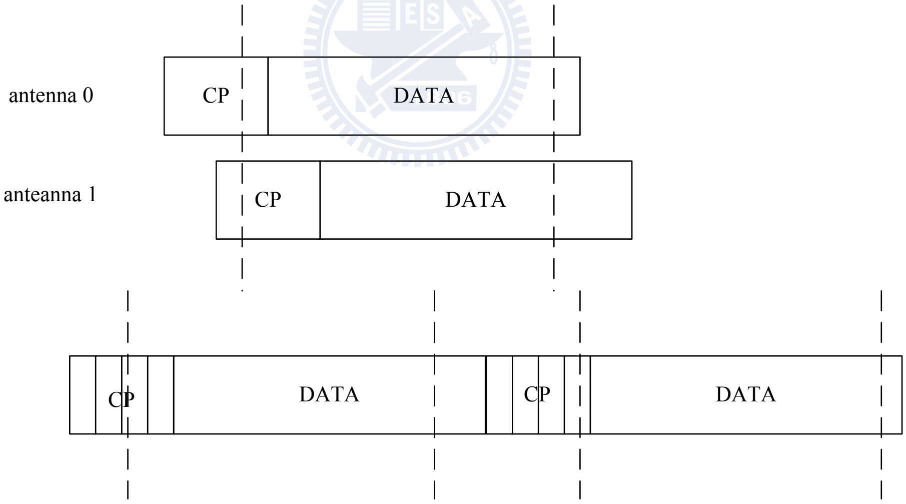


Fig. 2.22 Different channel delay for two antennas

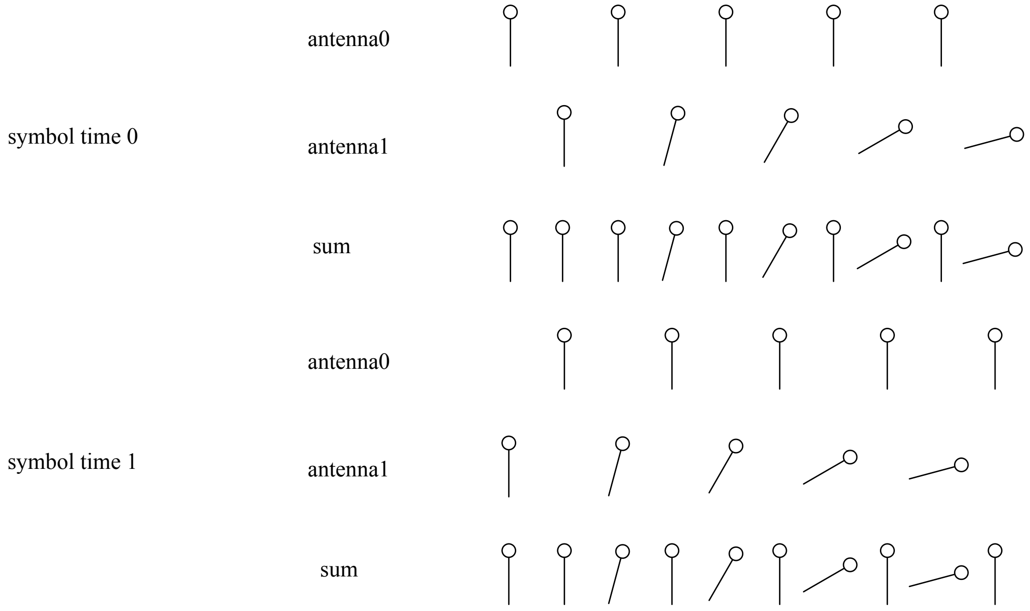


Fig. 2.23 Different phase rotation in 2 antennas

Fig. 2.23 shows that the locations of pilots from antenna 0 in symbol time 0 will be replaced by pilots from antenna 1 in symbol time 1. But the different basic phases of pilots are caused by the different channel delay and multipath gain. So as we can see, the pilot in corresponding location will be in different phase. Thus, although these two consecutive symbols have the same locations of pilots, the different basic phase makes the comparison meaningless in two consecutive symbols.

Chapter 3

SCO Estimation and Compensation in 802.16e

3.1 SCO Estimation

From the discussion in Section 2.3.2, we know SCO effects can be separated into ICI and phase rotation in frequency domain. The following section will mainly focus on the part of phase rotation for estimation and compensation. There are two reasons that we just control the part of phase rotation:

1. SCO effect on ICI is much smaller than that on phase rotation.
2. It is easy to estimate and compensate phase rotation effect, but the ICI effect is more complex.

The first reason will be simulated in the following graph (Fig. 3.1). In this figure, phase rotation effect is obvious more than ten times bigger than ICI effect in 802.16e for a 42 symbol frame without channel effect. The second reason is discussed in Section 2.3.2, and both of the effects have been illustrated.

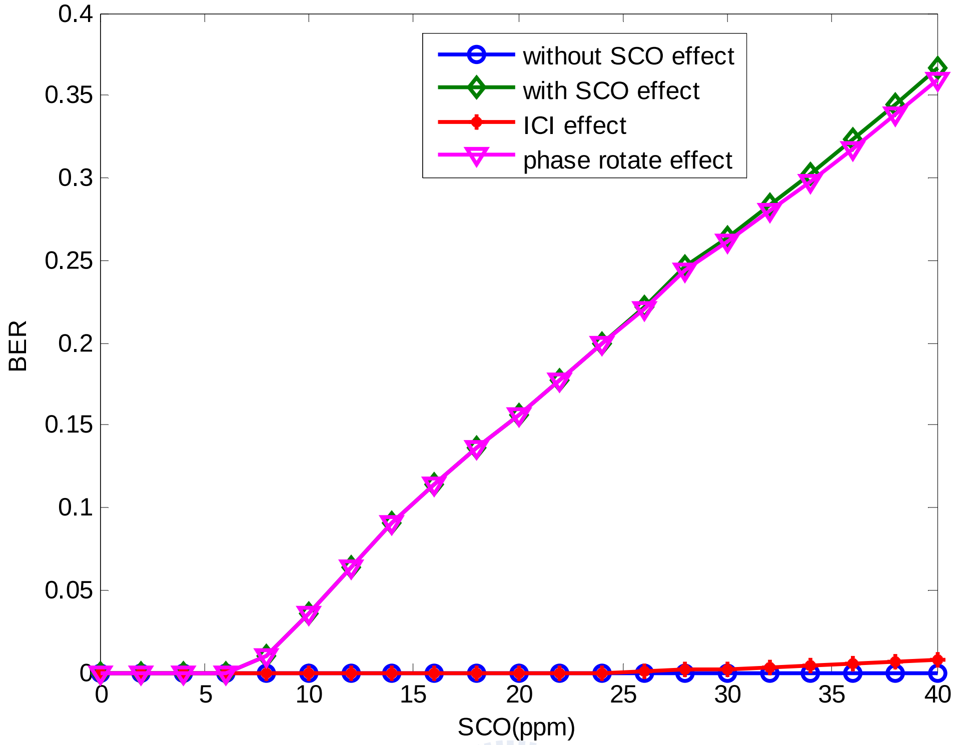


Fig. 3.1 Simulation of effect of ICI and phase rotation caused by SCO (perfect channel and no noise)

3.1.1 Detection of Phase Rotation in Frequency Domain

The phase rotation caused by SCO will change the phase of each subcarrier in the baseband of receiver. The subcarriers in the receiver are obviously through a channel. The final values of subcarriers in frequency domain with perfect synchronization will be multiplied with channel response and phase rotation as shown in Fig. 3.2.

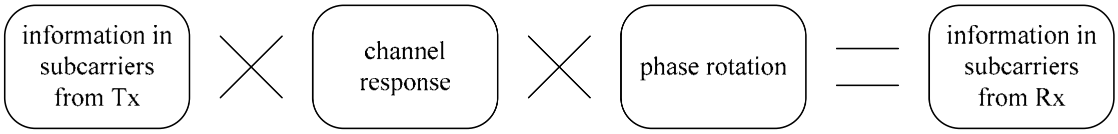

information in subcarriers from Tx
channel response
phase rotation
information in subcarriers from Rx

Fig. 3.2 Relations of Tx and Rx data in frequency domain with perfect synchronization

It is important to get the pure phase rotation without effects of channel response. However, in our baseband design as shown in Fig. 3.3, the channel estimation is after the part of synchronization, and the SCO estimation and compensation is before the part of channel estimation. If we need to resolve the phase rotation effect from the information of channel response, there must be a feedback loop in our design. Else we needed to change SCO estimation and compensation after the channel response. However, both of the solutions are not reasonable, for the feedback loop will cause requirement of speed in hardware, and the effect of will be interfered after channel estimation

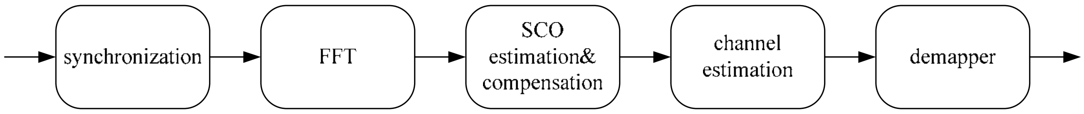


Fig. 3.3 802.16e baseband overview

3.1.2 Algorithm to Estimate SCO by Using Pilots

In order to estimate SCO, we must remove the channel response from the original value of subcarrier to get pure effect of SCO since the information of Rx is the multiplication of information of Tx, channel response, and phase rotation as shown in Fig. 3.2 shows. There is an easy way to get rid of unwanted information mentioned above. If we assume the channel responses do not vary very much in two consecutive symbols, then both of the channel responses in two consecutive symbols can be viewed as the same. Thus, the channel effect in phase can be eliminated by some operations. The pilots are good for SCO estimation, because the values of them are known. So, we can get the information of channel from by the pilots in the receiver by two consecutive symbols. If we take a multiplication of this time symbol with conjugated pre-symbol, as the following equations (Eqn. (3.1)) shows, the phase caused by pilot and channel responses will disappear.

$$
\begin{aligned}
R_{lk} &= H_k P_{lk} e^{j2\pi k \frac{\varepsilon}{N_{sc}}} \\
Z_{lk} &= R_{lk} R_{l-1k}{}^{*} \\
&= H_k P_{lk} e^{j2\pi k \frac{\varepsilon_k}{N_{sc}}} \left(H_k P_{l-1k} e^{j2\pi k \frac{\varepsilon_{k-1}}{N_{sc}}} \right)^{*} \\
&= \left|H_k\right|^2 \left|P_{lk}\right|^2 e^{j2\pi k(\varepsilon_k - \varepsilon_{k-1})} \\
&= \left|H_k\right|^2 \left|P_{lk}\right|^2 e^{j2\pi kN\delta}
\end{aligned}
\tag{3.1}
$$

where *l* is the symbol index, *k* is the subcarrier index, Z_{lk} is for finding value of phase, R_{lk} is the OFDM symbol after FFT with pilot only, P_{lk} is the transmitted pilot signal, H_k is the channel response of k^{th} subcarrier, and δ is the sampling timing offset, ε is the accumulation of sampling clock offset, which means the difference in starting symbol timing, in other words FSTO discussed in Section 2.3.4. We can get sampling clock offset δ by solving the equation. Eqn. (3.2) is the way to get the value of δ.

$$
\begin{aligned}
\phi_{1,l} &= \left[\sum_{k=1}^{K/2} \angle Z_{l,k} \right] = \left[\sum_{k=1}^{K/2} \angle R_{lk} R_{l-1k}{}^{*} \right] = \sum_{k=1}^{K/2} 2\pi kN\delta = 2\pi N\delta \star \frac{1}{2} * \frac{K}{2} * (\frac{K}{2} + 1) \\
\phi_{2,l} &= \left[\sum_{k=-1}^{-K/2} \angle Z_{l,k} \right] = \left[\sum_{k=-1}^{-K/2} \angle R_{lk} R_{l-1k}{}^{*} \right] = \sum_{k=-1}^{-K/2} 2\pi kN\delta = -2\pi N\delta \star \frac{1}{2} * \frac{K}{2} * (\frac{K}{2} + 1)
\end{aligned}
$$

$$
\delta = \frac{1}{2\pi N(\frac{K}{2} * (\frac{K}{2} + 1))} (\phi_{1,l} - \phi_{2,l}) \tag{3.2}
$$

3.1.3 Pilots Regulation in 802.16e and Solution

From the discussion above (Section 3.1.2) and pilot regulations in 802.16e mentioned in Section 2.3.6, we know the pilots are not very regular in two consecutive symbols, so Eqn. (3.1) and Eqn. (3.2) can not be used directly. Fig. 3.4 shows the difference in four symbols, where s is the transmitted signals, *V* is the SCO effects, and *ch0* and *ch1* is the channel responses of two Tx antenna to Rx antenna. It is obviously hard to find some regulation to solve *V* (SCO effects) directly. So the finding function is modified as Eqn. (3.3), which means when the pilots repeat every four symbols as discussed in Section 2.3.6, and assumes the channel response would not vary in four symbols much.

$$
\begin{aligned}
R_{lk} &= H_k P_{lk} e^{j2\pi k \frac{\varepsilon}{N_{sc}}} \\
Z_{lk} &= R_{lk} R_{l-4k}^{\;*} \\
&= H_k P_{lk} e^{j2\pi k \frac{\varepsilon_k}{N_{sc}}} \left(H_k P_{l-1k} e^{j2\pi k \frac{\varepsilon_{k-4}}{N_{sc}}} \right)^* \\
&= \left|H_k\right|^2 \left|P_{lk}\right|^2 e^{j2\pi k(\varepsilon_k - \varepsilon_{k-4})} \\
&= \left|H_k\right|^2 \left|P_{lk}\right|^2 e^{j2\pi k 4N\delta}
\end{aligned}
\qquad (3.3)
$$

	time slot 0		time slot 2	
antenna 0 (ch0)	s0 pilot 5+14k	-s1* 9+14k	s2 1+14k	-s3* 13+14k
antenna 1 (ch1)	s1 pilot 9+14k	s0* 5+14k	s3 13+14k	s2* 1+14k

rx_signal	s0+s1	-s1*+s0*	s2+s3	-s3*+s2*
rx_signal with channel response	ch0s0+ch1s1	ch0-s1*+ch1s0*	ch0s2+ch1s3	ch0-s3*+ch1s2*
rx_signal with SCO effect	s0+s1	V(-s1*+s0*)	V^2(s2+s3)	V^3(-s3*+s2*)
rx_signal with channel response & SCO effect	ch0s0+ch1s1	V(ch0-s1*+ch1s0*)	V^2(ch0s2+ch1s3)	V^3(ch0-s3*+ch1s2*)

Fig. 3.4 Phase rotation with channel response

3.2 SCO Compensation

The way to compensate SCO effect can be mainly partitioned into two kinds. One is time domain compensation which compensates the difference in time domain; the other is frequency domain compensation which rotates back the phase change caused by SCO.

3.2.1 Time Domain SCO Compensation

Time domain compensation can be viewed as resampling as shown in Fig. 3.5. We can use Sinc function to transfer the data from discrete time domain to continuous time domain, and then resample them with accurate frequency. However, in hardware design, especially in digital design, it is hard to make a real Sinc function to reconstruct the continuous signal. So some practical structures such as Lagrange Cubic filter will be introduced in Section 4.3.

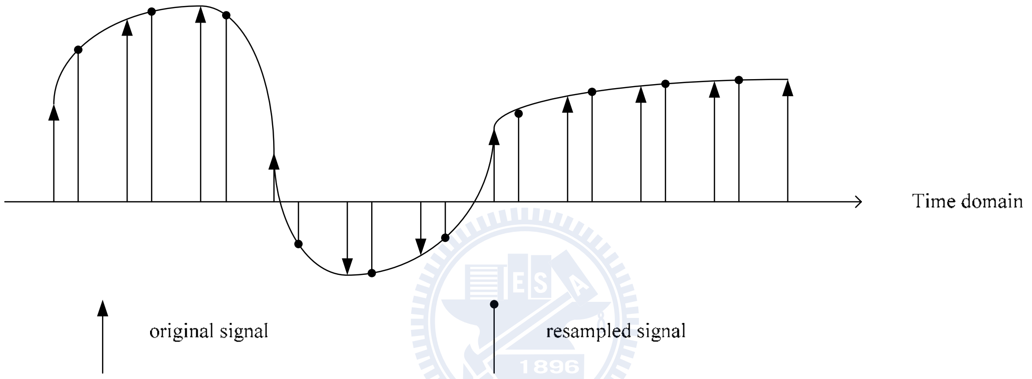


Fig. 3.5 Resample in time domain

3.2.2 Frequency Domain SCO Compensation

The compensation in frequency domain means only to compensate the effect of phase rotation in frequency domain. As SCO effects mentioned in Section 2.3.3, it can be partitioned into ICI and phase rotation. The simulation in Fig. 3.1 shows that the phase rotation effect is much larger than ICI effect.

3.2.3 Comparison of Time and Frequency Domain Designs

The ideal time domain compensation can eliminate ICI effect and phase rotation, whereas the frequency domain compensation can only eliminate the effect of phase rotation.

However, the time domain compensation is usually non-ideal, for a ideal low pass filter or a ideal Sinc function is hard to implement in hardware. So time domain compensation suffered from the error of some simplified design of filter. Frequency domain compensation is easier for implementation but suffered from the accuracy of phase rotator based on the design of CORDIC, because it need to rotate the angle and then transfer it to a complex number.

Fig. 3.6 shows the simulation of four different types of compensation with perfect known SCO value: ideal time domain compensation, ideal frequency domain compensation, Lagrange Cubic time domain compensation, CORDIC frequency domain compensation.

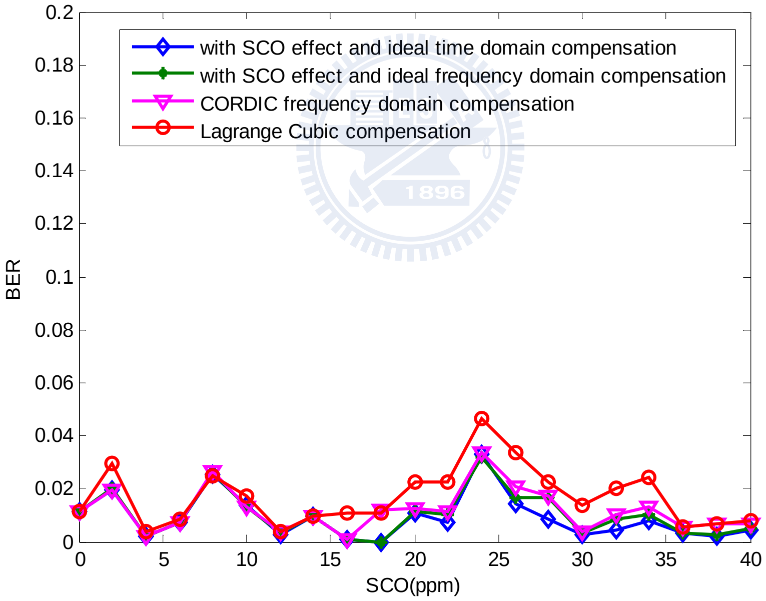


Fig. 3.6 Simulation and comparison of time and frequency domain compensation (Eb/N0=16dB 120km/hr)

3.3 Combination of SCO Estimation and Compensation

3.3.1 Difficulties in Time Domain Compensation

The way to estimate SCO by detecting phase rotation is in frequency domain.

However, if the time domain compensation method is chosen, it will cause a problem: the difference of domains for estimation and compensation. As the baseband receiver architecture proposed in 802.16e, FFT is a separator between time domain and frequency domain, and the calculation of frequency domain is after the calculation of time domain .Therefore, it needs a feedback loop as shown in Fig. 3.7.

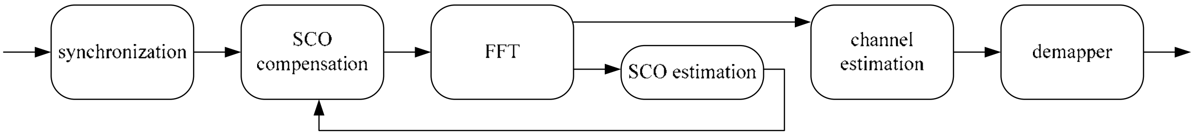
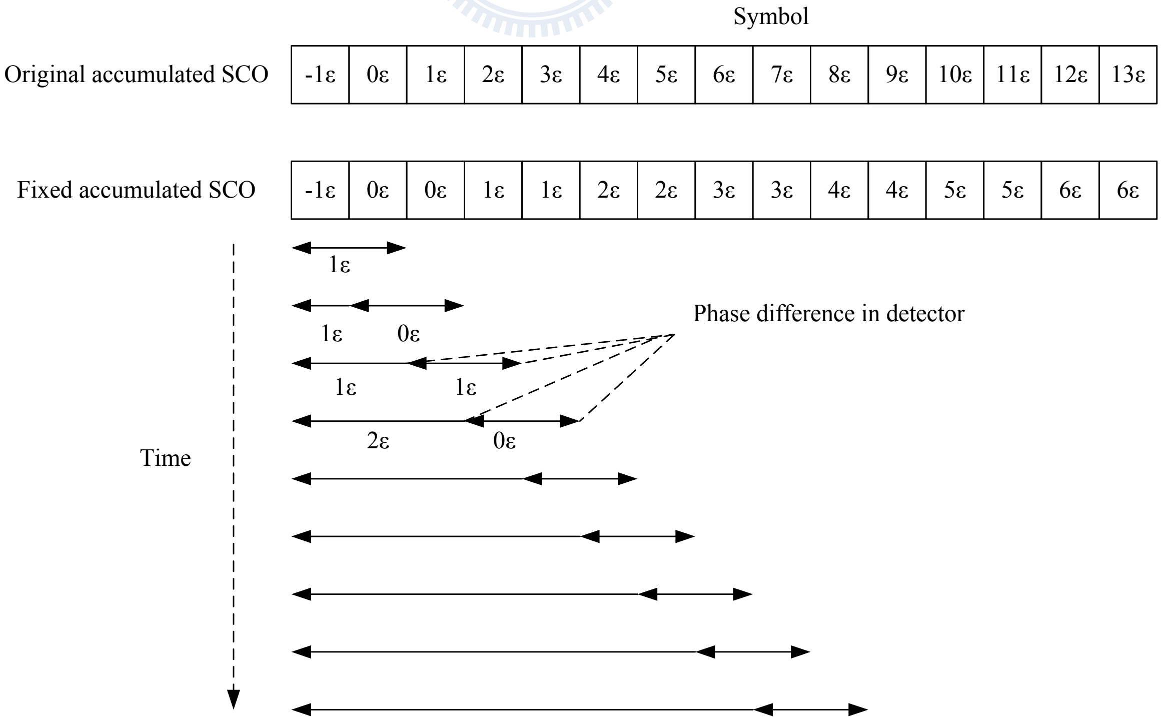


Fig. 3.7 Feedback from estimation to compensation

In our design it seems perfect if the SCO value is known, but in reality this structure does not work well. The reason is the un-fair comparison on the pilots which means some of the pilots are compensated and some of them are not, and this effect is shown in Fig. 3.8.



	Symbol														
Original accumulated SCO	-1ε	0ε	1ε	2ε	3ε	4ε	5ε	6ε	7ε	8ε	9ε	10ε	11ε	12ε	13ε
Fixed accumulated SCO	-1ε	0ε	0ε	1ε	1ε	2ε	2ε	3ε	3ε	4ε	4ε	5ε	5ε	6ε	6ε

Fig. 3.8 Errors caused by feedback

If we get SCO value from the first symbol time and the second symbol time, it will be corrected (ε), and then the third symbol time will be fixed, so SCO value gotten from second and third symbol will be 0. Thus, the forth symbol will not be fixed in reality.

This shows that if in the first symbol the subcarriers are fixed by the estimation value, but in the second symbol the subcarriers are not, the latter value of estimation will not be what we expected. In fact, it fixes about half only. In order to prevent this phenomenon, it requires more registers or memory to store the value of pilots which is not compensated. However the cost by doing so is large.

3.3.2 Frequency Domain Compensation

In order to estimate and compensate subcarriers in the same symbol time, large number of memory are needed, since the SCO value is acquired by phase rotation detection. So it needs all pilots in this time symbol to calculate, and thus the data in frequency domain need to be stored until the time when the SCO value is calculated so that the data can be fixed in time

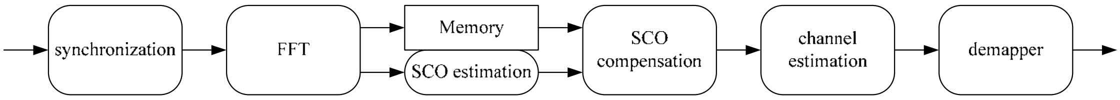


Fig. 3.9 Large memory frequency domain compensation

3.3.3 A Memory-efficiency Method in Frequency Domain Compensation

However the cost of using memory to save all subcarriers is large, if we can use the SCO value estimated in previous symbol to fix the current symbol, the memory will not be necessary as shown in Fig. 3.10. However, the condition is that the SCO value can’t change very much among symbol time. In our simulation as shown in Fig. 3.11, we will find that if the variation of SCO between two symbols is below 5 ppm, the performance degradation of using memory-reduced method can be neglected, and it is reasonable for our design, since the SCO value would not vary very much between two symbols, because the cause of SCO is mainly from timing jitter of oscillator and Doppler effects. These phenomenons do not vary much in time scale.

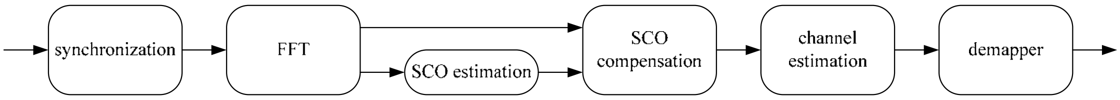


Fig. 3.10 Memory-efficiency frequency domain compensation

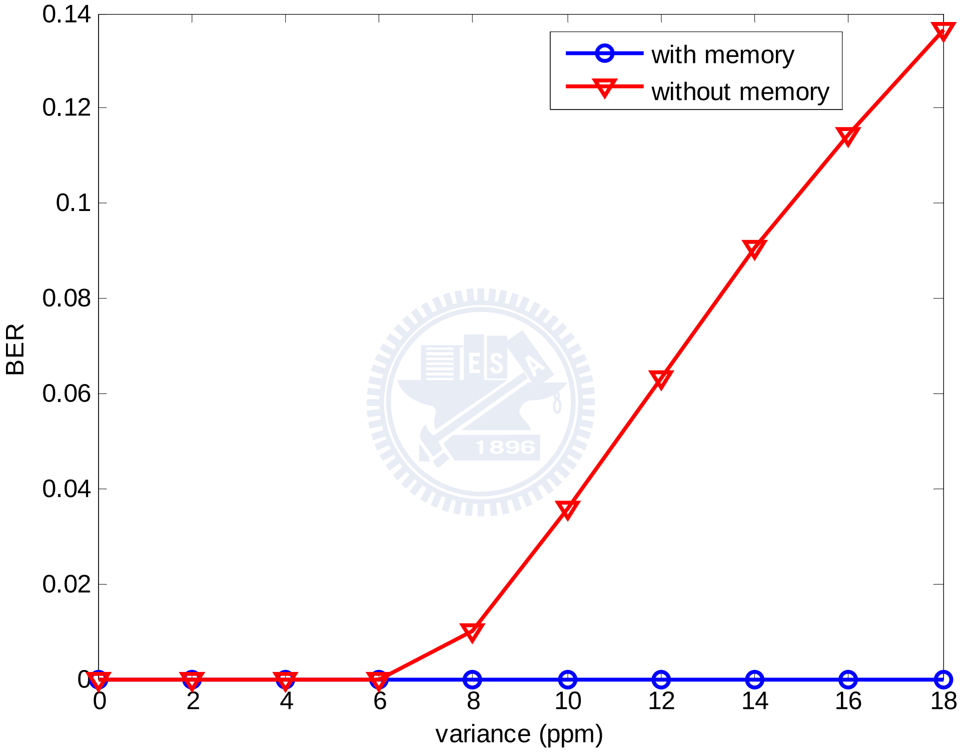


Fig. 3.11 Comparison of register and register-reduced way (perfect channel and no noise)

Chapter 4

Architecture Design

4.1 Overview of Baseband Receiver

The overall block diagram of 802.16e OFDMA baseband is shown in Fig. 4.1 [5]. It mainly contains four parts: synchronization, SCO compensation, FFT, and channel estimation. In the following, we will focus on SCO compensation circuits. The main issue now is hardware complexity and architecture. For example, many complicated mathematic operations which cost mass of hardware and power are used in these algorithms, such as angle computation. It may be implemented by using look-up table method which needs a lot of memory to store the values of angle. Therefore, we use coordinate rotational digital computer (CORDIC) scheme instead of the memory based circuit. It will only use adders, shifters and multiplexers to realize the angle calculation. Additionally, some mathematic and logical simplifications also successfully reduce the hardware cost and power. The detailed methods about implementation will be discussed in the following sections.

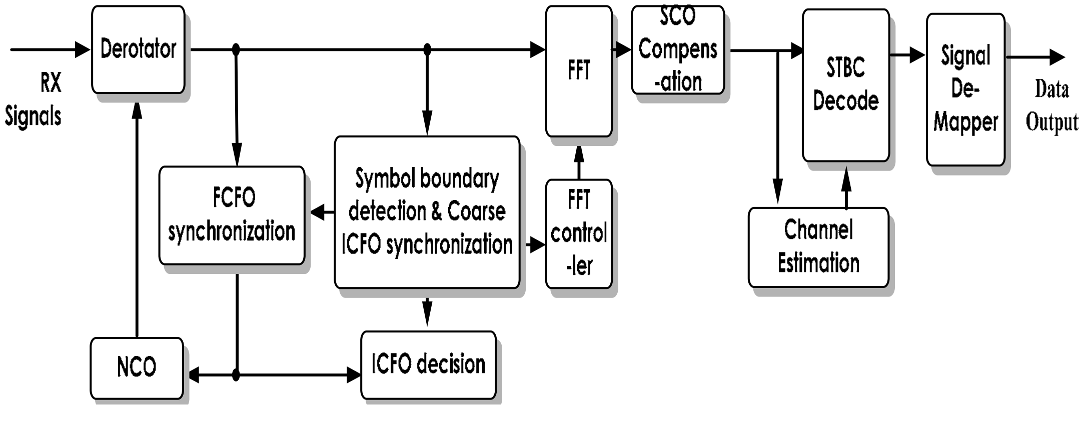


Fig. 4.1 Overall block diagram of baseband

4.2 Overview and Architecture of CORDIC

In our design, the phase of a complex number is needed for the estimation and compensation. So a good architecture to transform from complex number to phase or from phase to complex number is important for our hardware. The COordinate Rotational DIgital Computer (CORDIC) [12][13] algorithm is a well-known Digital Signal Processing (DSP) algorithm for computing between phase and complex number.

4.2.1 Mathematical Representation

The basic idea of CORDIC is phase rotation as shown in Eqn. (4.1), which rotate the vector (x_{in},y_{in}) with a phase θ. By taking the $\cos\theta$ from the matrix, we have Eqn. (4.2). As $\cos\theta$ is a positive number between -90 to 90 degrees, it does not affect the phase of new value of (x_{out}, y_{out}) without normalization. By assuming the $\tan\theta$ in the matrix is power of 2. We get Eqn. (4.3), which forms a basic vector rotation unit, where μ is the unit (+1 or -1) to determine whether $\tan\theta$ is $+2^{-i}$ or -2^{-i}. A general phase can be approached by the multiply of lots of $\tan\theta$ which is power of 2 as shown in Eqn. (4.4). Therefore, the total phase rotation contributed by the multiplications of $\tan\theta$ with power of 2 can be obtained by the accumulation of phase corresponding to each multiplication of $\tan\theta$ as shown in Eqn. (4.5). Thus, the final way to get the phase of a complex number by CORDIC can be shown as Eqn. (4.6), as K_i ($\cos\theta_i$) $=\sqrt{1+2^{-2i}}$ only affects the length of a complex number but not the phase of a complex number, so it can be neglected when we just want to know the phase to reduce the complexity of operations. In Eqn. (4.6), $z(i)$ is the residue of angle at i^{th} rotation. The CORDIC algorithm has two operation modes, the phase to complex number mode and the complex number to phase mode. In the phase to complex number mode, a vector is rotated by an angle θ to obtain the new vector. In every iteration i, the elementary angle is determined by angel remainder *z(i)* for added or subtracted. On the contrary, the complex number to phase mode tries to calculate the angle and the magnitude of the vector. For this propose, the vector is rotated toward the x-axis and μ_i is determined by the sign of *y(i)*.

$$\begin{bmatrix} x_{out} \\ y_{out} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\ \theta \end{bmatrix} \cdot \begin{bmatrix} x_{in} \\ y_{in} \end{bmatrix} \tag{4.1}$$

$$\begin{bmatrix} x_{out} \\ y_{out} \end{bmatrix} = \cos\theta \begin{bmatrix} 1 & -\tan\theta \\ \tan\theta & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{in} \\ y_{in} \end{bmatrix} \tag{4.2}$$

$$\begin{bmatrix} x_{(i)out} \\ y_{(i)out} \end{bmatrix} = K_i \begin{bmatrix} 1 & -\mu_i 2^{-i} \\ \mu_i 2^{-i} & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{(i)in} \\ y_{(i)in} \end{bmatrix} \tag{4.3}$$

$$\begin{bmatrix} x_{out} \\ y_{out} \end{bmatrix} = \prod_{i=0}^{N-1} K_i \begin{bmatrix} 1 & -\mu_i 2^{-i} \\ \mu_i 2^{-i} & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{in} \\ y_{in} \end{bmatrix} \tag{4.4}$$

$$\theta = \sum_{i=0}^{N-1} u_i \tan^{-1}(2^{-i}) \tag{4.5}$$

$$\begin{aligned} x(i+1) &= x(i) - \mu_i 2^{-i} y(i) \\ y(i+1) &= \mu_i 2^{-i} x(i) + y(i) \\ z(i+1) &= z(i) - \mu_i \tan^{-1}(2^{-i}) \end{aligned} \tag{4.6}$$

From the equations, there is one important element $\tan^{-1}(2^{-i})$ that is not easy to calculate. So we need to establish a table to record the value of $\tan^{-1}(2^{-i})$. However it is worth doing because for an N^{th} iteration calculation, we only need to record *N* values, but it needs lots of memory if we get the phase of a complex number by a look up table. The table of $\tan^{-1}(2^{-i})$ is shown at Table. 4.1. From the table the least error of phase must less than the smallest value of $\tan^{-1}(2^{-i})$, so if the iteration is high enough, the error will be small enough.

Table. 4.1 Values of $\tan^{-1}(2^{-i})$

i	2^{-i}	(DEG)
0	1.000000000	45.000000
1	0.50000000	26.565000
2	0.25000000	14.036000
3	0.12500000	7.125000
4	0.06250000	3.576300
5	0.03125000	1.789900
6	0.01562500	0.895170
7	0.00781250	0.447610
8	0.00390630	0.223810
9	0.00195310	0.111900
10	0.00097656	0.055953

4.2.2 Folded and Unfolded Structure

There are two possible structures to implement CORDIC: folded and unfolded. It need n iterations to approach the phase we want to solve, the folded structure means using one structure to execute each iteration by just changing the variable in each iteration as shown in Fig. 4.2, whereas unfolded structure means using n times of hardware to calculate the value of phase as shown in Fig. 4.3 .

Fig. 4.2 shows the detailed structure of the folded structure of CORDIC, in this structure we needs n iterations to solve the phase, every time after a phase rotation, the value of Xn, Yn and Zn will be recalculated and the sign of Yn will be used to determine the direction of rotation in the next loop.

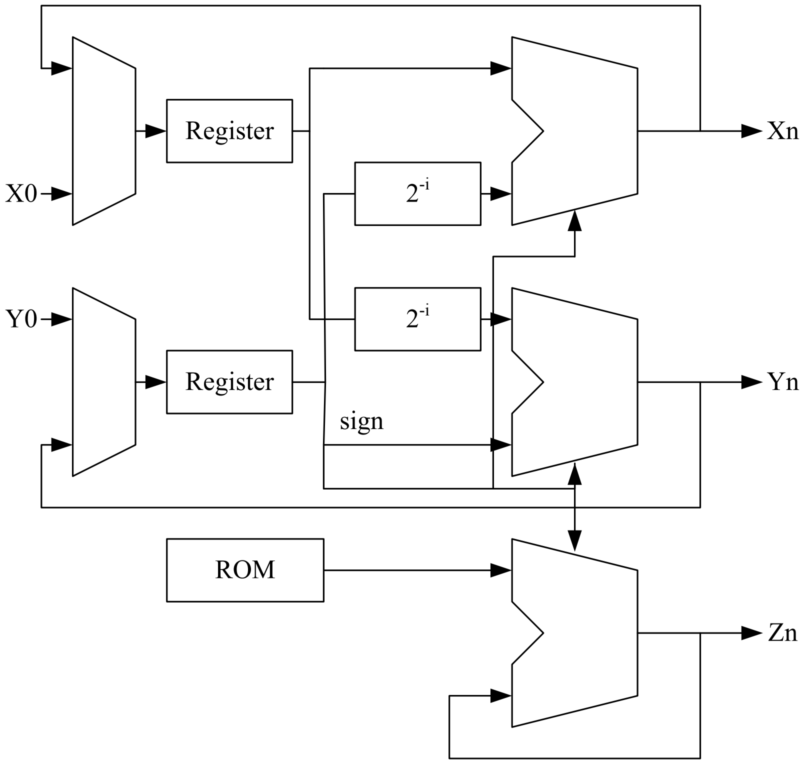


Fig. 4.2 Folded structure

Fig. 4.3 shows the detailed structure of the unfolded structure of CORDIC, it pass variables from one stage to another. It contains more hardware, but without registers and can be pipelined to speed up the throughput rate.

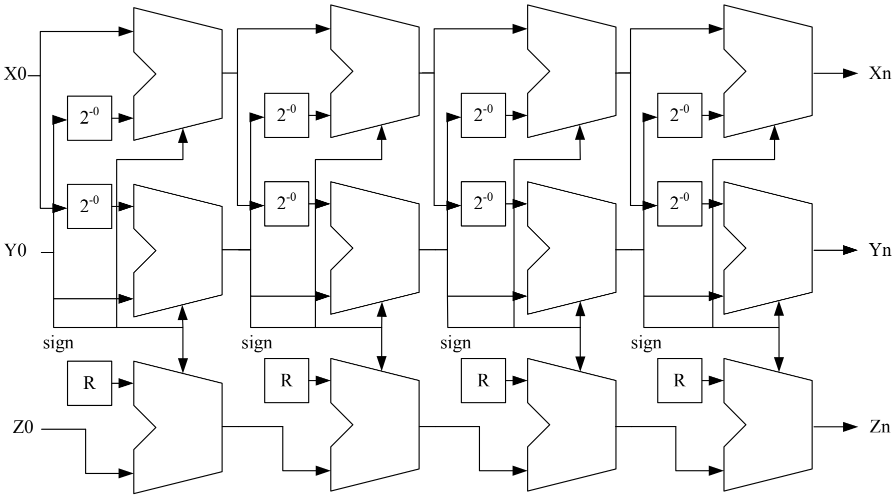


Fig. 4.3 Unfolded structure

4.3 Architecture of SCO Detector

Generally, the architecture of detector for SCO is based on the estimation method discussed in Section 3.1 and is shown in Fig. 4.4. Because the value of SCO is obtained by the difference of phase from the same pilots' time slots, the detector consists of a phase accumulator to accumulate the change of SCO in each time slot, a CORDIC to transform from complex number to phase, and a divider to get the average of estimated phase. However, transformation many times by CORDIC many times may corrupt the data, since the CORDIC is not lossless. There are two ways to reduce this effect: one is to replace addition in phase by multiplication in complex number, the other one is to use large word length in CORDIC to get more precise phase.

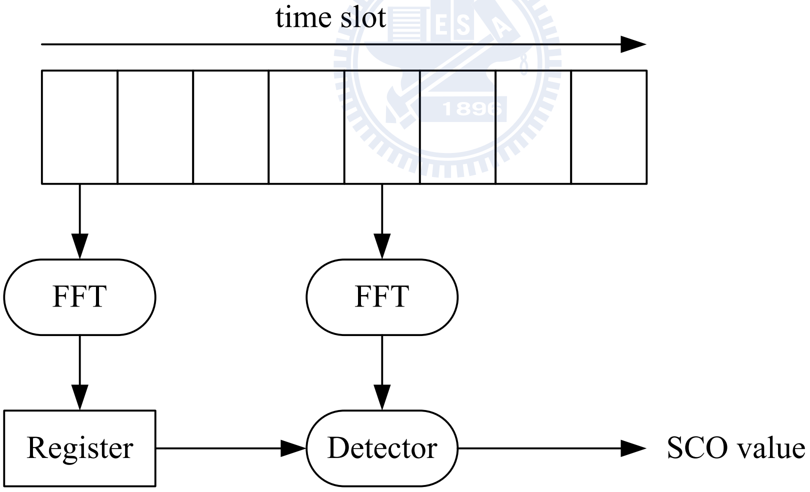


Fig. 4.4 Concept structure of detection

SCO value can be obtained by the difference of phase in symbols. Each time we compare the phase difference of pilots between two consecutive time slots as mentioned in Section 3.1, we get the value of SCO in this time slot. However we have to accumulate SCO value in this time slot with previous SCO values to get the total effect on phase rotation for us to do the compensation easily. The hardware of SCO detector is shown in Fig. 4.5. Inside the detector, there are three main parts: phase comparison, registers to store previous symbol pilots, and the architecture to average and normalize the value of SCO caused by phase comparison; the other hardware is the phase accumulator to implement the remembrance of total phase caused by SCO mentioned above.

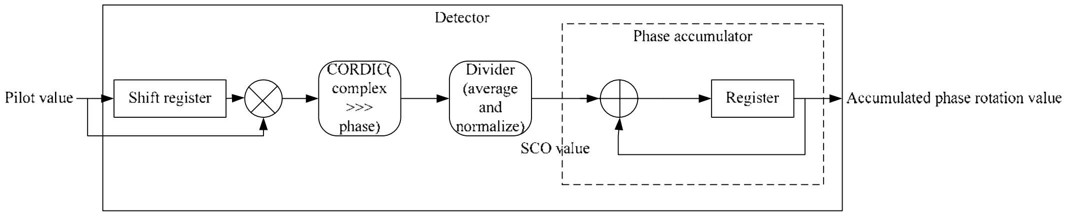


Fig. 4.5 Structure of detector

4.4 Foundation and Architecture of Interpolator

4.4.1 Introduction

In our design, an interpolator is needed for time domain compensation which means compensate the error between time scales as shown in Fig. 3.5. The ideal time domain compensation can eliminate the ICI effect if SCO value is known, whereas the frequency domain compensation can not.

4.4.2 Ideal Low Pass Filter – Sinc Function Filter

The best way to reconstruct signal is using Sinc function, which means the ideal low pass filter in frequency domain. From the sampling theory [8], the ideal low pass filter is needed to resample. Fig. 4.6 shows a Sinc function is used to cover the sampled signal to the original analog signal.

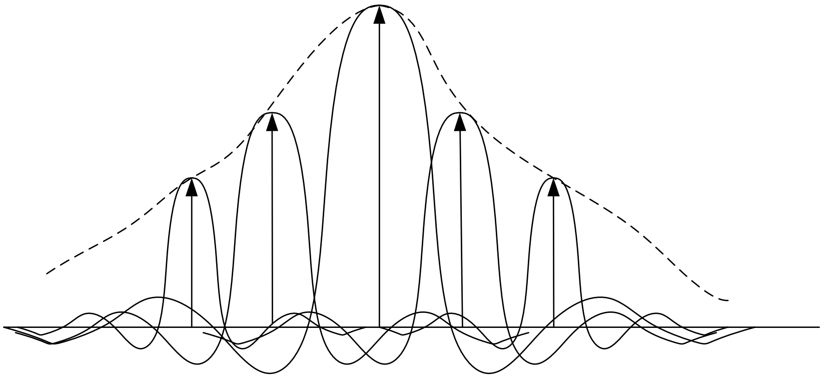


Fig. 4.6 Reconstruction of analog signal by Sinc function

The following mathematical representation illustrates the effect above. The sampled digital signals *x(nT$_s$)* go through the DAC and become an analog impulses, as shown in Eqn. (4.7). The impulses are filtered by a low pass filter, as Eqn. (4.8). Then, the continual analog signals are resampled at sample period T_i. Finally, the resample signal *x(kT$_i$)* is shown in Eqn. (4.9).

$$\begin{aligned} x_s &= r(t) \cdot \sum_{n=-\infty}^{\infty} \delta(t - nT_S) \\ &= \sum_{n=-\infty}^{\infty} r(nT_s) \cdot \delta(t - nT_S) \end{aligned} \tag{4.7}$$

$$\begin{aligned} X(t) &= x_s(t) \otimes Sinc(t) \\ &= \sum_{n=-\infty}^{\infty} x_s(nT_S) Sinc(t - nT_s) \end{aligned} \tag{4.8}$$

$$X(kT_i) = \sum_{n=-\infty}^{\infty} x_s(nT_S) Sinc(kT_i - nT_s) \tag{4.9}$$

However, it is difficult to implement Eqn. (4.9) in hardware architecture, for the following reasons:

1. It is difficult to generate Sinc function by simple computation.
2. The difference in time is a variable, so it can not use a table to generate Sinc function.
3. It needs infinite Sinc function to add up for generating the resampled value.

4.4.3 Lagrange Cubic Filter

The other way which is easier to implement in contrast to Sinc function is using Lagrange Cubic filter. However it is not an ideal low pass filter in frequency domain, and at higher frequency it has more variance in magnitude. However, in OFDM system, higher frequency is the guard band so it is appropriate to use in OFDM the system.

The original ideal low pass filter is shown in Eqn. (4.9), if we define $i = \lfloor kT_i / T_s \rfloor - n$, $m_k = \lfloor kT_i / T_s \rfloor$ and $u_k = kT_i / T_s - m_k$,

Then, Eqn. (4.9) becomes Eqn. (4.10)

$$y(kT_i) = \sum \left[x(nT_s) \right] \left[Sinc((i + u_k)T_s) \right] \tag{4.10}$$

The impulse response of ideal interpolation is Sinc function. To compute approximate coefficients of Sinc function, we replace it with *h(n)*, the well known polynomial based interpolation [16]. The Lagrange interpolation is a simple way to implement polynomial based interpolator.

The simplest FIR filter is the Lagrange interpolation [16] as shown as Eqn. (4.12):

$$h(i) = \prod_{k=0, k \neq i}^{L} \frac{D-k}{i-k}, i = 0,1,2.....L \tag{4.12}$$

where *L* is the order of coefficients of the filter, *D* is the fractional delay (μ_k) , *h(n)* is the interpolation coefficients. For *L* = 3, it is so called Lagrange Cubic interpolator and the interpolation coefficients are:

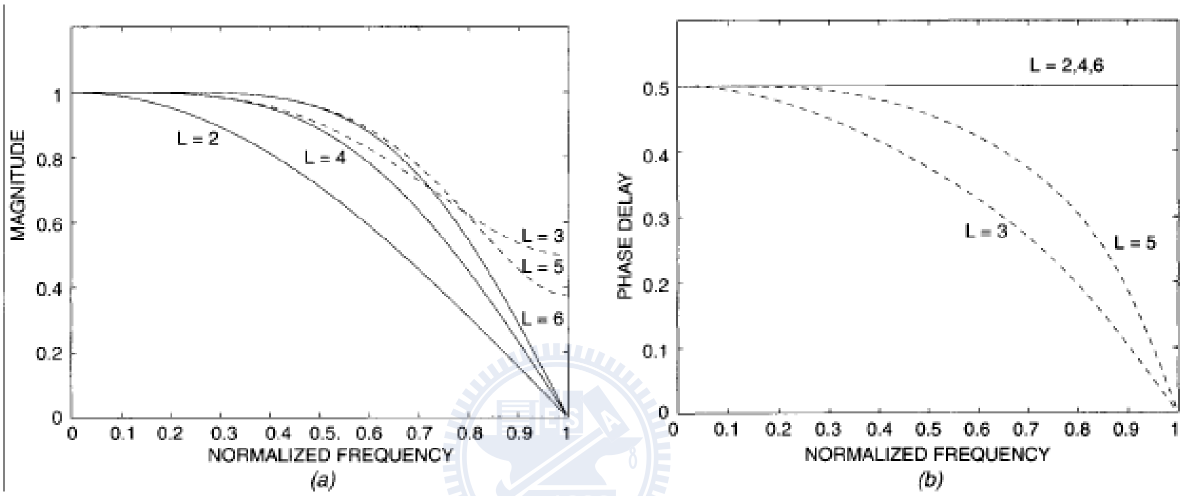
$$h(-2) = \frac{(D+1)D(D-1)}{6} = \frac{1}{6}D^3 - \frac{1}{6}D \tag{4.13}$$

$$h(-1) = \frac{(D+1)D(D-2)}{(-2))} = -\frac{1}{2}D^3 + \frac{1}{2}D^2 + D \tag{4.14}$$

$$h(0) = \frac{(D+1)(D-1)(D-2)}{(-2)} = \frac{1}{2}D^3 - D^2 + \frac{1}{2}D + 1 \tag{4.15}$$

$$h(1)=\frac{D(D-1)(D-2)}{(-6)}=-\frac{1}{6}D^3+\frac{1}{2}D^2-\frac{1}{3}D \tag{4.16}$$

Increasing the filter length of an interpolator can get better performance, as shown in Fig. 4.7 [16]. In 802.16e system, guard band is from 0.8 to 1 in normalized frequency, so the worst condition in this filter will not affect the performance seriously. A longer length interpolator also increases the hardware complexity. In the simulation platform, a four tap (*L*=3) Lagrange Cubic interpolator is adopted in.



(a) Magnitude and (b) phase delay responses of Lagrange interpolating filters of length *L*=2,3,4,5,6 with *d* =0.5

Fig. 4.7 Distortion of Lagrange filter in frequency domain [16]

An efficient implementation was proposed by Farrow [17] as shown in Fig. 4.8. The advantage of Farrow structure is to reduce the number of multipliers. The Farrow structure of Lagrange Cubic interpolator is shown in Fig. 4.9. According to different fraction delay, the interpolator needs to compute the corresponding filter coefficients. However, the generation of filter coefficient needs a lot of multipliers. The Farrow structure shares the multipliers in the generation equation. Hence, the Lagrange Cubic interpolator only needs three multipliers for coefficient generation.

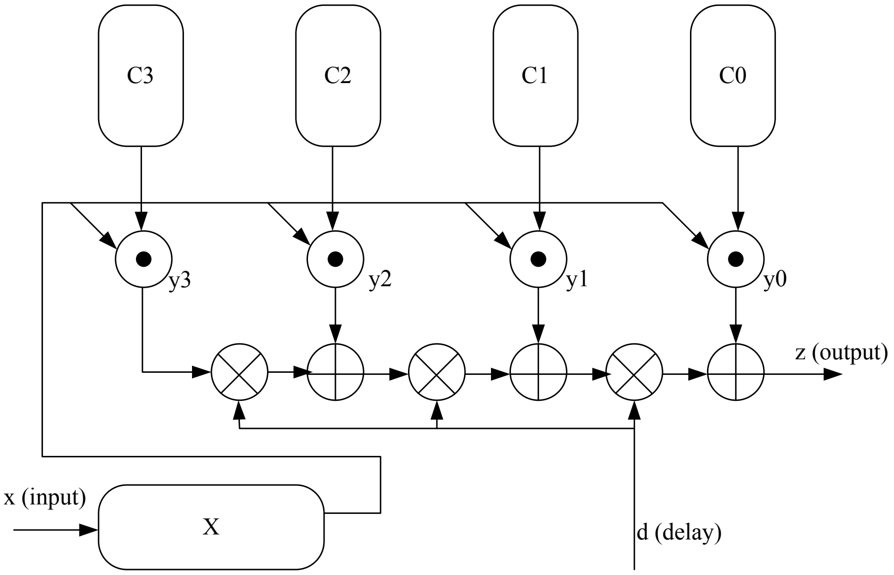


Fig. 4.8 Farrow structure

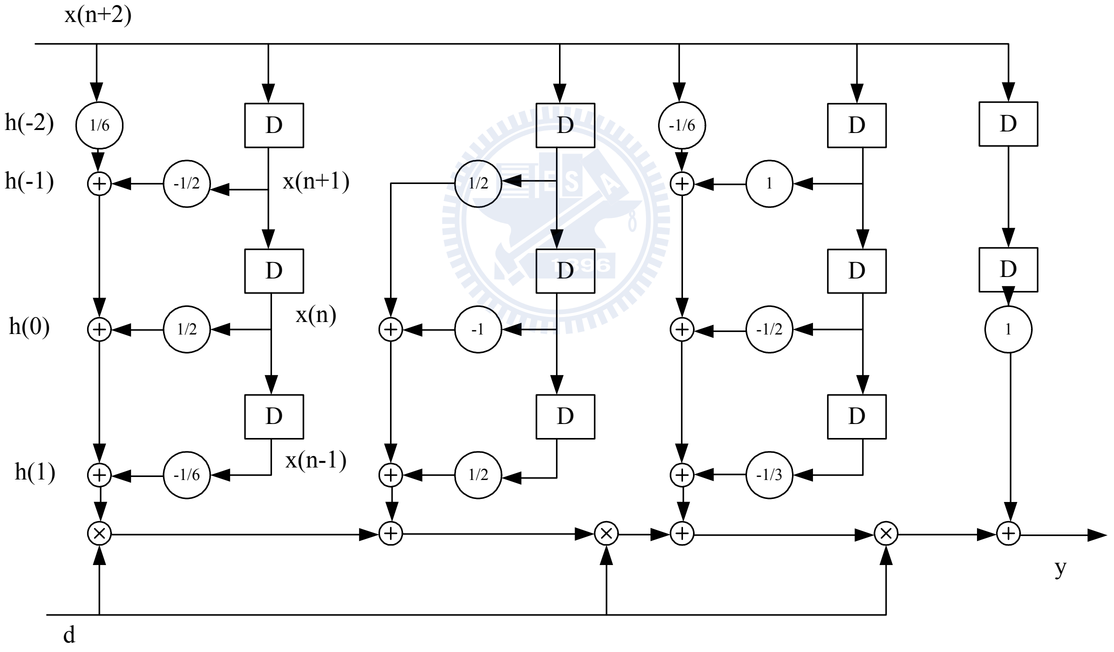


Fig. 4.9 Farrow structure of Lagrange Cubic filter

4.5 Architecture of Phase Rotator

4.5.1 CORDIC Based Rotator

Phase rotator is used to rotate each subcarrier with the angle caused by SCO effect. CORDIC based rotator is using another mode in CORDIC to transform from phase to complex value instead of general mode to transform from complex value to phase. CORDIC is used to generate the phase from the complex number in detector, but it can transfer in another way, just adding little hardware to check the sign value from the angle instead from complex value as shown in Fig. 4.10. In the following CORDIC, there is another control signal to determine which mode is used. In fact, if we select the phase to complex value mode, in each stage, we determine the phase is higher or lower, and then rotate complex number by adders and shifter from Eqn. (4.6).

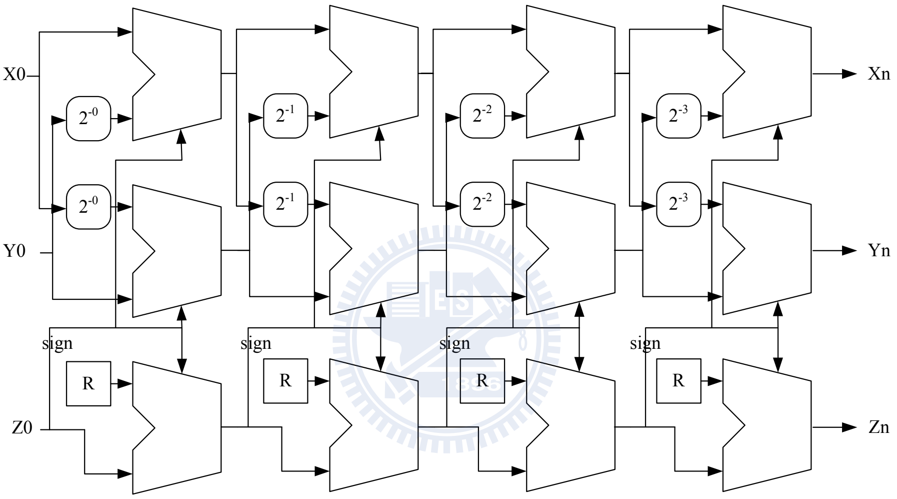


Fig. 4.10 CORDIC with phase to complex number mode

4.6 Architecture of Two Proposed Solutions

4.6.1 Compensation in Time Domain

Compensation in time domain method as shown in Fig. 3.7 needs a feedback loop to correct the signal as mentioned in Section 3.3.1, so it is not a real-time compensation method. The un-fair comparison effect in Section 3.3.1 can not be cancelled easily without adding lots of hardware. Fig. 4.11 shows the hardware architecture to eliminate both non real-time and un-fair comparison effect. However, it needs two FFTs and the hardware cost of FFT is very high .Thus, a reasonable architecture of compensation in time domain is shown at Fig. 4.12. It uses only one interpolator, FFT, one register file, and one detector. However, the performance of this architecture is about half of the perfect compensation for the un-fair comparison effect.

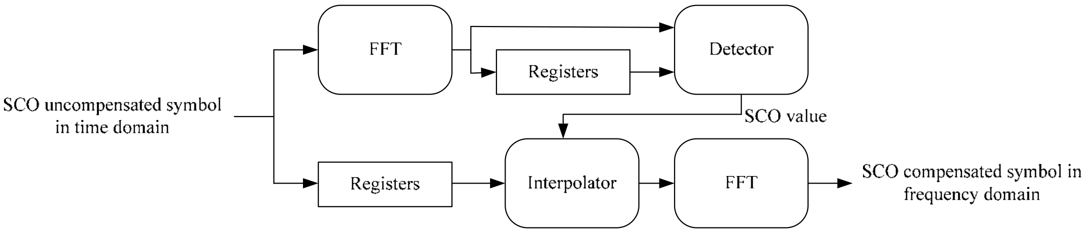

FFT
Detector
Registers
SCO uncompensated symbol in time domain
SCO value
Registers
Interpolator
FFT
SCO compensated symbol in frequency domain

Fig. 4.11 Architecture of compensation in time domain with two FFTs

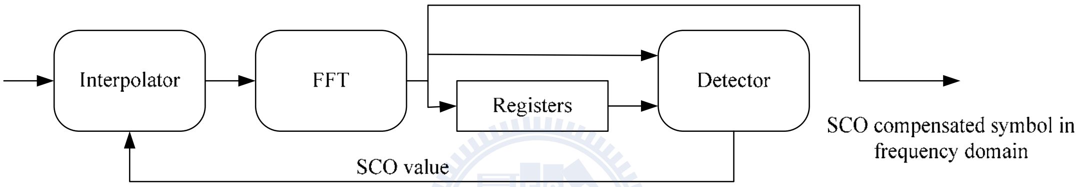


Fig. 4.12 Architecture of compensation in time domain

4.6.2 Compensation in Frequency Domain

The architecture of compensation in frequency domain is shown at Fig. 4.13. It needs register files to delay and store previous symbol subcarriers in frequency domain. It is a real-time compensation. The hardware of detector and rotator can be shared for they are almost the same.

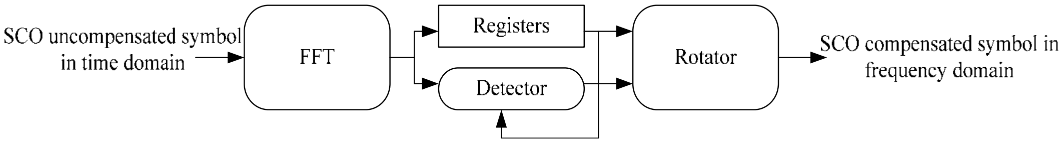

SCO uncompensated symbol in time domain
FFT
Registers
Detector
Rotator
SCO compensated symbol in frequency domain

Fig. 4.13 Architecture of compensation in frequency domain

4.6.3 Complexity Comparison

The complexity of compensation in frequency domain is much lower than compensation in time domain. From the discussion in Section 3.2, the most accurate way of compensation is the architecture of compensation in time domain with two FFTs, and it needs 1 symbol register, 1 pilot register, 2 FFTs, 1 CORDIC and 1 Lagrange Cubic filter. The worst compensation performance is the architecture of compensation in time domain with one FFT, and it needs 1 pilot registers, 1 FFT, 1 CORDIC and 1 Lagrange Cubic filter. The most efficient architecture is compensation in frequency domain, it needs 1 symbol register, 1 FFT and 1 CORDIC.

It is possible to reduce the memory usage in Fig. 4.13, and the way of memory reduction is mentioned in Section 3.3.3. The memory-efficiency architecture is to replace symbol register with pilot register which means only to store pilot values instead to all subcarriers in a symbol, but it will become a non real-time compensation which means using pre-symbol SCO value to fix current symbol. This architecture works when the variance of SCO of two successive symbols is within 6 ppm, as discussed in Section 3.3.3. Fig. 4.14 shows the modified architecture. It needs 1 pilot register, 1 FFT and 1 CORDIC.

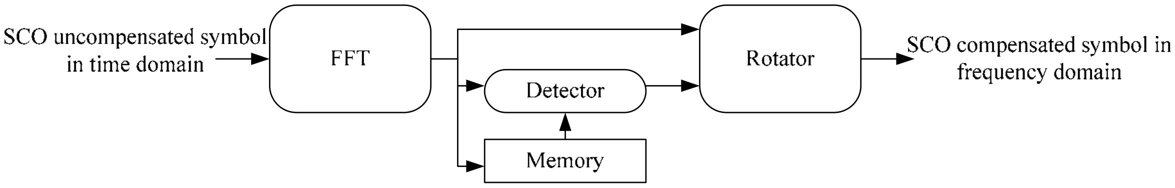


Fig. 4.14 Architecture of compensation in frequency domain with reduced register

4.6.4 Reduced Complex Multiplier Architecture

A general function for a complex multiplication expansion is shown at Eqn. (4.17). It needs two multipliers and one adder to calculate the real part and two multipliers and one adder for the imaginary part. Since the hardware cost of a multiplier is much large than the hardware cost of an adder. We can replace the calculation of imaginary part by the result of real part, and thus we need only one multiplier for imaginary part. The expansion can be further rewrite as Eqn. (4.18) to share the hardware of calculation for the real part. The reduced hardware architecture of complex multiplier is shown as Fig. 4.15.

$$(A+Bi)\times(C-Di) \tag{4.17}$$

$$\begin{aligned} &=(A\times C+B\times D)+i(B\times C-A\times D) \\ &=(A\times C+B\times D)+i\left[(A+B)\times(C-D)-A\times C+B\times D\right] \end{aligned} \tag{4.18}$$

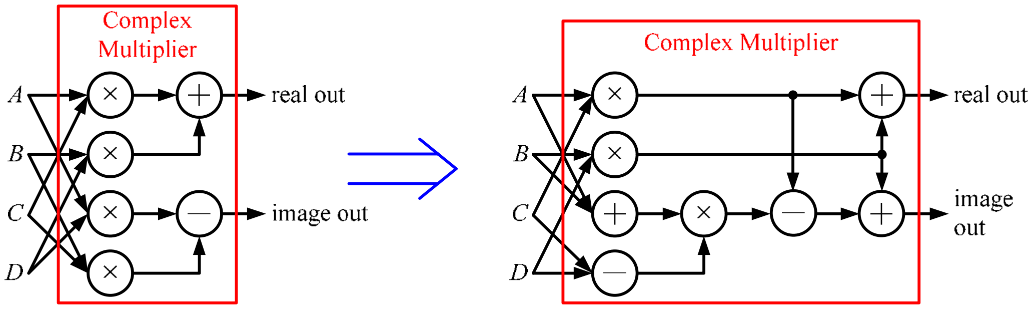


Fig. 4.15 Complex multiplier reduction

4.6.5 Conclusion

The compensation and estimation of SCO needs CORDIC structure. We choose reduced complex multiplier architecture to realize CORDIC in our system. The discussion of hardware cost in Section 4.6.3 can be replaced by the adders and multipliers. Thus, we replace the hardware of CORDIC and Lagrange Cubic filter by adders and multipliers for comparison. Table 4.2 shows the hardware complexity of each proposed solution of architecture.

Table 4.2 Hardware cost comparison of three architectures

Architectures Hardware cost	Time domain compensation	Frequency domain compensation	Frequency domain compensation with reduced memory
Adders	97	60	60
Multipliers	9	0	0
FFTs	2x1024size	1x1024size	1x1024size
Memory	1242x20bits	1034x20bits	138x20bits
Shifters	48	36	36

Chapter 5

Conclusions

In this thesis, we build an 802.16e simulation platform using C with floating point. This simulation platform contains symbol boundary detection, carrier frequency synchronization, sampling timing synchronization and frequency domain channel estimation. All synchronization loop works in digital domain. SCO compensation in 802.16e suffered some challenge from the diversity of pilots, and we take longer observation time to avoid it. In the simulation platform, although the phenomenon of estimating SCO is complicated in frequency domain, there is a simple way to estimate and compensate by phase rotation, and the errors between the simple method and the ideal method are small enough (less than 5%). Thus, SCO is estimated and compensated by CORDIC-base derotator. Besides, the architecture of SCO compensation can be further simplified if we choose the memory-efficiency version to implement. It works when SCO variance is below 6 ppm. This memory-efficiency algorithm reduces the usage of memory on subcarriers, and only pilots are stored in the memory. Thus, the usage of memory is reduced by 85 %.

Reference

[1] Part 16: Air interface for Fixed Broadband Wireless Access Systems, IEEE Std 802.16-2004, Oct. 2004

[2] Part 16: Air interface for Fixed and Mobile Broadband Wireless Access Systems, IEEE Std 802.16e-2005, Dec. 2005

[3] J.G. Andrews, A. Ghosh, R. Muhamed, Fundamental of WiMAX, 1st ed. Prentice Hill, 2007

[4] A. V. Oppenheim and R. W. Schafer, Discrete-time Signal Processing, 2nd ed. Prentice Hill, 1999

[5] M. Speth, S. Fechtel, G. Fock and H. Meyr, "Optimum Receiver Design for Wireless Broad-band Systems Using OFDM Part I," *IEEE Trans. Commun.*, vol. 47, no 11, pp.1668-1677, Nov. 1999

[6] M. Speth, S. Fechtel, G. Fock and H. Meyr, "Optimum Receiver Design for OFDM-Based Broadband Transmission Part II: A Case Study," *IEEE Trans. Commun.*, vol.49, no. 4, pp. 571-578, Apr. 2001

[7] R. V. Nee, OFDM for Wireless Multimedia Communication, 1st ed. Artech House, 2000

[8] N. Frederik, H. Francois and V. P. Liesbet, "Joint Carrier and Clock Offset Compensation in a Mobile 802.16e OFDMA Communication System," *IEEE Conf., Commun.*, pp. 51-60, Nov. 2006

[9] R. Venkataramani and Y. Bresler, "Sampling Theorems for Uniform and Periodic Nonuniform MIMO Sampling of Multiband Signals," *IEEE Trans. Signal Processing*, vol. 51, no 12, pp. 3152-3163, Dec. 2003

[10] Y.X. Xu, L. Dong and C. Zhang, "Sampling Clock Offset Estimation Algorithm Based on IEEE 802.11n," *IEEE Conf. Networ.*, pp.523 – 527, Apr. 2008

[11] J. Terry and J. Heiskala, OFDM Wireless LANs: A Theoretical and Practical Guide, 1st ed. Sams, 2002

[12] S. Nahm and W. Sung, "A Fast Direction Sequence Generation Method for CORDIC Processors," *IEEE Conf., Acoust.*, vol 1, pp. 635 – 638, Apr. 1997

[13] J. E. Volder, "The CORDIC Trigonometric Computing Technique," I*RE Trans. Electron. Computers*, vol. C-8, pp. 330–334, Sept. 1959

[14] J. Duprat and J. M. Muller, "The CORDIC Algorithm: New Results for Fast VLSI Implementation," *IEEE Trans. Computers*, vol. 42, no. 2, pp. 168-178, Feb. 1993

[15] M. M. Freda, J. F. Weng, and T. Le-Ngoc, "Joint Channel Estimation and

Synchronization for OFDM Systems," *IEEE Conf. Vehicu.*, vol. 3, no. 3, pp. 1673-1677, Sept. 2004

[16] T. I. Laakso, V. Valimaki, M. Karjalainen and U.K. Laine, "Splitting the Unit Delay," *IEEE Signal Processing Magazine*, vol. 13, no.1, pp. 30–60, Jan. 1996

[17] C. W. Farrow, "A Continuously Variable Digital Delay Element" *IEEE International Symposium on Circuits and Systems*, June 1988, pp. 2641 – 2645