國立交通大學
電子工程學系　電子研究所碩士班

應用於地面數位電視廣播系統之時序同步化設計

Timing Synchronization for DVB-T System

指導教授 ： 李鎮宜 教授

學生 ： 管成偉

中華民國九十三年九月

應用於地面數位電視廣播系統之時序同步化設計

研究生：管成偉　　指導教授：李鎮宜

摘　要

對於正交分頻多工的通訊系統而言，時序同步化是非常重要的，影響傳輸效能甚大，本論文的目的是設計一套時序同步化機制其中包含符號同步和取樣頻率同步。在此我們選擇地面數位電視廣播系統作為研究主題，我們在 Matlab 上建立了一個完整的模擬平台其中包含傳送端、通道模型以及接收端，我們會先對平台做一個完整介紹，並且把各種通道模型做一份完整的分析。

在同步系統的第一步，我們利用週期尖峰的特性偵測傳輸模式與防護區間的長度，之後我們對許多簡略符號同步的演算法做了一個詳盡的比較，為了找到更精確的符號邊界，我們提出了一套具有創新的尖峰偵測策略的低複雜度精準符號同步方法。在取樣頻率同步的部分，我們提出一套全數位的同步機制，其中包括一個內含內插與降頻功能的重新取樣器，以及新提出的以線性最小平方法理論為基礎的取樣頻率飄移 偵測設計，再加上可多級變動係數的循環濾波器。我們會秀出整個系統的效能以及對都卜勒效應的容忍度，最後，我們會對每塊模組做硬體架構上的說明。

Timing Synchronization for DVB-T System

Student : Cheng-Wei Kuang Advisor : Dr. Chen-Yi Lee

Institute of Electronics Engineering

National Chiao Tung University

ABSTRACT

The synchronization of symbol timing and sampling clock is very important for the OFDM communication system. In this thesis, a whole timing synchronization scheme including symbol synchronization and sampling clock synchronization is presented. The DVB-T (Digital Video Broadcasting – Terrestrial) system is chosen for the research topic. A complete simulation platform which contains transmitter, channel and receiver is built in Matlab. The overview of DVB-T system is presented first. Then the various channel distortions like multipath fading, Doppler, CFO and SCO will be analyzed clearly.

Employing flat peak period decision, the blind Mode/GI detection is exploited as the first stage of synchronization. A detail comparison of different coarse symbol synchronization algorithms is then presented. For more accurate symbol timing, we will illustrate the proposed low complexity fine symbol synchronization with new peak detection strategy. In sampling clock synchronization, a proposed resampler combining interpolator with decimator can reduce nearly half of power consumption. Subsequently, a SCO estimation based on LLS algorithm with the multi-stage tracking loop is proposed which improves the conventional designs. At last of this thesis, the architecture design of each block is presented respectively.

Contents

List of Tables

List of Figures

Chapter 1
Introduction

1.1 Motivation

Synchronization is one of the most important things in a communication system. In the mobile wireless channel, transmitted data suffers from several kinds of distortions caused by multipath fading, Doppler spread, AWGN, carrier frequency offset and sampling clock offset. These effects raise the difficulty of synchronization. In OFDM transmission systems, the task of synchronization consists of carrier frequency synchronization and timing synchronization. Timing synchronization is divided into symbol synchronization and sampling clock synchronization. The purpose of symbol synchronization is to find the correct position of symbol boundary. The inaccurate symbol timing can cause ISI that destroys the orthogonality of subcarriers, which degrades the system performance much. The objective of sampling clock synchronization is to adjust sampling clock frequency in the receiver. The sampling clock offset leads to the symbol timing drift. In the broadcasting communication system, the timing drift problem is more critical with respect to other packet-based system since the incoming data exists all the time. The timing drift can cause severe ISI effect if we do not perform sampling clock synchronization. In order to solve the timing synchronization problem, we choose the DVB-T (Digital Video Broadcasting – Terrestrial) system as the research topic. A complete timing synchronization scheme for DVB-T system will be proposed in this thesis.

1.2 *Introduction to ETSI DVB-T Standard*

In recent years, DTV (Digital TV) is widely adopted as the next-generation video broadcasting transmission technology. DTV can provide higher A/V quality and less transmission noise than conventional analog TV. The current developed DTV standards consist of DVB (Digital Video Broadcasting) in Europe, ATSC (Advanced Television Systems Committee) in U.S., ISDB (Integrated Services Digital Broadcasting) in Japan and DMB (Digital Multimedia Broadcasting) in China. The transmission modes of DTV include direct satellite broadcasting, cable and terrestrial broadcasting (over-the-air). In terrestrial broadcasting, particularly, video signal is transmitted against severer channel distortions such as multipath fading, co-channel interference and adjacent-channel interference. Since broadcasting transmission system is usually designed to operate within the UHF spectrum allocation for analogue transmissions, it has to provide sufficient protection against high levels of co/adjacent-channel interference emanating from existing PAL (Phase Alternative Line) / SECAM (SEquentiel Couleur Avec Memoire or sequential color with memory) services. Therefore, it is clearly that the terrestrial broadcasting has more challenges in research. The relative materials of terrestrial broadcasting systems are listed in Table 1.1

Standard	DVB-T	ATSC	ISDB-T	DMB-T
Full name	Digital Video Broadcasting for Terrestrial	Advanced Television Systems Committee	Integrated Services Digital Broadcasting for Terrestrial	Digital Multimedia Broadcasting for Terrestrial
Modulation	OFDM	8-VSB(Vestigial Sideband)	OFDM	OFDM
Area	Europe	U.S.	Japan	China

Table 1.1 Comparison of each DTV broadcasting standard

Terrestrial Digital Video Broadcasting (DVB-T) [1] has been subject to technical discussion for many years and undoubtedly been shown as a great success in delivering high quality and standard quality digital television by terrestrial means. DVB-T standard has been produced by European Telecommunication Standard Institute (ETSI) in Aug, 1997, and the second version is released in Jan, 2001. Although originating in Europe, it has been introduced in many countries around the world such as Taiwan. There are two other standards of different broadcasting modes, i.e., DVB-C (cable) and DVB-S (satellite) being specified by ETSI simultaneously. Both DVB-C and DVB-S are simpler system compared to DVB-T as listed in Table 1.2.

Broadcasting Mode	Terrestrial (DVB-T)	Satellite (DVB-S)	Cable (DVB-C)
Modulation	2K / 8K-COFDM	QPSK	16/32/64-QAM
Channel Bandwidth	5.71/6.66/7.61 MHz	26-54 MHz	2-7.86/7.96/7.92 MHz

Table 1.2 Comparisons of DVB-T, DVB-S and DVB-C

DVB-S has high channel bandwidth (26-54 MHz) and uses QPSK modulation. DVB-C applies 16/32/64 QAM modulation without convolutional code because of lower channel noise and interferences. Nevertheless, in order to provide the high data rate required for video transmission and resist severe channel distortion in DVB-T, concatenated-coded Orthogonal Frequency Division Multiplexing (OFDM) has been adopted into DVB-T in particular. OFDM is a very popular technology today due to its high data rate transmission capability with high bandwidth efficiency and its robustness to multipath distortion. It has been also chosen as the transmission technique of other communication systems such as ADSL, VDSL, XDSL, DAB and IEEE802.11a/g.

For coping with a multitude of propagation conditions encountered in the wireless

broadcasting channel, many parameters of OFDM for DVB-T can be dynamically changed according to channel conditions. The number of OFDM subcarriers can either be 2048 (2K) or 8192 (8K) so that the desired trade-off can be made between inter-symbol-interference (ISI) and Doppler spread. In the 2K mode, wide subcarrier spacing can reduce the distortion caused by Doppler frequency spread. In the 8K mode, long OFDM symbol duration can overcome large multipath fading. Other parameters like guard interval length, constellation mapping mode and coding rate of Viterbi can be also properly decided up to the broadcasting channel condition of the local area.

The transmission system is shown in Fig 1.1. It contains the blocks for source coding, outer coding and interleaving, inner coding and interleaving, mapping and modulation, frame adaptation and OFDM transmission.

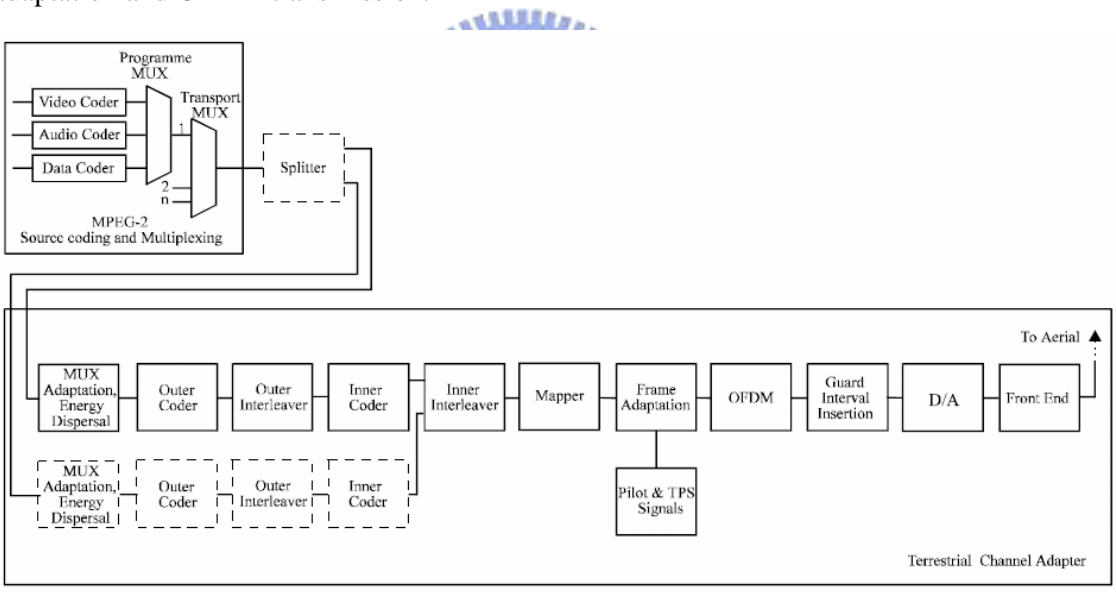


Fig 1.1 Function block diagram of DVB-T system

In the case of two-level hierarchy, the functional block diagram of the system must be expanded to include the modules shown in dashed in. The splitter separates the incoming data stream into the high-priority and the low-priority stream. These two bitstreams are mapped onto the signal constellation by the mapper and therefore the modulator has a corresponding

number of inputs.

As in the baseline systems for satellite and cable, source coding of video and sound is based on the ISO-MPEG2 standards. After the MPEG2 Transport Multiplexer, the packet may optionally be split into two data streams of different priority in case of hierarchical modulation/channel coding. Subsequently, a Reed Solomon (RS) shortened code (204, 188 t = 8) and a convolutional bytewise interleaving with depth I = 12 is applied to the error protected packets (outer interleaving). As Fig 1.1, the outer interleaver is followed by the inner coder. This coder is designed for a range of punctured convolutional codes (Viterbi), which allows code rates of 1/2, 2/3, 3/4, 5/6 and 7/8. If two-level hierarchical transmission is used, each of two parallel channel encoders has its own code rate. Afterward, the inner interleaver is block based bitwise interleaving.

The system uses OFDM transmission. All data carriers in one OFDM symbol are either QPSK, 16-QAM, 64-QAM, non-uniform-16-QAM or non-uniform-64-QAM. In addition to the transmitted data, an OFDM symbol contains scattered pilots, continual pilots and TPS (Transmission Parameter Signaling) pilots. These reference signals can be used for synchronization, channel estimation and transmission mode verification. The OFDM frame consists of 68 OFDM symbols and four frames constitute one super-frame. The frame structure involving distribution of scattered pilots is shown in Fig 1.2. Scattered pilots insert every 12 subcarriers and have an interval of 3 subcarriers in the next adjacent symbol. Continual pilots locate at fixed indices of subcarrier, which contains 177 pilots in 8K mode and 45 in 2K mode. Both scattered pilots and continual pilots are transmitted at a boosted power level of 16/9 whereas the power level of other symbols is normalized to 1.

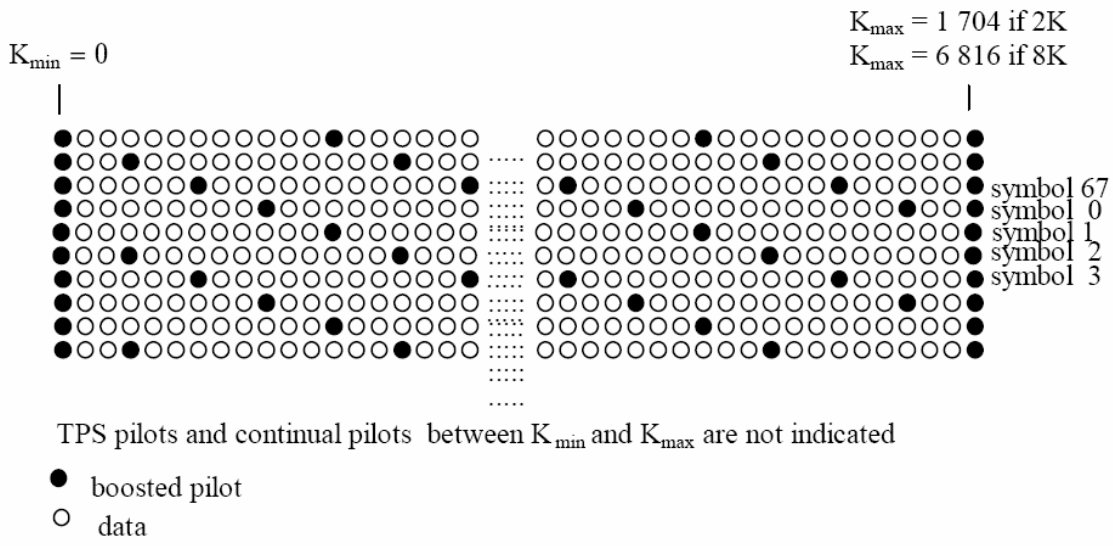


Fig 1.2 Frame structure

The TPS carriers are used for the purpose of signaling parameters related to the transmission scheme, i.e. to channel coding and modulation. The TPS is defined over 68 consecutive OFDM symbol and transmitted in parallel on 17 TPS carriers for the 2K mode and on 68 carriers for the 8K mode. Each OFDM symbol conveys one TPS bit which is differentially encoded in every TPS carriers. The TPS information contains frame number, constellation, hierarchy, code rate, guard interval, transmission mode and BCH error protection code. Unlike scattered and continual pilots, TPS pilots are transmitted at the normal power level of 1 with DBPSK modulation. The values of scattered pilots, continual pilots and TPS pilots are derived by PRBS sequence ($X^{11}+X^2+1$). The guard interval may have four values, i.e. 1/4, 1/8, 1/16 and 1/32. Guard interval 1/4 would occupy 25% of the usable transmission capacity and hence only be used in case of SFN operation with long distances between transmitter sites. In the case of smaller transmitter distances (local SFN) or non-SFN operation the smaller values of guard interval can be selected. In conclusion, DVB-T system has good flexibility for various transmission conditions, so that it becomes a successful technology for video broadcasting.

1.3 Thesis Organization

This thesis consists of 5 chapters. The reader is assumed to be familiar with OFDM theory and thus we'll just focus on timing synchronization of DVB-T system. The effect of timing error and the corresponding synchronization algorithms will be introduced in chapter 2. Chapter 3 provides the overview of simulation platform and we will explain the main blocks and the channel model. The simulation results and comparisons will be presented in this chapter. In chapter 4, we present the architecture design and discuss the considerations about hardware implementation. Finally, conclusion and future work are made in chapter 5.

Chapter 2
Timing Synchronization Algorithms

2.1 Introduction to Timing Offset

In OFDM system, timing offset would cause inter-symbol interference (ISI) which destroys the orthogonality of subcarriers. The timing offset can be divided into two parts: symbol timing offset and sampling clock offset. The symbol timing offset occurs when symbol synchronization finds incorrect OFDM symbol boundary, and sampling clock offset is caused by the difference between the sampling frequencies of the digital-to-analog converter (DAC) and the one of the analog-to-digital converter (ADC). Sampling clock offset can also lead to symbol timing drift. Unlike other packet based communication system such as 802.11a, DVB-T system is a continuous-data transmission. Therefore, sampling clock offset is a critical problem to be solved.

2.1.1 Effect of Symbol Timing Offset

The symbol synchronization of the OFDM system is to find the start of OFDM symbol, i.e. the FFT window position. Just as what is shown in Fig 2.1, we call Δ the ISI-free region. If the estimated start position of OFDM symbol is located within the ISI-free region, data will not be affected by inter-symbol interference (ISI). The effect of phase rotation caused by symbol timing offset can be easily corrected after FFT.

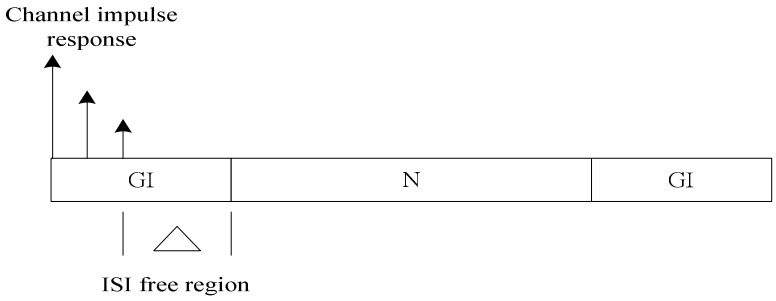


Fig 2.1 ISI-free region

Assume $x(n)$ represents received data in time domain, $X(k)$ is subcarrier after FFT operation for $x(n)$ with perfect symbol timing, and $\hat{X}(k)$ denotes subcarrier after FFT operation with symbol timing offset ε in the ISI-free region. The detail equations are demonstrated as follows.

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-i2\pi k \frac{n}{N}} \tag{2.1}$$

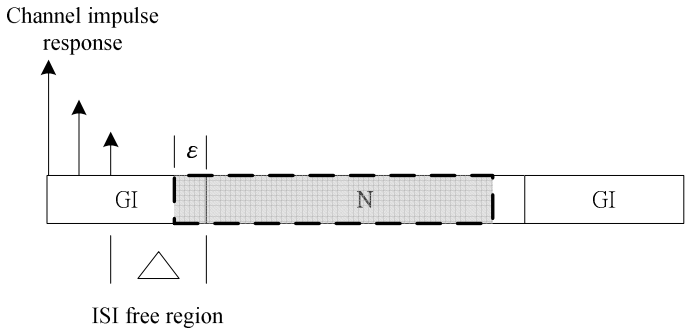
$$\hat{X}(k) = \sum_{n=0}^{N-1} x(n) e^{-i2\pi k \frac{n+\varepsilon}{N}} \tag{2.2}$$

$$\hat{X}(k) = \sum_{n=0}^{N-1} x(n) e^{-i2\pi k \frac{n}{N}} e^{-i2\pi k \frac{\varepsilon}{N}} \tag{2.3}$$

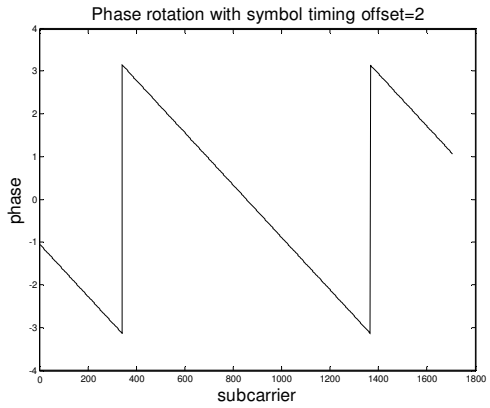
$$\hat{X}(k) = X(k) e^{i2\pi k\varepsilon / N} \tag{2.4}$$

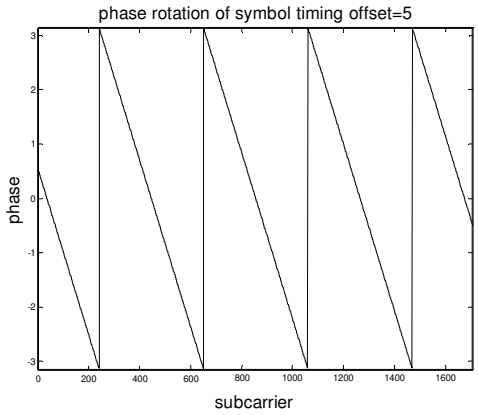
where k represents the subcarrier index, n denotes sample index in time domain, and N is the number of subcarriers in an OFDM symbol. Note the last term $e^{i2\pi k\varepsilon / N}$ in Eq(2.4), which exhibits the phase rotation. Therefore, we can conclude that the effect of symbol timing offset in the ISI-free region is phase rotation and unchanged magnitude of subcarrier, which can be compensated by equalizer completely. The phase rotation effect is shown in Fig 2.2. Fig 2.2(b) depicts the condition of symbol timing offset ε = 2 while Fig 2.2(c) shows the condition of ε = 5. As symbol timing offset ε is lager, the phase variation is severer. The additional variance of channel response due to timing error will increase the difficulty of channel estimation. In order to ease the load of channel estimation unit, the symbol timing effect should be as small as possible even the phase rotate effect can be completely corrected in

theory.



(a) Symbol timing offset ε in the ISI-free region

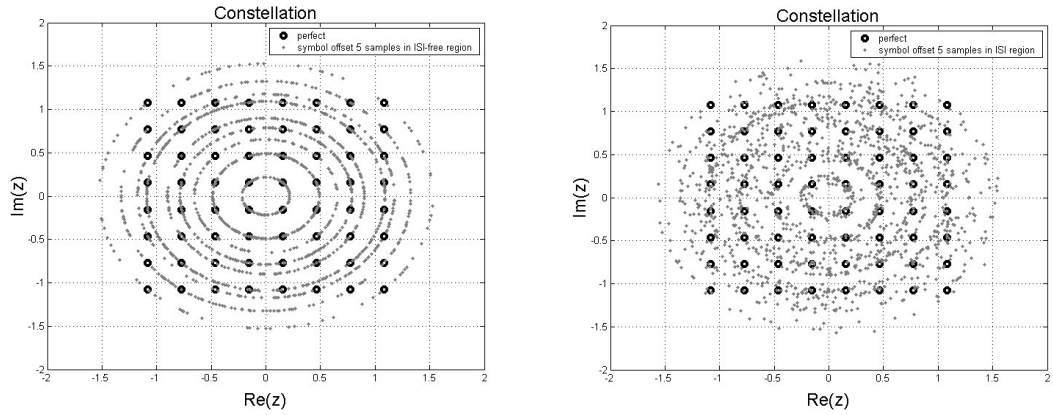




(b) Phase rotation due to symbol timing offset=2 (c) Phase rotation due to symbol timing offset=5

Fig 2.2 Phase rotation of symbol timing offset ε=2 and ε=5

On the other hand, if the estimated start position locates out of ISI-free region, the sampled OFDM symbol will contains some samples that belong to previous symbol or following symbol, which leads to the dispersion of signal constellation (ISI) and reduce system performance much. Therefore, the objective of symbol synchronization, first of all, is to avoid the estimated symbol boundary lying in ISI region and subsequently reduce the symbol timing offset as far as possible. The relative mapping constellations are depicted in Fig 2.3. Fig 2.3(a) shows the phase rotation effect due to symbol timing offset of 5 samples while Fig 2.3(b) shows the ISI effect which destroys the signal constellation heavily.



(a) Symbol offset 5 samples in the ISI-free region (b) Symbol offset 5 samples in the ISI region

Fig 2.3 Mapping constellation

2.1.2 Effect of Sampling Clock Offset

The sampling clock errors include the clock phase error and the clock frequency error. The effect of clock phase error is similar to the effect of symbol timing offset and hence we can regard the clock phase error as a fractional part of symbol timing offset. The major impact of sampling clock error is clock frequency error which causes phase rotation in frequency domain and symbol timing drift in time domain. We call the sampling clock frequency error "sampling clock offset (SCO)".

Sampling clock offset causes sampling timing change at every sample interval as shown in Fig 2.4. For example, if the sampling clock frequency is 8MHz, sampling clock offset is 10ppm, then the symbol timing has a drift of about 80 samples per second. Obviously, SCO is an important issue in broadcasting transmission system like DVB-T. Ignoring sampling clock synchronization would lead to severe timing drift.

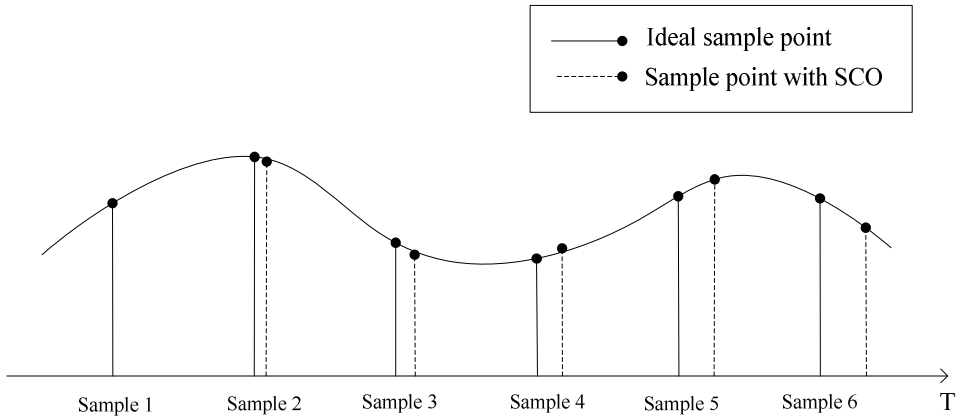


Fig 2.4 Sampling clock offset

Similar to symbol timing offset, sampling clock offset causes phase rotation in frequency domain. Furthermore, the amount of phase rotation is monotonous increasing as the symbol is conveying as shown in Fig 2.5.

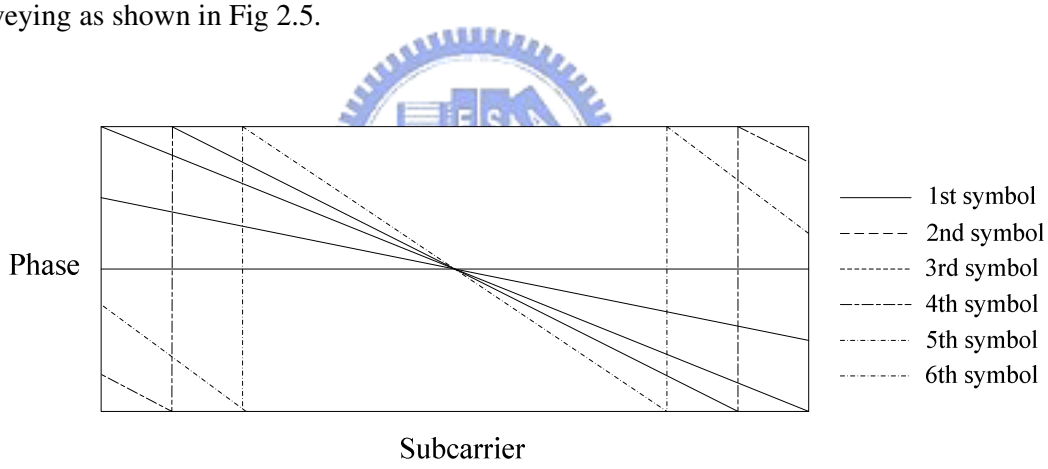


Fig 2.5 Phase rotation between consecutive OFDM symbols

To prove the phase rotation of SCO, we consider an OFDM system using IFFT with N-points. Each OFDM symbol consist of K (K < N) data subcarrier, $a_{l,k}$, where l denotes the OFDM symbol index and k denotes the subcarrier index, $-K/2 \le k \le K/2-1$, T is the sampling clock period and N_g is the number of guard interval samples. Then one OFDM symbol has total N_s samples, N_s is equal to $N + N_g$. The transmitted complex baseband signal for l-th symbol can be described by

$$s(t) = \frac{1}{N} \sum_{k=-K/2}^{K/2-1} z_{l,k} \cdot e^{\frac{j2\pi k(t-(N_g+l\cdot N_s)T)}{NT}} \tag{2.5}$$

In receiver, we assume that the carrier frequency is f' and sampling clock is T'. Therefore the carrier frequency offset Δf and the relative sampling clock offset ζ are designated as

$$\Delta f = f - f' \tag{2.6}$$

$$\zeta = (T' - T)/T \tag{2.7}$$

The l-th received symbol after sampling with the sampling clock T' and removing the guard interval can be represented as

$$\begin{aligned} r_l(n) &= e^{j2\pi\Delta f t_n} \frac{1}{N} \sum_{k=-K/2}^{K/2-1} a_{l,k} e^{\frac{j2\pi k(t_n-(N_g+l\cdot N_s)T)}{NT}} + v_l(n) \\ &= e^{j2\pi\Delta f(N_g+l\cdot N_s)(1+\zeta)T} e^{j2\pi\Delta f n(1+\zeta)T} \cdot \frac{1}{N} \sum_{k=-K/2}^{K/2-1} a_{l,k} e^{\frac{j2\pi kn}{N}(1+\zeta)} e^{\frac{j2\pi k}{N}(N_g+l\cdot N_s)\zeta} + v_l(n) \end{aligned} \tag{2.8}$$

where $t_n = (N_g+lN_s)T' + nT'$ and $v_l(n)$ is the complex Gaussian noise. Demodulation of the received samples via FFT yields the data symbol in frequency domain, $z_{l,k}$.

$$\begin{aligned} z_{l,k} &= \sum_{n=0}^{N-1} r_l(n) e^{-j2\pi kn/N} \\ &= e^{j2\pi\Delta f(N_g+l\cdot N_s)(1+\zeta)T} e^{j\frac{2\pi k}{N}(N_g+l\cdot N_s)\zeta} \cdot \alpha \cdot a_{l,k} + ICI + n_l(k) \end{aligned} \tag{2.9}$$

In acquisition process, the residual CFO has been estimated and pre-compensated in the time-domain. The ICI produced by the remaining CFO is smaller compared to Gaussian noise, which can be considered as a complex zero mean Gaussian noise. α is an attenuation factor which is close to 1. Considering the frequency selective fading channel and neglecting the factor α, $z_{l,k}$ is modified as

$$z_{l,k} = e^{j2\pi\Delta f(N_g+l\cdot N_s)(1+\zeta)T} e^{j\frac{2\pi k}{N}(N_g+l\cdot N_s)\zeta} \cdot H_l(k) \cdot a_{l,k} + n_l(k) \tag{2.10}$$

As we can see, the phase rotation occurs. The rotated phase is

$$\varphi_l(k) = 2\pi\Delta f(N_g + l\cdot N_s)(1+\zeta)T + \frac{2\pi k}{N}(N_g + l\cdot N_s)\zeta + \phi_l^H(k) \tag{2.11}$$

where $\phi_l^H(k)$ is the phase of fading channel $H_l(k)$.

If the channel is a slowly fading channel ($\phi_l^H(k) \approx \phi_{l-1}^H(k)$), the difference of rotated phases between two adjacent symbols is represented as

$$\begin{aligned} \varphi'_l(k) &= \varphi_l(k) - \varphi_{l-1}(k) \\ &= 2\pi\Delta f N_s T + 2\pi\Delta f N_s T\zeta + \frac{2\pi N_s k\zeta}{N} \\ &\approx 2\pi\Delta f N_s T + \frac{2\pi N_s k\zeta}{N} \end{aligned} \tag{2.12}$$

We can ignore the term $2\pi\Delta f N_s T\zeta$ since the SCO is usually less than $1.0\text{x}10^{-4}$. As Eq(2.12) indicates, CFO causes mean phase error as well as SCO causes linear phase error between two adjacent symbols.

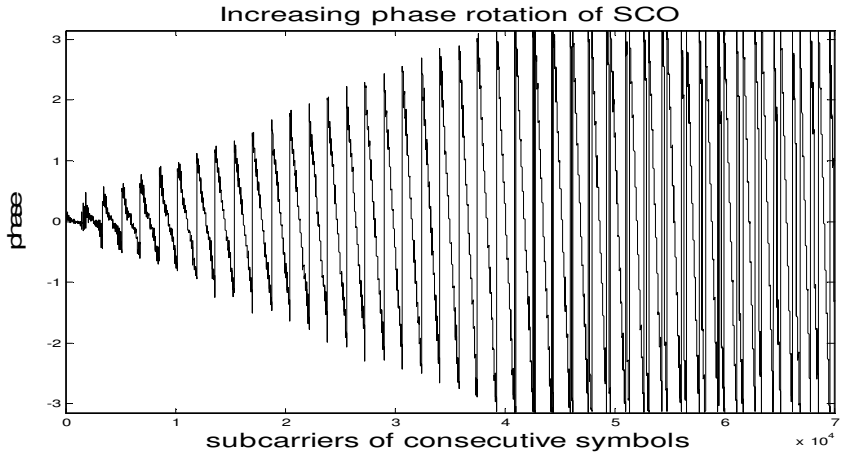


Fig 2.6 Phase rotation due to timing drift

Fig 2.6 demonstrates the phase rotation of timing drift due to sampling clock offset. In the former symbols, the total amount of phase rotation is limited in 2π (rads) since the drift point is less than one sample. After symbol timing drift exceeding one sample, phase rotation becomes severer increasingly. Regardless of the case of symbol timing drifting into ISI region, the violent phase variation still reduce the performance of channel estimation. If symbol timing drifts out of ISI-free region, inter-symbol interference is produced and hence system performance degrades much.

2.2 *Symbol Synchronization*

The purpose of symbol synchronization is to find the correct position of symbol boundary. Received symbol should synchronize to the first arriving path in order to take full advantage of the useful guard interval for the pre-FFT acquisitions. The inaccurate symbol timing caused by symbol synchronization error and sampling clock offset can induce ISI (inter-symbol interference) which destroys the orthogonality of subcarriers. The symbol synchronization process contains three parts: mode/GI detection, coarse symbol synchronization and fine symbol synchronization. In the first stage of synchronization flow, blind mode/GI detection must be done prior to the following synchronization operations since the receiver has no information about transmission mode and guard interval length of the received data. In general, to get precise symbol timing, we must divide the symbol synchronization into two parts: coarse symbol synchronization and fine symbol synchronization.

The goal of coarse symbol synchronization is to detect a coarse symbol boundary in the time domain before the FFT operation. After mode/GI detection, the transmission mode and guard interval (GI) length are well know and thus the cyclic property of GI can be adopted in coarse symbol synchronization. However, the coarse symbol synchronization is not exact enough so that the fine symbol synchronization of post-FFT operation is required in the frequency domain. Fine symbol synchronization is not only to estimate more accurate symbol timing but to ensure that the symbol timing do not drift into ISI region.

2.2.1 Mode / GI Detection

In order to perform timing and frequency synchronization as well as channel estimation, both the guard interval length and the correct number of subcarriers have to be determined.

Consequently, Mode/GI detection must be done prior to synchronization and channel estimation. In fact, there are few approaches to blind Mode/GI detection in the relative materials. That's probably because the transmission mode and GI length are assumed to known information in their researches. In particular, [5] proposes a blind Mode detection using variation-to-average ratio of moving sum but lacks GI detection. We therefore propose a joint Mode/GI detection method for the case of blind reception. Mode/GI detection can exploit the cyclic property of guard interval and then use maximum correlation method with minimum parameter GI = 1/32. Maximum correlation algorithm correlates cyclic prefixed part and useful part (where the guard interval is copied from) and then applies a moving window to seek the peak of moving sum. In order to ease the threshold decision, the normalization process is adopted in maximum correlation algorithm. The original correlation result divides its own power in addition. As a result, the normalized maximum correlation algorithm is derived as Eq(2.13).

$$K_{est} = \arg \max_{k} \frac{\left| \sum_{i=0}^{\frac{N_1}{32}-1} r(k-i) \cdot r^*(k-i-N) \right|}{\left| \sum_{i=0}^{\frac{N_1}{32}-1} r(k-i) \cdot r^*(k-i) \right|} \tag{2.13}$$

The received time domain sample is denoted by $r(k)$ and K_{est} is the estimation output of moving sum. Because of the normalization process, the moving sum is distributed in the interval of [0, 1]. The example of 2K moving sum operation is illustrated in Fig 2.7.

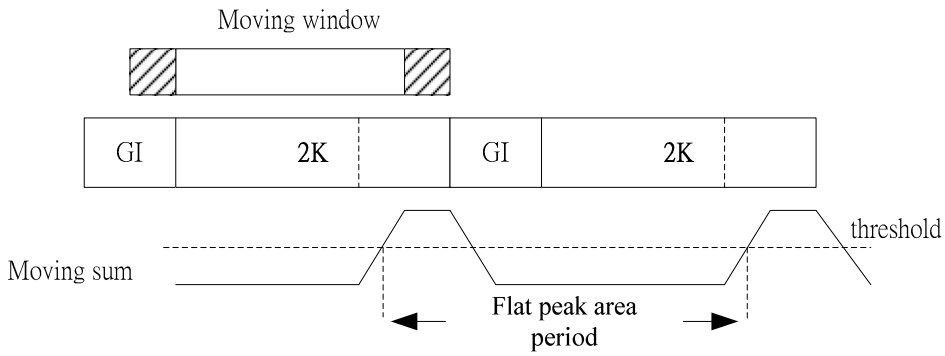


Fig 2.7 Periodic flat peak area

As we can see, the moving sum of applying moving window with GI = 1/32 causes a flat peak area if the mode selection is correct. Nevertheless, If 2K moving window is used in 8K transmission data, there would be no peak area appeared. The period of flat peak area in 2K mode can be either 2K*(1+1/4), 2K*(1+1/8), 2K*(1+1/16) or 2K*(1+1/32). To estimate the period of the peak area can obtain the information of GI length. First of all, a proper threshold has to be decided and then find the rising edge of flat peak area. The proper threshold would depend on channel if we do not use normalization. That's why the normalization process is introduced in Mode/GI detection. As a result, we calculate the period of rising edge and determine which case is rather close to the resulting estimated period. In summary, the blind Mode/GI detection is divided into two stages. First stage is 2K mode detection. If the peak area is detected, the transmission mode is therefore 2K and GI length can be derived by estimating the period of peak area. If no peak area appeared in first stage, the second stage of 8K mode detection following turns on. Similarly, the period of flat peak area in 8K mode is either 8K*(1+1/4), 8K*(1+1/8), 8K*(1+1/16) or 8K*(1+1/32).

The probability of false period determination is very small because the four cases of candidate periods have large difference of at least 64 samples with respect to others. In the multipath fading channel, the position of flat peak area will delay several samples and depend on the mean excess delay of multipath delay profile. However, the delay position will not

affect the determination of period so that this algorithm can be robust even in the strong multipath fading channel with low SNR condition.

2.2.2 Coarse Symbol Synchronization

The true design goal for coarse symbol synchronization is not to achieve the highest possible accuracy, but to meet the requirements of following operation such as AFC (automatic frequency control) and clock recovery process with a minimum implementation cost and fastest process time. The consideration for choosing suitable algorithm is to detect a coarse symbol boundary lying ISI-free region and to be tolerable to large frequency offset and potentially large sampling clock frequency deviations during acquisition. There are a lot of algorithms concerning coarse symbol synchronization. Almost all of algorithms exploit cyclic correlation based method such as maximum correlation [3], minimum mean square error [4], modified maximum likelihood [7] and double correlation [10]. In [8], a simple estimator of the minimum power difference is adopted. A joint coarse symbol synchronization and frequency acquisition is proposed in [9]. Three major synchronization algorithms of maximum correlation (MC), maximum likelihood (ML) and minimum mean square error (MMSE) are exhaustively analyzed in [6]. There are three algorithms to be discussed and compared as below.

a) Maximum Correlation (MC)

$$K_{est} = \arg \max_{k} \left| \sum_{i=0}^{Ng-1} r(k-i) \cdot r^{*}(k-i-N) \right| \tag{2.14}$$

Similar to Mode/GI detection, first algorithm of coarse symbol synchronization uses correlation method based on guard interval, which is denoted by Maximum Correlation algorithm. This algorithm is commonly used in GI-based symbol synchronization or frame synchronization of other transmission systems. Unlike Mode/GI detection, the length of the moving sum is the same as the length of guard interval in order to get best performance. The

operation is illustrated in Fig 2.8.

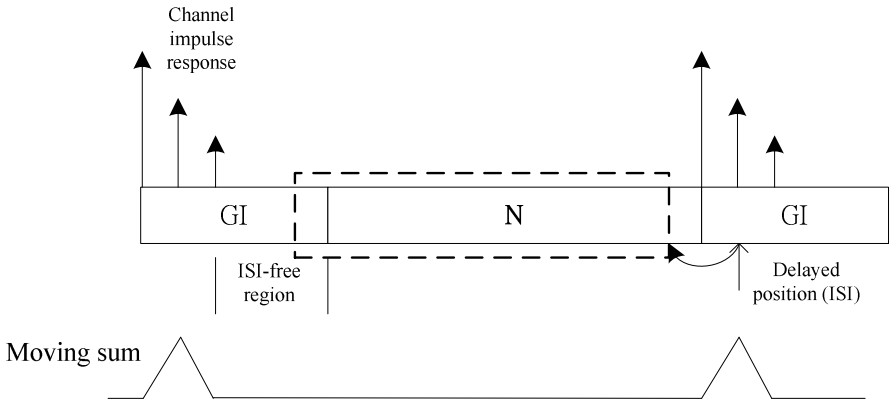


Fig 2.8 Delayed peak of moving sum

Due to multipath fading channel, the peak of moving sum will locate at a delayed position corresponding to mean excess delay of channel. The delayed position will cause the symbol boundary lie in ISI region. Therefore, a number of samples must be reserved for shifting forward while we decide the symbol boundary. Considering the Rayleigh channel specified by standard, which has a mean excess delay of 13 samples as a worst case, the shifting number of 20~30 samples should be applied in order to ensure the symbol boundary is safe in various types of channel. This maximum correlation method can resist the effect of large CFO and SCO so that the performance is acceptable. The major advantage is minimum implementation cost.

b) Normalized Maximum Correlation (NMC)

$$K_{est} = \arg \max_{k} \frac{\left| \sum_{i=0}^{Ng-1} r(k-i) \cdot r^{*}(k-i-N) \right|}{\left| \sum_{i=0}^{Ng-1} r(k-i) \cdot r^{*}(k-i) \right|} \tag{2.15}$$

Referring to the design of Mode/GI detection, the same algorithm except for different moving window length is applied in coarse symbol synchronization. The full guard interval length is taken as moving window length in place of Eq(2.13). The advantage of this algorithm is easy to set threshold for finding maximum peak because the moving sum is

normalized to 1. The accuracy is close to maximum correlation algorithm.

c) Minimum Mean Square Error (MMSE)

This algorithm is simplified from Maximum Likelihood (ML) algorithm [6]. The ML algorithm is written as

$$K_{est} = \arg \max_{k} \left(\left| \sum_{i=0}^{Ng-1} w_{k,1} \cdot r(k-i) \cdot r^{*}(k-i-N) \right| - \sum_{i=0}^{Ng-1} w_{k,2} \cdot \left(\left| r(k-i) \right|^{2} + \left| r(k-i-N) \right|^{2} \right) \right) \quad (2.16)$$

where $w_{k,1}$ and $w_{k,2}$ are parameters corresponding to the characteristics of transmitted data and channel. The basic concept of ML algorithm is to derive the log-likelihood function and obtain the ML solutions in an approximation value. The detail demonstrations can refer to [6].

Since ML algorithm requires known channel characteristic in advance, the MMSE algorithm approximates the parameters to a practical form. In fact, $w_{k,1}$ is close to 1 as well as $w_{k,2}$ is close to 1/2 so that the resulting algorithm can be rewritten as

$$K_{est} = \arg \max_{k} \left(\left| \sum_{i=0}^{Ng-1} r(k-i) \cdot r^{*}(k-i-N) \right| - \frac{1}{2} \sum_{i=0}^{Ng-1} \left(\left| r(k-i) \right|^{2} + \left| r(k-i-N) \right|^{2} \right) \right) \quad (2.17)$$

The algorithm is also similar to MC algorithm except for the second term in Eq(2.17). The difference between MMSE and MC is that we subtract its own mean power from original correlation power. Note the value of K_{est} is negative so that the perfect estimation value is 0. These three algorithms will be compared in Chapter 3.

2.2.3 Scattered Pilot Mode Detection

Before fine symbol synchronization and other operations in tracking mode, another acquisition operation has to be proceeded, which is scattered pilot mode detection. It is known that the distribution of scattered pilots has four modes.

$$k = K_{\min} + 3 \times (l \mod 4) + 12p \mid p \text{ int} eger,\ p \geq 0,\ k \in [K_{\min}; K_{\max}] \quad (2.18)$$

The four scattered pilot modes are drawn in Fig 2.9.

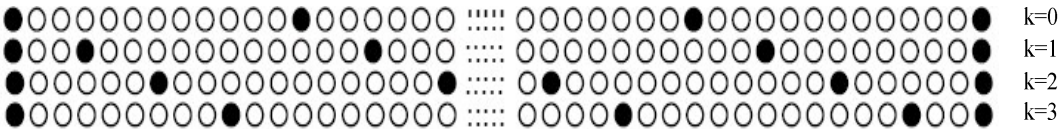
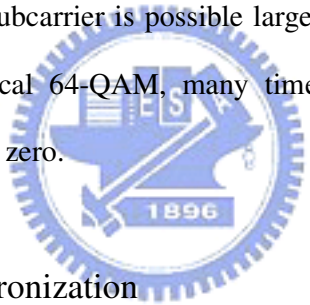


Fig 2.9 Four modes of scattered pilot position

The proposed scattered pilot mode detection exploits the property of boosted power level scattered pilots. Since the power level of scattered pilots is 16/9 while other data subcarrier is 1, we take one OFDM symbol and divide the subcarriers into 4 groups. Afterward, we accumulate the power of subcarrier belong to each group respectively as shown in Eq(2.19)

$$SP = \arg \max_{k} \left(\sum_{i=0}^{N/12-1} z(3\cdot k + 12\cdot i)\cdot z^{*}(3\cdot k + 12\cdot i) \right) \quad k = 0,1,2,3 \qquad (2.19)$$

Although the power of each subcarrier is possible larger than 16/9 such as maximum power level of 7/3 in non-hierarchical 64-QAM, many times of accumulations make the false detection rate almost reduce to zero.



2.2.4 Fine Symbol Synchronization

Since the coarse symbol synchronization is not able to provide the accuracy needed, a rather exact algorithm must be adopted in frequency domain. The algorithm requires that sampling and carrier frequency are already synchronized. Hence fine timing will be the last task in the synchronization scheme.

After the coarse symbol synchronization, the residual symbol timing offset ε becomes small and the symbol boundary locates in ISI-free region. We can assume this timing error is introduced by the physical channel whose first path time delay is $\varepsilon \cdot T$. The effect of path delay is illustrated in Fig 2.10.

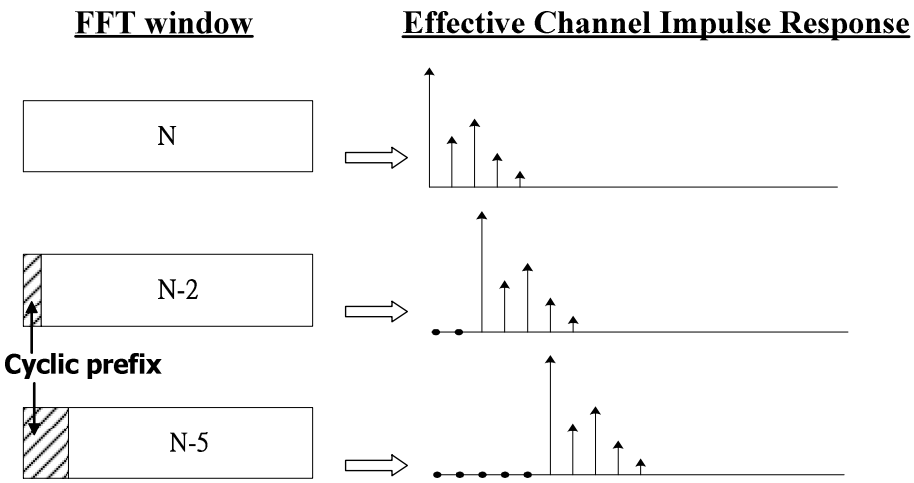


Fig 2.10 Effective channel impulse response due to inaccurate FFT window

As Fig 2.10 is depicted, the symbol timing offset in ISI-free region causes path time delay of effective channel impulse response. We assume that the signal is transmitted over multipath fading channel characterized by

$$h(\tau,t)=\sum_{l=0}^{L-1}h_l(t)\cdot\delta(\tau-\tau_l) \tag{2.20}$$

where $h_l(t)$ are the path complex gains, τ_l are the path time delay, and L is the total number of paths. Then the effective channel model due to symbol timing offset changes to be

$$h(\tau,t)=\sum_{l=0}^{L-1}h_l(t)\cdot\delta(\tau-\tau_l+\varepsilon\cdot T) \tag{2.21}$$

Therefore, the fine symbol timing can be maintained by acquiring the effective channel impulse response (CIR) and then estimating the main path delay. The path delay estimation task utilizes the channel frequency response (CFR) estimated by channel estimation unit and subsequently performs an IFFT to transform the CFR to CIR.

Before detail discussion of path time delay in CIR estimation, channel estimation should be introduced in advance. In channel estimation design, 2-D interpolation is generally used because of its robustness for mobile time-variant channel caused by Doppler spread. In 2D interpolation, channel gain estimations at scattered pilots are first interpolated over time-dimension so that three OFDM symbols must be kept in buffers. Then linear

interpolation is adopted in two adjacent symbols with same scattered pilot modes for estimating the channel gain of scattered pilot within the interval. Afterward, all other channel responses are obtained by frequency-dimension interpolation. This 2-D interpolation method deserves to be applied in DVB-T system even though large memory requirement. The time-dimension interpolation can overcome the severe time-variant channel and channel frequency response can be estimated effectively.

After time-dimension interpolation, a total of $K_{max}/3+1$ sub-sampled channel gain of scattered pilots are available. To provide a sufficiently accurate estimation, a zero padded IFFT of size *N/2* must be used. The operations of fine symbol synchronization are illustrated in Fig 2.11.

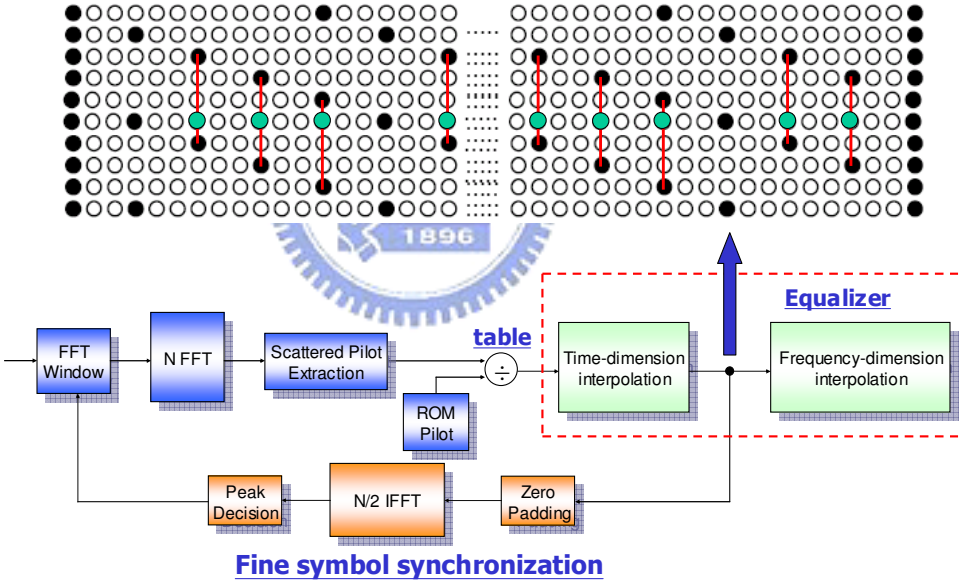


Fig 2.11 Structure of fine symbol synchronization

Fig 2.11 shows the process of fine symbol synchronization including zero padding, *N/2* IFFT and peak decision. The fine symbol synchronization utilizes the CFR of $K_{max}/3+1$ scattered samples after the time-dimension interpolation and then performs zero padding to size N/2. The changes of spectrum due to downsampling and zero padding are drawn in Fig 2.12 and Fig 2.13 respectively.

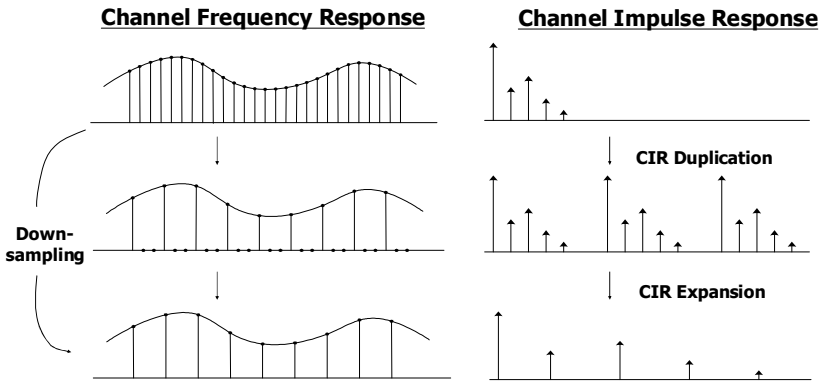


Fig 2.12 Downsampling of channel frequency response

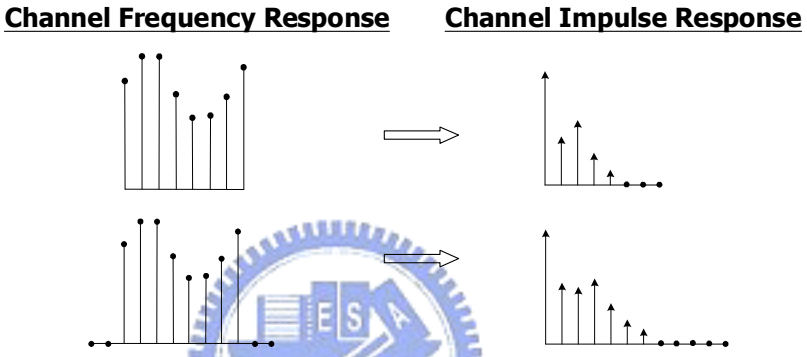


Fig 2.13 zero-padding of channel frequency response

Fig 2.12 shows a basic concept of downsampling in frequency domain. The resulting time domain response after sampling in frequency domain is CIR duplication. Then downsampling causes CIR expansion so that the original CIR can be computed with a little aliasing. It has been known that low pass filtering has to be done prior to downsampling for avoiding aliasing. However, the effect of aliasing is slight because the power of CIR usually centralizes in the prior paths and hence the low pass filtering can be neglected if the downsampling rate is not too large.

Fig 2.13 illustrates the effect of zero-padding. We can assume the zero padding in frequency domain as the spectrum compression and hence CIR expands according to the ratio of compression in spectrum. In our design, the channel gain of subcarrier is zero-padded from N/3+1 to N/2. In terms of spectrum, we can regard as spectrum compression with a ratio of

2/3 and thus CIR will expand with a ratio of 3/2. Therefore, the resolution of estimated CIR is 2/3. The resulting sampling resolution is

$$T' = T\frac{N}{3N_{est}} = 2/3T \tag{2.22}$$

In order to promote the peak detection, the square operation is performed with the resulting estimated CIR. After the square operation, the difference between channel impulses will be more apparent and hence the probability of wrong peak detection can be reduced. As a result, the expression of peak detection can be represented as

$$\hat{S}_k = |h_k|^2 \tag{2.23}$$

$$\hat{\delta} = \frac{2}{3}\arg\max_k(\hat{S}_k) \tag{2.24}$$

It's obvious that the fine symbol synchronization with N/2 IFFT method dominates the complexity of overall synchronization system. Therefore, we proposed a low complexity solution of fine symbol synchronization. The size of IFFT can be reduced by downsampling the estimated channel frequency response. However, the aliasing will occur if the downsampling rate is too large. In order to eliminate aliasing, we apply a 16-point average window as lowpass filter in advance. We then downsample the CFR by 16 so that the complexity can be reduced substantially. The proposed low complexity fine symbol synchronization design is depicted as Fig 2.14. The simulation results will be shown in Chapter 3.

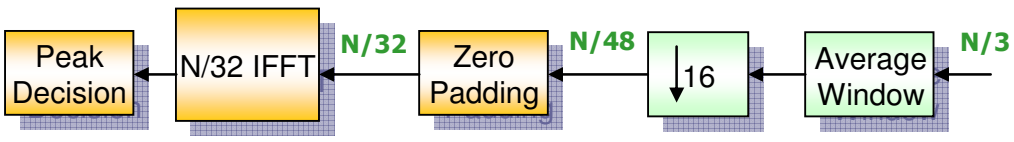


Fig 2.14 Proposed low complexity fine symbol synchronization

2.3 *Sampling Clock Synchronization*

The purpose of the sampling clock synchronization is to alter the sampling frequency and sampling phase. If the sampling timing recovery is properly operating, it will provide the downstream processing blocks with the samples at the highest SNR available. The proposed sampling clock synchronization is an all-digital timing recovery loop which is shown in Fig 2.15. After ADC conversion, the oversampled signal is passed through a resampler which consists of an interpolator and a decimator. The interpolator is able to generate samples within those actually sampled by ADC. By generating these intermediate samples, the interpolator can adjust the sampling timing as needed. The purpose of decimator is to get different sampling rate of received data. In our receiver design, the decimator downsamples the received signals from twofold-oversampling to original sampling rate.

The information of timing error is computed by sampling clock offset estimator. The sampling clock offset estimator can use a number of different algorithms to generate a detected sampling clock offset. The control signal of resampler is formed by filtering the estimated SCO, i.e., denoted by $\hat{\zeta}$, using an PI loop filter. Timing processor receives the filtered SCO $\hat{\zeta}'$ and computes the corresponding parameters required by resampler, basepoint m_k and fractional timing offset μ_k.

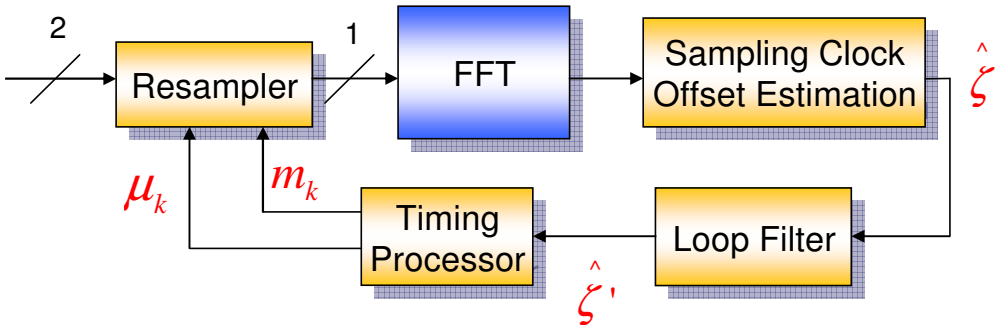


Fig 2.15 Structure of sampling clock synchronization

2.3.1 Sampling Clock Offset Estimation

As previous mentioned, the effect of sampling clock offset in frequency domain is phase rotation which increases every symbol. Referring to Eq(2.12), the difference of phase rotation between two consecutive symbols can be represented as

$$\begin{aligned}\varphi'_l(k) &= \varphi_l(k) - \varphi_{l-1}(k) \\ &\approx 2\pi\Delta f N_s T + \frac{2\pi N_s k\zeta}{N}\end{aligned} \tag{2.25}$$

where Δf is residual carrier frequency offset, ζ is sampling clock offset, N_s is OFDM symbol length equal to $N + N_g$ and k denotes the subcarrier index, $-K/2 \le k \le K/2-1$.

We can assume carrier frequency offset Δf causes mean phase error and sampling clock offset ζ causes linear phase error between two consecutive symbols. If we take two adjacent continual pilots of arbitrary two consecutively received OFDM symbols, the phase rotation is shown in Fig 2.16. The total phase rotation includes the effects of symbol timing offset, CFO and sampling clock offset. In the previous symbol, the magnitude of phase rotation due to symbol timing offset is proportional to subcarrier index. In the current symbol, the effect of CFO and sampling clock offset are accumulated in the phase of previous symbol, where the sampling clock offset induces linear phase and CFO generates mean phase. Thus, we have to estimate the sampling clock offset as well as residual CFO by computing the phase rotation between two consecutive symbols.

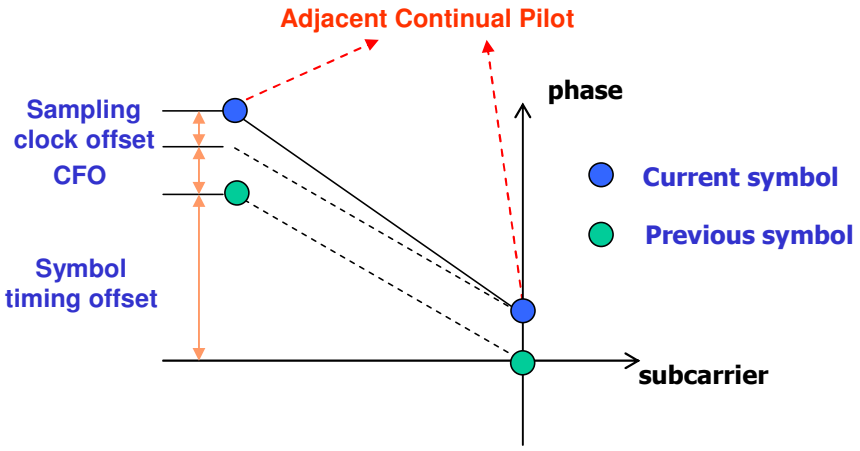


Fig 2.16 Phase rotation between two consecutive symbols

Since the phase rotation is proportional to subcarrier index, the continual pilots which have fixed position of index are exploited to estimate SCO. In general, CFO and SCO are usually estimated jointly because their effects of phase rotation are uncorrelated. To illustrate the subject of phase estimation, an example is depicted in Fig 2.17.

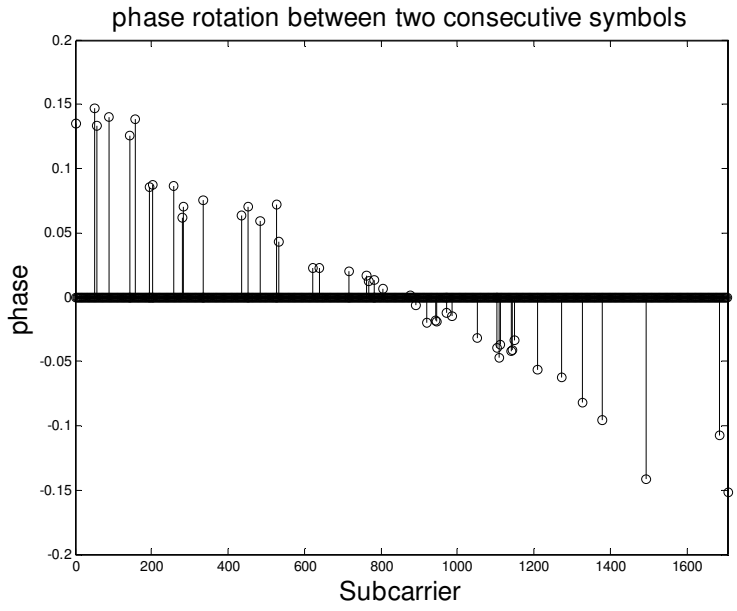


Fig 2.17 Phase error between two consecutive symbols

Fig 2.17 shows the phase error between two consecutive symbols in the 2K mode.

Because of the distortions caused by AWGN, residual ICI and time-variant multipath fading channel, the effect of phase rotation is not as perfect as theoretical value. Therefore, the task of SCO estimation is to find the total phase rotation over the whole OFDM symbol spectrum for averaging the estimation noise. In reference [3], a joint SCO and CFO tracking algorithm is proposed shown as

$$\Delta \hat{f} = \frac{1}{2\pi(1+N_g/N)} \cdot \frac{1}{2} \cdot (\varphi_{2,l} + \varphi_{1,l}) \tag{2.26}$$

$$\hat{\zeta} = \frac{1}{2\pi(1+N_g/N)} \cdot \frac{1}{M/2} \cdot (\varphi_{2,l} - \varphi_{1,l}) \qquad \varphi_{1\|2,l} = \arg\left[\sum_{k \in C_{(1\|2)}} z_{l,k} \cdot z^*_{l-1,k} \right] \tag{2.27}$$

where l denotes symbol length, k denotes subcarrier index and $z_{l,k}$ represents subcarrier after FFT operation. Let C_1 denotes the set of continual pilots which index in the left half $k \in [-(M-1)/2, 0)$ and C_2 the set of continual pilots which index in the right half $k \in (0, (M-1)/2]$ of the OFDM symbol spectrum. Applying correlation of continual pilots in two consecutive symbols and accumulation of the correlation results in two parts lead to the so-called CFD/SFD (carrier frequency detector / sampling frequency detector) algorithm. The summation of $\varphi_{2,l}$ and $\varphi_{1,l}$ can compute mean phase error while the subtraction of $\varphi_{2,l}$ and $\varphi_{1,l}$ produces the linear phase. As a result, SCO and CFO can be estimated jointly by multiplying different constants. However, this algorithm to divide continual pilots into left and right parts and then to accumulate the correlation results is suitable only in the equally distributed pilots. In DVB-T system, the distribution of continual pilots is not exactly equal as shown in Fig 2.17, so that this jointly CFD/SFD algorithm will always generate an unavoidable error in the SCO estimation. In order to improve this problem, a new SCO estimation algorithm is proposed.

As previous mentioned, the purpose of SCO estimation is to compute the phase rotation over the whole OFDM spectrum. Therefore, the subject can be simplified to find the slope of the linear regression line as shown in Fig 2.18 and hence we introduce the linear least square

regression calculator to estimate it. In order to illustrate the linear least square (LLS) method, we assume the candidate regression line is $y = f(x; a_0, a_1) = a_0 + a_1 x$. The process of LLS regression is to minimize the summation of square errors such as

$$E = \sum_{i=1}^{K} [y_i - f(x_i)]^2 = \sum_{i=1}^{K} [y_i - (a_0 + a_1 x_i)]^2 \tag{2.28}$$

Then we have to compute the gradients of $\frac{\partial E}{\partial a_0}$ and $\frac{\partial E}{\partial a_1}$. After letting the gradients be zero, a_0 and a_1 can be computed as optimal value.

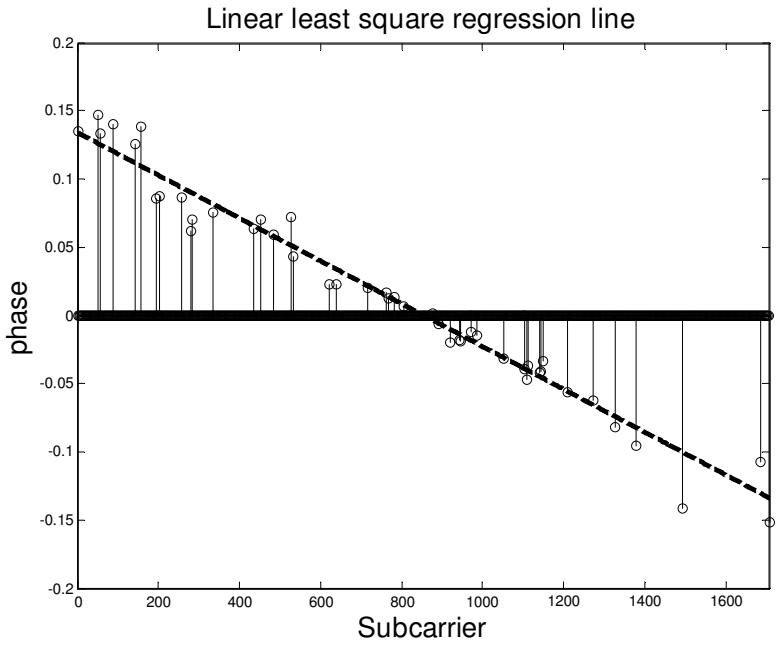


Fig 2.18 Linear least square line

In fact, we can also suppose each x_i passes through this candidate line such as

$$\begin{aligned} a_0 + a_1 x_1 &= y_1 \\ a_0 + a_1 x_2 &= y_2 \\ &\vdots \\ a_0 + a_1 x_K &= y_K \end{aligned}$$

It can be represented as matrix form

$$\begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ & \vdots \\ 1 & x_K \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_K \end{bmatrix}$$
$$A \qquad \theta \qquad y$$

We have to evaluate θ in terms of minimizing square error E

$$E = \left\| y - A\theta \right\|^2 = (y - A\theta)^T (y - A\theta) \tag{2.29}$$

As a result, the theoretical value of θ can be computed by

$$\theta = (A^T A)^{-1} A^T y \tag{2.30}$$

The slope θ_2 is linear phase error and thus can be applied in SCO estimation. On the other hand, the resulting shift θ_1 can be regarded as mean phase error so that the CFO estimation can be also done in linear least square regression method. This joint SCO/CFO estimation method can replace the CFD/SFD algorithm for avoiding the error caused by unequally distributed continual pilots. The LLS algorithm can be sown as

$$\Delta \hat{f} = \frac{1}{2\pi(1+N_g/N)} \cdot \sum_{k=-M/2}^{M/2-1} B_{1,k} \cdot y_{l,k} \tag{2.31}$$

$$\hat{\zeta} = \frac{1}{2\pi(1+N_g/N)} \cdot \sum_{k=-M/2}^{M/2-1} B_{2,k} \cdot y_{l,k} \qquad y_{l,k} = \arg\left[z_{l,k} \cdot z_{l-1,k}^{*} \right]$$

$$B = (A^T A)^{-1} A^T \tag{2.32}$$

$$A = \begin{bmatrix} 1 & k_1 \\ 1 & k_2 \\ & \vdots \\ 1 & k_M \end{bmatrix} \quad | \, k \in CP$$

where $B_{1,k}$ represents the first row of matrix B and $B_{2,k}$ the second row. k denotes the subcarrier index of continual pilot.

In DVB-T system, A represents the distributed indexes of continual pilots, which has been known before receiving signals. Therefore, the complicated matrix operation $(A^T A)^{-1} A^T$ is able to be computed in advance. As a result, the operations of LLS algorithm

are left to several multiplications and accumulations. The complexity is almost the same as CFD/SFD algorithm [3]. The performance improvement will be discussed in Chapter 3.

2.3.2 Resampler

The resampler consists of interpolator and decimator. The task of interpolation is to compute intermediate values between signal samples and decimation executes the downsample process which makes the 2-times oversampling signals received from ADC become signals with original sampling rate. In fact, the decimation process can be accomplished by the interpolation filter itself and hence the separate decimator is not required.

It has been well know that the band-limited input signal $x(t)$ or its samples $\{x(kT_i)\}$ at time $t=kT_i$) could be recovered by using the ideal filter with sinc function which has frequency response

$$h_I(t) = \frac{\sin \pi t / T_s}{\pi t / T_s} \tag{2.33}$$

and impulse response

$$H_I(f) = \begin{cases} T_s, & |f| < \frac{1}{2} T_s \\ 0, & otherwise \end{cases} \tag{2.34}$$

We assume the symbol timing $t = kT + \tau$ as $(m_k + \mu)T_s$, where τ is the timing offset, m_k is an integer and $0<\mu<1$. Subsequently, we can estimate $x(kT+\tau)$ by interpolation:

$$y[(m_k + \mu)T_s] = \sum_{n=-\infty}^{\infty} x[(m_k - n)T_s] h_n(\mu) \qquad (n = ..., -1, 0, 1, ...) \tag{2.35}$$

where $h_n(\mu)$ is the interpolation filter for fractional timing μ

$$\begin{aligned} h_n(\mu) &= h_I(nT_s, \mu T_s) \\ &= \frac{\sin \pi(nT_s + \mu T_s)/T_s}{\pi(nT_s + \mu T_s)/T_s} \qquad (n = ..., -1, 0, 1, ...) \end{aligned} \tag{2.36}$$

Conceptually, the ideal filter can be thought of as an FIR filter with an infinite number of taps.

The coefficient of taps are a function of μ. The structure of the sinc interpolation filter is shown in Fig 2.19.

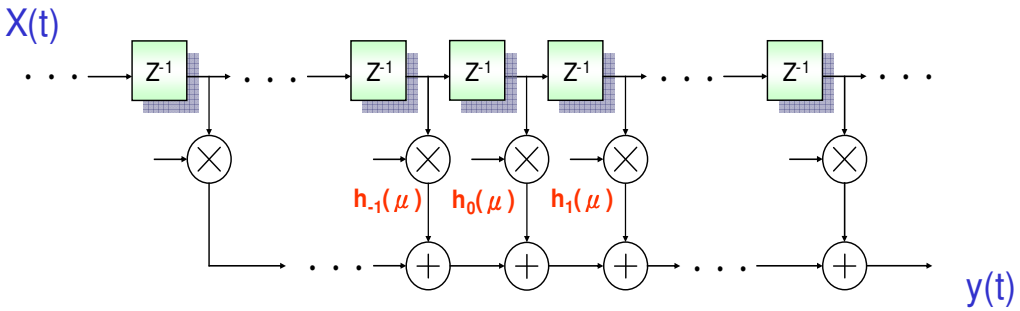
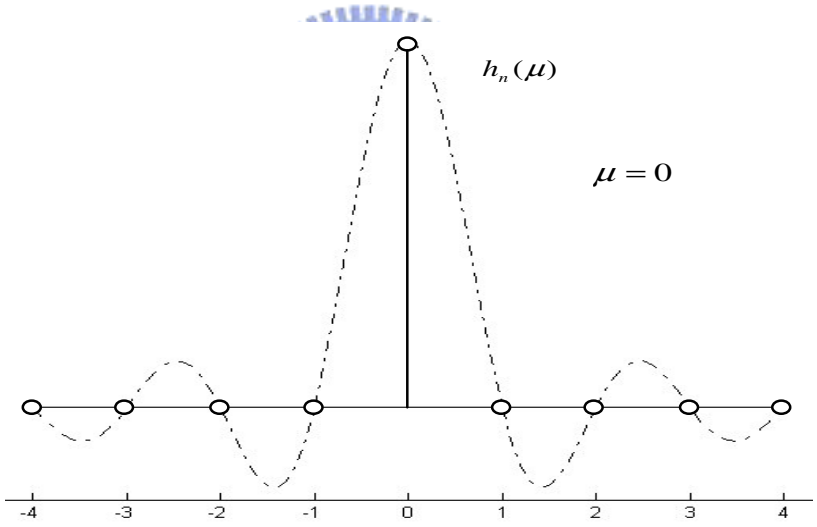
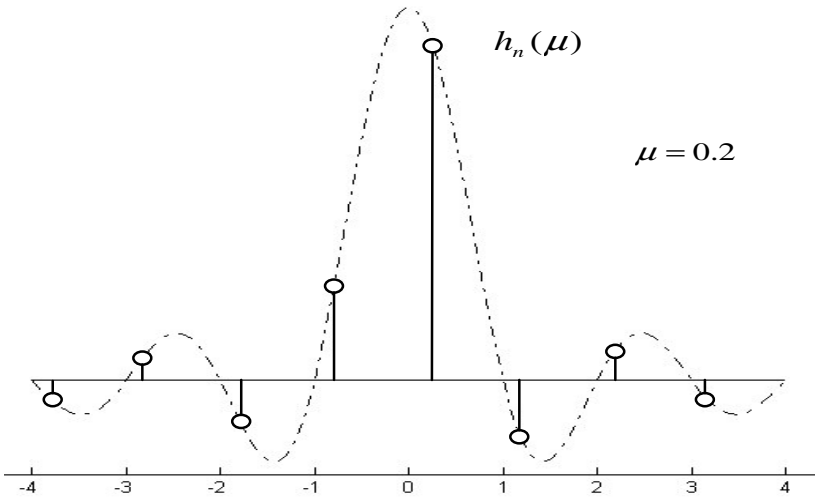


Fig 2.19 Ideal interpolator

The corresponding impulse response of $h_n(\mu)$ is illustrated in Fig Fig 2.20. Fig 2.20 shows the coefficient of each tap in the sinc interpolation filter.



(a) $h_n(\mu)$, $\mu = 0$



(b) $h_n(\mu)$, $\mu = 0.2$

Fig 2.20 Coefficients of $h_n(\mu)$ using sinc function

Theoretically, perfect signal recovery can be achieved by an ideal filter with infinite taps. However, for a practical receiver design the interpolator must be approximated by a finite-order FIR filter

$$y[(m_k + \mu)T_s] = \sum_{n=-I_1}^{I_2} x[(m_k - n)T_s]h_n(\mu) \tag{2.37}$$

The filter performs a linear combination of the (I_1+I_2+1) signal samples $x(nT_s)$ taken around the basepoint m_k. The truncated coefficients of the outside taps raise the side-lobe amplitude of frequency response. It is necessary in practice to window the ideal impulse response instead of removing the tail coefficients simply so as to make it finite.

There is a tradeoff between the main-lobe and side-lobe area when we are seeking the window function. The issue was considered in depth in a series of classic papers. The solution found previously are difficult to compute and therefore unattractive for filter design. However, Kaiser (1966, 1974) found that a near-optimal window could be formed using the zeroth-order modified Bessel function of the first kind, which is much easier to compute. The Kaiser window is defined as

$$w[n]=\begin{cases}\dfrac{I_0[\beta(1-[(n-\alpha)/\alpha]^2)^{1/2}]}{I_0(\beta)}, & 0\le n\le M,\\ 0, & otherwise,\end{cases} \tag{2.38}$$

where $\alpha = M/2$ and $I_0(\cdot)$ represents the zeroth-order modified Bessel function of the first kind which is defined by

$$I_0(z)=\sum_{k=0}^{\infty}\frac{(z^2/4)^k}{k!\Gamma(k+1)} \tag{2.39}$$

where $\Gamma(\cdot)$ is a gamma function.

The Kaiser has two parameters: the length *(M+1)* and a shape parameter β. The window length and shape can be adjusted to trade side-lobe amplitude for main-lobe width. The frequency response of the Kaiser window with *M=20* and different parameter of β is shown in Fig 2.21. If the window is taped more, the side-lobe of spectrum will be smaller but the main-lobe becomes wider, which is depicted in Fig 2.22. The Kaiser window permits the filter designer to predict in advance the values of M and β needed to meet a given frequency-selective filter specification.

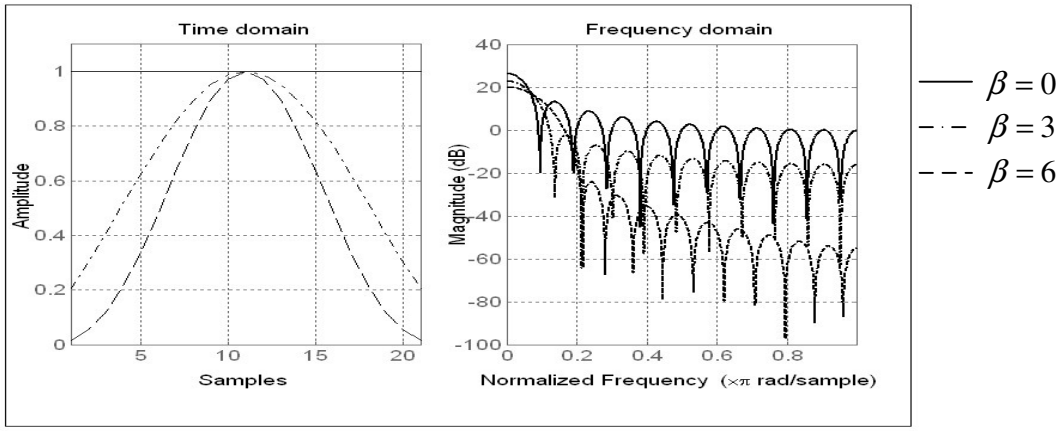


Fig 2.21 Frequency response of Kaiser window with different values of β

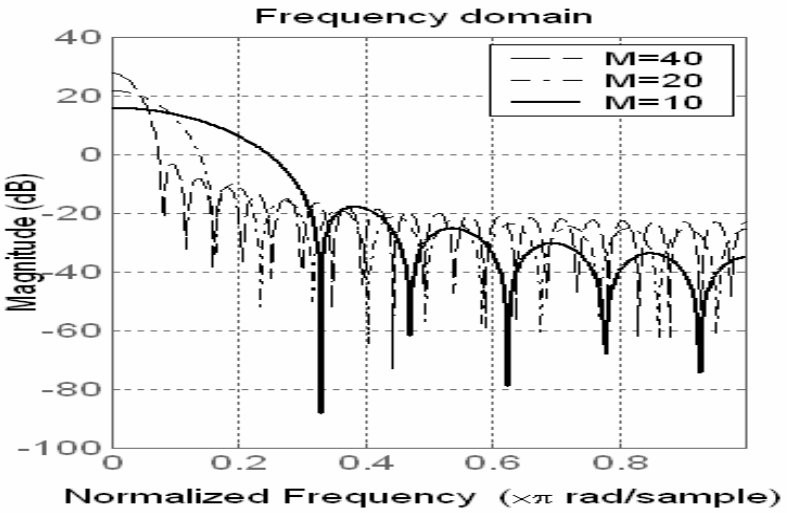
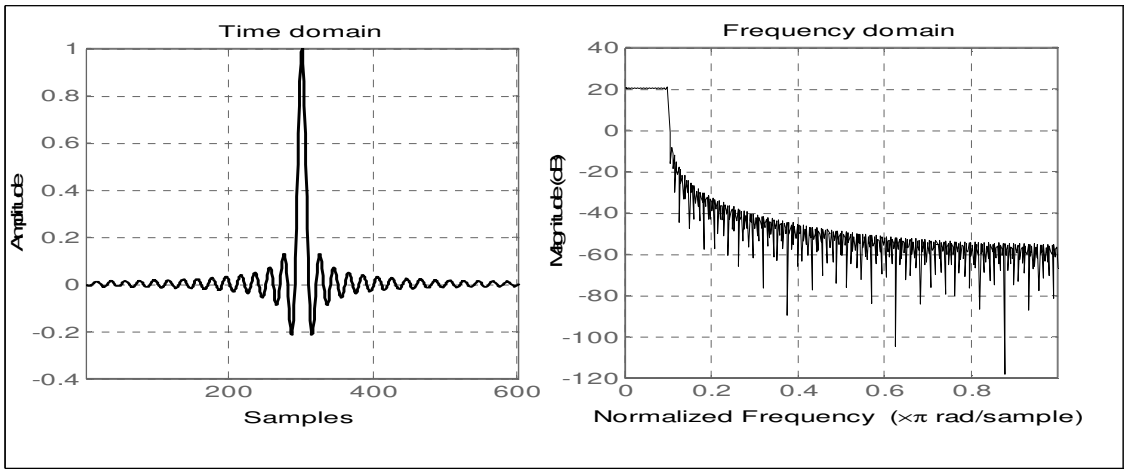


Fig 2.22 Frequency response of Kaiser window with different taps of M

In order to obtain better performance of interpolation, the sinc function truncated by a Kaiser window is used in interpolation filter design. The resulting function $h_n(\mu)$ is defined as

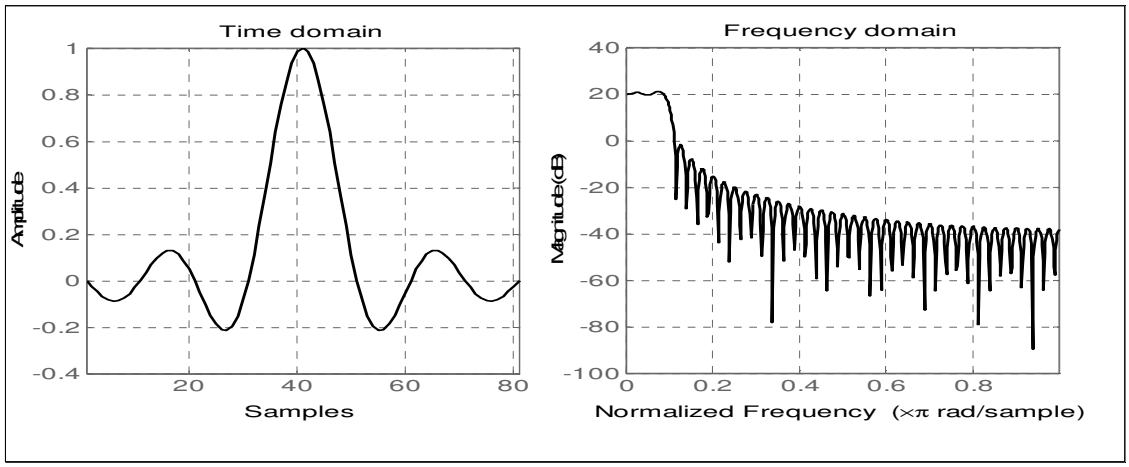
$$h_n(\mu) = \frac{\sin \pi(nT_s + \mu T_s)/T_s}{\pi(nT_s + \mu T_s)/T_s} \cdot \frac{I_0[\beta(1-[n/\alpha]^2)^{1/2}]}{I_0(\beta)} \qquad -\frac{M}{2} \le n \le \frac{M}{2} - 1 \qquad (2.40)$$

where $\alpha = M/2$. The frequency response of 61 taps sinc function which can be regarded as infinite approximately is depicted as Fig 2.23(a). If we truncate sinc function at the third zero crossing to the left and right of the origin, we obtain the frequency response shown in Fig 2.23(b). Note that the relative side-lobe attenuation enhances to 1.3db from 0.2db and the start of stopband raises from -20db to 0db. Fig 2.23(c) shows the frequency response of the sinc function truncated with a Kaiser window. Obviously, its side-lobe becomes slow and relative side-lobe attenuation is reduced so that the performance is improved. The relative simulation performance will be mentioned in Chapter 3.



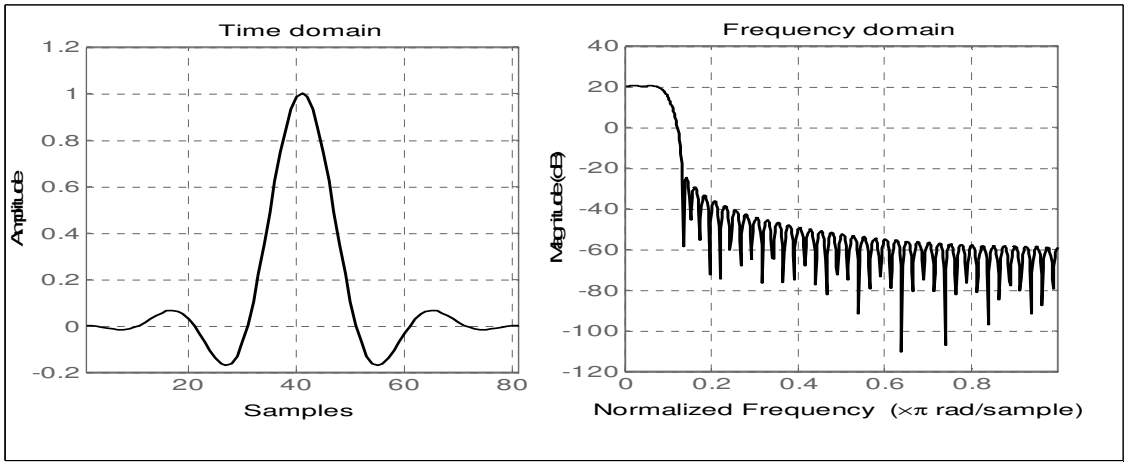
(a) Frequency response of 61-tap sinc function

(Relative side-lobe attenuation: 0.2db, Main-lobe width: 0.19849)



(b) Frequency response of 9-tap sinc function

(Relative side-lobe attenuation: 1.3db, Main-lobe width: 0.18945)



(c) Frequency response of 9-tap sinc function truncated by Kaiser window

(Relative side-lobe attenuation: 0.1db, Main-lobe width: 0.18164)

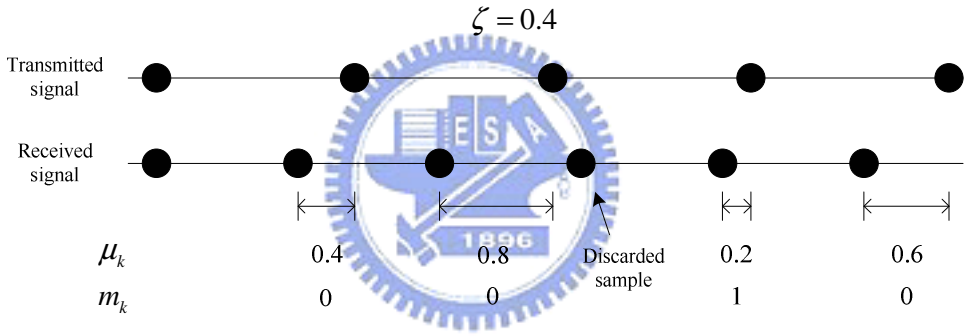
Fig 2.23 Frequency response of 61-tap sinc function, 9-tap sinc function and sinc function truncated by Kaiser window

2.3.3 Timing Processor

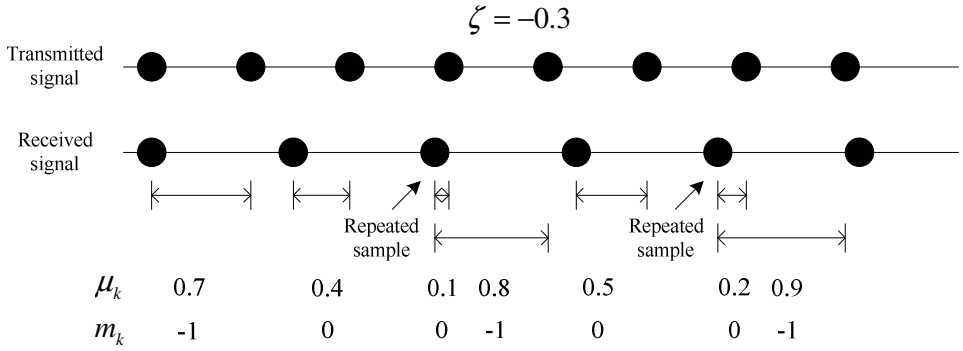
This section focuses on the control of the resampler. In SCO tracking loop, sampling clock offset will be computed every two OFDM symbols. The task of timing processor is to determine the basepoint m_k and the corresponding fractional delay μ_k as the time-variant coefficients based on the output of sampling clock offset estimator. The basic concept of interpolator control has been already described in [12]. In this section, the simpler and clearer structure will be illustrated for hardware consideration.

To recovery the sampling timing, it has to compute the corresponding basepoint m_k and the fractional delay μ_k every sample based on the loop-filtered SCO ζ'. The relation between the received samples, transmitted samples, m_k and μ_k is illustrated in Fig 2.24. Fig 2.24(a) illustrates the condition of SCO = 0.4 while Fig 2.24(b) shows the condition of SCO = -0.3. Note the basepoint m_k not really represents the index of basepoint and it indicates the behavior of the FIR filter instead. In the condition of m_k = 0, FIR filter

operates normally with the following sample as basepoint. The case of m_k = 1 represents that current received sample must be discarded and basepoint is replaced by next sample as shown in Fig 2.24(a). Note that the fractional delay μ_k is defined in the interval of [0, 1). For the hardware consideration, it means that the FIR filter should bypass a sample when the computed m_k = 1. Contrary to the condition of positive SCO as Fig 2.24(a), negative SCO causes that the basepoint must apply the same sample twice, which is denoted by m_k = -1 as shown in Fig 2.24(b). The repeated samples imply that a lot of buffers will be needed. This is not desirable in the practical filter design. Fortunately, this problem can be easily solved by merging decimation process with the interpolator. The subject will be discussed in the later section.



(a) SCO = 0.4



(b) SCO = -0.3

Fig 2.24 Interpolation control

For a given ζ, it is easy to find out the relation between m_k, μ_k and ζ. The sampling timing varies with the SCO ζ. Taking Fig 2.24(a) for example, the first received sample from original synchronized sample has a timing offset of 0.4 comparing with transmitted first sample. In the second sample, the difference of sampling timing between received data and transmitted data becomes 0.8, double of 0.4. Similarly, the third sample should get the offset value of 1.2 which implies the sampling timing offset exceeds one sample. Since we define $0 \le \mu_k < 1$, we should take the fractional part of μ_k, 0.2, and the integer part becomes m_k so as to apply the next sample as basepoint. The recursive estimation is represented by

$$\begin{aligned} m_k &= \lfloor \mu_{k-1} + \zeta' \rfloor \\ \mu_k &= \mu_{k-1} + \zeta' - m_k \end{aligned} \tag{2.41}$$

where ζ is replaced by ζ' since the timing processor actually receives the loop-filtered SCO ζ' rather than SCO ζ.

In order to illustrate the decimation process merging within interpolator, we take an example as Fig 2.25. There are two exaggerated cases of different ζ' listed in Fig 2.25. Left case describes the behavior of timing processor in the condition of positive ζ'=0.6 and initial timing offset μ_0=0. Right case also shows the computation of corresponding m_k and μ_k in the condition of negative ζ'=-0.2 and μ_0=0.3. The calculations of m_k and μ_k are according to Eq(2.41).

	$\zeta' = 0.6, \mu_0 = 0$		$\zeta' = -0.2, \mu_0 = 0.3$	
k	m_k	μ_k	m_k	μ_k
1	0	0.6	0	0.1
2	1	0.2	-1	0.9
3	0	0.8	0	0.7
4	1	0.4	0	0.5
5	1	0.0	0	0.3
6	0	0.6	0	0.1
7	1	0.2	-1	0.9
8	0	0.8	0	0.7

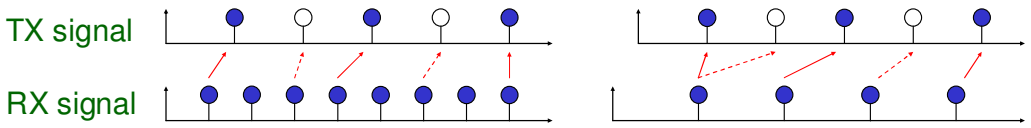


Fig 2.25 Parameter computation in timing processor

In the decimation process, the interpolated samples just only have to output one sample for every two samples, which is shown as each m_k bounding by the rectangle. For the case of positive ζ' in the original interpolator, the current sample has to be discarded and then uses next sample as basepoint if the computed m_k is 1. After applying the concept of decimation, the operations have something changed. We introduce the definitions of first zero and second zero. For example, $m_k|_{k=1}$ in the left case is a first zero. First zero can be followed by another zero which is denoted as second zero. When the computed m_k is an odd-zero, the decimation controller turns on and hence an output sample is generated. The decimation controller will turn off when m_k is an even-zero. If m_k is generated as 1, it implies sampling offset exceeds one sample. There are two different conditions of m_k=1. The one is m_k=1 follows an even-zero and another is that follows an odd-zero. In the first condition, m_k=1 should be assumed as an even-zero and hence two samples have to be discarded. In the second condition, m_k=1 should be regarded as an odd-zero so that the decimation controller has to turn on after bypassing one sample.

The condition of negative ζ' causes the repeated basepoints, which is undesirable in practical filter design. In the original interpolator, the computations of m_k and μ_k are listed as right case of Fig 2.25. Similar to the case of positive ζ', the definitions of odd-zero and

even-zero is the same. In additional, it is necessary to consider whether the next m_k is -1 when the current computed m_k is zero. In fact, there are many combinations of m_k=0, 1, -1. In order not to miss any case, the full cases of $\{m_{k-1}, m_k, m_{k+1}\}$ are listed in Table 2.1. Note that the case of {0_odd, 0_even, -1_odd} and {0_odd, -1_even, 0_odd} represent the repeated basepoint is needed in the original interpolator design. However, by merging decimation process we can avoid this problem and thus each sample generates an output sample at most. The operation of two above-mentioned cases is to compute m_{k+2}, μ_{k+2}. Then shift right the former $\{m_{k-1}, m_k, m_{k+1}\}$ to $\{m_k, m_{k+1}, m_{k+2}\}$, and subsequently apply $\{m_{k+1}, \mu_{k+1}\}$ as new $\{m_k, \mu_k\}$ so as to interpolate resulting sample.

{ m_{k-1}, m_k, m_{k+1} }	**Operation of Decimation Controller**
{ 0_odd, 0_even, 0_odd }	Bypass current sample
{ 0_even, 0_odd, 0_even }	Turn on current sample
{ 0_odd, 0_even, 1_odd }	Bypass current sample
{ 0_even, 1_odd, 0_even }	Bypass current sample and then turn on in the next sample
{ 1_odd , 0_even, 0_odd }	Bypass current sample
{ 0_even, 0_odd, 1_even }	Turn on current sample
{ 0_odd, 1_even, 0_odd }	Bypass two consecutive samples
{ 1_even, 0_odd, 0_even }	Turn on current sample
{ 0_odd, 0_even, -1_odd }	Compute m_{k+2}, u_{k+2} and shift right to { 0_even, -1_odd, 0_even }
{ 0_even, -1_odd, 0_even }	Turn on current sample
{ -1_odd , 0_even, 0_odd }	Bypass current sample
{ 0_even, 0_odd, -1_even }	Turn on current sample

{ 0_odd, -1_even, 0_odd }	Compute m_{k+2}, u_{k+2} and shift right to { -1_even, 0_odd, 0_even }
{ -1_even, 0_odd, 0_even }	Turn on current sample

Table 2.1 Decimation controller

To simplify Table 2.1, we can remove the consideration of m_{k-1} and reduce the cases as shown in Table 2.2, where X means the condition of don't care. Thus the timing processor merging decimation process within interpolator can be easily implemented in hardware. The 8-tap resampler as well as timing processor is depicted in Fig 2.26.

{ m_k, m_{k+1} }	**Operation of Decimation Controller**
{ 0_even, 0_odd }	Bypass current sample
{ 0_odd, X_even }	Turn on current sample
{ 0_even, 1_odd }	Bypass current sample
{ 1_odd, 0_even }	Bypass current sample and then turn on the next sample
{ 1_even, 0_odd }	Bypass two consecutive samples
{ 0_even, -1_odd }	Compute m_{k+2}, u_{k+2} and shift right to { -1_odd, 0_even }
{ -1_odd, 0_even }	Turn on current sample
{ -1_even, 0_odd }	Compute m_{k+2}, u_{k+2} and shift right to { 0_odd, 0_even }

Table 2.2 Simplified decimation controller

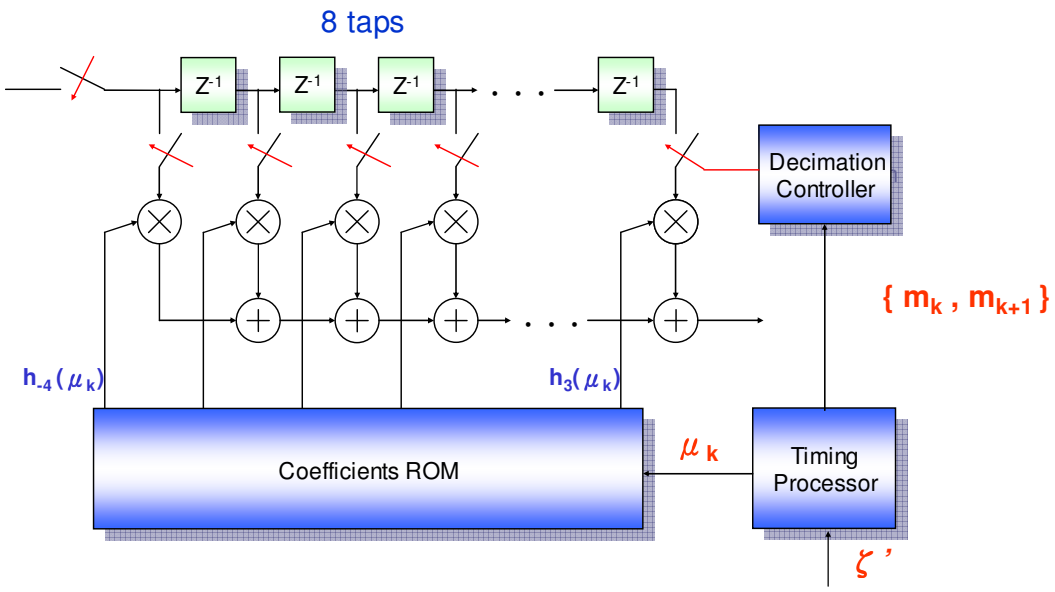


Fig 2.26 Structure of 8-tap resampler

2.3.4 Sampling Clock Tracking Loop

In the loop filter design, PI loop filter is commonly used in PLL design and timing recovery circuit [15]. In our receiver design, PI loop filter is chosen to track the estimated SCO error. The PI loop filter consists of two paths. The proportional path multiplies the estimated SCO by a proportional gain K_p. From control theory, it is known that a proportional path can be used to track out a phase error. However, it cannot track out a frequency error. For a sampling clock tracking loop to track out a sampling frequency error, a loop filter containing an integral path is needed. The integral path multiplies the estimated SCO by an integral gain K_I and then integrates the scaled error using an adder and a delay block. An IIR loop filter as shown in Fig 2.27 can track out both a sampling phase and a sampling frequency error.

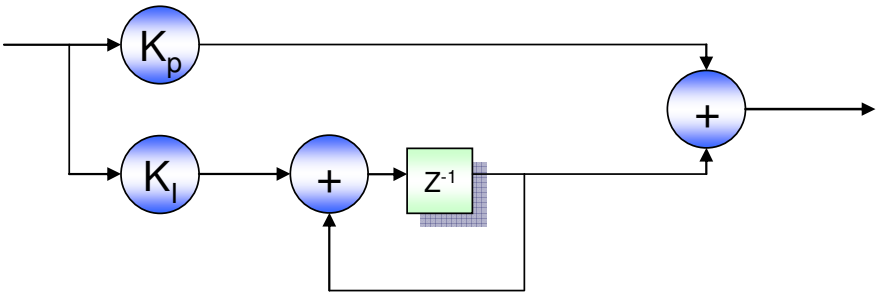


Fig 2.27 PI loop filter

The transfer function of PI loop filter can be represented as

$$H(z) = K_p + K_I \frac{Z^{-1}}{1 - Z^{-1}} \tag{2.42}$$

For a small loop delay and $K_I - K_P << K_P << 1$, the steady-state tracking error standard deviation is given by

$$\sigma(e') = (\sqrt{K_P / 2} \cdot \sigma(e)) \tag{2.43}$$

where e is the estimation error of SCO estimator and e' is the steady-state tracking error. The closed-loop tracking time constant is approximately given by

$$T_{loop} \approx 1 / K_p \tag{2.44}$$

So that there is a tradeoff between steady-state tracking error and tracking convergence speed. For hardware implementation, it is possible to change loop parameters during online reception. In the start of tracking period, a large value K_p is used for fast convergence speed and then switches to a small value in order to reduce the tracking error variation.

2.4 *Timing Synchronization Scheme*

This section illustrates the overall timing synchronization scheme, which contains synchronization control, flow chart, switch between acquisition mode and tracking mode. As mentioned before, timing synchronization consists of symbol synchronization and sampling clock synchronization. In acquisition mode, Mode/GI detection is first applied to decide

transition mode and guard interval length, which takes 2~3 OFDM symbols for evaluating the period of flat peak area. Subsequently, coarse symbol boundary lying in ISI-free region is detected by coarse symbol synchronization and thus downstream blocks can operate correctly. Coarse symbol synchronization has to take 1 symbol for correlation peak detection. In addition to coarse symbol synchronization, coarse fractional carrier frequency acquisition must be done prior to FFT operation. Therefore, the following step should be pre-FFT AFC in time-domain, which takes 1 OFDM symbol. After FFT, integer part of CFO is acquired by post-FFT AFC so that the subcarrier index is adjusted, which requires one OFDM symbol. Then the scattered pilot mode detection must precede the equalizer and the operations in tracking mode for being aware of scattered pilot positions. As above described, acquisition mode consists of Mode/GI detection, coarse symbol synchronization, pre-FFT AFC in time domain and post-FFT AFC, scattered pilot mode detection in frequency domain. The total OFDM symbols required by acquisition mode are 6~7 symbols. After the information of coarse symbol boundary, CFO and scattered pilot mode are realized so that acquisition mode turns off and then tracking mode starts.

In tracking mode, the residual CFO, sampling clock offset and fine symbol timing are evaluated all the time. The CFO and SCO tracking are processed by their own PI (proportional-integral) tracking loop. The joint CFO and SCO estimator computes the resulting ζ' and Δf in two consecutive symbols so that the tracking loop operates per two symbols. If two-stage CFO/SCO tracking loop is applied, the loop filter must retain the previous tracking value when loop parameter K_p changes to a smaller value. In order to keep the convergence of tracking achieve within a TPS frame which includes 68 OFDM symbols, the objective of tracking convergence speed is to obtain the best SNR before the next whole TPS frame as shown in Fig 2.28. Since the worse case of usable symbols in arbitrary receiving is 68 symbols, it is better to achieve convergence in 68 symbols.

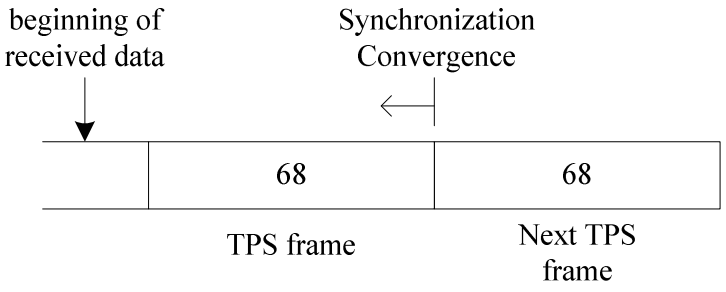


Fig 2.28 Convergence speed of synchronization considering TPS frame

As a result, the two-stage SCO/CFO tracking should change the loop filter parameter prior to 68 symbols. In our receiver design, we use a large K_p first and then change a smaller K_p in the 50-th symbol. The relative simulation will be discussed in Chapter 3.

Another topic of synchronization scheme is fine symbol synchronization control. Since the fine symbol synchronization changes the FFT window for avoiding timing drift, the effect of FFT window adjustment must be paid attention by other tracking blocks and channel estimation unit. The SCO/CFO tracking loop has to halt if FFT window changes. And the 2D interpolation channel estimation exploits memory to buffer three OFDM symbols so that the resulting phase shifts must be compensated in channel estimation unit. The flow chart of overall synchronization scheme is depicted as Fig 2.29.

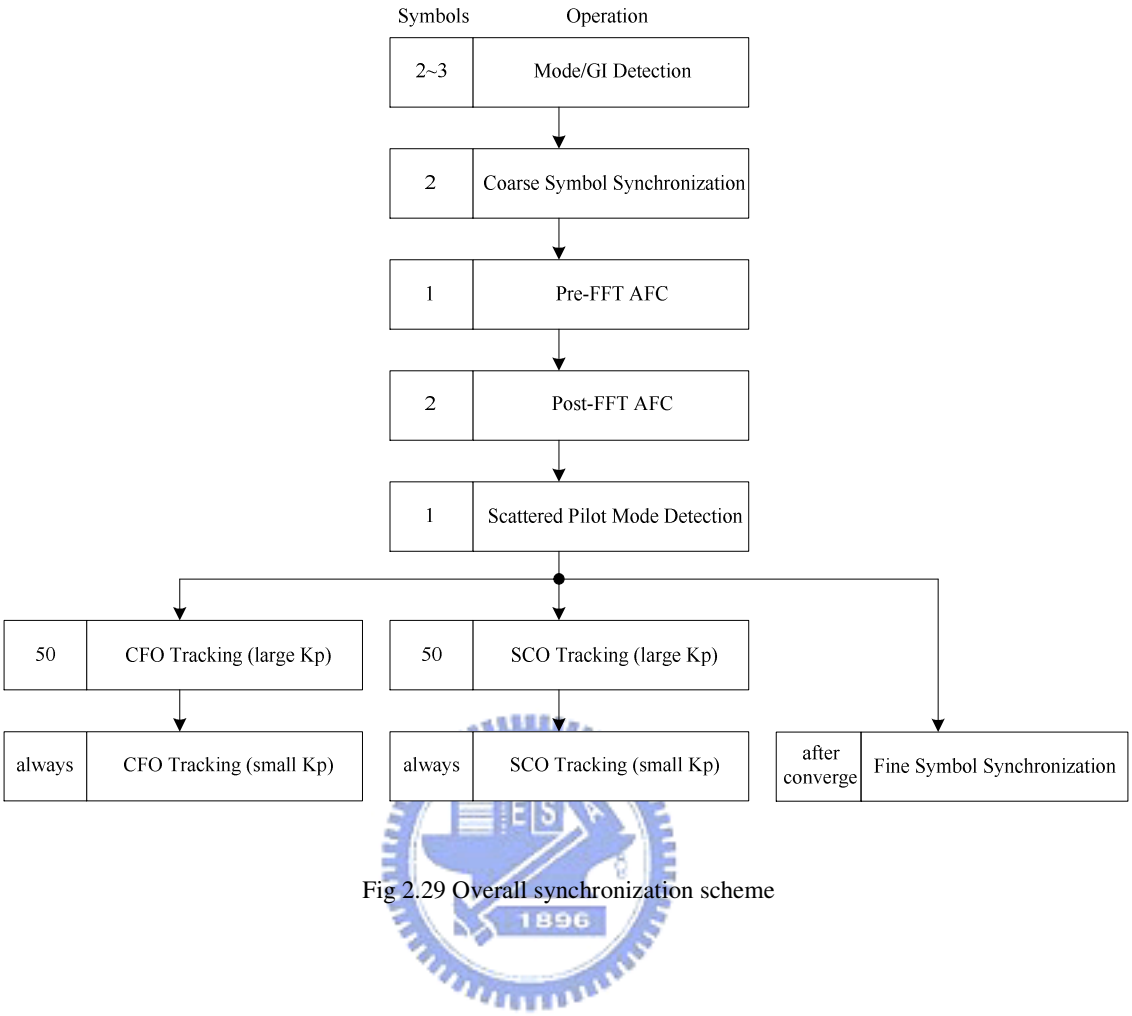


Fig 2.29 Overall synchronization scheme

Chapter 3
Simulation and Performance

3.1 Simulation Platform

In order to verify the performance of proposed algorithm, a complete DVB-T baseband simulation platform is developed in Matlab. The block diagram of DVB-T simulation platform is depicted in Fig 3.1.

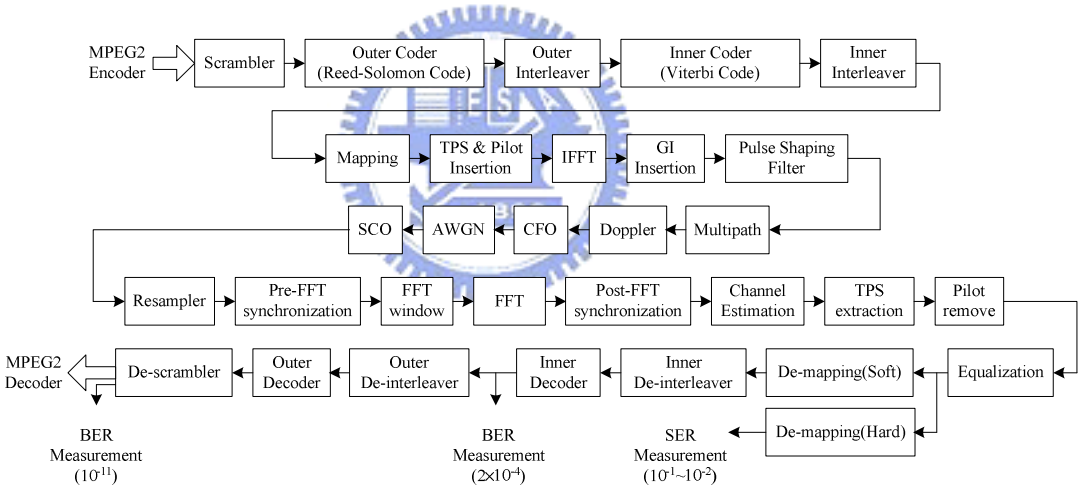


Fig 3.1 Block diagram of simulation platform

The platform consists of transmitter, channel and receiver. A typical transmitter receiving video data from MPEG2 encoder is fully established. Besides FEC blocks, constellation mapping, pilot insertion, IFFT modulation and GI insertion are built in order. The 2K and 8K with all other transmission modes are able to being selected as simulation parameters. In order to simulate discrete signal as far as continuously, upsampling and pulse shaping filter are adopted prior to entering channel. The upsampling rate is flexible and depends on the required

simulation accuracy. The roll-off factor of pulse-shaping filter is not defined in ETSI DVB-T standard so that a normal value of α=0.15 is used.

In the channel model, various channel distortions are introduced for simulating real mobile wireless environment, which contains multipath fading, Doppler frequency spread, AWGN, CFO and SCO. In fact, there are some other distortions such as co-channel interference, adjacent-channel interference and common phase error due to defective front-end receiving. However, these distortions are relatively small compared to effective time-variant channel response caused by Doppler spread, CFO and SCO, so that we can neglect those channel effects.

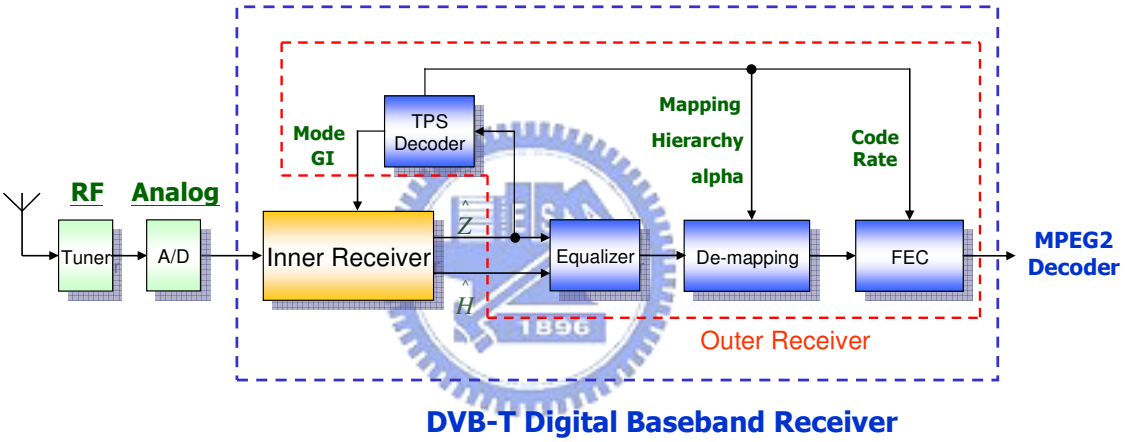


Fig 3.2 Overview of receiver design

In the receiver design, we focus on the baseband demodulation part between ADC and MPEG-2 decoder. The receiver can be divided into two portions, inner and outer receiver as depicted in Fig 3.2. Inner receiver copes with pre-FFT synchronization, FFT, post-FFT frequency synchronization, channel estimation and pilot removal. Then TPS check, de-mapping, inner de-interleaver, viterbi decoder, outer de-interleaver, RS decoder and de-scrambler are done in outer receiver. The transmission parameters computed by TPS decoder such as constellation mapping and code rate of viterbi send to downstream blocks in outer receiver. Afterwards, the bitwise output of FEC blocks enters to source decoding block, MPEG-2 decoder. Note the TPS check should operate all the time to prevent transmission

interruption. If TPS check error occurs, the inner receiver ought to reset and hence all blocks in acquisition mode restart. As for BER measurement, the quasi error-free condition is defined in ETSI standard [1] which means less than one uncorrected error event per hour, corresponding 10^{-11} after Reed-Solomon decoder and $2x10^{-4}$ after Viterbi decoder. Therefore, the BER should be measured both in the outputs of Viterbi and RS. In particular, the SER (symbol error rate) is usually applied as another performance measurement in several papers. As a result, we should exploit hard-decision demapping to measure SER in addition.

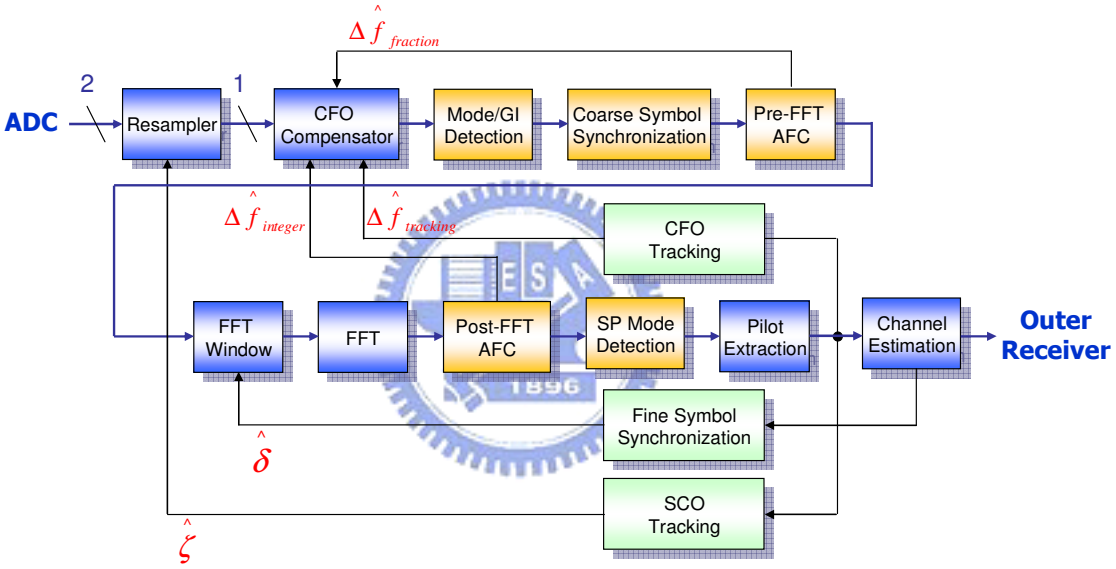


Fig 3.3 Structure of inner receiver

Fig 3.3 shows the detail structure of inner receiver. As mentioned in Chapter 2, synchronization task consists of symbol synchronization, frequency synchronization and sampling clock synchronization. Acquisition blocks operate in the initial synchronization period and turn off in tracking mode, and tracking blocks act all the while.

Our frequency synchronization design consults reference [3]. Like coarse symbol synchronization, pre-FFT frequency acquisition is based on guard interval correlation. Disregarding ISI and sampling timing error, the tail received sample and its cyclic prefix show the same property except for a phase rotation between guard and tail segments being

proportional to the fractional carrier frequency offset. Guard interval correlation samples thus become

$$x_n = r_n \cdot r_{n-N}^{*} \propto e^{j2\pi\Delta f} + noise \tag{3.1}$$

Given the coarse estimated symbol window $\hat{n}$, the ML frequency estimate [16] becomes

$$\Delta \hat{f} = \frac{1}{2\pi} \cdot \arg\left[\sum_{i=\hat{n}-N_g-1+x}^{\hat{n}} r(i) \cdot r^{*}(i-N) \right] \tag{3.2}$$

where x denote the forsaken samples distorted by ISI in multipath channel. Since the perfect coarse symbol window is impossible, we have to consider the ISI samples. In severe multipath fading channel as Rayleigh channel in DVB-T standard, long time delay profile raises the ISI effect which is illustrated in Fig 3.4. As a result, we must give up several beginning samples to reduce the ISI distortion. However, discarding too many samples will also degrade the averaging performance. The optimal value of x can be decided by simulation result.

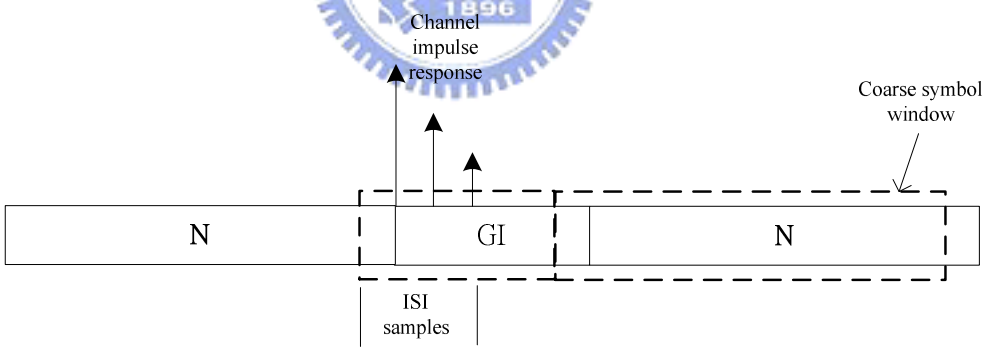


Fig 3.4 ISI effect on CFO acquisition

Post-FFT integer carrier frequency acquisition refers to [3]. Because of pre-FFT acquisition, the residual fractional offset Δf is small so that the ICI noise in this stage is also small. We assume the integer carrier frequency offset n_I (subcarrier spacings), which causes spectrum shift in frequency domain. The integer CFO must now be detected using continual pilots which are all boosted in power. Correlating FFT output samples of two consecutive

OFDM symbols *l-1*, *l* and a particular set $k \in C + m$ are accumulated. The maximum absolute value of accumulation result then yields the estimated integer carrier frequency offset

$$\hat{n}_I = \arg \max_{m \in I} \left| \sum_{k \in C+m} z_{l,k} \cdot z^*_{l-1,k} \right| \tag{3.3}$$

where *C* denotes the positions of continual pilots and *I* represents the search range which is typical given by $[-n_{I,\max}, n_{I,\max}]$. Considering small offset Δf and ζ, the probability of false detection ($\hat{n}_I \neq n_I$) is very small.

The channel estimation unit must estimate both the channel response and any residual phase errors caused by imperfect synchronization. In the channel estimation design of DVB-T system, it is common to use two-dimensional interpolation method such as [3] in order to estimate the mobile time-variant channel. The channel response is generated by interpolation in time and frequency dimension respectively. In time direction, channel gain estimates at scattered pilot are first interpolated so that channel gain estimates are available at every third subcarrier in every OFDM symbol as depicted in Fig 3.5. Subsequently, channel response estimates at all other subcarriers are obtained by interpolating the resulting time-interpolated channel gain in frequency direction. In time-dimension interpolation, four complete OFDM symbols have to be stored for each noncausal tap. Considering the memory requirement, interpolation in time dimension exploits linear interpolation so that only the storage of three additional OFDM symbols is needed. As for frequency-dimension interpolation, it is general to adopt Wiener filter approach. In general, the high-complexity frequency direction interpolation deserves since the system performance is usually dominated in channel estimation unit.

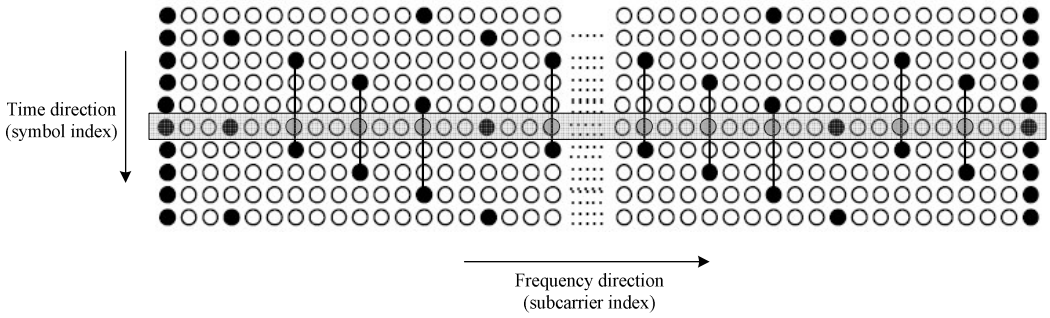


Fig 3.5 2D interpolation in channel estimation unit design

3.2 Channel Model

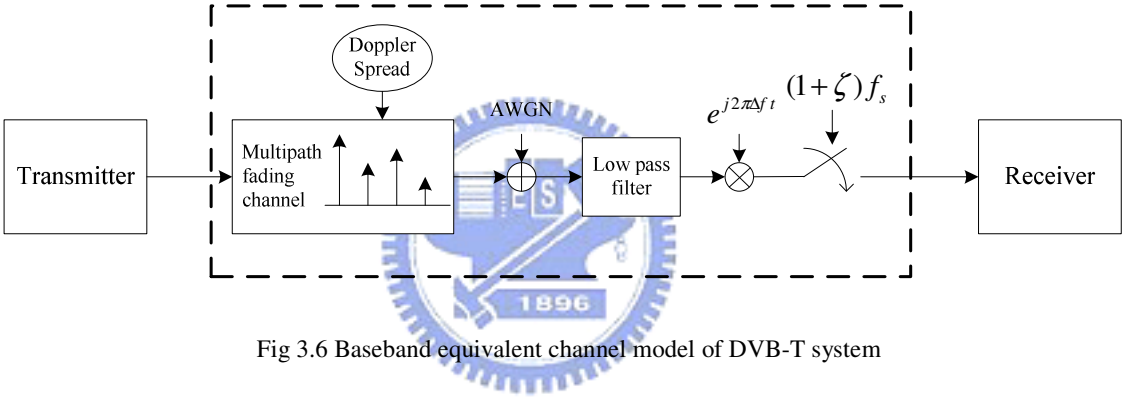


Fig 3.6 Baseband equivalent channel model of DVB-T system

Fig 3.6 shows the typical baseband equivalent channel model of DVB-T system. The transmitted data passes through multipath fading, Doppler spread, AWGN, RF lowpass filter, carrier frequency offset and sampling clock offset. The effects of inter-channel interference (co- and adjacent-channel interference) and common phase error are neglected in our simulation. In fact, the overall system performance represented as “BER versus SNR” shows nearly no difference. The detail illustration of each channel distortions will be shown in the following sections.

3.2.1 Multipath Fading Channel Model

In the wireless transmission, transmitted data is received through several paths with

different time delay and power decay. This is so-called multipath fading. Two types of channel are specified by ETSI DVB-T standard. Fixed reception condition is modeled by Ricean channel (Ricean factor = 10db) while portable reception is modeled by Rayleigh channel. The full 20-tap Ricean and Rayleigh channel was used with floating point tap magnitude and phase values and with tap delay accuracies rounded to within 1/2 of duration (7/64 μ s/2) for practical discrete simulation.

The major difference between Rayleigh and Ricean channel is the main path (The line of sight ray). In a Rayleigh fading channel, the received signals consist of several reflected signals with similar powers because there is no main path in Rayleigh channel. The rms delay of Rayleigh channel is about 12 sample time. This characteristic will cause serious synchronization error. A time delay and subcarrier distortion of frequency domain both occur in a Rayleigh fading channel. The frequency response and impulse response for each subcarrier are shown in Fig 3.7. The frequency response is not flat over the entire frequency region and some parts are severely distorted. The Ricean channel model defines Ricean factor K (the ration of the power of the direct path to the reflected paths) is given as

$$K = \frac{\rho_0^2}{\sum_1^N \rho_i^2} \tag{3.4}$$

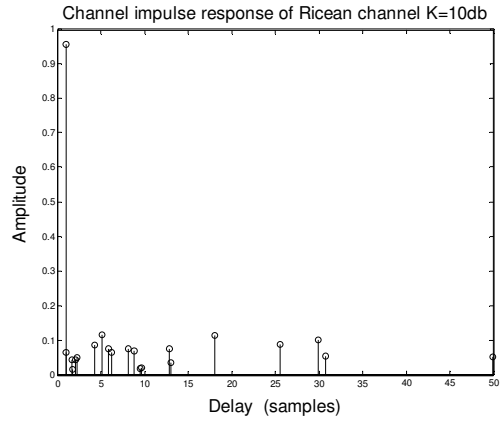
where ρ_i is the attenuation of the i'th path. The channel models can be generated from the following equation where x(t) and y(t) are input and output signals respectively

$$\text{Rayleigh:} \quad y(t) = k\sum_{i=1}^{N} \rho_i e^{-j\theta_i} x(t-\tau_i) \qquad k = \frac{1}{\sqrt{\sum_{i=1}^{N} \rho_i^2}} \tag{3.5}$$

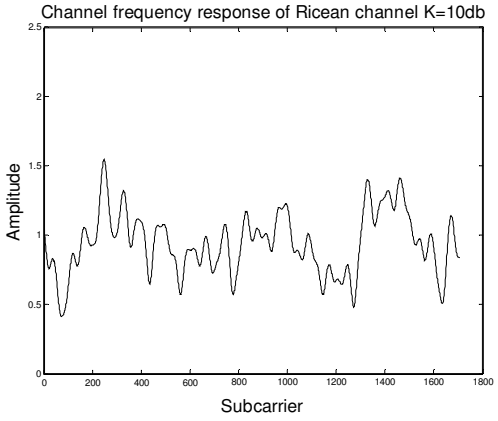
$$\text{Ricean:} \quad y(t) = \frac{\rho_0 x(t) + \sum_{i=1}^{N} \rho_i e^{-j\theta_i x(t-\tau_i)}}{\sqrt{\sum_{i=0}^{N} \rho_i^2}} \tag{3.6}$$

where N is the number of echoes equals to 20, θ_i is the phase shift from scattering of the i'th path and τ_i is the relative delay of the i'th path. The detail value of above parameters is

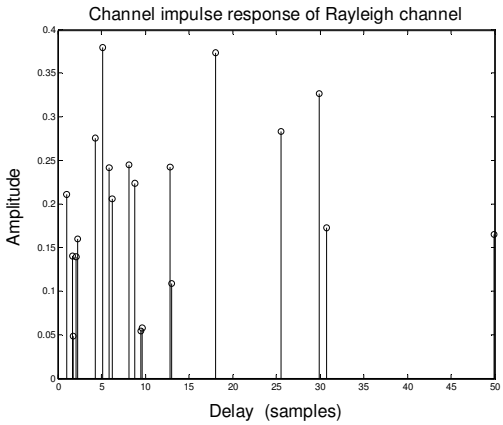
listed in table B.1 of [1]. The rms delay of Ricean channel (k=10db) and Rayleigh channel is respectively 0.4491 μs (about 4 samples) and 1.4426 μs (about 13 samples). The channel impulse response and channel frequency response of Ricean channel (K=10db) and Rayleigh channel are shown in Fig 3.7 respectively.



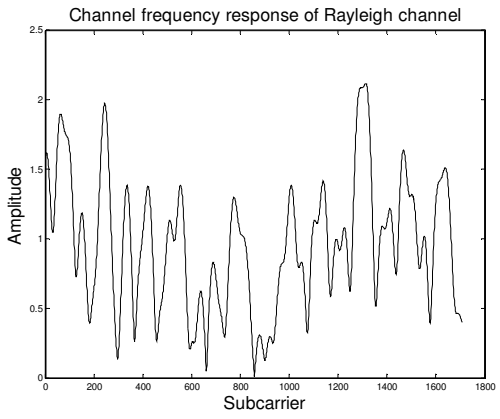
(a) Channel impulse response of Ricean channel



(b) Channel frequency response of Ricean channel



(c) Channel impulse response of Rayleigh channel



(d) Channel frequency response of Rayleigh channel

Fig 3.7 Channel response of Ricean channel (K=10db) and Rayleigh channel

In addition to Rayleigh and Ricean channel, a statistical channel model WSSUS (Wide Sense Stationary Uncorrelated Scattering) [17] is adopted in our simulation. The power delay profile is measured in two different areas in Germany (Berlin and Darmstadt) with a system bandwidth of 8 MHz and at carrier frequencies of 714 and 920 MHz. We can regard this channel model as a real case of transmission environment. WSSUS channel model provides

several type of channel model which contains *Non Line Of Sight (NLOS)* –models and *Type K (TypK)*-models. NLOS models have a very small Rice factor ($K \le -20db$) and TypK models have a Rice factor around the 50% of its model category. The detail descriptions of each channel model have been discussed in [17].

3.2.2 Doppler Frequency Spread Model

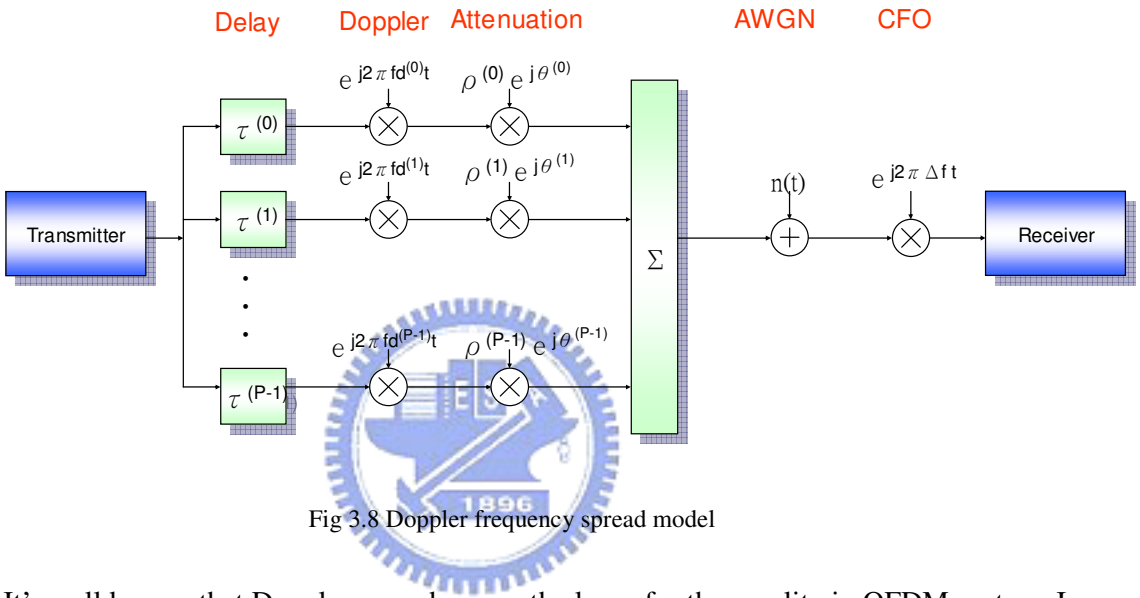


Fig 3.8 Doppler frequency spread model

It's well known that Doppler spread causes the loss of orthogonality in OFDM system. In DVB-T system, a mobile radio channel including Doppler spread must be considered. A simplified Doppler frequency spread model [11] is depicted in Fig 3.8. First, we initially assume a channel with a known and fixed number of paths P such as Ricean, Rayleigh or WSSUS with a Doppler frequency $f_d^{(k)}$, attenuation $\rho^{(k)} e^{j\theta^{(k)}}$, and time delay $\tau^{(k)}$. Every path has different amplitude, Doppler frequency, and time delay. Since each path has its own Doppler frequency, how to decide the statistical distribution for f_d is important. There are two commonly used Doppler frequency PDFs, uniform and classical. Obviously uniform case uses uniform distribution to model Doppler spread, and classical case uses Jakes' Doppler spectrum.

In some papers, a worst case of two-side Doppler spectrum is exploited, which shifts in

frequency each even-indexed channel tap by + f_d, and each odd-indexed channel tap by - f_d. When we compare the performance with other papers, we should pay attention to the definition of Doppler Spectrum.

The PDF of Jakes' Doppler spectrum [18] is derived as below.

$$p(f_d) = \frac{1}{\pi \cdot f_{d\max} \cdot \sqrt{1-\left(\frac{f_d}{f_{d\max}}\right)^2}} \qquad |f_d| < f_{d\max} \tag{3.7}$$

After transformation of random variable, we can obtain each f_d by the following equation.

$$f_d = \cos(2\pi \cdot rand(1)) \cdot f_{d\max} \tag{3.8}$$

The resulting spectrum is shown in Fig 3.9. The type of Doppler spread (uniform or Jakes') affects the performance very much. Because each path gets different f_d in each simulation case, the amount of the lost orthogonality will be not the same. Therefore, we should fix each f_d in each simulation and comparison.

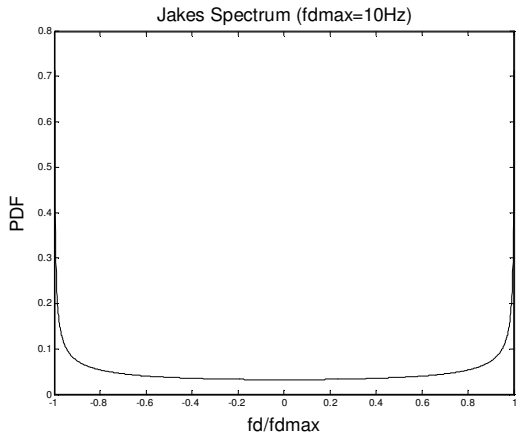


Fig 3.9 Jakes' spectrum with $f_{d\max} = 10\ Hz$

3.2.3 Carrier Frequency Offset Model

In the RF front-end, carrier frequency offset always exists between transmitter and receiver due to the mismatch between transmitter and receiver oscillators or channel Doppler frequency shift. The signal distortion with carrier frequency offset ε can be derived as

$$y(t) = x(t) \cdot e^{j2\pi\varepsilon t} \tag{3.9}$$

where x(t) is transmitted signal and r(t) is received signal. In terms of discrete time signal, we rewrite Eq(3.9) as Eq(3.10) [19]

$$y(n) = x(n) \cdot e^{j2\pi\Delta f \frac{n}{N}} \tag{3.10}$$

where N denotes IFFT points and Δf is the relative frequency offset of the channel (the ratio of the actual frequency offset to intercarrier spacing). In frequency domain, ICI will occur due to linear phase error in time domain.

3.2.4 Sampling Clock Offset Model

The model of sampling clock offset is built based on the concept of sinc interpolation. The input digital signals can interpolate the intermediate value between two consecutive samples using the shifted value of sinc function. We assume sampling period is T_s and sampling clock offset is ζ. Then the sampling phase can be represented as $nT_s + n\zeta$. The resulting signal after analog-to-digital converter (ADC) can be derived as

$$\begin{aligned} r_{ADC}(nT_s) &= r(nT_s) * \sin c(\frac{nT_s + n\zeta}{T_s}) \\ &= \sum_{k=-N}^{N} r(nT_s - kT_s) \cdot \sin c(k + \frac{n\zeta}{T_s}) \end{aligned} \tag{3.11}$$

where 2N+1 represents taps of FIR interpolator, k is the sampling point index, $r(\cdot)$ denotes received signal with perfect sampling and $r_{ADC}(\cdot)$ is received signal with sampling clock offset ζ. Note T_s should be twofold-oversampling period.

3.3 Performance

		Required C/N for BER = 2×10^{-4} after Viterbi QEF after Reed-Solomon			Bitrate (Mbit/s)			
Modu-lation	Code rate	Gaussian channel	Ricean channel (F_1)	Rayleigh channel (P_1)	Δ/T_U = 1/4	Δ/T_U = 1/8	Δ/T_U = 1/16	Δ/T_U = 1/32
QPSK	1/2	*3,1*	*3,6*	*5,4*	*4,98*	*5,53*	*5,85*	*6,03*
QPSK	2/3	*4,9*	*5,7*	*8,4*	*6,64*	*7,37*	*7,81*	*8,04*
QPSK	3/4	*5,9*	*6,8*	*10,7*	*7,46*	*8,29*	*8,78*	*9,05*
QPSK	5/6	*6,9*	*8,0*	*13,1*	*8,29*	*9,22*	*9,76*	*10,05*
QPSK	7/8	*7,7*	*8,7*	*16,3*	*8,71*	*9,68*	*10,25*	*10,56*
16-QAM	1/2	*8,8*	*9,6*	*11,2*	*9,95*	*11,06*	*11,71*	*12,06*
16-QAM	2/3	*11,1*	*11,6*	*14,2*	*13,27*	*14,75*	*15,61*	*16,09*
16-QAM	3/4	*12,5*	*13,0*	*16,7*	*14,93*	*16,59*	*17,56*	*18,10*
16-QAM	5/6	*13,5*	*14,4*	*19,3*	*16,59*	*18,43*	*19,52*	*20,11*
16-QAM	7/8	*13,9*	*15,0*	*22,8*	*17,42*	*19,35*	*20,49*	*21,11*
64-QAM	1/2	*14,4*	*14,7*	*16,0*	*14,93*	*16,59*	*17,56*	*18,10*
64-QAM	2/3	*16,5*	*17,1*	*19,3*	*19,91*	*22,12*	*23,42*	*24,13*
64-QAM	3/4	*18,0*	*18,6*	*21,7*	*22,39*	*24,88*	*26,35*	*27,14*
64-QAM	5/6	*19,3*	*20,0*	*25,3*	*24,88*	*27,65*	*29,27*	*30,16*
64-QAM	7/8	*20,1*	*21,0*	*27,9*	*26,13*	*29,03*	*30,74*	*31,67*
NOTE: Figures in italics are approximate values. Quasi Error Free (QEF) means less than one uncorrected error event per hour, corresponding to BER = 10^{-11} at the input of the MPEG-2 demultiplexer.								

Table 3.1 Required C/N for non-hierarchical transmission

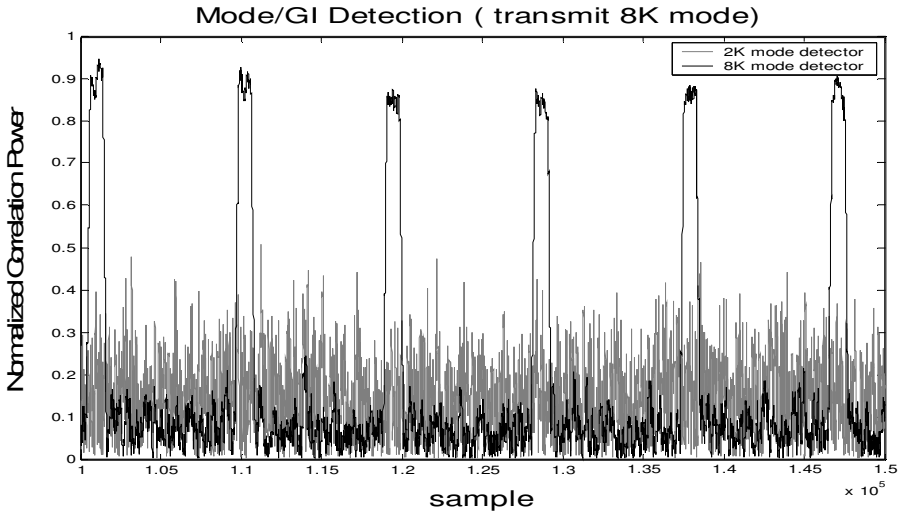
Unlike many other modern communication systems, including GSM based systems, 802.11 based systems and the Bluetooth system, DVB-T does not have a conformance level as such, i.e. a performance that must be met under specific channel conditions and/or input power levels. Instead, presented in the standard is the performance of the system under various conditions where perfect knowledge of channel information is known at the receiver and no phase noise is presented in the receiver. It is generally accepted that an implementation margin of 1.5 to 3db is tolerated, but this is not defined in the standard.

Table 3.1 gives simulated performance anticipating “perfect channel estimation and without phase noise” of channel coding and modulation combinations, and are subject to confirmation by testing. These results are given for Gaussian, Ricean and Rayleigh channels. In the Gaussian channel interferences are caused by noise, while in the Ricean and Rayleigh channels interferences are dominated by echoes. Fixed reception (directional antenna) is modeled by the Ricean channel while portable reception (omni-directional antenna) is modeled by the Rayleigh channel. Associated useful bit rates available are also indicated as a function of guard interval to active symbol duration for the three difference values of guard

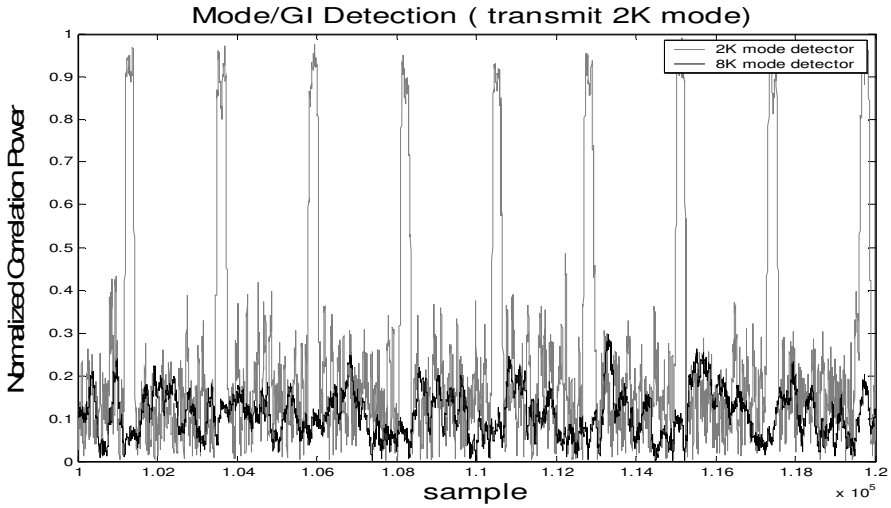
interval.

3.3.1 Mode/GI Detection

The simulated flat peak area in Mode/GI detection block is depicted in Fig 3.10. Fig 3.10(a) shows the 2K mode detector and 8K mode detector respectively in the 8K mode transmission. As we can see, the flat peak area only occurs in 8K mode transmission after normalized maximum correlation. Similarly, 2K mode transmission is illustrated in Fig 3.10(b). After deciding proper threshold, the period of flat peak area can be estimated so that we can obtain the correct transmission mode and guard interval length.



(a) 8K mode transmission



(b) 2K mode transmission

Fig 3.10 Flat peak area in Mode/GI detection

The performance of Mode/GI detection is depicted in Fig 3.11 in terms of false detection rate versus SNR. The definition of SNR is E_s / N_0, where E_s is the energy per channel symbol and $N_0 / 2$ is the two-sided power spectral density of the AWGN. Both performances in 2K and 8K mode are shown in Fig 3.11. The transmission mode is 64-QAM and GI=1/8. The simulation environment is Rayleigh fading channel, Doppler spread = 70 Hz, CFO=20ppm and SCO=20ppm. Since the Mode/GI detection is the first stage of acquisition mode, it has to bear all channel distortions without any compensation. As we can see, 8K mode has much lower false detection rate than 2K mode because of longer symbol duration. Longer symbol duration makes the difference between possible values of period lager so that the possibility of false detection becomes small apparently. In general, error-free condition in 64-QAM can be achieved when SNR > 8db in 2K mode and >3db in 8K mode. As for other constellation mapping like 16-QAM or QPSK, required SNR of error-free condition is much smaller.

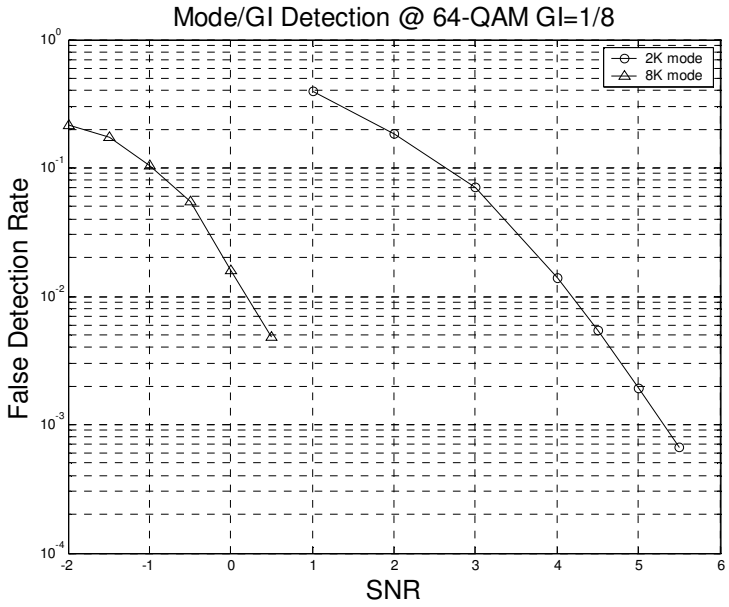
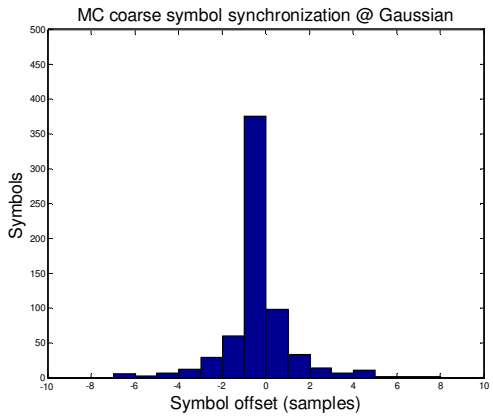


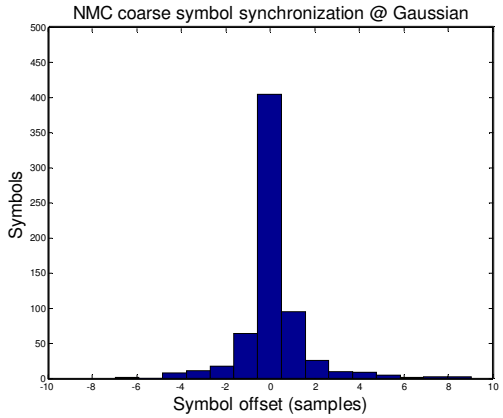
Fig 3.11 False Mode/GI detection rate versus SNR

3.3.2 Coarse Symbol Synchronization

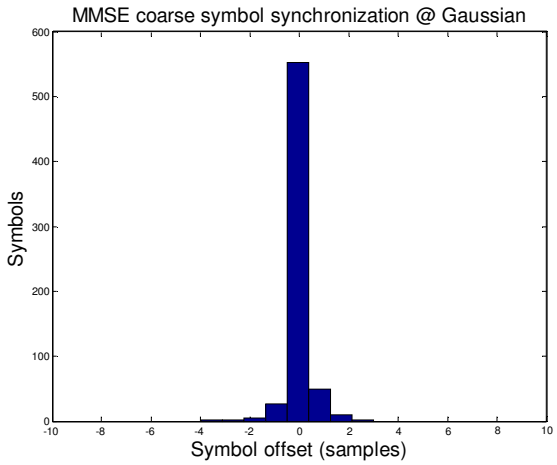
As mentioned in Chapter 2, there are numerous algorithms of coarse symbol synchronization. The design target of coarse symbol synchronization is to find symbol boundary and ensure it lying ISI-free region. In order to analyze the performance of each algorithm, we take 650 OFDM symbols to estimate the resulting symbol offset and depict a histogram to show the statistic distribution. As a first acquisition stage of synchronization, coarse symbol synchronization has to tolerate severe channel distortions such as large CFO and large SCO before tracking operations. Therefore, we assume the simulated channel condition as Doppler spread 70 Hz, CFO=20ppm, SCO=20ppm and AWGN=10db. The three cases of Gaussian channel, Ricean channel (K=10db) and Rayleigh channel are shown below.



(a) MC algorithm in Gaussian channel

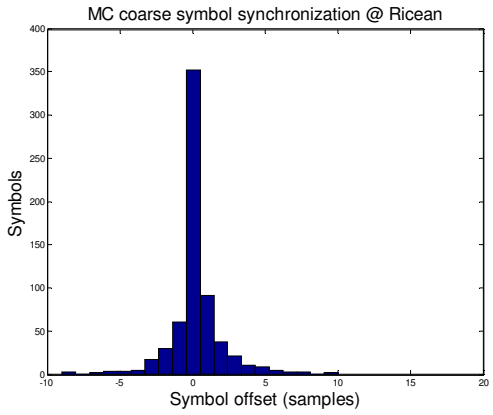


(b) NMC algorithm in Gaussian channel

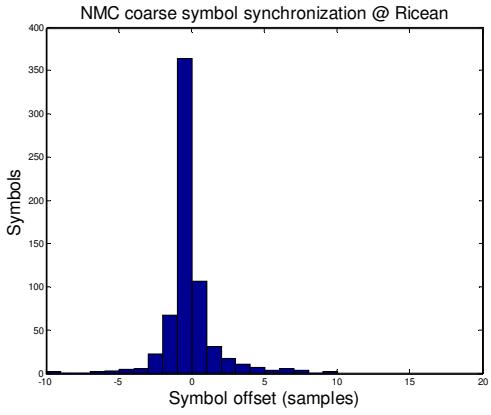


(c) MMSE algorithm in Gaussian channel

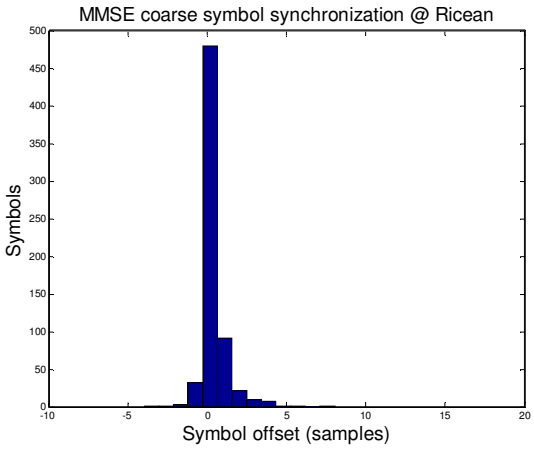
Fig 3.12 Histogram of estimated symbol offset in Guassian channel



(a) MC algorithm in Ricean channel

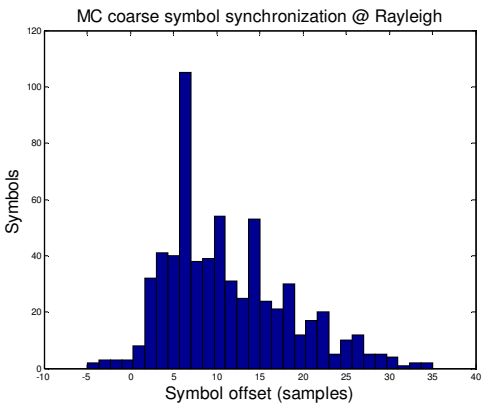


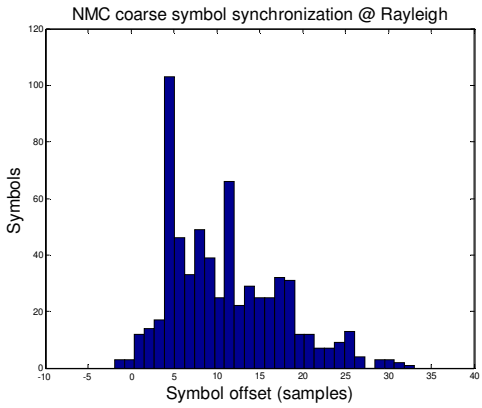
(b) NMC algorithm in Ricean channel



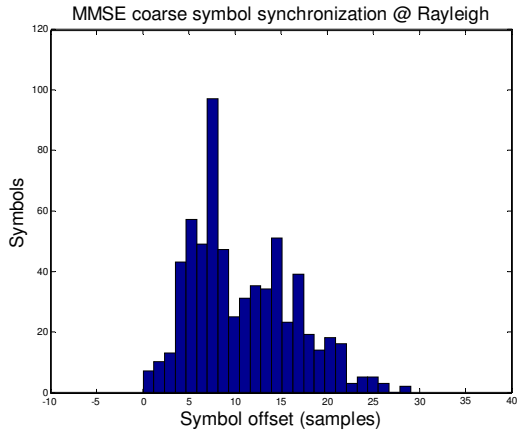
(c) MMSE algorithm in Ricean channel

Fig 3.13 Histogram of estimated symbol offset in Ricean channel





(a) MC algorithm in Rayleigh channel (b) NMC algorithm in Rayleigh channel



(c) MMSE algorithm in Rayleigh channel

Fig 3.14 Histogram of estimated symbol offset in Rayleigh channel

Fig 3.12 shows the estimated symbol offset in the Gaussian channel. Since the Gaussian channel does not have the property of multipath fading, the mean excess delay is regarded as zero. As a result, the estimated symbol offset centralizes in the interval of [-4, 4]. Comparing the performances of MC, NMC and MMSE algorithm, we can find the difference between MC and NMC is very small. And MMSE algorithm has a slightly better performance than other two algorithms. The simulation result in the Ricean channel (K=10db) is depicted in Fig 3.13. Unlike Gaussian channel, Ricean channel has rms delay spread of 0.4491 μs (about 4 samples) so that the ISI effect is much heavier than Gaussian channel. The resulting estimated symbol offset is distributed in the larger interval of [-8, 8]. As shown in Fig 3.13 clearly, MMSE has a more centralized distribution and just in the interval of [-5, 5]. As for Rayleigh channel, the strong multipath fading degrades the performance of coarse symbol synchronization much due to severe ISI effect. Since Rayleigh channel has rms delay spread of 1.4426 μs (about 13 samples), the estimated symbol offset is distributed in the interval of [-5, 35] as shown in Fig 3.14. A large delayed peak causes large symbol timing offset. Apparently, MMSE algorithm has better performance again. In summary, MMSE algorithm has an advantage of performance among these three algorithms. Concerning the hardware

implementation, MMSE has the highest hardware cost so that there exists a tradeoff between performance and hardware complexity.

In order to provide the reliable symbol boundary, the estimated delayed peak position must shift several samples in advance and set this shifted position as symbol boundary. As for how many shifted samples are enough, we have to consider the worst case of channel and ensure the symbol boundary being safe in this channel. In DVB-T system, ETSI standard [1] defines a Rayleigh channel for portable reception so that we can assume it as the worst channel. To check Fig 3.14(c) again, MMSE algorithm has a largest 28-sample-delayed peak. As a result, the symbol boundary must shift left 30 samples at least.

In order to detect the periodic peaks, an appropriate threshold must be decided. Since the power level of the correlation result varies depend on the channel characteristics. The severer channel yields the lower correlation power. Therefore, an adaptive threshold decision strategy has to be developed. We propose a simple peak detection algorithm. An adaptive threshold can be computed by

$$\Gamma = m + K \cdot \sigma \tag{3.12}$$

where m denotes the mean of resulting correlation power and σ represents standard deviation. The definition of K depends on the amount of noise and the variation of signals. For very noisy signals, it is best to lower it. In DVB-T system, K=1 is a suitable parameter value for peak detection algorithm. As a result, the coarse symbol synchronization has to spend some OFDM symbols to evaluate the mean and standard deviation of correlation power and therefore the peak can be correctly detected.

3.3.3 Scattered Pilot Mode Detection

As the last stage of acquisition mode, scattered pilot mode detection estimates the positions of scattered pilots prior to other operations in tracking mode like CFO/SCO tracking and fine symbol synchronization. The scattered pilot mode detection must have low

false-detection rate since the inaccurate scattered pilot position yields wrong operations of the downstream blocks in tracking mode. The scattered pilot mode detection must tolerate residual fractional ICI, large symbol offset and potentially large SCO. Therefore, we assume the simulation environment as 2K mode, GI=1/8, 64-QAM, Rayleigh channel, Doppler spread=70Hz, residual fractional CFO=0.02 (the ratio of carrier frequency offset to subcarrier spacing), and SCO=20ppm. The false detection rate versus SNR is depicted as Fig 3.15

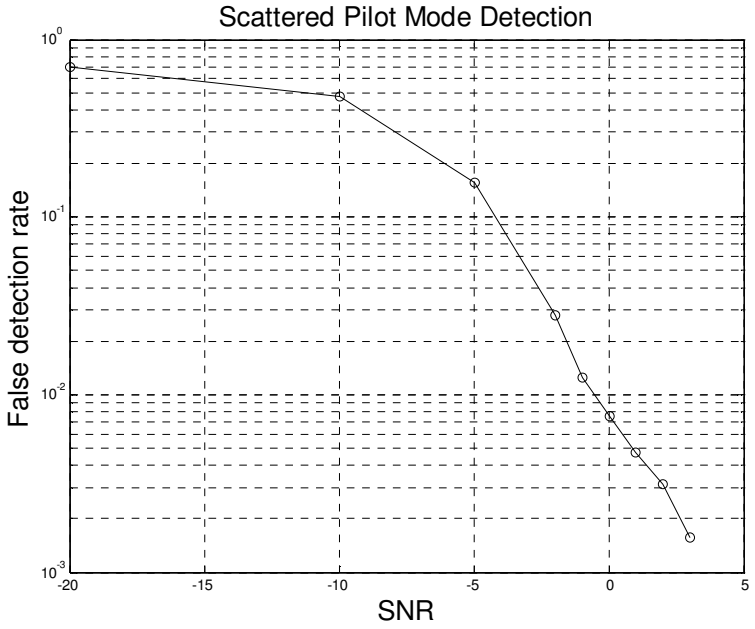


Fig 3.15 False detection rate of scattered piloted mode detection

The error-free condition can be achieved when SNR>5db and 8K mode will get better performance because of more subcarriers it has. In summary, the scattered pilot mode detection algorithm is strongly reliable in normal transmission.

In fact, this operation can probably be neglected if we disable tracking operations until TPS decoding is completed. Since the TPS bits contain frame information, the position of scattered pilots is able to be obtained as long as the TPS bits are decoded. However, the convergence of synchronization will delay a TPS frame, i.e. 64 OFDM symbols, at least. Therefore, there exists a tradeoff between hardware complexity and convergence speed.

3.3.4 Fine Symbol Synchronization

As illustrated in Chapter 2, fine symbol synchronization exploits IFFT to estimate the effective channel impulse response. Time-dimension interpolated scattered pilots are used to assume as sub-sampled channel frequency response. After zero padding, IFFT generates the effective channel impulse response with resolution of 3/2. To estimate the path delay can yields the symbol timing offset caused by inaccurate coarse symbol synchronization or sampling clock offset. The performance of path delay estimation depends on the channel type such as Gaussian, Ricean and Rayleigh. Furthermore, Doppler spread also dominates the accuracy of symbol timing offset estimation. We discuss these simulation performances in different channel condition respectively.

(1) Static Gaussian channel

In the static Gaussian channel, the first path is also the strongest path. Therefore, path delay can be easily estimated by detect the maximum $\hat{S}_k$ as shown in Fig 3.16

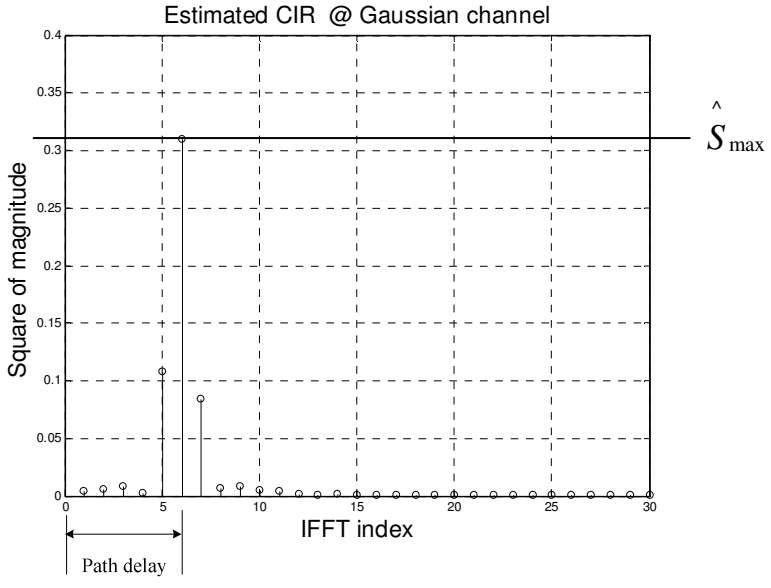


Fig 3.16 Estimated channel impulse response in Gaussian channel

The simulation environment is 2K mode, 64-QAM, GI=1/8, residual CFO=0.02 (ratio of actual CFO to subcarrier spacing), SCO=20ppm and AWGN=10db. The detected path delay of $\hat{S}_{\max}$ can reliably represent the symbol timing offset by restoring the resolution of estimated CIR. After multiplying by 2/3, the actual path delay caused by symbol timing offset can be obtained. Fig 3.17 shows the path delay estimation in the case of SCO=20ppm without timing recovery. X-axis denotes the timing drift caused by sampling clock offset. We calculate the actual symbol timing offset as depicted in dotted line. And the estimated symbol timing drift by IFFT path delay estimation is shown in solid curve. The solid curve appears in a shape of stair since the resolution of estimated CIR is not infinite. For increasing the resolution, we can pad more zeros to the channel gain and use more points of IFFT. As Fig 3.17 is depicted, the estimated symbol timing offset is very close to actual symbol timing offset so that the fine symbol synchronization can be achieved in Gaussian channel.

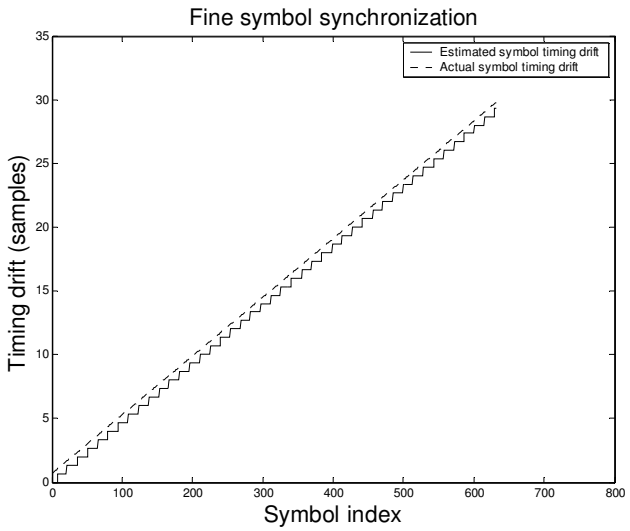


Fig 3.17 Path delay estimation in Gaussian channel

(2) Ricean channel with Doppler spread 70 Hz

Similar to Gaussian channel, Ricean fading channel (K=10db) has a main path as shown

in Fig 3.7(a). After applying Doppler spread, the channel has time-variant channel impulse response. The time-variant CIR can probably cause the error of path delay estimation. Fig 3.18 shows the estimated CIR and then we can detect $\hat{S}_{\max}$ which yields path delay.

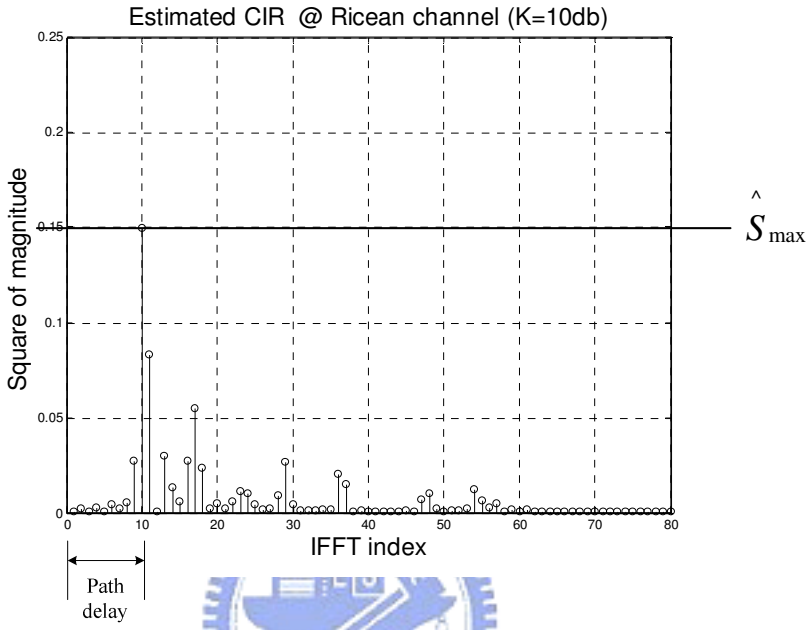


Fig 3.18 Estimated channel impulse response in Ricean channel

The estimation result is depicted in Fig 3.19. Comparing to Fig 3.17, the accuracy of estimation is slightly worse in some symbol index. The inaccurate path delay estimation is caused by time-variant CIR due to Doppler spread. However, the estimation error is tolerable if we take some samples in safe region that means we can define several sample offset like 5 samples as the target of symbol synchronization instead of complete synchronization.

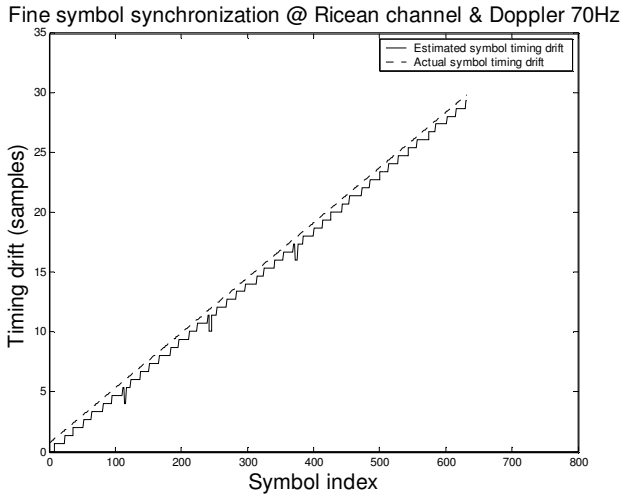


Fig 3.19 Path delay estimation in Ricean channel

(3) Rayleigh channel with Doppler spread 70 Hz

Unlike Gaussian or Ricean channel with a direct path, Rayleigh fading channel has no main path as shown in Fig 3.7. Therefore, the previous criterion of maximum detection is not suitable for Rayleigh channel since the peak probably occurs during a large interval. Considering the Doppler spread, time-variant CIR causes the false estimation of path delay. As a result, another criterion has to be used in Rayleigh channel. Instead of finding the maximum $\hat{S}_k$, we first set a threshold and select the first strong path whose energy must exceed the threshold value Γ. Hence, the estimated path delay is

$$\hat{\delta} = \frac{2}{3}\min\left\{k \mid \hat{S}_k > \Gamma, \quad k = 0, \cdots, M-1\right\} \tag{3.13}$$

where the M-point IFFT is used and δ is the estimated symbol timing offset.

How to choose Γ is an important issue. In order to make Γ less sensitive to noise, an adaptive threshold is needed. We define the threshold as

$$\Gamma = \hat{S}_{\max}/16 \tag{3.14}$$

The parameter of 16 is determined by simulation and is also feasible for hardware

implementation. The resulting path delay estimation is depicted in Fig 3.20. In fact, this criterion can be applied in either Gaussian channel or Ricean channel and a slightly smaller path delay will be detected. The smaller path delay will not affect the fine symbol synchronization much.

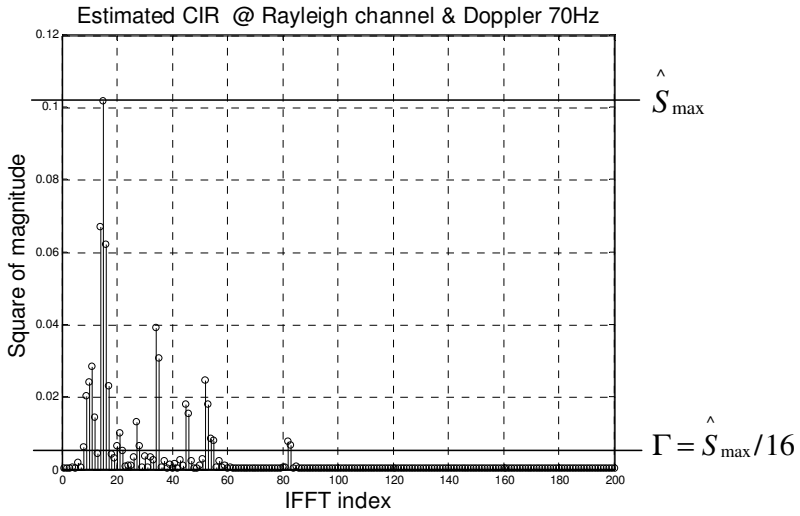


Fig 3.20 Estimated channel impulse response in Rayleigh channel

The simulation performance is shown in Fig 3.21. As we can see, the estimated symbol timing error is apparently larger than the case of Gaussian and Ricean fading channel. Similar to Ricean channel, the strategy of symbol timing adjustment can not be a complete synchronization. Fine symbol synchronization has to provide a buffer zone of about 5 samples for avoiding estimation error. Note the major target of fine symbol synchronization is to ensure that the symbol timing do not drift to ISI region due to residual SCO.

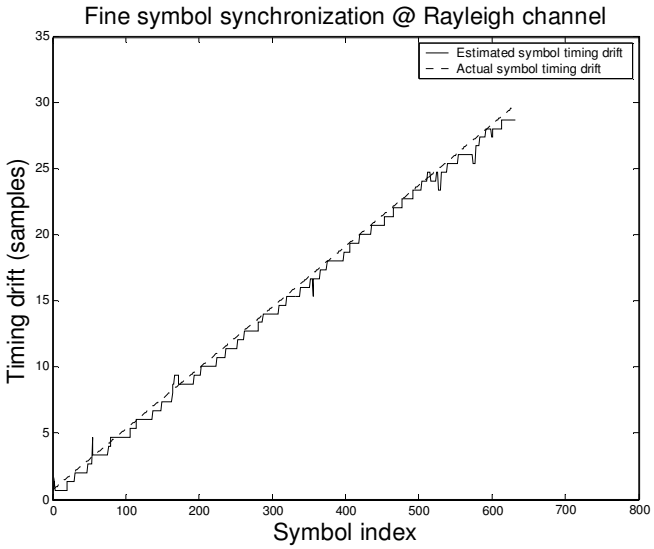


Fig 3.21 Path delay estimation in Rayleigh channel

We have designed different criteria for different channels to detect the path delay. The criterion of maximum detection works for the Gaussian and Ricean fading channel. The criterion of Eq(3.14) works for the Rayleigh fading channel as well as the Gaussian and Ricean channel. The simple criterion of maximum detection can be used if the receiver knows in advance that channel has a direct path such as the Gaussian or Ricean channels. In practical situations where the operating environment varies widely, the receiver may not know the type of channels. In such situation, we can always use the criterion of Eq(3.14) for the multipath Rayleigh fading channels.

It's obvious that the fine symbol synchronization with N/2 IFFT method dominates the complexity of overall synchronization system. Therefore, we proposed a low complexity design of fine symbol synchronization. In order to reduce the size of IFFT, we first downsample the estimated channel frequency response and hence the IFFT with fewer points can be adopted. Nevertheless, Downsampling causes the estimation error due to aliasing. The performances of fine symbol synchronization with varied downsampling rate in 2k mode are shown in Fig 3.22.

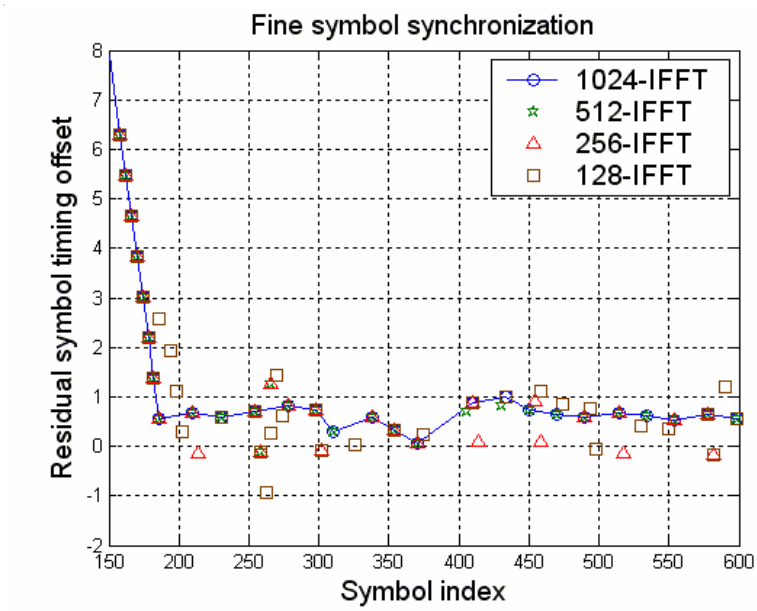


Fig 3.22 Fine symbol synchronization

As we can see, the 1024-IFFT achieves the best performance which keeps the symbol timing during 0 and 1. The larger downsampling rate is adopted, the larger estimation error is generated. Note the estimation will fail as the downsampling rate exceeds 8, i.e. 64-IFFT in 2k mode, because of huge aliasing. In 2k mode, the smallest size of IFFT can be adopted in fine symbol synchronization is 128-point. However, the large variance of estimation results is not reliable enough. Hence, we apply a method using 16-point average window as a lowpass filter prior to downsampling by 16. The aliasing effect can be eliminated so that the estimation variance is reduced. The performance of proposed fine symbol synchronization with N/32 IFFT is depicted in Fig 3.23. The proposed method can keep the symbol timing offset less than 3 samples. The comparisons between proposed method, conventional method and downsampled conventional method are listed in Table 3.2. The root mean square error of proposed method is between N/4 IFFT and N/8 IFFT. However, the complexity reduction is up to 26.67 times from N/2 IFFT to N/32 IFFT.

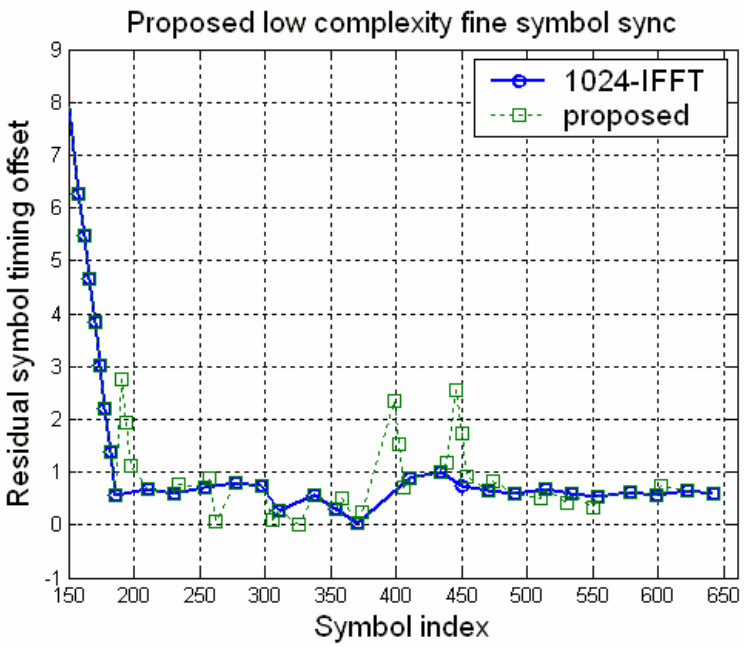


Fig 3.23 Proposed low complexity fine symbol synchronization

	RMSE (sample)	Complexity in 2K mode	Complexity reduction ratio (Radix-2)
N/2 IFFT [3]	0.3556	1024-IFFT	1
N/4 IFFT	0.3686	512-IFFT	2.22
N/8 IFFT	0.4845	256-IFFT	5
N/16 IFFT	0.5320	128-IFFT	11.43
Proposed N/32 IFFT	0.4548	64-IFFT + 16-point average window	26.67

Table 3.2 Comparisons between proposed and conventional fine symbol synchronization

3.3.5 Sampling Clock Synchronization

Sampling clock synchronization consists of three blocks: Resampler, SCO estimator and loop filter. These three blocks constitute a timing recovery loop which tracks out the SCO and

adjusts the timing error. In our receiver design, we propose the fully digital sampling clock synchronization. The relative simulations of each block are illustrated in the following section.

(1) Resampler

As mentioned previously, the resampler exploits FIR filter to interpolate the intermediate points between two consecutive samples. Since the number of taps in FIR filter plays an important role in power consumption, we should reduce the taps of FIR filter as far as possible. However, the truncated sinc function raises the side-lobe amplitude in frequency domain so that the interpolated signals are not as perfect as ideal interpolated signals. In order to improve the finite-tap FIR filter, a Kaiser window is applied as truncation window. The simulations of different numbers of taps with direct truncated sinc function or sinc function truncated by Kaiser window are shown in Fig 3.24.

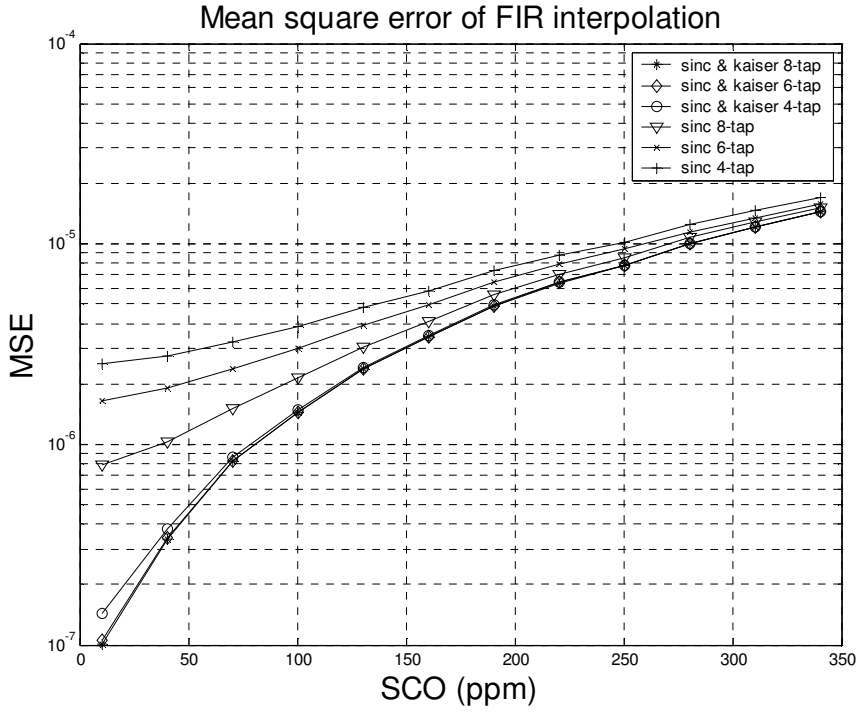


Fig 3.24 MSE of FIR interpolation

For evaluating the performance of interpolated signals, mean square error (MSE) is used to measure the error between transmitted signals and interpolated signals. In this figure, the shape parameter value β of Kaiser window is decided as 4 by simulation. As we can see, the performance is apparently improved after applying Kaiser window. Comparing the sinc function and sinc function truncated by Kaiser with the same taps, we can find a large amount of MSE improvement. Considering the number of taps, the more taps yield the more accurate signals in the case of sinc function. However, the MSE curve tends to saturate at 8-tap in the case of sinc function with Kaiser window. As a result, we can keep the number of taps to 6-tap or 8-tap for the least interpolation error.

(2) Sampling clock offset estimation

There are many SCO estimation algorithms in relative researches, and most of them estimate the CFO jointly. We exploit the LLS (linear least square) algorithm to find the regression line so that the SCO and CFO can be jointly obtained. As mentioned in Chapter 2, reference [14] proposed a "CFD/SFD algorithm" to do joint CFO and SCO tracking. And reference [13] mentioned a "1-D LLS algorithm" which is also based on linear least square method. Table 3.3 lists the one-shot simulation results of the above two algorithms and our proposed algorithm. Three channels are simulated which are Gaussian channel, Ricean channel with Doppler spread 70Hz and Rayleigh channel with Doppler spread 70Hz. We assume the initial SCO is 20 ppm, the residual fractional CFO is 0.02 (subcarrier spacing) and AWGN is 15db.

Initial SCO = 20ppm, residual CFO = 0.02 (subcarrier spacing)						
	Gaussian		Ricean + Doppler spread 70 Hz		Rayleigh + Doppler spread 70 Hz	
SCO estimation	Mean error	Standard deviation	Mean error	Standard deviation	Mean error	Standard deviation
CFD/SFD[14]	2.7990	4.4483	1.9976	7.5357	3.0060	8.3300
1-D LLS[13]	8.8590	4.8811	8.5975	16.0110	7.5885	15.7430
Proposed	0.2182	4.4695	0.4312	16.0934	0.3119	15.9228
CFO estimation	Mean error	Standard deviation	Mean error	Standard deviation	Mean error	Standard deviation
CFD/SFD[14]	0.0012	0.0028	0.0009	0.0040	-0.0024	0.0039
1-D LLS[13]	0.0017	0.0028	0.0016	0.0074	-0.0021	0.0072
Proposed	-0.0002	0.0027	-0.0003	0.0074	-0.0042	0.0072

Table 3.3 Joint SCO and CFO estimation (one shot)

In order to evaluate the estimation error, we compute the mean of one-shot estimation SCO and its standard deviation. As we can see, the CFD/SFD algorithm [14] will generate a negative mean error due to the unequally distributed continual pilots. In contrast to CFD/SFD algorithm, the 1-D LLS algorithm [13] has a larger positive mean error with respect to other two algorithms. Our proposed algorithm has the most accurate mean estimation value but the worse standard deviation than CFD/SFD algorithm. The joint CFO estimation is also listed in **錯誤! 找不到參照來源。** for reference. It is apparently that our proposed algorithm has the best performance in terms of mean value.

For observing the tracking error, we simulate these three algorithms in 648 OFDM symbols. The simulation result is depicted as Fig 3.25. The mean square errors of residual SCO are 0.1487, 0.2133, and 0.1275 respectively. Our proposed algorithm gets the best

performance, and the next is CFD/SFD algorithm.

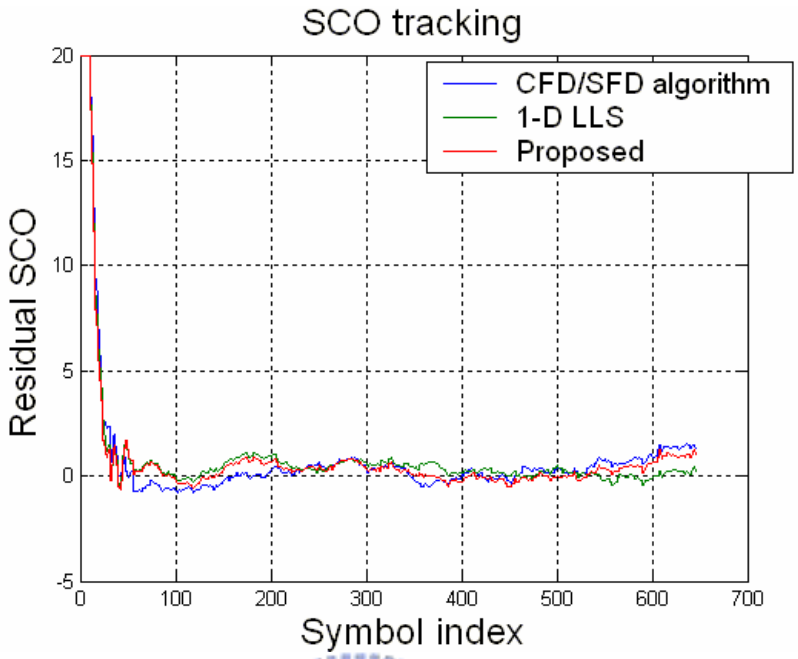
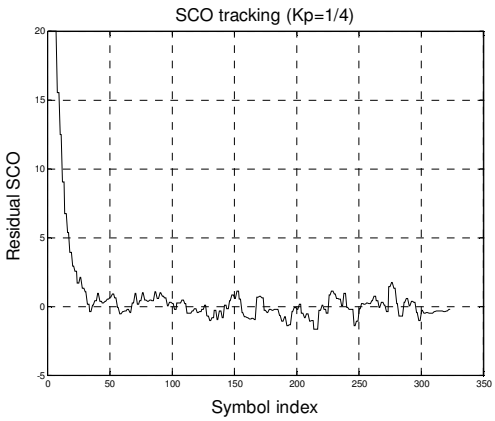


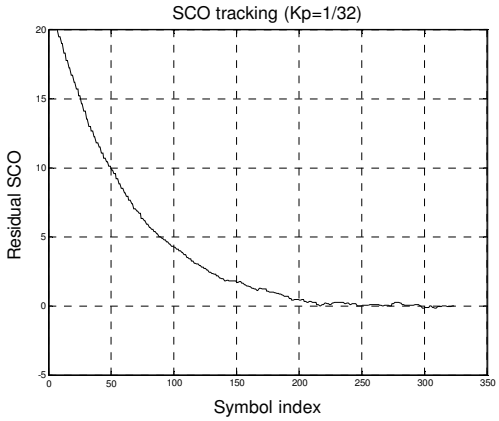
Fig 3.25 SCO tracking

(3) Loop filter

How to choose the parameters of loop filter is a tradeoff between convergence speed and steady-state standard deviation of estimation result. Fig 3.26 shows the convergence speed and corresponding steady-state standard deviation in the different values of K_p.



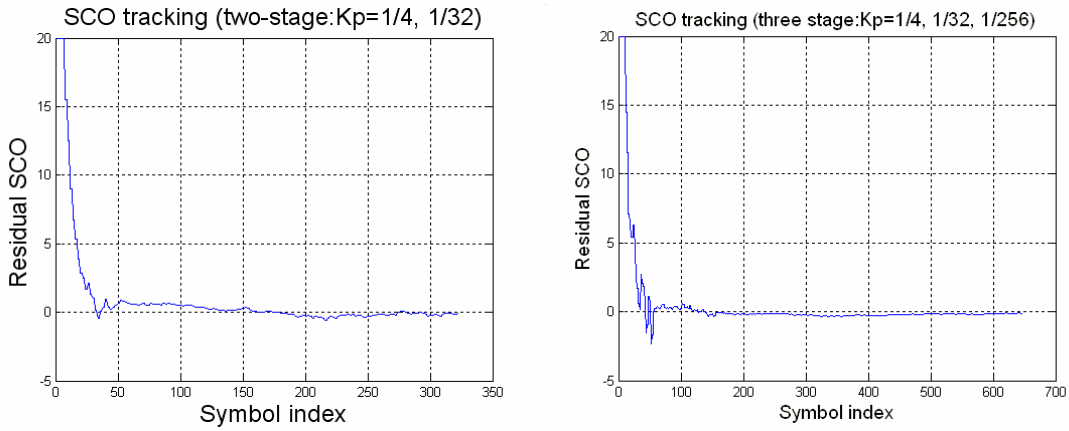
(a) $K_p = \frac{1}{4}$



(b) $K_p = \frac{1}{32}$

Fig 3.26 SCO tracking with different parameter

In above simulation, we assume initial SCO=20 ppm and AWGN=15db. As Fig 3.26(a) is depicted, large K_p yields fast convergence speed of less than 50 samples. However, a huge steady-state mean square error of 0.6714 ppm is also produced which reduce system performance much. And Fig 3.26(b) shows the small K_p causes slow fast convergence speed of more than 200 samples as well as a small mean square error of 0.1287 ppm. As mentioned, the convergence speed requests to be less than one TPS frame which contains 68 OFDM symbols. In order to look after both sides, a multi-stage SCO tracking is adopted as shown in Fig 3.27.



(a) Two-stage (b) Three-stage

Fig 3.27 Multi-stage SCO tracking

Fig 3.27(a) shows two-stage SCO tracking. The initial tracking period uses a large K_p like 1/4 for fast convergence. Afterward, we switch to a small K_p like 1/32 for decreasing steady-state standard deviation. The convergence can be achieved during less than 68 symbols and small mean square error in steady-state can be obtained. In general, the multi-stage SCO tracking is feasible if we do not concern the hardware cost. Multi-stage operation will yield more robust tracking system. Three-stage SCO tracking is depicted in Fig 3.27(b). The third

stage applies a very small K_p 1/256 so that the steady-state standard deviation can be further reduced. In the mobile reception, the multi-stage tracking is needed because of Doppler frequency shift.

3.3.6 Overall System Performance

The overall system performance will be shown in this section. For separating fixed reception from portable reception, we divide the overall system performance into two parts: static channel and mobile channel.

(1) Static channel

The overall system performance in static Gaussian channel, Ricean channel and Rayleigh channel are depicted in Fig 3.28, Fig 3.29 and Fig 3.30 respectively. The transmission works in 2k mode, GI=1/8, code rate=2/3.

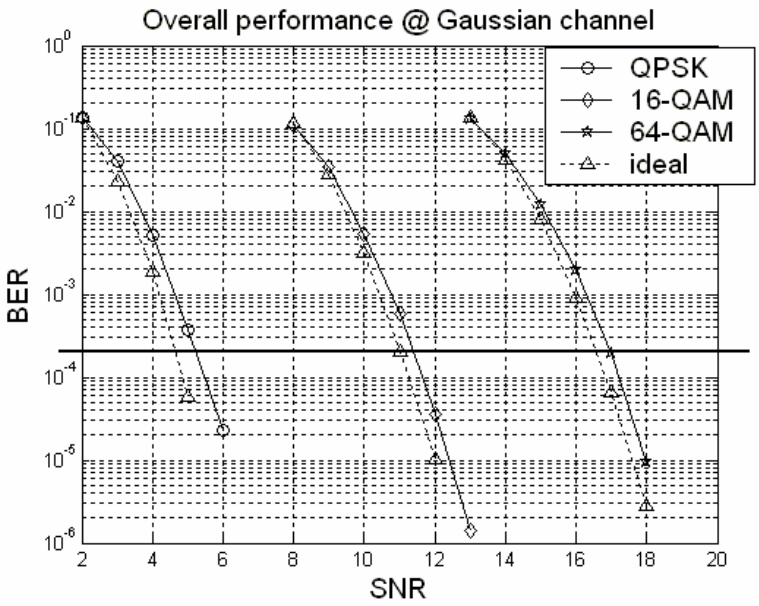


Fig 3.28 Overall system performance in static Gaussian channel

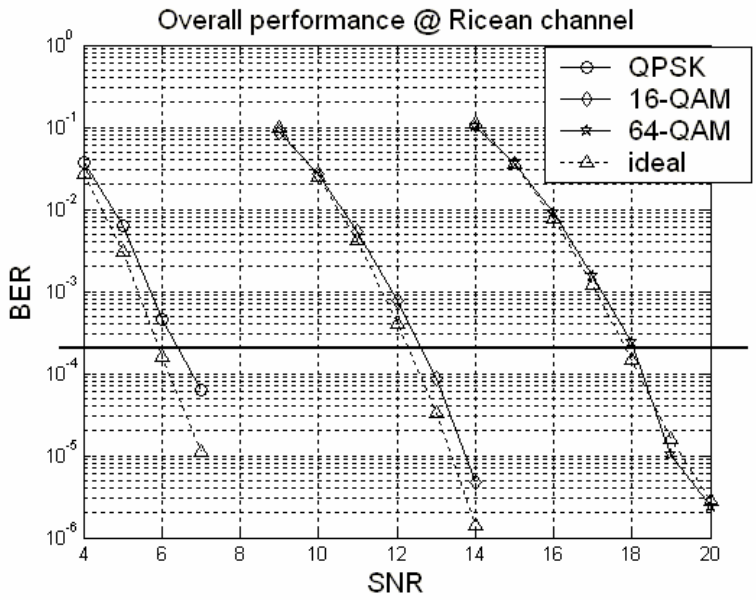


Fig 3.29 Overall system performance in static Ricean channel

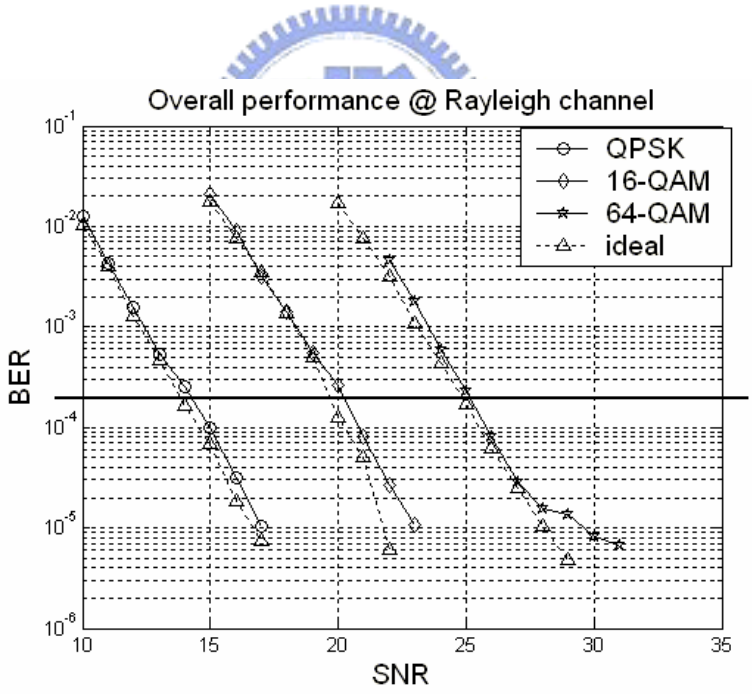


Fig 3.30 Overall system performance in static Rayleigh channel

It's clear that the frequency selective channel leads to an unavoidable degradation of several decibels with respect to nonselective AWGN channel. Severe frequency selectivity caused by very long channel delay profiles means the OFDM spectrum exhibits many notches

which degrades the channel estimation gain and causes the noise enhancement. In the Rayleigh channel, the ideal system performance has SNR degradation of about 8dB compared with AWGN channel. The SNR loss of each case is listed in Table 3.4.

	Gaussian	Ricean	Rayleigh
QPSK	0.38	0.50	0.44
16-QAM	0.38	0.33	0.56
64-QAM	0.41	0.21	0.31

Table 3.4 SNR loss in static Gaussian, Ricean and Rayleigh channel

(2) Mobile channel

In the case of mobile reception condition, the Doppler spread has to be considered. For providing practical environment simulation, we assume the maximum Doppler frequency as 70 Hz which is corresponding to velocity of 150 km/h. The overall system performance in Rayleigh channel with Doppler spread 70Hz is shown in Fig 3.31. The simulation performs in 2k mode, GI=1/8 and code rate=2/3.

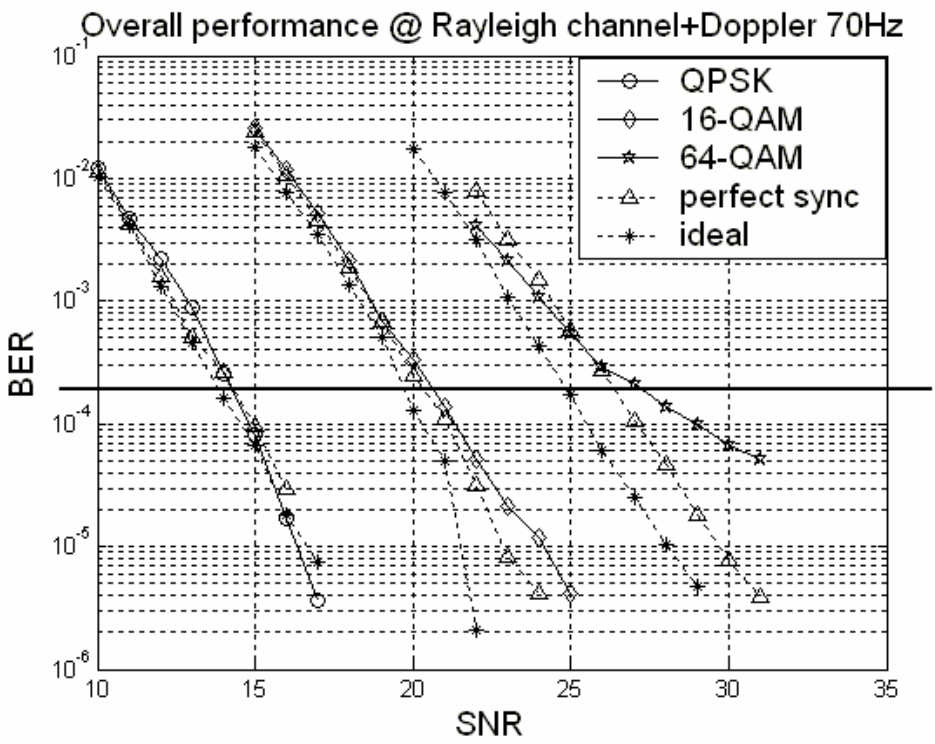


Fig 3.31 Overall system performance in Rayleigh channel with Doppler frequency 70Hz

The Doppler tolerance varies with transmission mode and constellation mapping. Total SNR loss in 64-QAM is 2.27dB which is larger than 0.42dB in QPSK. A further interest is the additional degradation caused by the synchronization algorithms. The synchronization loss is less than 1dB in every transmission mode. The detail values of synchronization loss and total SNR loss are listed in Table 3.5.

	QPSK	16-QAM	64-QAM
Synchronization loss	~0	0.32	0.81
Total SNR loss	0.42	0.92	2.27

Table 3.5 Synchronization loss and total SNR loss in mobile channel

Chapter 4
Architecture of Timing Synchronization Scheme

This chapter will discuss some considerations of hardware implementation in DVB-T system. The architecture of each timing synchronization block is constituted based on the concepts of hardware sharing.

4.1 Architecture of Symbol Synchronization

Since the basic concepts of every blocks of pre-FFT acquisition are all based on guard interval correlation, the architecture designs of Mode/GI detection, coarse symbol synchronization and pre-FFT AFC can combined based on hardware sharing. Fig 4.1, Fig 4.2 and Fig 4.3 shows the basic modules of Mode/GI detection, coarse symbol synchronization and pre-FFT AFC respectively.

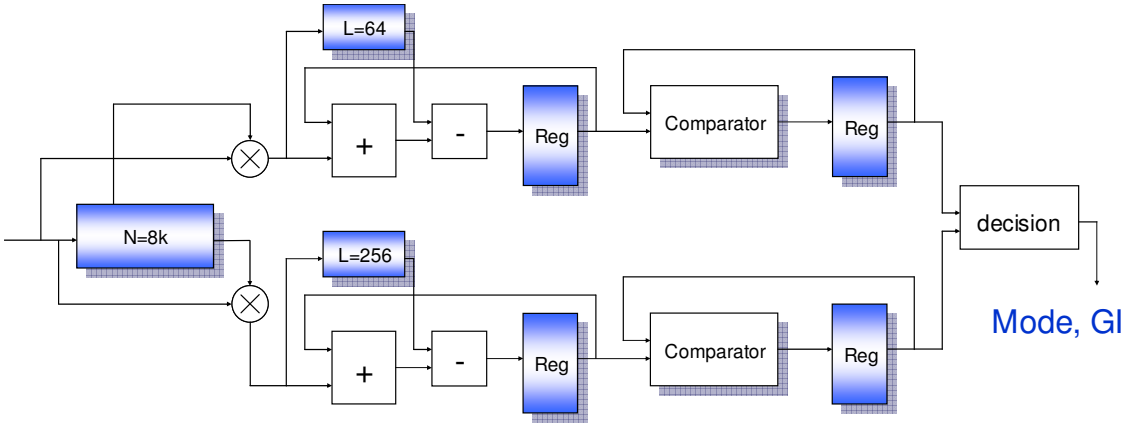


Fig 4.1 Architecture of Mode/GI detection

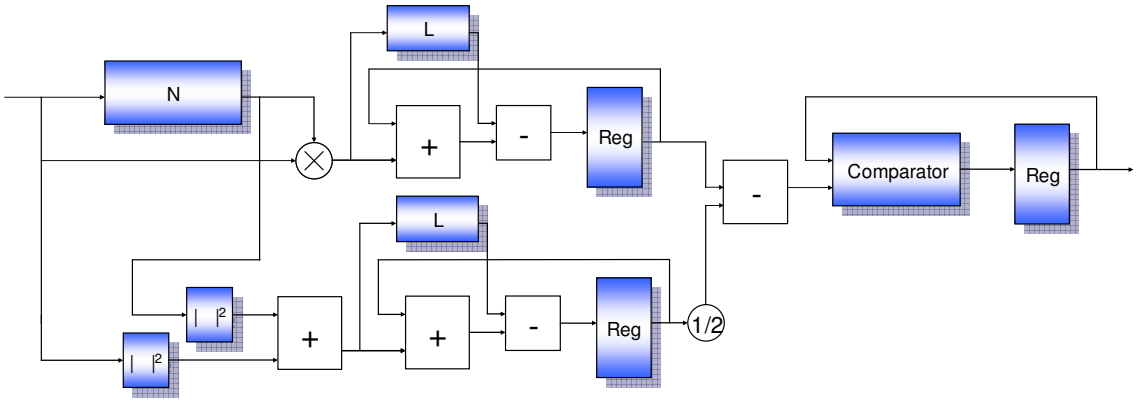


Fig 4.2 Architecture of coarse symbol synchronization (MMSE)

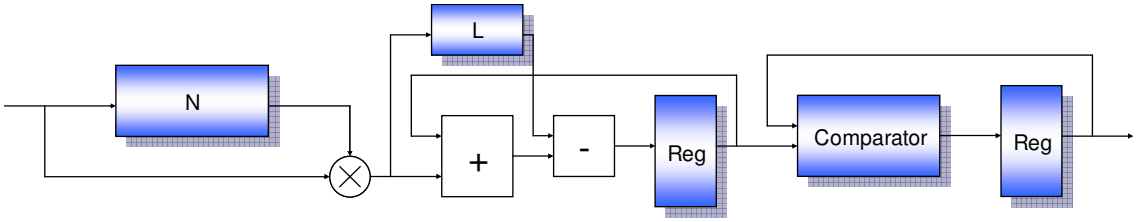


Fig 4.3 Architecture of pre-FFT AFC

As we can see, these three blocks have the common operations which contain correlation and moving sum. And the operations would not overlap since they are all operations in acquisition mode. Therefore, hardware sharing is certainly practical and necessary.

As for fine symbol synchronization, the major operation is IFFT. In DVB-T system, symbol duration is much longer than other communication systems. In general, FFT operation time can be much shorter than OFDM symbol duration if we speedup our FFT design. As a result, the 4k-point IFFT operation can utilize the FFT block without any other hardware cost. However, this strategy raises the FFT operation rate so that the gate count of FFT block enlarges deservedly. Another consideration of IFFT operation is that it may suffice to implement an IFFT which contains only one butterfly unit. Since the timing changes slowly, the speed of IFFT is not important. Hence the fine symbol synchronization can execute one times per 3 symbols for example. So that the main hardware concern is the memory needed. In summary, the hardware implementation of IFFT design can probably choose one of these

two strategies.

The architecture of the last acquisition operation – scattered pilot mode detection is depicted in Fig 4.4. The architecture consists of four paths which accumulate the subcarrier power. And then a comparator is applied to choose the maximum mode.

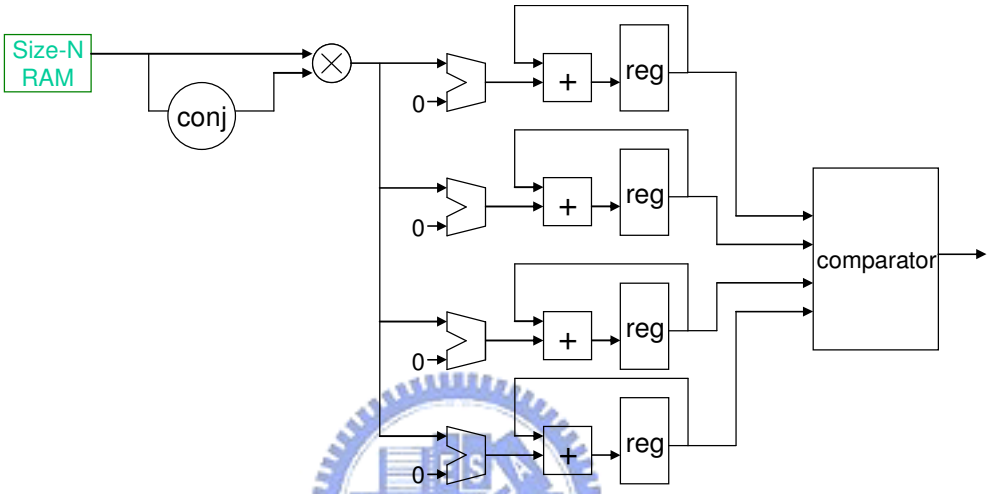


Fig 4.4 Architecture of scattered pilot mode detection

4.2 *Architecture of Sampling Clock Synchronization*

Sampling clock synchronization consists of three blocks: resampler, SCO estimation and loop filter. The architecture of resampler is depicted in Fig 2.26. Since the resampler operates all the time, how to reduce its power consumption is very important. As illustrated in Chapter 2, resampler combines interpolator and decimator so that the power consumption reduces nearly half.

The architectures of SCO estimator and loop filter are shown in Fig 4.5 and Fig 2.27 respectively. Note the *Arg* operation in SCO estimator should exploit the ROM table instead of using divider.

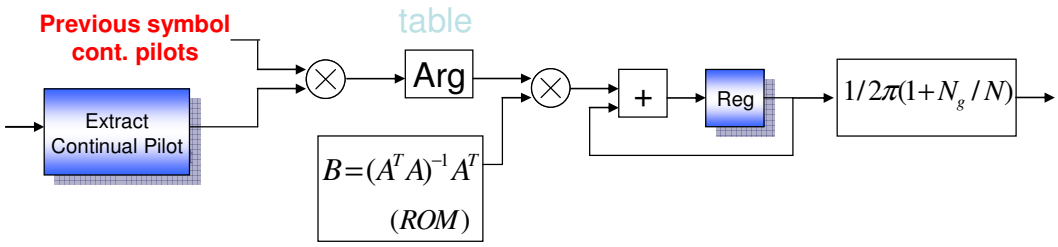


Fig 4.5 Architecture of joint SCO and CFO estimator

In the baseband of DVB-T system, the most important issue in hardware implementation is the efficiency of memory sharing. DVB-T system, in particular, has very large size of OFDM symbols so that the memory requirement is huge. Furthermore, in channel estimation design, the 2-D interpolation method is commonly exploited for resisting Doppler spread. The 2-D interpolation channel estimation acquires the memory of three and 1/3 complete OFDM symbols. If the receiver supports 8K mode transmission, it leads the amazing required memory size. In [20], an embedded DRAM unit performs all memory operations such as buffering three OFDM symbols in order to enable channel time interpolation, extracting scattered pilot for channel estimation and fine symbol synchronization, and extracting TPS pilot for TPS detection. Separated memory should not be applied in DVB-T system due to its inefficiency.

Another thought about hardware implementation of DVB-T system is to exploit a DSP processor. Since the basic operations within the baseband DVB-T system are similar like correlation and accumulation, the DSP processor probably can perform in an efficient way to compute all the operations. Unlike ASIC design, DSP processor based design will lead to a very complicated control of data path. But the advantage of fewer hardware cost is expectable.

Chapter 5
Conclusion and Future Work

The complete DVB-T digital baseband receiver design is accomplished in Matlab. The simulation platform can evaluate the overall system performance in each transmission mode and each channel model. Our system can resist strong multipath channel, large CFO, SCO, AWGN and Doppler frequency spread. The timing synchronization scheme including symbol synchronization and sampling clock synchronization is presented. The proposed low complexity fine symbol synchronization can reduce the computational complexity effectively. The conventional method of N/2 IFFT has been improved by applying an average window and therefore N/32 IFFT can be adopted in fine symbol synchronization. The symbol timing offset is kept in less than 3 samples so that the ISI can be avoided. On the other hand, a new sampling clock offset estimation algorithm based on linear least square error is proposed in this thesis. The proposed SCO estimation can overcome the problem of unequally distributed continual pilots in conventional designs and hence the steady-state mean square error of estimation results can be improved. The sampling clock synchronization can limit the residual SCO to $\pm 1 ppm$. With multi-stage synchronization loop, the system achieves high performance due to small parameter K_p of loop filter. The overall timing synchronization achieves a high tolerance to severe channel distortion including Rayleigh channel, Doppler spread 70Hz, CFO 20ppm and SCO 20ppm.

In the future, the channel estimation unit can be improved by applying Wiener interpolation since the system performance is heavily dominated by channel estimation gain. The resampler design can probably apply the polynomial interpolation to reduce the hardware

cost instead of classical FIR interpolation. There are several other interpolation techniques such as Farrow structure and Lagrange interpolation. We have to find out the best method of interpolation for DVB-T system in the future. In the equalization design, there are some robust techniques like MAP (maximum a posteriori) equalizer and SOVA (soft-output Viterbi algorithm) equalizer. Instead of conventional equalizer without any feedback, these noise prediction equalizers can gain several decibels of system performance especially in strong multipath channel such as Rayleigh channel. In the recent relative materials [22][23], we can discover the required C/N values in ETSI standard have some errors. Therefore, in our future work, we have to consult more other researches for system performance verification. At last, the fixed point simulation should be performed so that the hardware implementation is able to verify our proposed algorithms in silicon-proven.

Bibliography

[1] ETSI, "Digital Video Broadcasting: Framing Structure, Channel coding, and Modulation for Digital Terrestrial Television," *European Telecommunication Standard* EN 300 744 V1.4.1, Jan, 2001.

[2] 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, Nov. 1999.

[3] 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, Apr. 2001

[4] B. Yang, K. Letaief, R. Cheng and Z. Cao, "Timing Recovery for OFDM Transmission," *IEEE Journal on Select. Commun.*, vol. 18, No. 11, Nov. 2000

[5] S. Chen, W. He, H. Chen and Y. Lee, "Mode Dectection, Synchronization, and Channel Estimation for DVB-T OFDM Receiver," *GLOBECOM '03. IEEE*, vol. 5, Dec. 2003.

[6] D. Lee and K. Cheun, "Coarse Symbol Synchronization Algorithms for OFDM Systems in Multipath Channels," *IEEE Commun. Letters*, vol. 6, No. 10, Oct. 2002.

[7] D. Landstorm, S. Wilson, J. Beek, P. Odling and P. Borjesson, "Symbol Time Offset Estimation in Coherent OFDM Systems," *IEEE Trans. Commun.*, vol. 50, No. 4, Apr. 2002.

[8] J. Kim and D. Han, "Adaptive Frame Selection Algorithm For DVB-T," *IEEE Trans. Consumer Electronics*, vol.48, Aug. 2002.

[9] M. Speth, F. Classen and H. Meyr, "Frame Synchronization of OFDM Systems in Frequency Selective Fading Channel," *Vehicular Technology Conference'97 IEEE*, vol. 3, May. 1997.

[10] A. Palin and J. Rinne, "Enhanced Symbol Synchronization Method for OFDM System in SFN Channels," *IEEE Globecom'98*, vol. 5, 8-12 Nov. 1998

[11] P. Robertson and S. Kaiser, "The Effects of Doppler Spreads in OFDM(A) Mobile Radio Systems," *Vehicular Technology Conf.*, vol. 1, 19-22 Sept. 1999

[12] H. Meyr, M. Moeneclaey and S. Fechtel: *Digital Communication Receivers*, Wiley Interscience, 1997.

[13] S. Liu and J. Chong, "A Study of Joint Tracking Algorithms of Carrier Frequency Offset and Sampling Clock Offset for OFDM-based WLANs," *IEEE International Conf. Communications, Circuits and Systems and West Sino Expositions*, vol. 1, July. 2002.

[14] S. Fechtel, "OFDM Carrier and Sampling Frequency Synchronization and Its Performance on Stationary and Mobile Channels," *IEEE Trans. Consumer Electronics*, vol. 46, No. 3, Aug 2000.

[15] Louis Litwin, "Matched Filtering and Timing Recovery in Digital Receivers, " *RF design*, pp32-48, Sept. 2001.

[16] T. Schmidl and D. Cox, "Robust frequency and timing synchronization for OFDM," *IEEE Trans. Commun*., vol. 45, Dec. 1997.

[17] S. Bug, C. Wengerter, I. Gaspard and R. Jakoby, "WSSUS-Channel Models for Broadband Mobile Communication Systems," *Vehicular Technology Conf.*, vol. 2, May. 2002.

[18] W. C. Jakes, *Microwave Mobile Communications*. John Wiley & Sons, Inc., 1974.

[19] P. H. Moose, "A Technique for Orthogonal Frequency Division Multiplexing Frequency Offset Correction," *IEEE Trans. Commun.*, vol. 42, No. 10, Oct. 1994.

[20] S. Fechtel and P. Schollhorn, "Advanced Receiver Chip for Terrestrial Digital Video Broadcasting: Architecture and Performance," *IEEE Trans. Consumer Electronics*, vol .44, No. 3, Aug. 1998

[21] A. Oppenheim, R. Schafer and J. Buck, Discrete-Time Signal Processing. Prentice Hall International, Inc., 1998

[22] R. Schramn, “DVB-T – C/N values for portable single and diversity reception,” EBU Technical review, April, 2004

[23] Notes of DVB-H Meeting #16 10th and 11th June 2004.