# A run-time complexity adaptive channel estimator for CMMB

Yu, Z.

*Award date:*
2011

Eindhoven University of Technology
Department of Mathematics and Computer Science

# A Run-time Complexity Adaptive Channel Estimator for CMMB

Zhibin Yu
(0730770)

Supervisor: Prof. Dr. C.H. van Berkel
Dr. Hong Li

July, 2011, Eindhoven

# Abstract

CMMB (China Multimedia Mobile Broadcasting) is a digital TV broadcasting standard in China. It adopts OFDM as the modulation scheme in its physical layer. Channel estimation is the most computation intensive task in a CMMB baseband receiver. In order to save the power consumption in the baseband receiver, the thesis proposes a run-time complexity adaptive channel estimator, in which the processing load for channel estimation per CMMB time slot can be scaled according to the run-time estimated channel quality.

The thesis is organized as follows: first, the CMMB standard and the CMMB receiver platform are introduced. Then, the OFDM baseband model for CMMB is formulated and two important properties of a multi-path channel are analyzed: short-delay limitation and energy leakage. And then, practical channel estimation algorithms with scalable processing loads and scalable estimation accuracies are developed. Next, a novel noise variance estimation algorithm and a novel delay spread estimation algorithm using CMMB preamble, are developed. Later on, a power manager, which specifies the selection criterion under given channel quality parameters, is designed. Finally, the derived algorithms are mapped onto NXP vector DSP engine (VDSP), and the number of processing cycles for each algorithm is derived. The obtained load profile is further inserted into the Matlab test bench for integrated simulation. The simulation is performed under a pre-defined scenario which is consistent with CMMB standard, and the simulation results show more than 50% of run-time load reduction is achieved comparing with a fixed channel estimator. This results in a longer battery life in high quality channels while maintaining the satisfactory reception quality in poor quality channels.

# Key Words:

China multimedia mobile broadcasting (CMMB); orthogonal frequency division multiplexing (OFDM); channel estimation; channel quality estimation; vector digital signal processor (VDSP); complexity adaption

# Glossary

| | |
|---|---|
| AWGN | Additive White Gaussian Noise |
| BER | Bit Error Rate |
| BPSK | Binary Phase Shift Keying |
| CE | Channel Estimation |
| CDR | Correct Detection Rate |
| CIR | Channel Impulse Response |
| CQE | Channel Quality Estimation |
| CMMB | China Multimedia Mobile Broadcasting |
| CFO | Carrier Frequency Offset |
| CP | Cyclic Prefix |
| DFE | Digital Front End |
| DFS | Dynamic Frequency Scaling |
| DFT | Discrete Fourier Transform |
| DVS | Dynamic Voltage Scaling |
| FFT | Fast Fourier Transform |
| FIR | Finite Impulse Response |
| IDFT | Inverse Discrete Fourier Transform |
| IFFT | Inverse Fast Fourier Transform |
| LDPC | Low Density Parity Check |
| LMMSE | Least Minimal Mean Square Error |

| | |
|---|---|
| LS | Least Square |
| LTI | Linear Time Invariant |
| MSE | Mean Square Error |
| OFDM | Orthogonal Frequency Division Multiplexing |
| PDP | Power Delay Profile |
| PN | Pseudo Noise |
| QAM | Quadrature amplitude modulation |
| RF | Radio Frequency |
| SDR | Software Defined Radio |
| SFO | Sampling Frequency Offset |
| SIMD | Single Instruction Multiple Data |
| SNR | Signal Noise Ratio |
| SP | Scattered Pilot |
| TS | Time Slot |
| TSVD | Truncated Singular Value Decomposition |
| VDSP | Vector Digital Signal Processor |
| VLIW | Very Long Instruction Word |
| WUSSUS | Wide-Sense Stationary Uncorrelated Scattering |

# Contents

# 1. Introduction

In this chapter, first, the CMMB standard is briefly introduced. And then, the platform for CMMB baseband processing is introduced. Finally, CMMB baseband processing tasks on the target platform is described.

## 1.1. CMMB Overview

CMMB is short for China Multimedia Mobile Broadcasting. It's a digital TV broadcasting standard specified by the State Administration of Radio, Film, and Television (SARFT) of China. CMMB uses the carrier frequencies range 2635 – 2660 MHz (S-band) for satellite broadcasting, and the carrier frequencies range 470 – 862 MHz (UHF band) for additional terrestrial broadcasting. In the physical layer, CMMB adopts Orthogonal Frequency Division Multiplexing (OFDM) as the modulation scheme and LDPC-RS code as the channel code. CMMB offers 3 different bit-symbol mapping schemes and 2 bandwidth modes to satisfy different data rate requirements. The coded data rates of CMMB under different configurations are shown in Table 1.1.

**Table 1.1 Coded data rates of CMMB**

| Bit-symbol mapping scheme | Data rates in 1k mode (2MHz bandwidth) | Data rates in 4k mode (8MHz bandwidth) |
|---|---|---|
| BPSK | 1.10664M bps | 4.42656M bps |
| 4-QAM | 2.21328M bps | 8.85312M bps |
| 16-QAM | 4.42656M bps | 17.70624M bps |

The CMMB standard in physical layer is in details described in [1]. In this section, basic knowledge of OFDM, the CMMB frame structure and the CMMB spectrum structure are introduced, which help to understand the concepts of this study.

### 1.1.1. OFDM

OFDM is a multi-carrier modulation method, which uses a large number of closely-spaced orthogonal subcarriers to carry data. For CMMB, each OFDM useable subcarrier carries a complex-valued data symbol mapped by BPSK or 4-QAM or 16-QAM [1]. The subcarriers are assumed to be frequency-domain signals. OFDM modulation is performed by transforming the frequency-domain signals into the time-domain signals using IDFT, and OFDM demodulation is performed by transforming the received time-domain signals back into the frequency domain using DFT. One key feature of the OFDM method is to migrate the inter-sample interference caused by a multi-path channel in the time domain to be multiplicative frequency selective fading in the frequency domain. And if the channel frequency response is known or can be accurately estimated in the OFDM receiver, the signal distortions caused by frequency selective fading can be well equalized. Benefit from this feature, OFDM has the advantage to be robust against multi-path propagations and can easily adapt to severe channel conditions without complex time-domain equalization [2]. The OFDM based band model for CMMB is mathematically formulated in Chapter 3.

### 1.1.2. CMMB Frame Structure

The CMMB data steam is organized by frames. Each CMMB frame lasts for 1 second and consists of 40 time slots (TS) as synchronization units. Each TS lasts for 25ms and is headed with a preamble and then followed by 53 OFDM data blocks. Each CMMB Preamble contains two synchronization signals, which are two identical IDFT modulated PN sequences with unit power. Each OFDM data block contains a modulated useful data body. The CMMB data structure is shown in Figure 1.1.

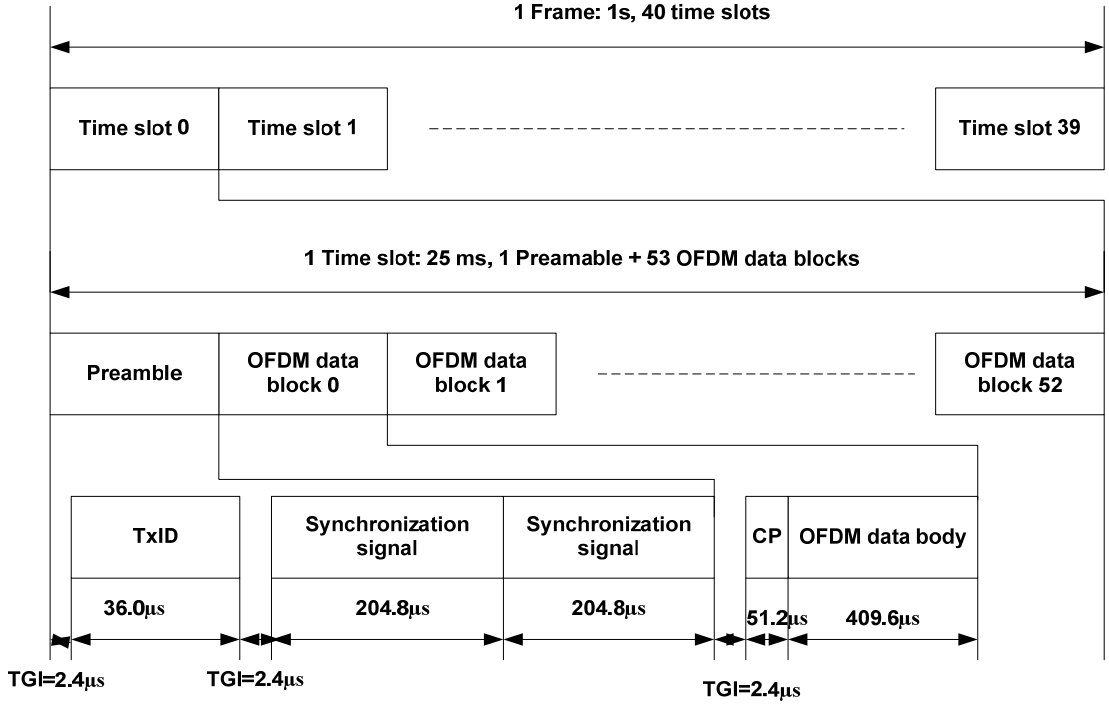


**Figure 1.1 CMMB frame structure**

### 1.1.3. CMMB Spectrum Structure

The spectrum structure of CMMB data body is shown in Figure 1.2, where the usable-subcarriers are guarded by null-subcarriers at two guard bands.

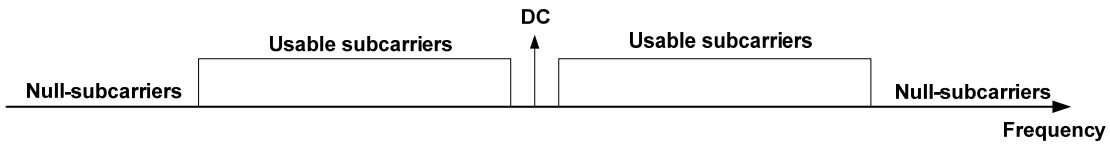


**Figure 1.2 CMMB spectrum structure**

In order to enable the baseband processing such as synchronization and equalization, two types of pilots are inserted into the usable-subcarrier band: continuous pilot and scattered pilot. Continuous pilots are pre-defined symbols with the same pattern in every OFDM data block, and are often used for carrier frequency offset (CFO) and sampling frequency offset (SFO) estimation and tracking. Scattered pilots (SP) are also pre-defined symbols and are inserted into OFDM data body in a higher density, which are often used for channel estimation [3]. The pattern of SPs within OFDM data

bodies is shown in Figure 1.3, which is an interlaced pattern in order to improve the data utilization efficiency. The distance between two neighboring SPs within one OFDM data block is 8 subcarriers.

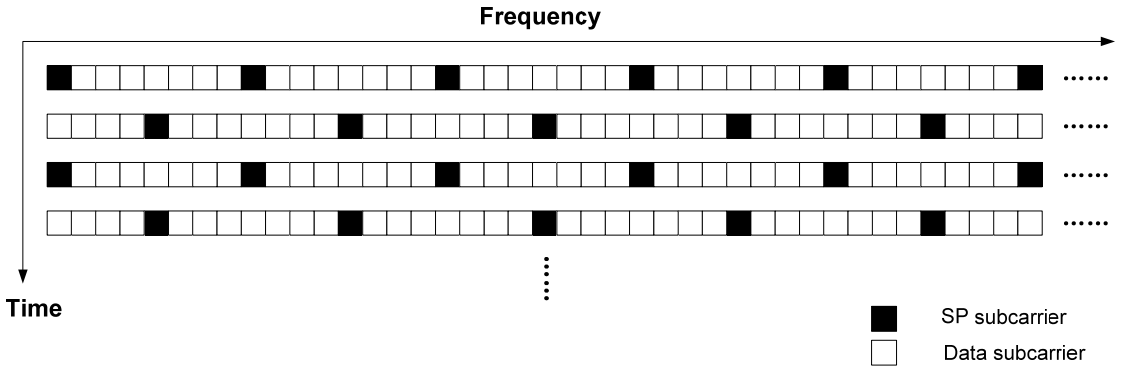


**Figure 1.3 SP pattern in the usable-subcarrier band**

## 1.2. Platform Overview

The CMMB receiver is realized on a software defined radio (SDR) platform. The platform can be abstracted into Figure 1.4, where the gray scale of a hardware block shows the programmable flexibility of it. It works as this: the tuner receives radio frequency (RF) signals, applies frequency down-conversion, and over-samples the analog waveform into digital bits. The digital front-end (DFE) sub-samples the digital bits, performs pre-filtering and then sends the data to vector digital signal processor (VDSP) for baseband processing (the inner-receiver). The demodulated soft bits from VDSP are further sent to a codec for de-interleaving and channel decoding (the outer-receiver). Eventually, the decoded bits are sent out as standard video transport stream.

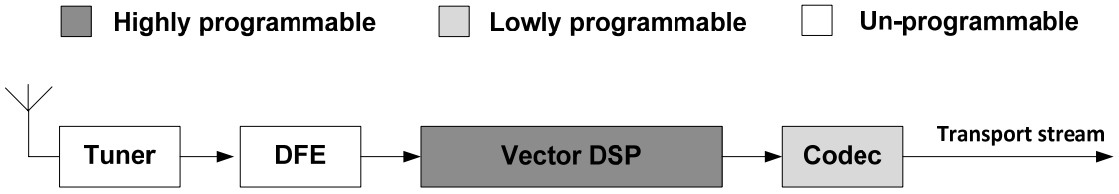


**Figure 1.4 Platform abstraction**

In Figure 1.4, VDSP is a customized digital signal processor which contains a Vectra LX DSP engine as well as the extended application specific function units for OFDM baseband processing [4]. The maximal clock speed of VDSP is 369MHz (at 1.08 volts) [5]. VDSP combines the very long instruction word (VLIW) and single instruction multiple data (SIMD) to explore the instruction-level parallelism and the data-level parallelism. The VDSP block diagram is shown in Figure 1.5.

Programming VDSP is done by using Tensilica-provided toolset, which is compatible with C language. The extended function units in VDSP are supported by Tensilica Instruction Extension (TIE) set [5]. Typical TIE instructions are 4-way complex multiply-addition, 4-way complex conjugate multiply-addition, 8-way parallel load, 8-way parallel store, 8-way sum of square, 8-way maximum and radix-4 butterfly operations. These instructions are extremely suitable for digital signal

processing algorithms such as data correlation, FIR filtering, power computing, peak searching, FFT and IFFT, and can be finished in one clock cycle.

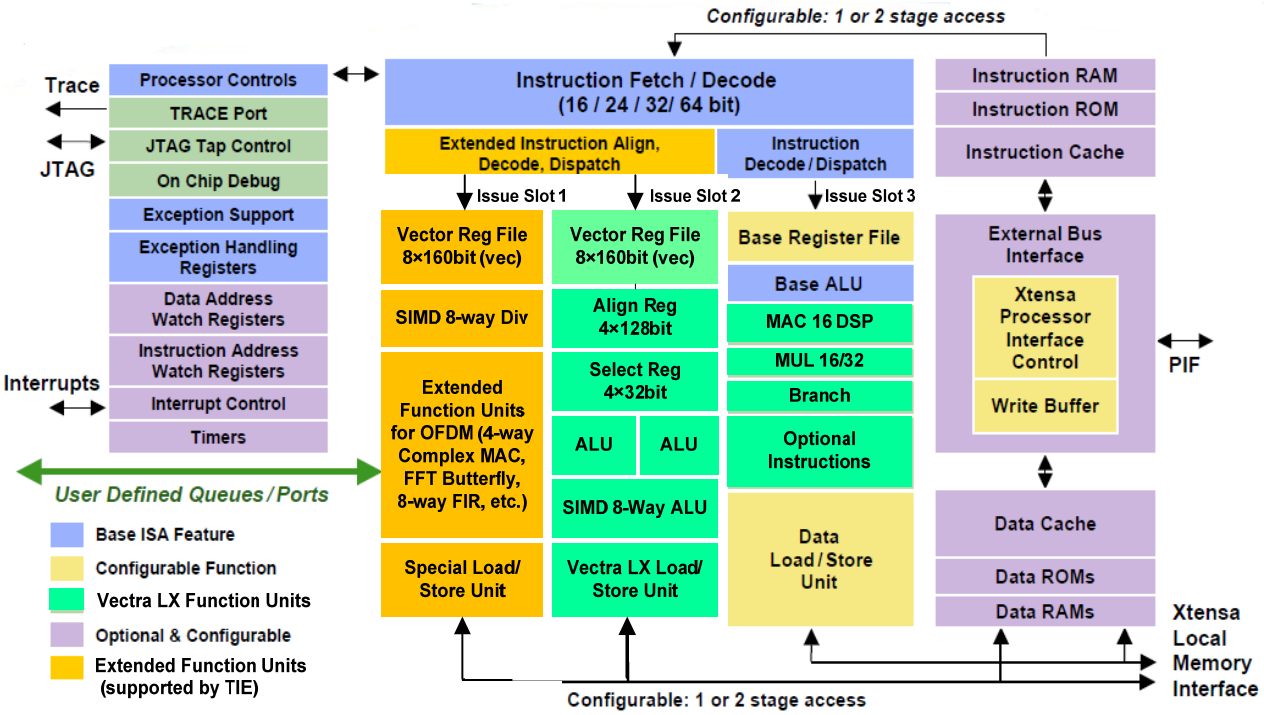


**Figure 1.5 VDSP block diagram**

VDSP provides the highly programmable flexibility to be able to support the baseband processing for many wireless communication standards such as DVB-T and CMMB. However, as a trade-off for flexibility, VDSP has limited computation capacity and higher power consumption comparing with ASIC chips. Such computation inefficiency and power inefficiency of VDSP are usually caused by the overhead of instruction fetch & decode, branch control, and register load & store. Therefore, how to design efficient baseband processing algorithms for VDSP is a challenge.

## 1.3. Baseband Processing Overview

For CMMB, typical baseband processing tasks in VDSP are showed in Figure 1.6, where VDSP runs in two modes: the acquisition & synchronization mode and the demodulation mode. The acquisition & synchronization mode aims at acquiring the FFT window location of CMMB data stream and coarsely estimating the CFO. It's only run when the system is set-up in the beginning; The demodulation mode aims at continuously demodulating the CMMB data stream and is run in the majority of time.

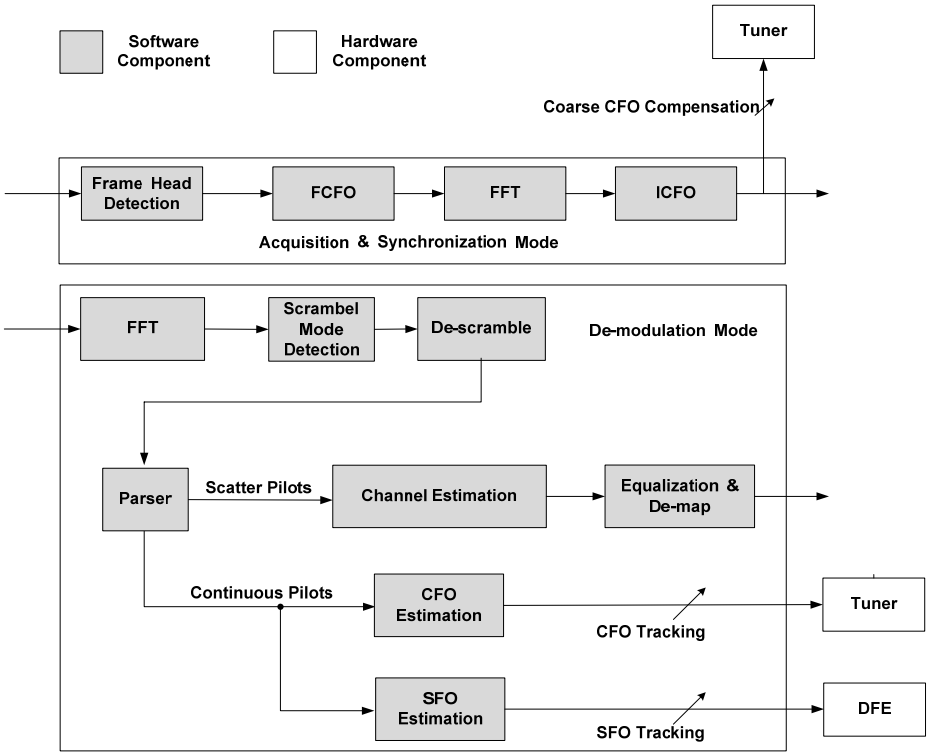


**Figure 1.6 Baseband processing tasks mapped onto VDSP**

Among the baseband processing tasks in VDSP, channel estimation is crucial for the reception quality. That is because radio propagation suffers from the fading of wireless channels. Channel estimation aims to estimate current channel transfer function in order to make compensations for channel fading. Channel estimation is also computation intensive because in order to track the change of a wireless channel, it has to be played for every OFDM block. [6] analyzes the arithmetic complexity of each task in the CMMB inner-receiver and concludes that channel estimation (18-tap FIR Wiener) takes up about 50% of the total computation load and is the most computation intensive task in the CMMB inner-receiver. And therefore it consumes most of power in the CMMB inner-receiver. The proportion for the power consumption caused by channel estimation in the CMMB receiver platform is illustrated in Figure 1.7.

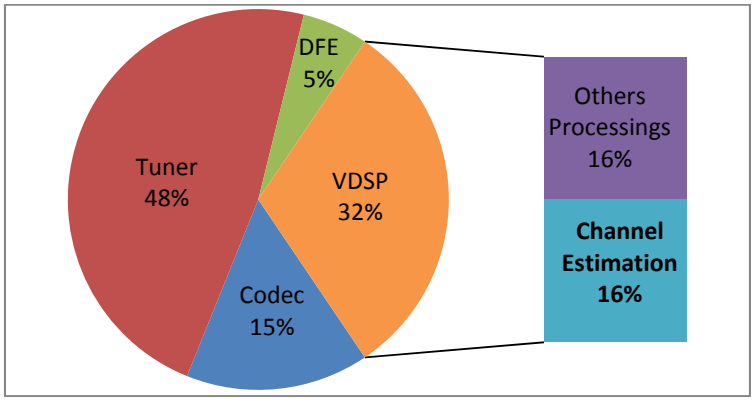


**Figure 1.7 Coarse power distributions in the receiver platform**

## 1.4. Chapter Conclusion

CMMB adopts OFDM as the modulation scheme in its physical layer. In the CMMB receiver platform, VDSP works as the CMMB inner-receiver. VDSP provides highly programmable flexibility for baseband processing, but has limited computation capacity and higher power consumption comparing with ASIC chips. Channel estimation is crucial for the reception quality and is the most computation intensive task in the CMMB inner-receiver.

# 2. Problem Statement

In this chapter, the motivation of this thesis is introduced. And then, the key problems to solve in the thesis work are presented. Finally, the approaches which are used in the thesis work to solve the problems are described.

## 2.1. Motivation

In VDSP (the inner-receiver), in order to guarantee the reception quality, channel estimation algorithm has to be performed in order to deal with signal distortion and noise interference which are introduced by mobile radio propagation. However, the limited battery capacity becomes contradict to the highly demanded processing load in VDSP. One idea to solve this contradiction is to build a complexity adaptive architecture as shown in Figure 2.1.

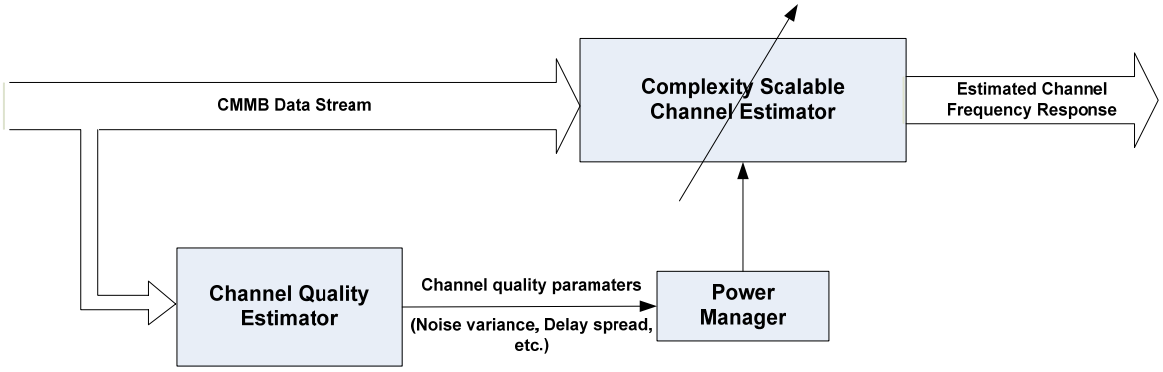


**Figure 2.1 Motivation architecture of run-time complexity adaptive channel estimator for CMMB**

In Figure 2.1, channel quality parameters, which reflect the degree of channel fading and the degree of noise interference in the current wireless communication channel, are firstly estimated by a channel quality estimator. And then, a power manager chooses an appropriate channel estimation algorithm so as to scale the processing load accordingly in the run-time: when the interference is light, simple channel estimation algorithms are used in the sense of power saving; when the interference is heavy, complex channel estimation algorithms are used in order to guarantee the reception quality.

## 2.2. Problem

In order to realize the architecture in Figure 2.1, the following problems are treated in this thesis work.

- How to develop a set of channel estimation algorithms for CMMB with scalable processing loads and scalable estimation accuracies? The derived algorithms should be practical to be implemented in VDSP.

- How to estimate channel quality parameters (typically noise variance and delay spread) using CMMB data structure? The derived algorithms should be practical to be implemented in VDSP, and should introduce small computation overheads.
- How to design a power manager which aims to optimize the trade-off between the processing load and the channel estimation accuracy?
- How to verify the idea and how to evaluate the performance of the derived run-time complexity adaptive channel estimator in the design phase? More specifically, how to evaluate the cycle reduction factor comparing with a fixed channel estimator in the design phase?

## 2.3. Approach

The primary approaches used in the thesis work are mathematical analysis and computer simulation. The design procedure is as follows:

Firstly, the OFDM baseband model for CMMB is formulated. And then, channel estimation algorithms and channel quality estimation algorithms are designed and separately evaluated in Matlab. Next, based on simulation results, a power manager is designed according to a pre-defined scenario, in which the power manager is able to select the most economical channel estimation algorithm according to the estimated channel quality. Later on, the derived algorithms are mapped onto a VDSP simulator, and the number processing cycles for each algorithm are recorded. Finally, the obtained load profile for each algorithm module is inserted into a Matlab test bench for integrated simulation in the pre-defined scenario, and the run-time cycle reduction factor comparing with a fixed channel estimator is evaluated.

## 2.4. Chapter Conclusion

This chapter introduces the motivation, problem and approach of the thesis. It's important to distinguish two notations: channel estimation (CE) and channel quality estimation (CQE). Channel estimation is an indispensable processing stage in the OFDM inner-receiver, aiming to deal with frequency selective fading in wireless communication; Channel quality estimation is an extra processing stage to determine the degree of distortion and noise interference in the current communication environment. The estimated channel quality parameters are used as an excitation to scale the run-time complexity of the channel estimator.

# 3. Analysis

In this chapter, the OFDM baseband model for CMMB standard is formulated. The properties of wireless fading channel are analyzed. A solid understanding of those properties can further help to design channel estimation algorithms and channel quality estimation algorithms.

## 3.1. System Model

The OFDM baseband model for CMMB is similar to many OFDM systems such as DVB-T [7], and 802.16b [8], which are all modulated with null-subcarriers on two band sides. Figure 3.1 shows the OFDM baseband model, which is a general model also used in [9]. The model and can be divided into three parts: the transmitter, the fading channel with additive noise, and the receiver.

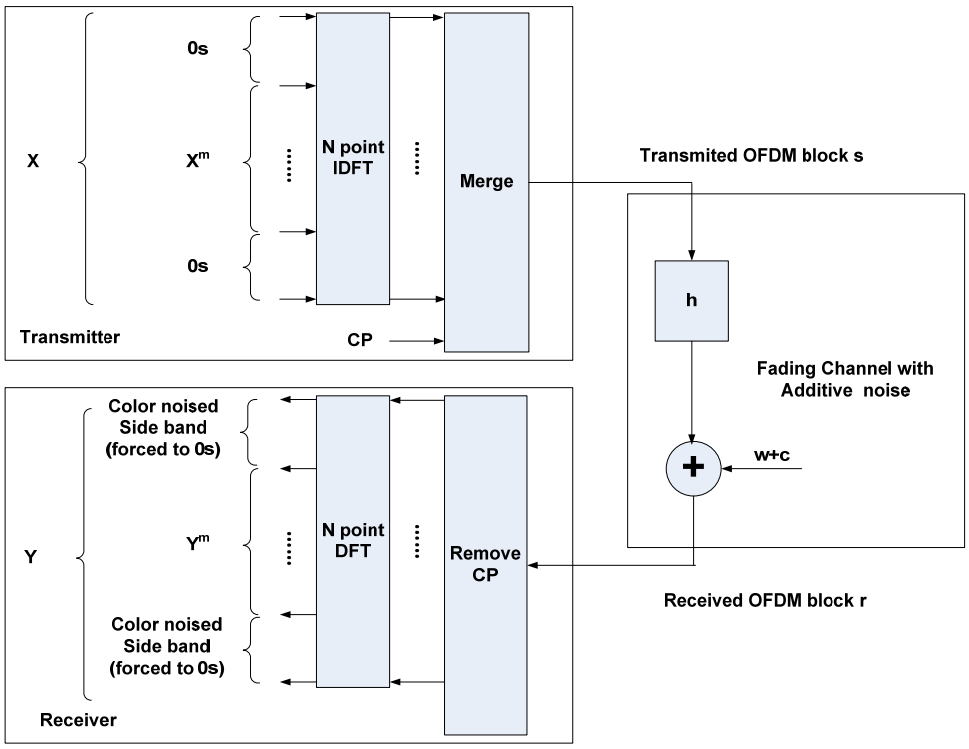


**Figure 3.1 OFDM baseband model for CMMB**

### 3.1.1. Transmitter

OFDM blocks are generated by a transmitter in Figure 3.1. $\mathbf{X}^{\mathrm{m}}$ is a vector with the length of $N_m$, consisting of $N_m$ complex-valued data symbols to be transmitted. $\mathbf{X}^{\mathrm{m}}$ is assigned into the usable-subcarrier band of the n$^{\mathrm{th}}$ OFDM block. And then, two groups of null-subcarriers, which are 0s, are padded into the edge bands of $\mathbf{X}^{\mathrm{m}}$ to form a N-point vector $\mathbf{X}$. For CMMB, $N$ and $N_m$ are even

and the number of null-subcarriers in each edge is $K = \frac{N - N_m}{2}$. Therefore, $\mathbf{X}$ can be expressed in the following form:

$$\mathbf{X} = [\mathbf{0_K}, \mathbf{X^m}, \mathbf{0_K}]^T \tag{3-1}$$

Where $(\bullet)^T$ is the matrix transpose operation.

$\mathbf{X}$ is further modulated by N-point IDFT. After modulation, we get the time-domain vector in the following form:

$$\mathbf{x} = IDFT(\mathbf{X}, N) = \mathbf{F'^H} \cdot \mathbf{X} \tag{3-2}$$

Where $(\bullet)^H$ is the matrix conjugate transpose operation, $\mathbf{F}'$ is a $N_m \times N$ "truncated" DFT matrix due to null-subcarrier modulation, where "truncated" means removing first $K$ rows and last $K$ rows out of a $N \times N$ DFT matrix. $\mathbf{F}'$ is defined as in the following form:

$$\mathbf{F}' = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & W_N^K & \cdots & W_N^{K(N-1)} \\ 1 & & \cdots & \\ 1 & W_N^{N-K-1} & \cdots & W_N^{(N-K-1)(N-1)} \end{bmatrix} \tag{3-3}$$

Where $W_N = e^{-j\frac{2\pi}{N}}$.

Then, cyclic prefix (CP) denoted as $\mathbf{x^G}$, which is a repetition part of $\mathbf{x}$ with the length of $G$, is merged together with the $\mathbf{x}$ so that one OFDM block is generated in the following form:

$$\mathbf{s} = [\mathbf{x^G}, \mathbf{x}]^T \tag{3-4}$$

### 3.1.2. Fading Channel with Additive Noise

For the channel part in Figure 3.1, the transmitted OFDM blocks are firstly faded by a time-variant multi-path channel. It can be modeled by correlating the time-domain OFDM block $\mathbf{x}$ with a channel impulse response (CIR) vector $\mathbf{h}$. $\mathbf{h}$ has the length of N, and is in the following form

$$\mathbf{h} = [h(0), h(1), \cdots, h(L-1), 0 \cdots 0]^T \tag{3-5}$$

Where $\mathbf{h}(l)$ is the $l^{th}$ path gain during the time interval of $n^{th}$ OFDM block. $L-1$ is the maximal index of the non-zero path gain in $\mathbf{h}$. $L$ is also called delay spread. We assume that $L < G$ and that within one OFDM block, $\mathbf{h}$ is constant. These are common assumptions in many OFDM systems and are one of the main reasons to use CP for inter-block protection [10].

After correlating with $\mathbf{h}$, the result is further added with a zero mean Gaussian white noise vector $\mathbf{w}$. In real communication systems, the signal vector are also interfered by a colored noise vector in

the null-subcarrier band, denoted as $\mathbf{c}$. That is why null-subcarriers are used for side band protection.

Based on above assumptions, the signals after passing through the fading channel with additive noise can be expressed in the following form:

$$\mathbf{r}(k) = \sum_{l=0}^{L-1} \mathbf{s}(G+k-l) \cdot \mathbf{h}(l) + \mathbf{w}(k) + \mathbf{c}(k) \qquad 0 \le k \le N-1 \tag{3-6}$$

### 3.1.3. Receiver

For the OFDM receiver in Figure 3.1, we assume that the synchronization is perfect. Then after removing CP, performing N-point DFT demodulation and forcing the null-subcarriers in the edge bands to be zeros, we can get the demodulated symbol vector $\mathbf{Y}$ in the following form:

$$\mathbf{Y} = [\mathbf{0_K}, \mathbf{Y^m}, \mathbf{0_K}]^T = \mathbf{H} \cdot \mathbf{X} + \mathbf{W} \tag{3-7}$$

In (3-7), $\mathbf{H}$ is a $N \times N$ diagonal matrix, whose diagonal vector is in the following form:

$$diag(\mathbf{H}) = [\mathbf{0_K}, \mathbf{H^m}, \mathbf{0_K}]^T \tag{3-8}$$

Where $\mathbf{H^m}$ is the frequency domain counterpart of $\mathbf{h}$ within the usable-subcarrier band and can be expressed in the following form:

$$\mathbf{H^m} = \mathbf{F'} \cdot \mathbf{h} \tag{3-9}$$

Also in (3-7), $\mathbf{W}$ is a band-limited white noise vector within the usable-subcarrier band, in the following form:

$$\mathbf{W} = [\mathbf{0_K}, \mathbf{W^m}, \mathbf{0_K}]^T \tag{3-10}$$

Where $\mathbf{W^m} = \mathbf{F'} \cdot \mathbf{w}$ (3-11)

Using (3-1), (3-7), (3-8) and (3-10), we can derive the relationship between transmitted symbol vector $\mathbf{X^m}$ and demodulated symbol vector $\mathbf{Y^m}$, in the following form:

$$\mathbf{Y^m}(k) = \mathbf{H^m}(k) \cdot \mathbf{X^m}(k) + \mathbf{W^m}(k), \qquad 0 \le k \le N_m - 1 \tag{3-12}$$

From (3-12), we can see that in order to recover the transmitted symbol vector $\mathbf{X^m}$ from the demodulated symbol vector $\mathbf{Y^m}$, it's necessary to have a estimate of $\mathbf{H^m}$, which is the channel frequency response in the usable-subcarrier band. This is exactly what a channel estimator aims for. For CMMB, in order to enable channel estimation, scattered pilots (SP), which are simply 1-valued, are inserted into specific usable-subcarrier locations based on a pre-defined pattern. For the subcarriers which transmit SPs, they can be expressed in the following form:

$$\begin{aligned}\mathbf{Y}^{\mathrm{m}}(\mathbf{P}(k)) &= \mathbf{H}^{\mathrm{m}}(\mathbf{P}(k))\mathbf{X}^{\mathrm{m}}(\mathbf{P}(k)) + \mathbf{W}^{\mathrm{m}}(\mathbf{P}(k)) \\ &= \mathbf{H}^{\mathrm{m}}(\mathbf{P}(k)) + \mathbf{W}^{\mathrm{m}}(\mathbf{P}(k)), \qquad 0 \le k \le N_p - 1\end{aligned} \tag{3-13}$$

Where $\mathbf{P}$ is the pattern vector consisting of the indexes all SPs, and $N_p$ is the number of SPs.

The goal of a channel estimator is to use the subset of $\mathbf{Y}^{\mathrm{m}}(p)\ \mathrm{with}\ p \in \mathbf{P}$ as the input, to generate the estimate of $\mathbf{H}^{\mathrm{m}}$ as the output. Denote the estimated channel frequency response as $\widehat{\mathbf{H}^{\mathrm{m}}}$, one widely used metric to evaluate the performance of a channel estimator is the mean square error (MSE), which is specified in the following form:

$$\begin{aligned}MSE &= E\ (\ \frac{1}{N_m}\sum_{k=0}^{N_{\mathrm{m}}-1}\left|\mathbf{H}^{\mathrm{m}}(k) - \widehat{\mathbf{H}^{\mathrm{m}}}(k)\right|^2\ ) \\ &= \frac{1}{M}\sum_{n=1}^{M}(\ \frac{1}{N_m}\sum_{k=0}^{N_{\mathrm{m}}-1}\left|\mathbf{H}^{\mathrm{m}}(k) - \widehat{\mathbf{H}^{\mathrm{m}}}(k)\right|^2\ )\end{aligned} \tag{3-14}$$

Where M is number of evaluated OFDM blocks in the testing assemble.

## 3.2. Channel Model

The system model in section 3.1 needs a concrete specification of $\mathbf{h}$, which can emulate a real multi-channel channel and can be used as the reference to evaluate the performance of the derived channel estimator. Typical urban 6-tap channel (TU6) is defined by COST-207[11] based on large amount of experimental measurements on terrestrial propagation in an urban area, and is used by CMMB standard as a reference channel model. Therefore, TU6 is used in our simulation and evaluation work. The TU6 channel profile in continuous time domain is specified in Table 3.1, we can see that in the TU6 channel model the longest path delay is $\tau_{\max} = 5us$.

**Table 3.1 TU6 channel profile**

| Tap number | Delay (us) | Power (dB) | Fading model |
|---|---|---|---|
| 1 | 0.0 | -3 | Rayleigh |
| 2 | 0.2 | 0 | Rayleigh |
| 3 | 0.5 | -2 | Rayleigh |
| 4 | 1.6 | -6 | Rayleigh |
| 5 | 2.3 | -8 | Rayleigh |
| 6 | 5.0 | -10 | Rayleigh |

In order to adopt the TU6 channel profile specified in Table 3.1 into CMMB baseband model, the continuous tap delays should be translated into discrete indexes. Denote the sampling rate of CMMB as $fs$, then the tap delay in the discrete baseband model can be transformed as follows:

$$l_i = \left\lceil \tau_i \cdot fs \right\rfloor \tag{3-15}$$

Where $\lceil \cdot \rceil$ is ceiling operation, $\tau_i$ is time delay of $i^{th}$ tap in continuous time and $l_i$ is the corresponding delay index in the discrete baseband model. The power delay profile (PDP) of TU6 with discrete indexes is shown in Figure 3.2, reflecting the average amplitude square in each discrete delay tap.

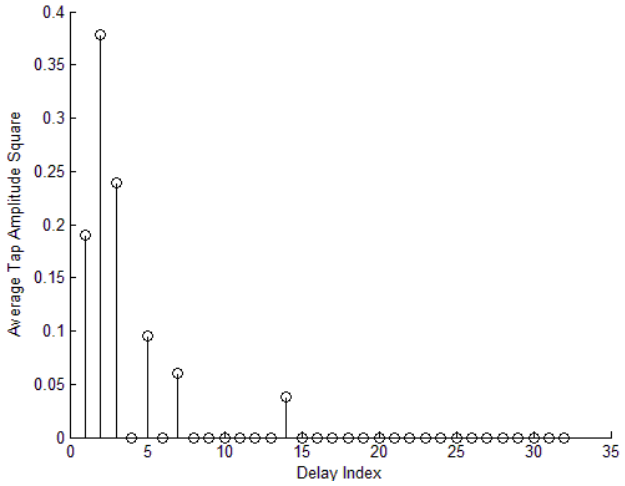


**Figure 3.2 Power delay profile of TU6 with discrete indexes**

## 3.3. Channel Properties

As analyzed in section 3.1, the goal of a channel estimator is to estimate $\mathbf{H}^{\mathrm{m}}$. Since $\mathbf{H}^{\mathrm{m}}$ is the frequency domain counterpart of $\mathbf{h}$ within the usable-subcarrier band following (3-9), a further analysis on the properties of $\mathbf{h}$ and its impact on $\mathbf{H}^{\mathrm{m}}$ can help the design of channel estimation algorithms and channel quality estimation algorithms.

### 3.3.1. Short-delay Limitation

Consider the CIR which is denoted as $\mathbf{h}$ in (3-5), it'short-delay limited. It means that the energy of a CIR is located within the short delay paths (shorter than $L$ ) in the time domain. That is fair because normally a longer path delay is, the weaker the path gain will be. Consider TU6 model, the delay spread in 1K mode CMMB ( $fs = 2.5MHZ$ ) can be calculated as:

$$L = \left\lceil \tau_{\max} \cdot fs \right\rceil + 1 = 14 \qquad (3\text{-}16)$$

We can see that indeed $L \ll N$ (in 1k mode CMMB, $N = 1024$), which verifies this property.

### 3.3.2. Energy Leakage

As analyzed in subsection 3.3.1, the CIR has short-delay limitation property. However, in real systems as the OFDM inner-receiver, the reconstructed CIR has the property that the energy of channel taps is actually leaked to longer delay paths. There are two reasons which cause the energy leakage: Firstly, when the product of a tap delay and the sampling frequency is not an integer, the energy of such tap is actually leaking into long delay paths in the baseband model. Such

phenomenon has been analytically analyzed by [12]; Secondly, for OFDM modulation schemes with null-subcarriers, such as CMMB, energy leakage happens due to the null-subcarrier modulation: Suppose the noise is zero and the transmitted symbol vector $\mathbf{X}^{\mathrm{m}}$ is an all-one vector, then (3-12) can be rewritten as the follows:

$$\mathbf{Y}^{\mathrm{m}}(k) = \mathbf{H}^{\mathrm{m}}(k), \qquad 0 \le k \le N_{m} - 1 \tag{3-17}$$

(3-17) means $\mathbf{H}^{\mathrm{m}}$ is perfectly reconstructed in the receiver. If we further transform the obtained $\mathbf{H}^{\mathrm{m}}$ back into the time domain by N-point IDFT, using (3-9) we can derive that:

$$\begin{aligned} \mathbf{h}^{\mathrm{m}} &= IDFT(\mathbf{H}^{\mathrm{m}}, N) \\ &= \mathbf{F'}^{\mathbf{H}}(\mathbf{F'} \cdot \mathbf{h}) = (\mathbf{F'}^{\mathbf{H}}\mathbf{F'}) \cdot \mathbf{h} \end{aligned} \tag{3-18}$$

Note that $\mathbf{F'}$ is a truncated version of the $N \times N$ DFT matrix due to null-subcarrier modulation, and $\mathbf{F'}^{H}$ is conjugated transpose of $\mathbf{F'}$. There is an important feature that $(\mathbf{F'}^{\mathbf{H}}\mathbf{F'})$ is not a diagonal matrix [9], which disperses the compacted energy of $\mathbf{h}$ into longer delay elements. Therefore, the long delay elements in $\mathbf{h}^{\mathrm{m}}$ are no longer zeros.

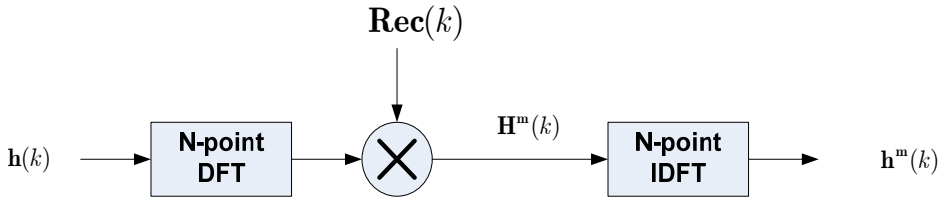


**Figure 3.3 Illustration of energy leakage due to null-subcarrier modulation**

The energy leakage phenomenon due to null-subcarrier modulation and expression (3-18) can be further illustrated by Figure 3.3, where $\mathbf{Rec}(k)$ is a rectangle windowing function defined as:

$$\mathbf{Rec}(k) = \begin{cases} 1, & K \le k \le N - K - 1 \\ 0, & 0 \le k \le K - 1 \quad or \quad N - K \le k \le N - 1 \end{cases} \tag{3-19}$$

In Figure 3.3, we can see that $\mathbf{H}^{\mathrm{m}}$ is the windowed version of the real channel frequency response, resulting that $\mathbf{h}^{\mathrm{m}}$ is the convolution result of $\mathbf{h}$ with a $sinc$ function [13]. Therefore, the long delay elements in $\mathbf{h}^{\mathrm{m}}$ are no longer zeros, but the energy leaked values from the short delay elements of $\mathbf{h}$. Figure 3.4 and Figure 3.5 show an example of the tap amplitudes of $\mathbf{h}^{\mathrm{m}}$ and $\mathbf{h}$.

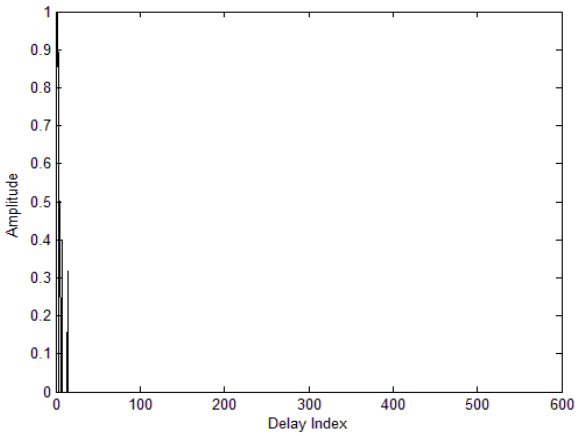


**Figure 3.4 Original CIR: $\mathbf{h}$**

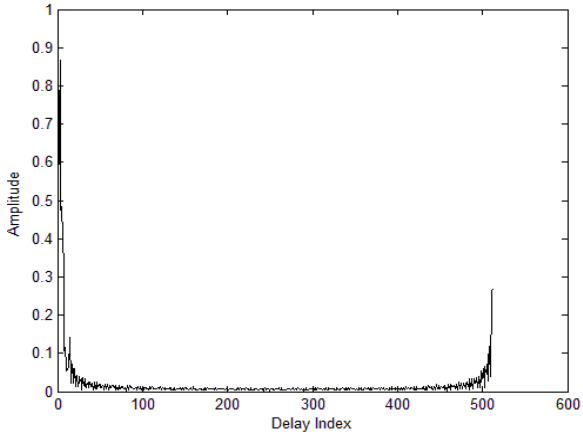


**Figure 3.5 Reconstructed CIR: $\mathbf{h}^{\mathrm{m}}$**

## 3.4. Chapter Conclusion

This chapter explains the OFDM baseband model for CMMB, while analytically formulates the channel estimation problem and the performance metric. TU6 channel model is inserted into the baseband model and can be used as a reference to evaluate the performance of derived algorithms. Two properties of a multi-path channel are analyzed: short-delay limitation and energy leakage. These two properties do not contradict to each other because short-delay limitation is a nature property of a CIR during radio propagations, while energy leakage is an important property of the reconstructed CIR in the baseband inner-receiver. Understanding these two properties can benefit the design of channel estimation algorithms as well as channel quality estimation algorithms.

# 4. Design

In this chapter, the architecture of the run-time complex adaptive channel estimator is detailed. Then, a set of channel estimation algorithms using SPs within CMMB data bodies are designed. Next, channel quality estimation algorithms using CMMB preamble are designed. Finally, a power manager which provides thresholds to switch channel estimation algorithms is designed.

## 4.1. Architecture

There are two alternative architectures for run-time complexity adaptive channel estimator: the feedforward architecture and the feedback architecture.

The feedforward architecture is directly derived from the motivation diagram in Figure 2.1, and is detailed into the diagram in Figure 4.1.

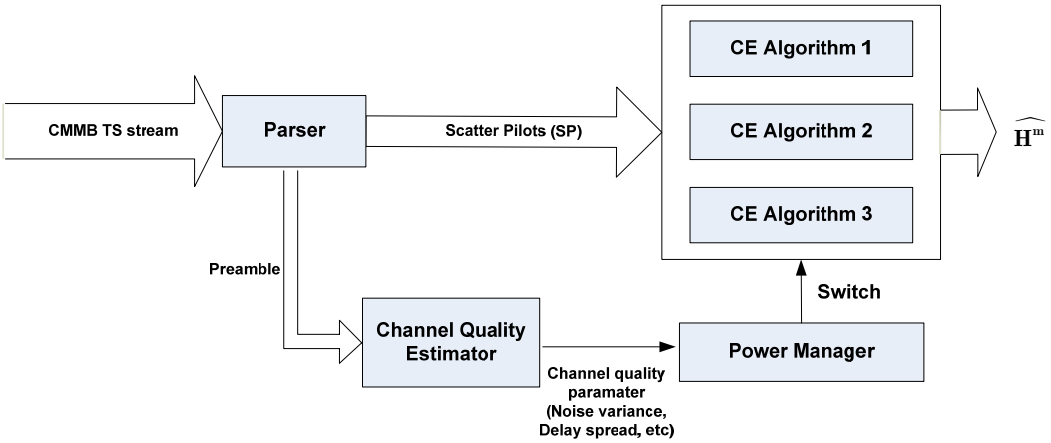


**Figure 4.1 Run-time complexity adaptive channel estimator in the feedforward architecture**

In the architecture of Figure 4.1, the channel quality is assumed to be constant within every OFDM time slot (TS), whose time duration is 25ms. This assumption is fair because the change of channel quality is normally caused by the change of environment temperature, or due to physical movement. Within 25ms, those changes can be seen subtitle.

Based on the assumption, in Figure 4.1, the preamble of every CMMB TS is extracted by a parser for channel quality estimation (CQE). The parser also extracts the SPs within 53 subsequent OFDM data blocks in the current TS, for channel estimation (CE) in every OFDM data block. A set of CE algorithms with different operation loads and different estimation accuracies are available to be selected, and a power manager reads the estimated channel quality parameters, sorts them by pre-defined thresholds and chooses a CE algorithm for channel estimation for the following 53 OFDM blocks in the current TS.

As an alternative, the run-time complexity adaptive channel estimator can also be designed in the feedback architecture as shown in Figure 4.2.

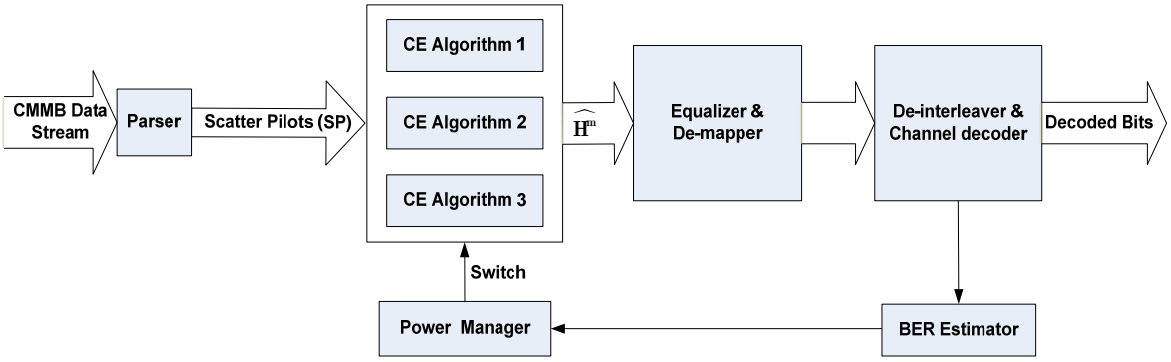


**Figure 4.2 Run-time complexity adaptive channel estimator in the feedback architecture**

The architecture in Figure 4.2 directly uses the estimated bit error rate (BER) to indicate the channel quality, which seems a more direct indication of the reception quality. However, there are following drawbacks of this architecture:

1) An accurate BER estimator is difficult to realize. That's because the un-coded data in the transmitter is unknown in the receiver. Therefore, in order to obtain an accurate BER estimation, the decoded bits needs to be re-interleaved and re-encoded for comparison. CMMB uses LDPC as its channel code, whose encoding process is far more complex than the decoding process, so that it's difficult to implement a LDPC encoder in VDSP.

2) Even an accurate BER estimator can be realized by specific hardware, the run-time requirement is difficult to fulfil. That is because the encoded bits are interleaved, and because a large data set is required to estimate a very small BER (For CMMB, $BER \leq 3 \times 10^{-6}$). Thus the estimated BER only indicates an average channel quality during a past time interval whose length is determined by the interleaving depth and the size of BER estimation data set, not an immediate channel quality for the current TS.

3) BER can not reflect the details of the channel quality, so that the selection of CE algorithms may be not optimal. That is because different CE algorithms might have different sensitivities to different channel quality parameters. The feedforward architecture plays a pre-analysis of channel quality parameters which can help to choose the most appropriate CE algorithm. However, in the feedback architecture, it's impossible.

Based on above analysis, the feedforward architecture in Figure 4.1 is chosen in the design.

## 4.2. Channel Estimation Algorithms

Channel Estimation (CE) for CMMB is done by interpolating scattered pilots (SP) within OFDM data bodies. In the following subsections, grouping of SPs is firstly introduced. Then, by using grouped SPs in a training cluster, three channel estimation algorithms for CMMB are presented, which are Linaer estimation, Wiener estimation and Transform domain LS estimation. Finally, the performance scalability of CE algorithms in the metric of MSE, and the complexity scalability in the metric of number of arithmetic operations (complex additions and complex multiplications) are both analyzed.

### 4.2.1. SP grouping

In the CMMB transmitter, SPs are inserted into usable-subcarrier band according to a pre-defined pattern. The SP pattern is already shown in Figure 1.3, where the SPs are arranged in an interlaced fashion, and the distant between two neighboring SPs within one OFDM data block is 8 subcarriers.

The SPs need to be extracted and grouped into a training cluster to estimate the channel frequency response in the usable-subcarrier band. The indexes of all SPs in the training cluster are stored in the pattern vector $\mathbf{P}$ . The grouping of the SPs is defined in two modes: the static mode where the channel is weakly time-varying, and the mobile mode where channel is strongly time-varying. The two grouping modes are shown in Figure 4.3: In the static mode, the grouped SPs in the training cluster are the combination of SPs in current OFDM data block and the SPs in previous OFDM data block; while in the mobile mode, the grouped SPs are only collected from the current OFDM data block.

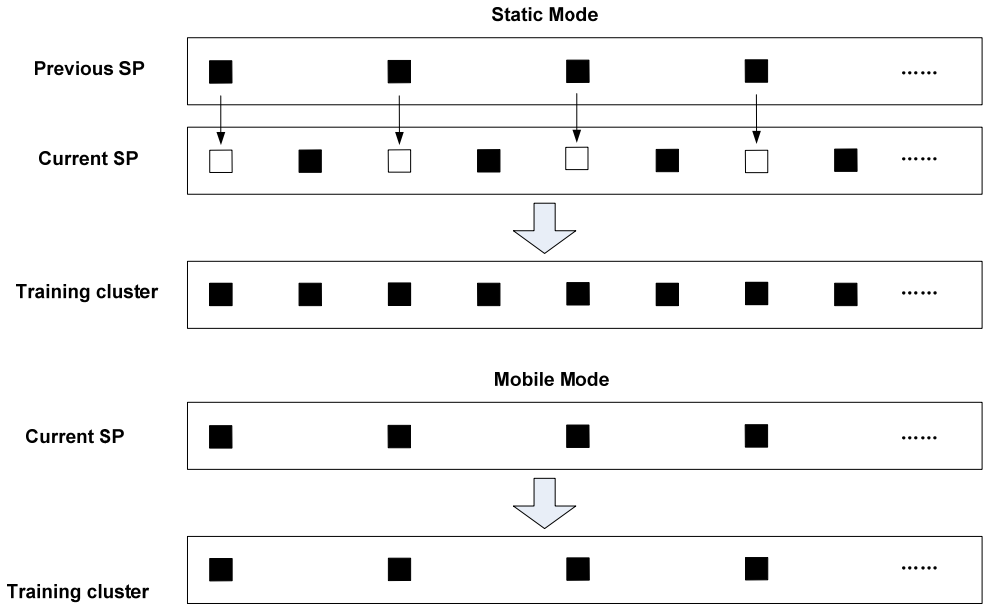


**Figure 4.3 SP grouping in the static mode and the mobile mode**

The identification of the static mode and the mobile mode is determined by the Doppler spread, and the methods to estimate Doppler spread and the methods for mode switching are presented in many literatures [14][15][16], and are not treated in this thesis. Also note that, in the static mode, the current OFDM data block together with SPs of the previous OFDM data block need to be buffered; in the mobile mode, only the current OFDM data block needs to be buffered.

### 4.2.2. Linear Estimation

Linear estimation is the simplest way to estimate the channel frequency response using received SPs. This is done by making weighted summation of two nearest SPs in the training cluster. The weights are inversely proportional to the pilot distance. Linear estimation under the static and the mobile mode are shown separately in Figure 4.4.

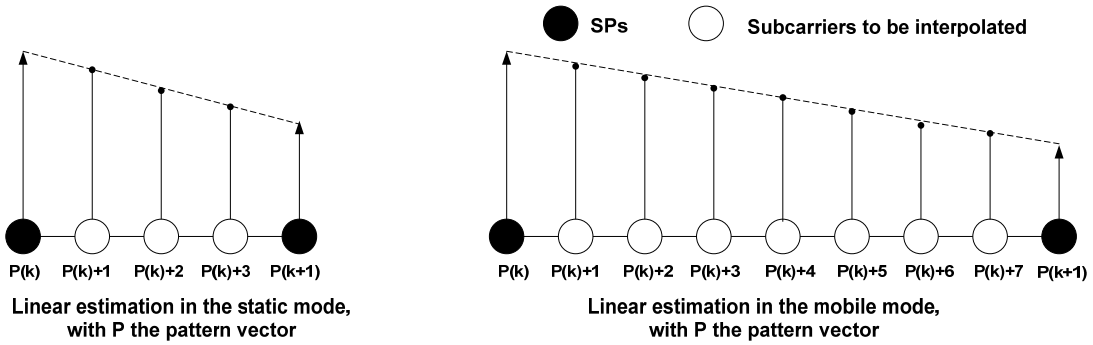


**Figure 4.4 Linear estimation in the static and the mobile mode**

### 4.2.3. Wiener Estimation

The ideal Full-rank Wiener estimator in the matrix form is proposed in [17] and is represented in subsection 4.2.3.1. It's claimed to be optimal because it aims to minimize the mean square error (MSE) of the estimated channel frequency response. In subsection 4.2.3.2, the problems to realize the ideal full-rank wiener estimator in VDSP are discussed, and a low-rank mismatched Wiener estimator in FIR filtering form, which can be realized in VDSP, is further derived.

#### 4.2.3.1. Ideal Full-rank Wiener Estimation in Matrix Form

Expression (3-13) can be rewritten into the matrix form in the following form:

$$\mathbf{Y}_{\mathbf{p}}^{\mathbf{m}} = \mathbf{H}_{\mathbf{p}}^{\mathbf{m}} + \mathbf{W}_{\mathbf{p}}^{\mathbf{m}} \tag{4-1}$$

Where $\mathbf{Y}_{\mathbf{p}}^{\mathbf{m}}$ is a vector filled with grouped SPs and $\mathbf{H}_{\mathbf{p}}^{\mathbf{m}}$ is the channel frequency response in the SP locations within the useable-subcarrier band, with $\mathbf{Y}_{\mathbf{p}}^{\mathbf{m}}(k) = \mathbf{Y}^{\mathbf{m}}(\mathbf{P}(k))$ and $\mathbf{H}_{\mathbf{p}}^{\mathbf{m}}(k) = \mathbf{H}^{\mathbf{m}}(\mathbf{P}(k))$

For the ideal full-rank Weiner estimator, the estimated channel response is expressed as a linear combination of the elements in $\mathbf{Y}_{\mathbf{p}}^{\mathbf{m}}$, in the following form:

$$\widehat{\mathbf{H}^{\mathbf{m}}} = \mathbf{C} \cdot \mathbf{Y}_{\mathbf{p}}^{\mathbf{m}} \tag{4-2}$$

Where $\mathbf{C}$ is a $N_m \times N_p$ coefficient matrix, $N_p$ is the number of grouped SPs in the training cluster and $N_m$ is the number of usable-subcarriers.

The goal of the ideal wiener estimator is find a optimal coefficient matrix $\mathbf{C}$ which minimizes the mean square error between $\widehat{\mathbf{H}^{\mathbf{m}}}$ and $\mathbf{H}^{\mathbf{m}}$, and it can be expressed in the following form:

$$\begin{aligned}\mathbf{C}_{\mathbf{LMMSE}} &= \arg\min_{\mathbf{C}} \left\{ E\left(\frac{1}{N_m} \sum_{k=0}^{N_m-1} \left| \mathbf{H}^{\mathbf{m}}(k) - \widehat{\mathbf{H}^{\mathbf{m}}}(k) \right|^2 \right) \right\} \\ &= \arg\min_{\mathbf{C}} \left\{ E[(\mathbf{H}^{\mathbf{m}} - \widehat{\mathbf{H}^{\mathbf{m}}}) \cdot (\mathbf{H}^{\mathbf{m}} - \widehat{\mathbf{H}^{\mathbf{m}}})^H] \right\}\end{aligned} \tag{4-3}$$

Using (4-1), (4-2), (4-3), and using the assumption that $\mathbf{H}^{\mathrm{m}}$ and $\mathbf{W}^{\mathrm{m}}$ are uncorrelated, the Wiener coefficient matrix $\mathbf{C}_{\mathbf{LMMSE}}$ is derived the in the following form [17]:

$$\mathbf{C}_{\mathbf{LMMSE}} = \mathbf{R}_{\mathbf{H^m H_p^m}}(\mathbf{R}_{\mathbf{H_p^m H_p^m}} + \sigma_w^2\mathbf{I})^{-1} \tag{4-4}$$

Where $\mathbf{R}_{\mathbf{H^m H_p^m}}$ is the cross-correlation matrix between $\mathbf{H}^{\mathrm{m}}$ and $\mathbf{H}_{\mathbf{p}}^{\mathbf{m}}$, with the size $N_m \times N_p$; $\mathbf{R}_{\mathbf{H_p^m H_p^m}}$ is the autocorrelation matrix of $\mathbf{H}_{\mathbf{p}}^{\mathbf{m}}$, with the size $N_p \times N_p$; $\sigma_w^2$ is the noise variance; $\mathbf{I}$ is a unit matrix, with the size $N_p \times N_p$.

The elements in $\mathbf{R}_{\mathbf{H^m H_p^m}}$ and $\mathbf{R}_{\mathbf{H_p^m H_p^m}}$ can be further expressed in the following from:

$$\begin{aligned}\mathbf{R}_{\mathbf{H^m H_p^m}}(k,j) &= E\left\{\mathbf{H}^{\mathrm{m}}(k)\mathbf{H}_{\mathbf{p}}^{\mathbf{m}}(j)^*\right\} \\ &= E\left\{\mathbf{H}^{\mathrm{m}}(k)\mathbf{H}^{\mathrm{m}}(\mathbf{P}(j))^*\right\} = \mathbf{R}_{\mathbf{H^m H^m}}(k,\mathbf{P}(j))\end{aligned} \tag{4-5}$$

$$\begin{aligned}\mathbf{R}_{\mathbf{H_p^m H_p^m}}(i,j) &= E\left\{\mathbf{H}_{\mathbf{p}}^{\mathbf{m}}(i)\mathbf{H}_{\mathbf{p}}^{\mathbf{m}}(j)^*\right\} \\ &= E\left\{\mathbf{H}^{\mathrm{m}}(\mathbf{P}(i))\mathbf{H}^{\mathrm{m}}(\mathbf{P}(j))^*\right\} = \mathbf{R}_{\mathbf{H^m H^m}}(\mathbf{P}(i),\mathbf{P}(j))\end{aligned} \tag{4-6}$$

Where $(\cdot)^*$ is the scalar conjugate operation and $\mathbf{R}_{\mathbf{H^m H^m}}$ is the autocorrelation matrix of $\mathbf{H}^{\mathrm{m}}$ with the size $N_m \times N_m$. $\mathbf{R}_{\mathbf{H^m H^m}}$ can be computed in the following form:

$$\mathbf{R}_{\mathbf{H^m H^m}} = E\left\{\mathbf{H}^{\mathrm{m}}(\mathbf{H}^{\mathrm{m}})^{\mathrm{H}}\right\} = E\left\{\mathbf{F'h}\cdot(\mathbf{F'h})^{\mathrm{H}}\right\} = \mathbf{F'}\cdot E\left\{\mathbf{hh}^{\mathrm{H}}\right\}\cdot\mathbf{F'}^{\mathrm{H}} \tag{4-7}$$

#### 4.2.3.2. Mismatched Low-rank Wiener Estimation in FIR Filtering Form

Although the ideal Wiener estimator in the matrix form is optimal in the sense of minimizing the mean square error (MSE), it's infeasible to be implemented in VDSP. The problems are in the following:

1) In order to compute the optimal Wiener coefficient matrix $\mathbf{C}_{\mathbf{LMMSE}}$ using (4-4), channel statistics information ($\mathbf{R}_{\mathbf{H^m H^m}}$ and $\sigma_w^2$) is needed, which is usually unknown in a real system. Furthermore, the computation of $\mathbf{C}_{\mathbf{LMMSE}}$ in (4-4) involves a matrix inversion operation with a size $N_p \times N_p$ (for CMMB in 4k mode, $N_p = 624$), which is too expensive to be done in VDSP in the run-time.
2) In order to compute the estimate of channel frequency response $\widehat{\mathbf{H}^{\mathrm{m}}}$ using expression (4-2), a matrix with $N_m \times N_p$ coefficients must be buffered. And a matrix multiplication with

the size of $N_m \times N_p$ must be played in the run-time. Both of them are also too expensive to be done in VDSP.

In order to solve problem 1), it's possible to derive a robust estimator with a mismatched channel model [18]: That is to replace $\sigma_w^2$ by a constant average noise variance, and to replace the real channel vector $\mathbf{h}$ in (4-7) by a constant mismatched channel vector $\mathbf{h}^{mis}$, which has uniform gain on each channel taps within a pre-defined maximal delay spread. $\mathbf{h}^{mis}$ can be expressed in the following form:

$$\mathbf{h}^{mis}(i) = \begin{cases} 1, & 0 \le i \le L_{\max} \\ 0 & L_{\max} < i \le N \end{cases} \qquad (4\text{-}8)$$

Where $L_{\max}$ is the assumed maximal delay spread, and can be the length of CP: $L_{\max} = G$.

Therefore, using (4-7), a mismatched $\mathbf{R}_{\mathbf{H^m H^m}}$ can be rewritten in the following form:

$$\mathbf{R}^{mis}_{\mathbf{H^m H^m}} = \mathbf{F'} \cdot E\left\{\mathbf{h} \cdot \mathbf{h}^{\mathbf{H}}\right\} \cdot \mathbf{F'}^{\mathbf{H}} = \mathbf{F'} \cdot [\mathbf{h}^{mis}(\mathbf{h}^{mis})^{\mathbf{H}}] \cdot \mathbf{F'}^{\mathbf{H}} \qquad (4\text{-}9)$$

Replacing $\mathbf{R}_{\mathbf{H^m H^m}}$ in (4-5) and (4-6) by $\mathbf{R}^{mis}_{\mathbf{H^m H^m}}$ in (4-9), a sub-optimal wiener coefficient matrix can be offline-computed and pre-stored in the memory.

In order to solve problem 2), the ideal is to reduce the rank of the estimator. That is to apply channel estimation on every individual subcarriers in the usable-subcarrier band, so that each subcarrier of $\mathbf{H^m}$ is estimated by only a local training cluster containing only SPs in its neighbourhood. Denote $\mathbf{P'}$ as the local pattern vector corresponding to the $k^{th}$ subcarrier $\mathbf{H^m}(k)$. Denote the size of $\mathbf{P'}$ as $N_l$, then $\mathbf{P'}$ stores the indexes of $N_l$ nearest neighbouring SPs in the neighbourhood of $\mathbf{H^m}(k)$. The elements in $\mathbf{P'}$ can be further expressed in the following form:

$$\mathbf{P'}(i) = k + \mathbf{S}_k(i) \qquad 0 \le i \le N_l - 1 \qquad (4\text{-}10)$$

Where $\mathbf{S}_k$ consists of the relative offset between $k$ and $\mathbf{P'}(i)$.

Now the estimate of $\mathbf{H^m}(k)$ can be rewritten in the following form:

$$\widehat{\mathbf{H^m}}(k) = \mathbf{c} \cdot \mathbf{Y}^{\mathbf{m}}_{\mathbf{p'}} \qquad (4\text{-}11)$$

Where $\mathbf{Y}^{\mathbf{m}}_{\mathbf{p'}}$ is the local training vector corresponding to $\mathbf{H^m}(k)$, with the size of $N_l \times 1$, filled with received SPs whose indexes are stored in $\mathbf{P'}$. And $\mathbf{c}$ is a coefficient matrix with the size of $1 \times N_l$.

The low-rank Wiener estimator aims to find an optimal $\mathbf{c}$ which minimizes the MSE on every individual subcarrier (MSE between $\widehat{\mathbf{H}^{\mathbf{m}}}(k)$ and $\mathbf{H}^{\mathbf{m}}(k)$), and results in the following form:

$$\begin{aligned}\mathbf{c}_{\mathbf{LMMSE}} &= \arg\min_{\mathbf{C}}\left\{E(\left|\mathbf{H}^{\mathbf{m}}(k)-\widehat{\mathbf{H}^{\mathbf{m}}}(k)\right|^2)\right\} \\ &= \arg\min_{\mathbf{C}}\left\{E[(\mathbf{H}^{\mathbf{m}}(k)-\widehat{\mathbf{H}^{\mathbf{m}}}(k))\cdot(\mathbf{H}^{\mathbf{m}}(k)-\widehat{\mathbf{H}^{\mathbf{m}}}(k))^*]\right\}\end{aligned} \tag{4-12}$$

Similarly like (4-4), $\mathbf{c}_{\mathbf{LMMSE}}$ can be derived in the following form:

$$\mathbf{c}_{\mathbf{LMMSE}} = \mathbf{R}_{\mathbf{H}_k^{\mathbf{m}}\mathbf{H}_{\mathbf{p}'}^{\mathbf{m}}}(\mathbf{R}_{\mathbf{H}_{\mathbf{p}'}^{\mathbf{m}}\mathbf{H}_{\mathbf{p}'}^{\mathbf{m}}}+\sigma_w^2\mathbf{I})^{-1} \tag{4-13}$$

Where $\mathbf{R}_{\mathbf{H}_k^{\mathbf{m}}\mathbf{H}_{\mathbf{p}'}^{\mathbf{m}}}$ is the cross-correlation matrix between $\mathbf{H}^{\mathbf{m}}(k)$ and $\mathbf{H}_{\mathbf{p}'}^{\mathbf{m}}$, with the size $1\times N_l$; $\mathbf{R}_{\mathbf{H}_{\mathbf{p}'}^{\mathbf{m}}\mathbf{H}_{\mathbf{p}'}^{\mathbf{m}}}$ is the autocorrelation matrix of $\mathbf{H}_{\mathbf{p}'}^{\mathbf{m}}$, with the size $N_l\times N_l$; $\sigma_w^2$ is the noise variance; $\mathbf{I}$ is a unit matrix, with the size $N_l\times N_l$.

Using (4-10), the elements in $\mathbf{R}_{\mathbf{H}_k^{\mathbf{m}}\mathbf{H}_{\mathbf{p}'}^{\mathbf{m}}}$ and $\mathbf{R}_{\mathbf{H}_{\mathbf{p}'}^{\mathbf{m}}\mathbf{H}_{\mathbf{p}'}^{\mathbf{m}}}$ can be further expressed in the following from:

$$\begin{aligned}\mathbf{R}_{\mathbf{H}_k^{\mathbf{m}}\mathbf{H}_{\mathbf{p}'}^{\mathbf{m}}}(j) &= E\left\{\mathbf{H}^{\mathbf{m}}(k)\mathbf{H}_{\mathbf{p}'}^{\mathbf{m}}(j)^*\right\} \\ &= E\left\{\mathbf{H}^{\mathbf{m}}(k)\mathbf{H}^{\mathbf{m}}(\mathbf{P}'(j))^*\right\} = \mathbf{R}_{\mathbf{H}^{\mathbf{m}}\mathbf{H}^{\mathbf{m}}}(k,\mathbf{P}'(j)) \\ &= \mathbf{R}_{\mathbf{H}^{\mathbf{m}}\mathbf{H}^{\mathbf{m}}}(k,k+\mathbf{S}_k(j)) = \mathbf{R}_{\mathbf{H}^{\mathbf{m}}\mathbf{H}^{\mathbf{m}}}(0,\mathbf{S}_k(j)) \\ &\approx \mathbf{R}^{mis}_{\mathbf{H}^{\mathbf{m}}\mathbf{H}^{\mathbf{m}}}(0,\mathbf{S}_k(j))\end{aligned} \tag{4-14}$$

$$\begin{aligned}\mathbf{R}_{\mathbf{H}_{\mathbf{p}'}^{\mathbf{m}}\mathbf{H}_{\mathbf{p}'}^{\mathbf{m}}}(i,j) &= E\left\{\mathbf{H}_{\mathbf{p}'}^{\mathbf{m}}(i)\mathbf{H}_{\mathbf{p}'}^{\mathbf{m}}(j)^*\right\} \\ &= E\left\{\mathbf{H}^{\mathbf{m}}(\mathbf{P}'(i))\mathbf{H}^{\mathbf{m}}(\mathbf{P}'(j))^*\right\} = \mathbf{R}_{\mathbf{H}^{\mathbf{m}}\mathbf{H}^{\mathbf{m}}}(\mathbf{P}'(i),\mathbf{P}'(j)) \\ &= \mathbf{R}_{\mathbf{H}^{\mathbf{m}}\mathbf{H}^{\mathbf{m}}}(k+\mathbf{S}_k(i),k+\mathbf{S}_k(j)) = \mathbf{R}_{\mathbf{H}^{\mathbf{m}}\mathbf{H}^{\mathbf{m}}}(\mathbf{S}_k(i),\mathbf{S}_k(j)) \\ &\approx \mathbf{R}^{mis}_{\mathbf{H}^{\mathbf{m}}\mathbf{H}^{\mathbf{m}}}(\mathbf{S}_k(i),\mathbf{S}_k(j))\end{aligned} \tag{4-15}$$

Note that in (4-14) and (4-15), the ideal autocorrelation matrix $\mathbf{R}_{\mathbf{H}^{\mathbf{m}}\mathbf{H}^{\mathbf{m}}}$ is replaced by the mismatched one $\mathbf{R}^{mis}_{\mathbf{H}^{\mathbf{m}}\mathbf{H}^{\mathbf{m}}}$ from (4-9).

Now consider the CMMB pilot pattern in Figure 1.3, which is regular. We see that there are only 4 alternatives for $\mathbf{S}_k$ in the static mode, and only 8 alternatives for $\mathbf{S}_k$ in the mobile mode.

Therefore, $\mathbf{S}_k$ in the static mode can be expressed in the following form:

$$\mathbf{S}_k(i) = \begin{cases} -4 \cdot \left\lfloor \frac{N_l}{2} \right\rfloor + 4i + 4 & k \mod 4 = 0 \\ -4 \cdot \left\lfloor \frac{N_l}{2} \right\rfloor + 4i + 3 & k \mod 4 = 1 \\ -4 \cdot \left\lfloor \frac{N_l}{2} \right\rfloor + 4i + 2 & k \mod 4 = 2 \\ -4 \cdot \left\lfloor \frac{N_l}{2} \right\rfloor + 4i + 1 & k \mod 4 = 3 \end{cases} \qquad 0 \le i \le N_l - 1 \qquad \text{(4-16)}$$

And $\mathbf{S}_k$ in the mobile mode can be expressed in the following form:

$$\mathbf{S}_k(i) = \begin{cases} -8 \cdot \left\lfloor \frac{N_l}{2} \right\rfloor + 8i + 8 & k \mod 8 = 0 \\ -8 \cdot \left\lfloor \frac{N_l}{2} \right\rfloor + 8i + 7 & k \mod 8 = 1 \\ -8 \cdot \left\lfloor \frac{N_l}{2} \right\rfloor + 8i + 6 & k \mod 8 = 2 \\ -8 \cdot \left\lfloor \frac{N_l}{2} \right\rfloor + 8i + 5 & k \mod 8 = 3 \\ -8 \cdot \left\lfloor \frac{N_l}{2} \right\rfloor + 8i + 4 & k \mod 8 = 4 \\ -8 \cdot \left\lfloor \frac{N_l}{2} \right\rfloor + 8i + 3 & k \mod 8 = 5 \\ -8 \cdot \left\lfloor \frac{N_l}{2} \right\rfloor + 8i + 2 & k \mod 8 = 6 \\ -8 \cdot \left\lfloor \frac{N_l}{2} \right\rfloor + 8i + 1 & k \mod 8 = 7 \end{cases} \qquad 0 \le i \le N_l - 1 \qquad \text{(4-17)}$$

Figure 4.5 shows an example of $\mathbf{S}_k$ in the static mode when $N_l = 6$.

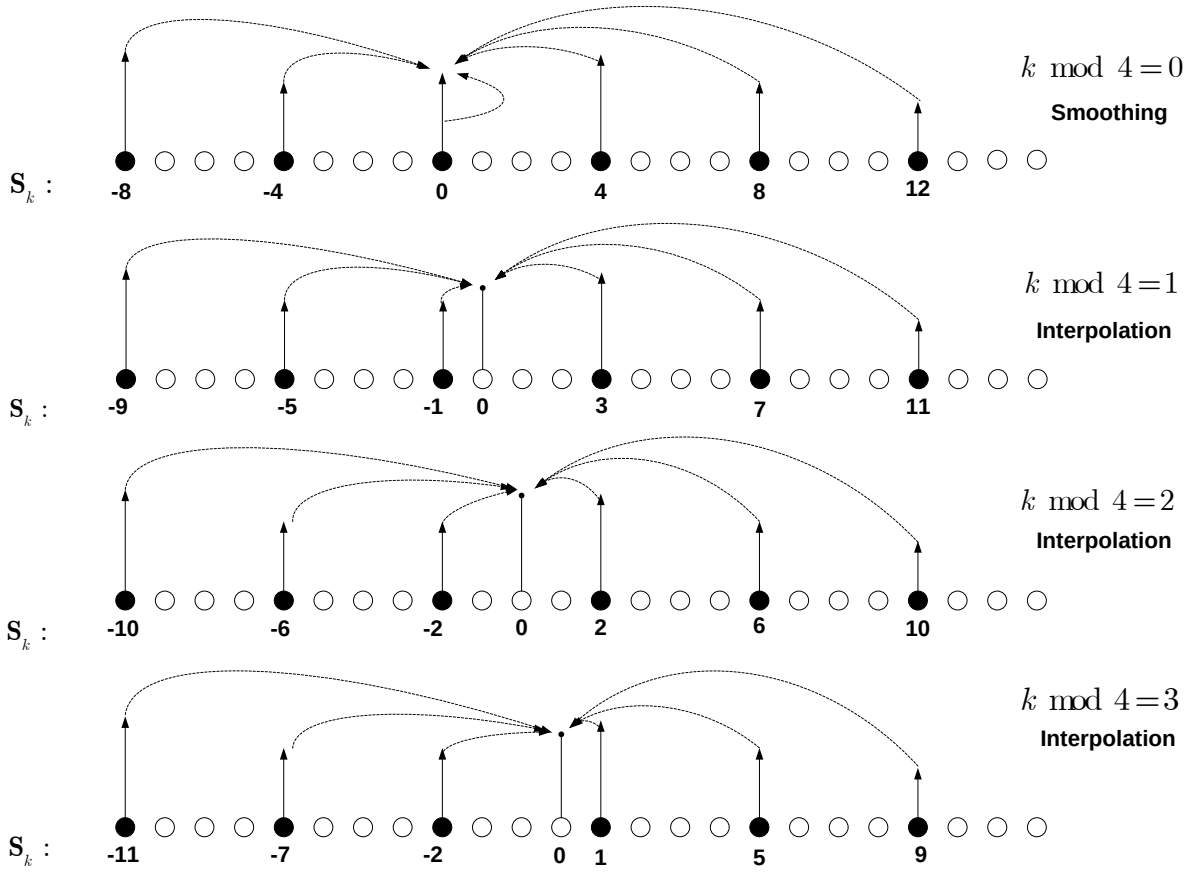


**Figure 4.5 Illustration of $\mathbf{S}_k$ in the static mode when $N_l = 6$**

Using SP pattern of CMMB and using (4-13), (4-14) and (4-15), we see that there are only 4 alternatives for $\mathbf{c}_{\mathbf{LMMSE}}$ in the static mode and only 8 alternatives for $\mathbf{c}_{\mathbf{LMMSE}}$. Two benefits are obtained: first, the coefficients to be buffered are reduced from $N_m \times N_p$ to $4 \times N_l$ in the static mode and $8 \times N_l$ in the mobile mode. Second, the matrix multiplication in (4-11) is transformed into 4 FIR filters in the static mode and 8 FIR filters in the mobile mode.

The implementation diagram of the derived mismatched low-rank Wiener estimator in the FIR filtering form is shown in Figure 4.6.

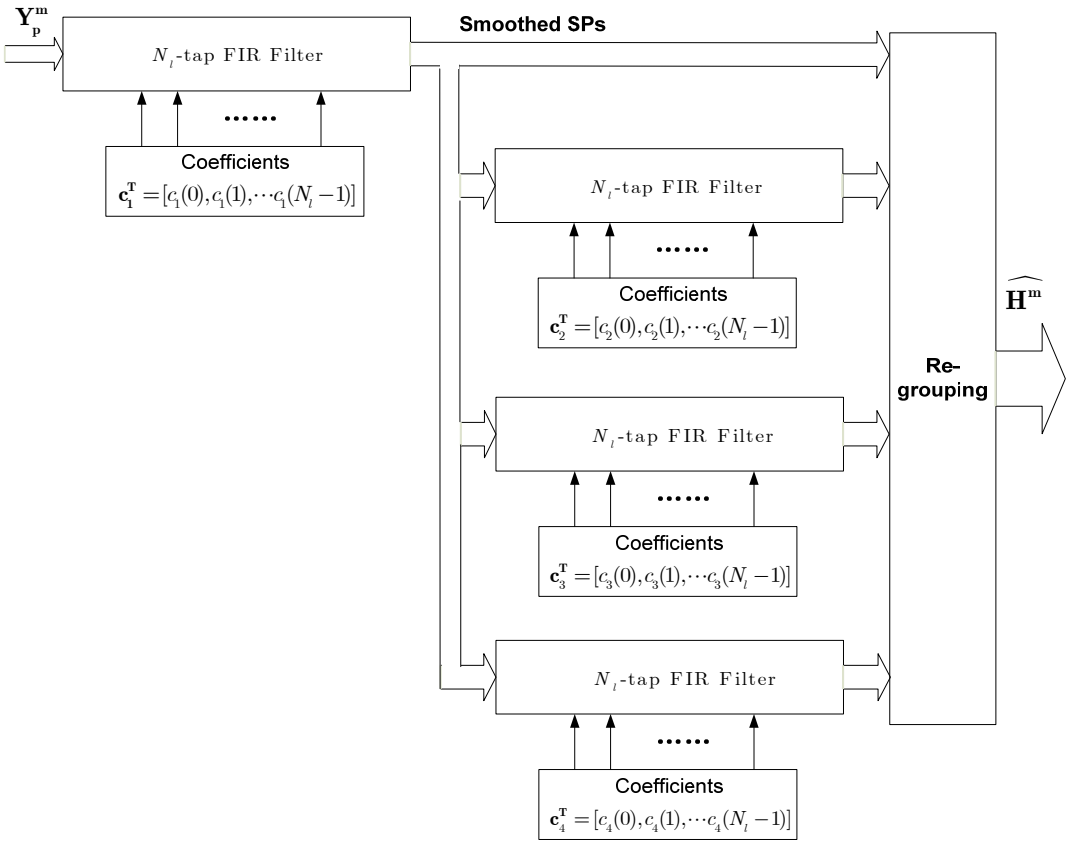


**Figure 4.6 Implementation diagram of mismatched low-rank Wiener estimator in FIR form (static)**

### 4.2.4. Transform Domain LS estimation

The naive transform domain estimator was firstly proposed in [19]. It's based on the fact that: the pilots subsample the channel transfer function in the frequency domain, resulting in a repetitive waveform in the time domain. Therefore, if we zero-pad the received SPs and transform the sequence back into the time domain, we can get repetitive waveforms of the original CIR. Due to the short-delay limitation feature of CIR as analyzed in subsection 3.3.1, the repetitive CIRs are not aliased with each other so that an accurate estimate of CIR, denoted as $\widehat{\mathbf{h}}$ , can be obtained by setting the waveforms longer than delay spread to be zeros. $\widehat{\mathbf{h}}$ can be further transformed back into frequency domain so that an accurate estimate of channel frequency response, denoted as $\widehat{\mathbf{H}}$, can be reconstructed.

However, naive transform domain estimator introduces large estimation error for CMMB. That's because CMMB is null-carrier modulated, so that SPs only locates in the usable-subcarrier band. As analyzed in subsection 3.3.2, if we transform the zero-padded SPs into time domain by N-point IDFT, we get the repetitive version of $\mathbf{h}^m$ , not $\mathbf{h}$ . $\mathbf{h}^m$ is no longer short-delay limited, but has the energy leaked into longer delay paths. As a result, the waveforms of repetitive $\mathbf{h}^m$ are aliased, so that it's impossible to get an accurate estimate of CIR by simply setting the longer delay paths to be zeros.

The energy leaking problem caused by null-carrier modulation is solved by [9], which plays a quantitative analysis and tries to find an optimal estimate of $\mathbf{h}$ by minimizing the square error between the actual channel frequency response on SP locations and the received SPs. It works as follows:

Rewrite $\mathbf{h}$ in (3-5) in the following form:

$$\mathbf{h} = [h(0), h(1), \cdots, h(L_{\max} - 1), 0 \cdots 0]^{\mathbf{T}} = [\mathbf{h_L}, \mathbf{0_{N\text{-}L}}]^{\mathbf{T}} \tag{4-18}$$

Where $\mathbf{h_L}$ consists of channel tap gains of $\mathbf{h}$ within an assumed maximal delay spread $L_{\max}$.

Then, the estimate of $\mathbf{h_L}$ tries to meet the following minimization criteria:

$$\begin{aligned}
\widehat{\mathbf{h_L}} &= \arg\min_{\mathbf{h}_L} \left\{ \left( \frac{1}{N_p} \sum_{k=0}^{N_p - 1} \left| \mathbf{H_p^m}(k) - \mathbf{Y_p^m}(k) \right|^2 \right) \right\} \\
&= \arg\min_{\mathbf{h}_L} \left\{ \left\| \mathbf{H_p^m} - \mathbf{Y_p^m} \right\|^2 \right\} \\
&= \arg\min_{\mathbf{h}_L} \left\{ \left\| \mathbf{F_p}' \cdot \mathbf{h_L} - \mathbf{Y_p^m} \right\|^2 \right\} \\
&= \arg\min_{\mathbf{h}_L} \left\{ (\mathbf{F_p}' \cdot \mathbf{h_L} - \mathbf{Y_p^m}) \cdot (\mathbf{F_p}' \cdot \mathbf{h^m} - \mathbf{Y_p^m})^{\mathbf{H}} \right\}
\end{aligned} \tag{4-19}$$

Where $\mathbf{F_p}' = \mathbf{F}'(\mathbf{P}, [0 : L_{\max} - 1])$, which extracts the first $L_{\max}$ columns of $\mathbf{F}'$, and extracts the rows of $\mathbf{F}'$ according to the pilots pattern vector $\mathbf{P}$. The size of $\mathbf{F_p}'$ is $N_p \times L_{\max}$, and $\mathbf{F_p}'^{\mathbf{H}}$ is the conjugate transpose version of $\mathbf{F_p}'$.

The minimization criteria in (4-19) results in a least square (LS) estimate of $\mathbf{h_L}$, which is expressed in the following form:

$$\widehat{\mathbf{h_L}} = \underbrace{(\mathbf{F_p}'^{\mathbf{H}} \cdot \mathbf{F_p}')^{-1}}_{\mathbf{B}} \cdot \underbrace{\mathbf{F_p}'^{\mathbf{H}} \cdot \mathbf{Y_p^m}}_{\mathbf{A}} \tag{4-20}$$

And the LS estimate of $\mathbf{h}$ can be directly expressed as:

$$\hat{\mathbf{h}} = [\widehat{\mathbf{h_L}}, \mathbf{0_{N\text{-}L}}]^{\mathbf{T}} \tag{4-21}$$

Finally, the estimate of channel frequency response in the usable-subcarrier band, which is $\widehat{\mathbf{H^m}}$, is obtained using (3-9), that is to transform $\hat{\mathbf{h}}$ back into frequency domain, in the following form:

$$\widehat{\mathbf{H^m}} = \mathbf{F}' \cdot \hat{\mathbf{h}} \tag{4-22}$$

Based on above analysis, the key step of the transform domain LS estimator is to compute $\widehat{\mathbf{h}_{\mathrm{L}}}$ using (4-20), which can be further divided into two steps: step A and step B.

For $\mathbf{A} = \mathbf{F}_{\mathrm{p}}{}'^{\mathrm{H}} \cdot \mathbf{Y}_{\mathrm{p}}^{\mathrm{m}}$, it can be realized by zero-padding $\mathbf{Y}_{\mathrm{p}}^{\mathrm{m}}$ into a sequence with the length of N, applying N-point IFFT, and removing the delay paths longer than $L_{\max}$.

For $\mathbf{B} = (\mathbf{F}_{\mathrm{p}}{}'^{\mathrm{H}} \cdot \mathbf{F}_{\mathrm{p}}{}')^{-1}$, it is a constant matrix with the size $L_{\max} \times L_{\max}$, whose value depends only on the pilot pattern vector $\mathbf{P}$, and can be offline-computed and pre-stored in the memory.

As a result, the implementation diagram of transform domain LS estimator is shown in Figure 4.7.

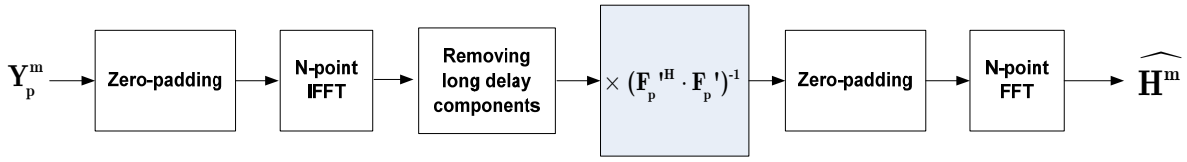


**Figure 4.7 Implementation diagram of transform domain LS estimator**

However, there is still one problem to map the LS estimator in Figure 4.7 onto VDSP: Let $\mathbf{T} = \mathbf{F}_{\mathrm{p}}{}'^{\mathrm{H}} \cdot \mathbf{F}_{\mathrm{p}}{}'$, then the implementation of LS estimator in Figure 4.7 involves a matrix multiplication with a constant matrix $\mathbf{T}^{-1}$. The problem to map the LS estimator onto VDSP is that $\mathbf{T}^{-1}$ is ill-conditioned [9], meaning that the condition number (the ratio between largest element amplitude and smallest non-zero element amplitude) of $\mathbf{T}^{-1}$ exceeds the representation range of a 16-bit machine like VDSP. For instance, for CMMB in the static mode, set $L_{\max}$ as 32, then numerical computation shows that: $\dfrac{\max\left\{|\,\mathbf{T}^{-1}(i,j)\,|\right\}}{\min\left\{|\,\mathbf{T}^{-1}(i,j)\,|\right\}} = 10^{13} \gg 2^{16}$.

Such ill-conditioned matrix inversion problem is solved by applying truncated singular value decomposition (TSVD) on $\mathbf{T}$. That is to find a sub-optimal lower-rank approximation of $\mathbf{T}^{-1}$ with a far smaller condition number [20]. It works as follows:

First, apply regular singular value decomposition on $\mathbf{T}$, such that:

$$\mathbf{T} = \mathbf{U} \cdot \mathbf{S} \cdot \mathbf{V}^{\mathrm{H}} \tag{4-23}$$

Where $\mathbf{U}$ and $\mathbf{V}$ are unitary matrixes and $\mathbf{S}$ is diagonal matrix.

Second, truncate the diagonal values in $\mathbf{S}$, that is:

$$\mathbf{S}'(i,i) = \begin{cases} \mathbf{S}(i,i)\ , & |\mathbf{S}(i,i)| \geq \theta \\ 0, & |\mathbf{S}(i,i)| < \theta \end{cases} \tag{4-24}$$

Where $\theta$ is a pre-defined threshold, and experiments show that a threshold between 0.01 and 0.001 can reach a compromise between the accuracy and a small condition number.

Third, denote that $\mathbf{T}' = \mathbf{U} \cdot \mathbf{S}' \cdot \mathbf{V}^{\mathbf{H}}$, and make the pseudo-inversion of $\mathbf{T}'$ to be the approximation of $\mathbf{T}^{-1}$, that is:

$$\mathbf{T}^{-1} \approx pinv(\mathbf{T}') = pinv(\mathbf{U} \cdot \mathbf{S}' \cdot \mathbf{V}^{\mathbf{H}}) = \mathbf{V} \cdot pinv(\mathbf{S}') \cdot \mathbf{U}^{\mathbf{H}} \qquad (4\text{-}25)$$

Where $pinv(\mathbf{S}')$ performs the pseudo-inversion operation on $\mathbf{S}'$, in which only the non-zero diagonal elements are inversed.

Using TSVD, numerical computation shows that the condition number of $\mathbf{T}^{-1}$ is reduced from $10^{13}$ down to $10^{2}$. As a result, the mapping problem of the transform domain LS estimator in VDSP is solved.

### 4.2.5. Scalability Analysis

- **MSE performance scalability**

The MSE performance under different SNRs for different CE algorithms is shown in Figure 4.8.

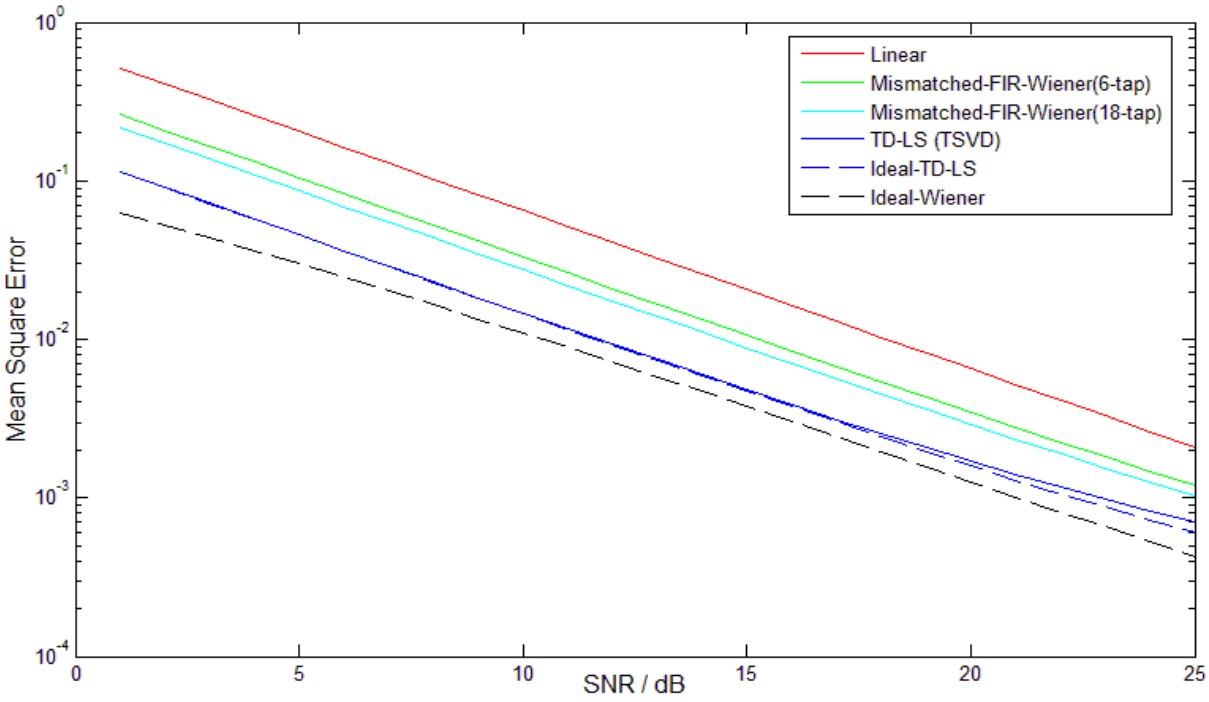


**Figure 4.8 MSE performances for different CE algorithms**

In Figure 4.8, the SNR ranges from 0 to 25 dB, a typical range for wireless communications. The dash lines stand for the MSE performance of the ideal Wiener estimator and the ideal time domain LS estimator (TD-LS), which are infeasible to be implemented in VDSP and only used as benchmarks. The solid lines stand for the MSE performance of three practical CE algorithms: the linear estimator, the Mismatched-FIR-Weiner estimator (6-tap and 18-tap) and the TSVD based TD-LS estimator. For the TSVD based TD-LS estimator, we see when SNR is below 20dB, the MSE performance of the TSVD based TD-LS estimator is quite close to the ideal TD-LS estimator and also closer to the ideal Wiener estimator than two other practical CE algorithms. When SNR is between 20dB and 25dB, the

TSVD based TD-LS estimator introduces a small error floor, but still outperforms other two practical CE algorithms. The error floor reflects the accuracy loss by using TSVD. For the Mismatched-FIR-Weiner estimator, we see that the MSE performance slightly scales with the filtering length. However, such fine-gained scalability is limited: the 18-tap FIR Wiener estimator has only 0.5dB gain comparing with the 6-tap FIR Wiener estimator, but introduces 3 times of arithmetic complexity.

For the MSE performance, we get the relationship that: TD-LS< Mismatched-FIR-Weiner<Linear, where "<" means a lower MSE can be achieved under the same SNR. The MSE performance gap between TD-LS and Mismatched-FIR-Weiner is about 3dB and the gap between Mismatched-FIR-Weiner and Linear is about 3 dB.

- **Complexity scalability**

The arithmetic complexity for 3 CE algorithms is shown in Table 4.1, which lists the estimated number of complex additions and multiplications in 1k mode and 4k mode of CMMB. Recapture that $N$ is the total number of subcarriers, $N_m$ is the number of usable-subcarriers, $N_p$ is the number of SPs in the training cluster, $N_l$ is the FIR filter length and $L_{\max}$ is the assumed maximal delay spread. Note that the operation count for FFT is based on the N-point radix-4 FFT, that is: $N \log_2 N$ complex additions and $\frac{3}{8} N \log_2 N$ complex multiplications [21].

**Table 4.1 Arithmetic complexity of CE algorithms for CMMB**

| CE algorithm | Complex additions | | Complex multiplications | |
|---|---|---|---|---|
| Linear | $(N_m - N_p)$ | 1k: 472 | $(N_m - N_p)$ | 1k: 472 |
| | | 4k:1888 | | 4k: 1888 |
| Mismatched-FIR-Wiener | $4 \cdot N_l \cdot N_p$ | 1k: 3984 (6-tap) | $4 \cdot N_l \cdot N_p$ | 1k: 3984 (6-tap) |
| | | 4k: 15936 (6-tap) | | 4k: 15936 (6-tap) |
| TD-LS | $2N \log_2 N + L_{\max}^2$ | 1k: 21504 | $0.75N \log_2 N + L_{\max}^2$ | 1k: 8704 |
| | | 4k: 65536 | | 4k: 37888 |

Based on the results in Table 4.1, we can see that, for arithmetic complexity, TD-LS> Mismatched-FIR-Weiner> Linear, where ">" means more arithmetic operations. For example in CMMB 1k mode, the ratio of number of complex additions between the linear estimator and the TD-LS estimator is about 3%. And the ratio of number of complex additions between the 6-tap Mismatched-FIR-Wiener estimator and the TD-LS estimator is about 19%.

## 4.3. Channel Quality Estimation Algorithms

Channel quality estimator aims to estimate the channel quality parameters, typically as noise variance, SNR and delay spread. It uses the CMMB preamble in the head of every CMMB TS. A preamble consists of two identical OFDM blocks modulated by two pre-defined unit-powered PN sequences. For the CMMB preamble, (4-1) can be rewritten in the following form:

$$\mathbf{Y}_1^{\mathbf{m}}(k) = \mathbf{H}_1^{\mathbf{m}}(k) \cdot \mathbf{X}^{\mathbf{m}}(k) + \mathbf{W}_1^{\mathbf{m}}(k), \qquad 0 \le k \le N_m - 1 \qquad \text{(4-26)}$$

$$\mathbf{Y}_2^{\mathbf{m}}(k) = \mathbf{H}_2^{\mathbf{m}}(k) \cdot \mathbf{X}^{\mathbf{m}}(k) + \mathbf{W}_2^{\mathbf{m}}(k), \qquad 0 \le k \le N_m - 1 \qquad \text{(4-27)}$$

Where $\mathbf{X}^{\mathbf{m}}$ is a pre-defined PN sequence with unit power, that is $|\mathbf{X}^{\mathbf{m}}(k)|^2 = 1$; $\mathbf{H}_1^{\mathbf{m}}$ and $\mathbf{H}_2^{\mathbf{m}}$ are channel frequency response within the usable-subcarrier band, during the interval of OFDM block 1 and OFDM block 2 of the preamble, with the delay spread of $L_1$ and $L_2$ respectively; $\mathbf{W}_1^{\mathbf{m}}$ and $\mathbf{W}_2^{\mathbf{m}}$ are white Gaussian noise in the usable-subcarrier band, with the noise variance $\sigma^2(\mathbf{W}_1^{\mathbf{m}})$ and $\sigma^2(\mathbf{W}_2^{\mathbf{m}})$.

Based on assumption in section 4.1, channel quality parameters are assumed to be constant in a total CMMB time slot. Since the duration of CMMB preamble is smaller the CMMB time slot length, it's fair to assume that $\sigma^2(\mathbf{W}_1^{\mathbf{m}}) = \sigma^2(\mathbf{W}_2^{\mathbf{m}})$ and $L_1 = L_2$.

A noise variance estimator aims to estimate the noise variance in the following form:

$$\widehat{\sigma_W^2} = \sigma^2(\mathbf{W}_1^{\mathbf{m}}) = \sigma^2(\mathbf{W}_2^{\mathbf{m}}) \qquad \text{(4-28)}$$

Note that using the estimated noise variance, the estimated SNR can be immediately computed as:

$$\widehat{SNR} = 10 \lg \frac{\sigma^2(\mathbf{Y}_n^{\mathbf{m}}) - \widehat{\sigma_W^2}}{\widehat{\sigma_W^2}} \qquad n = 1 \ or \ 2. \qquad \text{(4-29)}$$

Where $\sigma^2(\mathbf{Y}_{\mathbf{n}}^{\mathbf{m}})$ is the variance of received OFDM blocks, reflecting the total power (noise power plus signal power).

A delay spread estimator aims to estimate the delay spread in the following form:

$$\widehat{L} = L_1 = L_2 \qquad \text{(4-30)}$$

### 4.3.1. Noise Variance Estimation

By observing received OFDM blocks $\mathbf{Y}_1^{\mathbf{m}}$ and $\mathbf{Y}_2^{\mathbf{m}}$ in (4-26) and (4-27), we see that noise variance estimation suffers from the fading of a frequency selective channel. Many existing noise variance estimation approaches proposed in literatures need the estimate of channel statistics as a prerequisite [22] [23] [24], so that the performance depends on the accuracy of channel statistics estimates. [25] proposes a noise variance estimation approach by using cyclic prefix (CP). It does not need channel statistics information, but the estimate needs thousands of OFDM blocks to be accurate, therefore not applicable for our run-time adaptive scenario. [26] proposes an approach by using the two identical adjacent OFDM blocks in the synchronization preamble. It is robust against frequency selective channel but is only applicable when the channel is time-invariant. [27] proposes a transform domain approach, taking the fact that the maximal delay of channel taps stays within

the length of cyclic prefix (CP), so that the noise subspace and the channel subspace can be de-coupled. However, as analyzed in subsection 3.3.1, in null-subcarrier modulated OFDM systems as CMMB, the energy of channel taps are leaked into the delay paths longer than the CP length. As a result, the estimated noise variance is enlarged.

In this subsection, first, existing approaches in [26] and [27] are represented and used for CMMB preamble. And then, a novel variance estimation approach is presented, and the simulation shows that it can get rid of the drawbacks of both of these existing approaches.

#### 4.3.1.1. Frequency Domain Differencing

Consider the two demodulated OFDM blocks expressed by (4-26) and (4-27), the estimation of noise variance can be affected by the fading of frequency selective channel, whose statistics is normally unknown. In order to walk around channel statistics, the approach proposed by [26] assumes that the fading channel is constant during the two OFDM blocks, that is $\mathbf{H}_1^{\mathbf{m}} = \mathbf{H}_2^{\mathbf{m}}$. And then, based on the fact that the subtraction of two Gaussian variables is again a Gaussian variable with twice of the variance, the noise variance can be estimated as:

$$\begin{aligned} \hat{\sigma}_W^{\,2} &= \frac{1}{2}\sigma^2(\mathbf{W}_1^{\mathbf{m}} - \mathbf{W}_2^{\mathbf{m}}) \\ &= \frac{1}{2}\sigma^2(\mathbf{Y}_1^{\mathbf{m}} - \mathbf{Y}_2^{\mathbf{m}}) = \frac{1}{2N_{\mathrm{m}}}\sum_{k=0}^{N_{\mathrm{m}}-1} |\, \mathbf{Y}_1^{\mathbf{m}}(k) - \mathbf{Y}_2^{\mathbf{m}}(k)\,|^2 \end{aligned} \tag{4-31}$$

However, such approach is only feasible when the channel is little time-varied. Under fast time-varing channel, the difference between $\mathbf{H}_1^{\mathbf{m}}$ and $\mathbf{H}_2^{\mathbf{m}}$ enlarges the estimated noise variance!

#### 4.3.1.2. Transform Domain De-coupling

A different approach to walk around the channel statistics is to estimate the noise variance in the time domain by using only one OFDM block, so as to be robust to time-varying effect. In this approach, a received OFDM block in the preamble is firstly multiplied by the conjugated version of the pre-known PN sequence in the following form:

$$\begin{aligned} \mathbf{Y}^{\mathbf{m}\prime}(k) &= \mathbf{Y}_n^{\mathbf{m}}(k)\cdot\mathbf{X}^{\mathbf{m}}(k)^* \\ &= \mathbf{H}_n^{\mathbf{m}}(k)\cdot|\,\mathbf{X}^{\mathbf{m}}(k)\,|^2 + \mathbf{W}_n^{\mathbf{m}}(k)\cdot\mathbf{X}^{\mathbf{m}}(k)^* \\ &= \mathbf{H}_n^{\mathbf{m}}(k) + \mathbf{W}_n^{\mathbf{m}\prime}(k) \end{aligned} \tag{4-32}$$

$$0 \le k \le N_m - 1\,;\ n = 1\ or\ 2.$$

Where $\mathbf{W}_n^{\mathbf{m}\prime}$ is still white noise with the same variance as $\mathbf{W}_n^{\mathbf{m}}$.

Furthermore, $\mathbf{Y}^{\mathbf{m}\prime}$ is transformed into time domain again by N-point IDFT, and by using (3-18) and (4-32) we get the following form:

$$\mathbf{y}^{\mathbf{m}\prime} = IDFT(\mathbf{Y}^{\mathbf{m}\prime}, N) = IDFT(\mathbf{H}_n^{\mathbf{m}} + \mathbf{W}_n^{\mathbf{m}\prime}, N)$$

$$= \underbrace{IDFT(\mathbf{H}_n^{\mathbf{m}}, N)}_{\mathbf{h}_n^{\mathbf{m}}} + \underbrace{IDFT(\mathbf{W}_n^{\mathbf{m}\prime}, N)}_{\mathbf{w}_n^{\mathbf{m}\prime}} \tag{4-33}$$

$$= \underbrace{(\mathbf{F}'^{\mathbf{H}}\mathbf{F}') \cdot \mathbf{h}_n}_{\mathbf{h}_n^{\mathbf{m}}} + \underbrace{\mathbf{F}'^{\mathbf{H}} \cdot \mathbf{W}_n^{\mathbf{m}\prime}}_{\mathbf{w}_n^{\mathbf{m}\prime}} \qquad n = 1 \ or \ 2$$

(4-33) can be decomposed into two parts, the part of multi-path channel in the time domain $\mathbf{h}_n^{\mathbf{m}} = (\mathbf{F}'^{\mathbf{H}}\mathbf{F}') \cdot \mathbf{h}_n$, and the part of noise in the time domain: $\mathbf{w}_n^{\mathbf{m}\prime} = \mathbf{F}'^{\mathbf{H}} \cdot \mathbf{W}_n^{\mathbf{m}\prime}$.

For $\mathbf{w}_n^{\mathbf{m}\prime}$, it's the transform domain counterpart of band-limited Gaussian white noise, whose energy can be assumed to be uniformly distributed around the whole band.

For $\mathbf{h}_n^{\mathbf{m}}$, [27] assumes that $\mathbf{h}_n^{\mathbf{m}}$ is an approximation of the CIR in time domain, whose energy is limited within the short-delay paths lower than the CP length: $G$. Based on this assumption, the noise variance is estimated by using only the samples of high-pass delay paths longer than G, in the following form:

$$\hat{\sigma}_W^{\ 2} = \sigma^2(\mathbf{W}_n^{\mathbf{m}}) = \frac{N}{N_{\mathrm{m}}(N-G)} \sum_{k=G}^{N-1} (|\ \mathbf{y}^{\mathbf{m}\prime}(k)\ |^2) \tag{4-34}$$

However, reconsider $\mathbf{h}_n^{\mathbf{m}} = (\mathbf{F}'^{\mathbf{H}}\mathbf{F}') \cdot \mathbf{h}_n$, as analyzed in subsection 3.3.2, the energy of $\mathbf{h}_n^{\mathbf{m}}$ is no-longer short-delay limited, but leaked into the delay paths longer than $G$. As a result, using (4-34), the estimated noise variance is enlarged!

#### 4.3.1.3. Transform Domain Differencing

The novel approach combines the ideas of previous two approaches together, so as to resolve the problems introduced by both of them. Consider the two demodulated OFDM training blocks expressed by (4-26) and (4-27), the two demodulated OFDM blocks are firstly multiplied by the conjugated version of the pre-defined PN sequence, and transformed into the time domain as:

$$\mathbf{y}_1^{\mathbf{m}\prime} = IDFT(\mathbf{Y}_1^{\mathbf{m}\prime}, N) = \mathbf{h}_1^{\mathbf{m}} + \mathbf{w}_1^{\mathbf{m}\prime} \tag{4-35}$$

$$\mathbf{y}_2^{\mathbf{m}\prime} = IDFT(\mathbf{Y}_2^{\mathbf{m}\prime}, N) = \mathbf{h}_2^{\mathbf{m}} + \mathbf{w}_2^{\mathbf{m}\prime} \tag{4-36}$$

Based on the previous analysis in subsection 3.2.2, we know that the energy of channel taps is still leaked into delay paths longer than G. In order to remove the leaked channel taps, we further make difference of the long delay paths of $\mathbf{y}_1^{\mathbf{m}\prime}$ and $\mathbf{y}_2^{\mathbf{m}\prime}$ in the following form:

$$\Delta\mathbf{y}(k) = \mathbf{y}_1^{\mathbf{m}\prime}(k) - \mathbf{y}_2^{\mathbf{m}\prime}(k) \qquad G \le k \le N - G - 1 \tag{4-37}$$

Such differencing operation in long delay paths in time domain has the following effects:

1) For the part of the leaked channel taps: when the channel taps of adjacent OFDM blocks are time-invariant, the leaked energy can be totally removed by differencing; when the multi-path channel is time-varying, there is an extra energy which is introduced by channel difference. However, IDFT migrates the majority of such extra energy into low delay paths so that the effect of time-varying is greatly reduced. Also, the channel leakage energy in long delay paths is inhibited because the channel taps in adjacent OFDM blocks are still correlated.
2) For part of the noise: based on the fact that differencing of two Gaussian variables results in a new Gaussian variable with 2 times of the variance, the noise energy is magnified, which also helps to improve the estimation precision.

Therefore, the noise variance can be estimated as:

$$\hat{\sigma}_W^{\ 2} = \frac{1}{2}\sigma^2(\Delta \mathbf{y}) = \frac{N}{2N_m(N-2G)} \sum_{k=G}^{N-G-1} (\mid \mathbf{y}_1^{\mathbf{m}\prime}(k) - \mathbf{y}_2^{\mathbf{m}\prime}(k) \mid^2) \qquad (4\text{-}38)$$

The implementation diagram of the novel noise variance estimator is shown in Figure 4.9, in which N-point IDFT is implemented by N-point IFFT.

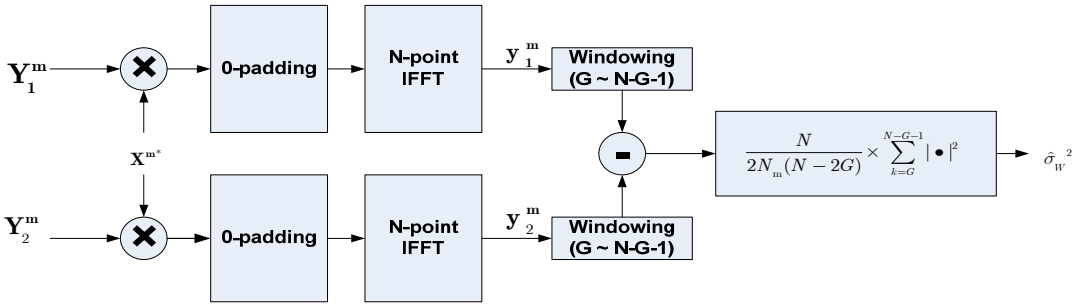


**Figure 4.9 Implementation diagram of the noise variance estimator**

#### 4.3.1.4. Performance

The three noise variance approaches are compared by the simulation under TU6 channel with different Doppler frequencies. For the convenience of evaluation, the average signal noise power in OFDM transmitter is normalized to 1, so that average signal noise ratio (SNR) can be computed as follows, which is used as the metric to performance evaluation.

$$SNR(\hat{\sigma}_W^2) = 10\lg(\frac{1}{\hat{\sigma}_W^2}) \qquad (4\text{-}39)$$

The simulation is run in three modes: the static mode in which the Doppler frequency is 2.4HZ (1km/h when the central carrier frequency is 2.6GHZ); the time-varying mode in which the Doppler frequency is 144.4HZ (60km/h when the central carrier frequency is 2.6GHZ); the fast time-varying mode in which the Doppler frequency is 288.9HZ (120km/h when the central carrier frequency is 2.6GHZ). Figure 4.10, Figure 4.11 and Figure 4.12 show the relationships between the actual SNR

and the estimated SNR (which is determined by the value of estimated noise variance) in these 3 modes: the ideal curve should be a line with slope of 45 degrees, which is the solid line in the three figures. The curve labeled with ”FD-Diff” stands for the frequency domain differencing approach; the curve labeled with “TD-Naive” stands for the time domain de-coupling approach; and the curve labeled with “TD-Diff” stands for the novel approach by transform domain differencing.

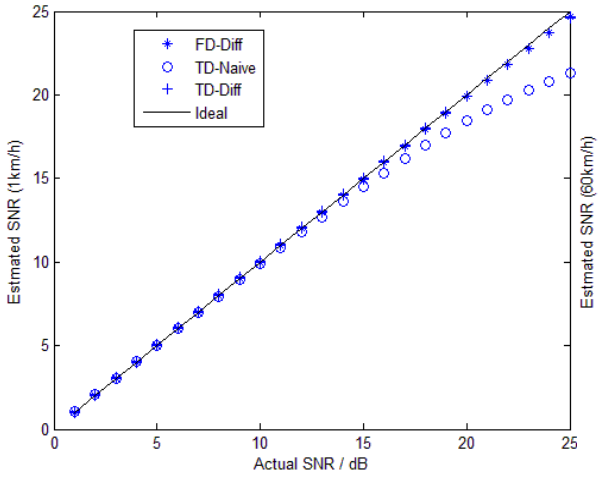


**Figure 4.10 SNR curves in the static mode**

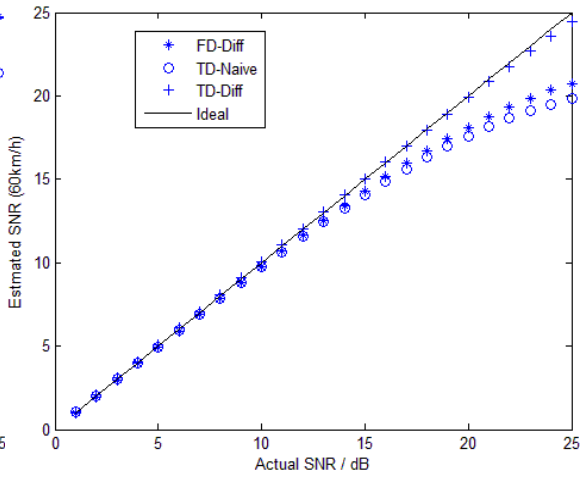


**Figure 4.11 SNR curves in the time-varying mode**

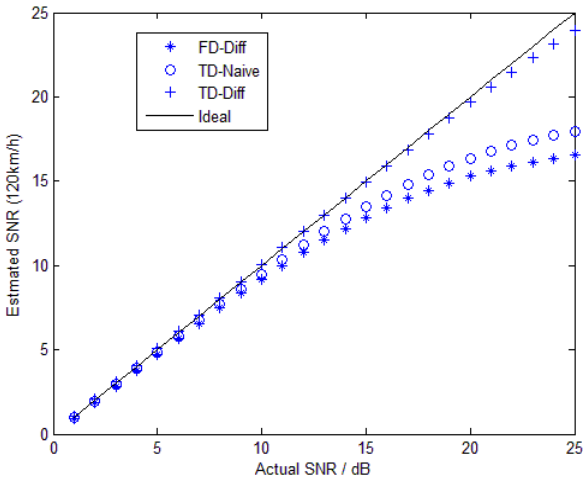


**Figure 4.12 SNR curves in the fast time-varying mode**

In the static mode as shown in Figure 4.10, we see that the both curves of “TD-Diff” and “FD-Diff” coincide with the ideal curve, while the curve of “TD-Naive” falls below the ideal curve when the SNR is higher than 15dB. That’s because in the static mode, the fading channel during the adjacent two OFDM blocks is constant and the differencing in time domain and frequency domain can both remove the total channel energy. For “TD-Naive”, however, the leaked channel energy into noise subspace enlarges the estimated noise variance, and such enlarged value becomes critical when SNR becomes high.

In the time-varying mode as shown in Figure 4.11, we see that the curve of “TD-Naive” still falls down below the ideal curve. That’s because the leaked energy still exists, enlarging the estimated

noise variance. We can also see that the curve of our "TD-Diff" approach still coincides with the ideal curve, while the curve of "FD-Diff" falls down below the ideal curve when the SNR is higher than 15dB. That's because the fading channel between adjacent OFDM blocks are no longer identical. For our "TD-Diff" approach, the majority of such difference is migrated into short delay paths and does not contribute to the final result, while for "FD-Diff" approach, however, such difference is totally taken into computation of noise variance so that estimated noise variance is enlarged. Such enlarged value becomes critical when SNR becomes high.

In the fast time-varying mode as shown in Figure 4.12, we see that, although a little bit falls down below the ideal curve, our "TD-Diff" approach still keeps close to the ideal curve, showing the robustness against fast time-varying channel. We can also see that, the curve of "FD-Diff" approach further falls down, with the performance even poorer than "TD-Naive" approach, which is caused by the even larger difference between $\mathbf{H}_1^{\mathrm{m}}$ and $\mathbf{H}_2^{\mathrm{m}}$.

### 4.3.2. Delay Spread Estimation

For delay spread estimation, [28] proposes an approach by using the correlation function between cyclic prefix and the identical part in un-demodulated OFDM data block, and by observing the relationship between correlation length and the correlation value. However, it has to be played over hundreds of OFDM blocks to be accurate, so as not applicable for our run-time adaptive scenario. [29] proposes an approach to estimate the delay spread in the frequency domain, by using relationship between the time-domain delay spread and the frequency-domain correlation. However, the technique is based on the assumption of an exponentially decaying power delay profile, which is only suitable for indoor channel model, but introduces large estimation errors under urban model like TU6. [30] proposes an approach to estimate delay spread in the time domain by simply making thresholds on the noised CIR , while the threshold value is determined by the noise power in long delay paths. The problem of this approach is that, as analyzed early in subsection 3.3.2, the energy of the re-constructed CIR after IDFT is leaked into long delay paths, so that the threshold value is not accurate and a large estimation error is introduced. [31] proposes another time-domain approach where delay spread estimation is based on the ratio between channel energy in short delay paths and the noise energy in long delay paths. The problem of this approach is that the energy ratio is different for different channel profiles, and is unknown for the estimator in real situations.

In this subsection, a novel delay spread approach is proposed. The approach is played in time domain and uses 3 techniques to solve the energy leaking problem and to improve the estimation accuracy: re-windowing, energy smoothing, and two-stage searching. It is further compared with the existing approach in [31], and shows the performance out-performing.

#### 4.3.2.1. Technique 1: Re-windowing

As analyzed in subsection 3.3.2, a rectangular window function is multiplied with the channel frequency response in the frequency domain to keep only the usable-subcarriers. Such windowing in the frequency domain causes the energy leaking of CIR in the time domain. One technique to migrate this problem is to multiply another window function with smoothing transition edges in

frequency domain (such as Hanning window), so that the side lobe levels in time domain decrease much faster than those of a rectangular window [13].

Therefore, for the received OFDM blocks in the preamble as shown in (4-26) and (4-27), a new window function is applied. The new window function is a Hanning window function multiplied with conjugate version of $\mathbf{X}^{\mathrm{m}}$ . Such re-windowing results in the following form:

$$\begin{aligned}\mathbf{Y}^{\mathrm{hann}}(k) &= \mathbf{Y}_n^{\mathrm{m}}(k)\cdot[\mathbf{X}^{\mathrm{m}}(k)^*\cdot\mathbf{Hann}(k)] \\ &= \mathbf{H}_n^{\mathrm{m}}(k)\cdot\mid\mathbf{X}^{\mathrm{m}}(k)\mid^2\cdot\mathbf{Hann}(k)+\mathbf{W}_n{}'(k)\cdot\mathbf{Hann}(k) \\ &= \mathbf{H}_n^{\mathrm{m}}(k)\cdot\mathbf{Hann}(k)+\mathbf{W}_n{}'(k)\cdot\mathbf{Hann}(k) \\ 0\le k\le N_m-1;\ n&=1\ or\ 2.\end{aligned} \tag{4-40}$$

$\mathbf{Y}^{\mathrm{hann}}$ is further transformed into the time domain by N-point IDFT, which can be done by zero-padding on the two edges to be a sequence of N, and doing N-point IFFT:

$$\mathbf{y}^{\mathrm{hann}} = IDFT(\mathbf{Y}^{\mathrm{hann}}, N) \tag{4-41}$$

In Figure 4.13, the waveform of $\mathbf{y}^{\mathrm{hann}}$ is compared with that of $\mathbf{y}^{\mathrm{m}\prime}$ in (4-35). And we can see that by re-windowing, the leaked energy in longer delay paths ( $k \geq L$ ) is inhibited.

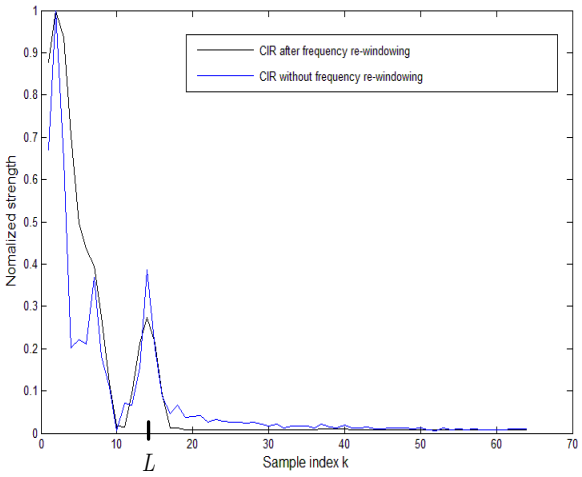


**Figure 4.13 Energy leaking inhibition by re-windowing**

#### 4.3.2.2. Technique 2: Energy Smoothing

The time-domain CIR $\mathbf{y}^{\mathrm{hann}}$ contains noises which may cause glitches that can be mis-recognized to be a channel path, and would cause mis-estimation of the delay spread if the location of a glitch is longer than the actual delay spread. Such problem is solved by applying an energy smoothing operation. That is to computing the energy of $\mathbf{y}^{\mathrm{hann}}$ within a sliding window. The window length is

chosen to be CP length, which is assumed larger than maximal delay spread. The sequence after energy smoothing can be expressed in the following form:

$$\mathbf{y}^{\mathbf{E}}(k) = \sum_{i=0}^{G-1} |\, \mathbf{y}^{\mathbf{hann}}(i+k) \,|^2 \quad 0 \le k \le N-G \tag{4-42}$$

Figure 4.14 shows the waveform of noised CIR before and after energy smoothing, and we can see that the noise glitches are removed after energy smoothing.

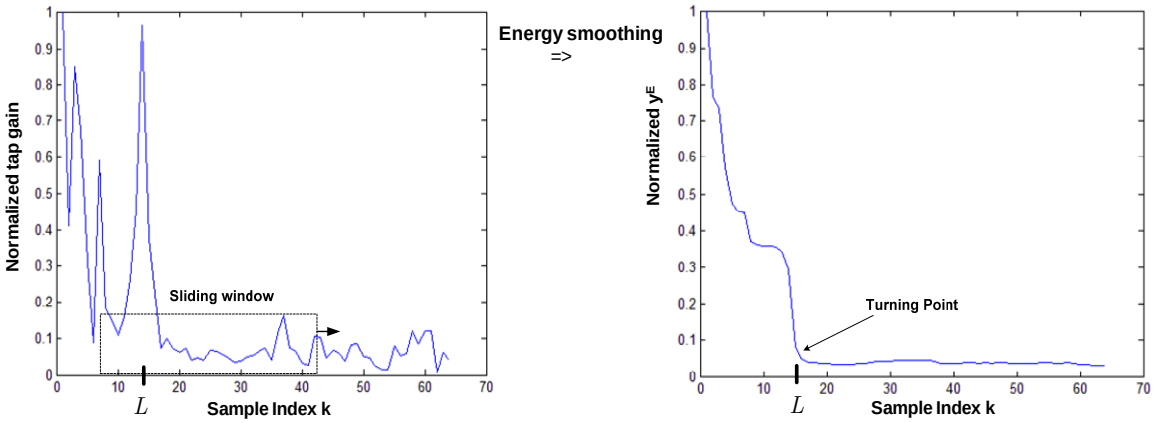


**Figure 4.14 Glitches reduction by energy smoothing**

Also in Figure 4.14, extra bonus can be found that, after energy smoothing, a waveform with decreasing strengths is obtained, which eases to apply the thresholds. Furthermore, the longest delay path which indicates the delay spread now becomes a turning point in $\mathbf{y}^{\mathbf{E}}$. The further processing is to find the location of such turning point.

#### 4.3.2.3. Technique 3: Two-Stage Searching

A Two-stage searching strategy is applied on $\mathbf{y}^{\mathbf{E}}$ to find the turning point which indicates the longest delay path: coarse searching by adaptive threshold, and fine searching by maximal gradient.

- **Coarse searching by adaptive thresholds**

This step aims to return a coarse range [K1, K2] within which the turning point locates. The range is obtained by setting two thresholds for $\mathbf{y}^{\mathbf{E}}$: $Th$ and 2 times of $Th$, while $Th$ is the noise power in the long delay paths and can be computed in the following form:

$$Th = \frac{1}{N-2G} \sum_{k=G}^{N-G-1} \mathbf{y}^{\mathbf{E}}(k) \tag{4-43}$$

And the step of coarse searching by adaptive thresholds is in Figure 4.15.

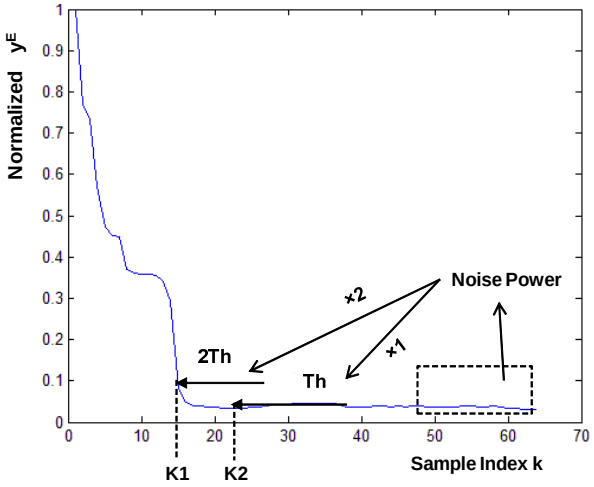


**Figure 4.15 Coarse searching by adaptive thresholds**

- **Fine searching by maximal gradient**

The step aims to return the exact index of the turning point within the coarse range, which is the estimated delay spread. An important feature of a turning point is that it has a peak gradient. Therefore, the turning point is fine searched in the given range by finding the index with maximal gradient. Gradient computation can be realized by simply making difference with a subsequent neighbor point. The mathematical expression of fine searching is in the following form:

$$\widehat{L} = \max_{K1 \le k \le K2} \left\{ \Delta \mathbf{y}^{\mathbf{E}}(k) \right\} \tag{4-44}$$

Where $\Delta \mathbf{y}^{\mathbf{E}}(k) = \mathbf{y}^{\mathbf{E}}(k) - \mathbf{y}^{\mathbf{E}}(k+1) \qquad 0 \le k \le G-1$

The total implementation diagram of our novel Delay spread estimator is shown in Figure 4.16.

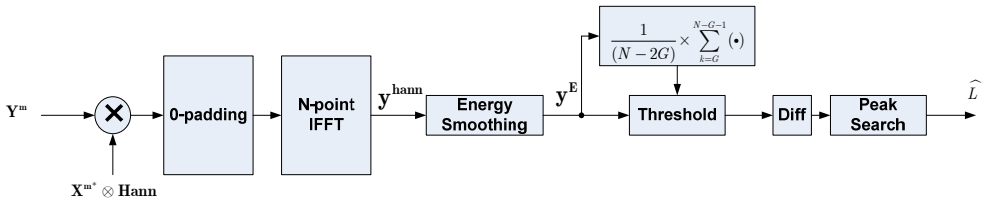


**Figure 4.16 Implementation diagram of the delay spread estimator**

#### 4.3.2.4. Performance

The novel approach is compared with the energy ratio approach in [31] for delay spread estimation using CMMB preamble. In order to make sure of the robustness, the simulation is performed under different channel profiles with different delay spreads. The performance is evaluated in the metric of correct detection rate (CDR), where the delay spread is considered to be correctly detected if the following criteria is fulfilled:

$$\left| \widehat{L} - L \right| \le 1 \tag{4-45}$$

Figure 4.17, Figure 4.18 and Figure 4.19 show the correct detection rate under 3 different LTI (linear time-invariant) channels, in which the gain of channel taps keeps constant. We see that under LTI channels, the energy ratio approach is better than the novel approach. That is because for the energy ratio approach, the estimator is assumed to know the energy ratio of every testing channel model and that the ratio always keeps constant. However, this is impossible in real situations.

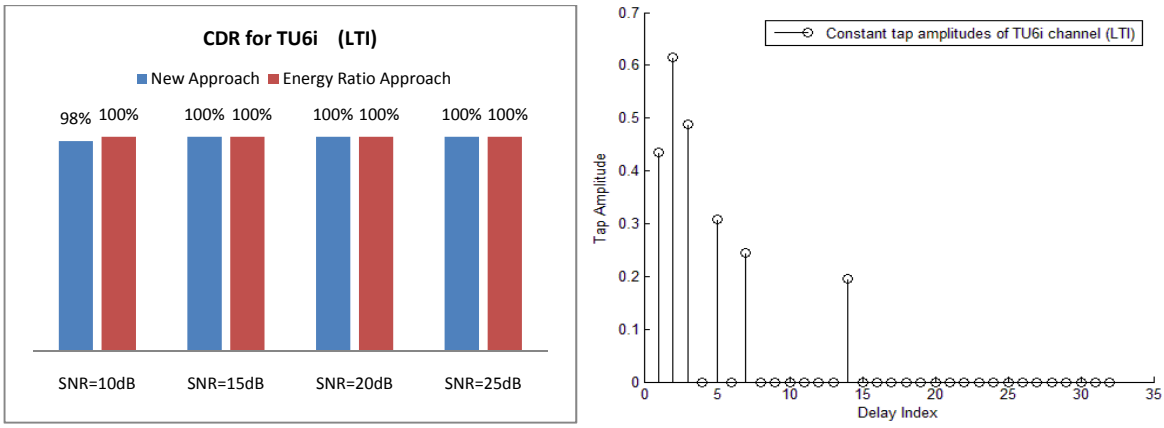


**Figure 4.17 CDR in TU6i channel**

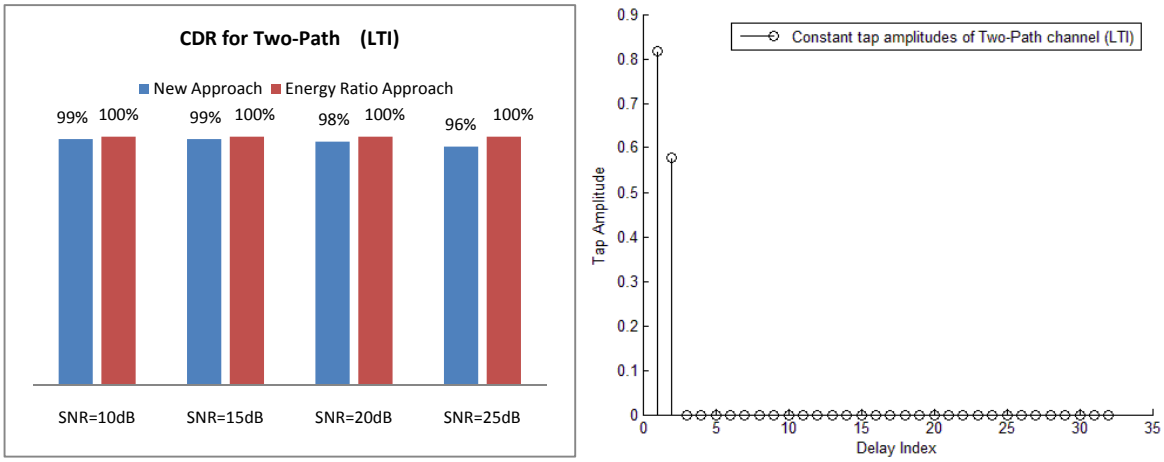


**Figure 4.18 CDR under Two-Path channel**

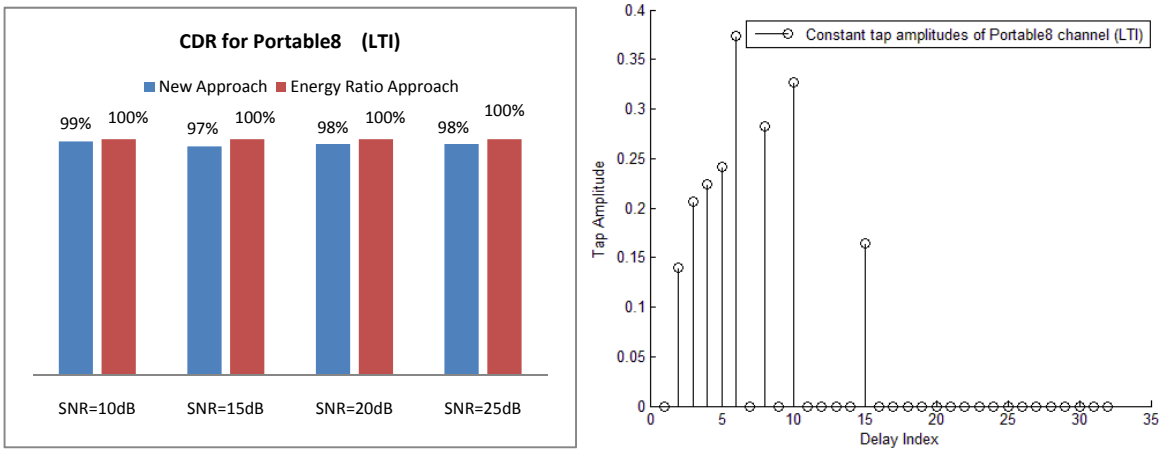


**Figure 4.19 CDR under Portable8 channel**

Figure 4.20 and Figure 4.21 show the correct detection rate under 3 different WSSUS (wide sense stationary uncorrelated scattering) channels, in which the gain of channel taps are time-variant. We see that, under time-variant channel, our novel approach outperforms the energy ratio approach. That is because the energy ratio is time-variant which leads to temporary mismatch for the energy ratio approach, while our approach does not need any information of the channel model. Another finding is that the SNR do not drastically affects the correct detection rate. That is because the noise energy only builds an offset to the total energy profile and can be subtracted.

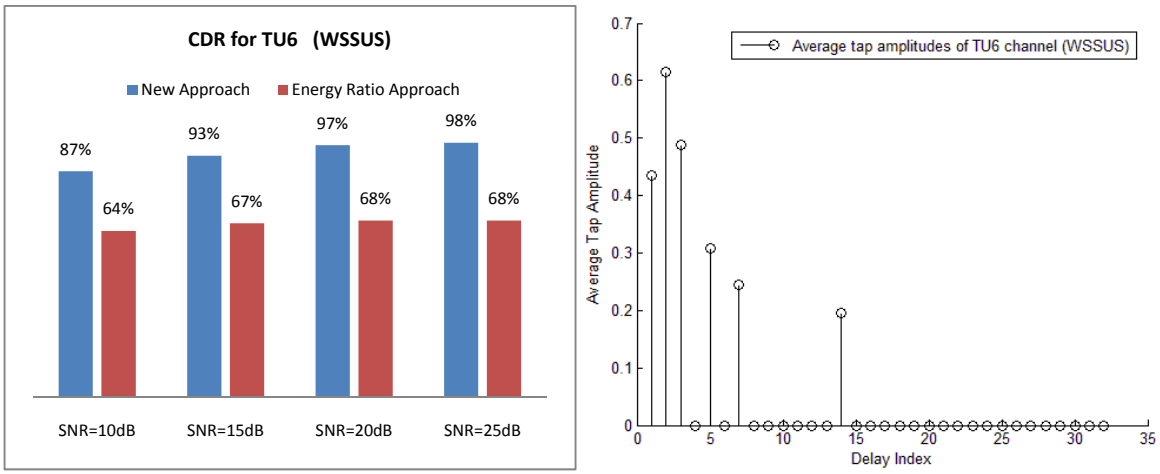


**Figure 4.20 CDR under TU6 channel (1km/h)**

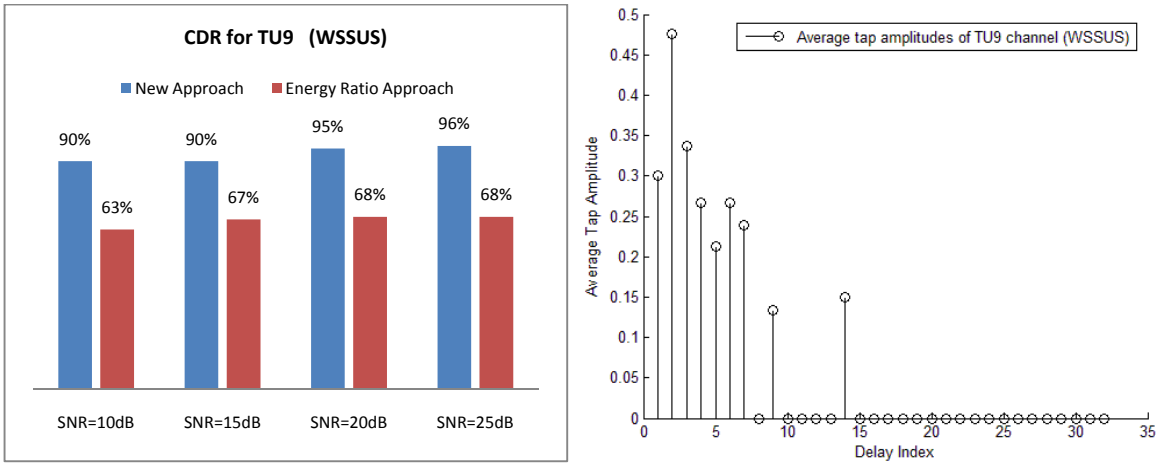


**Figure 4.21 CDR under TU9 channel (1km/h)**

Figure 4.22 and Figure 4.23 further show the effects of time-varying speed on correct detection rate. And we see that the effect is quite little. That is because time-varying speed only affects the changing rate of the gains on each channel tap, while the delay spread is not affected.

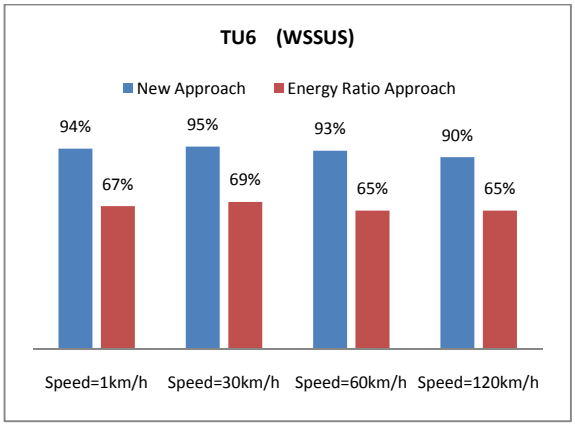


**Figure 4.22 CDR under TU6 channel (15dB)**

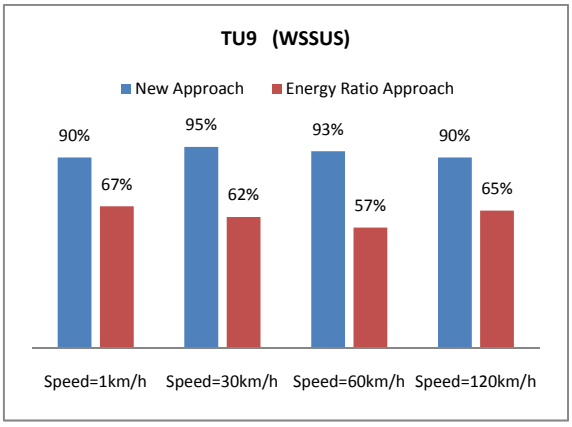


**Figure 4.23 CDR under TU9 channel (15dB)**

### 4.3.3. Computation Overheads Analysis

The complexity adaptive channel estimator aims at run-time load reduction and power saving. However, channel quality estimation (CQE) introduces computation overheads, which must be analyzed. Consider CMMB TS structure, each TS is headed with a preamble and then followed by 53 OFDM data blocks. CQE has to be performed only once while CE has to be performed 53 times per TS. Therefore, the computation overheads per OFDM data block can be calculated by multiplying the computation loads of CQE algorithms with a factor of 1/53.

Table 4.2 shows the computation overheads for CQE algorithms, in the number of arithmetic operations (complex additions and complex multiplications).

**Table 4.2 Computation overheads of CQE algorithms**

| CQE algorithms | Complex additions (Overhead) | | Complex multiplications (Overhead) | |
|---|---|---|---|---|
| Noise variance estimation | $\frac{2N(2+\log_2 N)-4G}{53}$ | 1k: 202<br>4k: 966 | $\frac{N(3+0.75\log_2 N)-2G}{53}$ | 1k: 89<br>4k:415 |
| Delay spread estimation | $\frac{N(2+\log_2 N)}{53}$ | 1k: 106<br>4k: 502 | $\frac{N(2+0.375\log_2 N)}{53}$ | 1k:52<br>4k: 236 |

Comparing the computation overheads of CQE algorithms in Table 4.2 with the arithmetic complexity of the most computation intensive CE algorithm in Table 4.1, we see that that the computation overheads are subtitle. For example in CMMB 1k mode, the total number of complex additions of the noise variance estimator and the delay spread estimator is less than 2% of that of the TD-LS estimator.

## 4.4. Power manager

The power manager sorts the estimated channel quality parameters according to the pre-defined thresholds, and chooses one of the three CE algorithms accordingly. The thresholds should be specified in such a way that, under a given channel quality, the power manager can select the most

computation economic CE algorithm which meets the BER (bit error rate) requirement of the standard. There are two ways to derive the thresholds: the experimental approach on real platforms, and the analytical approach under pre-defined scenarios.

### 4.4.1. The Experimental Approach on Real Platforms

In the testing phase, the thresholds for the channel quality parameters can be obtained by doing experiments on real platforms. Such experimental approach is shown in Figure 4.24, in which PC-A works as a CMMB transmitter to modulate the raw bits into CMMB data stream and send it to the channel emulator. The channel emulator emulates a real fading channel and adds the additive noise according to the setting by PC-B. The distorted and noised signals are received by the receiver board where the baseband processing including CE is applied. The transmitted raw bits from PC-A are compared with the demodulated bits from receiver board in PC-B, and BER is computed. Starting from the simplest CE algorithm in the receiver board, PC-B keeps on worsening the channel quality by decreasing SNR and increasing delay spread in the channel emulator, while at the same time monitoring the BER, when the BER exceeds the requirements of CMMB, a set of the channel quality parameter thresholds for the CE algorithm is obtained. Then PC-B selects a more accurate CE algorithm in the board and further worsens the channel quality in channel emulator until BER exceeds the requirements again. By repeating this procedure, the channel quality parameter thresholds for all CE algorithms can be obtained.

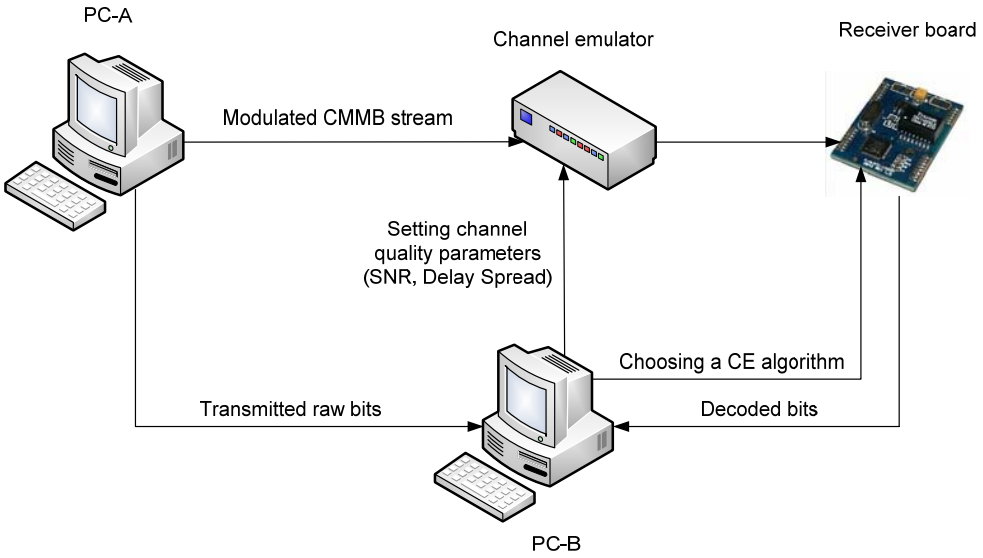


**Figure 4.24 Experimental approach to obtain the channel quality parameter thresholds**

### 4.4.2. The Analytical Approach under Pre-defined Scenarios

However, in the current design phase, since the subsequent components such as channel equalizer and channel decoder are still missing, it is impossible to directly obtain the thresholds by doing experiments on a real platform. In order to demonstrate the original ideal in the design phase, the thresholds are analytically derived under a pre-defined scenario. The scenario is specified in Table 4.3, and is consistent with CMMB standard [1].

**Table 4.3 Pre-defined scenario consistent with CMMB standard**

| | |
|---|---|
| Channel model: | TU6 in the static mode (1km/h) |
| CMMB mode | 1K mode |
| Equalizer type: | Zero-forcing Equalizer |
| Symbol mapping scheme: | 4-QAM |
| LDPC rate: | ½ |
| BER requirement: | $\leq 3\times10^{-6}$ |

In this pre-defined scenario, since delay spread is always constant in TU6 channel, SNR is the only tuneable channel quality parameter. The SNR thresholds are derived in the following procedure: the BER requirements for the total CMMB receiver are firstly transformed into a MSE requirement for the channel estimator. And then, the thresholds are obtained by examining the MSE-SNR curves of different CE algorithms under the given scenario.

#### 4.4.2.1. A Pessimistic MSE Requirement

Under the given scenario in Table 4.3, the first step to design the power manager is to transform the BER requirement for the total CMMB receiver into the MSE requirement for the channel estimator. It's a backward engineering process as shown in Figure 4.25.

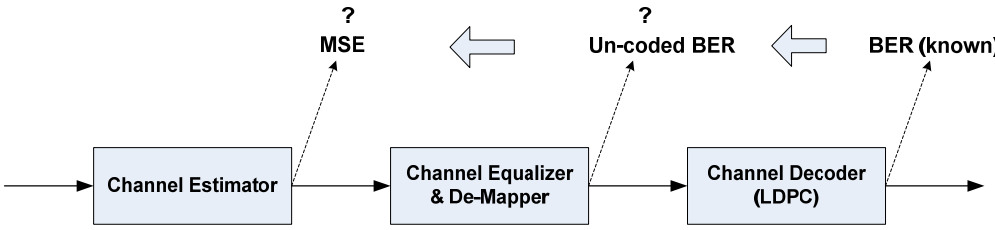


**Figure 4.25 Backward engineering to derive MSE requirement from BER requirement**

- **From the coded BER requirement to the un-coded BER requirement**

[32] and [33] report the BER curves for CMMB LDPC decoder in ½ rate, and show that $BER \leq 3\times10^{-6}$ can be achieved if $\frac{Eb}{N0} \leq 1.7dB$, where $\frac{Eb}{N0}$ is the signal noise ratio in the AWGN transmission model. The signal noise ratio requirement under the AWGN model can be transformed into un-coded BER requirement under the binary symmetric channel (BSC) model [34], in the following form:

$$P_{un-coded} = \int_{\sqrt{\frac{Eb}{N0}}}^{\infty} \frac{1}{\sqrt{2\pi}} \exp(-\frac{\alpha^2}{2}) d\alpha$$

$$= \frac{1}{2} erfc(\sqrt{\frac{Eb}{N0}}) \qquad (4\text{-}46)$$

$$\leq \frac{1}{2} erfc(\sqrt{10^{\frac{1.7}{10}}}) = 0.0427$$

Therefore, the un-coded BER requirement is derived.

- **From the un-coded BER requirement to the MSE requirement**

[35] takes the mathematical analysis on the relationships among MSE, SNR and un-Coded BER, based on the assumption of using zero-forcing equalizers. It concludes that for 4-QAM with Gray bit mapping, the MSE, SNR and un-coded BER follow the following relationship:

$$P_{un-coded} \approx \frac{1}{2}[1 - \sqrt{\frac{\pi\alpha}{2\beta}} \exp(\frac{\alpha\gamma + 2}{2\beta\gamma}) erfc(\sqrt{\frac{\alpha\gamma + 2}{2\beta\gamma}})] \quad (4\text{-}47)$$

Where $P_{un-coded}$ is un-coded BER, $\alpha$ is the average amplitude of channel frequency response, $\beta$ is MSE of the channel estimator, and $\gamma$ is SNR.

The relationships among $P_{un-coded}$, $\gamma$ (SNR) and $\beta$ (MSE) can be shown in Figure 4.26.

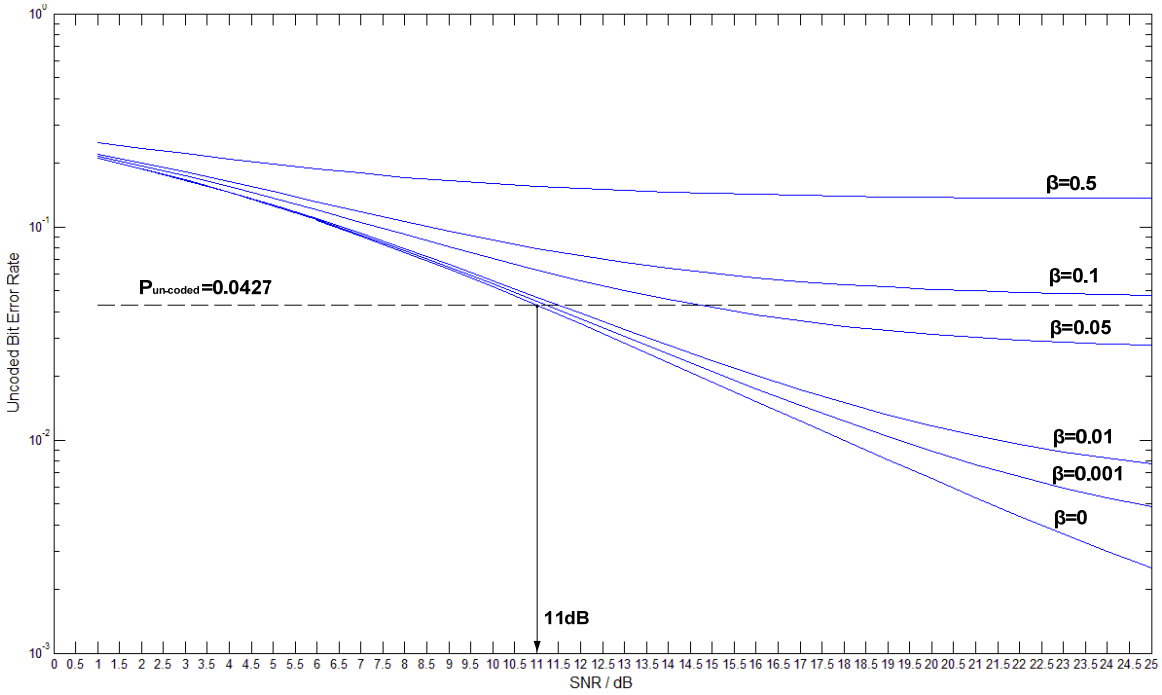


**Figure 4.26 Relationships among $P_{un-coded}$, $\gamma$ (SNR) and $\beta$ (MSE)**

In Figure 4.26, we see that, if the SNR is smaller than 11dB, no matter how small $\beta$ it is, the un-coded BER requirement ( $P_{un-coded} \leq 0.0427$ ) can never be fulfilled. Therefore, a necessary condition to meet the un-coded BER requirement is that:

$$\gamma \geq 11dB \quad (4\text{-}48)$$

Under such neccessary condition, we need to find a pessimistic MSE requirement. That is done by forcing $\gamma = 11dB$ in (4-47), and keeps on decreasing $\beta$ values until $P_{un-coded} \leq 0.0427$ is just met. And the numerical computation in Matlab shows that:

$$\beta \leq 0.018 \quad (4\text{-}49)$$

There are 2 implications of (4-48) and (4-49): firstly, for the zero-forcing equalizer, if the SNR is smaller than 11dB, no matter how accurate the channel estimation is, the un-coded BER requirement is not fulfilled; Secondly, the if SNR is larger than 11dB, then the , $MSE \leq 0.018$ is a sufficient condition (thus pessimistic) to fulfill the un-coded BER requirement.

Based on above analysis, a pessimistic MSE requirement for the channel estimator is specified in the following form:

$$MSE \leq 0.018 \tag{4-50}$$

#### 4.4.2.2. Pessimistic SNR Thresholds

Based on the scenario specified in Table 4.3 and the pessimistic MSE requirement derived in (4-50), the thresholds for SNR can be obtained by examining the MSE-SNR curves as shown in Figure 4.27. Note that in order to take consideration of precision loss due to fixed-point computation in VDSP, the MSE curves in Figure 4.27 are based on 16-bit fixed-point version of CE algorithms in Matlab, which conforms to the bit precision of VDSP.

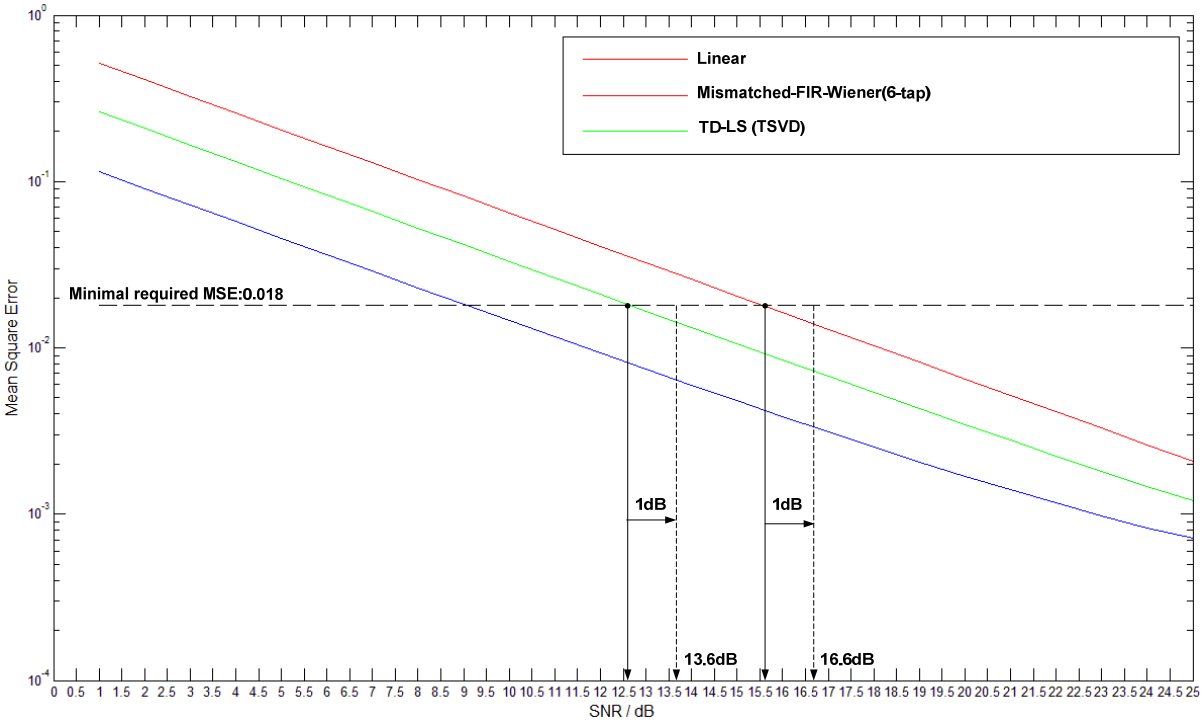


**Figure 4.27 Pessimistic thresholds by MSE-SNR curves and MSE requirement**

In Figure 4.27, the horizontal line stands for the minimal required MSE, and it intersects with the three MSE-SNR curves. Then, starting from the intersection points, two vertical arrows are derived indicating the SNR thresholds. Finally, the threshold values are further right shifted by 1dB. That is because for the derived noise variance and SNR estimation algorithm, simulation shows the worst case deviation is smaller than 1dB. The right shifting makes the derived thresholds pessimistic, but guarantees that the MSE requirement can always be fulfilled.

Therefore, in the pre-defined scenario in Table 4.3, the two SNR threshold values for algorithm switching are that:

$$\begin{aligned} Th1 &= 13.6dB \\ Th2 &= 16.6dB \end{aligned} \tag{4-51}$$

The derived threshold values are pessimistic because the MSE requirement is pessimistic. "Pessimistic" means, in the pre-defined scenario, by using $Th1$ and $Th2$ in the power manager, the final BER requirement can be guaranteed in the pre-defined scenario. However, the lower threshold values may still exist, which can reduce more run-time loads while at the same time still guarantee the final BER requirement. The integrated simulation in the next chapter will reuse the scenario in Table 4.3 and will use the derived pessimistic thresholds to evaluate the run-time load reduction factor comparing with a fixed channel estimator.

## 4.5. Chapter Conclusion

This chapter presents the detailed design of the run-time complexity adaptive channel estimator for CMMB. The feedforward architecture is chosen in order to fulfill the run-time requirements. For channel estimation, the linear estimator, the Wiener estimator and time domain LS estimator are designed independently. In particular, in order to be practical to be mapped in VDSP, the Wiener estimator is designed in a mismatched low-rank FIR form, while the time domain LS estimator uses TSVD to solve the ill-conditioned matrix inversion problem. Simulation shows that the derived 3 practical CE algorithms have scalable estimation accuracies, while the analysis on arithmetic complexity shows that they also have scalable complexities. For channel quality estimation, a novel algorithm for noise variance estimation and a novel algorithm for delay spread estimation are presented, and simulations show that they outperform existing algorithms. A power manager needs the specification of thresholds of the channel quality parameters. The experimental approach can easily derive thresholds of all channel quality parameters but it is only feasible in the testing phase when all components in the CMMB receiver is available. In the current design phase, the thresholds can only be derived by pessimistic analysis under a pre-defined scenario. The scenario used is consistent with CMMB standard, under which SNR is used as the only channel quality parameter to switch CE algorithms. SNR thresholds are obtained firstly by analytically deriving a pessimistic MSE requirement from the BER requirement of the CMMB standard, and then by examining the intersection points between MSE requirement curve and MSE-SNR curves of different CE algorithms.

# 5. Results

In this chapter, the derived algorithms are translated into C code and implemented in a VDSP simulator, and a load profile is obtained from the VDSP simulator. Then, the derived load profile is inserted into the Matlab test bench, and the integrated simulation is performed under a pre-defined scenario. Finally, the load reduction factor comparing with a fixed channel estimator is evaluated.

## 5.1. Load Profile

In order to evaluate the cycle reduction factor of our complexity adaptive channel estimator in VDSP, the derived channel estimation algorithms and channel quality estimation algorithms are further mapped by C code onto VDSP, and a VDSP simulator reports the cycle number of each algorithm module.

For the convenience of evaluation, each algorithm module is implemented into a function. For channel estimation, three channel estimation functions have the same specification of input and output. The input is a pointer pointing to the SP buffer which stores received SPs. The output is a pointer pointing to the channel buffer which stores the estimated channel frequency response in the useable-subcarrier band. For channel quality estimation functions, the input is the pointer pointing to the preamble buffer, which contains the two identical OFDM blocks in the preamble. The output of the SNR estimator is the estimated SNR value, and the output of the delay spread estimator is the estimated delay spread value.

For algorithm mapping, typical function modules such as Radix-4 FFT, Radix-4 IFFT,FIR filtering, peak searching and memory copy (aligned and unaligned) are already implemented in an existing VDSP code library and are reused in our implementation. The main contributor to the number of computation cycles are FFT, IFFT and FIR filtering. The FFT and IFFT are configured to be DFT length of CMMB and the filtering length of FIR filter is configured to be the Wiener filter length. For the TD-LS estimator, the FFT module and the IFFT module are separately reused once. For the FIR-Wiener estimator, a 6-tap FIR filtering module is reused for 4 times. For the SNR estimator, the IFFT module is reused twice. For the delay spread estimator, the IFFT module is reused once.

The normalized cycle numbers obtained from VDSP simulator for each algorithm module are showed in Figure 5.1. Note that the cycle numbers of CQE algorithms (SNR estimation and delay spread estimation) are further normalized by a factor of 1/53. That is because the execution frequency of CQE algorithms is 1/53 of that of CE algorithms. Also note that the computation overheads introduced by the power manager is also included.

In Figure 5.1 we see that the loads of three CE algorithms in VDSP scale very well. The processing load of the 6-tap FIR-Wiener estimator is roughly 33% of that of the TD-LS estimator. And the processing load of the linear estimator is roughly 8% of the TD-LS estimator. These two ratios are higher than the theoretical estimated ratios (19% and 3%) in subsection 4.2.5, meaning that the TD-LS estimator is more fit in VDSP. The reason is that, benefit from the hardware function unit of radix-4 butterfly within the VDSP core, FFT and IFFT computations are more efficient.

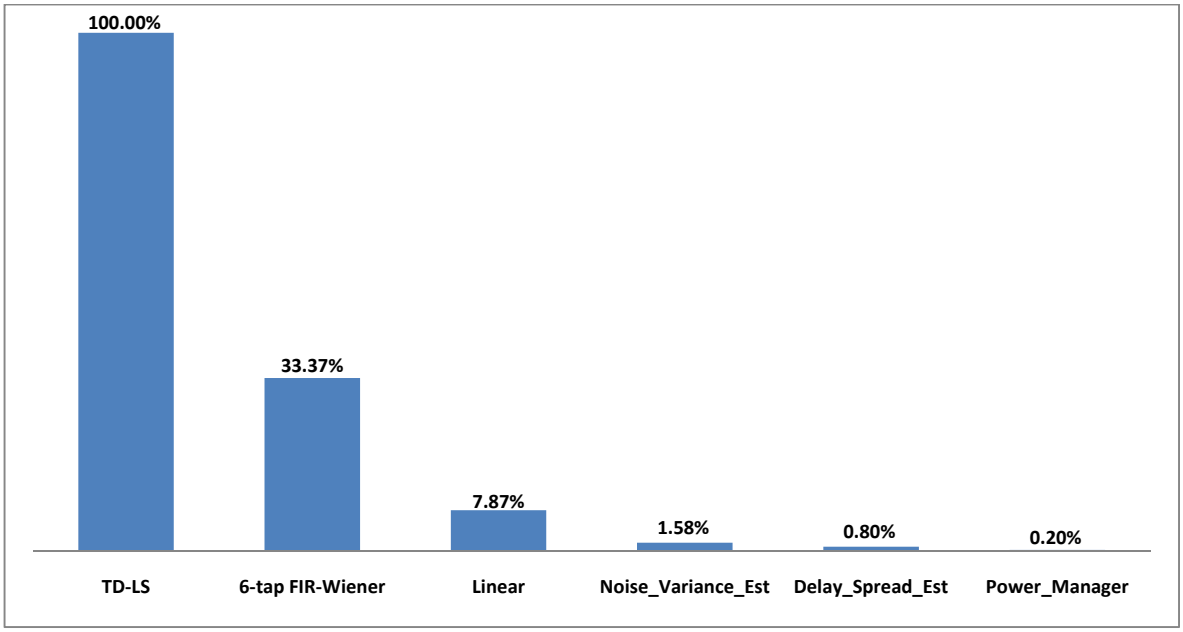


**Figure 5.1 Normalized cycles of algorithm modules in VDSP**

In Figure 5.1 we can also see that the total computation overhead is less than 3% of the TD-LS estimator, such ratio is close to the theoretical computation overhead estimates (2%) in subsection 4.3.3, meaning that indeed the computation overhead introduced by CQE algorithms and the power manager is subtitle.

## 5.2. Intergrated Simulation

The algorithm modules of our run-time complex adaptive channel estimator are integrated into a Matlab test bench and the obtained load profile for the algorithm modules is inserted into the test bench for integrated simulation. The simulation scenario is specified in Table 5.1. It follows the pre-defined scenario from Table 4.3 but modifies the BER requirement to be the derived pessimistic MSE requirement.

**Table 5.1 Scenario for integrated simulation**

| | |
|---|---|
| Channel model: | TU6 in the static mode (1km/h) |
| CMMB mode | 1K mode |
| Equalizer type: | Zero-forcing Equalizer |
| Symbol mapping scheme: | 4-QAM |
| LDPC rate: | ½ |
| MSE requirement: | $\leq 0.018$ |
| SNR assumption: | Constant within one TS (25ms) |
| SNR distribution: | Model 1: Uniform distribution (11dB~20dB) |
| | Model 2: Normal distribution(11dB~20dB) |

In table 5.1, two SNR distributions models are used for simulation: uniform distribution and normal distribution, both ranging from 11dB to 20dB. Such SNR range is chosen based on [36], which takes large amounts of experimental measurements on CMMB signals in real urban environments, and

reports that more than 85% of measurement points have a SNR ranging from 10dB to 20dB. Furthermore, based on the analysis in subsection 4.4.2, for zero-forcing equalizer in our scenario, the minimal SNR to satisfy the BER requirement is 11dB. Therefore, the SNR range used in our simulation scenario ranges between 11dB and 20dB. Since [36] does not report the exact distribution of SNR within such range, two frequently used distributions model: uniform distribution and normal distribution, are in our simulation scenario. They are further shown in Figure 5.2 and Figure 5.3.

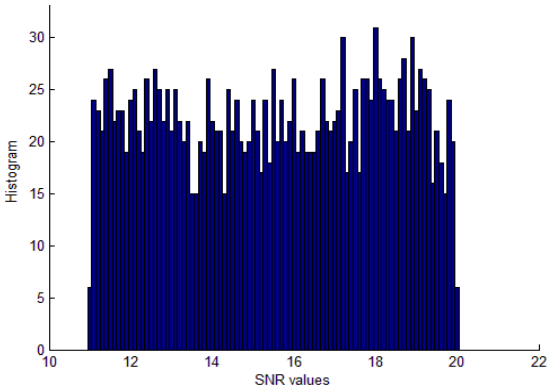


**Figure 5.2 Uniform-distributed SNR**

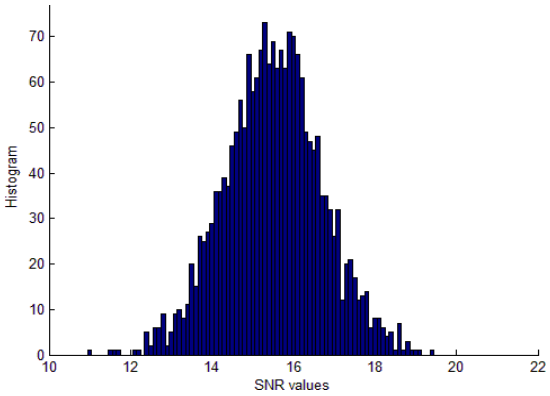


**Figure 5.3 Normal-distributed SNR**

The simulation is performed over a sample set of 2000 CMMB TS, under 4 different configurations for CE algorithms. The configurations of CE algorithms are showed in Table 5.2.

**Table 5.2 Configurations of CE algorithms**

| **Configurations** | **CE algorithms** |
|---|---|
| Config.1 (3 CE algorithms) | (TD-LS)+(FIR-Wiener)+(Linear) |
| Config.2 (2 CE algorithms) | (TD-LS)+(FIR-Wiener) |
| Config.3 (2 CE algorithms) | (TD-LS)+ (Linear) |
| Config.4 (Fixed channel estimator) | (TD-LS) |

In the simulation, two metrics are evaluated: MSE per TS, and normalized average cycles per TS.

First consider the metric “MSE per TS”. For the uniform-distributed SNR, The MSE per TS in 4 different configurations is shown in from Figure 5.4 to Figure 5.7; While for the normal-distributed SNR, the MSE per TS in 4 different configurations is shown in from Figure 5.8 to Figure 5.11. We see that in all configurations, the MSE in every TS is always lower than 0.018, meaning that the MSE requirement is always fulfilled.

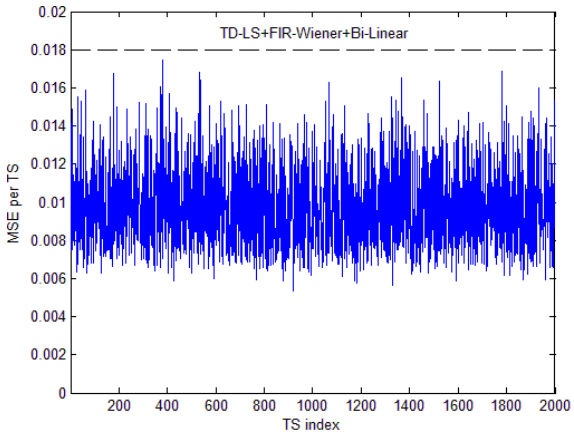


**Figure 5.4 Config.1 (Uniform-distributed SNR)**

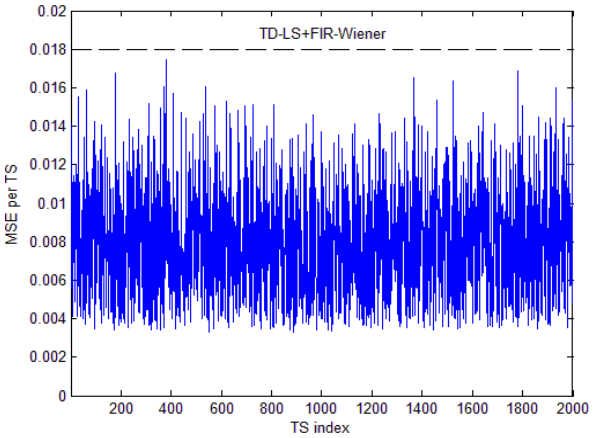


**Figure 5.5 Config.2 (Uniform-distributed SNR)**

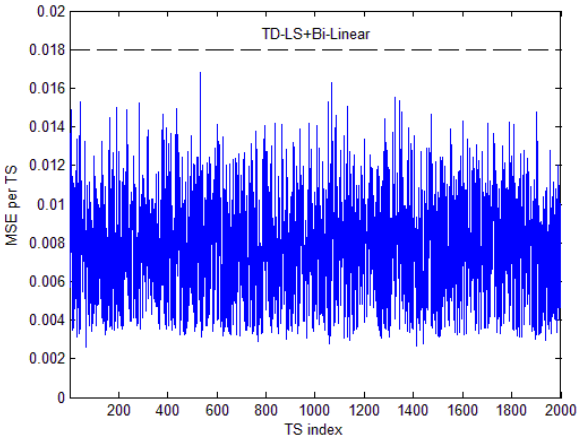


**Figure 5.6 Config.3 (Uniform-distributed SNR)**

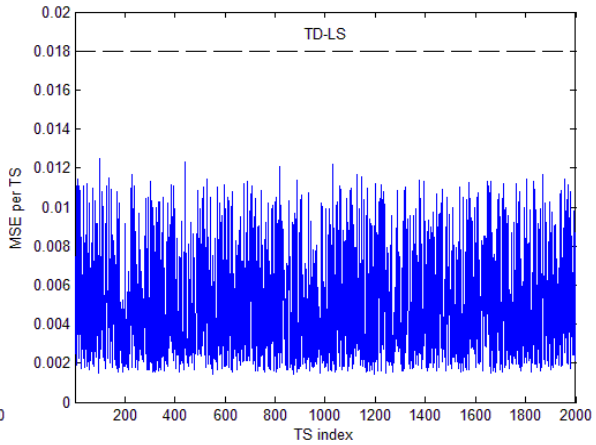


**Figure 5.7 Config.4 (Uniform-distributed SNR)**

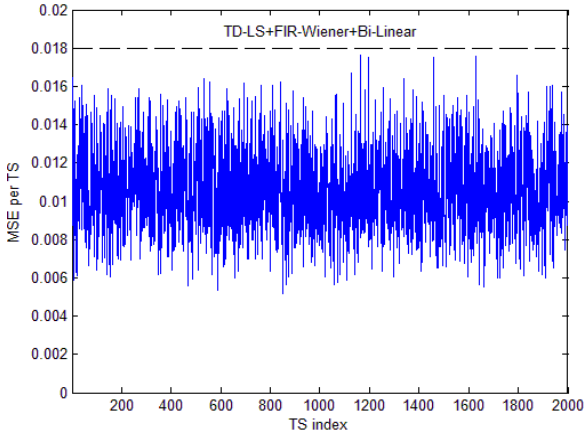


**Figure 5.8 Config.1 (Normal-distributed SNR)**

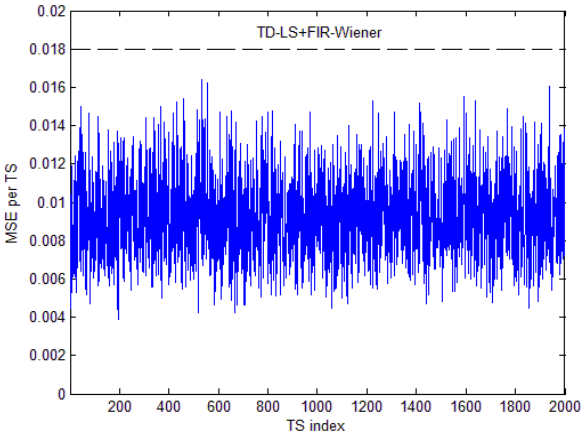


**Figure 5.9 Config.2 (Normal-distributed SNR)**

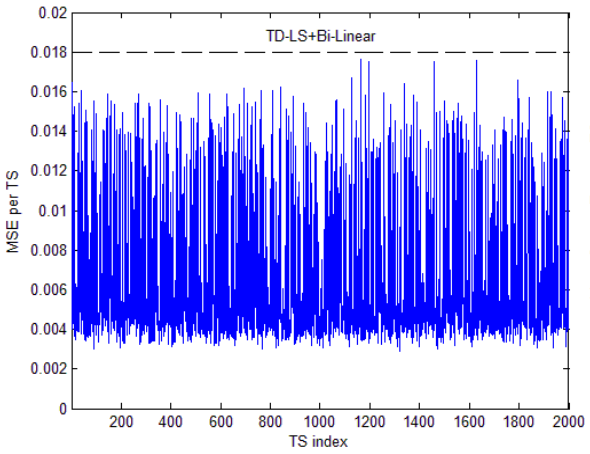


**Figure 5.10 Config.3 (Normal-distributed SNR)**

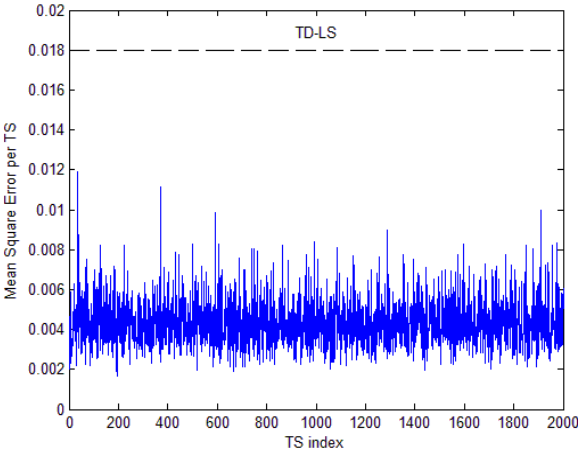


**Figure 5.11 Config.4 (Normal-distributed SNR)**

Then consider the metric "normalized average cycles per TS". For the uniform-distributed SNR, the normalized average cycles per TS in 4 configurations are shown together in Figure 5.12; While for the normal-distributed SNR, the normalized average cycles per TS under the 4 configurations are shown together in Figure 5.13; In Figure 5.12 and Figure 5.13, each normalized average cycle number per TS in every configuration is represented by a column, and the different colour proportions in a column reflects the cycle contribution of each algorithm modules, including the computation overheads. We see in Figure 5.12 and Figure 5.13 that, under the uniform-distributed SNR model, the simulation shows that, comparing with a fixed channel estimator (TD-LS), the maximal run-time cycle reduction factor is 55.58% (100%-44.42%=55.58%); under the normal-distributed SNR, the simulation shows that, comparing with a fixed channel estimator (TD-LS), the maximal run-time cycle reduction factor is 65.06% (100%-34.94%=65.06%). The cycle reduction factor under the normal-distributed SNR model is higher because the FIR-Wiener is more frequently selected, and because the load difference between the TD-LS estimator and the FIR-Wiener estimator is larger than that between the FIR-Wiener estimator and the linear estimator.

Therefore, we can conclude that under the pre-defined simulation scenario, more than 50% of run-time load reduction can be achieved comparing with a fixed channel estimator (the TD-LS estimator). Since the power consumption is proportional with the number of run-time cycles per TS, we can further conclude that: under the pre-defined simulation scenario, more than 50% of power saving can be achieved comparing with a fixed channel estimator (the TD-LS estimator).

Additionally, the power saving for the VDSP inner-receiver can be estimated. [6] reports that channel estimation (18-tap FIR-Wiener) takes about 50% of total processing load in the inner-receiver. Although 18-tap FIR-Wiener is not mapped onto VDSP, its processing load in VDSP can be estimated to be 3 times that of the 6-tap FIR Wiener because the filtering length is 3 times longer. Meanwhile, as shown in Figure 5.1, we can find that the processing load of the TD-LS estimator is also 3 times of 6-tap FIR Wiener. Therefore, we can make the assumption that the TD-LS estimator takes about 50% total processing load in VDSP. Based on this assumption, we can then estimate that our run-time complexity adaptive channel estimator can achieve about 25% (50%×50%=25%) of power saving in the VDSP inner-receiver.

Finally, the power saving for the total CMMB receiver platform can be estimated. By using power distribution diagram in Figure 1.7, we can estimate that our run-time complexity adaptive channel estimator can achieve about 8% (50%×16%=8%) power saving for the total CMMB receiver.

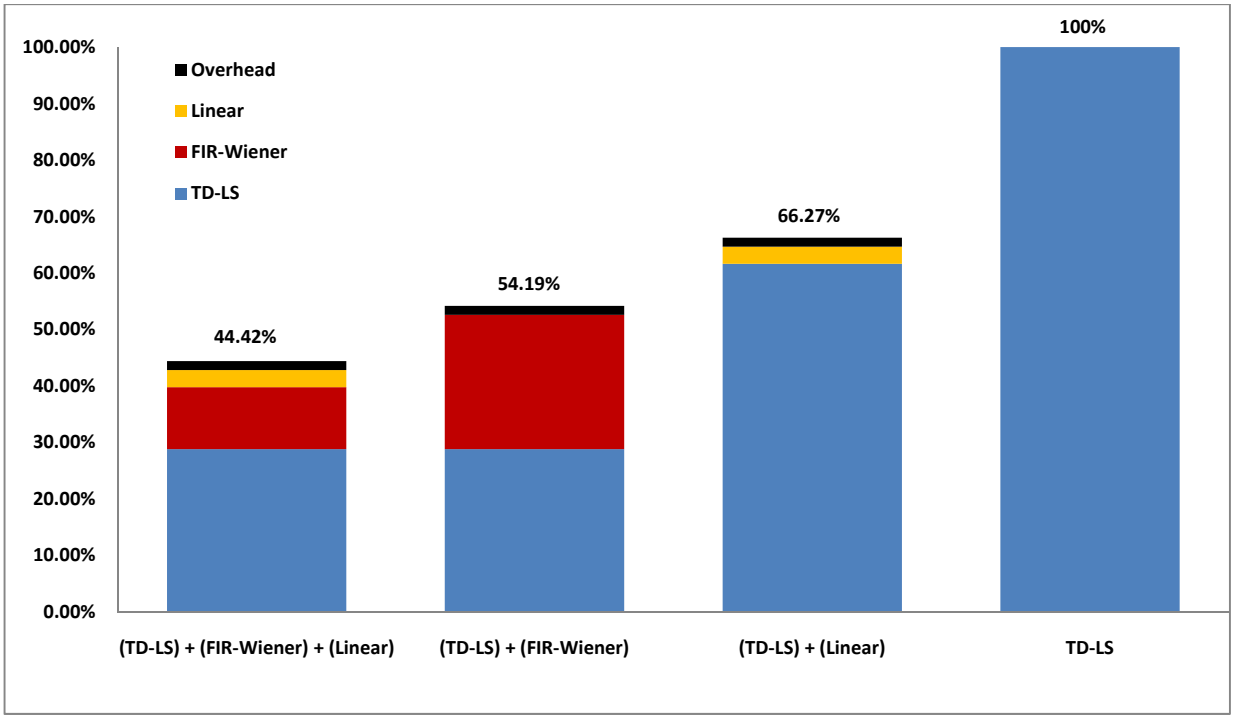


**Figure 5.12 Normalized average cycle number per TS (Uniform-distributed SNR)**

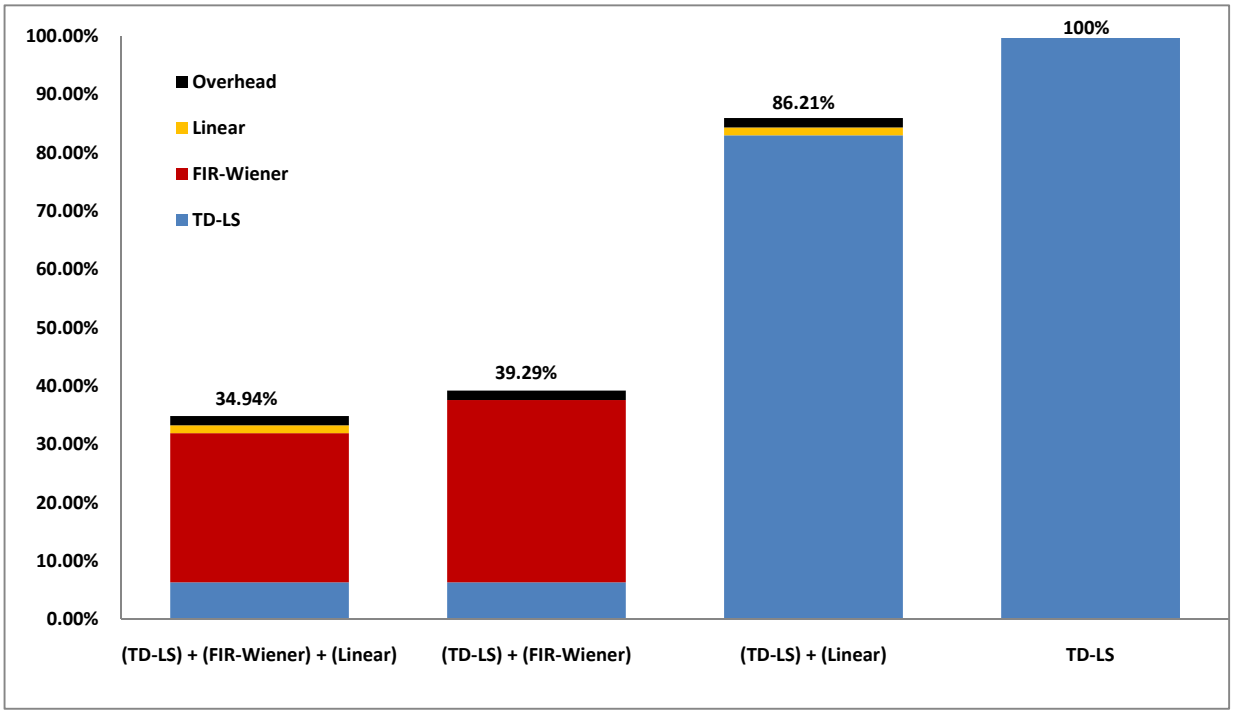


**Figure 5.13 Normalized average cycle number per TS (Normal-distributed SNR)**

## 5.3. Chapter Conclusion

This chapter presents the load profile of channel estimation algorithms and channel quality estimation algorithms in VDSP, and presents the performance of our run-time complexity adaptive channel estimator by integrated simulation. The load profile is obtained by running C code in the VDSP simulator, and is further inserted into the Matlab test bench for integrated simulation. The integrated simulation is performed under a pre-defined scenario, with two alternative SNR distribution models. The simulation shows that, under both SNR distribution models, the MSE requirement can always be fulfilled. The simulation also shows that, more than 50% of run-time load reduction can be achieved comparing with a fixed channel estimator. Consequently, it means more than 50% of power saving can be achieved comparing with a fixed channel estimator. Also, it means approximately 25% of power saving can be achieved in the VDSP inner-receiver. And it further means approximately 8% of power saving can be achieved in the total CMMB receiver platform.

# 6. Conclusion and Further Work

In this chapter, the final conclusion is made and further work for improvements is described.

## 6.1. Conclusion

This report presents a run-time complexity adaptive channel estimator for CMMB. The idea is to run-time scale the processing load of the channel estimator according to the run-time estimated channel quality parameters, in the sense of run-time load reduction and power saving. Computer simulation shows that, under a pre-defined scenario which is consist with CMMB standard, the run-time load can be reduced by more than 50% comparing with a fixed channel estimator, while still fulfilling the MSE requirement. This results in a longer battery life in high quality channels while maintaining satisfactory reception quality in poor quality channels.

The main contributions of this thesis are as follows:

1) Originally proposed the concept of run-time complexity adaptive channel estimator in the feedforward architecture.
2) Developed a set of practical channel estimation algorithms for CMMB, which are feasible to be mapped in VDSP. In particular, make the mathematical derivation of the mismatched Wiener estimator in FIR filtering form using CMMB pilot pattern; and solve the DSP mapping problem for the time domain LS estimator using TSVD.
3) Proposed a novel noise variance estimation algorithm and a novel delay spread estimation algorithm using CMMB preamble, which outperform existing algorithms.
4) Theoretically derived a pessimistic MSE requirement for the CMMB channel estimator.
5) Evaluated the performance of the run-time complexity adaptive channel estimator by computer simulation. Under a pre-defined scenario, the simulation shows more than 50% of run-time cycle reduction can be achieved comparing with a fixed channel estimator, while at the same time fulfilling the MSE requirement.

## 6.2. Further Work

Further research work for improvements can be as follows:

1) Use the channel model with time-variant delay spread for the integrated simulation, and improve the power manager: Although delay spread estimation algorithm is developed, in current design phase the power manager only uses SNR thresholds as a criterion to scale the load of the channel estimator. Applying both SNR and delay spread as selection criterions can be done by finding the thresholds according to the MSE requirement, in a 3D chart (MSE, SNR and delay spread).

2) Explore the fine-gained scalability: current complexity scalability is explored by choosing a different CE algorithm, however, fine-gained scalability within a CE algorithm can also be explored. Typically for FIR-Wiener estimator, the estimation accuracy can be scaled by scaling the FIR filtering length, or by skipping/keeping the SP smoothing process. By exploring the fine-gained scalability, higher load reduction factor is expected.
3) Explore the scalability together with a channel equalizer: channel equalization is another computation intensive task in an OFDM inner-receiver. By joint designing a scalable channel estimator and a scalable channel equalizer, higher load reduction factor is expected.
4) Implement the run-time complexity adaptive channel estimator on the real platform, and combine the method with dynamic frequency scaling (DFS) and dynamic voltage scaling (DVS): Instead of doing simulation in Matlab under a pre-defined scenario, the run-time complexity adaptive channel estimator can be implemented and evaluated in the real platform. The channel parameter thresholds in the power manager can be experimentally derived using the approach introduced in subsection 4.4.1.

   For power consumption, the active power in VDSP can be expressed in the following form [37]:

$$P_{active} = C_{switch} \cdot V_{dd}^2 \cdot f_c \qquad (6\text{-}1)$$

   Where $V_{dd}$ is the supply voltage for VDSP, $C_{switch}$ is a constant switching capacitance which is determined by the hardware circuits, and $f_c$ is the operation clock frequency of VDSP.

   When implemented in VDSP, the run-time complexity adaptive channel estimator can in the run-time scale the VDSP load per CMMB time slot. If the run-time load is reduced, the operation clock frequency can be dynamically reduced in the same factor, which is called dynamic frequency scaling (DFS). Using (6-1), we see that the active power is therefore reduced in the same factor.

   In addition, a very interesting thing is that, benefit from the decreasing of operation clock frequency $f_c$, it's further possible to perform dynamical voltage scaling (DVS) [38], so that the supply voltage $V_{dd}$ can be dynamically reduced, resulting in a further more power reduction!

# 7. References


[1] GYIT 220.1-2006. “China Mobile Multimedia Broadcasting Standard, Part I: Framing Structure, Channel Coding and Modulation for Broadcasting Channel”, *SARFT,* pp.1-20, Oct. 2006.

[2] Z. Wang and G.B. Giannakis, “Wireless Multicarrier Communications: Where Fourier Meets Shannon,” *in IEEE Signal Processing Magazine*, vol. 17, pp. 29–48, May 2000.

[3] H. Lu. “A Synchronization Scheme for China Multimedia Mobile Broadcasting standard: CMMB”, *Master thesis in Eindhoven University of Technology*, Aug. 2010.

[4] Tensilica Inc., “A Second-Generation High-performance DSP Engine”, http://www.tensilica.com/uploads/pdf/epf_2004_DSP.pdf, May, 2004.

[5] Berkeley Design Technology, Inc. “An Independent Analysis of the Tensilica Xtensa LX Processor with Vectra LX”, http://www.tensilica.com/uploads/pdf/BDTI-tensilica_wp-1.pdf, 2005.

[6] S.Sun and Z.Cao, “Feasibility Analysis of Key Modules in STiMi Receiver on EVP”, *Technical Note PR-TN 2008/00300, Internal Report of Philips Research East Asia*, pp.13-13, June 2008.

[7] ETSI-EN-300-744, “Digital Video Broadcasting (DVB): Frame Structure, Channel Coding and Modulation for Digital Terrestrial Television (DVB-T),” *ETSI, Tech. Report*, Nov. 2004.

[8] T. Cooklev, “Wireless Communication Standards, A Study of 802.11, 802.15, and 802.16”, *in IEEE Press*, 2004.

[9] X.G. Doukopoulos and R. Legouable. “Robust Channel Estimation via FFT Interpolation for Multicarrier Systems”, *in Proc. IEEE 65$^{th}$ Vehicular Technology Conference 2007,* pp.1861-1865, April 2007.

[10] A. Goldsmith, “Wireless Communications”, *Cambridge University Press*, 2005.

[11] Commission of the European Communities, "Digital Land Mobile Radio Communications: COST-207 Final Report", Ch. 2, 1988.

[12] J.J. van de Beek, O. Edfors, M. Sandell, S.K. Wilson and P.O. Börjesson, "On Channel Estimation in OFDM Systems", *in Proc. IEEE Vehic. Technol. Conf.*, vol 2, pp.815-819, Chicago, July 1995.

[13] A. Oppenheim, R. Schafer and J. Buck, "Discrete-time Signal Processing", Prentice Hall, New Jersey, 2nd Edition, 1999.

[14] A. Filippi, S.A. Husen and S. Baggen, "Automatic Switch between Static and Mobile Operation Modes in DVB-T/H Receivers", *in Proc. IEEE 63rd Vehicular Technology Conference,* May, 2006.

[15] T. Yucek, R.M.A. Tannious and H. Arslan," Doppler Spread Estimation for Wireless OFDM Systems", *in IEEE Sarnoff Symposium Advances in Wired and Wireless Communication*, April. 2005.

[16] Q. Lu, W. Chen and B. Long, "A Doppler Spread Estimator Design for Mobile OFDM Systems", *in IEEE International Conference on Computational Science,* Nov. 2006.

[17] P. Hoeher, S. Kaisa and P. Robertsan, "Two-dimensional Pilot-Symbol-Aided Channel Estimation by Wiener Filtering", *in Proc of ICASSP*, pp1845-1848, April 1997.

[18] P. Hoeher, S. Kaisa and P. Robertsan, "Pilot-Symbol-Aided Channel Estimation in Time and Frequency", *in Proc. Communication Theory. Mini-Conference (CTMC) within IEEE Global Telecommunications. Conference (GLOBECOM '97), Phoenix USA,* pp.90-96, 1997.

[19] Y. Zhao and A.Huang. "A Novel Channel Estimation Method for OFDM Mobile Communication Systems based on Pilot Signals and Transform domain processing", *in Proc. IEEE 47th Vehicular Technology Conference,* pp.2089-209, May 1997.

[20] P. C. Hansen, "The Truncated SVD as a Method for Regularization". *BIT: vol 27, pp.534–553,* 1987.

[21] Jones, D. "Radix-4 FFT Algorithms", www.cnx.org

[22] D. Pauluzzi and N. Beaulieu, "A Comparison of SNR Estimation Techniques for the AWGN Channel," *in IEEE Trans. Commun*., vol.48, pp. 1681–1691, Oct 2000.

[23] T. Yücek and H. Arslan, "MMSE Noise Power and SNR Estimation for OFDM Systems," *in IEEE Sarnoff Symposium*, Mar. 2006.

[24] S. Boumard, “Novel Noise Variance and SNR Estimation Algorithm for Wireless MIMO OFDM Systems*”, in Proc. of IEEE Global Telecommunications Conference*, vol.3, pp.1330–1334, Dec. 2003.

[25] T. Cui and C. Tellambura, “Power Delay Profile and Noise Variance Estimation for OFDM”, *in IEEE Commun. Lett.*, Jan. 2006.

[26] G. Ren, H. Zhang and Y. Chang, “SNR Estimation Algorithm based on the Preamble of OFDM Systems in Frequency Selective Channels”, *in IEEE Trans. Commun,* vol.57, Aug. 2009.

[27] Y. Wang, L. Li, P. Zhang, Z. Liu and M. Zhou. “A New Noise Variance Estimation Algorithm for Multiuser OFDM Systems”, *in IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, Dec. 2007.

[28] C.R.N. Athaudage and A.D.S. Jayalath, “Delay-spread Estimation using Cyclic-prefix in Wireless OFDM Systems”, *in Proc. IEEE, Communications. Vol.151,* Nov. 2004.

[29] H. Arslan and T. Yucek, “Delay Spread Estimation for Wireless Communication Systems”, *in IEEE Symposium on Computers and Communications*, pp.282-287, Jun. 2003.

[30] M. Xue, M. Se, B. Hyung and J. Choi , “Improved Channel Estimation Method based on Time-Domain Processing for OFDM Systems”, *in IEEE 7$^{th}$ international conference on Wireless Communications*, *Networking and Mobile Computing* (*WiCOM 2007*), Sept. 2007.

[31] Y. Zhao and A.Huang, “A Novel Channel Estimation Method for OFDM Mobile Communication Systems based on Pilot signals and Transform-domain processing”, *in Proc. IEEE 47$^{th}$ Vehicular Technology Conference 1997,* pp.2089-2903, May 1997.

[32] Continuous-bits Ltd., “ASIC IP-core for Decoding CMMB’s LDPC Codes”, http://www.continuousbits.com/CMMBdecoderDataBrief.pdf.

[33] K. Zhang, X. Huang and Z. Wang, “A Dual-rate LDPC Decoder for China Multimedia Mobile Broadcasting Systems”, *in IEEE Trans. Consumer Electronics*, vol.6, May 2010.

[34] G.D. Forney, “Concatenated Codes”, *M.I.T. Press*, Cambridge, MA, USA, 1966.

[35] H. Cheon and D. Hong, ”Effect of Channel Estimation Error in OFDM-based WLAN”, *in IEEE Communications Letters*, vol.6, pp.190 – 192, May 2002.

[36] M. Dai. ” Wireless Network Test and Analysis for CMMB”, *Publishing House of Electronic industry*, Beijing, June, 2009.

[37] A.P. Chandrakasan, S. Sheng, R.W. Brodersen.“ Low-power CMOS digital design“, *in IEEE Journal of Solid-State Circuits*, pp.473 - 484 Apr. 1992

[38] T. Pering, T. Burd, and R. Brodersen. “Dynamic Voltage Scaling and the Design of a Low-power Microprocessor System”, *in International Symposium on Computer Architecture (ISCA),* 1998.