國 立 交 通 大 學

電信工程學系碩士班

碩士論文

應用於 MIMO-OFDM 系統之前置編碼搜尋與時域通道資訊回傳

Precoder search and time domain CSI feedback in MIMO-OFDM systems

研 究 生：徐子瀚

指導教授：吳 文 榕 博士

中華民國九十七年八月

應用於 MIMO-OFDM 系統之前置編碼搜尋與時域通道資訊回傳

Precoder search and time domain CSI feedback in MIMO-OFDM systems

研 究 生：徐子瀚　　Student：Tzu-Han Hsu

指導教授：吳文榕 博士　　Advisor：Dr. Wen-Rong Wu

國 立 交 通 大 學

電信工程學系碩士班

碩 士 論 文

A Thesis
Submitted to Department of Communication Engineering
College of Electrical and Computer Engineering
National Chiao-Tung University
in Partial Fulfillment of the Requirements
for the Degree of
Master of Science
In
Communication Engineering
August 2008
Hsinchu, Taiwan, Republic of China

中華民國九十七年八月

應用於 MIMO-OFDM 系統之前置編碼搜尋與時域通道資訊回傳

Precoder search and time domain CSI feedback in MIMO-OFDM systems

研究生：徐子瀚　　　　　　　　指導教授：吳文榕 教授

國立交通大學電信工程學系碩士班

摘要

前置編碼(Precoding)，為一項能夠有效提升 MIMO-OFDM 系統傳輸效能之技術。在實際的系統中，前置編碼矩陣會先選定，並且只回傳該選定矩陣之編號。然而，候選之矩陣數量可能很多，因此需要高度計算量來搜尋出最佳的前置編碼矩陣。在本篇論文中，為了解決此問題，我們首先提出一個低複雜度的前置編碼搜尋演算法。與完全搜尋(exhaustive search)比較，此方法可以降低 80%左右的搜尋複雜度，並且其效能損失是在可以接受的範圍之內。此外，在 MIMO-OFDM 的系統中，通道資訊(Channel state information, CSI)常常能起到很大的作用。然而，這些必須從接收端回傳的 CSI，通常需要相當多的資料量。在本篇論文的第二部份，我們提出一個時域通道資訊回傳方法來降低回傳資料量。在某些通道情況下，此方法只需要少量的回傳資料。最後，針對時變的通道環境，我們提出一個機制，將誤差脈衝編碼調變(Differential Pulse Code Modulation, DPCM)應用於時域通道資訊回傳方法。由此，回傳的資料量可以更加地減低。

Precoder search and time domain CSI feedback in

MIMO-OFDM systems

Student: Tzu-Han Hsu　　　　Advisor: Dr. Wen-Rong Wu

Department of Communication Engineering
National Chiao-Tung University

Abstract

Precoding is an effective technique improving the performance of MIMO-OFDM systems. In practical systems, the precoding matrices are pre-determined and only the index of the selected matrix is fed back. Since the number of precoding matrices may be large, the search for the optimum precoder requires high computational complexity. In this thesis, we first propose a low-complexity precoder searching algorithm to solve the problem. Compared to the exhaustive search, the proposed searching method can reduce about 80% searching complexity with acceptable performance loss. Channel state information (CSI) is useful in MIMO-OFDM communication systems. However, the information has to be fed back from the receiver and this will requires a large amount of data. In the second part of the thesis, we propose a time domain CSI feedback scheme to lower the feedback data bits. Under some channel conditions, the proposed method only requires a small amount of feedback data. Finally, for time-varying channels, we propose to use a differential pulse code modulation (DPCM) scheme in our time domain CSI feedback method such that the required feedback data can be further reduced.

誌謝

本篇論文得以順利完成，首先要感謝我的指導教授 吳文榕博士。當我遇到困難與瓶頸時，老師總是耐心地給我意見與指正，讓我能往正確的方向前進。除了老師在專業領域上的淵博知識另我獲益匪淺之外，更重要的，從老師身上學習到的，是一種解決問題的纏鬥精神，與看待研究的嚴謹態度。由此，再次感謝老師兩年來的教導。

此外，也要感謝實驗室的許兆元學長、謝弘道學長、曾凡碩學長與林鈞陶學長，經常不吝給我研究上的寶貴意見。還有同屆的碩士班同學：何軒廷、賴允仁、賴廉承。若沒有你們與我共同努力，互相砥礪，也不會有今天的成果。

當然也感謝我在電信系的好同學們，因為你們，讓我的碩士生活增添許多有趣回憶。也感謝電信系壘的隊友們，讓我在研究之餘，能放鬆心情在球場上盡情揮灑汗水。

最後，誠摯感謝始終在我背後支持我的雙親。謹以此文獻給你們。

Catalog

List of Tables

List of Figures

Chapter 1 Introduction

In recent years, the demand for high data rate wireless communication increases rapidly, e.g., high quality, real-time video and audio data streams. For this purpose, many prospective techniques have been considered to increase channel capacity and link reliability.

Multiple-input multiple output (MIMO) transceivers, created by having multiple antennas at both the transmitter and the receiver, promises high spectral efficiency and high reliability wireless communication links. Different space-time modulation schemes can be chosen to exploit the benefits offered by MIMO channels, such as space-time coding (STC) and spatial multiplexing (SM). Spatial multiplexing (SM) is a simple and practical space-time modulation scheme that allows MIMO wireless systems to obtain high spectral efficiency by dividing single bit stream into multiple substreams sent over different antennas.

Unfortunately, SM is sensitive to the condition of the MIMO channels. When the channel matrix becomes ill-conditioned, the performance of SM becomes poor. In narrowband channels, linear precoding, a technique that pre-multiplying the transmitted data streams by a precoding matrix, chosen based on channel information, is one way to guard against rank deficiencies in the channel and to improve error performance. Optimal precoder under different performance criteria has been proposed in [1]. As mentioned, we need full channel state information (CSI) at transmitter to conduct precoding. However, when the forward and reverse channels are not reciprocal, full CSIT may be may be difficult to obtain due to limited bandwidth of feedback channel. Thus, a codebook-based limited feedback precoding scheme is often used. The main idea is that the receiver only sends the binary index of optimal precoder chosen from a finite set of precoding matrices, called codebook, known to both the receiver and the transmitter. The codeword selection criteria and codebook design criteria are also discussed in [1]. The practical codebook construction method used in [1] is described in [3], using the Fourier-based designs. The work in [4] proposes a method to recursively quantize the precoding matrix with Householder reflection, and the codebook is vector-wise and can be designed using vector quantization (VQ) algorithm.

The precoding technique proposed for narrowband channels can be easily extended to frequency selective channels by using orthogonal frequency division multiplexing (OFDM). The combination of MIMO and OFDM, known as MIMO-OFDM, converts a broadband MIMO channels into a series of parallel narrowband MIMO channels, one for each OFDM subcarrier. The codebook-based limited feedback precoding scheme can be performed independently at each subcarrier. However, in nonreciprocal channels, this requires that the receiver computes and sends the index of optimal precoding matrix for every active subcarrier. Thereby, the feedback data generally grows in proportion to the number of active subcarriers. To solve this problem, some techniques such as clustering or interpolation [5] are developed. These methods exploit the correlation between adjacent subcarriers and feedback the information about precoding matrices for only a fraction of all subcarriers.

There is another problem in precoding; the receiver must search for the best precoder from the codebook for each active subcarrier. Obviously, the searching complexity is proportional to the number of active subcarriers and the number of codewords in a codebook. In general, exhaustive codeword searching will require large computational complexity. In this thesis, we propose a low complexity codeword searching algorithm. For any arbitrary codebook, we first partition the codebook with a distance comparison algorithm, and perform a tree search algorithm to find the desired codeword. The simulation shows that while the performance is only slightly affected, the searching complexity is decreased significantly.

Apart from precoding, many schemes can be applied in transmitters to improve the performance of MIMO-OFDM systems provided the CSI is available. For example, we can conduct bit-loading, resource management, multiuser diversity, and dirty paper coding. Apparently, if CSI for all subcarrier are fed back, the data amount will be very high. Taking the advantage of the sparse nature of wireless channels, we then propose a time domain CSI feedback method. The simulation shows that, under some channel conditions, the time domain CSI feedback method only requires a small amount of data. Even in the application of precoding, our method is comparable to the conventional precoder feedback scheme such as clustering. Finally, for realistic time-varying channel, we propose a differential pulse code modulation (DPCM) scheme in our time- domain CSI feedback method such that the required feedback data can be further reduced.

The rest of the thesis is organized as follows. In Chapter 2, we first review the linear precoding scheme, including 1) system model, 2) criteria for precoding, 3) codebook design and construction, and 4) limited feedback schemes. At the end of Chapter 2, we propose a low complexity codeword searching algorithm. In Chapter 3, we propose a time domain CSI feedback method. For realistic time-varying channel, a modified time domain CSI feedback applying DPCM is also proposed. In Chapter 4, simulation results are reported and analyzed. Finally, Chapter 5 gives some conclusions and potential topics for future works.

Chapter 2 Precoding in MIMO-OFDM systems

2.1 System model

A simplified block diagram for precoding in MIMO-OFDM transceiver is shown below.

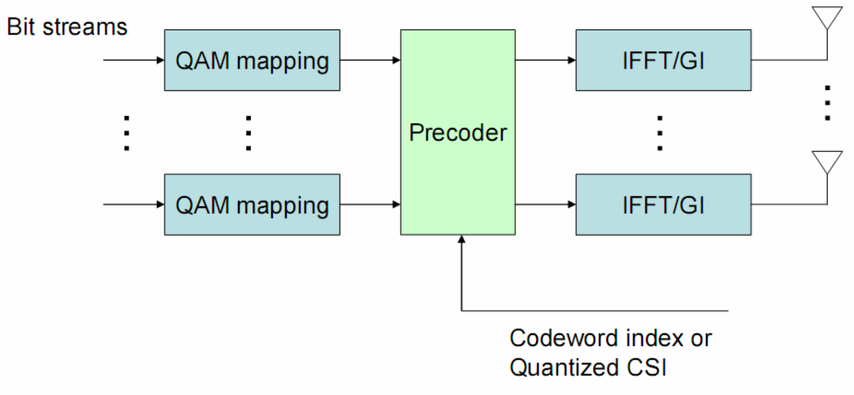

Figure 2-1 MIMO-OFDM Transmitter with Precoding

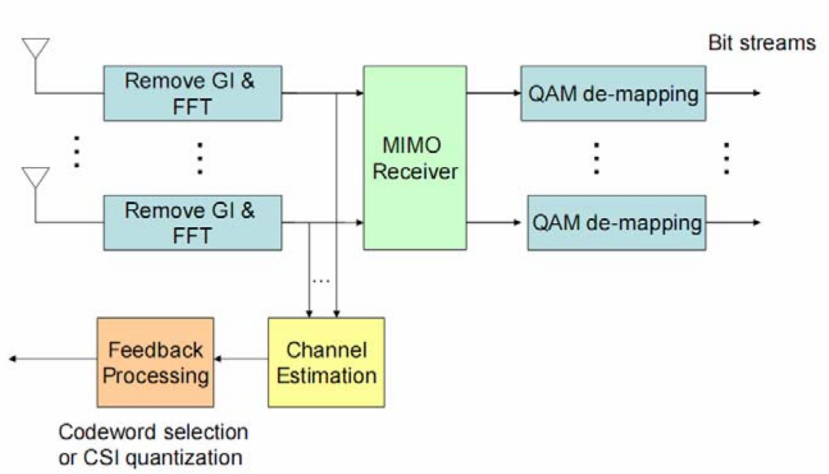

Figure 2-2 MIMO-OFDM Receiver with Precoding

The bit stream to be transmitted is first divided into M different bit streams and sent into QAM mapper. Each of the M bit streams is then modulated independently using the same constellation, e.g. QPSK or 16-QAM. This yields a symbol vector at time k as:

$$\mathbf{s}_k = [s_{k,1}\ s_{k,2} \cdots s_{k,M}]^T \tag{2-1}$$

For convenience, we assume that the M data streams are equally-powered and independent to each other. That is,

$$E[\mathbf{s}_k \mathbf{s}_k^*] = \mathbf{I}_M \tag{2-2}$$

The notation $\mathbf{A}^*$ denotes complex-conjugate of matrix **A.**

The symbol vector $\mathbf{s}_k$ is then multiplied by an $N_t \times M$ precoding matrix **F** (which is chosen as a function of the channel using criteria to be described) producing a length M_t vector $\mathbf{x}_k$

$$\mathbf{x}_k = \sqrt{\frac{E_s}{M}} \cdot \mathbf{F} \cdot \mathbf{s}_k \tag{2-3}$$

where N_t is the number of transmit antennas, and we assume that $N_t > M$. E_s is the total transmit energy at time k. Then the symbol $\mathbf{x}_k$ is sent into the MIMO channel, and the received signal, of dimension $N_r \times 1$, can be written as:

$$\begin{aligned} \mathbf{r}_k &= \mathbf{H} \cdot \mathbf{x}_k + \mathbf{n}_k \\ &= \sqrt{\frac{E_s}{M}} \cdot \mathbf{H} \cdot \mathbf{F} \cdot \mathbf{s}_k + \mathbf{n}_k \end{aligned} \tag{2-4}$$

where **H** is the $N_r \times N_t$ channel matrix and $\mathbf{n}_k$ is the $N_r \times 1$ noise vector. N_r is the number of receive antennas. We assume that the entries of **H** are independent and identically distributed (i.i.d.) and the distribution is *CN* (0,1). Similarly, the entries of $\mathbf{n}_k$ are also i.i.d. and the distribution is *CN* $(0,N_0)$.

Here, assume that the channel matrix **H** can be estimated perfectly, we consider two kinds of MIMO receivers:

1. Maximum likelihood (ML) receiver:

The ML receiver is the optimal MIMO receiver and is a nonlinear receiver. That is, we cannot perform ML decoding with simple matrix operations. The ML receiver solves the optimization problem shown below:

$$\tilde{\mathbf{s}} = \arg\min_{\mathbf{s}\in w^M} \left\| \mathbf{r} - \sqrt{\frac{E_s}{M}}\mathbf{HFs} \right\| \tag{2-5}$$

where $\tilde{\mathbf{s}}$ is the decoded symbol vector, w^M is the multidimensional constellation of QAM mapping. Therefore, ML receiver needs exhaustive search and that results in high computational complexity.

2. Linear receiver (ZF and MMSE):

A Linear receiver applies an $M \times N_r$ matrix **G**, chosen according to some criterion, to produce the decoded symbol vector $\tilde{\mathbf{s}}$,

$$\tilde{\mathbf{s}} = \mathbf{G} \cdot \mathbf{r}_k \tag{2-6}$$

We consider two different criteria**:** zero-forcing (ZF) and minimum mean square error (MMSE). It can be shown that the optimum ZF receiver is

$$\mathbf{G} = (\mathbf{HF})^{+} \tag{2-7}$$

where “+” denotes matrix pseudo-inverse. The ZF receiver is a simple linear receiver, but it has the noise enhancement problem when signal to noise power ratio (SNR) is low. It can also be shown that the optimum MMSE receiver is

$$\mathbf{G} = [\mathbf{F}^*\mathbf{H}^*\mathbf{H}\mathbf{F} + \frac{MN_0}{E_s}\mathbf{I}_M]^{-1}\mathbf{F}^*\mathbf{H}^* \tag{2-8}$$

The MMSE receiver considers both the noise and the channel effect, and has better performance. However, its computational complexity is higher. Note that, if we set $\sigma^2 = 0$, then the MMSE receiver will become a ZF receiver.

Once the receiver has estimated the channel matrix **H**, it needs to feed back the information to the transmitter such that a precoding matrix can be determined. Two different feedback schemes can be chosen:

1. Directly feedback the quantized channel state information (CSI)

We can feedback the quantized coefficients of channel response either on frequency domain or time domain. The transmitter can then calculate the precoder **F** with some performance criteria. However, when the number of active subcarriers in the OFDM system is large, it may not be practical to feed back the frequency domain channel response of each active subcarrier. Nevertheless, under the condition that the time domain channel only consists of few taps (i.e., sparse channels), the time domain CSI feedback may be an efficient and feasible way. The time domain feedback CSI method will be described in chapter 3.

2. Feedback the quantized optimal precoder **F**

Since only the information about optimal precoder is necessary for the transmitter to perform precoding, a reasonable solution is to quantize the optimal precoder $\mathbf{F}_{\text{opt}}$ rather than the full channel matrix $\mathbf{H}$. Thus, a codebook-based limited feedback scheme is often used. The idea is that the receiver only sends the index of optimal precoder chosen from a finite set of precoding matrices, called codebook, known to both the receiver and the transmitter. Although the data amount for feedback is also proportional to the number of active subcarriers, some techniques such as clustering or interpolation can be applied to reduce the total feedback data.

We have briefly described the system model of MIMO-OFDM precoding system at Section 2.1, but we haven't answered the question that what is the *selection criterion* for optimal precoder. We will do that at Section 2.2.

2.2 Criteria for precoding

In this section, we discuss the criteria used for choosing the optimal precoding matrix from a given codebook. We will outline the criteria based on the ML receiver, or linear receivers such as ZF and MMSE. The criterion for mutual information maximization is also included. Using these precoder selection criteria, we will show that the optimal un-quantized precoding matrix $\mathbf{F}_{\text{opt}}$ for linear receiver is just the *first M columns of right singular matrix of* $\boldsymbol{H}$.

1. ML receiver

Equation (2-5) indicates the criterion for the ML receiver to optimize. Note that for a given channel matrix, the ML receiver will give the minimum-error-rate. Thus, we have to find an optimal precoder yielding the minimum the error rate. However, a closed-form expression of the probability of symbol vector error is difficult to derive. One approach is to use the property that the probability of the symbol vector error can be upper bounded when the SNR is high. The bound, called the vector union bound, is solely a function of the receive minimum distance $d_{min,R}$ of the multidimensional constellation w^M, which is given by

$$\begin{aligned} d_{\min,R} &= \min_{\mathbf{s}_1,\mathbf{s}_2 \in w^M : \mathbf{s}_1 \neq \mathbf{s}_2} \|\mathbf{r}_1 - \mathbf{r}_2\| \\ &= \min_{\mathbf{s}_1,\mathbf{s}_2 \in w^M : \mathbf{s}_1 \neq \mathbf{s}_2} \sqrt{\frac{E_s}{M}} \|\mathbf{HF}(\mathbf{s}_1 - \mathbf{s}_2)\| \end{aligned} \tag{2-9}$$

Thus, the precoder selection criterion for ML receiver can be approximated by picking **F** from the codebook ***C*** using (2-9),

ML Selection Criterion (ML-SC): Pick **F** such that

$$\mathbf{F} = \arg\max_{\mathbf{F}_i \in \mathbf{C}} d_{\min,R} \tag{2-10}$$

2. Linear receiver

- Zero forcing (ZF)

Equation (2-7) indicates the ZF linear receiver, and we want to find an optimal precoder to minimize the error rate. It was shown in [6] that the SNR of the k_{th} substream for ZF receiver is given by

$$SNR_k = \frac{E_s}{MN_0 \cdot [\mathbf{F}^* \mathbf{H}^* \mathbf{HF}]^{-1}_{k,k}} \tag{2-11}$$

where $\mathbf{A}^{-1}_{k,k}$ is entry (k,k) of $\mathbf{A}^{-1}$. In [6], it is shown that in order to minimize a bound on the average probability of symbol vector error, the minimum substream SNR must be maximized. However, the substream SNR is often difficult to estimate.

For this reason, [6] shows that the minimum SNR for ZF receiver is bounded as

$$\begin{aligned} SNR_{\min} &= \min_{1 \le k \le M} SNR_k \\ &\ge \lambda_{\min}^2\{\mathbf{HF}\} \cdot \frac{E_s}{MN_0} \end{aligned} \tag{2-12}$$

where $\lambda_{\min}\{\mathbf{HF}\}$ is the minimum singular value of **HF.** Therefore, from (2-12), the precoder selection criterion for ZF receiver can be approximated by picking **F** from the codebook ***C*** maximizing the minimum singular value of **HF**.

Minimum Singular Value Selection Criterion (MSV-SC): Pick **F** such that

$$\mathbf{F} = \arg\max_{\mathbf{F}_i \in \mathbf{C}} \lambda_{\min}\{\mathbf{HF}_i\} \tag{2-13}$$

- Minimum mean square error (MMSE)

Equation (2-8) indicates the MMSE linear receiver, and we want to find an optimal precoder to minimize the mean square error (MSE). The MSE matrix can be expressed as follows:

$$\overline{\mathbf{MSE}}(\mathbf{F}) = (\mathbf{I}_M + \frac{E_s}{MN_0}\mathbf{F}^*\mathbf{H}^*\mathbf{HF})^{-1} \tag{2-14}$$

Therefore, the precoder selection criterion for MMSE receiver can be described as picking **F** from the codebook ***C*** to maximize the trace or determinant of MSE matrix.

Mean Squared Error Selection Criterion (MSE-SC): Pick **F** such that

$$\mathbf{F} = \arg\min_{\mathbf{F}_i \in \mathbf{C}} trace(\overline{\mathbf{MSE}}(\mathbf{F}i)) \tag{2-15}$$

3. Capacity criterion

The mutual information for an uncorrelated complex Gaussian source, with a channel matrix **H,** and a precoder matrix **F,** can be expressed as

$$Capacity(\mathbf{F}) = \log_2 \det(\mathbf{I}_M + \frac{E_s}{MN_0}\mathbf{F}^*\mathbf{H}^*\mathbf{H}\mathbf{F}) \qquad (2\text{-}16)$$

With this capacity expression, we can state the capacity inspired precoder selection criterion as picking **F** from the codebook ***C*** to maximize the mutual information.

Capacity Selection Criterion (Capacity-SC): Pick **F** such that

$$\mathbf{F} = \arg\max_{\mathbf{F}_i \in \mathbf{C}} Capacity(\mathbf{F}_i) \qquad (2\text{-}17)$$

Several important results have been proved in [1] and [2],

(a) The optimal un-quantized precoding matrix $\mathbf{F}_{\mathrm{opt}}$ has unit-norm column vectors that are orthogonal to each other. That is, $\mathbf{F}_{opt} \subset U(N_t, M)$, where $U(N_t, M)$ denotes the set of $N_t \times M$ matrices with orthonormal columns. Therefore, when designing the codebook, each codeword $\mathbf{F}_i$ in the codebook must also be contained within $U(N_t, M)$. In other words, codebook design is just the quantization to the set $U(N_t, M)$. We will discuss codebook design issues in Section 2.3

Let the singular value decomposition (SVD) of channel matrix **H** be given by

$$\mathbf{H} = \mathbf{V_L}\mathbf{\Sigma}\mathbf{V_R}^* \qquad (2\text{-}18)$$

where $\mathbf{V_L} \subset U(N_t, N_t)$ is called left singular matrix, $\mathbf{V_R} \subset U(N_r, N_r)$ is called right singular matrix, $\mathbf{V_L}$ and $\mathbf{V_R}$ are both unitary matrices (i.e., $\mathbf{V_L^*} = \mathbf{V_L^{-1}}$, $\mathbf{V_R^*} = \mathbf{V_R^{-1}}$), and $\mathbf{\Sigma}$ is an $N_r \times N_t$ diagonal matrix with $\lambda_k\{\mathbf{H}\}$ denoting the k_{th} largest singular value of **H.**

(b) The optimal precoder over $U(N_t, M)$ for **MSV-SC**, **MSE-SC**, and **Capacity-SC** is

$$\mathbf{F}_{opt} = \overline{\mathbf{V}}_\mathbf{R} \qquad (2\text{-}19)$$

where $\overline{\mathbf{V}}_\mathbf{R}$ is the first M columns of the right singular matrix $\mathbf{V_R}$.

Equation (2-19) is a mathematical result, and we can look at this result from a more intuitive way. With SVD, we can easily see that the MIMO channel matrix can be decomposed to R equivalent parallel SISO channel, where R is the rank of the channel matrix and also the number of nonzero singular values. The diagonal matrix $\mathbf{\Sigma}$ indicates the gain of each parallel SISO channel, e.g. $\lambda_k\{\mathbf{H}\}$ denotes the power gain of k_{th} parallel SISO channel. If we want to send M data streams over MIMO channel **H**, where M≦R. Undoubtedly, the best strategy is to choose M strongest equivalent parallel SISO channels for transmission. From (2-19), we see that the optimal precoder is the first M columns of the right singular matrix $\mathbf{V_R}$, and it just corresponds to the M largest singular values and also the M strongest equivalent parallel SISO channels.

Equation (2-18) and (2-19) are important results, and they can be used to calculate the optimal precoding matrix (with known channel matrix **H**). They also give some hints to design a good codebook (or we can say, how to well quantize the optimal precoder $\mathbf{F}_{opt}$ with finite codewords). In the next section, we will describe the codebook design criteria and practical codebook construction method.

2.3 Codebook design and construction

2.3.1 Codebook design criteria

Before stating the codebook design criteria, we present some relevant background about finite sets in $U(N_t, M)$. The set $U(N_t, M)$ defines the *complex Stiefel manifold* [7] of real dimension $2N_tM - M^2$. Each matrix in $U(N_t, M)$ represents an M-dimensional subspace of N_t-dimensional complex vector space. The set of all M-dimensional subspaces spanned by matrices in $U(N_t, M)$ is the *complex Grassmann manifold*, denoted as $g(N_t, M)$.

What is the difference between $U(N_t, M)$ and $g(N_t, M)$? We answer this question with non-uniqueness of the optimal precoding matrix $\mathbf{F}_{opt}$. From (2-19), we know that optimal precoder consists of first M columns of the right singular matrix $\mathbf{V_R}$. However, if $\mathbf{F}_{opt}$ is multiplied by any M × M unitary matrix $\mathbf{U}_n$, that is, $\mathbf{F}' = \mathbf{F}_{opt}\mathbf{U}_n$, $\mathbf{F}'$ will also be an optimal precoding matrix. For example, if $\mathbf{F}_{opt}$ maximizes $\lambda_{min}\{\mathbf{HF}\}$, then so does $\mathbf{F}_{opt}\mathbf{U}_n$ for any M × M unitary matrix $\mathbf{U}_n$. Same result can be obtained for the MSE-SC and Capacity-SC. Therefore, the optimal precoding matrix $\mathbf{F}_{opt}$ is not unique. If we refer a matrix $\mathbf{X}$ in the set $U(N_t, M)$ as optimal precoding matrix, then $\mathbf{X}\mathbf{U}_n$ for any M × M unitary matrix $\mathbf{U}_n$ can be referred as a M-dimensional subspace spanned by the matrix $\mathbf{X}$. All the M-dimensional subspaces spanned by matrices in $U(N_t, M)$ is denoted as $g(N_t, M)$. Therefore, $U(N_t, M)$ is the set of matrices, and $g(N_t, M)$ is the set of subspaces.

Our codebook $\boldsymbol{C}$, which consists of a finite number of matrices chosen from $U(N_t, M)$, thus represents a set, or packing, of subspaces in the Grassmann manifold $g(N_t, M)$. Determination of the set of L matrices that maximize the minimum subspace distance (where distance can be chosen as a number of different ways) is known as *Grassmannian subspace packing* [8], [9]. First, we give three defined distances between two subspaces $\mathbf{F}_1$ and $\mathbf{F}_2$.

1. Chordal distance:

$$\begin{aligned} d_{chord}(\mathbf{F}_1, \mathbf{F}_2) &= \frac{1}{\sqrt{2}} \left\| \mathbf{F}_1\mathbf{F}_1^* - \mathbf{F}_2\mathbf{F}_2^* \right\|_F \\ &= \sqrt{M - \sum_{i=1}^{M} \lambda_i^2 \{\mathbf{F}_1^*\mathbf{F}_2\}} \end{aligned} \tag{2-20}$$

where $\| \ \|_F$ denotes the matrix Frobenius norm

2. Projection two-norm distance:

$$\begin{aligned} d_{proj}(\mathbf{F}_1, \mathbf{F}_2) &= \left\| \mathbf{F}_1\mathbf{F}_1^* - \mathbf{F}_2\mathbf{F}_2^* \right\|_2 \\ &= \sqrt{1 - \lambda_{\min}^2 \{\mathbf{F}_1^*\mathbf{F}_2\}} \end{aligned} \tag{2-21}$$

where $\| \ \|_2$ denotes the matrix two-norm

3. Fubini-Study distance:

$$d_{FS}(\mathbf{F}_1, \mathbf{F}_2) = \arccos \left| \det(\mathbf{F}_1^*\mathbf{F}_2) \right| \tag{2-22}$$

Details about these definitions and implications can be found in [8]. In [1], the author derives the codebook design criteria under different precoder selection criteria. The codebook design criteria are given below:

- Criterion 1: If ML-SC, MSV-SC, or MSE-SC with trace is used, the codebook should be designed such that the minimum projection two-norm distance between codewords is maximized. It can be expressed as follows:

$$\textit{Maximize} \ \min_{\mathbf{F}_i \neq \mathbf{F}_j} \left\| \mathbf{F}_i \mathbf{F}_i^* - \mathbf{F}_j \mathbf{F}_j^* \right\|_2 \tag{2-23}$$

- Criterion 2: If Capacity-SC or MSE-SC with determinant is used, the codebook should be designed such that the minimum Fubini-Study distance between codewords is maximized. It can be expressed as follows:

$$\textit{Maximize} \ \min_{\mathbf{F}_i \neq \mathbf{F}_j} \left| \det(\mathbf{F}_i^* \mathbf{F}_j) \right| \tag{2-24}$$

However, finding good packings in the Grassmann manifold for arbitrary N_t (number transmit antennas), M (number of data streams), and L (number of codewords in the codebook), is difficult. In Section 2.3.2, we describe one simple codebook construction method [3] yielding codebook with large minimum distances.

2.3.2 Codebook construction method

In [3], it is shown that, maximizing the minimum distance between two subspaces $\mathbf{F}_i$ and $\mathbf{F}_j$, $i \neq j$, is equivalent to minimizing the maximum correlation between them. A Fourier-based construction method based on this result is described below.

We begin with M = 1 (number of data stream = 1). For this case, a precoding matrix with size $N_t \times M$ becomes a precoding matrix with size $N_t \times 1$ (also called a beamforming vector).

With Fourier-based construction, the L vector codewords can be expressed as:

$$F_i = \frac{1}{\sqrt{N_t}} \begin{bmatrix} 1 \\ e^{j\frac{2\pi}{L}(i-1)} \\ e^{j\frac{2\pi}{L}2(i-1)} \\ \vdots \\ e^{j\frac{2\pi}{L}(N_t-1)(i-1)} \end{bmatrix} , \quad i = 1,2,\ldots,L \tag{2-25}$$

where i indicates the index of codeword, and L denotes the total number of codewords within the codebook. For this choice, we obtain the correlation R_{ij} between i_{th} codeword and j_{th} codeword as:

$$R_{ij} = \left|\mathbf{F}_j^* \mathbf{F}_i\right| = \begin{cases} 1 & (i = j) \\ \dfrac{1}{N_t}\left|\sum_{t=1}^{N_t} e^{j\frac{2\pi}{L}(t-1)(i-j)}\right| = \left|\dfrac{\sin(\pi(i-j)N_t/L)}{N_t\sin(\pi(i-j)/L)}\right| & (i \neq j) \end{cases} \tag{2-26}$$

We observe that:

1. The correlation between $\mathbf{F}_i$ and $\mathbf{F}_j$ depend only on (i – j) mod L; the correlation structure of the codebook is therefore circulant and it suffices to consider $\left|\mathbf{F}_1^* \mathbf{F}_j\right|$, for j = 2,3,…,L. That is, to find the maximum correlation, we don't need to calculate C_2^L correlations between all codewords.

2. The correlation structure behaves roughly like a sinc function. With the fact described at the beginning of Section 2.3.2, we want to find a good set of codewords to minimize the maximum correlation between codewords. Figure 2-3 shows the correlation structure of codewords chosen in (2-25) for N_t = 6, L = 64. The maximum correlation is 0.986.

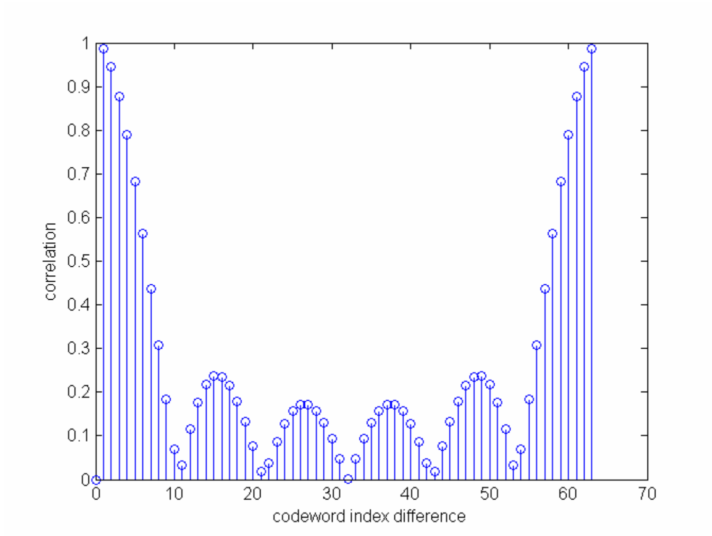

Figure 2-3 Correlation structure of codewords in (2-25) as a function of (i-j)

Since the maximum correlation approaches unity, it is obviously a poor choice for codewords in (2-25), especially when L is large. However, we are not necessarily constrained to choose first N_t rows of the L × L DFT matrix as is done in (2-25). To lower the correlation between neighbors, we may consider choosing another set of N_t components. We thus let

$$F_i = \frac{1}{\sqrt{N_t}} \begin{bmatrix} 1 \\ e^{j\frac{2\pi}{L}u_1(i-1)} \\ e^{j\frac{2\pi}{L}u_2(i-1)} \\ \vdots \\ e^{j\frac{2\pi}{L}u_{N_t}(i-1)} \end{bmatrix} \quad , \quad i = 1,2,...L \tag{2-27}$$

where $0 \le u_1, u_2, ..., u_{N_t} \le L-1$

By (2-26) and (2-27) , we wish to find $u_1, u_2, ..., u_{N_t}$ achieving

$$\min_{0 \le u_1,\ldots u_{N_t} \le L-1} \max_{j=2,\ldots L} \frac{1}{N_t} \left| \sum_{t=1}^{N_t} e^{j\frac{2\pi}{L} u_t (j-1)} \right| \tag{2-28}$$

The minimization problem in (2-28) can be seen as an aperiodic array design problem [10]. Despite much effort, there has never been a completely satisfactory way to design aperiodic arrays: for small arrays one can use exhaustive search, whereas, for large arrays, random search strategies seem to be the only resort. With random search, a good choice for $u_1, u_2, \ldots, u_{N_t}$ can be found. Figure 2-4 shows the correlation structure of codewords chosen in (2-27) for $N_t = 6$, $L = 64$. Here $u_1, u_2, \ldots, u_{N_t}$ = [1 18 23 39 46 57]. The maximum correlation is decreased to 0.5604.

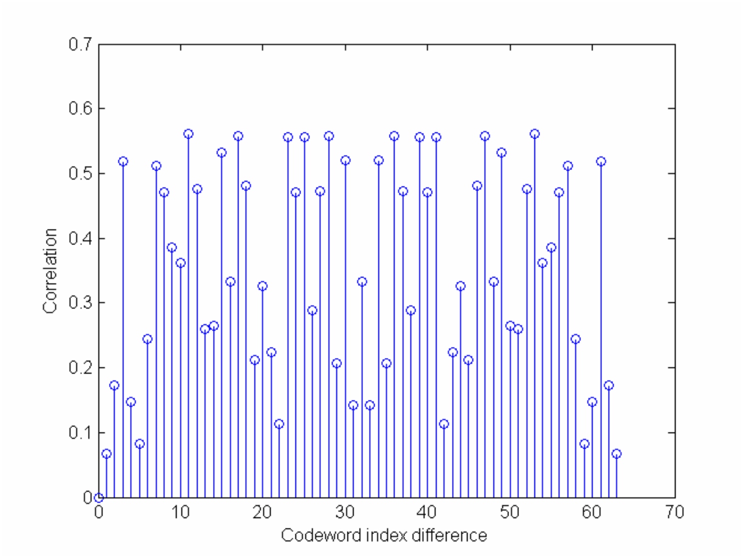

Figure 2-4 Correlation structure of codewords in (2-27) as a function of (i-j)

Now, we can extend this single data stream case (M = 1) to multiple data streams case (M > 1). In the single data stream case discussed above, each vector codeword can be written as

$$\mathbf{F}_i = \Theta^{i-1} \mathbf{F}_1 \quad , \quad i = 1, 2, \ldots, L \tag{2-29}$$

where Θ is a $N_t \times N_t$ diagonal matrix whose diagonal elements are $e^{j2\pi u_1/L}, e^{j2\pi u_2/L}, \ldots, e^{j2\pi u_{N_t}/L}$ and $\mathbf{F}_1$ is $\frac{1}{\sqrt{N_t}}$ times a vector of all ones. Note that Θ is an unitary matrix and $\Theta^{N_t} = \mathbf{I}_{N_t}$. Therefore, (2-29) can be rewritten as

$$\mathbf{F}_i = \begin{bmatrix} e^{j2\pi u_1/L} & 0 & \cdots & 0 \\ 0 & e^{j2\pi u_2/L} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & e^{j2\pi u_{N_t}/L} \end{bmatrix}^{i\text{-}1} \cdot \frac{1}{\sqrt{N_t}} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}, \quad i = 1, 2, \ldots, L \qquad (2\text{-}30)$$

Geometrically, the construction can be interpreted as rotating an initial vector through N_t dimensional complex space using a matrix which is the L_{th} root of unity.

For multiple data streams case (M > 1), let the initial matrix $\mathbf{F}_1$ be a $N_t \times M$ matrix with $\mathbf{F}_1^* \mathbf{F}_1 = \mathbf{I}_M$, and construct the L codewords by applying (2-29) again. For the M > 1 case, this construction can be interpreted as rotating an initial M-dimension subspace using an L_{th} root of unity to form L different M-dimensional subspaces. A simple method to build a starting matrix $\mathbf{F}_1$ is to choose M distinct columns of a $N_t \times N_t$ DFT matrix, and this ensures that $\mathbf{F}_1^* \mathbf{F}_1 = \mathbf{I}_M$.

Now extending (2-28) to multiple data streams case (M > 1). We wish to find $u_1, u_2, \ldots, u_{N_t}$ achieving

$$\min_{0 \le u_1, \ldots u_{N_t} \le L-1} \; \max_{j=2,\ldots L} \left\| \mathbf{F}_1^* \mathbf{F}_j \right\|_F \qquad (2\text{-}31)$$

For finding a good choice of $u_1, u_2, \ldots, u_{N_t}$, random search can be applied. Example codebooks using this Fourier-based construction with different N_t (number transmit antennas), M (number of data streams), and L (number of codewords in the codebook) can be downloaded at [11].

2.4 Limited feedback schemes

In MIMO-OFDM systems, the broadband channel is converted into multiple narrowband channels such that a subcarrier can be used in a channel. Each subcarrier can then perform precoding, independently. However, feeding back the information about precoding matrix for each active subcarrier requires a large amount of data. While codebook-based precoding techniques can be used, the feedback data still grow in proportion to the number of active subcarriers. To solve this problem, some techniques such as clustering and interpolation [5] can be applied.

2.4.1 Clustering

In general, adjacent subchannels in an OFDM system are correlated. As a result, optimal precoders corresponding to neighboring subchannels are also correlated. Using the precoder correlation, the amount of feedback information can be reduced.

A simple approach is to combine the neighboring subcarriers into a cluster and use the quantized optimal precoder corresponding to the center subcarrier for all the subcarriers in that cluster. This method is referred to as ***clustering***.

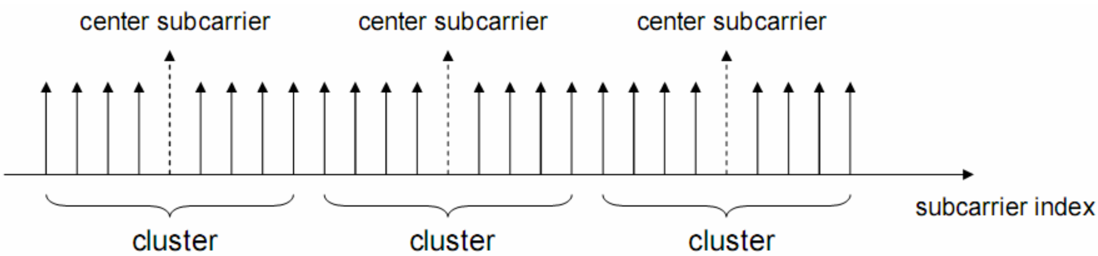

Figure 2-5 Clustering

Suppose that total N_s active subcarriers are divided into clusters as shown in Figure 2-5. Each cluster has K subcarriers, and the receiver only feedbacks the indices of precoders corresponding to center subcarriers. Thus, the feedback information can be reduced to 1/K by clustering. However, because the precoder used for all K subcarriers within a cluster is the quantized optimal precoder corresponding to the center subcarrier, the subcarriers near the cluster boundary will experience larger performance degradation.

2.4.2 Interpolation

To improve the performance of precoding near the cluster boundary, [5] proposes an interpolation scheme. First, the receiver obtains quantized optimal precoders $\{\mathbf{F}(1), \mathbf{F}(K+1), \ldots \mathbf{F}(N-K+1)\}$, where $\mathbf{F}(k)$ means the quantized optimal precoder for the k_{th} subcarrier. Then we send the indices of them to the transmitter and the transmitter determines the precoders for all subcarriers through interpolation of the transmitted precoders. Unfortunately, there are two difficulties for the interpolation scheme.

1. It is not trivial to interpolate the precoders, because the optimal precoder must have orthonormal columns. After interpolation, the orthonormality may not always hold.

2. As we mentioned at Section 2.3.1, the optimal precoder is not unique. That is, if **F'** = $\mathbf{F}_{opt}$ $\mathbf{U}_n$,where $\mathbf{U}_n$ is a M × M unitary matrix, **F'** is also an optimal precoding matrix. Because the precoder is calculated independently for each subcarrier, the unitary matrix $\mathbf{U}_n$ for each subcarrier is also arbitrarily determined. However, the choice of unitary matrix $\mathbf{U}_n$ has a substantial influence on the performance of an interpolator.

Based on these observation, [5] proposes the following interpolation algorithm.

$$\mathbf{Z}(iK+m)=(1-c_m)\mathbf{F}(iK+1)+c_m\mathbf{F}((i+1)K+1)\mathbf{Q}_i \tag{2-32}$$

$$\hat{\mathbf{F}}(iK+m;\mathbf{Q}_i)=\mathbf{Z}(iK+m)\{\mathbf{Z}^*(iK+m)\mathbf{Z}(iK+m)\}^{-\frac{1}{2}} \tag{2-33}$$

where $\mathbf{F}$(N+1) = $\mathbf{F}$(1) , and $\mathbf{Q}_i$ is a M × M unitary matrix. $c_m=(m-1)/K$, $1\le m\le K$. We can see that (2-32) is simply a linear interpolator with an additional matrix $\mathbf{Q}_i$. After interpolation, a projection is then required. In equation (2-33), $\hat{\mathbf{F}}$ is the projection of $\mathbf{Z}$ into $U(N_t,M)$ with respect to the Frobenius norm, and thus it ensures that the orthonormality will hold for the precoder after interpolation. The role of the unitary matrix $\mathbf{Q}_i$ is to solve the non-uniqueness problem. $\mathbf{Q}_i$ can be found in number of ways, such as maximizing the capacity

$$\mathbf{Q}_i=\arg\max_{\bar{\mathbf{Q}}\in C_Q}\sum_{m=1}^{K}Capacity(\hat{\mathbf{F}}(iK+m\ ;\bar{\mathbf{Q}})) \tag{2-34}$$

where $\mathbf{C}_Q$ is a codebook for unitary matrix $\mathbf{Q}_i$. Note that, a large size of $\mathbf{C}_Q$ will cause a higher computational complexity for the search of the best $\mathbf{Q}_i$ and more feedback data (for sending the information about $\mathbf{Q}_i$). In [5], a suggested codebook for $\mathbf{Q}_i$ which contains 4 codewords is shown below:

$$\begin{bmatrix}1&0\\0&1\end{bmatrix}\ \begin{bmatrix}-1&0\\0&-1\end{bmatrix}\ \begin{bmatrix}j&0\\0&j\end{bmatrix}\ \begin{bmatrix}-j&0\\0&-j\end{bmatrix} \tag{2-35}$$

Clustering and interpolation both exploit the correlation between neighboring subcarriers. If the channel has a large coherent bandwidth, the data amount for feedback can be significantly reduced by clustering scheme. The interpolation scheme proposed in [5] can further improve the performance of clustering. However, due to the difficulties we mentioned above, the interpolation scheme requires additional feedback information, and also higher computational complexity. When the coherent bandwidth becomes small, these techniques will apparently suffer performance degradation.

2.5 Proposed codeword search method

For conventional codebook-based precoding, an exhaustive search is required to find the optimal codeword in the codebook. That is, if we have a codebook with L codewords (size = L), we then need to conduct the same operation for L times to find the optimal codeword. At this section, we propose a codeword search method which can reduce about 80% searching complexity with acceptable performance loss.

We use a sub-optimal codeword selection criterion which minimizes the chordal distance (Equation 2-20) between the chosen codeword and the ideal (un-quantized) optimal precoder.

$$\mathbf{F} = \arg\min_{\mathbf{F}_i \in \mathbf{C}} d_{Chordal}\{\mathbf{F}_i, \mathbf{F}_{opt}\} \tag{2-36}$$

The simulation shows that this criterion has performance comparable to MSV-SC. With this distance-based codeword selection criterion, a low complexity codeword search method becomes possible.

The proposed codeword search method is composed of the following two steps:

1. **Codebook partition:** Given any appropriately designed codebook with L codewords, we first partition this codebook with a simple distance comparison

algorithm. After this partition step, the codebook will have a tree structure. Note that, this is an off-line step.

2. **Codeword searching:** With the partitioned codebook known by both the transmitter and the receiver, a tree searching algorithm can be performed to find an optimal codeword within this partitioned codebook.

In codebook partition step, we first find two codewords which have maximum chordal distance. This can be done with an exhaustive search manner. With the two farthest codewords, the other codewords then can be partitioned into two groups with a simple distance comparison algorithm. Assume X is a codeword,

If $d_{chordal}(\mathbf{A},\mathbf{X}) < d_{chordal}(\mathbf{B},\mathbf{X})$, then **X** will be referred to group ***a***.

If $d_{chordal}(\mathbf{A},\mathbf{X}) > d_{chordal}(\mathbf{B},\mathbf{X})$, then **X** will be referred to group ***b***.

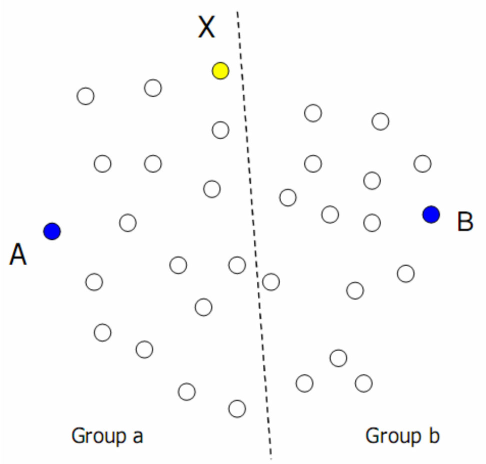

Figure 2-6 Partition the codebook into two groups

Then, we can further find two farthest codeword **C** and **D** in group ***a***, and two farthest codeword **E** and **F** in group ***b***. Following the distance comparison algorithm, group ***a*** can be further partitioned into group ***c*** and group ***d***, and group ***b*** can be partitioned into group ***e*** and group ***f***.

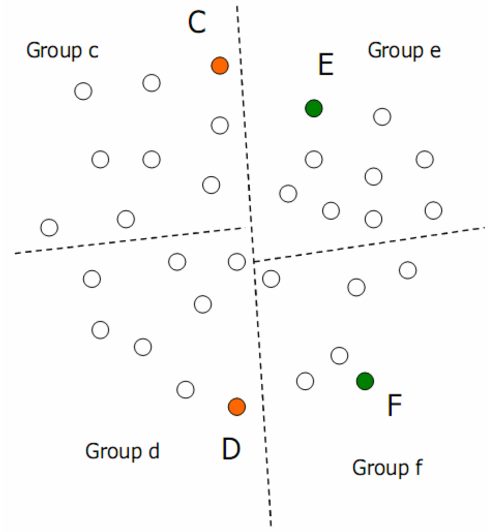

Figure 2-7 Partition the codebook into 4 groups

Finally, the codebook will be partitioned into 2, 4, 8, … ,2^k groups , where the integer k can be defined as the ***searching depth***. The farthest codewords within each group must be recorded during the partition process. Note that groups at depth k are all contained in groups at depth k-1, in other words, the partitioned codebook has a nested structure.

Figure 2-8 shows an example of codebook partition with codebook size = 64. The maximum searching depth k is 3. Note that this codebook is partitioned unequally (number of codewords in each group at same depth is not equal).

<table>
<tr><td colspan="8">L = 64</td></tr>
<tr><td colspan="4">30</td><td colspan="4">34</td></tr>
<tr><td colspan="2">19</td><td colspan="2">11</td><td colspan="2">18</td><td colspan="2">16</td></tr>
<tr><td>9</td><td>10</td><td>7</td><td>4</td><td>11</td><td>7</td><td>8</td><td>8</td></tr>
</table>

Figure 2-8 Example of codebook partition

After codebook partition step, the tree structure is determined and stored. In online codeword search, we can then perform a tree search algorithm to locate the optimal codeword within this partitioned codebook. First, we calculate the ideal optimal precoder $\mathbf{F}_{opt}$ by SVD of channel matrix $\mathbf{H}$. Then, we find a codeword $\mathbf{F}_i$ which has minimum chordal distance to $\mathbf{F}_{opt}$. The first step is to compare $d_{chordal}(\mathbf{A},\mathbf{F}_{opt})$ and $d_{chordal}(\mathbf{B},\mathbf{F}_{opt})$, where $\mathbf{A}$ and $\mathbf{B}$ are two farthest codewords within the codebook (the largest group). If $\mathbf{F}_{opt}$ is nearer to the codeword $\mathbf{A}$, the optimal codeword is assumed to be within group $\boldsymbol{a}$. Then, further compare $d_{chordal}(\mathbf{C},\mathbf{F}_{opt})$ and $d_{chordal}(\mathbf{D},\mathbf{F}_{opt})$, where $\mathbf{C}$ and $\mathbf{D}$ are two farthest codewords within the group $\boldsymbol{a}$. If $\mathbf{F}_{opt}$ is nearer to the codeword $\mathbf{C}$, the optimal codeword is assumed to be within group $\boldsymbol{c}$. Repeat this process, and finally we can find the optimal codeword. Using the algorithm, we can significantly reduce the searching complexity.

Assume a codebook can be partitioned equally at each level, then we can easily express the complexity of the proposed codeword searching algorithm as follows:

$$Searching\ complexity = 2k + \frac{L}{2^k} \qquad (2\text{-}37)$$

Table 2-1 shows the searching complexity for L = 64 and L = 128.

Searching complexity		
	L = 64	L = 128
k = 0	64	128
k = 1	34	66
k = 2	20	36
k = 3	14	24
k = 4	12	16
k = 5	12	14
k = 6	12	14
k = 7		14

Table 2-1 Searching complexity for equally partitioned codebook

Note that k = 0 corresponds to exhaustive search. Note that this complexity indicates the number of chordal distance calculations. Although it's difficult to partition the codebook equally, simulation shows that the average searching complexity with unequally partitioned codebook will approach to this result listed in table 2-1.

One problem with the proposed algorithm is that codeword searching error will occur under some situations.

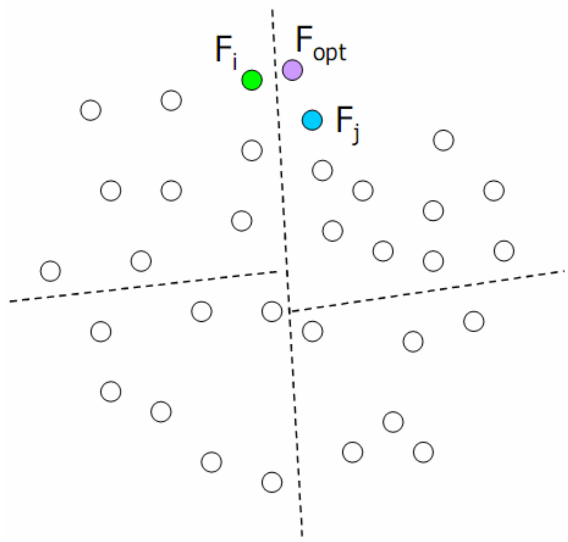

Figure 2-9 Codeword searching error

As we show in Figure 2-9, the codeword $\mathbf{F}_i$ has minimum chordal distance to ideal optimal precoder $\mathbf{F}_{opt}$. However, the optimal codeword is determined in the wrong group. This is because $\mathbf{F}_i$ and $\mathbf{F}_{opt}$ are too close to the partition edge. In this case, if we assume searching depth is two, the codeword chosen by tree search algorithm will be $\mathbf{F}_j$, which is a sub-optimal codeword. The codeword searching error causes performance loss compared to exhaustive search.

Table 2-2 shows the average complexity ratio and average distance error. The complexity ratio is defined as the complexity of proposed tree search algorithm divided by complexity of exhaustive search. The complexity of exhaustive search is equal to the size of codebook, L. The distance error can be defined as:

$$d_{chordal}(\mathbf{F}_j, \mathbf{F}_{opt}) - d_{chordal}(\mathbf{F}_i, \mathbf{F}_{opt}) \tag{2-38}$$

where $\mathbf{F}_i$ is the exhaustively searched codeword , and $\mathbf{F}_j$ is the codeword chosen by tree search algorithm. If there is no codeword searching error, the distance error will be zero.

Average complexity ratio		
	L = 64	L = 128
k = 3	0.2177	0.1725
k = 4	0.1866	0.1257
k = 5		0.1099

Average distance error		
	L = 64	L = 128
k = 3	0.0927	0.0803
k = 4	0.1333	0.1150
k = 5		0.1450

Table 2-2 Searching complexity ratio and distance error

From table 2-2, we can realize that increasing the size of codebook, L, will decrease the complexity ratio and distance error. For instance, if L = 64, k = 3, the complexity for tree search algorithm only requires 21.77% of that for exhaustive search. If we increase L to 128, the complexity ratio can be further reduced to 17.25%. Besides, increasing the searching depth k will decrease the complexity ratio but cause higher distance error. For L = 64, k = 3 or k = 4 will be good choices. Further increase the searching depth will not reduce complexity but will incur serious performance loss (high probability of codeword searching error).

In order to lower the probability of codeword searching errors, we can modify the original slightly. As the codebook partition method described above, we first find two codewords which have maximum chordal distance. The other codewords then can be partitioned into two groups with a modified distance comparison algorithm.

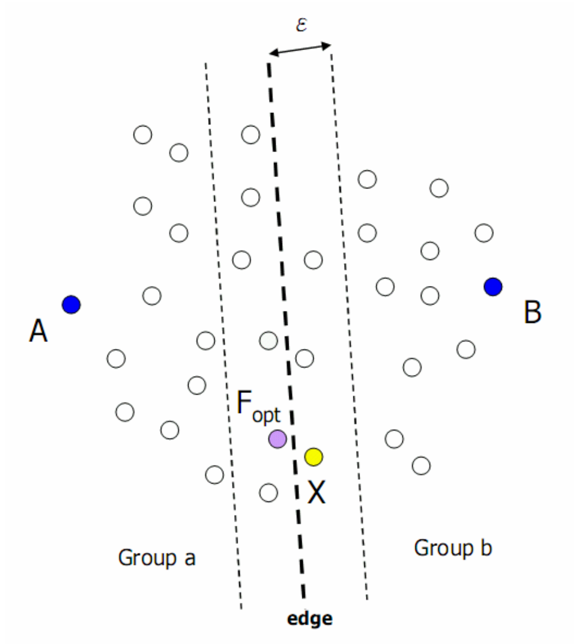

Figure 2-10 Modified codebook partition

Define a new factor called the overlap threshold, denoted as ε . Assume that **X** is a codeword,

If $\left|d_{chordal}(\mathbf{A},\mathbf{X})-d_{chordal}(\mathbf{B},\mathbf{X})\right|<\varepsilon$, then **X** will be referred to both groups, ***a*** and ***b***.

If $\left|d_{chordal}(\mathbf{A},\mathbf{X})-d_{chordal}(\mathbf{B},\mathbf{X})\right|>\varepsilon$, then the algorithm is unchanged:

$d_{chordal}(\mathbf{A},\mathbf{X})<d_{chordal}(\mathbf{B},\mathbf{X})$, then **X** will be referred to group ***a***.

$d_{chordal}(\mathbf{A},\mathbf{X})>d_{chordal}(\mathbf{B},\mathbf{X})$, then **X** will be referred to group ***b***.

As shown in Figure 2-10, **X** is the nearest codeword to the ideal optimal precoder $\mathbf{F}_{opt}$. However, with the original partition method, they will be partitioned into different groups. It will cause codeword searching error when performing tree search algorithm. For the modified partitioned method, we refer **X** to both groups and thus avoiding the error. Obviously, a higher overlap threshold will result in a higher searching complexity since each group size is enlarged. Table 2-3 shows the average complexity ratio and average distance error with this modified codebook partition algorithm for ε = 0.05

Average complexity ratio		
	L = 64	L = 128
k = 3	0.3199	0.2528
k = 4	0.2663	0.1878

Average distance error		
	L = 64	L = 128
k = 3	0.0520	0.0462
k = 4	0.0723	0.0642

Table 2-3 Searching complexity ratio and distance error with modified codebook partition algorithm

Compared to table 2-2, we can find that the average distance error is smaller for different codebook size and different searching depth. However, the complexity ratio increases significantly, also. Thereby, how to choose an appropriate overlap threshold becomes a critical problem.

Chapter 3 Time domain CSI feedback

In this chapter, we propose time domain CSI feedback methods. Under some channel conditions, the proposed time domain CSI feedback method only requires a small amount of data. Even in the application of procoding, our method is comparable to the conventional precoder feedback scheme such as clustering. For time varying channels, we incorporate a differential pulse code modulation (DPCM) scheme in our time domain CSI feedback method such that the required feedback data can be further reduced.

3.1 Least squares method

We begin with an example of a 4 by 2 MIMO channel model shown in Figure 3-1. As we can see, the MIMO channel contains 8 single input single output (SISO) channels. We refer each SISO channel from one transmit antenna to one receive antenna as a Tx-Rx channel pair.

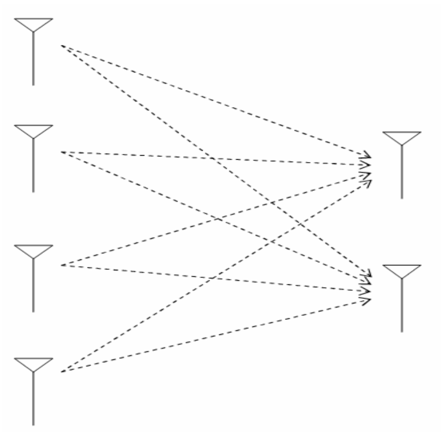

Figure 3-1 4 by 2 MIMO channel model

Figure 3-2 shows a typical time domain channel response for one Tx-Rx channel pair.

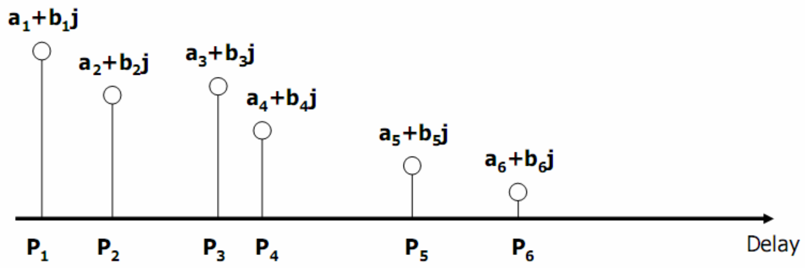

Figure 3-2 A time domain channel response

A simple way to quantize the time domain channel response is to directly quantize the complex value and delay for each channel tap, individually. That is, quantize a_1, b_1, a_2, b_2, …, a_6, b_6, P_1,P_2,…, P_6 , if there are 6 channel taps. However, it may require large amount of quantization bits. Therefore, we propose to quantize the overall time domain channel response, jointly. For this purpose, we first shorten the channel taps by removing insignificant taps for each Tx-Rx channel pair. Then, we sort the shortened channel based on magnitude, as shown in Figure 3-3.

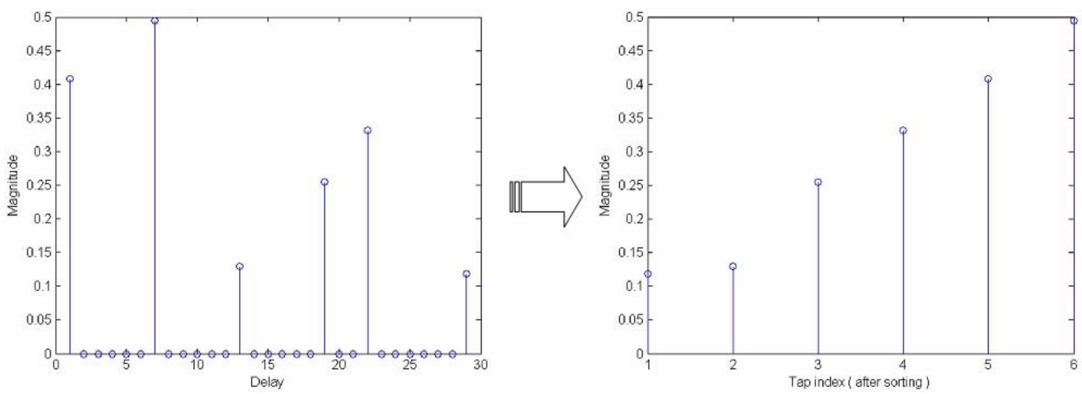

Figure 3-3 Shortening and sorting the time domain channel response

After shortening and sorting, the magnitude of time domain channel response will have high correlation between each channel tap. Thus, we can apply least squares

(LS) to fit these sorted taps with a straight line or a higher-order polynomial curve and thus avoid the quantization of each channel tap. Notice that, the delay information is quantized before shortening. The delay information fed from receiver back to transmitter can be used to recover the original taps before shortening and sorting. Besides, the sorting operation is based on magnitude, thus the phase information must be quantized with other scheme. Because the phase for each tap is i.i.d. and has uniform distribution, we simply apply an uniform quantizer for the phase information.

Least squares (LS) method is a well-known curve fitting method. Given N observed data, we can find a straight line or a higher-order polynomial curve to fit these data with a minimum squared errors. Figure 3-4 shows an example of linear fitting:

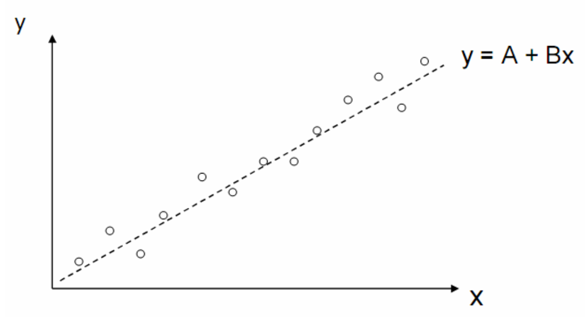

Figure 3-4 First-order polynomial least squares fitting

We express the N observed real-valued data as:

$$\bar{\mathbf{X}} = \begin{bmatrix} x_1 & x_2 & \cdots & x_N \end{bmatrix}^T \tag{3-1}$$

The LS method finds a parameter vector $\bar{\boldsymbol{\theta}}$ to minimize the squared error vector $J(\bar{\boldsymbol{\theta}})$, which can be written as:

$$J(\bar{\boldsymbol{\theta}}) = (\bar{\mathbf{X}} - \mathbf{K}\bar{\boldsymbol{\theta}})^T (\bar{\mathbf{X}} - \mathbf{K}\bar{\boldsymbol{\theta}}) \tag{3-2}$$

where $\mathbf{K}\overline{\boldsymbol{\theta}}$ is the fitting vector, and $\mathbf{K}$ is a known observation matrix. For the first-order polynomial fitting, $\mathbf{K}$ can be written as:

$$\mathbf{K} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ \vdots & \vdots \\ 1 & N-1 \end{bmatrix} \tag{3-3}$$

For the second-order polynomial fitting, $\mathbf{K}$ can be written as:

$$\mathbf{K} = \begin{bmatrix} 1 & 0 & 0^2 \\ 1 & 1 & 1^2 \\ \vdots & \vdots & \vdots \\ 1 & N-1 & (N-1)^2 \end{bmatrix} \tag{3-4}$$

If the gram matrix $\mathbf{K}^T\mathbf{K}$ is non-singular, then the least squares solution will be:

$$\hat{\boldsymbol{\theta}}_{LS} = (\mathbf{K}^T\mathbf{K})^{-1} \cdot \mathbf{K}^T\overline{\mathbf{X}} \tag{3-5}$$

Figure 3-5 shows an example of the first-order polynomial fitting for 6 sorted channel taps.

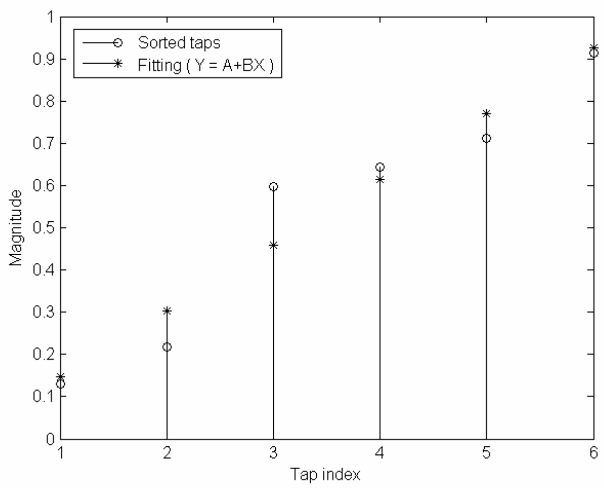

Figure 3-5 First-order polynomial fitting for sorted channel taps

A second-order polynomial fitting for 6 sorted channel taps is shown in Figure 3-6.

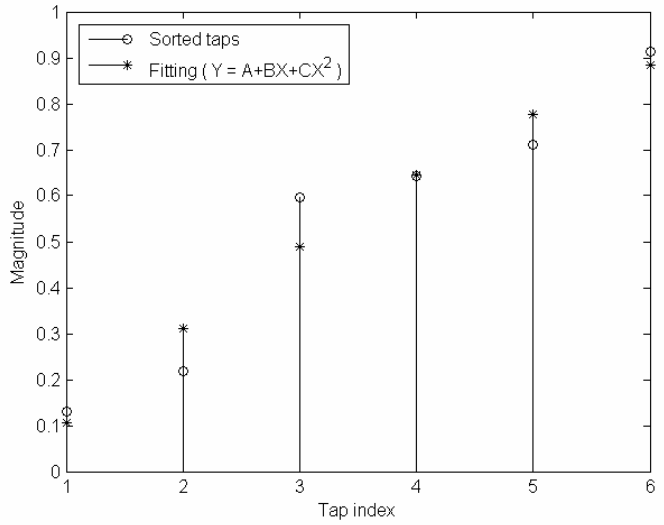

Figure 3-6 Second-order polynomial fitting for sorted channel taps

Obviously, the second-order polynomial fitting has higher accuracy (lower squared-errors) compared to the first-order polynomial fitting.

Now we can briefly describe the time domain feedback scheme with LS fitting method as follows:

1. Quantize the delay and phase information first. Then, shorten and sort the channel taps using their magnitude for each Tx-Rx channel pair.

2. Apply LS method to fit the sorted magnitude response for each Tx-Rx channel pair.

3. Feedback the quantized LS parameters together with the quantized phase and delay information to the transmitter side.

At transmitter, we can reconstruct the time domain channel response using the LS parameters, quantized phases, and delay information. If precoding is conducted at transmitter, the precoding matrix for each active subcarrier can be obtained by performing singular value decomposition (SVD) on the reconstructed frequency domain channel response.

3.2 Discrete cosine transform method

In this section, we propose another time domain CSI feedback scheme using discrete cosine transform (DCT). Assume that x(n) is a real-valued sequence. A one-dimensional DCT can be expressed as follows.

$$y(k) = w(k)\sum_{n=0}^{N-1} x(n)\cos\frac{\pi(2n+1)k}{2N}, \quad 0 \le k \le N-1$$
$$where \quad w(k) = \begin{cases} \sqrt{1/N} & k = 0 \\ \sqrt{2/N} & 1 \le k \le N-1 \end{cases} \tag{3-6}$$

It has been shown that many physical signals can be accurately reconstructed using only a few of their DCT coefficients. Therefore, it is useful in data compression. We can easily extend the one dimensional DCT in (3-6) to two-dimensional DCT as:

$$B_{pq} = \alpha_p \alpha_q \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} A_{mn} \cos\frac{\pi(2m+1)p}{2M}\cos\frac{\pi(2n+1)q}{2N}, \quad \begin{matrix} 0 \le p \le M-1 \\ 0 \le q \le N-1 \end{matrix}$$

$$where \quad \alpha_p = \begin{cases} \sqrt{1/M}, & p = 0 \\ \sqrt{2/M}, & 1 \le p \le M-1 \end{cases} \qquad \alpha_q = \begin{cases} \sqrt{1/N}, & q = 0 \\ \sqrt{2/N}, & 1 \le q \le N-1 \end{cases}$$

(3-7)

Comparing (3-6) and (3-7), we can see that the two-dimensional DCT is equivalent to two one-dimensional DCTs, performed along one dimension followed by another DCT in the other dimension. Two-dimensional DCT is a very common method for image compression.

As discussed in Section 2.1, shortening and sorting for each Tx-Rx channel pair will generate correlations between the channel taps. Therefore, a joint quantization strategy such as LS method proposed in the previous section can be applied to quantize the channel taps with only a few parameters. Note that, each Tx-Rx channel pair is quantized independently in the LS fitting method. However, correlations also exist between different Tx-Rx channel pairs. For small separation distance between the receive antennas, the channel pairs from one transmit antenna to different receive antennas are often very similar. Therefore, this inspires us to apply a two-dimensional quantization scheme such as DCT to quantize multiple Tx-Rx channel pairs.

After shortening and sorting to each Tx-Rx channel pair, we can collect the sorted taps for all channel pairs and regard them as a two-dimensional response. An example is shown in Figure 3-7.

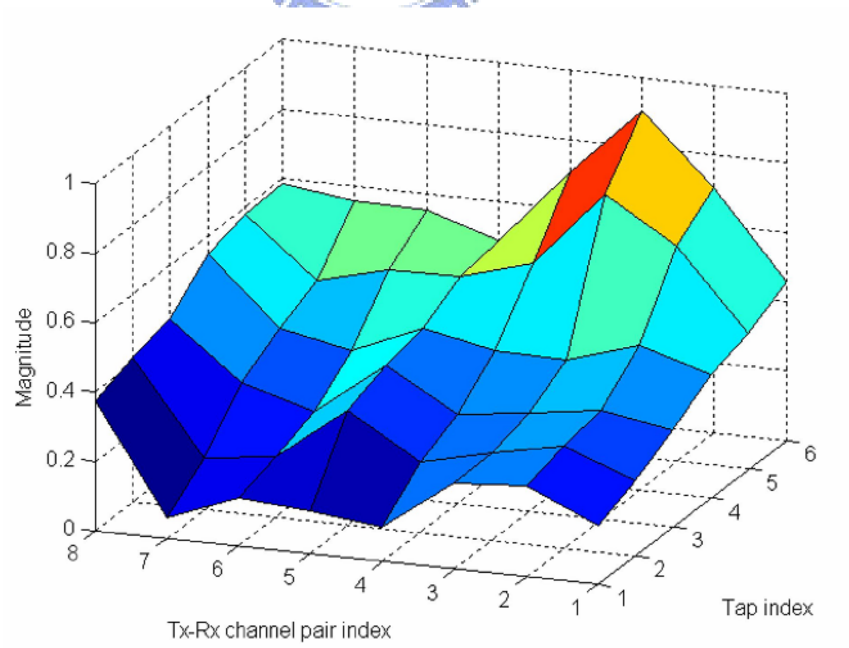

Figure 3-7 Sorted taps for the entire MIMO channel

Now we can perform two-dimensional DCT to the sorted magnitude response in Figure 3-7. The transformation result is shown in Figure 3-8.

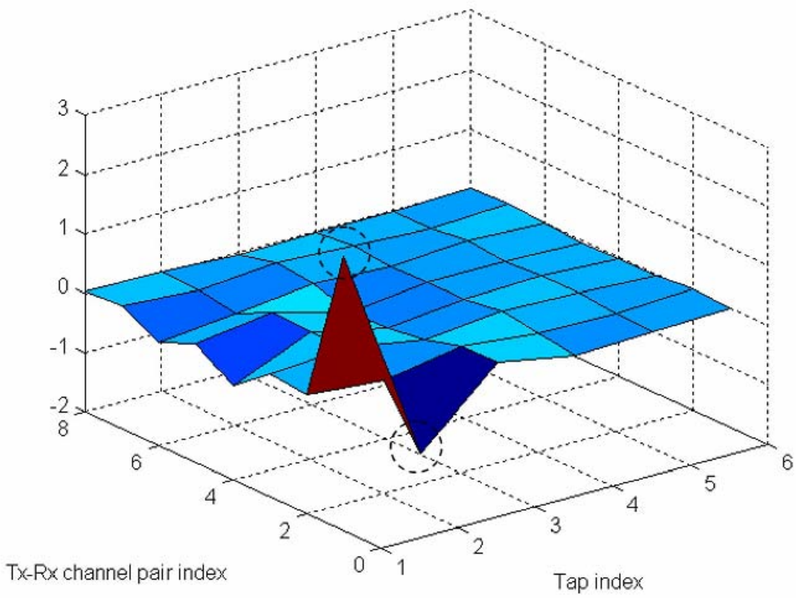

Figure 3-8 Two-dimensional DCT

Apparently, there are two significant parameters at positions (1,1) and (1,2). We can extract these two parameters to reconstruct the sorted magnitude response by inverse two-dimensional DCT, as shown in Figure 3-9.

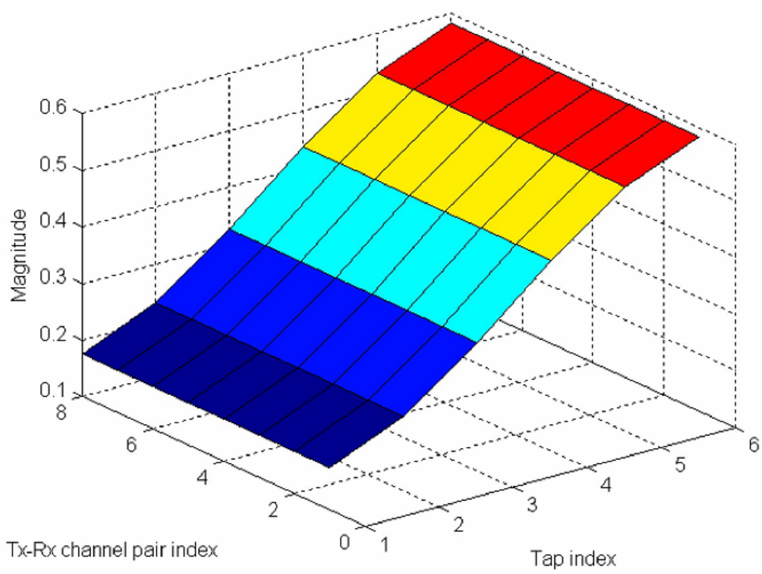

Figure 3-9 Reconstructed magnitude response with two DCT parameters

Since the response in Figure 3-9 is reconstructed with two most significant parameters, distortion is unavoidable. Figure 3-10 shows the reconstructed sorted magnitude response with six most significant parameters. Distortion is obviously lowered compared to that in Figure 3-9.

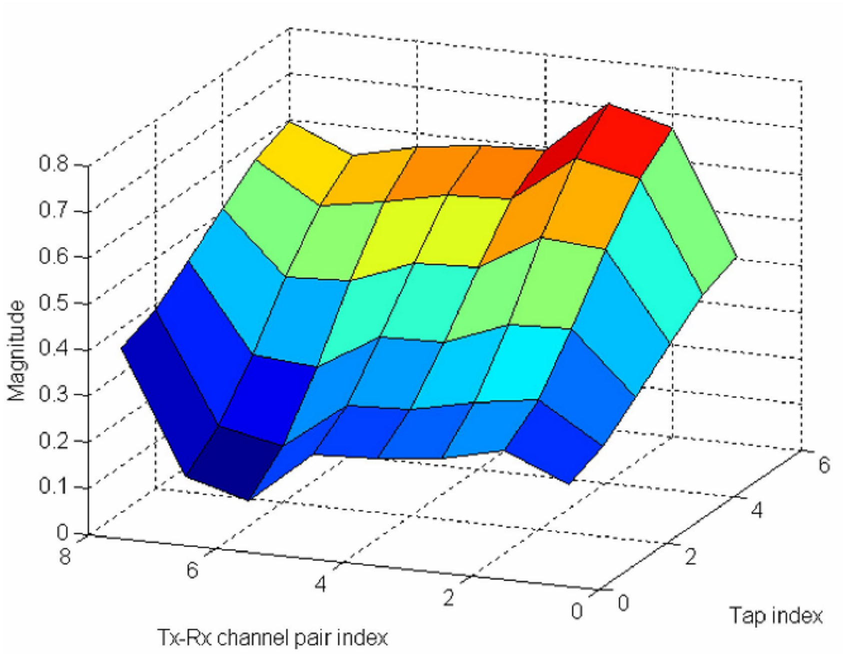

Figure 3-10 Reconstructed magnitude response with six DCT parameters

We briefly describe the proposed time domain feedback scheme with DCT as follows:

1. Quantize the delay information and phase information first. Then, shorten and sort the channel taps based on magnitude for each Tx-Rx channel pair.

2. Apply DCT to the entire sorted magnitude response, and extract the most significant parameters.

3. Feedback the quantized DCT parameters together with the quantized phase and

delay information to the transmitter side.

3.3 Differential pulse code modulation

Differential pulse code modulation (DPCM) is a technique which is often used in speech coding or audio coding. It exploits the correlations between input signals and the quantization bits can be reduced significantly compared to conventional pulse code modulation (PCM). Conventional PCM is an instantaneous quantization scheme. That is, to quantize the signal at different time independently. When the signal to be quantized varies slowly, conventional PCM is not efficient.

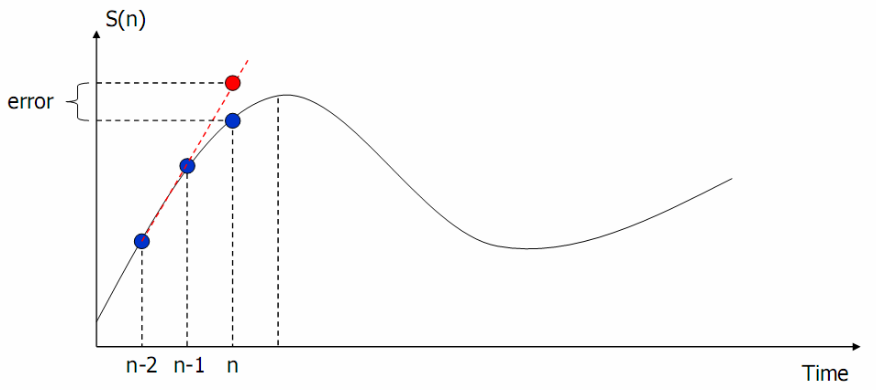

Figure 3-11 Quantization to the prediction error

The main idea of DPCM is to quantize the prediction error of signal, rather than the instantaneous signal itself. For example, assume we have exact value for S(n-2) and S(n-1), then S(n) can be predicted with a linear predictor. Let the predicted signal be $\tilde{S}(n)$. If the signal varies slowly with time, the prediction error $S(n)-\tilde{S}(n)$ is often very small and can be quantized efficiently. Figure 3-12 shows the block diagram of a simple DPCM scheme:

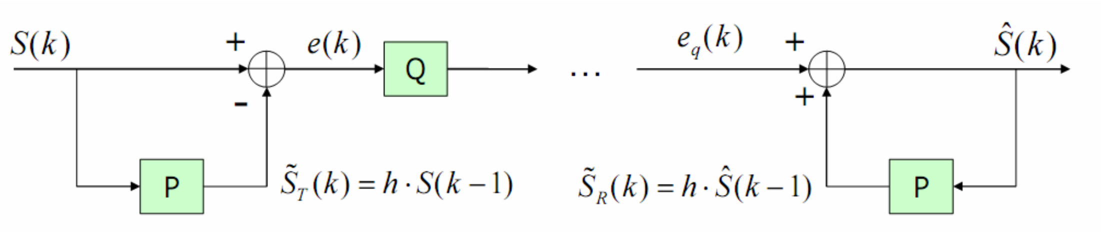

Figure 3-12 Open-loop DPCM

It is an open-loop DPCM scheme, where **P** denotes predictor, **Q** denotes the quantizer, and k is the discrete time index. However, this open-loop scheme will cause an accumulation of reconstruction errors. At transmitter, the prediction error e(k) can be expressed as:

$$e(k) = S(k) - \tilde{S}_T(k) \tag{3-8}$$

where S(k) is the input signal, and $\tilde{S}_T(k)$ is the prediction signal at transmitter.

$$\tilde{S}_T(k) = h \cdot S(k-1) \tag{3-9}$$

From (3-8) and (3-9), we can write the input signal S(k) as:

$$S(k) = h \cdot S(k-1) + e(k) \tag{3-10}$$

Iterating (3-10) for k =1,2,…K, we have

$$\begin{aligned} S(1) &= h \cdot S(0) + e(1) \\ S(2) &= h \cdot S(1) + e(2) = h^2 \cdot S(0) + h \cdot e(1) + e(2) \\ &\vdots \\ S(K) &= h^K \cdot S(0) + \sum_{i=0}^{K-1} h^i \cdot e(K-i) \end{aligned} \tag{3-11}$$

Equation (3-11) expresses the input signal S(K) in terms of the initial value S(0) and prediction error e(k)'s. Now we turn to the receiver side. At receiver, the reconstructed signal $\hat{S}(k)$ can be written as:

$$\hat{S}(k) = \tilde{S}_R(k) + e_q(k) \tag{3-12}$$

where $\tilde{S}_R(k)$ is the prediction signal at receiver and $e_q(k)$ is the quantized prediction error. That is,

$$\tilde{S}_R(k) = h \cdot \hat{S}(k-1) \tag{3-13}$$

$$e_q(k) = e(k) - q(k) \tag{3-14}$$

where q(k) denotes the quantization error. Using (3-13) and (3-14), we can rewrite (3-12) as:

$$\begin{aligned} \hat{S}(k) &= h \cdot \hat{S}(k-1) + e_q(k) \\ &= h \cdot \hat{S}(k-1) + e(k) - q(k) \end{aligned} \tag{3-15}$$

Iterating (3-15), for k =1,2,…K, we have

$$\begin{aligned} &\hat{S}(1) = h \cdot \hat{S}(0) + e(1) - q(1) \\ &\hat{S}(2) = h^2 \cdot \hat{S}(0) + h \cdot [e(1) - q(1)] + [e(2) - q(2)] \\ &\qquad\qquad \vdots \\ &\hat{S}(K) = h^K \cdot \hat{S}(0) + \sum_{i=0}^{K-1} h^i \cdot [e(K-i) - q(K-i)] \end{aligned} \tag{3-16}$$

Equation (3-16) express the reconstructed signal $\hat{S}(k)$ in terms of the initial value $\hat{S}(0)$, prediction error e(k)'s, and quantization error q(k)'s. Assume that $S(0) = \hat{S}(0)$. Comparing (3-11) and (3-16), we can write the reconstruction error

$S(k) - \hat{S}(k)$ as:

$$S(k) - \hat{S}(k) = \sum_{i=0}^{k-1} h^i \cdot q(k-i) \tag{3-17}$$

Therefore, for an open-loop DPCM scheme, the reconstruction error will accumulated, as shown in equation (3-17).

In order to solve this problem, a closed-loop DPCM shown in Figure 3-13 is often applied:

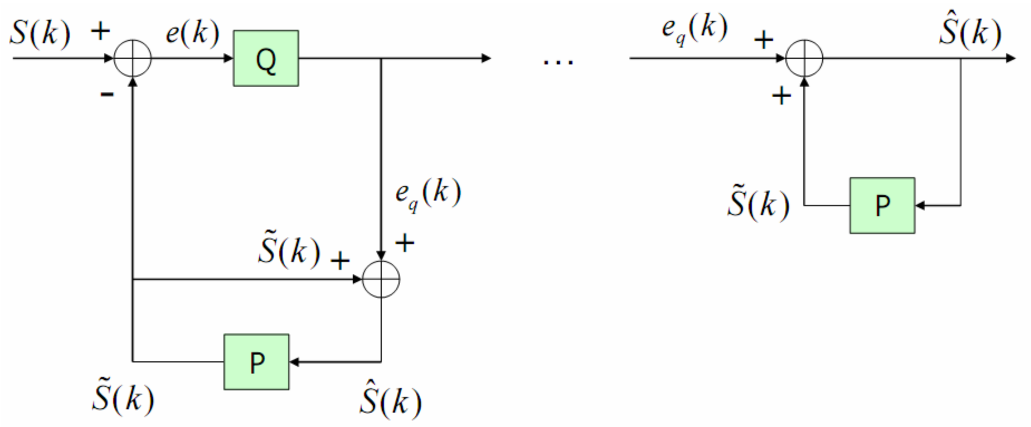

Figure 3-13 Closed-loop DPCM

For closed-loop DPCM, the prediction signal at transmitter and that at receiver are made identical. That is, $\tilde{S}_R(k) = \tilde{S}_T(k) = \tilde{S}(k)$. For this purpose, we need to reconstruct the signal $\hat{S}(k)$ at transmitter first, and predict the signal $\tilde{S}(k)$ based on the previous reconstructed signal. We can express the reconstruction error $S(k) - \tilde{S}(k)$ for closed-loop DPCM as:

$$\begin{aligned} S(k) - \hat{S}(k) &= S(k) - [\tilde{S}(k) + e_q(k)] \\ &= S(k) - [\tilde{S}(k) + e(k) - q(k)] \\ &= S(k) - [S(k) - e(k) + e(k) - q(k)] \\ &= q(k) \end{aligned} \tag{3-18}$$

From (3-18), we can find that the reconstruction error at time = k is identical to that at time = k. Therefore, for a closed-loop DPCM, the reconstruction error accumulation will not happen.

So far, we have described how to apply DPCM to quantize a slowly-varying signal. In many scenarios, the variation of channel taps is slow. We can then apply the DPCM method to further reduce the feedback data. Here, we use an example to demonstrate the effectiveness of the method. We use the spatial channel model (SCM) [12], provided by 3GPP, as our time-varying channel model. The SCM channel model gives 6 non-zero taps for each Tx-Rx channel pair and their values change with time. For our application, we let the speed for mobile station be 20 km/hr.

We assume that the channel is quasi-stationary, which means that the channel is time-invariant in one OFDM-symbol. For our system, one frame consists of 10 OFDM symbols. For each frame, only the CSI for the first OFDM symbol is fed back to the transmitter. Besides, the delay for each tap is assumed to be time-invariant. That is, only the magnitude and phase information of the first OFDM symbol are fed back to the transmitter for each frame.

Now we can combine the DPCM scheme with the time domain CSI feedback schemes described in Section 3.1 and 3.2. The parameter for LS or DCT method now can be considered as a signals varied slowly with time. Figure 3-14 and 3-15 show a variation of the two LS parameters A and B for linear fitting (Y=A+BX).

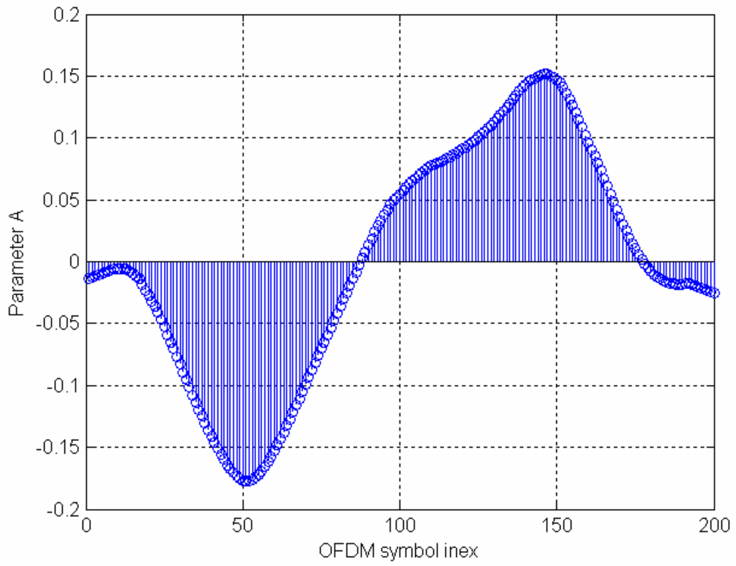

Figure 3-14 Variation of parameter A

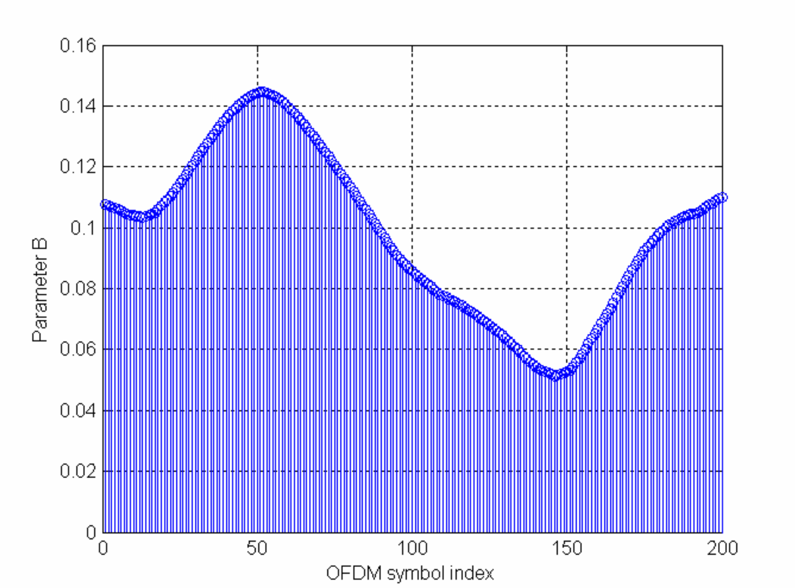

Figure 3-15 Variation of parameter B

Figure 3-16 shows the phase variation for one tap.

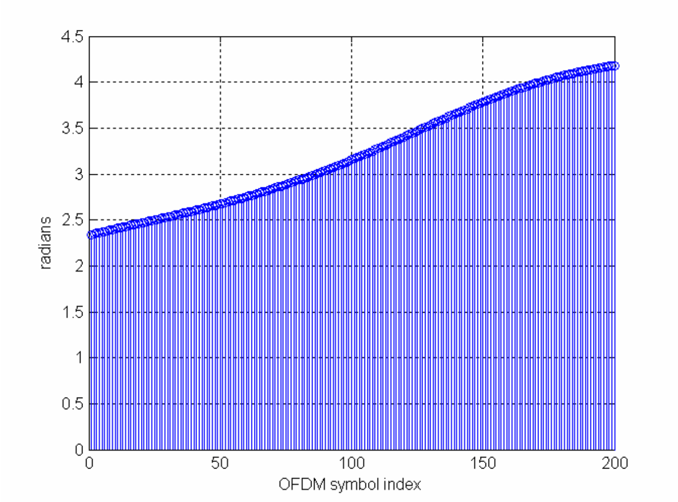

Figure 3-16 Variation of phase for one tap

Then, we quantize the parameters and phases by DPCM scheme with a linear predictor. We give two bits for each frame to quantize one parameter. Figure 3-17 and 3-18 shows the reconstructed parameters A and B at transmitter.

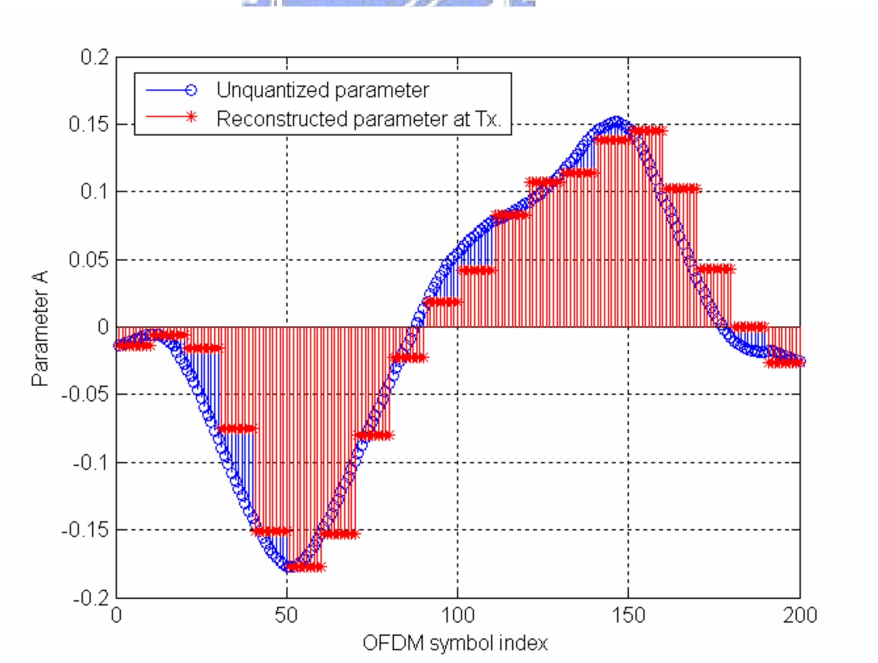

Figure 3-17 Reconstructed parameter A at Tx.

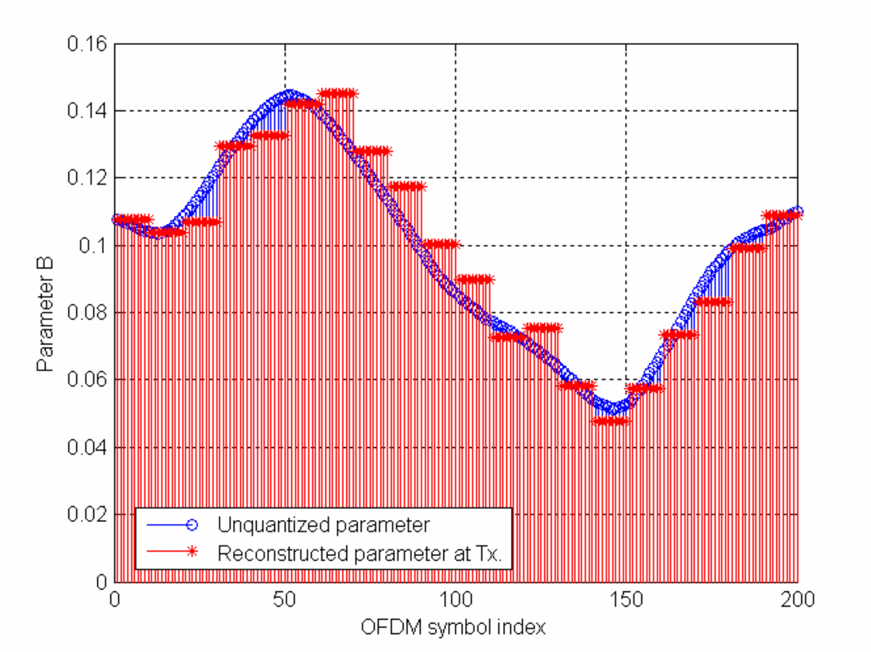

Figure 3-18 Reconstructed parameter B at Tx.

In Figure 3-19, we show the reconstructed phase with the one-bit DPCM. Figure 3-20 shows the reconstructed phase with the two-bit DPCM.

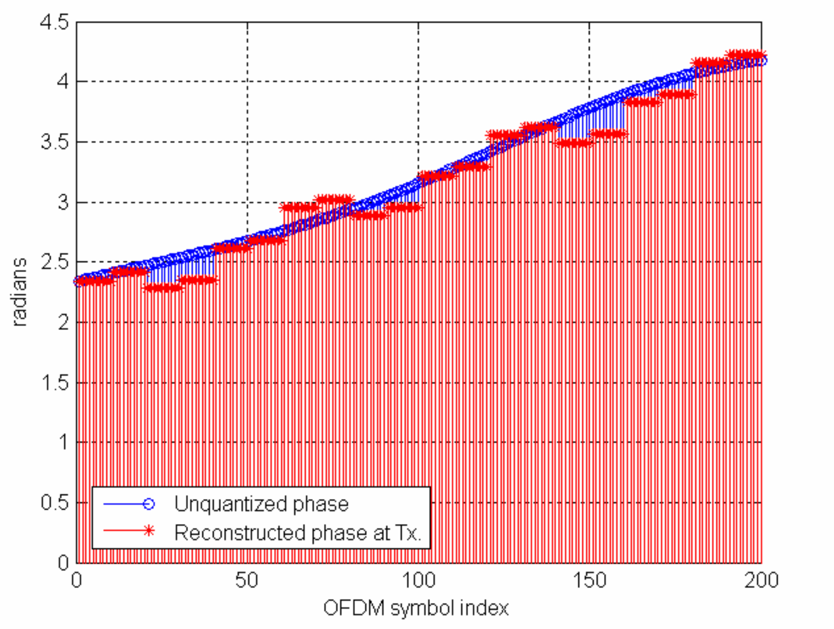

Figure 3-19 Reconstructed phase at Tx. (1 bit)

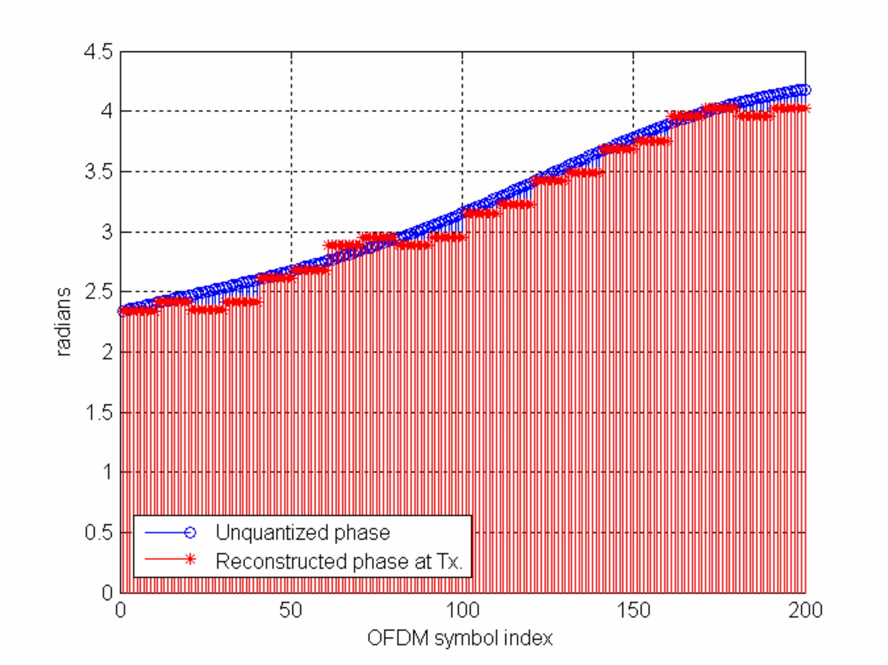

Figure 3-20 Reconstructed phase at Tx. (2 bits)

We now summarize the time-domain feedback scheme with time-varying channel as follows:

1. Shorten and sort the channel taps based on magnitude for each Tx-Rx channel pair.

2. Apply LS method to fit the sorted magnitude response for each Tx-Rx channel pair.

3. For the first two frames, the LS parameters for each channel pair and phases for each tap are quantized with conventional PCM and sent back to the transmitter. After that, with a linear predictor, we can apply DPCM to quantize the prediction error of the time-varying parameters and phases.

4. Feedback the quantized prediction error of LS parameters for and phases to the transmitter side.

Chapter 4 Simulations

4.1 Precoding

In this section, we report some simulation results to evaluate the performance of the proposed codeword search method. A simplified MIMO-OFDM system with precoding is constructed. Two independent data streams (M = 2) are sent over a 4 × 2 system (N_t = 4, N_r = 2). The QAM size is 16, the FFT size is 512, and the cyclic prefix (CP) size is 64. For simplicity, an uncoded system and a basic zero forcing (ZF) receiver are conducted. Besides, we assume that perfect channel estimation can be obtained at receiver. Also, the feedback channel is error-free and has zero delay.

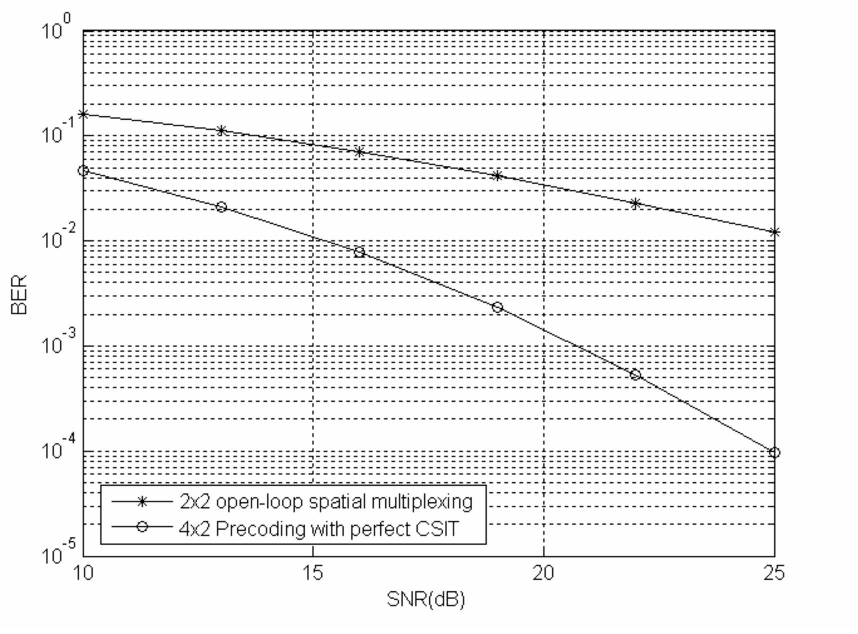

Figure 4-1 BER comparison for 2x2 open-loop SM and 4x2 precoding with perfect CSIT

We assume that the channel experiences a block Rayleigh fading, and each Tx-Rx channel pair (one SISO channel) has 6 taps and fixed delay [1, 22, 23, 26, 51, 56]. Figure 4-1 shows BER comparison for 2 × 2 open-loop spatial multiplexing (SM) system and 4×2 precoding system with perfect CSI at transmitter (CSIT).

As we can see, precoding can significantly improve the system performance.

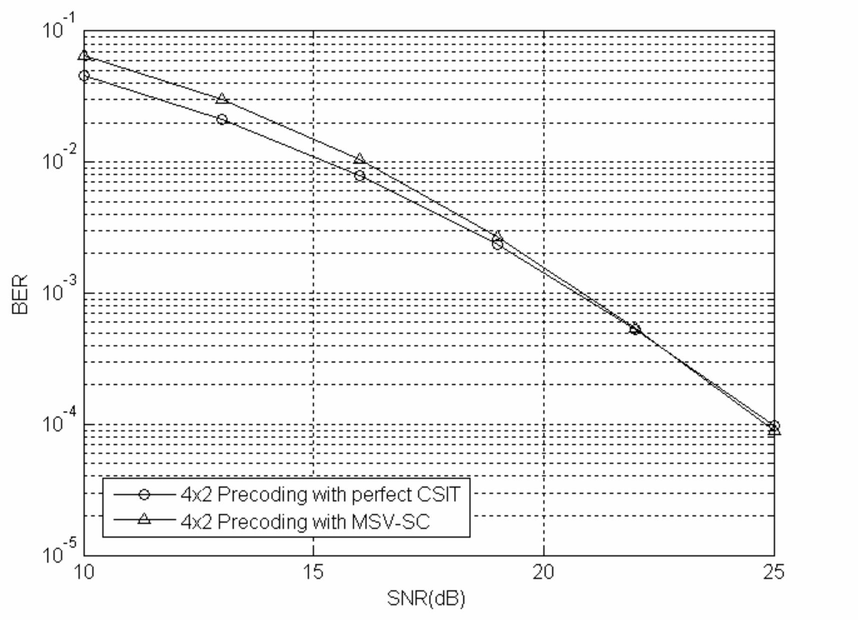

Figure 4-2 BER comparison for precoding with perfect CSIT and precoding with MSV-SC (L=64)

Figure 4-2 shows the performance comparison between codebook-based precoding and ideal precoding. For codebook-based precoding, the codebook size L is 64 and the codeword selection criterion is MSV-SC. Exhaustive search is conducted to find the optimal codeword for each subcarrier.

For the proposed codeword search method, we use a sub-optimal codeword selection criterion which minimizes the chordal distance (See (2-20)) between the chosen codeword and the ideal (un-quantized) optimal precoder (See (2-36)). Figure 4-3 shows the BER performance comparison between MSV-SC and the minimum chordal distance selection criterion. We can see that the two selection criteria have comparable performance.

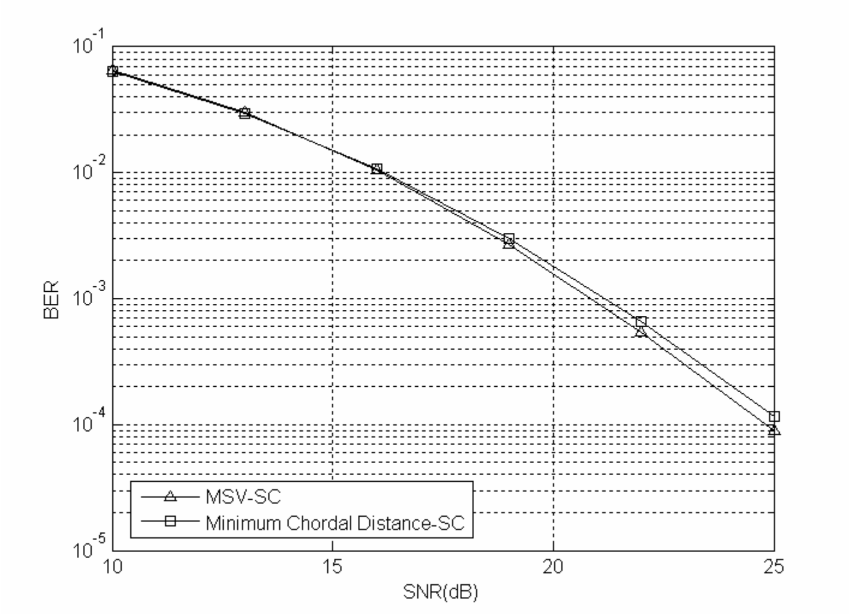

Figure 4-3 BER comparison for MSV-SC and minimum chordal distance-SC

Figure 4-4 shows the BER performance for the proposed codeword search method. The minimum chordal distance selection criterion is used, and the codebook size L is 64. As discussed in Section 2.5, increasing the search depth k will decrease the searching complexity, but the probability of codeword searching error will also increase. From Figure 4-4, we can see that the proposed codeword searching method with k = 3 has only about 1dB performance loss compared to exhaustive search. If we increase searching depth k to 4, the performance will further degrade about 1 dB.

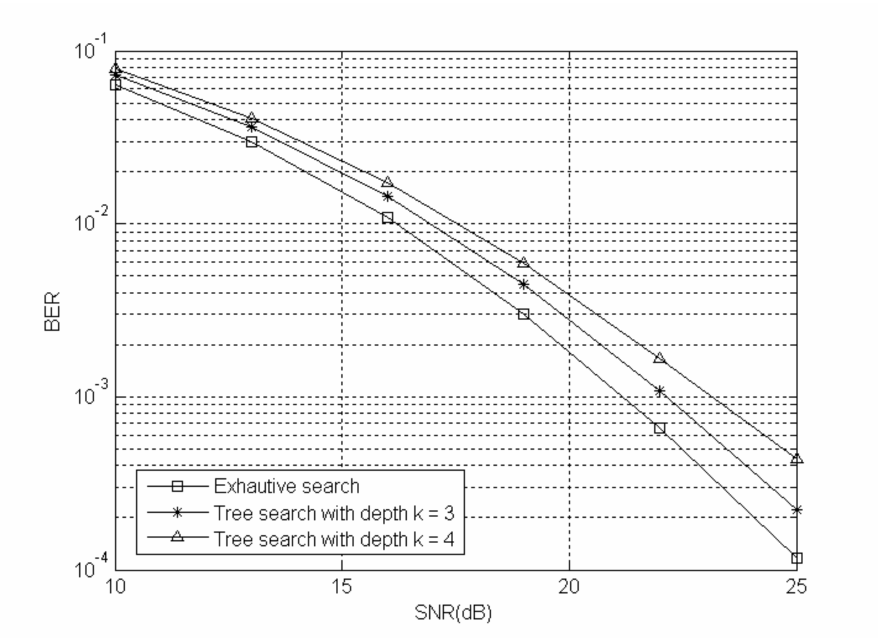

Figure 4-4 BER comparison between exhaustive search and tree search (L=64)

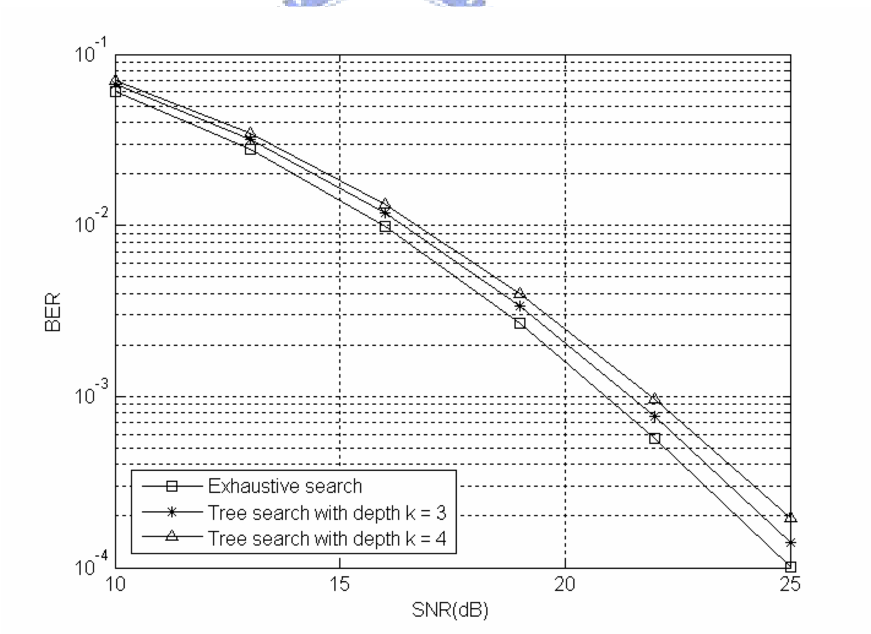

Figure 4-5 BER comparison between exhaustive search and tree search (L=128)

Figure 4-5 shows the BER performance for the proposed codeword search method with codebook size L = 128. Comparing Figure 4-4 and Figure 4-5, we can find that increasing the codebook size will decrease the probability of codeword searching error, and thus improve the performance.

As discussed in Section 2.5, using a modified codebook partition algorithm with the new factor ε, called overlap threshold, can lower the probability of codeword search error. Figure 4-6 shows the BER performance for the modified codebook partition algorithm with L=64, ε =0.05. Notice that ε =0 corresponds to original codebook partition algorithm. With the factor ε =0.05 , the performance of the proposed tree search algorithm will be comparable to the exhaustive searching scheme.

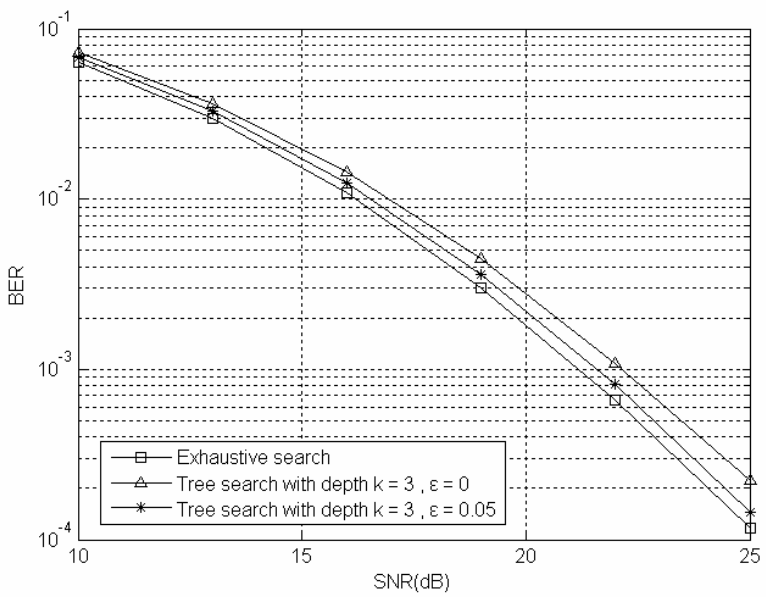

Figure 4-6 BER performance for modified codebook partition algorithm with ε =0.05 (L=64)

Figure 4-7 shows the BER performance for the modified codebook partition algorithm with L=128. The result is similar to the case for L = 64.

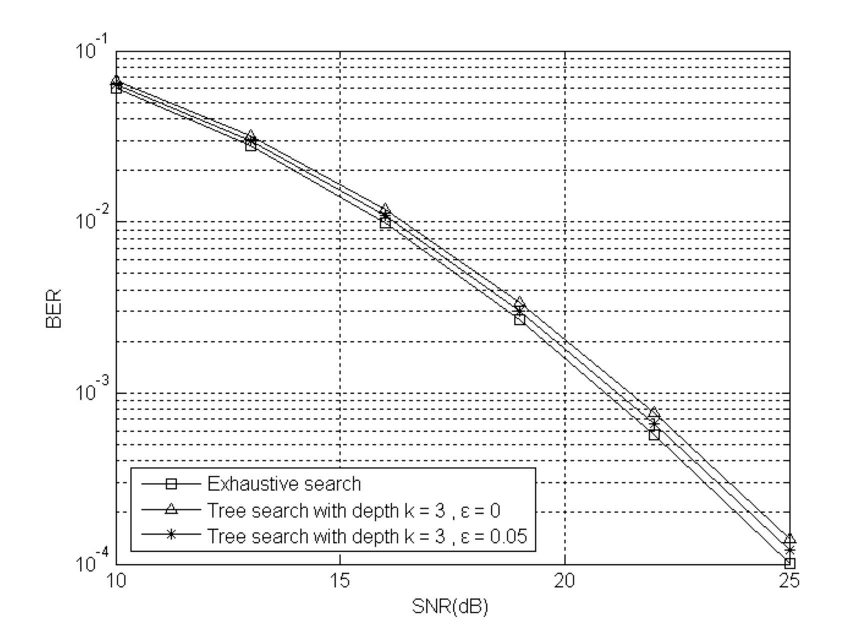

Figure 4-7 BER performance for modified codebook partition algorithm with ε =0.05 (L=128)

4.2 Time domain CSI feedback

In this section, we report some simulation results to evaluate the performance of the proposed time domain feedback methods described in Chapter 3. Although the channel information can be used in many transmitter processing schemes, we only consider the application of precoding. We use two different delay profiles for comparison:

1. Large delay spread → [1, 22, 23, 26, 51, 56]

2. Median delay spread → [1, 7, 13, 19, 22, 29]

To evaluate the efficiency of the proposed time-domain feedback algorithm, we calculate the total amount of feedback data at each frame. We give each LS or DCT parameters 5 bits for quantization. As the phase of each tap, it is given 3 bits for quantization, and the delay is 6 bits. Then, for feedback scheme with the first-order polynomial LS fitting, the magnitude information needs $5\times2\times(4\times2)$ = 80 bits, the phase information needs $3\times6\times(4\times2)$ = 144 bits, and the delay information requires $6\times6\times(4\times2)$ = 288 bits. Thus, total feedback data for the first-order polynomial LS scheme requires 512 bits. For DCT scheme with two parameters, the quantization bits required for magnitude can be reduced to 5×2 = 10 bits, and the total feedback data bits can be reduce to 442 bits.

We use the clustering technique for comparison. The cluster size is set as 8 and the codebook size is 64. So, each precoder index requires 6 bits. Assume that the number of active subcarriers is equal to the FFT size, then the total feedback data bits will be $(512/8)\times6$ = 384 bits. We summarize the required total feedback data bits for different schemes in table 4-1.

Scheme	LS	DCT	Clustering
bits	512	442	384

Table 4-1 Total feedback data bits for different schemes

Figure 4-8 shows the BER performance for time domain CSI feedback scheme with the LS method. A large delay spread model is chosen for simulations. We can see that the LS fitting scheme has about 2dB performance loss compared to precoding with perfect CSIT. Besides, increasing the order of the fitting curve will not give a substantial performance gain. This is because we quantize each phase with only 3 bits; although we have accurate magnitude, the quantization error for phase information still results in performance degradation.

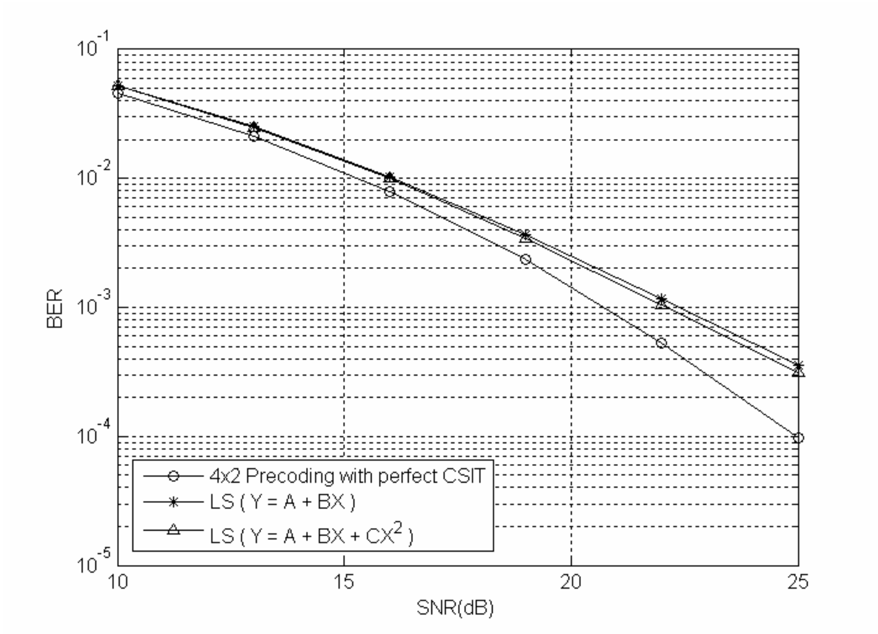

Figure 4-8 BER performance for the CSI feedback scheme with the LS method

Figure 4-9 shows the BER performance for time domain CSI feedback scheme with the DCT method. A large delay spread model is also chosen.

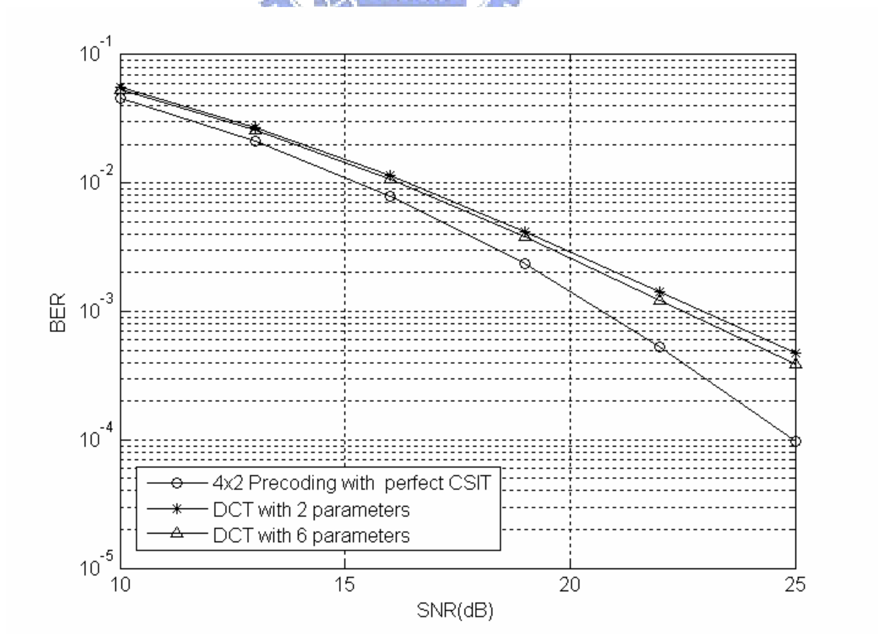

Figure 4-9 BER performance for the CSI feedback scheme with the DCT method

As we can see the DCT scheme has about 2.5 dB performance loss compared to precoding with perfect CSIT. As discussed above, increasing the number of extracted parameters of DCT will not give substantial performance gain.

Now we compare the time domain CSI feedback methods with the conventional precoder index feedback scheme, clustering. Figure 4-10 shows the BER comparison between different feedback schemes under large delay spread model. From Figure 4-10, we can see that clustering technique suffers significant performance degradation. This is because a large delay spread indicates a small coherent bandwidth, and the rapid variation of the frequency response. This will let the subcarriers near the cluster boundary suffer higher error probability.

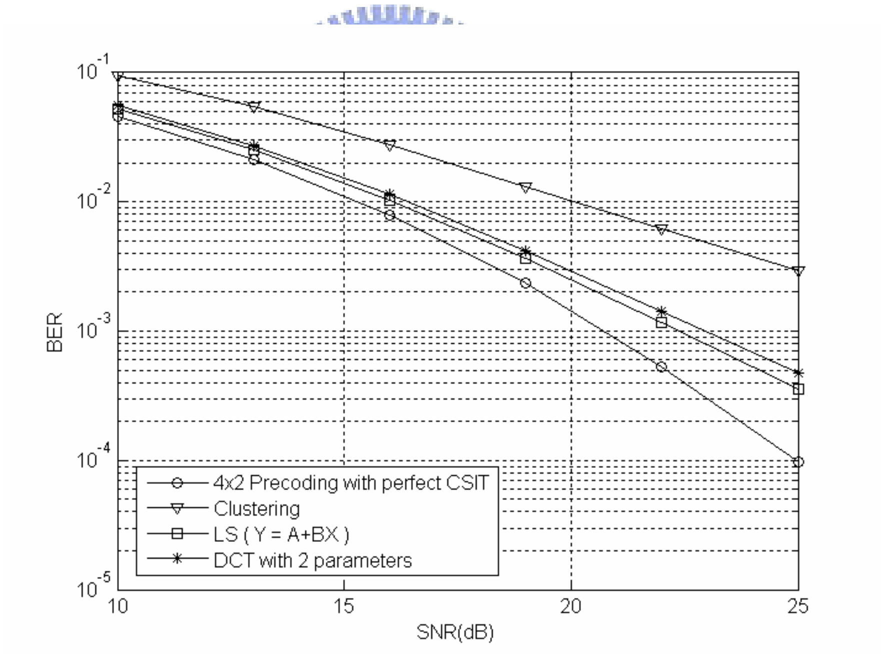

Figure 4-10 BER comparison under large delay spread for different feedback schemes

Figure 4-11 shows BER comparison between different feedback schemes under median delay spread model. The clustering technique has significant performance improvement. Besides, it is worthwhile to notice that different delay spread model will not affect the performance for time domain CSI feedback schemes.

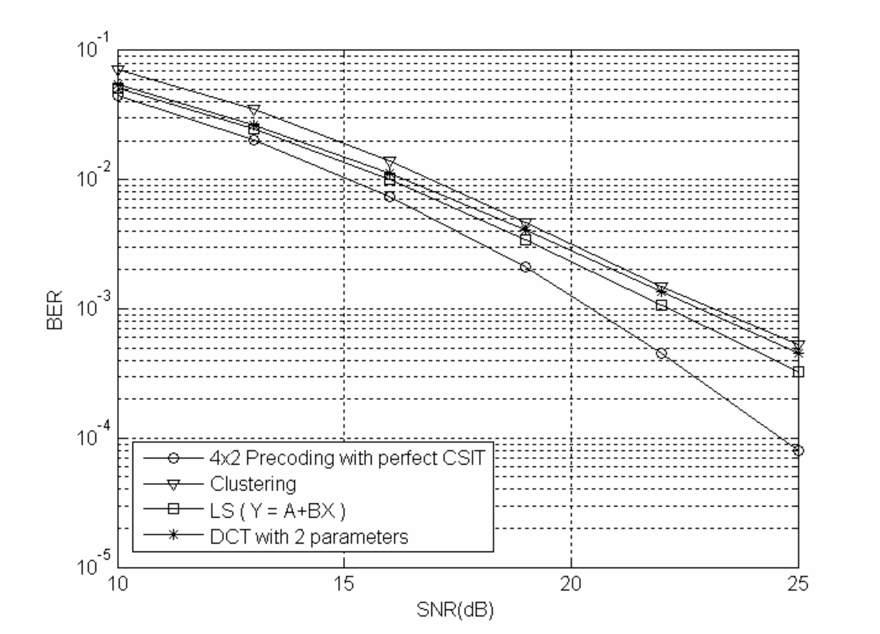

Figure 4-11 BER comparison under median delay spread for different feedback schemes

For time varying channel, a DPCM scheme can be included to further reduce the amount of feedback data for the proposed time domain CSI feedback methods. As mentioned, we use the spatial channel model (SCM) for the time-varying channel model. The relevant assumptions and settings have been described in Section 3.3. Since we assume that the delay for each tap will not change with time, only the magnitude and phase information will be fed back to the transmitter.

Figure 4-12 shows the BER performance of a 4 × 2 MIMO-OFDM precoding system with the proposed time varying CSI feedback scheme. Here, the LS method is combined with the DPCM scheme. In the simulation, a first-order polynomial LS fitting (Y = A+BX) is applied. For each frame, if we quantize the prediction error of one parameter with 2 bits and quantize the prediction error of one phase with 1 bit, the total amount of feedback data will be 2x2x(4x2) + 1x6x(4x2) = 80 bits. If we increase the quantization bits for one phase to 2 bits, the total amount of feedback data will become 128 bits.

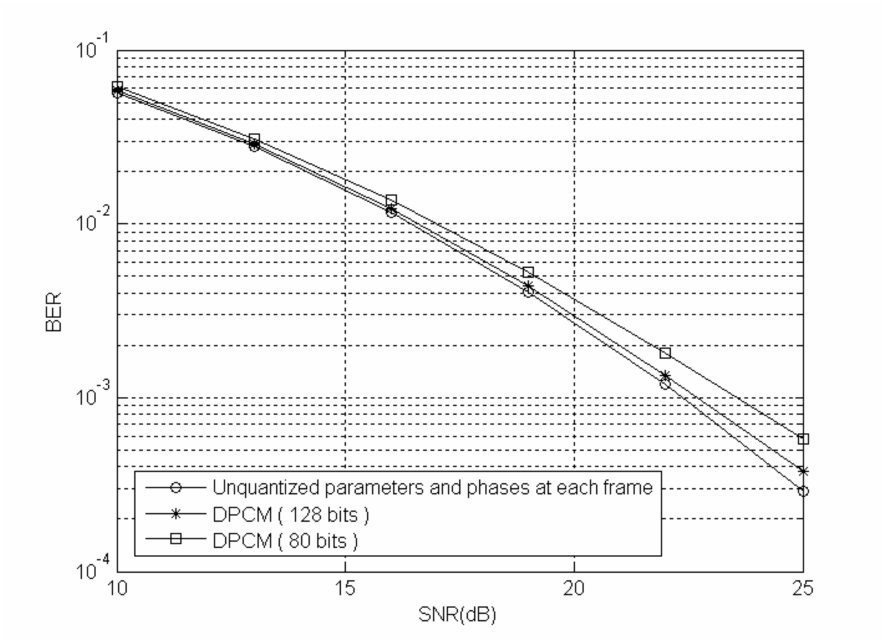

Figure 4-12 BER performance for time domain CSI feedback with LS and DPCM

Under the assumption of a slowly-varying channel (MS speed is 20km/hr), Figure 4-12 shows that the performance of the LS and DPCM combined scheme can approach to that of the un-quantized LS method. For the conventional PCM, if each parameter is quantized with 5 bits and each phase is quantized with 3 bits, the total amount will be 224 bits. From Figure 4-12, we can find that only 80~120 quantization bits are required with the DPCM scheme.

Chapter 5 Conclusions

In this thesis, we consider the precoder search and time domain CSI feedback problems. For the precoder search, we propose a low-complexity precoder searching algorithm, which consists of a codebook partition step and a codeword searching step. Compared to the exhaustive search, the proposed searching method can reduce about 80% searching complexity with acceptable performance loss. The performance of the proposed searching method can be further improved by modifying the codebook partition algorithm, but the complexity will also increase.

For time domain CSI feedback, we propose two methods for efficient feedback data compression. Under some channel conditions, the proposed method only requires a small amount of feedback data. Even in the application of precoding, our method is comparable to the conventional precoder feedback scheme such as clustering. For realistic time-varying channel, we also propose to use a differential pulse code modulation (DPCM) scheme in our method such that the required feedback data can be further reduced.

For precoding, we only consider how to find an optimal codeword within a constructed codebook. Directly designing and constructing an appropriate codebook in which the optimal codeword can be found fast and easily may serve as a potential research topic. For the CSI feedback, the proposed time domain method can work well in typical wireless channels. For channels with a lot of nonzero taps or large delay spread, our method may still result in a large amount of total feedback data. Thereby, how to exploit the spatial and time domain correlation and to well quantize the time domain MIMO channel response remains to be another potential topic for future research.

Reference

[1] David J. Love and Robert W. Health, Jr., “Limited feedback Unitary Precoding for Spatial Multiplexing Systems,” *IEEE Trans. Inf. Theory, vol.51, no.8, pp. 2967-2976, Aug. 2005.*

[2] A. Scaglione, P. Stoica, S. Barbarossa, G. B. Giannakis, and H. Sampath, “Optimal sesigns for space-time linear precoders and decoders,” *IEEE Trans. Signal Process., vol. 50, no.5, pp. 1051-1064, May 2002.*

[3] B. M. Hochwald, T.L. Marzetta, T. J. Richardson, W. Sweldens, and R. Urbanke, “Systematic design of unitary space-time constellations,” *IEEE Trans. Inf. Theory, vol. 46, no. 6, pp. 1962-1973, Sep. 2000.*

[4] Qinghua Li and Xintian Eddie Lin, “Compact feedback for MIMO-OFDM Systems over Frequency Selective Channels,” *Proc. IEEE VTC 2005-Spring, vol.1, pp.187–191, May/June 2005.*

[5] Jihoon Choi and Robert W. Health, Jr., “Interpolation based unitary precoding for spatial multiplexing MIMO-OFDM with limited feedback,” *IEEE Commun. Society Globecom, vol.1, Dec. 2004, pp.214 – 218.*

[6] Robert W. Health, Jr., S. Sandhu, and A. Paulraj, “Antenna selection for spatial multiplexing systems with linear receivers,” *IEEE Commun. Lett., vol.5, no.4, pp. 142-144, Apr. 2001.*

[7] L. Zheng and D. N. C. Tse, “Communication on the Grassmann manifold: A geometric approach to the noncoherent multiple-antenna channel,” *IEEE Trans, Inf, Theory, vol.48, no.2, pp. 359-383, Feb. 2002.*

[8] A. Barg and D. Y. Nogin, "Bounds on packings of spheres in the Grassmann manifold," *IEEE Trans. Inf. Theory, vol.48, no.9, pp. 2450-2454, Sep. 2002.*

[9] J. H. Conway, R. H. Hardin, and N. J. A. Sloane, "Packing lines, planes, etc.: Packings in Grassmannian spaces," *Exper. Math., vol.5, pp. 139-159, 1996.*

[10] Y. T. Lo and S. W. Lee, *Antenna Handbook: Theory, Applications, and Design*, New York: Van Nostrand Reinhold, 1988.

[11] D.J. Love.(2004) Tables of Complex Grassmannian Packings. [Online]. Available: http://www.ece.purdue.edu/~djlove/grass.html

[12] 3GPP TR 25.996, "Spatial channel model for Multiple Input Multiple Output (MIMO) simulations," *v6.1.0, Sept.2003*