LAPPED TRANSFORMS BASED ON DLS AND DLC BASIS FUNCTIONS AND APPLICATIONS

By

JINGYUN LI

B. Eng. Dalian Maritime University

M. Eng. McMaster University

A Thesis

Submitted to the School of Graduate Studies

in Partial Fulfilment of the Requirements

for the Degree

Ph. D.

McMaster University

To my parents, my husband and our daughter

LAPPED TRANSFORMS BASED ON DLS AND DLC BASIS FUNCTIONS AND APPLICATIONS

PH. D. (1997) MCMASTER UNIVERSITY
(Electrical and Computer Engineering) Hamilton, Ontario

TITLE: **Lapped Transforms Based on DLS and DLC Basis Functions and Applications**

AUTHOR: Jingyun Li
B. Eng. Dalian Maritime University
M. Eng. McMaster University

SUPERVISOR(S): Dr. P. Yip
Professor, Department of Mathematics and Statistics

NUMBER OF PAGES: xvii, 164

ABSTRACT

During the past decade, discrete block transforms have evoked considerable interest in the signal processing area. Discrete block transform is realized by first dividing the sample sequence of a signal to be processed into a series of blocks. The transform operation is then carried out by taking the inner products between the finite-length signal and a set of basis functions. This produces a set of coefficients which may then be further processed in compression, quantization, and encoding. Generally, block transforms can be divided into two types. One is called "nonlapped transform"; the other, "lapped transform". In the first system, the blocks are contiguous and no overlap (conjunction) is involved. The second system has overlapped operations: i.e., the data blocks may be overlapped.

In signal or image compression, it is well known that the non-lapped transforms produce an artifact called the "blocking effect" in the reconstructed signal or image. The blocking effect is partially due to the sharp cutoff of the data block. In lapped transforms where the data blocks overlap, the gradual decay of the basis functions in the overlapped region helps to reduce this blocking effect.

In this dissertation, we use a bell function to establish the discrete local sine (DLS) transform and the discrete local cosine (DLC) transform. The basis functions of this general type have a smooth cutoff and are shown to have orthogonality properties. The orthonormality and lapped orthogonality of DLS/DLC transforms indicate that DLS and DLC belong to the family of the lapped block transform system. We derive sufficient and necessary conditions for perfect reconstruction for lapped block transforms based on DLS/DLC basis functions, although, in general, lapped transform matrices are not unitary. These conditions are useful for designing optimum lapped block transforms in special applications. For example, in image and speech coding, we can design an optimal lapped transform system with the perfect signal reconstruction property.

Other properties of the transforms are also discussed. These include scaling-in-time, shifting-in-time, difference-in-time, the uniqueness property, and the convolution property. Fast algorithms for implementing DLS and DLC are developed. Based on Given's rotations and butterfly operations, we decompose an Mth-order DLS or DLC with length $M + L$ into a set of sparse matrices, where L is the length of the overlapped region. Because of the partially recursive nature of the structure, DLS and DLC fast algorithms can be implemented with parallel processors.

As examples, we consider two applications using lapped transforms. The first application is in acoustic echo cancellation. We use block transforms to implement a subband acoustic echo canceller. However, due to the frequency aliasing problem in a filter bank system [36], the direct application of block transforms in an echo canceller does not function very well. We propose an improved method. By changing the subsampling rate in block transforms, echo residuals can be reduced significantly. Moreover, we develop an optimum lapped transform using a criterion of maximum energy concentration. With the optimum designed lapped transform, an obvious performance improvement can be observed from echo suppression. Image compression [65], a tool for efficiently encoding a picture (two dimensional data), is the second application of lapped orthogonal tranforms discussed in this thesis. We develop a new sub-optimal lapped transform based on DLS and DLC for image coding. The results are compared with those obtained from using the lapped orthogonal transform (LOT) and modulated lapped transform (MLT). Computer simulation results show significant improvement in the reduction of block effects.

Acknowledgement

The author wishes to express sincere gratitude to Dr. P.Yip for his constant encouragement, continued assistance and expert guidance and supervision; to supervisory committee members Dr. K.M. Wong and Dr. J. Litva for their encouragement, supervison and support, throughout the course of this work; to the Communication Research Laboratory (CRL) and staff, for providing good facilities.

It is the author's pleasure to acknowledge the inspiring discussions with her colleague Y. Zhou. Many thanks go to other friends for their valuable help. Especially, I would like to express my thanks to Mrs. Ruth Katz who uses her spare time to check my thesis in English. I would also like to acknowledge the financial support through Dr. K.M. Wong provided by Telecommunication Research Institute of Ontario and Department of Electrical and Computer Engineering (McMaster University).

Finally, I am indebted to my husband, Qu Jin, for his constant understanding, encouragement, assistance and support.

Contents

List of Figures

List of Tables

List of Short Terms and Symbols

AR: auto-regressive

bpp: bits per pixel

DCT: discrete cosine transform

DCT-IV: discrete cosine transform type IV

DFT: discrete Fourier transform

DLC: discrete local cosine

DLS: discrete local sine

DST: discrete sine transform

EPE: energy packing efficiency

HT: Hāar transform

KLT: Kahunan-Loéve transform

LOT: lapped orthogonal transform

MLT: modulated lapped transform

PR: perfect reconstruction

ST: slant transform

WHT: Walsh-Hadamard transform

$\boldsymbol{\Phi}$: transform matrix

$\boldsymbol{I}$: identity matrix

$\tilde{\boldsymbol{I}}$: reverse identity matrix

$\overline{\boldsymbol{I}}$: a matrix defined by $\overline{\boldsymbol{I}} = \begin{bmatrix} -\boldsymbol{I} & \vdots & \boldsymbol{O} \\ \cdots & \vdots & \cdots \\ \boldsymbol{O} & \vdots & \boldsymbol{I} \end{bmatrix}$

$\boldsymbol{O}$: zero matrix

$\boldsymbol{W}$: shift matrix

$\boldsymbol{G}$: Given's rotation matrix

$\boldsymbol{B}$: butterfly matrix

$\boldsymbol{S}_M$: $M \times M$ DST-IV matrix

$\boldsymbol{C}_M$: $M \times M$ DCT-IV matrix

$\boldsymbol{x}$: input signal

$\boldsymbol{X}$: transform of input signal

$\hat{x}$: reconstructed signal

μ: number of multiplication

$\mathcal{A}$: number of addition

Chapter 1

Introduction

1.1 Discrete Transform Signal Processing

During the past decade, discrete block transforms have evoked considerable interest in the signal processing area. Some typical transforms are discrete Fourier transform (DFT), discrete cosine transform (DCT) [66], Walsh-Hadamard transform (WHT) [62], Haar transform (HT) [2] and slant transform (ST) [63], as well as block transforms with optimum performance, such as the Kahunan-Loéve transform (KLT). All these transforms can be viewed as "cutoff block transforms", so named because the data sequence is processed in contiguous blocks with no overlap. The transform operation is carried out by taking the inner products between the finite-length signal and a set of basis functions [1]. As a result, a set of coefficients is produced, which are then passed on for further processing such as compression, quantization, and encoding, [34] [64] [16] [58]. In practice, many of these transforms have efficient implementations as cascades of "butterfly" computations [66] [78] [23] [45] [15] [25] [26]. They have found applications in such diverse areas as speech and image coding [18] [27], data compression [22], pattern recognition [33], and classification [21], to name just a few.

Most discrete block transforms represent a signal on a set of subbands in the frequency domain [75]. They become a convenient tool for analysis and treatment of a set of data. The relation between transform coding and subband coding can be found in [65], where subband coding is based on multirate filter banks [20] [7] and transforms use matrix operations [66]. We can treat a block transform as a set of decimated filter banks

[19] and the inverse transform as a set of expanded filter banks. A transform and its inverse correspond to the impulse responses of analysis filters and synthesis filters in the multirate filter banks [50] respectively. On the other hand, a filter bank can be viewed as a transform with overlapped operation in which the transform of a block obtains not only the data in that block but also the data in its neighboring blocks [76]. This relation allows for easy signal processing in the transform domain when using multirate filter banks [20].

Both block transforms and filter banks can be used in signal coding problems [65]; but each method has its own characteristics. In data compression, block processing causes discontinuities (nonsmoothness) between adjacent blocks in the reconstructed signal. The discontinuities are referred to as blocking effects [51], [53]. In image processing, the blocking effect results in a reconstructed image that seems to be built up by small tiles across block boundaries, whereas in speech coding the blocking effect results in audible periodic tones [72]. In using filter banks, the filter can have many tap weights, and it is possible to build a filter in which the impulse responses decay smoothly to zero. The blocking effects are thus reduced [71]. However, the development of such systems is not easy. We need to design a long filter, because the longer the filter is, the less is the boundary effect. In real time processing, a fast computation operation for filter bank does not exist. In addition, the filter bank works well only for a small number of subbands because high computational complexity is involved in the design of perfect reconstruction M-bank filter banks [35]. A perfect reconstruction system is defined as the one in which the output signal (or the reconstructed signal) is a delayed copy of the original signal [72].

Based on the inter-relation between transforms and filter banks, we are able to develop a flexible time-frequency transform. Such a transform has not only good frequency resolution but also good time resolution. References to the main issues in this area can be found in [49], [6], [48] and [80]. Cosine modulated systems with perfect reconstruction property have been proposed recently from different approaches [52] [54] [42] [43] [76]. All of them belong to a set of "uniform paraunitary filter banks" [72] [76]. Lapped Orthogonal Transform (LOT) [49] (a special case of the lapped block transform systems), modulated lapped transform (MLT) [52] and local sine transform [6] belong to the family

of lapped block transforms, in which a signal is processed in each block, and only two adjacent blocks are involved in the computation. As a result, blocking effects can be reduced, and fast algorithms may be developed. Wavelet and wavelet packet transforms are free from the blocking effect [77], but they require more intense computation than some block transforms such as DCT and DST. In this thesis, we will focus on lapped block transforms.

1.2 Major Contributions of Thesis

Block transform and filter bank are closely related, but each may have their own characteristics. Transform techniques are usually based on orthogonal linear transforms. Because the basis functions are mutually orthogonal and the blocks are chosen so that they do not overlap, the coders employing the block transforms are typically classified as transform coders [65]. The length of each basis function in a block can be chosen as short as desired. For N basis functions, the shortest length in a block transform is N. Most block transforms take the shortest length for their basis functions. However, because of the abrupt "cutoff" at the boundary, some undesirable artifacts, such as the blocking effect, may appear in the reconstructed signal. On the other hand, filter banks process the signal by convolving the input signal with a set of bandpass filters and then decimating the result. Each decimated subband signal contains the information of the signal in a particular portion of the frequency domain. To recover the signal, the subband signals are upsampled, filtered, and re-combined. Thus, coders using filter banks are essentially subband coders. Because of the absence of blocking effects, subband coding is still a widely used technique in signal processing. Looking at the blocking effect in transform coding, one may wish to merge filter banks with block transforms to alleviate the abrupt change at the boundary. The cosine modulated system (also called as MLT), which is the expanded LOT, is one such scheme [52]. Local sine and cosine basis functions [17] [6] based on a selected bell function are another type of smooth cutoff system. Because an arbitrarily smooth cutoff is involved, transforms based on local sine and cosine basis functions may be considered as "smooth block transforms". Their basis functions are orthogonal to each other, and not only do they reduce blocking effects, but they also

improve subband localization.

In this thesis, using the bell function proposed originally in [6], we derive a discrete local sine (DLS) transform and a discrete local cosine (DLC) transform. The basis functions which involve arbitrarily smooth cutoffs are shown to be orthogonal. In addition, we prove that the basis functions are orthogonal in the overlapped portion - meaning that the cross terms will not introduce errors when the data is reconstructed. By definition, DLS and DLC belong to the family of lapped orthogonal transforms which satisfy lapped orthogonal property. It is easily seen that the lapped transform cannot be unitary because the matrix of such transform is no longer square. Despite the lack of unitarity for the transform matrix, we have derived necessary and sufficient conditions for the perfect reconstruction by examining the product of the lapped transform matrix and its transpose. These conditions will be used in the design of optimum lapped block transforms. In image and speech coding, a best transform system can be designed with perfect signal reconstruction property. In this thesis, other properties of transforms are discussed, such as scale-in-time and shift-in-time properties, uniqueness property, difference-in-time and convolution properties.

In addition to transform properties, fast algorithms for the general types of DLS and DLC are considered and developed. These algorithms make DLS and DLC more suitable for parallel processing. After derivations of properties and algorithms, we focus on applications. In this thesis, we consider two applications of lapped transforms. The first application is acoustic echo cancellation. We use block transforms to implement the subband acoustic echo canceller. However, due to the frequency aliasing problem in filter bank system [36], direct application of block transform in an echo canceller may not work very well. In order to increase the echo suppression, we propose an improved method. By changing the subsampling rate in the block transforms, the echo residuals can be reduced significantly. Furthermore, we develop an optimum lapped transform by using a criterion with maximum energy concentration. Using the optimum lapped transform, an obvious improvement in echo suppression can be observed. Image compression [65], as a tool for efficiently encoding picture (two dimensional) data, is the second application of lapped orthogonal transforms discussed in this thesis. Through the transform, the data are compressed by discarding many other coefficients with lower energy. Comparing

with LOT and MLT, DLS and DLC are applied for image compression here. By further developing a new sub-optimal lapped transform with the highest decorrelation of the AR-model, the corresponding performance in image compression again indicates that such transform reduces the blocking effect significantly.

1.3 Outline of Thesis

The thesis is organized as follows. The first chapter briefly describes the history of discrete transform signal processing and states the problems to be solved in this thesis. In the second chapter, we review the multirate filter banks and their relations to lapped transform systems, followed by the development of the DLS and DLC transforms. Operational properties of these transforms are developed in Chapter 3. In the fourth chapter, we develop a fast algorithm for implementing a general type of DLS and DLC transforms. Applications — acoustic echo cancellation and image compression of lapped transforms — are presented in Chapters 5 and 6, respectively. The final chapter concludes the thesis with a summary of the contributions and some suggestions for further research.

Chapter 2

Multirate Filter Banks and Lapped Orthogonal Transforms

In this chapter, we first review the fundamentals of multirate signal processing such as decimation, interpolation and perfect reconstruction of filter banks. The relation between filter banks and block transforms is then discussed in Section 2.2. In Section 2.3, lapped orthogonal transform is introduced. Discrete local sine transform and discrete local cosine transform are proposed in Section 2.4.

2.1 Multirate Filter Banks

In multirate digital signal processing, a signal may be processed with different rates at the same time. This is different from the conventional digital signal processing system in which only a single sampling rate is used. The multirate system basically uses two operations, decimation and interpolation. We will now describe these two operations.

2.1.1 Decimation Operation

The decimation operation is also called downsampling or subsampling. A decimator retains sample points spaced only at certain distances. Consider an input discrete signal sequence $x(n)$ with a sampling period T. The process of reducing the sampling rate of $x(n)$ by an integer factor M can be described as follows: only those sample points of

$x(n)$ at every M-th period are retained; i.e.,

$$\frac{T_{new}}{T} = M, \tag{2.1}$$

where T_{new} is new sampling period. The new sampling rate is obtained by

$$F_{new} = \frac{1}{T_{new}} = \frac{1}{M\,T} = \frac{F}{M}, \tag{2.2}$$

where F is the old sampling rate. The output $v(n)$ of this process is given by

$$v(n) = x(M\,n). \tag{2.3}$$

It is easily seen that information will be lost after decimation. Let

$$x'(n) = \begin{cases} x(n) & n = 0, \pm M, \pm 2M, \cdots \\ 0 & otherwise, \end{cases} \tag{2.4}$$

which can also be written as

$$x'(n) = x(n)\left[\frac{1}{M}\sum_{l=0}^{M-1} e^{j2\pi ln/M}\right]. \tag{2.5}$$

Thus, we have

$$v(n) = x(Mn) = x'(Mn). \tag{2.6}$$

Now, let us analyze the process of decimation in the transform domain. Taking z-transform ($z = e^{j\omega}$) on both sides of (2.6) and letting $V(z)$ be z-transform of $v(n)$, we have

$$\begin{aligned} V(z) &= \sum_{n=-\infty}^{\infty} v(n)z^{-n} \\ &= \sum_{n=-\infty}^{\infty} x'(Mn)z^{-n} \\ &= \sum_{n=-\infty}^{\infty} x'(n)z^{-\frac{n}{M}} \\ &= \sum_{n=-\infty}^{\infty} x(n)\left[\frac{1}{M}\sum_{l=0}^{M-1} e^{j2\pi ln/M}\right] z^{-\frac{n}{M}} \\ &= \frac{1}{M}\sum_{l=0}^{M-1}\sum_{n=-\infty}^{\infty} x(n)\left[W^l z^{\frac{1}{M}}\right]^{-n} \\ &= \frac{1}{M}\sum_{l=0}^{M-1} X(W^l z^{\frac{1}{M}}), \end{aligned} \tag{2.7}$$

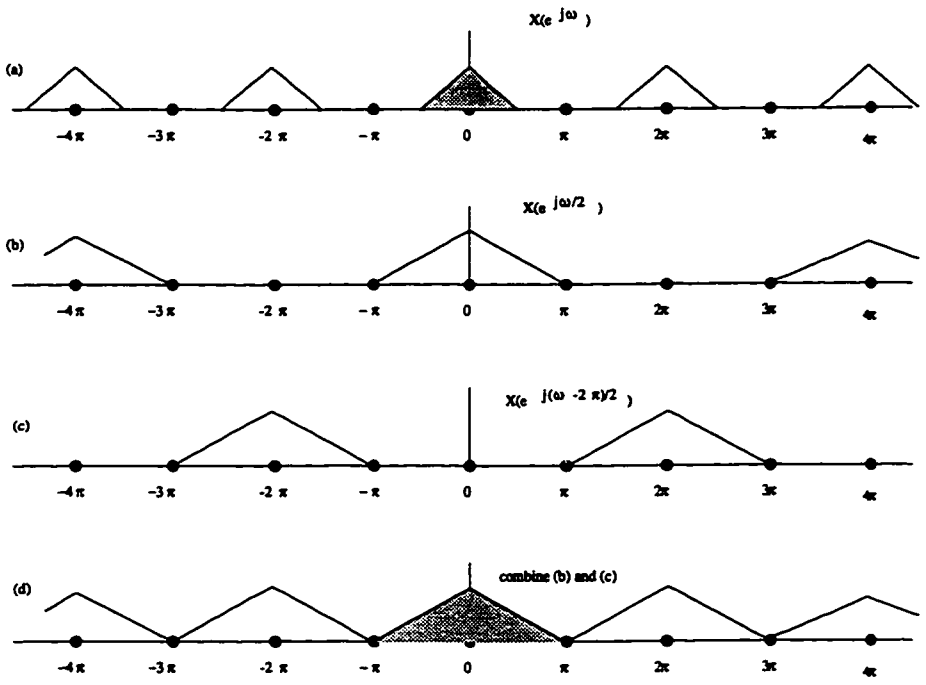


Figure 2.1: Frequency-domain effect of decimation with $M = 2$

where $W = e^{-j\frac{2\pi}{M}}$ and $X(z)$ is z-transform of $x(n)$. The effect of decimation in the frequency domain can be illustrated by considering the special case where $M = 2$ and $x(n)$ is a low-pass signal bandlimited at $\frac{\pi}{2}$. (2.7) becomes

$$V(z) = \frac{1}{2}\sum_{l=0}^{1} X(W^l z^{\frac{1}{2}}) = \frac{1}{2}\sum_{l=0}^{1} X(e^{j\frac{\omega-2\pi l}{2}}). \tag{2.8}$$

Figure 2.1 shows the decimation process for this special case. Furthermore, if $x(n)$ is low-pass signal bandlimited at $\frac{3\pi}{4}$, then after downsampling by 2, the spectrum is extended and overlapped. As a result, $x(n)$ cannot be perfectly recovered by $v(n)$ (shown in figure 2.2). This indicates that if the input signal is not a low-pass signal, or even if it is low-pass but the bandwidth is larger than $\frac{\pi}{2}$, then after downsampling by 2, $x(n)$ cannot be perfectly recovered from $v(n)$. For a general case, if $x(n)$ is not a low-pass signal at a region $|\omega| < \frac{\pi}{M}$ and if we want to recover the input signal from $v(n)$, the reconstructed signal may not hold the same information as the original (Detailed reconstruction procedures will be discussed in the next subsection). This problem is

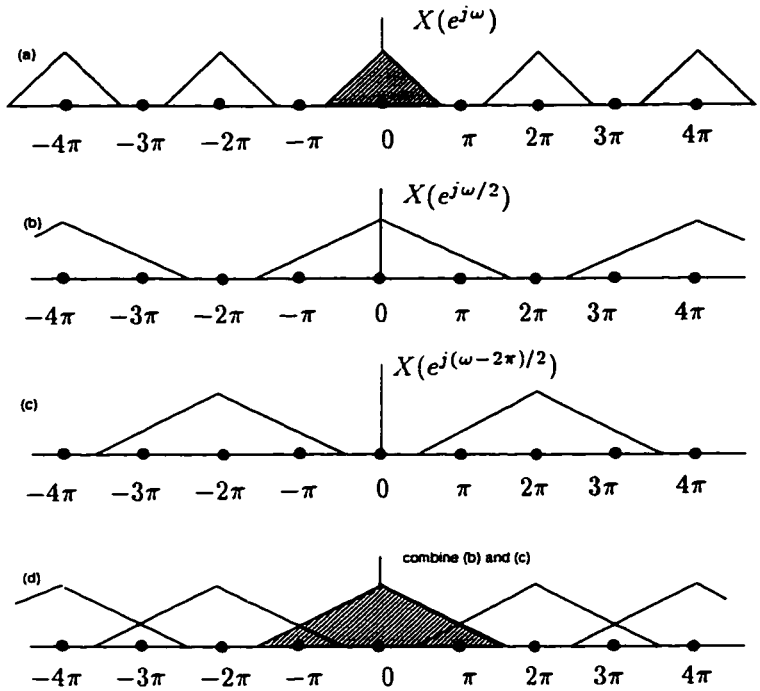


Figure 2.2: Frequency-domain effect of decimation with $M = 2$ and signal bandwidth larger than $\frac{\pi}{2}$

commonly called aliasing. In order to avoid aliasing at this lower sampling rate, a low-pass filter has to be added before the decimation operation. Such a system is shown in Figure 2.3. The circle with the downward pointing arrow in this diagram indicates that the output sequence of the filter is subsampled; i.e., only every M-th sample is retained. If we denote the impulse response of the filter as $h(n)$, we then have

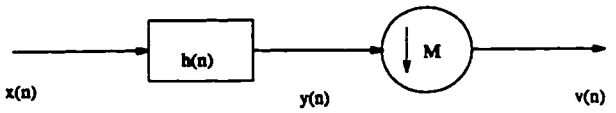
$$y(n) = \sum_{k=-\infty}^{\infty} h(k)x(n-k), \tag{2.9}$$

where $y(n)$ is the output of the filter. Thus, after downsampling, the output $v(n)$ is obtained as

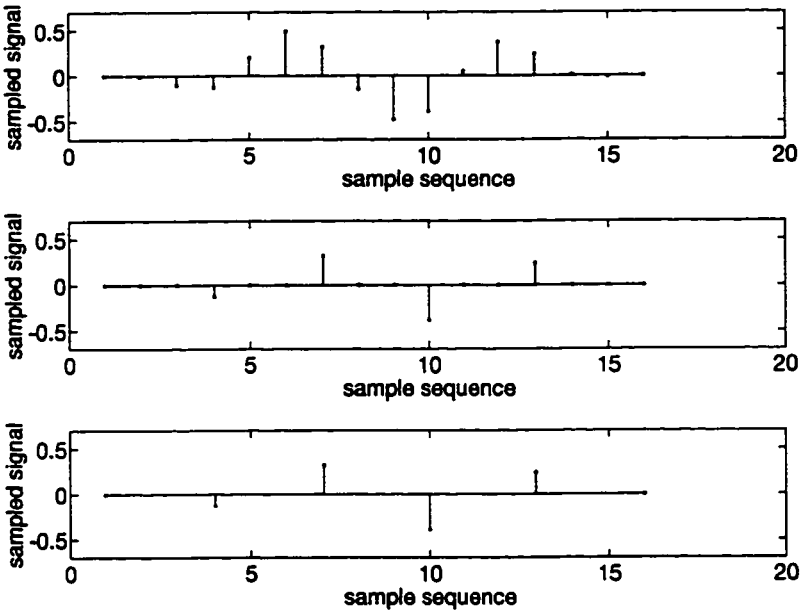
$$v(n) = y(Mn). \tag{2.10}$$

Combining (2.9) with (2.10), we have

$$v(n) = \sum_{k=-\infty}^{\infty} h(k)x(Mn-k). \tag{2.11}$$



(a)



(b)

Figure 2.3: (a). Subsampling system with low-pass filter. (b). Input and output waveforms. The first graph shows the signal $x(n)$; the second shows the signal $x'(n)$; and the last is the signal $v(n)$. Here, $M = 3$.

Taking z-transform on both sides of (2.11), and applying the results in (2.7), we have

$$V(z) = \frac{1}{M} \sum_{l=0}^{M-1} H(W^l z^{\frac{1}{M}}) X(W^l z^{\frac{1}{M}}), \tag{2.12}$$

where $H(z)$ is z-transform of $h(n)$. This is the result of low-pass filtering and decimation. The combination is sometimes called a decimation filter. A system of decimation filters consists of a number of properly chosen low-pass filters and decimators, such that aliasing can be reduced or even eliminated. The choice of filters depends on the perfect reconstruction property which implies that the reconstructed signal must be a delay copy of the original signal [72]. We will address this property in Section 2.1.3.

2.1.2 Interpolation Operation

The interpolation operation is also called upsampling. As opposed to subsampling, upsampling by M consists of placing $M-1$ zeros between every two consecutive samples of the input signal. Thus, we have

$$\frac{T_{new}}{T} = \frac{1}{M}. \tag{2.13}$$

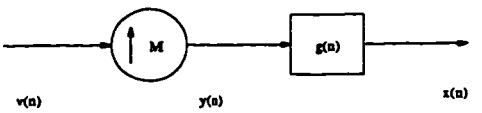
The new sampling rate is

$$F_{new} = \frac{1}{T_{new}} = \frac{M}{T} = M\ F. \tag{2.14}$$

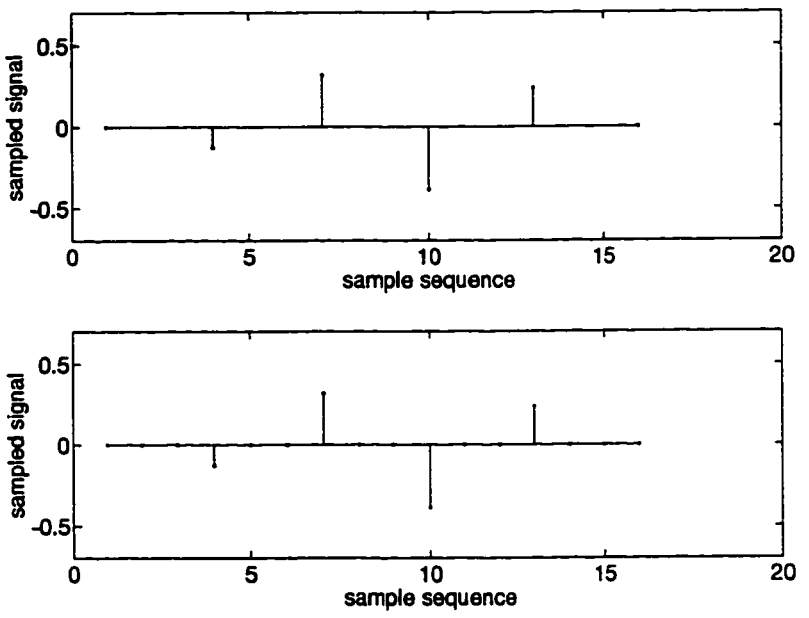
Obviously, interpolation is an inverse operation of decimation. However, no information is lost in interpolation and the original signal can be simply recovered by the process: subsampling by M. An interpolation filter can be described with the help of Figure 2.4. The circle with the upward pointing arrow in the diagram indicates that the input sequence is upsampled: i.e., $M-1$ zeros are inserted between two neighbouring sample points. If $v(n)$ is the input to the interpolator, $y(n)$ is the output of $v(n)$ after the upsampling process, and $x(n)$ is the output of the filter, and if the impulse response of the filter is $g(n)$, we then have

$$y(n) = \begin{cases} v(\frac{n}{M}) & m = 0, \pm M, \pm 2M, \cdots, \\ 0 & otherwise, \end{cases} \tag{2.15}$$

In the z-transform domain, we have



(a)



(b)

Figure 2.4: (a). Upsampling system with low-pass filter. (b). Input and output waveforms. The top shows the signal $v(n)$ and the bottom shows $x(n)$. Here $M = 3$.

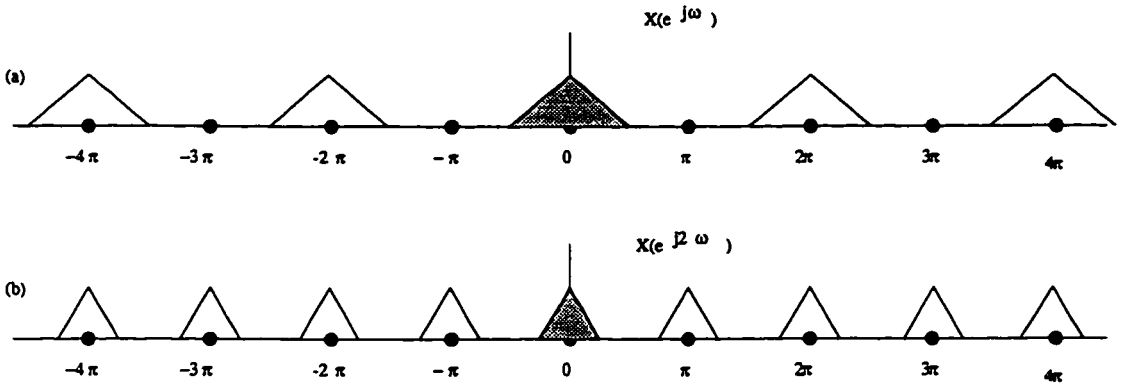


Figure 2.5: Frequency-domain effect of interpolation operation with $M = 2$

$$\begin{aligned} Y(z) &= \sum_{n=-\infty}^{\infty} y(n)z^{-n} \\ &= \sum_{n=-\infty}^{\infty} v(\frac{n}{M})z^{-n} \\ &= \sum_{n=-\infty}^{\infty} v(n)z^{-nM} \\ &= V(z^M), \end{aligned} \tag{2.16}$$

and the filter output is

$$X(z) = G(z)Y(z) = G(z)V(z^M). \tag{2.17}$$

Here, $G(z)$ is the z-transform of $g(n)$. (2.16) shows that in the frequency domain, after upsampling by M, the original spectrum $V(e^{j\omega})$ is compressed as $V(e^{jM\omega})$ and a set of copies, called images, is created. Figure 2.5 demonstrates this result. The compression in the frequency domain causes no information loss and the original signal can easily be recovered by using a low-pass filter $G(z)$ in (2.17).

2.1.3 Perfect Reconstruction for Filter Banks

With both decimation and interpolation in one system, a multirate filter bank can be constructed as shown in Figure 2.6. Block I consists of a set of band-pass filters and associated subsamplers. The input signal is decomposed into subband signals or sub-signals. Block II consists of upsamplers and the associated band-pass filters, provided

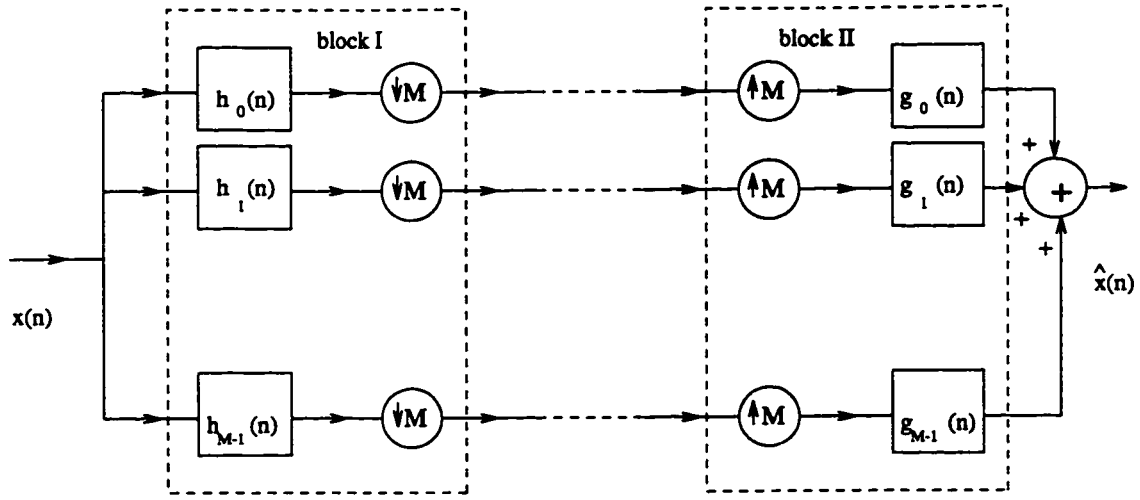


Figure 2.6: Subband coding system with M subbands

for the reconstruction of the original signal from its subband decomposition. Between Block I and Block II, additional processing such as noise suppression, quantization, compression or coding and decoding can be applied in each subband. Without these additional processings, the subband system should perfectly reconstruct the original signal in Block II. We now derive the "perfect reconstruction" condition imposed on the structures of the filters in this multirate filter bank.

Let $\hat{X}(Z)$ be the z-transform of $\hat{x}(n)$, the output of the multirate filter bank. From (2.12) and (2.17), we get

$$\begin{aligned} \hat{X}(z) &= \sum_{i=0}^{M-1} G_i(z) V_i(z^M) \\ &= \frac{1}{M} \sum_{l=0}^{M-1} [\sum_{i=0}^{M-1} G_i(z) H_i(W^l z)] X(W^l z). \end{aligned} \tag{2.18}$$

(2.18) can be rewritten as

$$\hat{X}(z) = \frac{1}{M}[\sum_{i=0}^{M-1} G_i(z) H_i(z)] X(z) + \frac{1}{M} \sum_{l=1}^{M-1} [\sum_{i=0}^{M-1} G_i(z) H_i(W^l z)] X(W^l z). \tag{2.19}$$

The first term in (2.19) is the distortion term, and the second is the aliasing term. To have perfect reconstruction (PR), the following conditions must be satisfied [72]:

$$\sum_{i=0}^{M-1} G_i(z) H_i(z) = M, \tag{2.20}$$

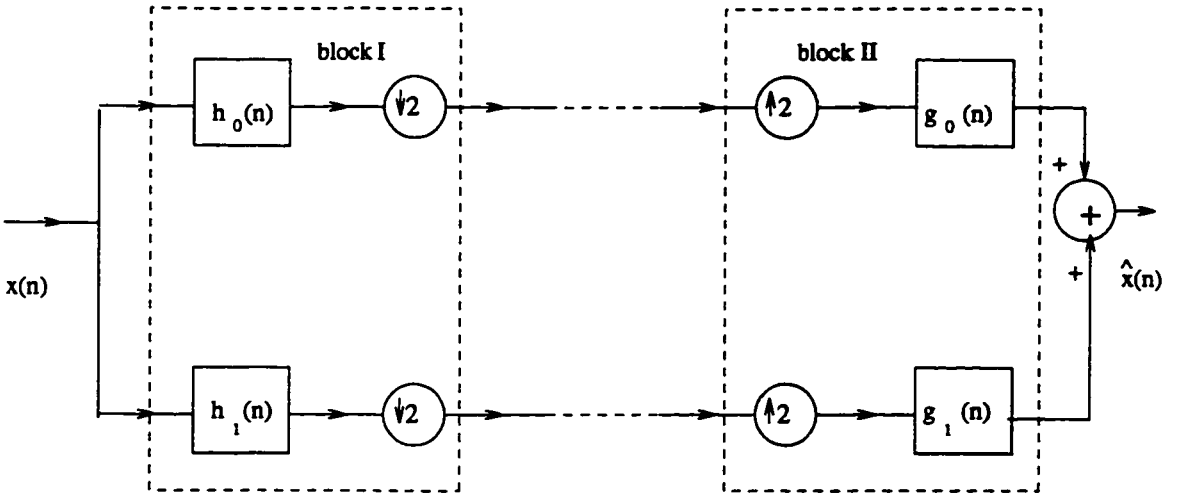


Figure 2.7: Subband coding system with 2 subbands

and

$$\sum_{i=0}^{M-1} G_i(z)H_i(W^l z) = 0\ ; \quad l = 1, \cdots, M-1. \tag{2.21}$$

For the special case of $M = 2$, as shown in Figure 2.7, (2.20) and (2.21) can be reduced to

$$\begin{cases} G_0(z)H_0(z) + G_1(z)H_1(z) = 2, \\ G_0(z)H_0(-z) + G_1(z)H_1(-z) = 0. \end{cases} \tag{2.22}$$

For the system shown in Figure 2.7, we first design $H_0(e^{j\omega})$ as a low-pass filter, and $H_1(e^{j\omega})$ as a high-pass filter with its frequency response symmetric to that of $H_0(e^{j\omega})$: i.e. $H_1(e^{j\omega}) = H_0(e^{j(\omega+\pi)})$ [71]. We then choose $G_0(e^{j\omega}) = H_0(e^{j\omega})$, and $G_1(e^{j\omega}) = -H_1(e^{j\omega})$. With this choice, the aliasing term is cancelled, and the multirate filter bank system is a perfect reconstruction system if $|H_0(e^{j\omega})|^2 + |H_0(-e^{j\omega})|^2 = 2$. Increasing the number of subbands in the filter bank means that more complicated procedures are required to design a PR system, as (2.20) and (2.21) become harder to satisfy. We may conclude that filter banks work well only on a small number of subbands because of the computational complexity of implementation in the perfect reconstruction requirement of filter banks. Another difficulty in real-time processing is the requirement for filters of fewer tap weights. Coupled with the PR condition, it implies that a fast computation operation may not easily be developed. This is one of the reasons for considering block

transforms instead of filter banks in subband processing.

2.2 Relation between Filter Banks and Block Transforms

Before talking about the relation between filter banks and block transforms, let us look at the block transform system. The purpose of block transform processing is to divide the sampled sequence of a signal into a series of blocks (Figure 2.8 shows two blocks, each with length 8, in a cutoff block system), and then apply transform on each individual blocked signal.

Consider a sample sequence of a real data $x(n)$, $n = 1, \cdots, K$. This set of data is divided into blocks and each block includes M samples. Let $\boldsymbol{x}_m$ represent data at the m-th block: i.e., $\boldsymbol{x}_m = [\tilde{x}_m(0), \tilde{x}_m(1), \cdots, \tilde{x}_m(M-1)]^T$ and $\tilde{x}_m(i) = x(mM+i)$; $m = 0, 1, \cdots, \frac{K}{M}$, assuming that K is an integer multiple of M. A transform matrix $\boldsymbol{\Phi}$ is defined as

$$\boldsymbol{\Phi} = [\boldsymbol{\phi}_0 \boldsymbol{\phi}_1 \cdots \boldsymbol{\phi}_{M-1}]; \tag{2.23}$$

with $\boldsymbol{\phi}_i$ being the i-th basis function of the transform, which is a real column vector with dimension M. We assume that $\{\boldsymbol{\phi}_i\}$ are orthonormal: i.e.,

$$\boldsymbol{\phi}_i^T \boldsymbol{\phi}_j \equiv \delta_{ij}. \tag{2.24}$$

Here T denotes the transpose operation. δ_{ij} is the Kronecker Delta function: i.e., $\delta_{ij} = 1$ when $i = j$, and $\delta_{ij} = 0$ for $i \neq j$. Furthermore, we have that $\boldsymbol{\Phi}$ is a unitary matrix: i.e.,

$$\boldsymbol{\Phi}\boldsymbol{\Phi}^T = \boldsymbol{\Phi}^T\boldsymbol{\Phi} = I, \tag{2.25}$$

where I is an identity matrix. Then the transformed coefficients of $\boldsymbol{x}_m$ are given by

$$\boldsymbol{X}_m = \boldsymbol{\Phi}^T \boldsymbol{x}_m. \tag{2.26}$$

Furthermore, the reconstructed data from $\boldsymbol{X}_m$ is obtained by

$$\hat{\boldsymbol{x}}_m = (\boldsymbol{\Phi}^T)^{-1} \boldsymbol{X}_m. \tag{2.27}$$

Because $\boldsymbol{\Phi}$ is a unitary matrix, we have

$$\boldsymbol{\Phi}^{-1} = \boldsymbol{\Phi}^T. \tag{2.28}$$

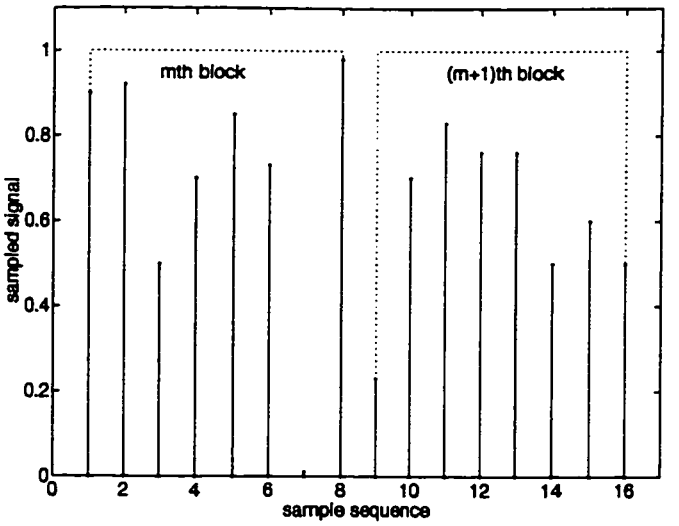


Figure 2.8: Cutoff block system

Thus, (2.26) becomes

$$\hat{\boldsymbol{x}}_m = \boldsymbol{\Phi} \boldsymbol{X}_m. \tag{2.29}$$

Substituting (2.26) and (2.25) into the above formula, we have

$$\hat{\boldsymbol{x}}_m = \boldsymbol{\Phi}\boldsymbol{\Phi}^T \boldsymbol{x}_m = \boldsymbol{x}_m. \tag{2.30}$$

(2.30) indicates that if a block transform is orthonormal, then it satisfies the perfect reconstruction conditions when a signal is processed by using this transform. Here is a complete picture for the processing of a signal using transforms: a signal is first divided into a series of blocks, and each block is transformed individually. The transformed signal in each block is then processed by further techniques, such as quantization, compression, encoding or decoding. In order to recover the original signal, we can apply an inverse transform to realize the reconstruction operation. Now we are going to show how the block transform system is related to filter banks.

Let us go back to (2.26). Rewriting the mth block processing, we have

$$X_m(k) = \sum_{n=0}^{M-1} \tilde{x}_m(n)\phi_k(n), \tag{2.31}$$

where $X_m(k)$ is the k-th element of $\boldsymbol{X}_m$, $\tilde{x}_m(n)$ is the n-th element of $\boldsymbol{x}_m$, and $\phi_k(n)$ is

the n-th element of ϕ_k. (2.31) can be written as follows:

$$\begin{aligned} X_m(k) &= \sum_{n=0}^{M-1} \tilde{x}_m(n)\phi_k[M-1-(M-1-n)] \\ &= \begin{cases} \sum_{n=0}^{M-1} \tilde{x}_m(n)\phi_k[i-(M-1-n)]; & i = M-1; \\ 0 & otherwise. \end{cases} \end{aligned} \tag{2.32}$$

Let $\tilde{X}_m(k,i)$ represent the first line in (2.32): i.e.,

$$\tilde{X}_m(k,i) = \sum_{n=0}^{M-1} \tilde{x}_m(n)\phi_k[i-(M-1-n)]. \tag{2.33}$$

(2.33) can be described as a convolution operation:

$$\tilde{X}_m(k,i) = \tilde{x}_m(n) * \phi_k(M-1-n), \tag{2.34}$$

where $*$ is a convolution operation. Thus, we have

$$X_m(k) = \begin{cases} \tilde{X}_m(k,i) & i = M-1; \\ 0 & otherwise. \end{cases} \tag{2.35}$$

It is easily seen that (2.34) is a filtering operation while (2.35) is a decimating one. The above results indicate that block transform systems are equivalent to a set of filter banks. In the analysis filter banks, the impulse response of the k-th subband is $\phi_k(M-1-n)$, which is obtained by reversing the basis function $\phi_k(n)$. The transformed coefficients are collected after filtering and subsampling by M (as shown in Figure 2.9). Thus, for the sampled sequence of the signal $x(n)$, the subsampled outputs at times $M-1, 2M-1, \cdots$, represent successive transform coefficient vectors corresponding to successive blocks of data. Therefore, it is seen that the filter bank system can be realized by a block transform system. Later, we will also see that this relation is not only limited to the cutoff block transform, but also applicable to the smooth block transform and wavelet transform. Similarly, in the synthesis filter banks, the impulse response is given by the inverse transform. The reconstructed signal can be obtained by the process: upsampling by M, and then filtering. Therefore, we conclude that a block transform can be viewed as a decimation filter bank and the inverse transform as an expanding filter bank. On the other hand, a filter bank can be viewed as a transform with the overlapped operation,

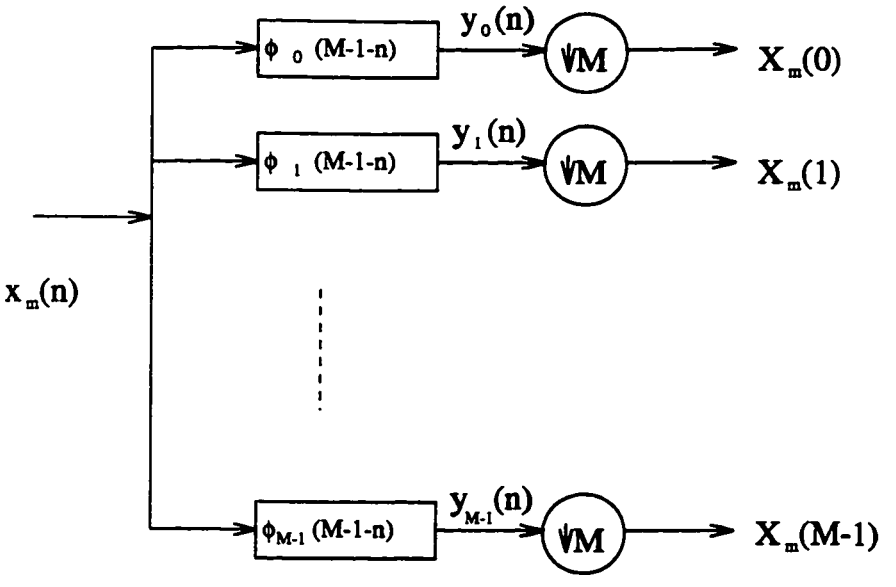


Figure 2.9: Orthonormal transform analyzer as a multirate filter bank

in which the transform of a block contains not only the data in that block but also the data in its neighbouring blocks. This relation allows for easy signal processing in the transform domain by using the multirate filter banks.

Even though block transforms and filter banks have a one-to-one correspondence, each one has its own characteristics. Traditional block transforms (cutoff block transforms) allow for the short length of the basis functions. Corresponding to the concept of filter banks, the impulse response does not go down to zero at the boundaries, because of the "cutoff" block processing. This causes discontinuities between two adjacent blocks in the reconstructed signal and creates so-called blocking effects [51], [53]. In image processing, the blocking effect results in a reconstructed image that seems to be built up of small tiles across block boundaries, whereas in speech coding, it will result in audible periodic tones [72]. Filter banks, as discussed in Section 2.1, show that with long impulse response filters, it is possible to build a filter in which the impulse responses decay smoothly to zero, and hence the blocking effects disappear [71]. In order to design a filter with the impulse response decaying to zero smoothly, we require that the filter be long. In general, the longer the filter length is, the less is the boundary effect. In practice, developing such systems is not easy; it requires the design of a long filter. Fast

algorithms may not be possible for real time processing. The relation between transforms and filter banks enables us to develop a flexible time-frequency transform with a given length for the impulse response filter. That implies that such transform has not only good frequency resolution but also good time resolution. Lapped orthogonal transform is one such transform.

2.3 Lapped Orthogonal Transform

As described in the last section, the traditional orthogonal block transform system has a perfect reconstruction property. That is, given a signal, in order to examine its properties on a finite sub-interval, we can map the signal into a transform domain by using a block transform. The conventional block transforms process data block-by-block. No overlapped operation is involved.

Lapped orthogonal transforms, proposed in [49], are slightly different from standard block transforms. It transforms N sample points of a block into M transformed coefficients, where $N \geq M$. That is, the transform matrix is $M \times N$ rather than $M \times M$. A non-square matrix indicates that the sample sequence of a signal is divided into a set of overlapped blocks (shown in Figure 2.10). Each block has length N and the length of basis functions is also N. However, the total number of basis functions is M. Because of overlapping, the blocking effect in data compression can be reduced. This will be demonstrated in detail in Chapter 6.

Consider a real signal $x(n)$, $n = 0, 1, \cdots, K$. Let $\boldsymbol{x}_m$ be data of the m-th block. The difference of the lapped orthogonal transforms from the traditional block transforms is that two adjacent blocks $\boldsymbol{x}_m$ and $\boldsymbol{x}_{m+1}$ have an overlapped portion. Let $\boldsymbol{x}_m$ be a column vector with dimension N. The transform coefficients of $\boldsymbol{x}_m$ can be written as

$$\boldsymbol{X}_m = \boldsymbol{\Phi}^T \boldsymbol{x}_m, \tag{2.36}$$

where $\boldsymbol{\Phi}$ is a lapped transform matrix with dimension $N \times M$, in which the column vectors correspond to basis functions with length N, $N \geq M$. $\boldsymbol{X}_m$ is a column vector with dimension M. (2.36) shows that lapped transform maps data with length N into a set of coefficients with length M. The relation between lapped transforms and filter

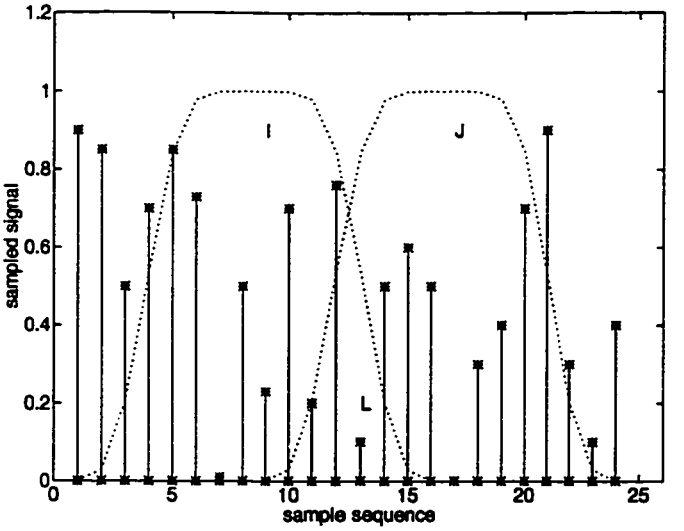


Figure 2.10: Smoothed block system — lapped block system

banks can also be described by Figure 2.9, with $\phi_i(M-1-n)$ being replaced with $\phi_i(N-1-n)$. It is worth noting that because $\boldsymbol{\Phi}$ is not a square matrix, it can no longer be unitary. Next, we will describe how to obtain the perfect reconstruction for lapped transforms.

Now let us look at how lapped transform works for a sequence of data $x(n)$ instead of a single block $\boldsymbol{x}_m$. Consider the overall transform matrix $\tilde{\boldsymbol{\Phi}}_w$, which is block diagonal. Each matrix $\boldsymbol{\Phi}$ corresponds to a block process. The subscript w represents processing the whole data set. We could also use different lengths of basis functions so that the size of one block matrix may be different from another. For simplicity, we consider the case where all blocks are of the same size. Here, $\boldsymbol{\Phi}$'s are positioned so that $\tilde{\boldsymbol{\Phi}}_w$ is a square matrix or nearly a square matrix provided $\{x(n)\}$ is long enough. $\tilde{\boldsymbol{\Phi}}_w$ is

$$\tilde{\boldsymbol{\Phi}}_w = \begin{bmatrix} \boldsymbol{\Phi} & & \boldsymbol{O} \\ & \boldsymbol{\Phi} & \\ & \ddots & \\ \boldsymbol{O} & & \boldsymbol{\Phi} \end{bmatrix}. \tag{2.37}$$

Now we describe how they are positioned. Considering three adjacent $\{\boldsymbol{\Phi}\}$s in (2.37), we denote the block matrix at left top as $\boldsymbol{\Phi}^{(l)}$, the middle as $\boldsymbol{\Phi}^{(m)}$, and the right bottom as

$\boldsymbol{\Phi}^{(r)}$, where the superscript (l), (m) and (r) represent left, middle and right respectively. $\boldsymbol{\Phi}^{(l)}$ is positioned $(N-M)$ rows above the $\boldsymbol{\Phi}^{(m)}$ block to its left, and $\boldsymbol{\Phi}^{(r)}$ is positioned $(N-M)$ rows below the $\boldsymbol{\Phi}^{(m)}$ to its right. To illustrate, we show some of the blocks in $\tilde{\boldsymbol{\Phi}}_w$ when $N=4$ and $M=2$ and the overlapped length is 2; i.e.,

$$\boldsymbol{\Phi}^T = \begin{bmatrix} a_{11} & a_{21} & a_{31} & a_{41} \\ a_{12} & a_{22} & a_{32} & a_{42} \end{bmatrix},$$

and,

$$\tilde{\boldsymbol{\Phi}}_w = \begin{bmatrix} \ddots & & & & \\ & \boldsymbol{\Phi}^{(l)} & & & \\ & & \boldsymbol{\Phi}^{(m)} & & \\ & & & \boldsymbol{\Phi}^{(r)} & \\ & & & & \ddots \end{bmatrix} \Leftrightarrow \begin{bmatrix} \ddots & & & & & & & \\ & a_{11} & a_{12} & & & & & \\ & a_{21} & a_{22} & & & & & \\ & a_{31} & a_{32} & a_{11} & a_{12} & & & \\ & a_{41} & a_{42} & a_{21} & a_{22} & & & \\ & & & a_{31} & a_{32} & a_{11} & a_{12} & \\ & & & a_{41} & a_{42} & a_{21} & a_{22} & \\ & & & & & a_{31} & a_{32} & \\ & & & & & a_{41} & a_{42} & \\ & & & & & & & \ddots \end{bmatrix}.$$

In order to get perfect reconstruction, $\tilde{\boldsymbol{\Phi}}_w$ must satisfy, assuming that its dimension is $K \times K$,

$$\tilde{\boldsymbol{\Phi}}_w \tilde{\boldsymbol{\Phi}}_w^T = \boldsymbol{I}_K \quad \Leftrightarrow \quad \tilde{\boldsymbol{\Phi}}_w^T \tilde{\boldsymbol{\Phi}}_w = \boldsymbol{I}_K. \tag{2.38}$$

(2.38) implies

$$\boldsymbol{\Phi}^T \boldsymbol{\Phi} = \boldsymbol{I}_M, \tag{2.39}$$

and

$$\boldsymbol{\Phi}^T \boldsymbol{W} \boldsymbol{\Phi} = \boldsymbol{O}_M, \tag{2.40}$$

where $\boldsymbol{W}$ is defined as an $M \times M$ one-block shift matrix

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{O}_1 & \boldsymbol{I}_L \\ \boldsymbol{O}_2 & \boldsymbol{O}_1^T \end{bmatrix}. \tag{2.41}$$

Here, $\boldsymbol{I}_L$ is an identity matrix with order L which is the length of the lapped portions. $\boldsymbol{O}_1$ is a $L \times (M-L)$ zero matrix, $\boldsymbol{O}_2$ is a $(M-L) \times (M-L)$ zero matrix. From the above

conditions, the second half in (2.38) can be obtained. The two conditions of (2.39) and (2.40) represent the orthonormality and the lapped orthogonality for a lapped transform.

Thus, for given data $x(n)$, $n = 1, \cdots, K$, with K being an integer multiple of M and $\boldsymbol{x}_w$ denoting a column vector with dimension K, the transformed coefficients of $\boldsymbol{x}_w$ are described as follows:

$$\boldsymbol{X}_w = \tilde{\boldsymbol{\Phi}}_w^T \boldsymbol{x}_w, \tag{2.42}$$

where $\boldsymbol{X}_w$ is a column vector with dimension K. The reconstructed signal is given by

$$\hat{\boldsymbol{x}}_w = \tilde{\boldsymbol{\Phi}}_w \boldsymbol{X}_w = \tilde{\boldsymbol{\Phi}}_w \tilde{\boldsymbol{\Phi}}_w^T \boldsymbol{x}_w. \tag{2.43}$$

Substituting (2.38) to (2.43), we then have

$$\hat{\boldsymbol{x}}_w = \boldsymbol{x}_w. \tag{2.44}$$

(2.44) shows that if a lapped transform has orthogonality (i.e., basis functions are orthogonal to each other in a block) and lapped orthogonality (i.e., the inner products of overlapped portions of any two basis functions in two adjacent blocks are equal to zero), then such a transform has the perfect reconstruction property. Usually, this kind of transform is called "lapped orthogonal transform". Using lapped orthogonal transforms, the blocking effect in data compression can be reduced. Lapped orthogonal transform, as an overlapped block transform system, can have different types of basis functions. LOT, proposed by [49], is developed based on the discrete cosine transform. The length of basis functions N is fixed at $2M$, where M is the number of basis functions. The length of lapped portions L is M. Thus, the length of the lapped portions is half the length of basis functions. Furthermore, a modulated lapped transform (MLT) [49] is proposed which is similar to LOT but with a modulated function at the edges. Also, it satisfies the requirements $N = 2M$, and $L = M$. Extended LOT [49] is an extended form of LOT for a longer length of basis functions: i.e., $N = 2PM$, where P is an integer. The length of lapped portions is $(2P - 1)M$.

2.4 DLS Transform and DLC Transform

As described before, the lapped orthogonal transforms, which merge filter banks into conventional block transform and let the boundaries decay to zero graduately, can reduce

the blocking effect. LOT is one such lapped orthogonal transform, and it has a limitation for the length of the overlapped portions. Here, we introduce the discrete local sine transform and the discrete local cosine transform, which allow for flexible length in the lapped regions and arbitrarily smooth cutoff at the edges.

Considering a signal $x(n)$, in order to analyse its properties on a finite sub-interval I, we could process $x(n)$ by using a block transform. To find a smooth cutoff, we borrow a set of continuous bases from Coifman and Meyer [17], [6], the local sine bases and the local cosine bases. Such bases allow for an arbitrarily smooth cutoff. By discretizing these bases, we derive a set of discrete local sine (DLS) basis functions and a set of discrete local cosine (DLC) basis functions. These are defined as the DLS and DLC transforms respectively, given by

- the DLS transform:

  Let $\boldsymbol{\Phi}_s$ be the DLS transform matrix, $\boldsymbol{\Phi}_s = [\boldsymbol{\phi}_0, \boldsymbol{\phi}_1, \cdots, \boldsymbol{\phi}_{M-1}]$. The basis function $\boldsymbol{\phi}_r(n)$ is

$$\phi_r(n) = \sqrt{\frac{2}{M}}\{b_I(n)\sin[\frac{2r+1}{2}\pi(\frac{n}{M}-\varepsilon)]\}, n \in [0, M+L-1]; r \in [0, M-1], \tag{2.45}$$

  where n represents the index of sample points, r represents the index of basis functions, $\varepsilon = \frac{L-1}{2M}$. M is the number of basis functions and L is the length of the lapped portion of the basis functions. $b_I(n)$ is called a bell function defined by

$$b(n) = \begin{cases} S_e(n), & n = 0, 1, \cdots, L-1, \\ 1 & n = L, \cdots, M-1, \\ C_e(n-M), & n = M, \cdots, M+L-1, \end{cases} \tag{2.46}$$

  where

$$S_e(n) = \sin[\frac{n\pi}{2(L-1)} - \frac{1}{4}\sin\frac{2n\pi}{L-1}], \quad n = 0, 1, \cdots, L-1, \tag{2.47}$$

$$C_e(n) = \cos[\frac{n\pi}{2(L-1)} - \frac{1}{4}\sin\frac{2n\pi}{L-1}], \quad n = 0, 1, \cdots, L-1. \tag{2.48}$$

  Figure 2.11 shows the bell function.

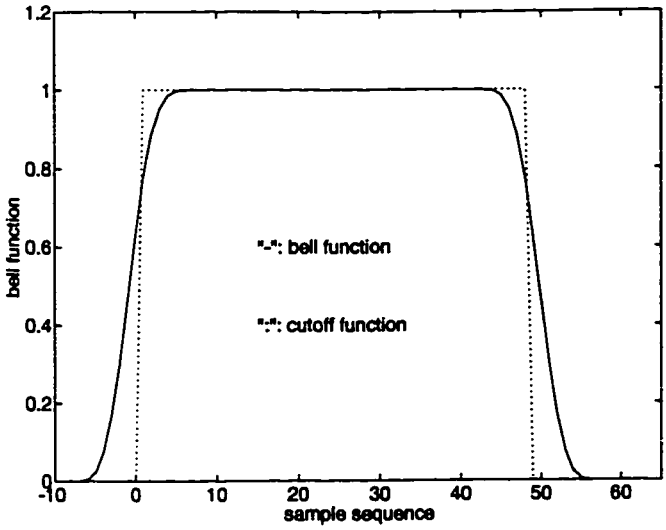


Figure 2.11: Bell function

- the DLC transform:

  Let Φ_c be the DLC transform matrix. $\Phi_c = [\phi_0, \phi_1, \cdots, \phi_{M-1}]$. The basis function ϕ_r is

$$\phi_r(n) = \sqrt{\frac{2}{M}}\{b_I(n)\cos[\frac{2r+1}{2}\pi(\frac{n}{M} - \varepsilon)]\}, n \in [0, M+L-1]; r \in [0, M-1] \tag{2.49}$$

  where $b_I(n)$ is defined as before.

In the next chapter, we will show that DLS and DLC transforms satisfy the orthogonality and lapped orthogonality properties, similar to (2.39) and (2.40)

$$\begin{cases} \Phi_s^T \Phi_s = I_M \\ \Phi_s^T W \Phi_s = O_M \end{cases}, \tag{2.50}$$

$$\begin{cases} \Phi_c^T \Phi_c = I_M \\ \Phi_c^T W \Phi_c = O_M \end{cases}. \tag{2.51}$$

The bases of DLS and DLC in time and frequency domains are shown in Figure 2.12 and 2.13, respectively where we have chosen $M = 8$ and $L = 8$.

For comparison, we note that the LOT is defined as

$$\mathbf{\Phi}_l = \frac{1}{2}\begin{bmatrix} D_e - D_o & D_e - D_o \\ J(D_e - D_o) & -J(D_e - D_o) \end{bmatrix}, \tag{2.52}$$

with D_e and D_o being $M \times \frac{M}{2}$ matrices containing the even and odd DCT functions of length M, respectively. $\mathbf{\Phi}_l$ also satisfies the lapped orthogonal properties [49]. The bases in time and frequency domains are presented in Figure 2.14. Figure 2.12 indicates that DLS transform, based on a bell function, belongs to smooth cutoff system, while LOT still has some boundary effects as shown in Figure 2.14 by examining the time domain basis functions. Figure 2.15 shows MLT defined in Chapter 5 of [49]. The results indicates that MLT has better frequency characteristics than DLS transform.

In this chapter, we discuss fundamentals of filter banks and their relations with block transforms. Then we go further into lapped orthogonal transforms. The relation between lapped transforms and filter banks is described in Figure 2.9, with $\phi_i(N-1-n)$ being replaced by $\phi_i(M-1-n)$. Computing a transform is equivalent to convolving the signal with each of the block basis functions and then subsampling by a factor equal to the number of basis functions. Also, lapped orthogonal transforms could reduce the blocking effect which appears in the application of data compression. Comparing with LOT, it is seen that the DLS and DLC transforms, as a smooth cutoff system, may provide a more efficient approach to reduce the blocking effect. Next, we will introduce the operational properties of lapped transforms based on DLS and DLC basis functions.

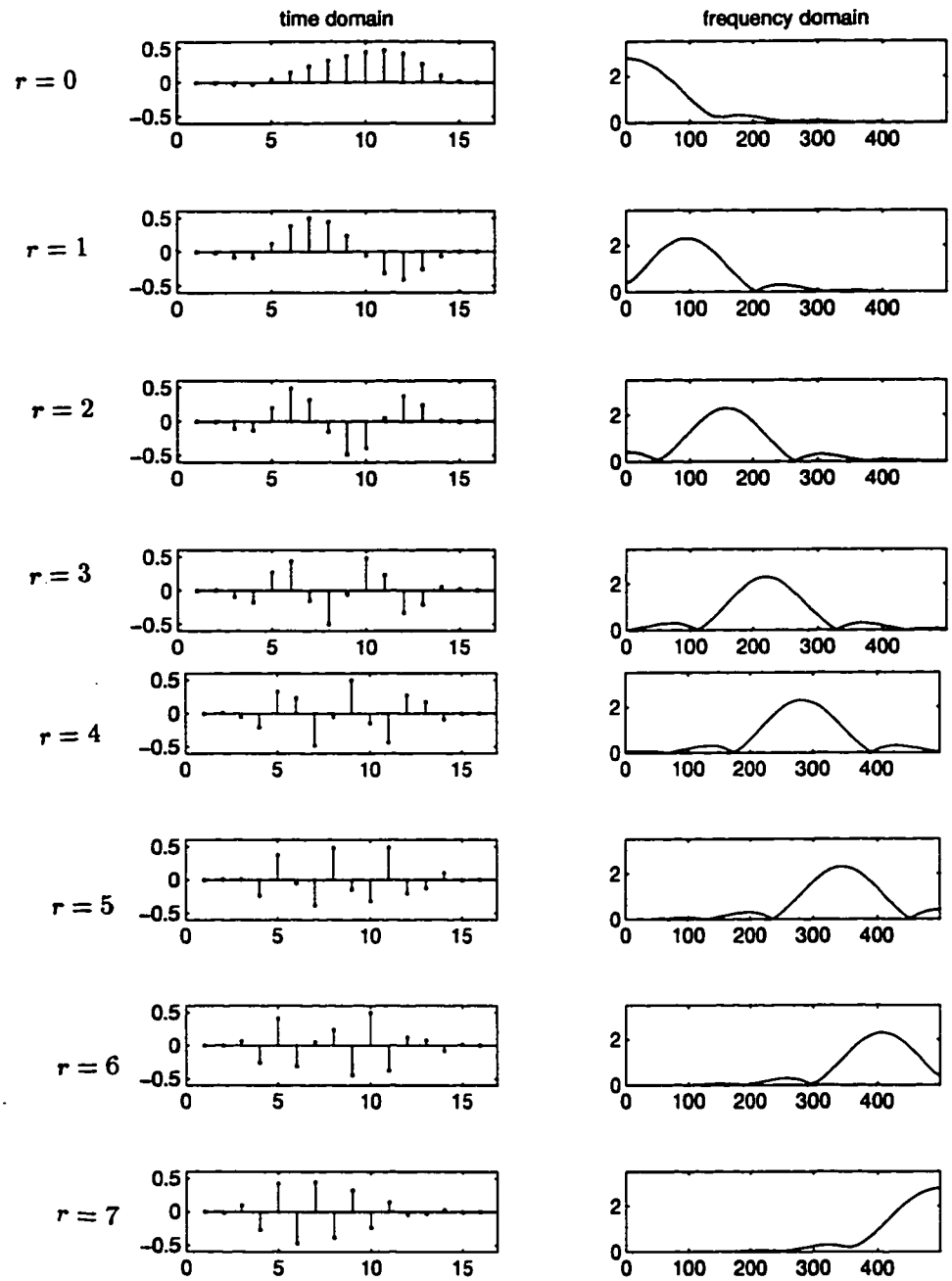


Figure 2.12: Bases in time and frequency domains ($M = 8$ and $L = 8$; 8-basis DLS)

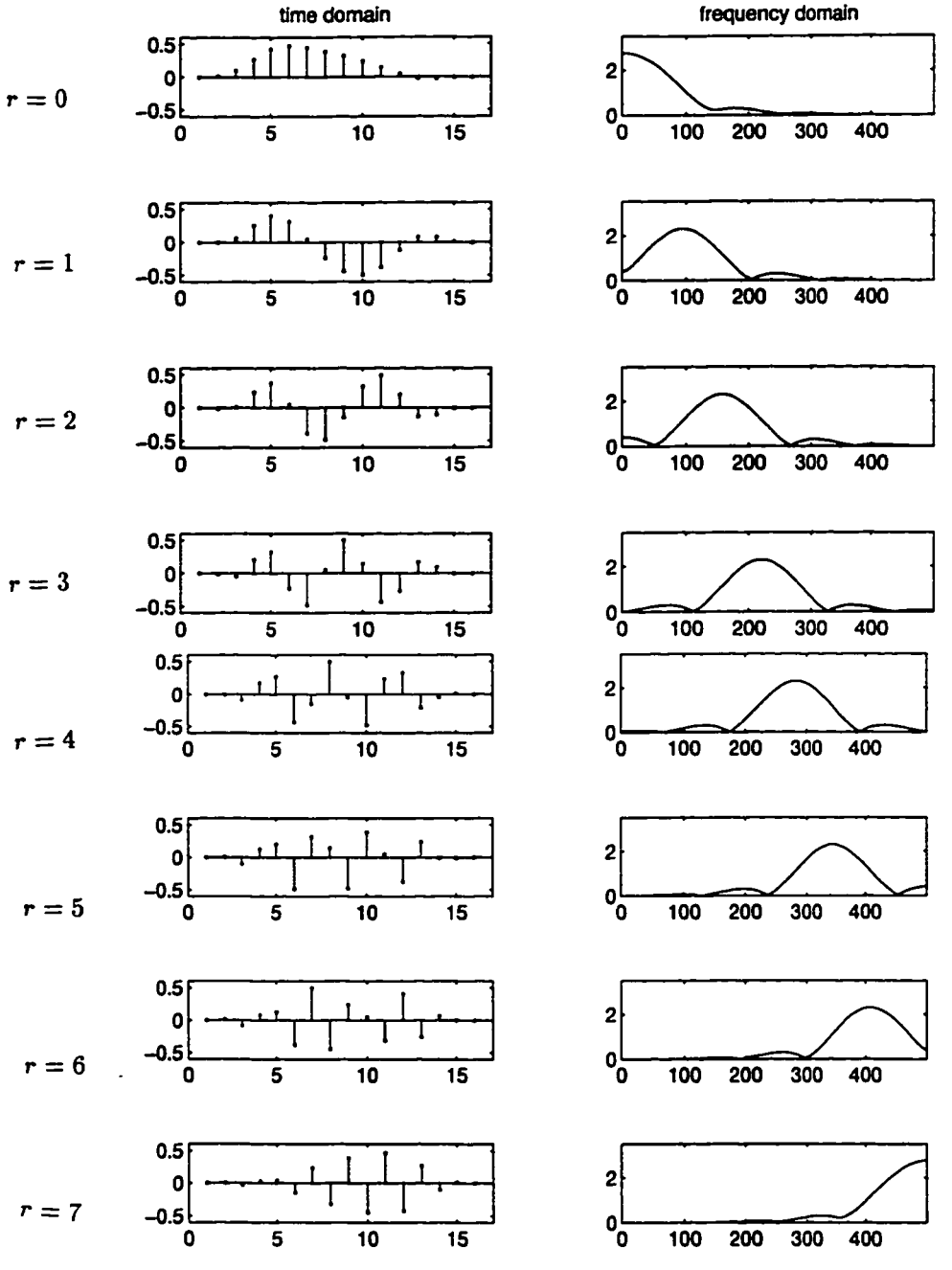


Figure 2.13: Bases in time and frequency domains ($M = 8$ and $L = 8$; 8-basis DLC)

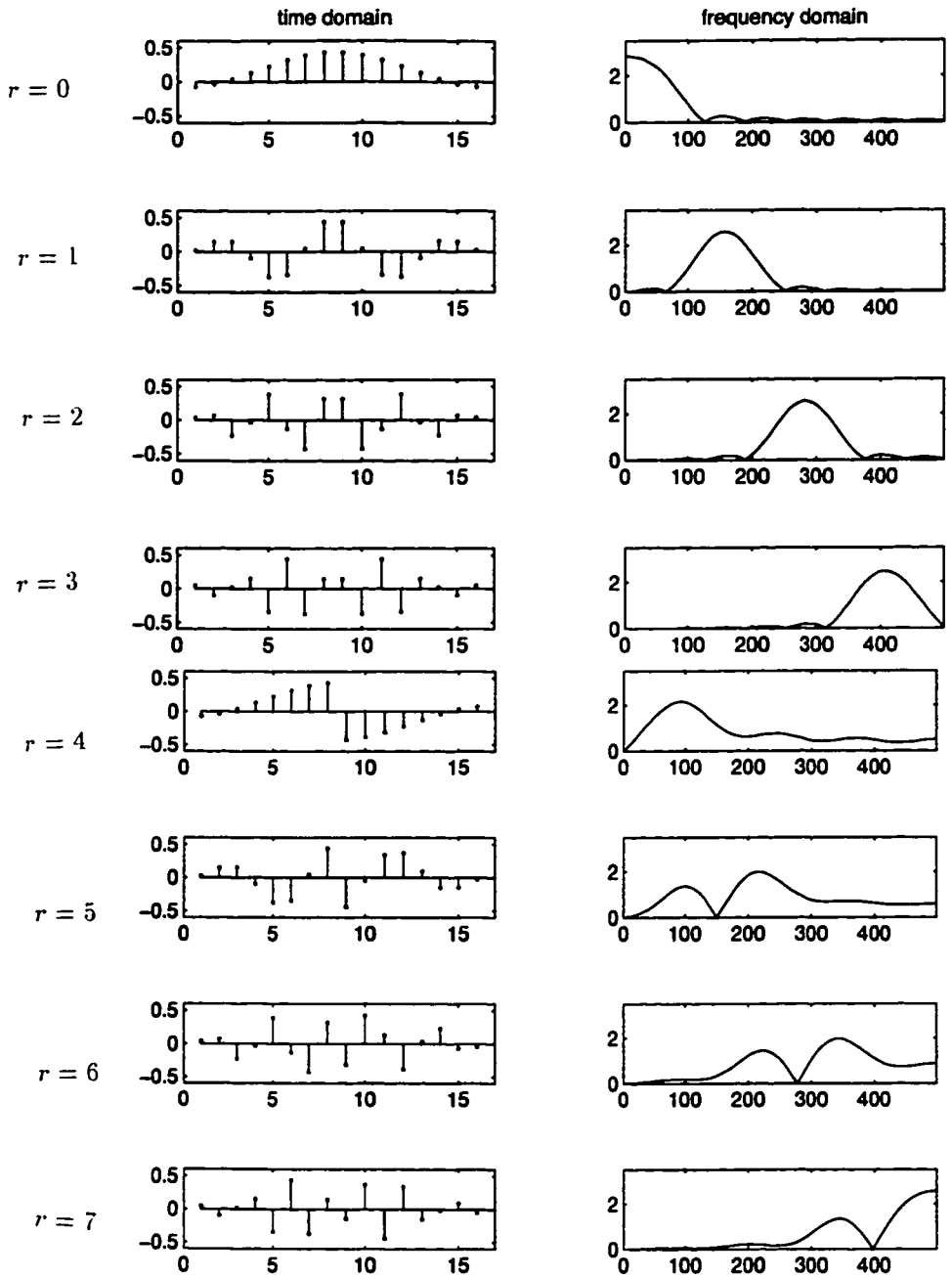


Figure 2.14: Bases in time and frequency domains ($M = 8$ and $L = 8$; 8-basis LOT)

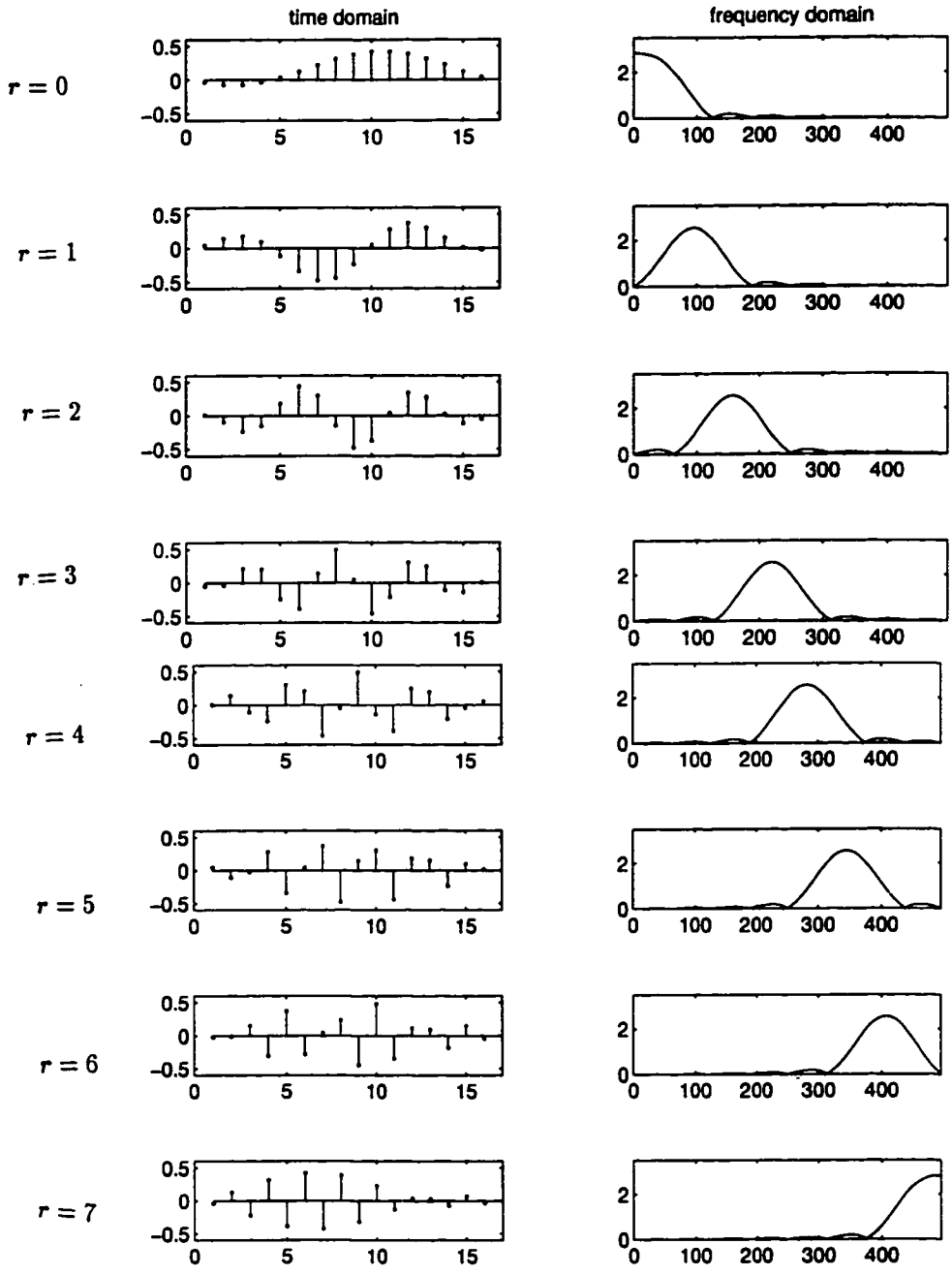


Figure 2.15: Bases in time and frequency domains ($M = 8$ and $L = 8$; 8-basis MLT)

Chapter 3

DLS and DLC Transforms: Properties and Performance Analysis

3.1 Introduction

In the last chapter, we reviewed the lapped orthogonal transforms and their relations with multirate filter banks. Because of the smooth cutoff, DLS and DLC transforms have the potential to significantly reduce the blocking effect. In this chapter, we derive important operational properties and performance limitations for the discrete local sine/cosine (DLS/DLC) transforms. In Sections 3.2, 3.3 and 3.4, we start with the DLS transform to analyse the unitarity property and to derive the orthogonality and lapped orthogonality properties. Following Section 3.4, we use a general form of lapped orthogonal transforms to discuss the perfect reconstruction property, scaling-in-time, shift-in-time, uniqueness properties and the convolution operation. According to the definitions of the DLS and DLC transforms in the last chapter, we rewrite equation (2.45) and (2.49) respectively as follows,

$$\text{DLS: } \phi_r(n) = \sqrt{\tfrac{2}{M}}\{b(n)\sin\tfrac{(2r+1)(2n-L+1)\pi}{4M}\} \quad n \in [0, M+L-1], r \in [0, M-1], \tag{3.1}$$

$$\text{DLC: } \phi_r(n) = \sqrt{\tfrac{2}{M}}\{b(n)\cos\tfrac{(2r+1)(2n-L+1)\pi}{4M}\} \quad n \in [0, M+L-1], r \in [0, M-1], \tag{3.2}$$

where M is the number of basis functions, L is the length of lapped portions ($L \leq M$) and r is an integer. $b(n)$ is a bell function defined in (2.46). Both the DLS and DLC transforms may use the same bell function. The only difference between them is that DLS uses the sine function and DLC uses the cosine function. Thus, we could use a common notation $\phi_r(n)$ to examine their matrix forms. Let the matrix form of $\phi_r(n)$ be $\boldsymbol{\Phi}$ with dimension $(M+L) \times M$. According to the characteristics of the DLS and DLC basis functions, we could divide the matrix $\boldsymbol{\Phi}$ into three parts:

$$\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\Phi}^{(t)} \\ \boldsymbol{\Phi}^{(m)} \\ \boldsymbol{\Phi}^{(h)} \end{bmatrix}, \tag{3.3}$$

where the superscript (h), (m) and (t) represent the head, middle, and tail portions, as shown in Figure 3.1, and

$$\begin{aligned} \boldsymbol{\Phi}^{(h)} &= [\boldsymbol{\phi}_0^{(h)}, \boldsymbol{\phi}_1^{(h)}, \cdots, \boldsymbol{\phi}_{M-1}^{(h)}], \quad \text{of dimension } L \times M, \\ \boldsymbol{\Phi}^{(t)} &= [\boldsymbol{\phi}_0^{(t)}, \boldsymbol{\phi}_1^{(t)}, \cdots, \boldsymbol{\phi}_{M-1}^{(t)}], \quad \text{of dimension } L \times M, \\ \boldsymbol{\Phi}^{(m)} &= [\boldsymbol{\phi}_0^{(m)}, \boldsymbol{\phi}_1^{(m)}, \cdots, \boldsymbol{\phi}_{M-1}^{(m)}], \quad \text{of dimension } (M-L) \times M, \end{aligned} \tag{3.4}$$

$$\begin{aligned} \boldsymbol{\phi}_r^{(h)} &= [\phi_r(M), \phi_r(M+1), \cdots, \phi_r(M+L-1)]^T, \\ \boldsymbol{\phi}_r^{(t)} &= [\phi_r(0), \phi_r(1), \cdots, \phi_r(L-1)]^T, \\ \boldsymbol{\phi}_r^{(m)} &= [\phi_r(L), \phi_r(L+1), \cdots, \phi_r(M-1)]^T. \end{aligned} \tag{3.5}$$

Starting with these definitions, we will derive some important operational properties in the following sections. First, we focus on the DLS basis functions; similar results can be obtained for the DLC basis functions.

3.2 Orthonormality

Consider the two basis functions $\boldsymbol{\phi}_r$ and $\boldsymbol{\phi}_t$. Their inner product is given by

$$\begin{aligned} < \boldsymbol{\phi}_r, \boldsymbol{\phi}_t > &= \sum_{n=0}^{M+L-1} \frac{2}{M} b^2(n) \sin \frac{(2r+1)(2n-L+1)\pi}{4M} \sin \frac{(2t+1)(2n-L+1)\pi}{4M} \\ &= \frac{1}{M} \sum_{n=0}^{M+L-1} b^2(n) \left[\cos \frac{(r-t)(2n-L+1)\pi}{2M} - \cos \frac{(r+t+1)(2n-L+1)\pi}{2M} \right] \end{aligned}$$

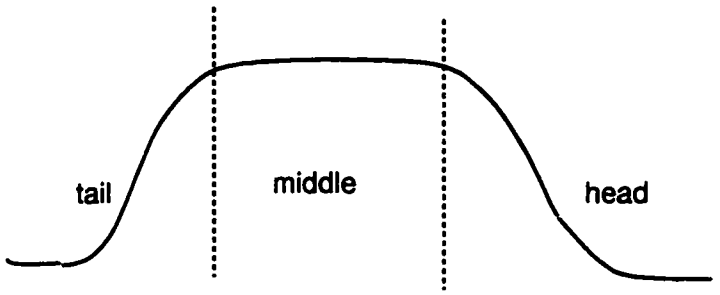


Figure 3.1: Expression of portions in one block

$$= \frac{1}{M}(T_1 + T_2 + T_3), \tag{3.6}$$

where

$$T_1 = \sum_{n=0}^{L-1} S_\epsilon^2(n) \left[\cos\frac{(r-t)(2n-L+1)\pi}{2M} - \cos\frac{(r+t+1)(2n-L+1)\pi}{2M}\right], \tag{3.7}$$

$$T_2 = \sum_{n=L}^{M-1} \left[\cos\frac{(r-t)(2n-L+1)\pi}{2M} - \cos\frac{(r+t+1)(2n-L+1)\pi}{2M}\right], \tag{3.8}$$

$$T_3 = \sum_{n=M}^{M+L-1} C_\epsilon^2(n-M) \left[\cos\frac{(r-t)(2n-L+1)\pi}{2M} - \cos\frac{(r+t+1)(2n-L+1)\pi}{2M}\right]. \tag{3.9}$$

Here, we have written out the bell functions for the appropriate ranges of the summation. It is easy to see that T_3 can be rewritten by substituting n with $n' = n - M$

$$\begin{aligned} T_3 &= \sum_{n'=0}^{L-1} C_\epsilon^2(n') \left[\cos\frac{(r-t)(2n'+2M-L+1)\pi}{2M} - \cos\frac{(r+t+1)(2n'+2M-L+1)\pi}{2M}\right] \\ &= \sum_{n'=0}^{L-1} C_\epsilon^2(n') \left\{\cos[(r-t)\pi + \frac{(r-t)(2n'-L+1)\pi}{2M}] \right. \\ &\quad \left. - \cos[(r+t+1)\pi + \frac{(r+t+1)(2n'-L+1)\pi}{2M}\right\}. \end{aligned} \tag{3.10}$$

Using this expression for T_3, we consider $< \phi_r, \phi_t >$ in the cases where r and t differ by an even integer or by an odd integer.

- CASE I: for $r-t=2k, k=0,1,\cdots,p$ and $p=M/2$ if M is even; $p=(M-1)/2$ if M is odd. In this case, T_3 reduces to

$$T_3=\sum_{n=0}^{L-1}C_\epsilon^2(n)\left[\cos\frac{k(2n-L+1)\pi}{M}+\cos\frac{(2k+2t+1)(2n-L+1)\pi}{2M}\right]. \tag{3.11}$$

Substituting (3.11) into (3.6) and using $S_\epsilon^2(n)+C_\epsilon^2(n)=1$, we have

$$\begin{aligned}<\phi_r,\phi_t> &= \frac{1}{M}\left\{\sum_{n=0}^{M-1}\cos\frac{k(2n-L+1)\pi}{M}\right.\\ &\quad -\sum_{n=0}^{L-1}[S_\epsilon^2(n)-C_\epsilon^2(n)]\cos\frac{(2k+2t+1)(2n-L+1)\pi}{2M}\\ &\quad \left.-\sum_{n=L}^{M-1}\cos\frac{(2k+2t+1)(2n-L+1)\pi}{2M}\right\}\\ &= \frac{1}{M}(P_I-P_{II}-P_{III}),\end{aligned} \tag{3.12}$$

where

$$P_I = \sum_{n=0}^{M-1}\cos\frac{k(2n-L+1)\pi}{M}, \tag{3.13}$$

$$P_{II} = \sum_{n=0}^{L-1}[S_\epsilon^2(n)-C_\epsilon^2(n)]\cos\frac{(2k+2t+1)(2n-L+1)\pi}{2M}, \tag{3.14}$$

$$P_{III} = \sum_{n=L}^{M-1}\cos\frac{(2k+2t+1)(2n-L+1)\pi}{2M}. \tag{3.15}$$

Through trigonometric property, P_I, P_{II} and P_{III} have the following results (refer to (A.4), (A.6) and (A.8) of Appendix A):

$$P_I = \begin{cases}0, & k\neq 0\\ M, & k=0\end{cases} \tag{3.16}$$

$$P_{II} = 0, \quad \text{for all } k, \tag{3.17}$$

$$P_{III} = 0, \quad \text{for all } k. \tag{3.18}$$

Combining (3.16), (3.17) and (3.18), we have for $r-t=2k$

$$<\phi_r,\phi_t>=\begin{cases}1 & r-t=2k=0;\\ 0 & r-t=2k\neq 0.\end{cases} \tag{3.19}$$

- CASE II: for $r-t=2k+1, k=0,1,\cdots,p$ and $p=M/2-1$ if M is even; $p=(M-3)/2$ if M is odd ($M\geq 3$). In this case, T_3 is equal to

$$T_3=\sum_{n=0}^{L-1}C_\epsilon^2(n)\left\{-\cos\frac{(2k+1)(2n-L+1)\pi}{2M}-\cos\frac{(k+t+1)(2n-L+1)\pi}{M}\right\}. \tag{3.20}$$

Substituting (3.20) into (3.6), we have

$$\begin{aligned} < \phi_r, \phi_t > \; = \; & \frac{1}{M}\left\{ -\sum_{n=0}^{M-1} \cos\frac{(k+t+1)(2n-L+1)\pi}{M} \right. \\ & + \sum_{n=0}^{L-1} [S_\epsilon^2(n) - C_\epsilon^2(n)] \cos\frac{(2k+1)(2n-L+1)\pi}{2M} \\ & \left. + \sum_{n=L}^{M-1} \cos\frac{(2k+1)(2n-L+1)\pi}{2M} \right\}. \end{aligned} \tag{3.21}$$

Similar to CASE I, for $r - t = 2k + 1$, we have

$$< \phi_r, \phi_t >= 0, \quad k = 0, 1, \cdots, p. \tag{3.22}$$

Therefore, combining (3.19) and (3.22), we have the following orthonormality property

$$< \phi_r, \phi_t >= \begin{cases} 1 & r = t; \\ 0 & r \neq t. \end{cases} \tag{3.23}$$

(3.23) indicates that the discrete local sine transform satisfies the orthonormality property. In the same way, we can develop the orthonormal property for the discrete local cosine transform. Using the orthogonal property, we obtain, for the transform matrix in (3.3)

$$\boldsymbol{\Phi}^T \boldsymbol{\Phi} = \boldsymbol{\Phi}^\dagger \boldsymbol{\Phi} = I_M,$$

where † represents the pseudo-inverse of a matrix, and the column vectors of $\boldsymbol{\Phi}$ correspond to the basis functions. I_M is an $M \times M$ identity matrix. Since the columns of $\boldsymbol{\Phi}$ are orthonormal to one another, $\boldsymbol{\Phi}$ is an orthogonal matrix: i.e., $\boldsymbol{\Phi}^T = \boldsymbol{\Phi}^\dagger$.

3.3 Lapped Orthogonality

Consider two basis functions $\phi_r^{(0)}$ and $\phi_t^{(1)}$, where the superscripts (0) and (1) represent two adjacent blocks (0) and (1), as shown in Figure 3.2. Their inner product over the lapped region is described as

$$\begin{aligned} < \phi_r^{(0)}, \phi_t^{(1)} > \; = \; & \frac{2}{M} \sum_{n=M}^{M+L-1} \{C_\epsilon(n-M) S_\epsilon(n-M) \\ & \sin\frac{(2r+1)(2n-L+1)\pi}{4M} \sin\frac{(2t+1)(2n-L-2M+1)\pi}{4M} \Big\}. \end{aligned} \tag{3.24}$$

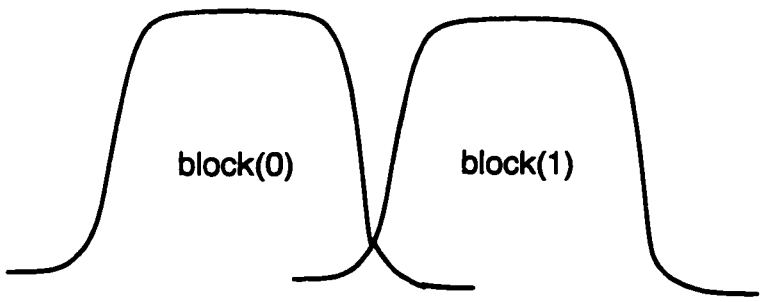


Figure 3.2: Two adjacent blocks (0) and (1)

Letting $n' = n - M$, substituting (2.47) and (2.48) into the above formula, and using trigonometric identities, we have

$$< \phi_r^{(0)}, \phi_t^{(1)} > = \frac{1}{M} \sum_{n'=0}^{L-1} \left\{ \sin\left(\frac{n'\pi}{L-1} - \frac{1}{2}\sin\frac{2n'\pi}{L-1}\right) \cos\frac{(2r+1)(2n'-L+1)\pi}{4M} \sin\frac{(2t+1)(2n'-L+1)\pi}{4M} \right\}. \tag{3.25}$$

Now, let us examine each factor inside the summation. It is easily seen that over the range $[0, L-1]$, the second factor is symmetric with respect to $n' = \frac{L-1}{2}$, while the third factor is anti-symmetric. For the first factor, let

$$f(n) = \sin\left(\frac{n\pi}{L-1} - \frac{1}{2}\sin\frac{2n\pi}{L-1}\right).$$

It can be easily proven that

$$f(L-1-n) = \sin\left(\frac{n\pi}{L-1} - \frac{1}{2}\sin\frac{2n\pi}{L-1}\right) = f(n) \quad \text{for all } n \in [0, L-1]. \tag{3.26}$$

Hence the first factor inside the summation of (3.25) is symmetric. Therefore, (3.25) is a summation over anti-symmetric terms with respect to $n' = \frac{L-1}{2}$. Thus

$$< \phi_r^{(0)}, \phi_t^{(1)} >= 0 \qquad \text{for all } r \text{ and } t. \tag{3.27}$$

(3.27) shows that the DLS basis functions in block (0) and (1) have lapped orthogonality property. This lapped orthogonality property can easily be extended to any two adjacent blocks (i) and (i+1). In such cases, n in (3.24) is shifted by iM. By replacing the variable n with $n'+iM$, exactly the same result as (3.25) can be obtained. Therefore, we conclude that the DLS transform also satisfies the lapped orthogonality property. Concerning the DLC transform, it is known that the only difference between the DLC and DLS is to change the sine function into the cosine function in (3.24). The bell function which is formed by $C_e(n-M)$ and $S_e(n-M)$ remains unchanged. Therefore, their inner products over the lapped region have the same form as (3.25) except with a different sign. In a similar way, we can derive lapped orthogonality for the DLC transform. Because of the lapped orthogonality, the cross terms between the adjacent blocks will not introduce an error when we reconstruct the original data from their transforms.

3.4 Unitarity

It is easy to see that the basis functions of lapped transforms do not form a square matrix. Therefore, the lapped transform matrix cannot be unitary. However, portions of the matrix may contain a property related to unitarity. This property may be useful in actual transforms; we examine this property here. Let $\boldsymbol{\Phi}$ be the DLS matrix, in which each column corresponds to a basis function. Now let $\boldsymbol{\psi}_n$ represent the n-th row vector in $\boldsymbol{\Phi}$ such that its r-th component is expressed as

$$\psi_n(r) = \sqrt{\frac{2}{M}} b(n) \sin \frac{(2r+1)(2n-L+1)\pi}{4M}. \tag{3.28}$$

The inner product of the two row vectors $\boldsymbol{\psi}_n$ and $\boldsymbol{\psi}_m$ is given by,

$$\begin{aligned}
<\boldsymbol{\psi}_n, \boldsymbol{\psi}_m> &= \frac{2}{M} \sum_{r=0}^{M-1} b(n)b(m) \sin \frac{(2r+1)(2n-L+1)\pi}{4M} \sin \frac{(2r+1)(2m-L+1)\pi}{4M} \\
&= \frac{1}{M} b(n)b(m) \sum_{r=0}^{M-1} \left[\cos \frac{(2r+1)(n-m)\pi}{2M} - \cos \frac{(2r+1)(n+m-L+1)\pi}{2M} \right] \\
&= \frac{1}{M} b(n)b(m) \left\{ Re \sum_{r=0}^{M-1} \left[W_{2M}^{-\frac{n-m}{2}} W_{2M}^{-(n-m)r} \right] \right. \\
&\quad \left. -Re \sum_{r=0}^{M-1} \left[W_{2M}^{-\frac{n+m-L+1}{2}} W_{2M}^{-(n+m-L+1)r} \right] \right\},
\end{aligned} \tag{3.29}$$

where $W_{2M} = e^{-j\frac{\pi}{M}}$. The value of this inner product depends on the relative values of m and n. Details are contained in Appendix B.

Using the results in Appendix B, we can obtain the matrix form of the inner product, $\boldsymbol{\Phi}\boldsymbol{\Phi}^T$. The results in Appendix B indicate that $\boldsymbol{\Phi}\boldsymbol{\Phi}^T$ is a block diagonal matrix, consisting of three blocks. The first block $\boldsymbol{R}_1$ and the last block $\boldsymbol{R}_2$ are respectively called the tail block and the head block, which correspond to the regions $[0, L-1]$ and $[M, M+L-1]$ in the bell function. The second block is an $(M-L) \times (M-L)$ identity matrix. Thus

$$\boldsymbol{\Phi}\ \boldsymbol{\Phi}^T = \begin{bmatrix} \boldsymbol{R}_1 & 0 & 0 \\ 0 & \boldsymbol{I} & 0 \\ 0 & 0 & \boldsymbol{R}_2 \end{bmatrix}. \tag{3.30}$$

Elements $r_{nm}^{(1)}$ and $r_{nm}^{(2)}$ of $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ are defined respectively by

$$r_{nm}^{(1)} = \begin{cases} -\frac{1}{2}\sin\left(\frac{n\pi}{L-1} - \frac{1}{2}\sin\frac{2n\pi}{L-1}\right), & \text{for } m+n = L-1 \text{ and } n \neq m, \\ \frac{1}{2}\left[1 - \cos\left(\frac{n\pi}{L-1} - \frac{1}{2}\sin\frac{2n\pi}{L-1}\right)\right], & \text{for } m+n \neq L-1 \text{ and } n = m, \\ 0, & \text{otherwise.} \end{cases} \tag{3.31}$$

Because $n \in [0, L-1]$, by shifting the time index n by M in (B.7) and (B.8), we have

$$r_{nm}^{(2)} = \begin{cases} -\frac{1}{2}\sin\left(\frac{n\pi}{L-1} - \frac{1}{2}\sin\frac{2n\pi}{L-1}\right), & \text{for } m+n = L-1 \text{ and } n \neq m, \\ \frac{1}{2}\left[1 + \cos\left(\frac{n\pi}{L-1} - \frac{1}{2}\sin\frac{2n\pi}{L-1}\right)\right], & \text{for } m+n \neq L-1 \text{ and } n = m, \\ 0, & \text{otherwise.} \end{cases} \tag{3.32}$$

It is obvious that both $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ are $L \times L$ symmetric matrices, with non-zero diagonal and cross-diagonal elements. (3.30) indicates that when $M \gg L$, $\boldsymbol{\Phi}$ is close to being a unitary matrix. Even though there is no exact unitarity property for the DLS transform, we can still reconstruct the original signal from its transform; perfect reconstruction condition is still possible. In the same way, we may examine the unitarity for DLC transforms. Next, we will discuss the perfect reconstruction property for both the DLS and DLC transforms. Before discussing this issue, let us define a new matrix $\tilde{\boldsymbol{\Phi}}$ of dimension $(M+L) \times 3M$ based on $\boldsymbol{\Phi}$, and the head and tail blocks $\boldsymbol{\Phi}^{(h)}$ and $\boldsymbol{\Phi}^{(t)}$, defined in (3.3). With this $\tilde{\boldsymbol{\Phi}}$, we have the following theorem.

Theorem 3.1 *Let*

$$\tilde{\boldsymbol{\Phi}} = \begin{bmatrix} \tilde{\boldsymbol{\Phi}}^{(h)} & \boldsymbol{\Phi} & \tilde{\boldsymbol{\Phi}}^{(t)} \end{bmatrix}, \tag{3.33}$$

where

$$\begin{aligned}
\tilde{\boldsymbol{\Phi}}^{(h)} &= \begin{bmatrix} \boldsymbol{\Phi}^{(h)} \\ \cdots \\ \boldsymbol{O}_M \end{bmatrix}, \\
\boldsymbol{\Phi}^{(h)} &= [\boldsymbol{\phi}_0^{(h)}, \boldsymbol{\phi}_1^{(h)}, \cdots, \boldsymbol{\phi}_{M-1}^{(h)}], \\
\tilde{\boldsymbol{\Phi}}^{(t)} &= \begin{bmatrix} \boldsymbol{O}_M \\ \cdots \\ \boldsymbol{\Phi}^{(t)} \end{bmatrix}, \\
\boldsymbol{\Phi}^{(t)} &= [\boldsymbol{\phi}_0^{(t)}, \boldsymbol{\phi}_1^{(t)}, \cdots, \boldsymbol{\phi}_{M-1}^{(t)}], \\
\boldsymbol{\Phi} &= [\boldsymbol{\phi}_0, \boldsymbol{\phi}_1, \cdots, \boldsymbol{\phi}_{M-1}].
\end{aligned} \tag{3.34}$$

$\boldsymbol{O}_M$ is a zero matrix with dimension $M \times M$. $\boldsymbol{\phi}_r^{(h)}$, $\boldsymbol{\phi}_r^{(t)}$ and $\boldsymbol{\phi}_r$ are column vectors defined at the beginning of this chapter, and are rewritten here as

$$\begin{aligned}
\boldsymbol{\phi}_r^{(h)} &= [\phi_r(M), \phi_r(M+1), \cdots, \phi_r(M+L-1)]^T, \\
\boldsymbol{\phi}_r^{(t)} &= [\phi_r(0), \phi_r(1), \cdots, \phi_r(L-1)]^T, \\
\boldsymbol{\phi}_r &= [\phi_r(0), \phi_r(1), \cdots, \phi_r(M+L-1)]^T.
\end{aligned} \tag{3.35}$$

With the above definitions, we have

$$\tilde{\boldsymbol{\Phi}}\, \tilde{\boldsymbol{\Phi}}^T = \boldsymbol{I}_{M+L}, \tag{3.36}$$

where $\boldsymbol{I}_{M+L}$ is an $(M+L) \times (M+L)$ identity matrix.

Proof: According to (3.3), we have

$$\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\Phi}^{(t)} \\ \boldsymbol{\Phi}^{(m)} \\ \boldsymbol{\Phi}^{(h)} \end{bmatrix}.$$

Similarly, $\tilde{\boldsymbol{\Phi}}$ can also be divided into three blocks; i.e.,

$$\tilde{\boldsymbol{\Phi}} = \begin{bmatrix} \boldsymbol{\Delta}_1 \\ \boldsymbol{\Delta}_2 \\ \boldsymbol{\Delta}_3 \end{bmatrix},$$

where $\boldsymbol{\Delta}_1$ and $\boldsymbol{\Delta}_3$ are $L \times 3M$ matrices defined by

$$\boldsymbol{\Delta}_1 = \left[\begin{array}{ccc} \boldsymbol{\Phi}^{(h)} & \boldsymbol{\Phi}^{(t)} & \boldsymbol{O} \end{array} \right],$$

$$\boldsymbol{\Delta}_3 = \left[\begin{array}{ccc} \boldsymbol{O} & \boldsymbol{\Phi}^{(h)} & \boldsymbol{\Phi}^{(t)} \end{array} \right].$$

$\boldsymbol{O}$ is an $L \times M$ zero matrix. $\boldsymbol{\Delta}_2$ is the middle part of $\boldsymbol{\Phi}$ with dimension $(M-L) \times 3M$ when both the head part and the tail part are cut,

$$\boldsymbol{\Delta}_2 = \left[\begin{array}{ccc} \boldsymbol{O} & \boldsymbol{\Phi}^{(m)} & \boldsymbol{O} \end{array} \right].$$

Here $\boldsymbol{O}$ is also a zero matrix with dimension $(M-L) \times M$. According to (3.30), we have

$$\boldsymbol{\Delta}_2 \, \boldsymbol{\Delta}_2^T = \boldsymbol{\Phi}^{(m)} [\boldsymbol{\Phi}^{(m)}]^T = \boldsymbol{I},$$

and

$$\boldsymbol{\Delta}_1 \, \boldsymbol{\Delta}_2^T = \boldsymbol{\Phi}^{(t)} [\boldsymbol{\Phi}^{(m)}]^T = \boldsymbol{O}_L,$$

$$\boldsymbol{\Delta}_1 \, \boldsymbol{\Delta}_3^T = \boldsymbol{\Phi}^{(t)} [\boldsymbol{\Phi}^{(h)}]^T = \boldsymbol{O}_L,$$

$$\boldsymbol{\Delta}_2 \, \boldsymbol{\Delta}_3^T = \boldsymbol{\Phi}^{(m)} [\boldsymbol{\Phi}^{(h)}]^T = \boldsymbol{O},$$

where $\boldsymbol{I}$ is an $(M-L) \times (M-L)$ identity matrix , $\boldsymbol{O}_L$ is an $L \times L$ zero matrix, and $\boldsymbol{O}$ is an $(M-L) \times L$ zero matrix. In addition, based on (3.31) and (3.32), it is easily found that

$$\boldsymbol{\Delta}_1 \, \boldsymbol{\Delta}_1^T = \boldsymbol{\Delta}_3 \, \boldsymbol{\Delta}_3^T = \boldsymbol{R}_1 + \boldsymbol{R}_2 = \boldsymbol{I}_L,$$

where $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ are defined in (3.30). (3.36) easily follows. □

The unitarity property in the above theorem leads to the perfect reconstruction condition of DLS and DLC transforms. This condition will be discussed in detail in the next section.

3.5 Sufficient and Necessary Conditions for Perfect Reconstruction

Definition 3.1 *A perfect reconstruction is defined as*

$$\tilde{\boldsymbol{\Phi}} \tilde{\boldsymbol{\Phi}}^T = \boldsymbol{I}_{(M+L)},$$

where $\tilde{\boldsymbol{\Phi}}$ is defined by (3.33) and $\boldsymbol{I}_{M+L}$ is an $(M+L) \times (M+L)$ identity matrix.

Lemma 3.1 *For $M \times M$ square matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, if*

$$\boldsymbol{A}\boldsymbol{A}^T + \boldsymbol{B}\boldsymbol{B}^T = \boldsymbol{I}_M \tag{3.37}$$

$$\boldsymbol{A}\boldsymbol{B}^T = \boldsymbol{O}_M. \tag{3.38}$$

where $\boldsymbol{I}_M$ is an $M \times M$ identity matrix and $\boldsymbol{O}_M$ is a zero matrix of the same size, we have

$$\boldsymbol{A}^T\boldsymbol{B} = \boldsymbol{O}_M;$$

$$\boldsymbol{A}^T\boldsymbol{A} + \boldsymbol{B}^T\boldsymbol{B} = \boldsymbol{I}_M;$$

Proof: Let $\boldsymbol{R} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{B} & \boldsymbol{A} \end{bmatrix}$. Then, we have from (3.37) and (3.38),

$$\boldsymbol{R}\boldsymbol{R}^T = \begin{bmatrix} \boldsymbol{A}\boldsymbol{A}^T + \boldsymbol{B}\boldsymbol{B}^T & \boldsymbol{A}\boldsymbol{B}^T + \boldsymbol{B}\boldsymbol{A}^T \\ \boldsymbol{B}\boldsymbol{A}^T + \boldsymbol{A}\boldsymbol{B}^T & \boldsymbol{A}\boldsymbol{A}^T + \boldsymbol{B}\boldsymbol{B}^T \end{bmatrix} = \boldsymbol{I}_{2M}.$$

Now since $\boldsymbol{R}$ is square matrix, its inverse is unique and is given by $\boldsymbol{R}^{-1} = \boldsymbol{R}^T$ from above. Thus, $\boldsymbol{R}^{-1}\boldsymbol{R} = \boldsymbol{R}^T\boldsymbol{R} = \boldsymbol{I}_{2M}$ from which we conclude

$$\boldsymbol{A}^T\boldsymbol{A} + \boldsymbol{B}^T\boldsymbol{B} = \boldsymbol{I}_M. \tag{3.39}$$

Pre-multiplying by $\boldsymbol{A}^T$ and post-multiplying by $\boldsymbol{A}$ on both sides of (3.37), we obtain

$$\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{A}^T + \boldsymbol{B}\boldsymbol{B}^T)\boldsymbol{A} = \boldsymbol{A}^T\boldsymbol{A}.$$

From the above formula, we have

$$\boldsymbol{A}^T\boldsymbol{B}\boldsymbol{B}^T\boldsymbol{A} = \boldsymbol{A}^T\boldsymbol{A} - \boldsymbol{A}^T\boldsymbol{A}\boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{A}^T\boldsymbol{A}(\boldsymbol{I}_M - \boldsymbol{A}^T\boldsymbol{A}).$$

Substituting (3.39) into the above formula, we have

$$\boldsymbol{A}^T\boldsymbol{B}\boldsymbol{B}^T\boldsymbol{A} = \boldsymbol{A}^T\boldsymbol{A}\boldsymbol{B}^T\boldsymbol{B}. \tag{3.40}$$

Using (3.38), (3.40) becomes

$$\boldsymbol{A}^T\boldsymbol{B}\boldsymbol{B}^T\boldsymbol{A} = \boldsymbol{O}_M. \tag{3.41}$$

Letting

$$\boldsymbol{G} = \boldsymbol{A}^T\boldsymbol{B},$$

(3.41) becomes $\boldsymbol{G}\boldsymbol{G}^T = \boldsymbol{O}_M$. According to Graybill [56], we have $\boldsymbol{G} = \boldsymbol{A}^T\boldsymbol{B} = \boldsymbol{O}_M$. Combining the above results and (3.39), the lemma is proven. □

The necessary and sufficient conditions for perfect reconstruction using lapped transforms are the result of the following two theorems.

Theorem 3.2 *Consider a lapped transform matrix $\boldsymbol{\Phi}$ which is defined in (3.3) and let $M = L$. Let $\tilde{\boldsymbol{\Phi}}$ be defined by (3.33). Then, we have*

$$\tilde{\boldsymbol{\Phi}}\tilde{\boldsymbol{\Phi}}^T = \boldsymbol{I}_{2M} \tag{3.42}$$

if and only if $\boldsymbol{\Phi}^T\boldsymbol{\Phi} = \boldsymbol{I}_M$ and $\boldsymbol{\Phi}^T\boldsymbol{W}\boldsymbol{\Phi} = \boldsymbol{O}_M$. $\boldsymbol{I}_M$ and $\boldsymbol{O}_M$ represent $M \times M$ identity and zero matrices. $\boldsymbol{W}$ is an $(M+L)\times(M+L)$ shift matrix defined in (2.41).

Proof: The sufficient condition is the direct result of Theorem 3.1. Now we will prove the necessary condition. Since $M = L$, the block $\boldsymbol{\Delta}_2$ in the proof of Theorem 3.1 vanishes and $\tilde{\boldsymbol{\Phi}}$ can be written as

$$\begin{bmatrix} \boldsymbol{\Phi}^{(h)} & \boldsymbol{\Phi}^{(t)} & \boldsymbol{O}_M \\ \boldsymbol{O}_M & \boldsymbol{\Phi}^{(h)} & \boldsymbol{\Phi}^{(t)} \end{bmatrix}, \tag{3.43}$$

where $\boldsymbol{\Phi}^{(h)}$ and $\boldsymbol{\Phi}^{(t)}$ are $M \times M$ square matrices. From $\tilde{\boldsymbol{\Phi}}\tilde{\boldsymbol{\Phi}}^T = \boldsymbol{I}_{2M}$, we can easily derive that

$$\begin{cases} \boldsymbol{\Phi}^{(h)}[\boldsymbol{\Phi}^{(h)}]^T + \boldsymbol{\Phi}^{(t)}[\boldsymbol{\Phi}^{(t)}]^T = \boldsymbol{I}_M \\ \boldsymbol{\Phi}^{(h)}[\boldsymbol{\Phi}^{(t)}]^T = \boldsymbol{\Phi}^{(t)}[\boldsymbol{\Phi}^{(h)}]^T = \boldsymbol{O}_M \end{cases}. \tag{3.44}$$

According to Lemma 3.1, we have

$$\begin{cases} [\boldsymbol{\Phi}^{(h)}]^T\boldsymbol{\Phi}^{(h)} + [\boldsymbol{\Phi}^{(t)}]^T\boldsymbol{\Phi}^{(t)} = \boldsymbol{I}_M \\ [\boldsymbol{\Phi}^{(h)}]^T\boldsymbol{\Phi}^{(t)} = [\boldsymbol{\Phi}^{(t)}]^T\boldsymbol{\Phi}^{(h)} = \boldsymbol{O}_M \end{cases}. \tag{3.45}$$

The first equation of (3.45) indicates the orthogonality property of lapped transform $\boldsymbol{\Phi}$, and the second equation corresponds to lapped orthogonality. Now, the theorem is proven. □

For $M > L$, we may use a similar idea to get sufficient and necessary conditions for perfect reconstruction. The result is presented in Theorem 3.3. This is based on Lemma 3.2 and Lemma 3.3 which we now introduce.

Lemma 3.2 *Consider two $L \times M$ matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ and an $(M-L) \times M$ matrix $\boldsymbol{Q}$. If the following equations are satisfied*

$$\boldsymbol{A}\boldsymbol{A}^T + \boldsymbol{B}\boldsymbol{B}^T = \boldsymbol{I}_L \tag{3.46}$$

$$\boldsymbol{Q}\boldsymbol{Q}^T = \boldsymbol{I}_{M-L} \tag{3.47}$$

$$\boldsymbol{A}\boldsymbol{B}^T = \boldsymbol{O}_L \tag{3.48}$$

$$\boldsymbol{B}\boldsymbol{Q}^T = \boldsymbol{A}\boldsymbol{Q}^T = \boldsymbol{O}_{L\times(M-L)} \tag{3.49}$$

$$\boldsymbol{Q}\boldsymbol{B}^T = \boldsymbol{Q}\boldsymbol{A}^T = \boldsymbol{O}_{(M-L)\times L}, \tag{3.50}$$

we have

$$\boldsymbol{A}^T\boldsymbol{A} + \boldsymbol{B}^T\boldsymbol{B} + \boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{I}_M \tag{3.51}$$

$$\boldsymbol{A}^T\boldsymbol{B} + \boldsymbol{B}^T\boldsymbol{A} = \boldsymbol{O}_M. \tag{3.52}$$

Proof: The proof can be found in Appendix C.

Lemma 3.3 *Consider two $L \times M$ matrices $\boldsymbol{A}$ and $\boldsymbol{B}$. If $\boldsymbol{A}^T\boldsymbol{B} + \boldsymbol{B}^T\boldsymbol{A} = \boldsymbol{O}_M$ and $\boldsymbol{A}\boldsymbol{B}^T = \boldsymbol{B}\boldsymbol{A}^T = \boldsymbol{O}_L$, then $\boldsymbol{A}^T\boldsymbol{B} = \boldsymbol{B}^T\boldsymbol{A} = \boldsymbol{O}_M$.*

Proof: The proof can be found in Appendix D.

With Lemmas 3.2 and 3.3, we are ready to prove the following theorem as a general case of Theorem 3.2.

Theorem 3.3 *Consider the general case of Theorem 3.2, $M > L$. Given a lapped transform matrix $\boldsymbol{\Phi}$ with $M > L$ and letting $\tilde{\boldsymbol{\Phi}}$ be defined by (3.33), we have*

$$\tilde{\boldsymbol{\Phi}}\tilde{\boldsymbol{\Phi}}^T = \boldsymbol{I}_{M+L}, \tag{3.53}$$

if and only if $\boldsymbol{\Phi}^T\boldsymbol{\Phi} = \boldsymbol{I}_M$ and $\boldsymbol{\Phi}^T\boldsymbol{W}\boldsymbol{\Phi} = \boldsymbol{O}_M$.

Proof: The proof can be found in Appendix E.

Theorem 3.3 indicates that although the DLS and DLC basis functions do not form a square matrix, if the matrix satisfies orthogonality and lapped orthogonality property, a perfect reconstruction can still be obtained. This property can also be applied to any lapped orthogonal transform.

3.6 Scaling-in-Time

For any basis functions used in signal processing, time and frequency resolutions are important characteristics. This is the context to be discussed in this section. Let the frequency resolution of a DLS transform with M basis functions be Δf and its sampling interval be Δt. From the definition of the DLS transform, it is known that they are related to the bandwidth (B) of the basis functions as

$$\Delta f = \frac{B}{M},$$
$$\Delta t = \frac{1}{2B}.$$

Then we have

$$\Delta t \Delta f = \frac{1}{2M}.$$

When Δt increases by a factor a; i.e., $\Delta t' = a\Delta t$, Δf will decrease by the same factor: i.e.,

$$\Delta f' = \frac{\Delta f}{a},$$

while the product of frequency resolution and the sampling interval remains a constant. This result indicates that for the DLS transform, scaling-in-time leads to scaling-in-frequency. This property is the same as the one in DCT [66], and it can be well applied in multiresoultion signal analysis, where we may increase the frequency resolution by increasing the sampling interval of the basis functions.

3.7 Shift-in-Time

In data processing, we often deal with sets of data with different time shifts or delays. It is better to have a transform which has a time shift invariant property. In this section, we will examine the lapped orthogonal transform for a set of data which is shifted in time. The property discussed in this section will also be used later for the derivation of the property: difference-in-time.

Let the input data sequence be $\{x(n)\}$, where $n = 0, 1, \cdots, M + L - 1$. We shift the input data sequence by 1 sample interval: i.e. let the shifted sequence be $\{x_+(n)\}$, where $x_+(n) = x(n+1)$, $n = 0, 1, \cdots, M + L - 1$. The corresponding lapped transformed

coefficients are given by

$$\begin{cases} X(r) = \sum_{n=0}^{M+L-1} \tilde{\phi}_r(n) x(n), \\ X_+(r) = \sum_{n=0}^{M+L-1} \tilde{\phi}_r(n) x_+(n), \end{cases}$$

where $\tilde{\phi}_r(n)$ is the n-th element of the r-th column vector in matrix $\tilde{\boldsymbol{\Phi}}$, as defined in (3.33). In matrix form of the above equations are

$$\begin{cases} \boldsymbol{X} = \tilde{\boldsymbol{\Phi}}^T \boldsymbol{x} \\ \boldsymbol{X}_+ = \tilde{\boldsymbol{\Phi}}^T \boldsymbol{x}_+, \end{cases} \tag{3.54}$$

where T is the transpose operation, and

$$\begin{cases} \boldsymbol{x} = [x(0), x(1), \cdots, x(M+L-1)]^T \\ \boldsymbol{x}_+ = [x(1), x(2), \cdots, x(M+L-1), x(M+L)]^T, \end{cases} \tag{3.55}$$

and

$$\begin{cases} \boldsymbol{X} = [X(0), X(1), \cdots, X(3M-1)]^T \\ \boldsymbol{X}_+ = [X_+(0), X_+(1), \cdots, X_+(3M-1)]^T. \end{cases}$$

Pre-multiplying $\tilde{\boldsymbol{\Phi}}$ on both sides of (3.54) and using Theorem 3.1, we have

$$\begin{cases} \boldsymbol{x} = \tilde{\boldsymbol{\Phi}}\, \boldsymbol{X}, \\ \boldsymbol{x}_+ = \tilde{\boldsymbol{\Phi}}\, \boldsymbol{X}_+. \end{cases} \tag{3.56}$$

Now, from (3.55) we may write

$$\boldsymbol{x}_+ = [(\boldsymbol{T}\, \boldsymbol{x})^T, x(M+L)]^T, \tag{3.57}$$

where $\boldsymbol{T}$ is an $(M+L-1) \times (M+L)$ shift matrix defined by

$$\boldsymbol{T} = \left[\begin{array}{cc} \boldsymbol{O} & \boldsymbol{I}_{M+L-1} \end{array} \right], \tag{3.58}$$

and $\boldsymbol{O}$ is a zero column vector with length $M+L-1$. Substituting the first equation in (3.56) into (3.57) and then substituting the result into the second equation of (3.54), we have

$$\begin{aligned} \boldsymbol{X}_+ &= \tilde{\boldsymbol{\Phi}}^T \boldsymbol{x}_+ \\ &= \tilde{\boldsymbol{\Phi}}^T [(\boldsymbol{T}\, \tilde{\boldsymbol{\Phi}}\, \boldsymbol{X})^T, x(M+L)]^T \\ &= \tilde{\boldsymbol{\Phi}}^T \left\{ [(\boldsymbol{T}\, \tilde{\boldsymbol{\Phi}}\, \boldsymbol{X})^T, 0]^T + [0, \cdots, 0, x(M+L)]^T \right\}, \end{aligned} \tag{3.59}$$

from which no simplifications can be made. It is clear that lapped transform does not have the time shift invariant property as does Fourier transform [41]. However, the property discussed in this section indicates that the transformed coefficients may be updated from the coefficients in the adjacent block immediately in front.

3.8 Uniqueness

For any transform, it is important to know whether the data and its transform are uniquely related or not. The uniqueness property of lapped transform is stated as follows.

Theorem 3.4 *Consider a column vector $\boldsymbol{x}$. Its lapped transform is given by*

$$\boldsymbol{X} = \tilde{\boldsymbol{\Phi}}^T \boldsymbol{x}, \tag{3.60}$$

where $\tilde{\boldsymbol{\Phi}}$ is defined in (3.33); i.e.,

$$\tilde{\boldsymbol{\Phi}}\, \tilde{\boldsymbol{\Phi}}^T = \boldsymbol{I}_{M+L}. \tag{3.61}$$

Then we have

1.

$$\boldsymbol{x} = \tilde{\boldsymbol{\Phi}}\, \tilde{\boldsymbol{\Phi}}^T \boldsymbol{x}, \quad \textit{for all } \boldsymbol{x}; \tag{3.62}$$

2. a).

$$\boldsymbol{X} = \tilde{\boldsymbol{\Phi}}^T \tilde{\boldsymbol{\Phi}}\, \boldsymbol{X}; \quad \textit{not for all } \boldsymbol{X}, \textit{ but only for } \boldsymbol{X} = \tilde{\boldsymbol{\Phi}}^T \boldsymbol{x}; \tag{3.63}$$

b). when $M \longrightarrow \infty$

$$\boldsymbol{X} = \tilde{\boldsymbol{\Phi}}^T \tilde{\boldsymbol{\Phi}}\, \boldsymbol{X}, \qquad \textit{for almost all } \boldsymbol{X}. \tag{3.64}$$

Proof: 1. We prove (3.62) as follows. Let $\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{\Phi}}\, \tilde{\boldsymbol{\Phi}}^T \boldsymbol{x}$. Using (3.61), we have $\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{\Phi}}\, \tilde{\boldsymbol{\Phi}}^T \boldsymbol{x} = \boldsymbol{I}_{M+L} \boldsymbol{x} = \boldsymbol{x}$. Therefore, for all $\boldsymbol{x}$ $\boldsymbol{x} = \tilde{\boldsymbol{\Phi}}\, \tilde{\boldsymbol{\Phi}}^T \boldsymbol{x}$, which proves (3.62).

2. a). We prove (3.63) as follows. Let

$$\boldsymbol{x} = \tilde{\boldsymbol{\Phi}}\, \tilde{\boldsymbol{X}}. \tag{3.65}$$

Then substituting (3.65) into (3.60), we have

$$\boldsymbol{X} = \tilde{\boldsymbol{\Phi}}^T \tilde{\boldsymbol{\Phi}}\, \tilde{\boldsymbol{X}}. \tag{3.66}$$

Since $\tilde{\boldsymbol{\Phi}}$ is not a matrix of full column rank, we have, in general $\tilde{\boldsymbol{\Phi}}^T\tilde{\boldsymbol{\Phi}} \neq \boldsymbol{I}$. The above equation implies that in general $\boldsymbol{X} \neq \tilde{\boldsymbol{X}}$, and $\boldsymbol{X} = \tilde{\boldsymbol{X}}$ only if $\tilde{\boldsymbol{X}} = \tilde{\boldsymbol{\Phi}}^T\boldsymbol{x}$. b) We prove (3.64) as follows. By definition we know that when $M \longrightarrow \infty$,$\tilde{\boldsymbol{\Phi}}^T\tilde{\boldsymbol{\Phi}} \longrightarrow \boldsymbol{I}$, i.e., $\tilde{\boldsymbol{\Phi}}$ approaches a unitary matrix. In such a case, we have $\boldsymbol{X} = \tilde{\boldsymbol{\Phi}}^T\tilde{\boldsymbol{\Phi}}\,\tilde{\boldsymbol{X}} \longrightarrow \tilde{\boldsymbol{X}}$. □

The above proof indicates that given $\boldsymbol{x}$, $\boldsymbol{X}$ is uniquely determined while the reverse is not true. This result means that different $\boldsymbol{x}$'s may have the same transform $\boldsymbol{X}$. However, we have shown that $\boldsymbol{x}$ and $\boldsymbol{X}$ are asymptotically one-to-one as $M \rightarrow \infty$.

3.9 Difference-in-Time

In signal processing, we often need to calculate the differences between successive samples. In DPCM (Differential Pulse Code Modulation), these differences are coded and then transmitted. In this section, we discuss the corresponding relation in the transform domain.

Consider the input data sequence $\{x(n)\}, n = 0, 1, \cdots, M + L - 1$. A difference sequence between two adjacent samples may be defined by

$$d(n) = x(n+1) - x(n), \quad n = 0, 1, \cdots, M + L - 1.$$

In matrix form, we have

$$\boldsymbol{d} = \boldsymbol{x}_+ - \boldsymbol{x},$$

where $\boldsymbol{x}$ and $\boldsymbol{x}_+$ are defined in (3.55). Here, we may use shift-in-time property to describe the difference in the transform domain: i.e.,

$$\boldsymbol{D} = \boldsymbol{X}_+ - \boldsymbol{X},$$

where $\boldsymbol{D}$ is the lapped transform of $\boldsymbol{d}$ and $\boldsymbol{X}_+$ is lapped transform of $\boldsymbol{x}_+$, which is given by (3.59). It is noted have that $\boldsymbol{X}_+$ can be updated from $\boldsymbol{X}$ as shown by (3.59). Thus, the computation for $\boldsymbol{D}$, although not as simple as in the case of Fourier transform, does not require evaluating $\boldsymbol{X}_+$ and $\boldsymbol{X}$ independently.

3.10 Convolution

Convolution is an important operation in signal processing and filtering. The Fourier transform has the property that a convolution in the time domain corresponds to multiplication in the transform domain. In this section, we discuss the relation between the circular convolution and the multiplication for lapped transforms. Consider two data sequences $\{x(n)\}$ and $\{y(n)\}$, $n = 0, 1, \cdots, M+L-1$. The circular convolution of $x(n)$ and $y(n)$ and its lapped transform are given respectively by

$$\begin{cases} a(i) = x(n) * y(n) = \sum_{n=0}^{M+L-1} x(n) y|i-n|_{mod(M+L)}, \\ A(r) = \sum_{i=0}^{M+L-1} a(i)\tilde{\phi}_r(i), \end{cases} \tag{3.67}$$

where $*$ is the convolution operation. Let $\boldsymbol{x} = [x(0), x(1), \cdots, x(M+L-1)]^T$, $\boldsymbol{y} = [y(0), y(1), \cdots, y(M+L-1)]^T$, $\boldsymbol{a} = [a(0), a(1), \cdots, a(M)]^T$, and $\boldsymbol{A} = [A(0), A(1), \cdots, A(M)]^T$. The matrix form of (3.67) is then

$$\begin{cases} \boldsymbol{a} = \boldsymbol{\chi}\, \boldsymbol{x} \\ \boldsymbol{A} = \tilde{\boldsymbol{\Phi}}^T \boldsymbol{a}, \end{cases} \tag{3.68}$$

where $\boldsymbol{\chi}$ is defined as

$$\boldsymbol{\chi} = \begin{bmatrix} y(0) & y(M+L-1) & \cdots & y(1) \\ y(1) & y(0) & & \\ \vdots & & \ddots & \vdots \\ y(M+L-1) & & \cdots & y(0) \end{bmatrix}.$$

Substituting the first equation of (3.68) into the second one, $\boldsymbol{A}$ can be expressed as

$$\begin{aligned} \boldsymbol{A} &= \tilde{\boldsymbol{\Phi}}^T \boldsymbol{\chi}\, \boldsymbol{x} \\ &= \tilde{\boldsymbol{\Phi}}^T (\boldsymbol{\chi}_1 + \boldsymbol{\chi}_2) \boldsymbol{x}, \end{aligned}$$

where $\boldsymbol{\chi}_1$ and $\boldsymbol{\chi}_2$ are the lower triangular and upper triangular components of $\boldsymbol{\chi}$ defined by

$$\chi_1 = \begin{bmatrix} y(0) & 0 & \cdots & 0 \\ y(1) & y(0) & & \\ \vdots & & \ddots & \\ y(M+L-1) & & \cdots & y(0) \end{bmatrix},$$

$$\chi_2 = \begin{bmatrix} 0 & y(M+L-1) & \cdots & y(1) \\ \vdots & 0 & \cdots & y(2) \\ \vdots & & \ddots & y(M+L-1) \\ 0 & & \cdots & 0 \end{bmatrix}.$$

Applying the shift-in-time property to each column of χ_1 and χ_2, we obtain the transform of circular convolution as

$$\boldsymbol{A} = \tilde{\boldsymbol{\Phi}}^T [\boldsymbol{E}^{(0)} \tilde{\boldsymbol{\Phi}}\, \boldsymbol{Y}, \boldsymbol{E}^{(1)} \tilde{\boldsymbol{\Phi}}\, \boldsymbol{Y}, \cdots, \boldsymbol{E}^{(M+L-1)} \tilde{\boldsymbol{\Phi}}\, \boldsymbol{Y}] \tilde{\boldsymbol{\Phi}}\, \boldsymbol{X}, \tag{3.69}$$

where $\boldsymbol{Y}$ is transform of $\boldsymbol{y}$ and $\boldsymbol{E}^{(i)} = \begin{bmatrix} \boldsymbol{O} & \boldsymbol{I}_i \\ \boldsymbol{I}_{M+L-i} & \boldsymbol{O} \end{bmatrix}$.

When two signals are circularly convoluted in time domain, (3.69) gives the corresponding relation in the transform domain. It can be observed that there does not exist a simple relation in the transform domain. However, (3.69) is still useful for the description of a circular convolution. From this relation, we understand how the transform is changed with a circular convolution. Linear convolution, a very common computation in signal processing, is related to the circular convolution through some manipulations [41].

3.11 Performance Analysis

In this section, we compare the performance of different transforms: DLS, DLC, MLT, LOT, and DCT. Some common criteria for evaluating a transform, such as variance distribution, energy packing efficiency (EPE) and coding gain, are used in the comparisons. Corresponding to these criteria, since DLC has same performance as DLS, we are not going to show the results using DLC.

3.11.1 Variance Distribution

Consider a lapped transform $\boldsymbol{\Phi}$ which is an $(M+L) \times M$ matrix. According to (2.36), the transform of a discretized signal $\boldsymbol{x}$ is given by

$$\boldsymbol{X} = \boldsymbol{\Phi}^T \boldsymbol{x}, \tag{3.70}$$

where $\boldsymbol{x}$ is a column vector with length $(M+L)$. From (3.70), and the definition for the auto-correlation matrix, $\boldsymbol{R}_{xx}$, the transform domain correlation $\boldsymbol{R}_{XX}$ is

$$\boldsymbol{R}_{XX} = \boldsymbol{\Phi}^T \boldsymbol{R}_{xx} \boldsymbol{\Phi}. \tag{3.71}$$

The traces of $\boldsymbol{R}_{XX}$ and $\boldsymbol{R}_{xx}$ are equal and represent the signal energy. It is known that in data compression [16], we would like to have the signal energy concentrated in a few diagonal elements, so that the original signal can be approximately recovered with only a few transform coefficients. The diagonal elements of $\boldsymbol{R}_{XX}$ represent the variances of the transform coefficients. A transform is judged to be efficient if significant variances occur in only a few coefficients. Figures 3.3 and 3.4 show the variance as a function of the index of the transform coefficient based on (3.71) for DCT, LOT, MLT, DLS transforms. Here, we choose a signal to be a first-order Markov signal with a correlation coefficient of $\rho = 0.9$ and the number of bases to be $M = 8$ and $M = 16$ respectively. For DLS transform, we use $L = 8$ and $L = 16$ respectively. The reason that we choose $L = M$ is that this choice gives the smoothest decay of the basis function so that the block effect is minimized. Also choosing $L = M$ makes DLS comparable with LOT and MLT which both have the length $L = M$.

From Figure 3.3, it can be observed that DLS has better performance than either DCT or LOT. DLS transform has its energy concentrated on the first three transform coefficients, and other transform coefficients have a very small energy. The variances for LOT do not decay in an uniform way. Sometimes they decrease very fast and sometimes the energy is almost unchanged between the adjacent coefficients. In general, the first few coefficients of the DLS contains more energy than those of the DCT, LOT and MLT. Similar results can be observed in Figure 3.4 where $M = 16$. In this case, the DLS has most of its energy concentrated in the first five to six transform coefficients.

3.11.2 Energy Packing Efficiency (EPE)

Energy packing efficiency, which describes the portion of energy contained in the first K coefficients of a total of M transform coefficients, is defined by [38]

$$EPE(K) = \frac{\sum_{p=0}^{K-1} \sigma_p^2}{\sum_{p=0}^{M-1} \sigma_p^2}, \tag{3.72}$$

where σ_p^2 is the p-th diagonal element in $\boldsymbol{R}_{XX}$. Figures 3.5 and 3.6 show EPE in cases of $M = 8$ and $M = 16$ respectively. The signal is again first-order Markov with $\rho = 0.9$; all the parameters are the same as in Figures 3.3 and 3.4. It can be observed that when the ratio $\frac{K}{M}$ is in the range from 0.1 to 0.5, there are obvious differences among different transforms. The DLS transform is consistently better (having higher EPE value for the same $\frac{K}{M}$) than both the DCT and LOT when $\frac{K}{M} > .2$ for $M = 8$ and when $\frac{K}{M} > .15$ for $M = 16$. Both figures show that the DLS transform has higher EPE values after the first transform coefficients because the first coefficients of the DLS transform has smaller energy than either of the DCT and LOT as shown in Figures 3.3 and 3.4. MLT is almost same as DLS. After the first coefficients, the DLS transform has higher energy concentration in fewer coefficients than either of the DCT and LOT. Also, after the first two coefficients, DLS and DLC transforms have almost same energy concentration, comparing with MLT.

3.11.3 Coding Gain

The transform coding gain G_{TC} [16] is defined as

$$G_{TC} = \frac{\frac{1}{M}\sum_{p=0}^{M-1} \sigma_p^2}{(\prod_{p=0}^{M-1} \sigma_p^2)^{1/M}}, \tag{3.73}$$

which is used in the analysis of the performance for various transforms in data compression. The numerator indicates the average energy in $\boldsymbol{X}$, and the denominator represents the geometric mean of transform coefficient variances $\{\sigma_p^2\}$. It can be seen that maximizing G_{TC} is equivalent to minimizing the denominator. Minimizing the denominator implies that only a few coefficients hold the most energy of a signal. This is the optimal criterion used in KLT. Table 3.1 shows the coding gains of different transforms for the same Markov-I signal as in Figures 3.3 to 3.6. Again, we choose $M = 8$ and $M = 16$

Table 3.1: **Coding gain for $M = 16$ and $\rho = 0.9$**

transform	*coding gain*
DCT (M=8)	4.2424
LOT (M=8, L=8)	4.2587
MLT (M=8, L=8)	4.7091
DLS (M=8, L=8)	4.3229
DLC (M=8, L=8)	4.3229
DCT (M=16)	4.7058
LOT (M=16, L=16)	4.6896
MLT (M=16, L=16)	5.0826
DLS (M=16, L=16)	4.9772
DLC (M=16, L=16)	4.9772

respectively for each transform in comparison. It can be observed that for a given size, DLS and DLC transforms have higher coding gains than either of DCT and LOT for the same M. The reason is similar to those for the variance distribution and energy packing efficiency: both the DLS and DLC transforms have higher energy concentration in fewer coefficients. However, MLT has the highest coding gain among all of these transforms.

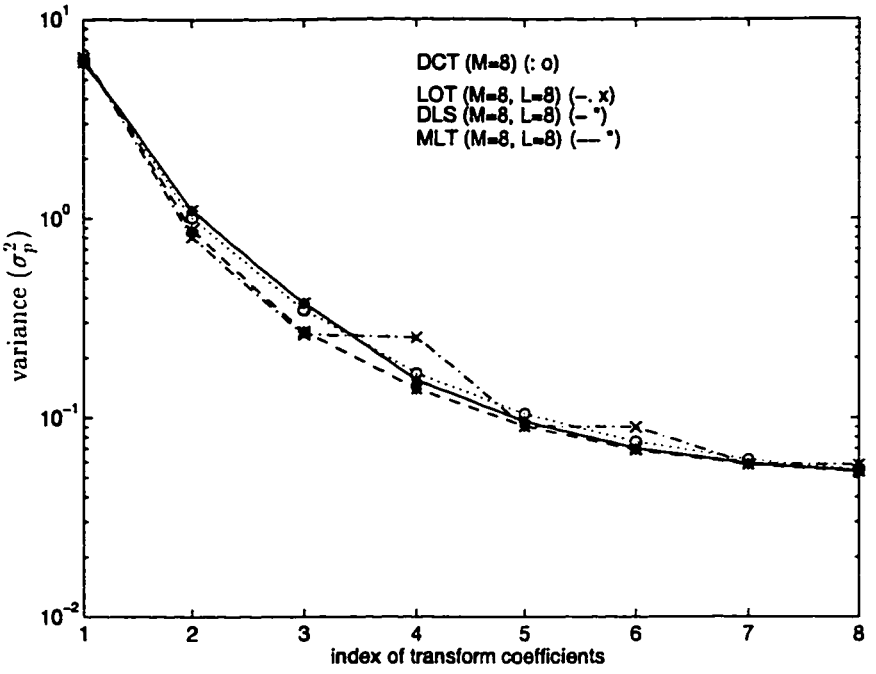


Figure 3.3: Variance distribution for $M = 8$ and $\rho = 0.9$

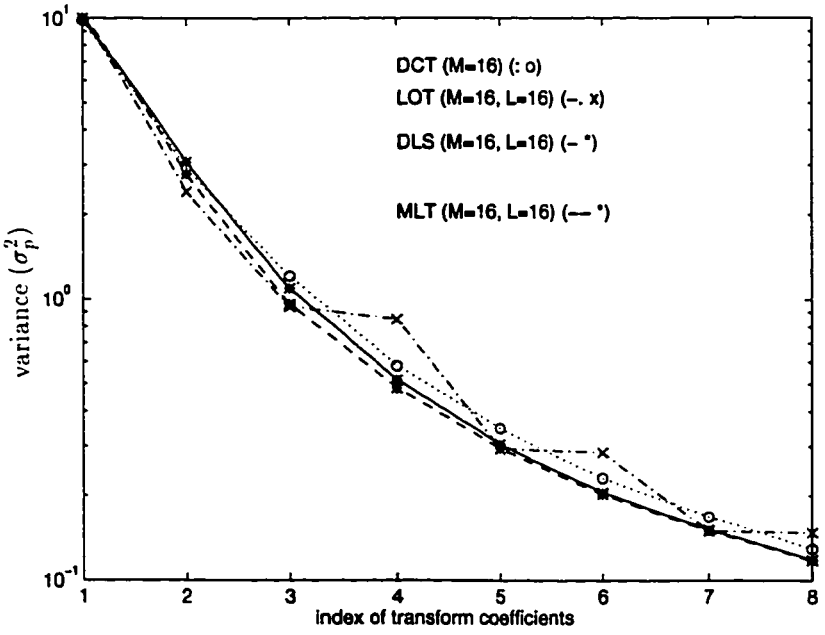


Figure 3.4: Variance distribution for $M = 16$ and $\rho = 0.9$

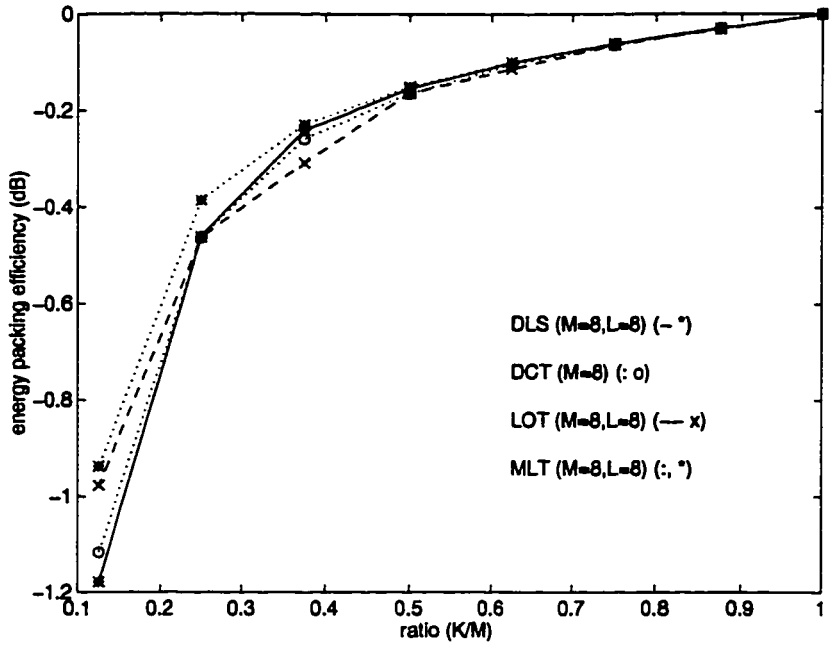


Figure 3.5: Energy packing efficiency for $M = 8$ and $\rho = 0.9$

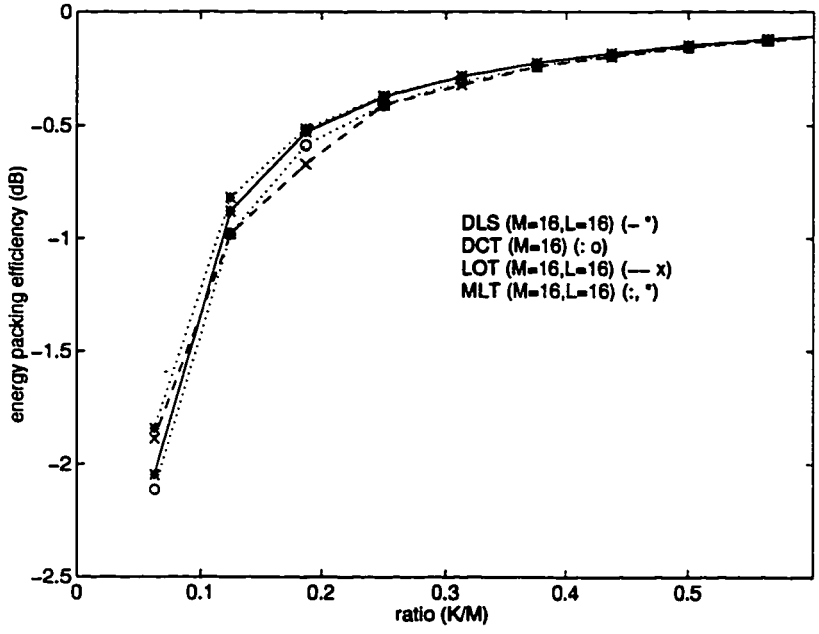


Figure 3.6: Energy packing efficiency for $M = 16$ and $\rho = 0.9$

Chapter 4

Fast Algorithms for Computing DLS and DLC Transforms

4.1 Introduction

In this chapter, we derive the fast algorithms for computing the DLS and DLC transforms. Given a vector $\boldsymbol{x}$ with length $M+L$, the transformed vector is $\boldsymbol{X} = \boldsymbol{\Phi}^T\boldsymbol{x}$. It is easy to see that such direct matrix computation is laborious and time consuming. However, by recognizing a particular structure of the matrix $\boldsymbol{\Phi}$, one may be able to significantly reduce the amount of computation required in the transformation. This computation reduction can be realized by decomposing $\boldsymbol{\Phi}^T$ into a set of sparse matrices. In this chapter we will describe this sparse matrix factorization. First $\boldsymbol{\Phi}^T$ is partitioned into $\boldsymbol{\Gamma}^{(1)}$, $\boldsymbol{\Gamma}^{(2)}$ and $\boldsymbol{\Gamma}^{(3)}$, which are the transposes of $\boldsymbol{\Phi}^{(t)}$, $\boldsymbol{\Phi}^{(m)}$ and $\boldsymbol{\Phi}^{(h)}$ in (3.3), respectively. By examining these matrix structures, $\boldsymbol{\Phi}^T$ can be factorized into a set of Given's rotations and operations involving the discrete sine and cosine transform of type IV [66] (DST-IV and DCT-IV). Sparse factorizations of DST-IV and DCT-IV are also developed. The whole algorithm allows us to decompose an M order DLS/DLC transforms with length $M+L$ into a set of sparse matrices in Given's rotations and butterfly operations. Finally, the fast algorithm can be implemented based on these sparse matrix decompositions. Because of the similarity between the DLS and DLC transforms, we will describe only the fast algorithm for the DLS transform; a similar algorithm can be obtained for the DLC transform.

4.2 Partitioned Matrices

A DLS basis function is defined in (2.45): i.e.,

$$\phi_r(n) = \sqrt{\frac{2}{M}} b(n) \sin \frac{(2r+1)(2n-L+1)\pi}{4M}; \quad n \in [0, M+L-1], r \in [0, M-1], \tag{4.1}$$

Assume that L is an integer power of 2. Consider the matrix form $\boldsymbol{\Phi}$ which can be partitioned into three sub-matrices $\boldsymbol{\Phi}^{(t)}$, $\boldsymbol{\Phi}^{(m)}$ and $\boldsymbol{\Phi}^{(h)}$. Let $\boldsymbol{\Gamma}^{(1)}$, $\boldsymbol{\Gamma}^{(2)}$ and $\boldsymbol{\Gamma}^{(3)}$ be, respectively, the transposes of $\boldsymbol{\Phi}^{(t)}$, $\boldsymbol{\Phi}^{(m)}$ and $\boldsymbol{\Phi}^{(h)}$. $\boldsymbol{\Phi}^T$ is then written as

$$\boldsymbol{\Phi}^T = \left[\begin{array}{ccc} \boldsymbol{\Gamma}^{(1)} & \boldsymbol{\Gamma}^{(2)} & \boldsymbol{\Gamma}^{(3)} \end{array} \right], \tag{4.2}$$

where T represents transpose. We call $\boldsymbol{\Gamma}^{(1)}$, $\boldsymbol{\Gamma}^{(2)}$ and $\boldsymbol{\Gamma}^{(3)}$ *partitioned matrices.* According to (2.45) to (2.48), we have

$$[\boldsymbol{\Gamma}^{(1)}]_{rn} = a_n^{(b)} [\boldsymbol{S}]_{rn}; \quad r \in [0, M-1]; \; n \in [0, L-1],$$

$$[\boldsymbol{\Gamma}^{(2)}]_{rn} = [\boldsymbol{S}]_{r(n+L)}; \quad r \in [0, M-1]; \; n \in [0, L-1],$$

$$[\boldsymbol{\Gamma}^{(3)}]_{rn} = b_n^{(b)} [\boldsymbol{S}]_{r(n+M)}; \quad r \in [0, M-1]; \; n \in [0, L-1],$$

where $\boldsymbol{S}$ is a matrix of sinusoids given by

$$[\boldsymbol{S}]_{rn} = \sin \frac{(2r+1)(2n-L+1)\pi}{4M},$$

and, $a_n^{(b)} = S_e(n)$ and $b_n^{(b)} = C_e(n-M)$ are the elements of the column vectors $\boldsymbol{a}^{(b)}$ and $\boldsymbol{b}^{(b)}$ representing the bell functions of the tail region and of the head region, respectively. All these notations will be used in the remaining part of this chapter.

4.3 Sparse Factorization of Partitioned Matrices

Consider an input sequence $\{x(n)\}$, $n = 0, 1, \cdots, M+L-1$. The transform coefficients are given by

$$X(r) = \sum_{n=0}^{M+L-1} \phi_r(n) x(n).$$

Substituting (4.1) into the above formula, we have

$$\begin{aligned} X(r) &= \sum_{n=0}^{L-1} \sqrt{\frac{2}{M}} \sin\left[\frac{n\pi}{2(L-1)} - \frac{1}{4}\sin\frac{2n\pi}{L-1}\right] \sin\frac{(2r+1)(2n-L+1)\pi}{4M} x(n) \\ &+ \sum_{n=L}^{M-1} \sqrt{\frac{2}{M}} \sin\frac{(2r+1)(2n-L+1)\pi}{4M} x(n) \\ &+ \sum_{n=M}^{M+L-1} \sqrt{\frac{2}{M}} \cos\left[\frac{(n-M)\pi}{2(L-1)} - \frac{1}{4}\sin\frac{(n-M)\pi}{L-1}\right] \sin\frac{(2r+1)(2n-L+1)\pi}{4M} x(n). \end{aligned} \tag{4.3}$$

Define the first term in the above formula to be T_1, the second to be T_2 and the third to be T_3. These corresponds to the elements in the respective matrices of $\boldsymbol{\Gamma}^{(1)}$, $\boldsymbol{\Gamma}^{(2)}$ and $\boldsymbol{\Gamma}^{(3)}$. We examine the characteristics of each term as follows.

(1). T_1 can be rewritten as

$$\begin{aligned} T_1 &= \sum_{n=0}^{\frac{L}{2}-1} \sqrt{\frac{2}{M}} \sin\left[\frac{n\pi}{2(L-1)} - \frac{1}{4}\sin\frac{2n\pi}{L-1}\right] \sin\frac{(2r+1)(2n-L+1)\pi}{4M} x(n) \\ &+ \sum_{n=\frac{L}{2}}^{L-1} \sqrt{\frac{2}{M}} \sin\left[\frac{n\pi}{2(L-1)} - \frac{1}{4}\sin\frac{2n\pi}{L-1}\right] \sin\frac{(2r+1)(2n-L+1)\pi}{4M} x(n) \\ &= \sum_{n=0}^{\frac{L}{2}-1} \sqrt{\frac{2}{M}} \sin\left[\frac{n\pi}{2(L-1)} - \frac{1}{4}\sin\frac{2n\pi}{L-1}\right] \sin\frac{(2r+1)(2n-L+1)\pi}{4M} x(n) \\ &+ \sum_{n=0}^{\frac{L}{2}-1} \sqrt{\frac{2}{M}} \sin\left[\frac{(L-1-n)\pi}{2(L-1)} - \frac{1}{4}\sin\frac{2(L-1-n)\pi}{L-1}\right] \\ &\cdot \sin\frac{(2r+1)(L-1-2n)\pi}{4M} x(L-1-n) \\ &= \sum_{n=0}^{\frac{L}{2}-1} \sqrt{\frac{2}{M}} [\sin(\theta_n)x(n) - \cos(\theta_n)x(L-1-n)] \sin\frac{(2r+1)(2n-L+1)\pi}{4M}, \end{aligned} \tag{4.4}$$

where $\theta_n = \frac{n\pi}{2(L-1)} - \frac{1}{4}\sin\frac{2n\pi}{L-1}$. (4.4) indicates that $\boldsymbol{\Gamma}^{(1)}$ can be partitioned as

$$\boldsymbol{\Gamma}^{(1)} = \left[\begin{array}{c:c} \boldsymbol{D}_1^{(1)} & \boldsymbol{E}_1^{(1)} \end{array}\right]. \tag{4.5}$$

We call this factorization the first step or the first layer of decomposition, indicated by the subscript 1. The elements of $\boldsymbol{D}_1^{(1)}$ and $\boldsymbol{E}_1^{(1)}$ are given by

$$\left[\boldsymbol{D}_1^{(1)}\right]_{rn} = \sin(\theta_n)\sin\frac{(2r+1)(2n-L+1)\pi}{4M}; \quad n \in [0, \frac{L}{2}-1]; \ r \in [0, M-1]$$

$$\left[\boldsymbol{E}_1^{(1)}\right]_{r(\frac{L}{2}-1-n)} = -\cos(\theta_n)\sin\frac{(2r+1)(2n-L+1)\pi}{4M}; \quad n \in [0, \frac{L}{2}-1]; \ r \in [0, M-1].$$

By examining the structures of $\boldsymbol{D}_1^{(1)}$ and $\boldsymbol{E}_1^{(1)}$, it is found that $\boldsymbol{\Gamma}^{(1)}$ can be further partitioned by using an $L \times L$ sparse matrix $\boldsymbol{F}_1$, which is structured as Given's rotations and described as

$$\boldsymbol{F}_1 = \begin{bmatrix} \cos\theta_0 & & & & & \sin\theta_0 \\ & \ddots & & & \iddots & \\ & & \cos\theta_{\frac{L}{2}-1} & \sin\theta_{\frac{L}{2}-1} & & \\ & & -\sin\theta_{\frac{L}{2}-1} & \cos\theta_{\frac{L}{2}-1} & & \\ & \iddots & & & \ddots & \\ -\sin\theta_0 & & & & & \cos\theta_0 \end{bmatrix}. \tag{4.6}$$

Post-multiplying by $\boldsymbol{F}_1^T$ in (4.5), we have

$$\boldsymbol{\Gamma}^{(1)} \cdot \boldsymbol{F}_1^T = \left[\begin{array}{c:c} \boldsymbol{D}_2^{(1)} & \boldsymbol{E}_2^{(1)} \end{array} \right], \tag{4.7}$$

where $\boldsymbol{D}_2^{(1)}$ is an $M \times L$ zero matrix, and

$$\left[\boldsymbol{E}_2^{(1)}\right]_{r,\frac{L}{2}-1-n} = -[\boldsymbol{S}]_{rn}; \quad r \in [0, M-1]; \ n \in [0, \frac{L}{2}-1]. \tag{4.8}$$

By changing the index n in (4.8), the rn-th element of $\boldsymbol{E}_2^{(1)}$ can be rewritten as

$$\left[\boldsymbol{E}_2^{(1)}\right]_{rn} = \sin\frac{(2r+1)(2n+1)\pi}{4M}; \quad n \in [0, \frac{L}{2}-1]. \tag{4.9}$$

(2). According to (4.3), T_2 can be written as

$$\begin{aligned} T_2 &= \sum_{n=L}^{M-1} \sqrt{\frac{2}{M}} \sin\frac{(2r+1)(2n-L+1)\pi}{4M} x(n) \\ &= \sum_{n=0}^{M-L-1} \sqrt{\frac{2}{M}} \sin\frac{(2r+1)(2n+L+1)\pi}{4M} x(n+L) \\ &= \sum_{n=0}^{\frac{M-L}{2}-1} \sqrt{\frac{2}{M}} \sin\frac{(2r+1)(2n+L+1)\pi}{4M} x(n+L) \\ &\quad + \sum_{n=\frac{M-L}{2}}^{M-L-1} \sqrt{\frac{2}{M}} \sin\frac{(2r+1)(2n+L+1)\pi}{4M} x(n+L) \\ &= \sum_{n=0}^{\frac{M-L}{2}-1} \sqrt{\frac{2}{M}} \sin\frac{(2r+1)(2n+L+1)\pi}{4M} x(n+L) \\ &\quad + \sum_{n=0}^{\frac{M-L}{2}-1} (-1)^r \sqrt{\frac{2}{M}} \cos\frac{(2r+1)(2n+L+1)\pi}{4M} x(M-L-n). \end{aligned} \tag{4.10}$$

According to (4.10), we can partition $\boldsymbol{\Gamma}^{(2)}$ into two sub-matrices $\boldsymbol{D}_1^{(2)}$ and $\boldsymbol{E}_1^{(2)}$:

$$\boldsymbol{\Gamma}^{(2)} = \left[\begin{array}{c:c} \boldsymbol{D}_1^{(2)} & \boldsymbol{E}_1^{(2)} \end{array} \right], \tag{4.11}$$

where

$$\left[\boldsymbol{D}_1^{(2)}\right]_{rn} = \sin \frac{(2r+1)(2n+L+1)\pi}{4M}; \quad n \in [0, \frac{M-L}{2} - 1] \;\; r \in [0, M-1] \tag{4.12}$$

$$\left[\boldsymbol{E}_1^{(2)}\right]_{r(\frac{M-L}{2}-1-n)} = (-1)^r \cos \frac{(2r+1)(2n+L+1)\pi}{4M}; \quad n \in [0, \frac{M-L}{2}-1] \;\; r \in [0, M-1]. \tag{4.13}$$

We are not going to do further decomposition for $\boldsymbol{\Gamma}^{(2)}$ right away. Instead, we leave the results here and continue to check T_3.

(3). Through (4.3), T_3 can be rewritten as

$$T_3 = \sum_{n=M}^{M+L-1} \sqrt{\frac{2}{M}} \cos \left[\frac{(n-M)\pi}{2(L-1)} - \frac{1}{4} \sin \frac{(n-M)\pi}{L-1} \right] \sin \frac{(2r+1)(2n-L+1)\pi}{4M} x(n). \tag{4.14}$$

Using the same partition method as before, T_3 can be divided as

$$\begin{aligned} T_3 = (-1)^r \sqrt{\frac{2}{M}} \Bigg\{ & \sum_{n=0}^{\frac{L}{2}-1} \cos \left[\frac{n\pi}{2(L-1)} - \frac{1}{4} \sin \frac{n\pi}{L-1} \right] \cos \frac{(2r+1)(2n-L+1)\pi}{4M} x(n+M) \\ & + \sum_{n=0}^{\frac{L}{2}-1} \sin \left[\frac{n\pi}{2(L-1)} - \frac{1}{4} \sin \frac{n\pi}{L-1} \right] \cos \frac{(2r+1)(2n-L+1)\pi}{4M} x(M+L-1-n) \Bigg\}. \end{aligned} \tag{4.15}$$

Similarly, (4.15) shows that $\boldsymbol{\Gamma}^{(3)}$ can also be partitioned into

$$\boldsymbol{\Gamma}^{(3)} = \left[\begin{array}{c:c} \boldsymbol{D}_1^{(3)} & \boldsymbol{E}_1^{(3)} \end{array} \right], \tag{4.16}$$

where

$$\left[\boldsymbol{D}_1^{(3)}\right]_{rn} = (-1)^r \cos\theta_n \cos \frac{(2r+1)(2n-L+1)\pi}{4M}; \quad n \in [0, \frac{L}{2} - 1] \;\; r \in [0, M-1],$$

$$\left[\boldsymbol{E}_1^{(3)}\right]_{r(\frac{L}{2}-1-n)} = (-1)^r \sin\theta_n \cos \frac{(2r+1)(2n-L+1)\pi}{4M}; \quad n \in [0, \frac{L}{2}-1] \;\; r \in [0, M-1].$$

Post-multiplying by matrix $\boldsymbol{F}_1^T$ in (4.16), we have

$$\boldsymbol{\Gamma}^{(3)} \cdot \boldsymbol{F}_1^T = \left[\begin{array}{c:c} \boldsymbol{D}_2^{(3)} & \boldsymbol{E}_2^{(3)} \end{array} \right], \tag{4.17}$$

where

$$\left[\boldsymbol{D}_2^{(3)}\right]_{rn} = (-1)^r C_{rn}; \quad n \in [0, \frac{L}{2} - 1] \;\; r \in [0, M-1], \tag{4.18}$$

$$C_{rn} = \cos \frac{(2r+1)(2n-L+1)\pi}{4M},$$

and $\boldsymbol{E}_2^{(3)}$ is an $L \times \frac{L}{2}$ zero matrix. Combining all the partitions of $\boldsymbol{\Gamma}^{(1)}$, $\boldsymbol{\Gamma}^{(2)}$ and $\boldsymbol{\Gamma}^{(3)}$, it is observed that $\boldsymbol{\Phi}^T$ can be partitioned by using the matrix $\boldsymbol{G}$

$$\boldsymbol{G} = \begin{bmatrix} [\boldsymbol{F}_1]^T_{L\times L} & \vdots & \boldsymbol{O}_{L\times(M-L)} & \vdots & \boldsymbol{O}_L \\ \boldsymbol{O}_{(M-L)\times L} & \vdots & \boldsymbol{I}_{M-L} & \vdots & \boldsymbol{O}_{(M-L)\times L} \\ \boldsymbol{O}_L & \vdots & \boldsymbol{O}_{L\times(M-L)} & \vdots & [\boldsymbol{F}_1]^T_{L\times L} \end{bmatrix},$$

which is seen to have a specific form based on Given's rotations, identity matrix, and null matrices. The total factorization can be described by the following relation

$$\boldsymbol{\Phi}^T\boldsymbol{G} = \begin{bmatrix} \boldsymbol{O}_{M\times\frac{L}{2}} & \vdots & \boldsymbol{E}_2^{(1)} & \vdots & \boldsymbol{D}_1^{(2)} & \vdots & \boldsymbol{E}_1^{(2)} & \vdots & \boldsymbol{D}_2^{(3)} & \vdots & \boldsymbol{O}_{M\times\frac{L}{2}} \end{bmatrix}. \tag{4.19}$$

As we will show, the inner part of the new matrix $\boldsymbol{\Phi}^T\boldsymbol{G}$ is an $M \times M$ matrix with symmetry properties of sine and cosine functions and can be identified as a DST-IV transform matrix. In addition, it is easy to show that $\boldsymbol{G}$ is unitary. The elements of $\boldsymbol{E}_2^{(1)}$, $\boldsymbol{D}_1^{(2)}$, $\boldsymbol{E}_1^{(2)}$ and $\boldsymbol{D}_2^{(3)}$ in (4.9), (4.12), (4.13) and (4.18) are, for all $r \in [0, M-1]$

$$\left[\boldsymbol{E}_2^{(1)}\right]_{rn} = \sin \frac{(2r+1)(2n+1)\pi}{4M}; \quad n \in [0, \frac{L}{2} - 1] \;\; r[0, M-1], \tag{4.20}$$

$$\begin{aligned} \left[\boldsymbol{D}_1^{(2)}\right]_{rn} &= \sin \frac{(2r+1)(2n+L+1)\pi}{4M}; \quad n \in [0, \frac{M-L}{2} - 1], \\ &= \sin \frac{(2r+1)(2n'+1)\pi}{4M}; \quad n' \in [\frac{L}{2}, \frac{M}{2} - 1]; \text{where } n' = n + L/2, \end{aligned} \tag{4.21}$$

$$\begin{aligned} \left[\boldsymbol{E}_1^{(2)}\right]_{rn} &= (-1)^r \cos \frac{(2r+1)(M-2n-1)\pi}{4M}; \quad n \in [0, \frac{M-L}{2} - 1], \\ &= \sin \frac{(2r+1)(2n'+1)\pi}{4M}; \quad n' \in [\frac{M}{2}, M - \frac{L}{2} - 1]; \text{where } n' = n + M/2, \end{aligned} \tag{4.22}$$

$$\begin{aligned} \left[\boldsymbol{D}_2^{(3)}\right]_{rn} &= (-1)^r \cos \frac{(2r+1)(2n-L+1)\pi}{4M}; \quad n \in [0, \frac{L}{2} - 1], \\ &= \sin \frac{(2r+1)(2n'+1)\pi}{4M}; \quad n' \in [M - \frac{L}{2}, M-1]; \text{where } n' = n + M - L/2. \end{aligned} \tag{4.23}$$

According to the definition of DST- IV [66], one sees that this new structure is just the M-point DST-IV. Denoting the inner non-zero portions in (4.19) as $\boldsymbol{\Theta}$, we have

$$\boldsymbol{\Theta} = \begin{bmatrix} \boldsymbol{E}_2^{(1)} & \vdots & \boldsymbol{D}_1^{(2)} & \vdots & \boldsymbol{E}_1^{(2)} & \vdots & \boldsymbol{D}_2^{(3)} \end{bmatrix} = \sqrt{\frac{M}{2}}\boldsymbol{S}_M, \tag{4.24}$$

where $\boldsymbol{S}_M$ is an $M \times M$ DST-IV matrix. Since $\boldsymbol{G}$ is unitary, we can write $\boldsymbol{\Phi}^T$ in factored form as

$$\boldsymbol{\Phi}^T = \left[\boldsymbol{O}_{M\times\frac{L}{2}} \ \vdots \ \sqrt{\tfrac{M}{2}}\boldsymbol{S}_M \ \vdots \ \boldsymbol{O}_{M\times\frac{L}{2}} \right] \boldsymbol{G}^T \tag{4.25}$$

where $\boldsymbol{G}^T$ is sparse and $\boldsymbol{S}_M$ can be made sparse. The sparse factorization of DST-IV is related to that of the DCT-IV matrix $\boldsymbol{C}_M$ through

$$\boldsymbol{S}_M = \boldsymbol{\Lambda}\boldsymbol{C}_M\tilde{\boldsymbol{I}}, \tag{4.26}$$

where

$$\boldsymbol{\Lambda} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & -1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & (-1)^{M-1} \end{bmatrix} \tag{4.27}$$

$$[\boldsymbol{C}_M]_{rn} = \cos\frac{(2r+1)(2n+1)\pi}{4M}, \tag{4.28}$$

and $\tilde{\boldsymbol{I}}$ is the reverse (or anti-diagonal) identity matrix. It remains to consider the sparse factorization for DCT-IV.

4.4 Sparse Factorization for DCT-IV

In this section, we examine in detail the factorization for $\boldsymbol{C}_M$. The structure of $\boldsymbol{C}_M$ is such that after applying Given's rotations and butterfly operations, many zero elements can be introduced in $\boldsymbol{C}_M$, and the required sparse matrix factors for $\boldsymbol{C}_M$ can be obtained. We will describe the factorization of $\boldsymbol{C}_M$ in the following stages.

- Stage 1 :

According to (4.28), we have

$$[\boldsymbol{C}_M]_{rn} = \cos\frac{(2r+1)(2n+1)\pi}{4M}, \quad r,n \in [0, M-1]. \tag{4.29}$$

Note that

$$\begin{aligned} [\boldsymbol{C}_M]_{r(M-n-1)} &= \cos\frac{(2r+1)[2(M-1-n)+1]\pi}{4M} \\ &= (-1)^r \sin\frac{(2r+1)(2n+1)\pi}{4M}. \end{aligned}$$

$[C_M]_{rn}$ and $[C_M]_{r(M-n-1)}$ are cosine and sine functions of the same angle up to a sign. A proper Given's rotation operation will reduce most of these elements to zeros.

Let F_2 be a matrix of Given's rotations defined by

$$F_2 = G_M^I = \begin{bmatrix} \cos\alpha_0 & & & & \sin\alpha_0 \\ & \ddots & & \iddots & \\ & & \cos\alpha_{\frac{M}{2}-1} & \sin\alpha_{\frac{M}{2}-1} & \\ & & -\sin\alpha_{\frac{M}{2}-1} & \cos\alpha_{\frac{M}{2}-1} & \\ & \iddots & & \ddots & \\ -\sin\alpha_0 & & & & \cos\alpha_0 \end{bmatrix}, \tag{4.30}$$

where $\alpha_n = \frac{2n+1}{4M}\pi$, $n \in [0, \frac{M}{2} - 1]$, and the superscript I refers to the first kind of Given's rotation matrix. We will use similar notations for other Given's rotations and butterfly matrices in the rest of this chapter. It can be easily verified that

$$F_2^{-1} = F_2^T.$$

Therefore, we have

$$C_M = (C_M F_2^T) F_2.$$

We can partition the columns of the product $C_M \cdot F_2^T$ into two blocks of equal size so that

$$C_M F_2^T = \left[D_3^{(1)} : E_3^{(1)} \right],$$

where

$$\begin{aligned} [D_3^{(1)}]_{rn} &= \cos\frac{(2r+1)(2n+1)\pi}{4M} \cdot \cos\alpha_n + (-1)^r \sin\frac{(2r+1)(2n+1)\pi}{4M} \cdot \sin\alpha_n \\ &= \begin{cases} \cos\frac{(2r)(2n+1)\pi}{4M} & \text{for even } r \\ \cos\frac{(2r+2)(2n+1)\pi}{4M} & \text{for odd } r \end{cases}, \end{aligned} \tag{4.31}$$

and

$$\begin{aligned} [E_3^{(1)}]_{r(\frac{M}{2}-n-1)} &= -\cos\frac{(2r+1)(2n+1)\pi}{4M} \cdot \sin\alpha_n + (-1)^r \sin\frac{(2r+1)(2n+1)\pi}{4M} \cdot \cos\alpha_n \\ &= \begin{cases} \sin\frac{(2r)(2n+1)\pi}{4M} & \text{for even } r, \\ -\sin\frac{(2r+2)(2n+1)\pi}{4M} & \text{for odd } r. \end{cases} \end{aligned} \tag{4.32}$$

From (4.31) and (4.32), we have for integer i's

$$[D_3^{(1)}]_{r(\frac{M}{2}-1-n)} = \begin{cases} (-1)^i \cos\frac{(2r)(2n+1)\pi}{4M} & \text{for } r = 2i, \\ (-1)^{i+1} \cos\frac{(2r+2)(2n+1)\pi}{4M} & \text{for } r = 2i+1, \end{cases} \tag{4.33}$$

$$[\boldsymbol{E}_3^{(1)}]_{rn} = \begin{cases} (-1)^{i+1} \sin \frac{(2r)(1+2n)\pi}{4M} & \text{for } r = 2i, \\ (-1)^{i+1} \sin \frac{(2r+2)(1+2n)\pi}{4M} & \text{for } r = 2i+1 \ . \end{cases} \tag{4.34}$$

Based on (4.31), (4.32), (4.33) and (4.34), it is easily seen that :

1). $|[\boldsymbol{D}_3^{(1)}]_{rn}| = |[\boldsymbol{D}_3^{(1)}]_{r(\frac{M}{2}-n-1)}|$;

2). $|[\boldsymbol{E}_3^{(1)}]_{rn}| = |[\boldsymbol{E}_3^{(1)}]_{r(\frac{M}{2}-n-1)}|$.

These two equalities indicate that proper butterfly operations will reduce most of these elements to zeros.

- Stage 2 :

Let

$$\boldsymbol{F}_3 = \begin{bmatrix} \boldsymbol{B}^I_{\frac{M}{2}} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{B}^I_{\frac{M}{2}} \end{bmatrix},$$

where $\boldsymbol{B}^I_{\frac{M}{2}}$ is a butterfly matrix, given by $\begin{bmatrix} \boldsymbol{I}_{\frac{M}{4}} & \tilde{\boldsymbol{I}}_{\frac{M}{4}} \\ \tilde{\boldsymbol{I}}_{\frac{M}{4}} & -\boldsymbol{I}_{\frac{M}{4}} \end{bmatrix}$ and $\boldsymbol{I}_k$ and $\tilde{\boldsymbol{I}}_k$ are $k \times k$ identity and reverse identity matrices respectively. Because $\boldsymbol{F}_3^{-1} = \frac{1}{2}\boldsymbol{F}_3^T$, we have

$$\begin{bmatrix} \boldsymbol{D}_3^{(1)} & \vdots & \boldsymbol{E}_3^{(1)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{D}_3^{(1)} & \vdots & \boldsymbol{E}_3^{(1)} \end{bmatrix} \cdot \frac{1}{2}\boldsymbol{F}_3^T \boldsymbol{F}_3.$$

Let

$$\frac{1}{2}\begin{bmatrix} \boldsymbol{D}_3^{(1)} & \vdots & \boldsymbol{E}_3^{(1)} \end{bmatrix} \cdot \boldsymbol{F}_3^T = \begin{bmatrix} \boldsymbol{D}_4^{(1)} & \vdots & \boldsymbol{D}_4^{(2)} & \vdots & \boldsymbol{E}_4^{(1)} & \vdots & \boldsymbol{E}_4^{(2)} \end{bmatrix}, \tag{4.35}$$

where $\boldsymbol{D}_4^{(1)}, \boldsymbol{D}_4^{(2)}, \boldsymbol{E}_4^{(1)}$ and $\boldsymbol{E}_4^{(2)}$ are $M \times \frac{M}{4}$ matrices. It can be shown that

$$[\boldsymbol{D}_4^{(1)}]_{rn} = \begin{cases} \cos \frac{(2r)(2n+1)\pi}{4M} & \text{for } r = 4i, \\ \cos \frac{(2r+2)(2n+1)\pi}{4M} & \text{for } r = 4i+3, \\ 0 & \text{otherwise;} \end{cases}$$

$$[\boldsymbol{D}_4^{(2)}]_{r(\frac{M}{4}-n-1)} = \begin{cases} \cos \frac{(2r)(2n+1)\pi}{4M} & \text{for } r = 4i+2, \\ \cos \frac{(2r+2)(2n+1)\pi}{4M} & \text{for } r = 4i+1, \\ 0 & \text{otherwise;} \end{cases}$$

and

$$[\boldsymbol{E}_4^{(1)}]_{rn} = \begin{cases} \sin \frac{(2r)(2n+1)\pi}{4M} & \text{for } r = 4i+2, \\ -\sin \frac{(2r+2)(2n+1)\pi}{4M} & \text{for } r = 4i+1, \\ 0 & \text{otherwise;} \end{cases} \tag{4.36}$$

$$[\boldsymbol{E}_4^{(2)}]_{r(\frac{M}{4}-n-1)} = \begin{cases} -\sin\frac{(2r)(2n+1)\pi}{4M} & \text{for } r = 4i, \\ \sin\frac{(2r+2)(2n+1)\pi}{4M} & \text{for } r = 4i+3, \\ 0 & \text{otherwise.} \end{cases} \tag{4.37}$$

Furthermore we have

$$[\boldsymbol{D}_4^{(2)}]_{rn} = \begin{cases} (-1)^i \sin\frac{(2r)(2n+1)\pi}{4M} & \text{for } r = 4i+2, \\ (-1)^i \sin\frac{(2r+2)(2n+1)\pi}{4M} & \text{for } r = 4i+1, \\ 0 & \text{otherwise;} \end{cases} \tag{4.38}$$

$$[\boldsymbol{E}_4^{(2)}]_{rn} = \begin{cases} (-1)^i \sin\frac{(2r)(2n+1)\pi}{4M} & \text{for } r = 4i, \\ (-1)^i \sin\frac{(2r+2)(2n+1)\pi}{4M} & \text{for } r = 4i+3, \\ 0 & \text{otherwise.} \end{cases} \tag{4.39}$$

Based on the results in (4.36) and (4.38), the inner matrices $[\boldsymbol{D}_4^{(2)} \vdots \boldsymbol{E}_4^{(1)}]$ can be factorized again. Therefore, we can reduce more elements into zeros by using a butterfly matrix defined as :

$$\boldsymbol{B}_{\frac{M}{2}}^{II} = \begin{bmatrix} \boldsymbol{I}_{\frac{M}{8}} & \vdots & \boldsymbol{O} & \vdots & -\boldsymbol{I}_{\frac{M}{8}} & \vdots & \boldsymbol{O} \\ \cdots & \vdots & \cdots & \vdots & \cdots & \vdots & \cdots \\ \boldsymbol{O} & \vdots & \boldsymbol{I}_{\frac{M}{8}} & \vdots & \boldsymbol{O} & \vdots & \boldsymbol{I}_{\frac{M}{8}} \\ \cdots & \vdots & \cdots & \vdots & \cdots & \vdots & \cdots \\ \boldsymbol{I}_{\frac{M}{8}} & \vdots & \boldsymbol{O} & \vdots & \boldsymbol{I}_{\frac{M}{8}} & \vdots & \boldsymbol{O} \\ \cdots & \vdots & \cdots & \vdots & \cdots & \vdots & \cdots \\ \boldsymbol{O} & \vdots & \boldsymbol{I}_{\frac{M}{8}} & \vdots & \boldsymbol{O} & \vdots & -\boldsymbol{I}_{\frac{M}{8}} \end{bmatrix}. \tag{4.40}$$

Our next problem is to decompose $\boldsymbol{D}_4^{(1)}$ and $\boldsymbol{E}_4^{(2)}$. According to results in (4.37) and (4.39), $[\boldsymbol{E}_4^{(2)}]_{rn}$ and $[\boldsymbol{E}_4^{(2)}]_{r(\frac{M}{4}-n-1)}$ have the same values up to a sign : i.e.,

$$|[\boldsymbol{E}_4^{(2)}]_{rn}| = |[\boldsymbol{E}_4^{(2)}]_{r(\frac{M}{4}-n-1)}|.$$

We also observe that

$$|[\boldsymbol{D}_4^{(1)}]_{rn}| = |[\boldsymbol{D}_4^{(1)}]_{r(\frac{M}{4}-n-1)}|.$$

These structures of $\boldsymbol{D}_4^{(1)}$ and $\boldsymbol{E}_4^{(2)}$ imply that a butterfly matrix of dimension $\frac{M}{4}$ will produce more zeros. The above results lead to the third stage of the decomposition.

- Stage 3:

Let

$$F_4 = \begin{bmatrix} B^I_{\frac{M}{4}} & \vdots & O & \vdots & O \\ \cdots & \vdots & \cdots & \vdots & \cdots \\ O & \vdots & B^{II}_{\frac{M}{2}} & \vdots & O \\ \cdots & \vdots & \cdots & \vdots & \cdots \\ O & \vdots & O & \vdots & B^I_{\frac{M}{4}} \end{bmatrix} \tag{4.41}$$

be the third factor extracted in C_M. Because $F_4^{-1} = \frac{1}{2}F_4^T$, we have

$$\begin{bmatrix} D_4^{(1)} & \vdots & D_4^{(2)} & \vdots & E_4^{(1)} & \vdots & E_4^{(2)} \end{bmatrix}$$
$$= \begin{bmatrix} D_4^{(1)} & \vdots & D_4^{(2)} & \vdots & E_4^{(1)} & \vdots & E_4^{(2)} \end{bmatrix} \cdot \frac{1}{2} F_4^T F_4.$$

Let us define:

$$\frac{1}{2}\begin{bmatrix} D_4^{(1)} & \vdots & D_4^{(2)} & \vdots & E_4^{(1)} & \vdots & E_4^{(2)} \end{bmatrix} \cdot F_4^T$$
$$= \begin{bmatrix} D_5^{(1)} & \vdots & D_5^{(2)} & \vdots & D_5^{(3)} & \vdots & D_5^{(4)} & \vdots & E_5^{(1)} & \vdots & E_5^{(2)} & \vdots & E_5^{(3)} & \vdots & E_5^{(4)} \end{bmatrix}.$$

The elements of the above matrix can be described as

$$[D_5^{(1)}]_{rn} = \begin{cases} \cos\frac{(2r)(2n+1)\pi}{4M} & \text{for } r = 8i, \\ \cos\frac{(2r+2)(2n+1)\pi}{4M} & \text{for } r = 8i+7, \\ 0 & \text{otherwise;} \end{cases}$$

$$[D_5^{(2)}]_{r(\frac{M}{8}-n-1)} = \begin{cases} \cos\frac{(2r)(2n+1)\pi}{4M} & \text{for } r = 8i+4, \\ \cos\frac{(2r+2)(2n+1)\pi}{4M} & \text{for } r = 8i+3, \\ 0 & \text{otherwise;} \end{cases} \tag{4.42}$$

$$[D_5^{(3)}]_{rn} = \begin{cases} -\sin\frac{(2r)(2n+1)\pi}{4M} & \text{for } r = 8i+6, \\ \sin\frac{(2r+2)(2n+1)\pi}{4M} & \text{for } r = 8i+1, \\ 0 & \text{otherwise;} \end{cases}$$

$$[D_5^{(4)}]_{rn} = \begin{cases} \sin\frac{(2r)(2n+\frac{M}{4}+1)\pi}{4M} & \text{for } r = 8i+2, \\ -\sin\frac{(2r+2)(2n+\frac{M}{4}+1)\pi}{4M} & \text{for } r = 8i+5, \\ 0 & \text{otherwise;} \end{cases}$$

and

$$[E_5^{(1)}]_{rn} = \begin{cases} \sin\frac{(2r)(2n+1)\pi}{4M} & \text{for } r = 8i+2, \\ -\sin\frac{(2r+2)(2n+1)\pi}{4M} & \text{for } r = 8i+5, \\ 0 & \text{otherwise;} \end{cases}$$

$$[\boldsymbol{E}_5^{(2)}]_{rn} = \begin{cases} -\sin\frac{(2r)(2n+\frac{M}{4}+1)\pi}{4M} & \text{for } r = 8i+6, \\ \sin\frac{(2r+2)(2n+\frac{M}{4}+1)\pi}{4M} & \text{for } r = 8i+1, \\ 0 & \text{otherwise;} \end{cases}$$

$$[\boldsymbol{E}_5^{(3)}]_{rn} = \begin{cases} -\sin\frac{(2r)(2n+1)\pi}{4M} & \text{for } r = 8i+4, \\ \sin\frac{(2r+2)(2n+1)\pi}{4M} & \text{for } r = 8i+3, \\ 0 & \text{otherwise;} \end{cases} \tag{4.43}$$

$$[\boldsymbol{E}_5^{(4)}]_{r(\frac{M}{8}-n-1)} = \begin{cases} \sin\frac{(2r)(2n+1)\pi}{4M} & \text{for } r = 8i, \\ -\sin\frac{(2r+2)(2n+1)\pi}{4M} & \text{for } r = 8i+7, \\ 0 & \text{otherwise.} \end{cases} \tag{4.44}$$

Simplifying (4.42) and (4.44), it is obtained that

$$[\boldsymbol{D}_5^{(2)}]_{rn} = \begin{cases} (-1)^i \sin\frac{(2r)(2n+1)\pi}{4M} & \text{for } r = 8i+4, \\ (-1)^i \sin\frac{(2r+2)(2n+1)\pi}{4M} & \text{for } r = 8i+3, \\ 0 & \text{otherwise;} \end{cases} \tag{4.45}$$

$$[\boldsymbol{E}_5^{(4)}]_{rn} = \begin{cases} (-1)^{i+1} \sin\frac{(2r)(2n+1)\pi}{4M} & \text{for } r = 8i, \\ (-1)^{i+1} \sin\frac{(2r+2)(2n+1)\pi}{4M} & \text{for } r = 8i+7, \\ 0 & \text{otherwise.} \end{cases}$$

Let us examine the characteristics of $\boldsymbol{D}_5^{(1)}$, $\boldsymbol{D}_5^{(2)}$, $\boldsymbol{D}_5^{(3)}$,$\boldsymbol{D}_5^{(4)}$, $\boldsymbol{E}_5^{(1)}$,$\boldsymbol{E}_5^{(2)}$, $\boldsymbol{E}_5^{(3)}$, and $\boldsymbol{E}_5^{(4)}$. First comparing (4.43) and (4.45), it is seen that the absolute values of the elements in matrix $\boldsymbol{D}_5^{(2)}$ are equal to those of the corresponding elements in matrix $\boldsymbol{E}_5^{(3)}$. Thus we can consider these two matrices (i.e., $\boldsymbol{D}_5^{(2)}$ and $\boldsymbol{E}_5^{(3)}$) together and use $\boldsymbol{B}_{\frac{M}{4}}^I$ to factorize them. To obtain the properties of other sub-matrices, we need to look at their respective structures. We find that $\boldsymbol{D}_5^{(1)}$ and $\boldsymbol{E}_5^{(4)}$ have characteristics similar to $\boldsymbol{D}_4^{(1)}$ and $\boldsymbol{E}_4^{(2)}$. Thus, for these two sub-matrices, we can use $\boldsymbol{B}_{\frac{M}{8}}^I$ as the decomposition factor. The similarity between other sub-matrices can be summarized as follows (see Figure 4.1):

$$\begin{aligned} &(1). \quad |[\boldsymbol{D}_5^{(1)}]_{rn}| = |[\boldsymbol{D}_5^{(1)}]_{r(\frac{M}{8}-n-1)}|; \\ &(2). \quad |[\boldsymbol{E}_5^{(4)}]_{rn}| = |[\boldsymbol{E}_5^{(4)}]_{r(\frac{M}{8}-n-1)}|; \\ &(3). \quad arcos|[\boldsymbol{D}_5^{(4)}]_{rn}| = arcsin|[\boldsymbol{E}_5^{(1)}]_{r(\frac{M}{8}-n-1)}|; \\ &(4). \quad arcsin|[\boldsymbol{D}_5^{(3)}]_{rn}| = arcos|[\boldsymbol{E}_5^{(2)}]_{r(\frac{M}{8}-n-1)}|; \\ &(5). \quad |[\boldsymbol{D}_5^{(2)}]_{rn}| = |[\boldsymbol{E}_5^{(3)}]_{rn}|; \end{aligned} \tag{4.46}$$

Figure 4.1 shows the grouping of the matrices diagrammatically based on the above

$$[\underbrace{\boldsymbol{D}_5^{(1)}}_{(1)} : \boldsymbol{D}_5^{(2)} : \boldsymbol{D}_5^{(3)} : \underbrace{\boldsymbol{D}_5^{(4)} : \boldsymbol{E}_5^{(1)}}_{(3)} : \boldsymbol{E}_5^{(2)} : \boldsymbol{E}_5^{(3)} : \underbrace{\boldsymbol{E}_5^{(4)}}_{(2)}]$$
$$\underbrace{\qquad\qquad\qquad\qquad}_{(4)}$$
$$\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad\qquad}_{(5)}$$

Figure 4.1: Layer notations

properties. We note that among these groups, (1), (2) and (5) require butterfly matrices for further factorization, and (3) and (4) require Given's rotations. The above results lead to the fourth stage.

- Stage 4:

In this stage, we use $\boldsymbol{B}^I_{\frac{M}{8}}$,$\boldsymbol{B}^{II}_{\frac{M}{4}}$ and $\boldsymbol{G}^{II}_{\frac{M}{2}}$ to decompose

$$\left[\begin{array}{ccccccccc} \boldsymbol{D}_5^{(1)} & \vdots & \boldsymbol{D}_5^{(2)} & \vdots & \boldsymbol{D}_5^{(3)} & \vdots & \boldsymbol{D}_5^{(4)} & \vdots & \boldsymbol{E}_5^{(1)} & \vdots & \boldsymbol{E}_5^{(2)} & \vdots & \boldsymbol{E}_5^{(3)} & \vdots & \boldsymbol{E}_5^{(4)} \end{array}\right], \tag{4.47}$$

i.e.,

$$\boldsymbol{F}_5 = \left[\begin{array}{ccccccccc} \boldsymbol{B}^I_{\frac{M}{8}} & \vdots & \boldsymbol{O} & \vdots & \boldsymbol{O} & \vdots & \boldsymbol{O} & \vdots & \boldsymbol{O} \\ \cdots & \vdots & \cdots & \vdots & \cdots & \vdots & \cdots & \vdots & \cdots \\ \boldsymbol{O} & \vdots & \boldsymbol{I}_{\frac{M}{8}} & \vdots & \boldsymbol{O} & \vdots & -\overline{\boldsymbol{I}}_{\frac{M}{8}} & \vdots & \boldsymbol{O} \\ \cdots & \vdots & \cdots & \vdots & \cdots & \vdots & \cdots & \vdots & \cdots \\ \boldsymbol{O} & \vdots & \boldsymbol{O} & \vdots & \boldsymbol{G}^{II}_{\frac{M}{2}} & \vdots & \boldsymbol{O} & \vdots & \boldsymbol{O} \\ \cdots & \vdots & \cdots & \vdots & \cdots & \vdots & \cdots & \vdots & \cdots \\ \boldsymbol{O} & \vdots & \boldsymbol{I}_{\frac{M}{8}} & \vdots & \boldsymbol{O} & \vdots & \overline{\boldsymbol{I}}_{\frac{M}{8}} & \vdots & \boldsymbol{O} \\ \cdots & \vdots & \cdots & \vdots & \cdots & \vdots & \cdots & \vdots & \cdots \\ \boldsymbol{O} & \vdots & \boldsymbol{O} & \vdots & \boldsymbol{O} & \vdots & \boldsymbol{O} & \vdots & \boldsymbol{B}^I_{\frac{M}{8}} \end{array}\right]. \tag{4.48}$$

Note that the placement of the individual blocks in $\boldsymbol{F}_5$ reflects the grouping of the partitions in Figure 4.1 with

$$\overline{\boldsymbol{I}}_{\frac{M}{8}} = \left[\begin{array}{ccc} -\boldsymbol{I}_{\frac{M}{16}} & \vdots & \boldsymbol{O} \\ \cdots & \vdots & \cdots \\ \boldsymbol{O} & \vdots & \boldsymbol{I}_{\frac{M}{16}} \end{array}\right].$$

We note that the matrix blocks $I_{\frac{M}{8}}$ and $-\overline{I}_{\frac{M}{8}}$ around the matrix $G^{II}_{\frac{M}{2}}$ make up the butterfly. We refer to this butterfly as an "inlay" block in the factor matrix F_5; i.e.,

$$B^{II}_{\frac{M}{4}} = \begin{bmatrix} I_{\frac{M}{8}} & \vdots & -\overline{I}_{\frac{M}{8}} \\ \cdots & \vdots & \cdots \\ I_{\frac{M}{8}} & \vdots & \overline{I}_{\frac{M}{8}} \end{bmatrix}.$$

In general, the "inlay" block is defined as follows:

Definition 4.1 *If a block matrix H can be partitioned as*

$$H = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix},$$

where H_{11}, H_{12}, H_{21}, and H_{22} all are matrices, the "inlay" block H is then defined as

$$\begin{bmatrix} H_{11} & O & H_{12} \\ O & D & O \\ H_{21} & O & H_{22} \end{bmatrix},$$

with D being an arbitrary matrix.

It can be noticed that $B^{II}_{\frac{M}{4}}$ is different from $B^{II}_{\frac{M}{2}}$ in (4.40): i.e., a minus sign is exchanged in the second column of $B^{II}_{\frac{M}{4}}$. Later, we will explain when this minus sign will be exchanged. Let $G^{II}_{\frac{M}{2}}$ be defined as

$$G^{II}_{\frac{M}{2}} = \begin{bmatrix} \cos\alpha_0 & & & & & & & & & & & & & & \sin\alpha_0 \\ & \ddots & & & & & & & & & & & & \iddots & \\ & & \cos\alpha_{\frac{M}{8}-1} & & & & & & & & & & \sin\alpha_{\frac{M}{8}-1} & & \\ & & & \cos\alpha_{\frac{M}{8}-1} & & & & & & & & -\sin\alpha_{\frac{M}{8}-1} & & & \\ & & & & \ddots & & & & & \iddots & & & & & \\ & & & & & \cos\alpha_0 & & & -\sin\alpha_0 & & & & & & \\ & & & & & & \sin\alpha_0 & & \cos\alpha_0 & & & & & & \\ & & & & & \iddots & & & & \ddots & & & & & \\ & & & \sin\alpha_{\frac{M}{8}-1} & & & & & & & & \cos\alpha_{\frac{M}{8}-1} & & & \\ & & -\sin\alpha_{\frac{M}{8}-1} & & & & & & & & & & \cos\alpha_{\frac{M}{8}-1} & & \\ & \iddots & & & & & & & & & & & & \ddots & \\ -\sin\alpha_0 & & & & & & & & & & & & & & \cos\alpha_0 \end{bmatrix}, \tag{4.49}$$

where $\alpha_n = \frac{4(\frac{M}{2}-1-2n)\pi}{4M}$. Now the factor matrix F_5 can be considered to be made up of three layers. The outermost layer is made up of the two $B^{I}_{\frac{M}{8}}$, the middle layer is made

up of the butterfly blocks around $G^{II}_{\frac{M}{2}}$ (equivalent in form to $B^{II}_{\frac{M}{4}}$) and the innermost layer represents the Given's rotations. The result of extracting the factor F_5 from the matrix (4.47) (i.e., post-multiplying the matrix in (4.47) by F_5) can be described in layers as follows:

1. The outermost layer: after multiplying $[B^{I}_{\frac{M}{8}}]^T$, this layer is divided into two sub-layers. These newly generated sub-layers can be further factorized by $B^{I}_{\frac{M}{16}}$ and $B^{II}_{\frac{M}{8}}$, i.e. $B^{I}_{\frac{M}{2^4}}$ and $B^{II}_{\frac{M}{2^{4-1}}}$. Note: the new layers are similar to the previous layers

$$\underbrace{[D_5^{(1)}:\underbrace{D_5^{(2)}:\ldots:E_5^{(3)}}_{middle layer}:E_5^{(4)}]}_{outermost layer}. \tag{4.50}$$

2. The middle layer: when using $[B^{II}_{\frac{M}{4}}]^T$, this layer (corresponding to $[D_5^{(2)}:\ldots:E_5^{(3)}]$) will be factorized into the structure similar to $[\ D_5^{(3)}\ :\ D_5^{(4)}\ :\ E_5^{(1)}\ :\ E_5^{(2)}\ ]$ with the number of columns being reduced by half. Thus, the middle layer can be further factorized by $G^{II}_{\frac{M}{4}}$. It can be seen that for subsequent steps, the two outermost layers can be factorized by $B^{I}_{\frac{M}{2^{m-1}}}$ (for $m > 1$) and $B^{II}_{\frac{M}{2^{m-2}}}$ (for $m > 2$), respectively, where m indicates the stage of factorizations.

$$B^{II}_{\frac{M}{2^{m-2}}} = \begin{bmatrix} I_{\frac{M}{2^{m-1}}} & \vdots & (-1)^{m-1}\overline{I}_{\frac{M}{2^{m-1}}} \\ \cdots & \vdots & \cdots \\ I_{\frac{M}{2^{m-1}}} & \vdots & (-1)^{m}\overline{I}_{\frac{M}{2^{m-1}}} \end{bmatrix}.$$

3. The innermost block:

$$\cdot\left[\ D_5^{(3)}\ :\ D_5^{(4)}\ :\ E_5^{(1)}\ :\ E_5^{(2)}\ \right].$$

Let :

$$\left[\ D_5^{(3)}\ :\ D_5^{(4)}\ :\ E_5^{(1)}\ :\ E_5^{(2)}\ \right]\cdot[G^{II}_{\frac{M}{4}}]^T = \left[\ D_6^{(1)}\ :\ D_6^{(2)}\ :\ E_6^{(1)}\ :\ E_6^{(2)}\ \right],$$

where

$$[D_6^{(1)}]_{rn} = \begin{cases} \cos\frac{(2r+4)(2n+1)\pi}{4M} & \text{for } r = 8i+6, \\ \cos\frac{(2r-2)(2n+1)\pi}{4M} & \text{for } r = 8i+1, \\ 0 & \text{otherwise;} \end{cases}$$

$$[\boldsymbol{D}_6^{(2)}]_{rn} = \begin{cases} (-1)^{i+1} \sin \frac{(2r-4)(2n+1)\pi}{4M} & \text{for } r = 8i+2, \\ (-1)^{i} \sin \frac{(2r+6)(2n+1)\pi}{4M} & \text{for } r = 8i+5, \\ 0 & \text{otherwise;} \end{cases}$$

$$[\boldsymbol{E}_6^{(1)}]_{r(\frac{M}{8}-n-1)} = \begin{cases} (-1)^{i} \cos \frac{(2r-4)(2n+1)\pi}{4M} & \text{for } r = 8i+2, \\ (-1)^{i+1} \cos \frac{(2r+6)(2n+1)\pi}{4M} & \text{for } r = 8i+5, \\ 0 & \text{otherwise;} \end{cases} \tag{4.51}$$

$$[\boldsymbol{E}_6^{(2)}]_{r(\frac{M}{8}-n-1)} = \begin{cases} \sin \frac{(2r+4)(2n+1)\pi}{4M} & \text{for } r = 8i+6, \\ -\sin \frac{(2r-2)(2n+1)\pi}{4M} & \text{for } r = 8i+1, \\ 0 & \text{otherwise.} \end{cases} \tag{4.52}$$

Simplifying $[\boldsymbol{E}_6^{(1)}]_{rn}$ and $[\boldsymbol{E}_6^{(2)}]_{rn}$ in (4.51) and (4.52), we have

$$[\boldsymbol{E}_6^{(1)}]_{rn} = \begin{cases} \cos \frac{(2r-4)(2n+1)\pi}{4M} & \text{for } r = 8i+2, \\ \cos \frac{(2r+6)(2n+1)\pi}{4M} & \text{for } r = 8i+5, \\ 0 & \text{otherwise;} \end{cases}$$

$$[\boldsymbol{E}_6^{(2)}]_{rn} = \begin{cases} (-1)^{i+1} \sin \frac{(2r+4)(2n+1)\pi}{4M} & \text{for } r = 8i+6, \\ (-1)^{i} \sin \frac{(2r-2)(2n+1)\pi}{4M} & \text{for } r = 8i+1, \\ 0 & \text{otherwise.} \end{cases}$$

Similarly to the previous decomposition, by comparing the structures in $[\boldsymbol{D}_6^{(1)}]_{rn}$ and $[\boldsymbol{D}_6^{(1)}]_{r(\frac{M}{8}-n-1)}$; $[\boldsymbol{D}_6^{(2)}]_{rn}$ and $[\boldsymbol{D}_6^{(2)}]_{r(\frac{M}{8}-n-1)}$; $[\boldsymbol{E}_6^{(1)}]_{rn}$ and $[\boldsymbol{E}_6^{(1)}]_{r(\frac{M}{8}-n-1)}$, and $[\boldsymbol{E}_6^{(2)}]_{rn}$ and $[\boldsymbol{E}_6^{(2)}]_{r(\frac{M}{8}-n-1)}$, we can use $\boldsymbol{B}_{\frac{M}{8}}^{I}$ to do further decomposition: i.e., let

$$\left[\boldsymbol{D}_7^{(1)} \vdots \boldsymbol{D}_7^{(2)} \vdots \boldsymbol{D}_7^{(3)} \vdots \boldsymbol{D}_7^{(4)} \vdots \boldsymbol{E}_7^{(1)} \vdots \boldsymbol{E}_7^{(2)} \vdots \boldsymbol{E}_7^{(3)} \vdots \boldsymbol{E}_7^{(4)} \right]$$
$$= \left[\boldsymbol{D}_6^{(1)} \vdots \boldsymbol{D}_6^{(2)} \vdots \boldsymbol{E}_6^{(1)} \vdots \boldsymbol{E}_6^{(2)} \right] \cdot [\boldsymbol{B}_{\frac{M}{2}}^{III}]^T,$$

where

$$\boldsymbol{B}_{\frac{M}{2}}^{III} = \begin{bmatrix} \boldsymbol{B}_{\frac{M}{8}}^{I} & \vdots & \boldsymbol{O} & \vdots & \boldsymbol{O} & \vdots & \boldsymbol{O} \\ \cdots & \vdots & \cdots & \vdots & \cdots & \vdots & \cdots \\ \boldsymbol{O} & \vdots & \boldsymbol{B}_{\frac{M}{8}}^{I} & \vdots & \boldsymbol{O} & \vdots & \boldsymbol{O} \\ \cdots & \vdots & \cdots & \vdots & \cdots & \vdots & \cdots \\ \boldsymbol{O} & \vdots & \boldsymbol{O} & \vdots & \boldsymbol{B}_{\frac{M}{8}}^{I} & \vdots & \boldsymbol{O} \\ \cdots & \vdots & \cdots & \vdots & \cdots & \vdots & \cdots \\ \boldsymbol{O} & \vdots & \boldsymbol{O} & \vdots & \boldsymbol{O} & \vdots & \boldsymbol{B}_{\frac{M}{8}}^{I} \end{bmatrix}. \tag{4.53}$$

- Stage 5:

Now we get into the fifth stage $\boldsymbol{F}_6$:

$$\boldsymbol{F}_6 = \begin{bmatrix} \boldsymbol{B}^I_{\frac{M}{16}} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \boldsymbol{O} & \boldsymbol{I}_{\frac{M}{16}} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \overline{\boldsymbol{I}}_{\frac{M}{16}} & \boldsymbol{O} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{H}^I_{\frac{M}{8}} & \boldsymbol{O} & \boldsymbol{H}^{II}_{\frac{M}{8}} & \boldsymbol{O} & \boldsymbol{O} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{B}^{III}_{\frac{M}{2}} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{H}^{II}_{\frac{M}{8}} & \boldsymbol{O} & \boldsymbol{H}^I_{\frac{M}{8}} & \boldsymbol{O} & \boldsymbol{O} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \boldsymbol{O} & \boldsymbol{I}_{\frac{M}{16}} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & -\overline{\boldsymbol{I}}_{\frac{M}{16}} & \boldsymbol{O} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{B}^I_{\frac{M}{16}} \end{bmatrix}, \tag{4.54}$$

where

$$\boldsymbol{H}^I_{\frac{M}{8}} = \begin{bmatrix} \cos\alpha_0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 & 0 & 0 \\ 0 & 0 & \cos\alpha_{\frac{M}{16}-1} & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos\alpha_{\frac{M}{16}-1} & 0 & 0 \\ 0 & 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & \cos\alpha_0 \end{bmatrix},$$

$$\boldsymbol{H}^{II}_{\frac{M}{8}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \sin\alpha_0 \\ 0 & 0 & 0 & 0 & \iddots & 0 \\ 0 & 0 & 0 & \sin\alpha_{\frac{M}{16}-1} & 0 & 0 \\ 0 & 0 & -\sin\alpha_{\frac{M}{16}-1} & 0 & 0 & 0 \\ 0 & \iddots & 0 & 0 & 0 & 0 \\ \sin\alpha_0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Let

$$G_{\frac{M}{4}}^{II} = \begin{bmatrix} H_{\frac{M}{4}}^{I} & \vdots & O & \vdots & H_{\frac{M}{4}}^{II} \\ \cdots & \vdots & \cdots & \vdots & \cdots \\ O & \vdots & T & \vdots & O \\ \cdots & \vdots & \cdots & \vdots & \cdots \\ -H_{\frac{M}{4}}^{II} & \vdots & O & \vdots & H_{\frac{M}{4}}^{I} \end{bmatrix}$$

denote the centre block in F_6 with $T = B_{\frac{M}{2}}^{III}$. Now we may summarize the general steps of factorization by noting how a particular factor chosen at one stage determines the form or forms of the factors in the next stages. Figure 4.2 illustrates the general principles involved. In this diagram, we note that the procedure is recursive and the recursive structures occur in "layers". Based on the above results and observations, we develop a general procedure of stage-by-stage factorization as follows:

Procedure of Stage-by-stage factorization:

(a). If a rotation matrix G_M^I is a block or an "inlay" block in the current factor matrix F_m (the "inlay" block is defined in Definition (4.1)), then in the next factor matrix F_{m+1}, two $B_{\frac{M}{2}}^I$ blocks will appear at the diagonal elements of the new matrix, and be located at the same location as G_M^I is located in F_m: i.e.,

$$\begin{array}{cc} \text{the } m\text{-th stage} & \text{the } (m+1)\text{-th stage} \\ G_M^I \quad \longrightarrow & \begin{bmatrix} B_{\frac{M}{2}}^I & \vdots & O \\ \cdots & \cdots & \cdots \\ O & \vdots & B_{\frac{M}{2}}^I \end{bmatrix}. \end{array}$$

(b). If in F_m two diagonal blocks are located as

$$\begin{bmatrix} B_t^I & \vdots & O \\ \cdots & \cdots & \cdots \\ O & \vdots & B_t^I \end{bmatrix},$$

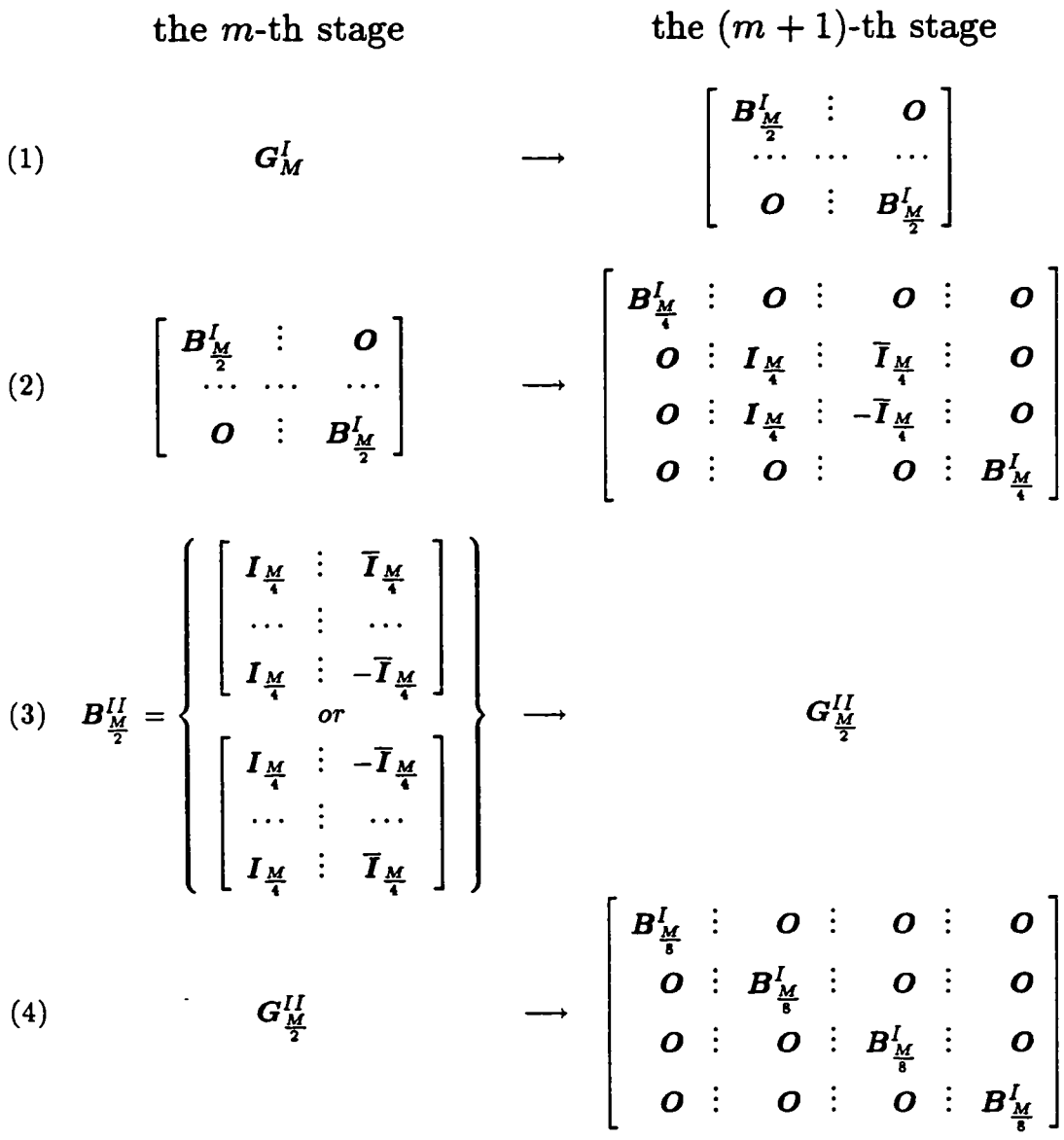


Figure 4.2: Recursive structures in layers

with t represents the dimension of the square matrix B_t^I, then this matrix can be factorized as

$$\begin{bmatrix} B_{\frac{t}{2}}^I & \vdots & O & \vdots & O & \vdots & O \\ O & \vdots & I_{\frac{t}{2}} & \vdots & \overline{I}_{\frac{t}{2}} & \vdots & O \\ O & \vdots & I_{\frac{t}{2}} & \vdots & -\overline{I}_{\frac{t}{2}} & \vdots & O \\ O & \vdots & O & \vdots & O & \vdots & B_{\frac{t}{2}}^I \end{bmatrix} \tag{4.55}$$

in F_{m+1}. Here, $I_{\frac{t}{2}}$ is a $\frac{t}{2} \times \frac{t}{2}$ identity matrix, and $\overline{I}_{\frac{t}{2}}$ is defined as

$$\overline{I}_{\frac{t}{2}} = \begin{bmatrix} -I_{\frac{t}{4}} & \vdots & O \\ \cdots & \vdots & \cdots \\ O & \vdots & I_{\frac{t}{4}} \end{bmatrix}.$$

More generally, the block matrix

$$\begin{bmatrix} B_{\frac{M}{2}}^I & \vdots & O & \vdots & O \\ O & \vdots & T_t & \vdots & O \\ O & \vdots & O & \vdots & B_{\frac{M}{2}}^I \end{bmatrix}. \tag{4.56}$$

can be decomposed into the following matrix in the subsequent stage

$$\begin{bmatrix} B_{\frac{M}{2^2}}^I & \vdots & O & \vdots & O & \vdots & O & \vdots & O \\ O & \vdots & I_{\frac{M}{2^2}} & \vdots & O & \vdots & -\overline{I}_{\frac{M}{2^2}} & \vdots & O \\ O & \vdots & O & \vdots & T_t & \vdots & O & \vdots & O \\ O & \vdots & I_{\frac{M}{2^2}} & \vdots & O & \vdots & \overline{I}_{\frac{M}{2^2}} & \vdots & O \\ O & \vdots & O & \vdots & O & \vdots & O & \vdots & B_{\frac{M}{2^2}}^I \end{bmatrix}. \tag{4.57}$$

where T_t represents a $t \times t$ inner layer matrix. This process continues until we reach B_1^I. The placement of the negative sign for the I and $\overline{I}$ matrices in (4.55) and (4.57) depends on the stage of the factorization, or $B_{\frac{M}{2^k}}^I$. We may denote the matrix $B_{\frac{M}{2^k}}^I$ as B_e^{II} for even number k, and B_o^{II} for odd number k. Then the partitions of $B_{\frac{M}{2^k}}^I$ are

shown as follows:

$$\boldsymbol{B}_o^{II} = \begin{bmatrix} \boldsymbol{I} & \vdots & \overline{\boldsymbol{I}} \\ \cdots & \vdots & \cdots \\ \boldsymbol{O} & \vdots & \boldsymbol{O} \\ \cdots & \vdots & \cdots \\ \boldsymbol{I} & \vdots & -\overline{\boldsymbol{I}} \end{bmatrix},$$

and

$$\boldsymbol{B}_e^{II} = \begin{bmatrix} \boldsymbol{I} & \vdots & -\overline{\boldsymbol{I}} \\ \cdots & \vdots & \cdots \\ \boldsymbol{O} & \vdots & \boldsymbol{O} \\ \cdots & \vdots & \cdots \\ \boldsymbol{I} & \vdots & \overline{\boldsymbol{I}} \end{bmatrix}.$$

(c). Both of the blocks $\boldsymbol{B}_o^{II}$ and $\boldsymbol{B}_e^{II}$, whether "inlay" or not, will lead to the block or "inlay" block involving the matrices $\boldsymbol{H}^I$ and $\boldsymbol{H}^{II}$ in the next factor matrix: i.e.,

the m-th stage the $(m+1)$-th stage

$$\left.\begin{array}{l} \boldsymbol{B}_o^{II} \rightarrow \begin{bmatrix} \boldsymbol{I} & \vdots & \boldsymbol{O} & \vdots & \overline{\boldsymbol{I}} \\ \boldsymbol{O} & \vdots & \mathcal{T}_t & \vdots & \boldsymbol{O} \\ \boldsymbol{I} & \vdots & \boldsymbol{O} & \vdots & -\overline{\boldsymbol{I}} \end{bmatrix} \\ \text{or} \quad \boldsymbol{B}_e^{II} \rightarrow \begin{bmatrix} \boldsymbol{I} & \vdots & \boldsymbol{O} & \vdots & -\overline{\boldsymbol{I}} \\ \boldsymbol{O} & \vdots & \mathcal{T}_t & \vdots & \boldsymbol{O} \\ \boldsymbol{I} & \vdots & \boldsymbol{O} & \vdots & \overline{\boldsymbol{I}} \end{bmatrix} \end{array}\right\} \longrightarrow \begin{bmatrix} \boldsymbol{H}^I & \vdots & \boldsymbol{O} & \vdots & \boldsymbol{H}^{II} \\ \boldsymbol{O} & \vdots & \mathcal{T}_t & \vdots & \boldsymbol{O} \\ -\boldsymbol{H}^{II} & \vdots & \boldsymbol{O} & \vdots & \boldsymbol{H}^I \end{bmatrix}.$$

The matrix made up of block of $\boldsymbol{H}^I$ and $\boldsymbol{H}^{II}$ as shown above is defined as $\boldsymbol{G}^{II}$, either as a block or as "inlay" block.

(d). If $\boldsymbol{G}_M^{II}$ is present in $\boldsymbol{F}_m$ as a block or "inlay" block, then $\boldsymbol{F}_{m+1}$ contains four $\boldsymbol{B}_{\frac{M}{4}}^I$ in its diagonal elements: i.e.,

the m-th stage the $(m+1)$-th stage

$$\begin{bmatrix} \boldsymbol{H}^I_{\frac{M}{2}} & \vdots & \boldsymbol{O} & \vdots & \boldsymbol{H}^{II}_{\frac{M}{2}} \\ \boldsymbol{O} & \vdots & \mathcal{T}_i & \vdots & \boldsymbol{O} \\ -\boldsymbol{H}^{II}_{\frac{M}{2}} & \vdots & \boldsymbol{O} & \vdots & \boldsymbol{H}^I_{\frac{M}{2}} \end{bmatrix} \longrightarrow \begin{bmatrix} \boldsymbol{B}^I_{\frac{M}{4}} & \vdots & \boldsymbol{O} & \vdots & \boldsymbol{O} & \vdots & \boldsymbol{O} & \vdots & \boldsymbol{O} \\ \boldsymbol{O} & \vdots & \boldsymbol{B}^I_{\frac{M}{4}} & \vdots & \boldsymbol{O} & \vdots & \boldsymbol{O} & \vdots & \boldsymbol{O} \\ \boldsymbol{O} & \vdots & \boldsymbol{O} & \vdots & \mathcal{T}_i & \vdots & \boldsymbol{O} & \vdots & \boldsymbol{O} \\ \boldsymbol{O} & \vdots & \boldsymbol{O} & \vdots & \boldsymbol{O} & \vdots & \boldsymbol{B}^I_{\frac{M}{4}} & \vdots & \boldsymbol{O} \\ \boldsymbol{O} & \vdots & \boldsymbol{O} & \vdots & \boldsymbol{O} & \vdots & \boldsymbol{O} & \vdots & \boldsymbol{B}^I_{\frac{M}{4}} \end{bmatrix}$$

(e). Now the factor matrix can be presented in layers, starting from the outermost layer. Steps (b), (c) and (d) are repeated until the size of the smallest blocks $\boldsymbol{B}^I$ and $\boldsymbol{B}^{II}$ is 1×1 and that of corresponding $\boldsymbol{G}^{II}$ is 4×4.

4.5 Factorization of DLS when $M = 4$ and $L = 2$

In this section, we show an example of factorization when $M = 4, L = 2$. We use the approach presented in the previous section to factorize DLS stage-by-stage.

For the case of $M = 4$ and $L = 2$, $\boldsymbol{\Phi}$ is written as

$$\boldsymbol{\Phi} = \begin{bmatrix} 0.0000 & 0.1379 & 0.3928 & 0.5879 & 0.6935 & 0.0000 \\ 0.0000 & 0.3928 & 0.6935 & 0.1379 & -0.5879 & 0.0000 \\ 0.0000 & 0.5879 & 0.1379 & -0.6935 & 0.3928 & 0.0000 \\ 0.0000 & 0.6935 & -0.5879 & 0.3928 & -0.1379 & 0.0000 \end{bmatrix}.$$

The first layer sparse matrix F_1 consists of Given's rotations and an identity matrix. By observing the Given's rotation in F_1, we note that F_1 is a 6×6 identity matrix which can be ignored. The 4×4 non-zero matrix inside $\boldsymbol{\Phi}$ is a DST-IV matrix. Let this matrix be denoted as $\boldsymbol{S}_4$, which can be transferred into DCT-IV matrix (denoted as $\boldsymbol{C}_4$) by the following relation:

$$\boldsymbol{C}_4 = \boldsymbol{\Lambda}\boldsymbol{S}_4\tilde{\boldsymbol{I}},$$

where $\boldsymbol{\Lambda}$ is defined by (4.27) and $\tilde{\boldsymbol{I}}$ is 4×4 reverse identity matrix. Then, the solution

of C_4 is written by

$$C_4 = \begin{bmatrix} 0.6935 & 0.5879 & 0.3928 & 0.1379 \\ 0.5879 & -0.1379 & -0.6935 & -0.3928 \\ 0.3928 & -0.6935 & 0.1379 & 0.5879 \\ 0.1379 & -0.3928 & 0.5879 & -0.6935 \end{bmatrix}.$$

The sparse matrix F_2 for the second layer factorization consists of Given's rotations, given by

$$F_2 = \begin{bmatrix} 0.9808 & 0.0000 & 0.0000 & 0.1951 \\ 0.0000 & 0.8315 & 0.5556 & 0.0000 \\ 0.0000 & -0.5556 & 0.8315 & 0.0000 \\ -0.1951 & 0.0000 & 0.0000 & 0.9808 \end{bmatrix}.$$

If we extract the factor F_2 from C_4, we have

$$C_4 F_2^T = \begin{bmatrix} 0.7071 & 0.7071 & 0.0000 & 0.0000 \\ 0.5000 & -0.5000 & -0.5000 & -0.5000 \\ 0.5000 & -0.5000 & 0.5000 & 0.5000 \\ 0.0000 & 0.0000 & 0.7071 & -0.7071 \end{bmatrix}.$$

Now, it is more clearly seen that the above matrix can be factorized by two butterfly matrices, given by

$$F_3 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix}.$$

Therefore, extracting F_3 , there remains

$$C_4 F_2^T F_3^T = \begin{bmatrix} 1.4142 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 1.0000 & -1.0000 & 0.0000 \\ 0.0000 & 1.0000 & 1.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 1.4142 \end{bmatrix}.$$

Now, we have one more stage to factorization by a sparse matrix $\boldsymbol{F}_4$, which is given by

$$\boldsymbol{F}_4 = \begin{bmatrix} \sqrt{2} & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & \sqrt{2} \end{bmatrix}.$$

Therefore, extracting F_4 , we have the final result

$$\frac{1}{2}\boldsymbol{C}_4\boldsymbol{F}_2^T\boldsymbol{F}_3^T\boldsymbol{F}_4^T = \begin{bmatrix} 1.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 1.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 1.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 1.0000 \end{bmatrix}.$$

Then, we have

$$\boldsymbol{C}_4 = 2\boldsymbol{F}_4\boldsymbol{F}_3\boldsymbol{F}_2.$$

In the above example, we first transfer the inner non-zero matrix of $\boldsymbol{\Phi}$ into a DCT-IV matrix, and then extract a sequence of sparse matrix factors from this DCT-IV matrix until the remainder is diagonal. This procedure gives us the sparse factorization.

4.6 Determination of Parameters

In the last two sections, we discussed the relation between DLS transform and DST-IV/DCT-IV, and presented a way to decompose the DLS transform for general M and L. In this section, we will discuss some relevant parameters in this matrix factorization. These parameters include the coefficients of the Given's rotations, the number of stages in the factorization, and the computational complexity.

4.6.1 Coefficients of Given Rotation

(1). $\boldsymbol{G}_M^I$

In (4.30), we have given the angles in $\boldsymbol{G}_M^I$ as :

$$\alpha_n = \frac{(2n+1)\pi}{4M}.$$

(2). $\boldsymbol{G}_t^{II}$

In general G_t^{II} is a $t \times t$ matrix. According to the rules of the previous section, we have $t = 2^k$ where $k = 1, 2, \cdots, (\log_2 M - 1)$ and $M \geq 2^3$. The general G_t^{II} can be represented as:

$$G_t^{II} = \begin{bmatrix} \cos\alpha_0 & & & & & & & & & & & & \sin\alpha_0 \\ & \ddots & & & & & & & & & & \iddots & \\ & & \cos\alpha_{\frac{t}{4}-1} & & & & & & & & \iddots & & \\ & & & \cos\alpha_{\frac{t}{4}-1} & & & & & & -\sin\alpha_{\frac{t}{4}-1} & & & \\ & & & & \ddots & & & & \iddots & & & & \\ & & & & & \cos\alpha_0 & & -\sin\alpha_0 & & & & & \\ & & & & & & \sin\alpha_0 & & \cos\alpha_0 & & & & \\ & & & & & \sin\alpha_{\frac{t}{4}-1} & & & & \ddots & & & \\ & & & & \iddots & & & & & & \cos\alpha_{\frac{t}{4}-1} & & \\ & & & -\sin\alpha_{\frac{t}{4}-1} & & & & & & & & \ddots & \\ & & \iddots & & & & & & & & & & \\ -\sin\alpha_0 & \iddots & & & & & & & & & & & \cos\alpha_0 \end{bmatrix} \cdot q$$

where $q = \begin{cases} 1 & ;t > 4 \\ \sqrt{2} & ;t = 4 \end{cases}$ and $\alpha_n = \frac{2^{\ell+2}[t-(2n+1)]\pi}{4M}$; $n = 0, 1, \cdots, \frac{t}{4} - 1$; $\ell = \log_2 \frac{M}{2t}$.

4.6.2 Number of Stages

The algorithm described in the previous section has a regular pattern of repetition after the first two layer factorizations (i.e., F_1 and F_2). The repetition occurs every time when a matrix of the type G_t^{II} appears. G_t^{II} occurs two stages after each pair of two B_n^I (where, $n \geq 4$) is generated. The first two layer factorizations are fixed for $M \geq 4$. After that, we start to count the number of stages of factorization when a matrix B_n^I first appears in sparse matrices. Let $[B_n^I]_1$ represents the first appearance of B_n^I matrix. For example, when $M = 4$, $[B_n^I]_1$ occurs at $n = 2$, and $[B_2^I]_1$ can only be further factorized into one more stage. It is known that before $[B_2^I]_1$, we have 2 stages, which is fixed for any M and L. Thus for $M = 4$, we have a total of 4 stages in sparse factorization. For $M = 8$, B_n^I first appears at $n = 4$, and $[B_4^I]_1$ can be further factorized by two more stages. We then obtain the first matrix of the form G_4^{II} denoted as $[G_4^{II}]_1$. Because $[G_4^{II}]_1$ reaches minimum dimension for such a type of Given's rotations, the factorization is finished. Therefore, combining all stages, we have a total of 5 stages for $M = 8$. Similarly, when $M = 16$, we have $[B_8^I]_1$, $[G_8^{II}]_1$, and $[B_2^I]_2$. From $[B_8^I]_1$ to $[G_8^{II}]_1$, we have 3 stages; from $[G_8^{II}]_1$ to $[B_2^I]_2$, 1 stage. Also, $[B_2^I]_2$ can be further factorized as in the case of $n = 2$ ($M = 4$). Thus, the total number of stages for $M = 16$ is 7. Based on the above

observations, we can generalize the number of stages as follows.

Let i be the number of repetition from $\boldsymbol{B}_n^I$ to $\boldsymbol{G}_n^{II}$. The total number of stages P, is then described as

$$P = \begin{cases} 2+(3i+2), & \text{for } M = 2^{2j} \text{ and } \frac{M}{4^{i+1}} = 1 \\ 3+(3i+2), & \text{for } M = 2^{2j+1} \text{ and } \frac{M}{4^{i+1}} = 2 \end{cases},$$

where j is an integer. Simplifying, we have

$$P = \begin{cases} \frac{3}{2}\log_2 M + 1, & M = 2^{2j}, \\ \frac{3}{2}\log_2 M + \frac{1}{2}, & M = 2^{2j+1}. \end{cases} \tag{4.58}$$

(4.58) shows the number of stages in sparse factorization of DLS matrix for a general case. We can use this result to design a flowgraph of DLS factorization. The number of stages will also help us to evaluate the computational complexity of DLS.

4.6.3 Computational Complexity

We have mentioned that $\boldsymbol{\Phi}$ can be factorized into a series of sparse matrices, which include mainly two parts: Given's rotations for the lapped portion, and DST-IV for the remaining portion. We also demonstrate the relation between DST-IV and DCT-IV, and the factorization of DCT-IV. We can easily test that DST-IV has structures similar to DCT-IV. The only difference requires rotating the Given's rotation matrices and exchanging sine and cosine functions. Therefore, the sparse matrices for DCT-IV can also be used for computing DST-IV. The computational complexity for DCT-IV is thus the same as that for DST-IV. In order to obtain the number of computations, we can calculate the number of computations for $[\boldsymbol{C}_M]$ first, and add them to the number of computations of the first layer factorization.

Now we are ready to count the number of computations. The Given's rotations in the first layer gives $3(L-2)$ multiplications and $3(L-2)$ additions. Now let us look at the computational complexity of $\boldsymbol{C}_M$. It is observed that the butterfly matrices $\boldsymbol{B}^I$, $\boldsymbol{B}^{II}$ and $\boldsymbol{B}^{III}$ have only addition operations; the Given's rotations have both addition operations and multiplication operations. To have identity matrix after extracting a set of sparse matrix from $\boldsymbol{C}_M$, we need the multiplication of the square roots on some diagonal elements. The square root operation adds one more multiplication. Report

[46] develops a sparse matrix factorization for DCT-II matrix. Section 2.2 in [46] is for the factorization of DCT-II which is similar to DCT-IV with sign differences for odd r, where r represents the r-th basis function in $\boldsymbol{C}_M$. Sign difference does not affect the number of stages and computational complexity. Thus we can use the results developed in Section 4.3 of [46]. Considering the repetition of $\boldsymbol{B}_n^I$, $\boldsymbol{G}_n^{II}$ and $\boldsymbol{B}_m^{II}$, and combining the square root operation, we have that for $\boldsymbol{C}_M$:

(1). The number of Multiplication ($\tilde{\mu}_M$) is:

$$\tilde{\mu}_M = \frac{M}{2}(\log_2 M + 2); \tag{4.59}$$

(2). The number of Addition ($\tilde{A}_M$) is:

$$\tilde{A}_M = \frac{3}{2} M \log_2 M. \tag{4.60}$$

Combining the above computation requirements with that in the first layer factorization, we have that

(1). The total number of multiplication (μ_M) is:

$$\mu_M = \tilde{\mu}_M + 3(L-2) = \frac{M}{2}(\log_2 M + 2) + 3(L-2); \tag{4.61}$$

The multiplication of ELT is given by [49],

$$\mu_M = \frac{M}{2}(2K + \log_2 M + 3), \tag{4.62}$$

where M is number of basis functions and K is overlapped factor.

(2). The total number of addition ($\mathcal{A}_M$) is:

$$\mathcal{A}_M = \tilde{A}_M + 3(L-2) = \frac{3}{2} M \log_2 M + 3(L-2). \tag{4.63}$$

The addition of ELT is given by [49],

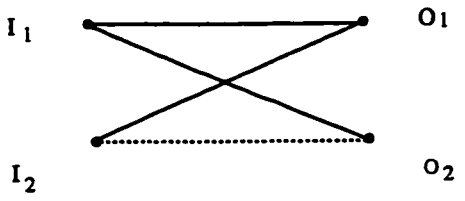
$$\mathcal{A} = \frac{M}{2}(2K + 3\log_2 M + 1). \tag{4.64}$$

The direct computation of DLS transform requires the computation which is on the order of M^2 for both multiplications and additions. Using the sparse matrix factorization, many more computations have been saved, and a fast and efficient algorithm is achieved.

4.7 Several Practical Flowgraphs

In the last two sections, we discussed how to compute the DLS and DLC transform in a fast and efficient way. In this section, we will show two examples for the implementation of DLS when $M = 4$, $L = 2$, and $M = 8$, $L = 4$. Figures 4.3 and 4.4 give the flowgraphs of $M = 4$, $L = 2$, and $M = 8$, $L = 4$ respectively. Some symbols in these figures are explained as follows:

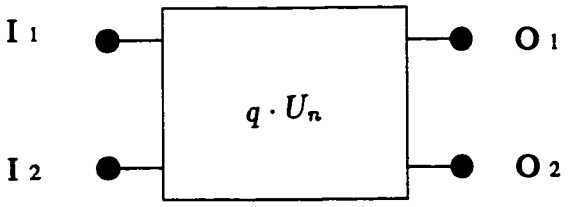
(1).



$$\textit{where,} \quad O_1 = I_1 + I_2,$$
$$O_2 = I_1 - I_2.$$

This block implies a butterfly operation which involves two additions.

(2).



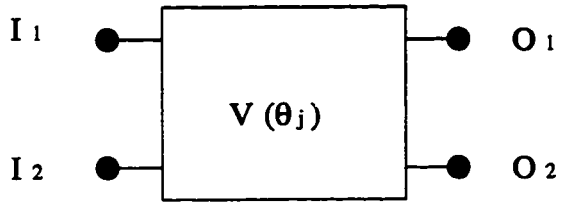
$$\textit{where,} \quad O_1 = I_1 \cdot q \cdot \sin \tfrac{n\pi}{4M} + I_2 \cdot q \cdot \cos \tfrac{n\pi}{4M},$$
$$O_2 = -I_1 \cdot q \cdot \cos \tfrac{n\pi}{4M} + I_2 \cdot q \cdot \sin \tfrac{n\pi}{4M}.$$

Parameters q and n are determined in part (2) of Section 4.6.1. This block represents Given's rotation operation which contains three multiplications and three additions [47].

(3).

Table 4.1: **Numbers of computation**

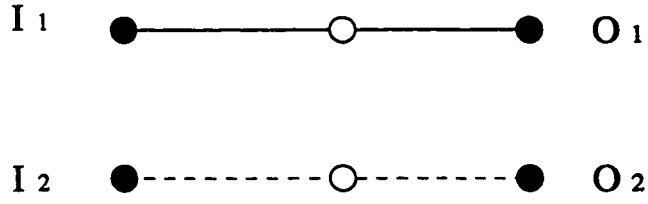
M	L	μ	$\mathcal{A}$
2	2	3	3
4	4	14	18
4	2	8	12
8	8	38	54
8	4	26	42
16	16	90	138
16	8	66	114
16	4	54	102
32	32	202	330
32	16	154	282
32	8	130	258
32	4	118	246



$$\textit{where,} \quad O_1 = I_1 \cdot \cos(\theta_j) + I_2 \cdot \sin(\theta_j),$$
$$O_2 = -I_1 \cdot \sin(\theta_j) + I_2 \cdot \cos(\theta_j).$$

Parameter θ_j is determined in (4.4) of Section 4.3. This block also represents Given's rotations and it contains three multiplications and three additions.

(4).



$$\textit{where}, O_1 = \sqrt{2} I_1 \text{ and } O_2 = -\sqrt{2} I_2.$$

This block contains one multiplication.

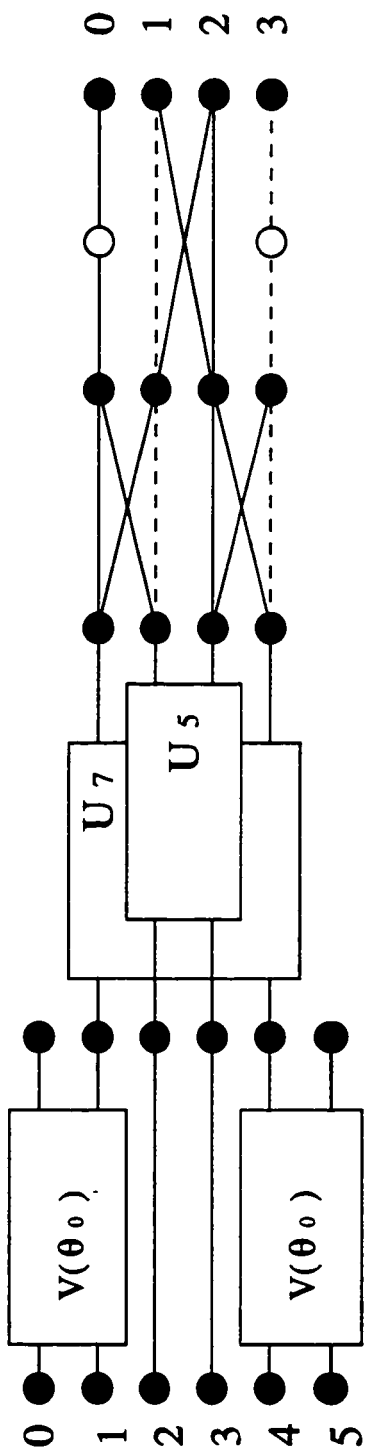


Figure 4.3: M=4, L=2 DLS transform

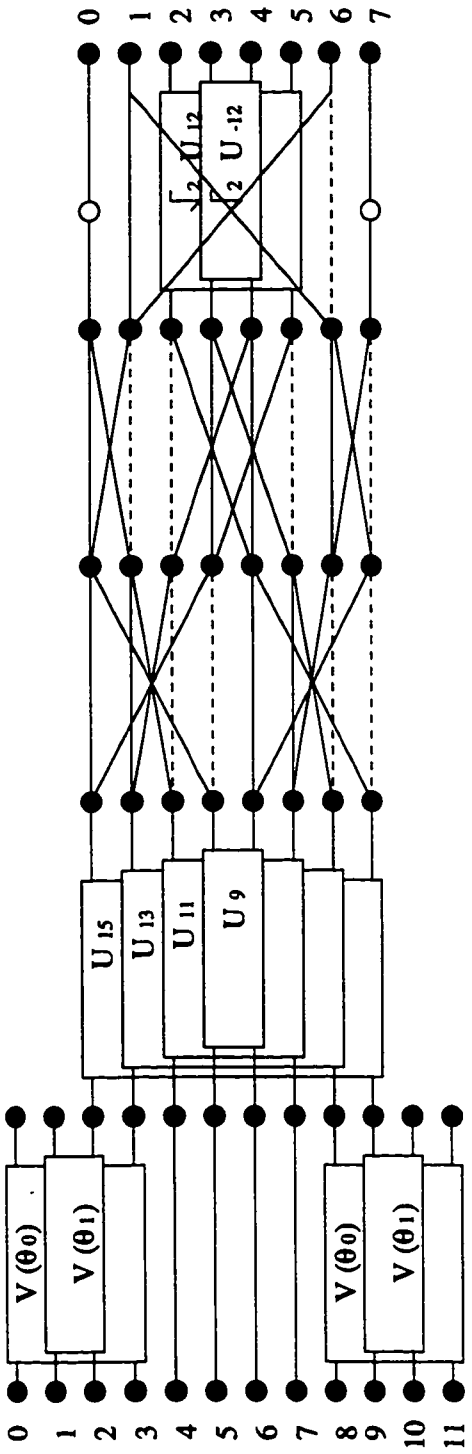


Figure 4.4: M=8, L=4 DLS transform

4.8 Conclusion

In this chapter, we introduce a fast algorithm for computing discrete local sine/cosine transforms. Table 4.1 shows the number of multiplications and additions for different M and L. Because of the partially recursive nature of the structure, we have a very regular architecture which may be implemented in a parallel processor.

Chapter 5

Lapped Transforms for Cancellation of Acoustic Echoes

The conventional adaptive filter is a tool used to cancel the acoustic echoes in a telephone network. It models the echo path to produce a replica of the echo and then subtracts it from the total incoming signal. Since teleconferencing and hands-free telephone systems are increasingly being used in modern communication systems, the traditional echo canceller may not perform well. The reason is that if we use the conventional method, adaptive filters with at least 1000 taps are required to achieve a significant reduction of the echo level [31]. Thus, this method results in very heavy computational burden and slow convergence rate with existing technologies. In teleconferencing/hands-free systems, the movement of the speaker modifies the echo path impulse response. Therefore, it is required that the algorithm used for the acoustic echo canceller must converge fast enough to track these changes [8].

To solve these problems, i.e., to reduce the computation load, several methods, based on multirate filter bank systems with the subsampled signals, have been proposed [13] [11]. It is well known that block transforms have many benefits in digital signal processing and can be used to replace the filter banks [74] . In this chapter, we use block transforms to implement the subband acoustic echo canceller. However, due to the frequency aliasing problem in the filter bank system [36], the direct application of block transform in an echo canceller may not work very well. To solve this problem, we propose an improved method — by changing the subsampling rate in the block transforms, the canceller can track the signal well. The simulation results indicate that transform systems achieve

significant reductions of the echo level with less computation requirements.

In this chapter, we will first overview the use and formulation of filter banks in a subband echo cancellation system. Then we show how to use block transforms to replace filter banks and implement the subband echo canceller. In Section 5.2, the use of different block transforms is studied. In Section 5.3, the problem of directly using block transforms in echo cancellation is discussed. A new optimum lapped transform, based on the criterion of maximum energy concentration, is developed. Using the newly designed lapped transform, we can improve the echo suppression significantly. Simulation results, shown in Section 5.5, are followed by discussions.

5.1 Subband Echo Canceller Using Filter Banks

In multirate digital signal processing, the two key operations are subsampling and upsampling; they are described as

- Subsampling — decimation by an integer factor M; i.e., reducing the sampling rate of $x(n)$ by an integer factor M. To avoid aliasing at this lower sampling rate, it is necessary to filter the signal $x(n)$ with a digital lowpass filter.

- Upsampling — interpolation by an integer factor M. As opposed to subsampling, upsampling by M consists of inserting $M-1$ zeros between two consecutive samples in the input signal.

In multirate systems, a set of filter operations must be involved before the subsampling and after the upsampling processes. This set of filters is called the filter bank. Usually filter banks consist of two parts: one is the analysis bank; the other, the synthesis bank. When a signal goes through the analysis bank, it will first be filtered to create a set of sub-signals, each of which contains the information in a certain frequency bandwidth [71] and is called a subband signal. Each subband signal has a small bandwidth and a reduced sample sequence in comparison with the original signal. Similarly, reconstruction is achieved by upsampling and filtering. It is known [37] that when filter banks are used in an echo cancellation system, the computation load of the echo canceller can be reduced. Figure 5.1 shows the basic structure of a subband acoustic echo canceller[13]. In that

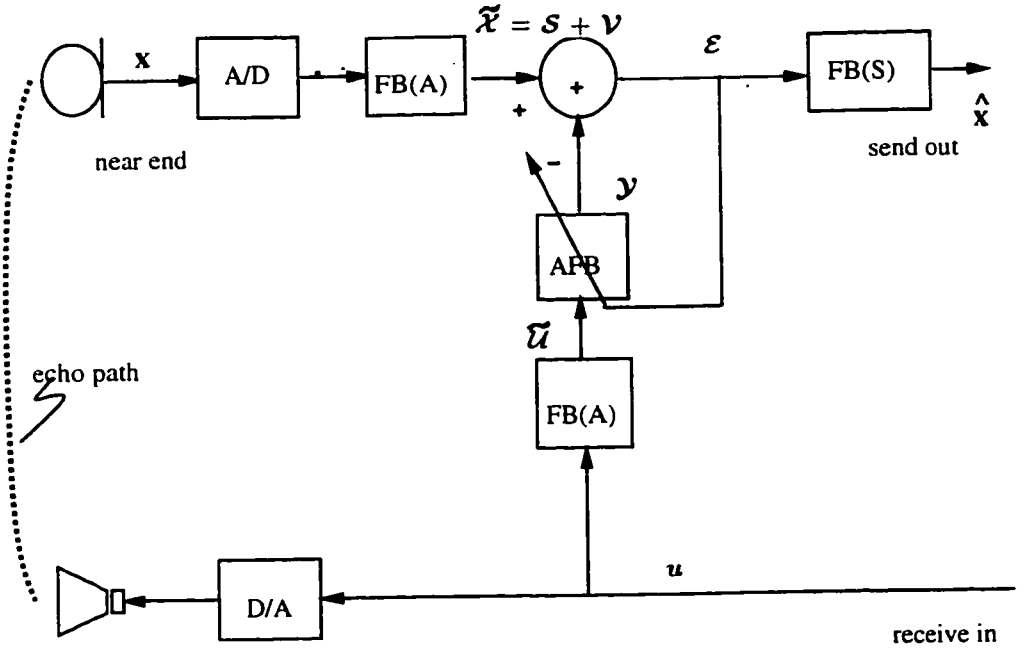


Figure 5.1: Acoustic echo canceller using filter banks

figure, FB(A) and FB(S) are the analysis and synthesis filter banks respectively, and AFB is the adaptive filter bank. Before discussing the principle of the subband echo canceller, we first define some variables which will be used in the rest of the chapter. Let $\tilde{\boldsymbol{U}} = [\tilde{\boldsymbol{u}}_0, \tilde{\boldsymbol{u}}_1, \cdots, \tilde{\boldsymbol{u}}_M]^T$ be an $M \times K$ matrix containing the reference signal after being processed by the analysis filter bank FB(A). Each element $\tilde{\boldsymbol{u}}_r$ in the matrix $\tilde{\boldsymbol{U}}$ is the output of filter bank FB(A) at the r-th subband, and it is a $K \times 1$ column vector with elements $\{\tilde{\boldsymbol{u}}_r(j)\}, j = 1, 2, \cdots, K$. $\boldsymbol{V}$ is an $M \times K$ matrix containing the echo after being decomposed by the same analysis filter bank FB(A). The echo is produced by the coupling between the loudspeaker and microphone. $\boldsymbol{S}$ is an $M \times K$ matrix, denoted as the near-end speech signal after being decomposed by FB(A). The near end speech is the desired signal to be sent out. The reference signal $\tilde{\boldsymbol{U}}$ and the error signal $\boldsymbol{\mathcal{E}}$ are sent into AFB to adjust the weights in the estimated echo path and to create the echo replica. In AFB, one adaptive filter is applied for each subband with a lower sampling rate. By adjusting the weights of the adaptive filter in each subband, we can create an echo replica and then subtract it from the microphone input signal $\tilde{\boldsymbol{X}} = \boldsymbol{S} + \boldsymbol{V}$. The

residual output $\mathcal{E}$ is the echo suppressed signal which is sent out after having been passed through the synthesis filter bank FB(S).

Let the jr-th element of $\widetilde{\mathcal{X}}$ be

$$\widetilde{x}_r(j) = s_r(j) + v_r(j),$$

and let $g_r^{(i)}(j)$ be the j-th weight of AFB in the r-th subband at time i. If the order of the adaptive filters at each subband is P, the output of the adaptive filter at subband r is

$$y_r(j) = \sum_{i=0}^{P-1} g_r^{(j)}(i)\tilde{u}_r(j-i+1).$$

Thus, the residual error $e_r(j)$ (the jr-th element of $\mathcal{E}$) is

$$e_r(j) = \widetilde{x}_r(j) - y_r(j).$$

To minimize the mean square error $e_r(j)$, the weights can be adaptively adjusted by *steepest-descent algorithm*[28], in which the weights are updated as

$$g_r^{(j+1)}(i) = g_r^{(j)}(i) + \mu e_r(j)\widetilde{x}_r(j-i+1),$$

where μ is the *step-size parameter* or *weighting constant* satisfying $0 < \mu < 1$ for the proper convergence of the adaptive algorithm. The above adaptive algorithm is also called *Least Mean Square*(LMS) algorithm. It is a simple and well known adaptive algorithm and we will not discuss it in detail here. The interested reader is referred to [28].

It is known that lapped transform can be used to replace the filter banks. In this chapter, we are mainly concentrating on the implementation of the subband echo canceller using block transforms.

5.2 Echo Canceller with Lapped Transforms

As discussed in Chapter 2, lapped transforms can be represented by a bank of M filters with length N, and the transform coefficients are obtained by downsampling the filtered signal by M. Therefore, lapped transforms can be cast into the filter bank structure. The frequency characteristics of block transforms show that both block transforms and filter

banks have a common feature: i.e., each subband signal is equivalent to each transformed signal obtained by the inner product between the signal and the corresponding basis function in the block transform. The advantages of using block transform instead of filter banks are

1. Block transforms are usually invertible and easy to compute. Thus, the synthesis stage in the filter banks can easily be implemented by the inverse of the block transforms. On the contrary, the general M-band perfect reconstructible filter banks are not easy to design;

2. Most block transforms have a short filter length which means that less delay will be introduced in the reconstructed signal.

3. Most block transforms have fast and parallel implementations. Thus fewer computations are involved in both signal decompositions and reconstructions.

The subband echo canceller using block transform can be implemented as shown in Figure 5.2. We use notations similar to those in Section 5.1. Let $x(k)$ be a mixture of near-end signal and echo with a sampling period of T. $v_r(j)$ is the transformed echo signal with a sampling period MT, where r refers the r-th basis function. $s_r(j)$ is the transformed near-end signal with a sampling period MT, and $\tilde{\boldsymbol{x}}_r(j)$ is the mixture of $s_r(j)$ and $v_r(j)$. Let $u(k)$ be the reference signal with the sampling period T. $\tilde{u}_r(j)$ is the transformed reference signal with the sampling period MT, and $y_r(j)$ is the replica of $v_r(j)$ by passing $\tilde{u}_r(j)$ through an adaptive filter. In a practical telephone system, a double detector must be present whose function is to detect the existence of the near-end speech signal. Because of the strong correlation between the speech signals, an adaptive filter cannot function very well when both the near-end signal and the echo signal are present – a situation referred to as double talk. When the double detector detects the existence of near-end speech, the adaptive filter weights are forced to either freeze or be completely inactive. The adaptive algorithm works when input contains only the echo signal; i.e., $\tilde{\boldsymbol{x}}_r(j) = v_r(j)$. Thus we have

$$y_r(j) = \sum_{i=0}^{P-1} g_r^{(j)}(i)\tilde{u}_r(j-i+1),$$

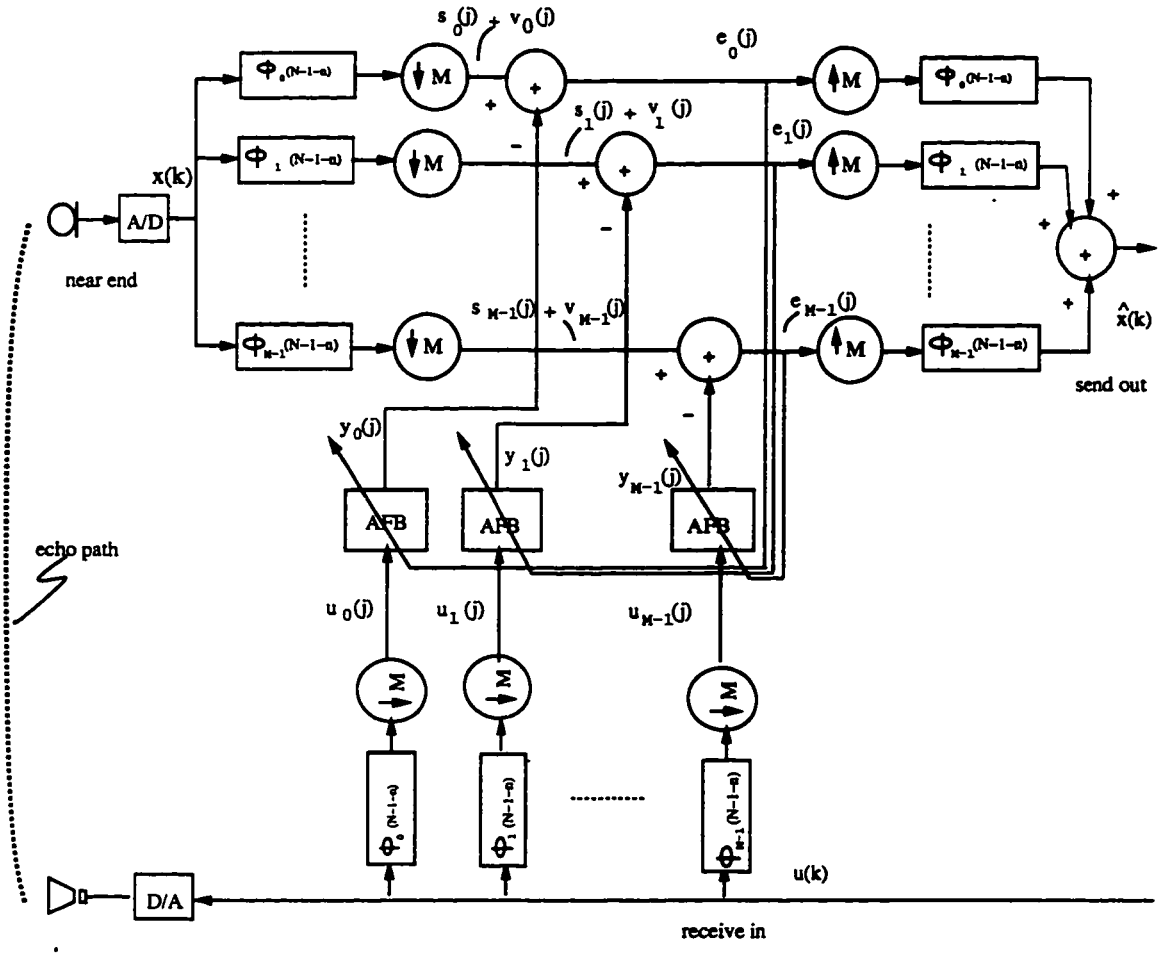


Figure 5.2: Acoustic echo canceller using lapped transform.

where $g_r^{(j)}(i)$ is the impulse response of adaptive filter in the r-th subband at time j. The error $e_r(j)$ is

$$e_r(j) = \tilde{x}_r(j) - y_r(j),$$

and the weights can be updated in the same way as the subband filter banks: i.e.,

$$g_r^{(j+1)}(i) = g_r^{(j)}(i) + \mu e_{r(j)} \tilde{x}_r(j - i + 1).$$

Figure 5.3 shows the simulation results for the echo canceller using two lapped transforms: the DLS and LOT. 8 basis functions are used in both transforms. Comparing with the original echoes shown in (a) of Figure 5.3, the echo residuals at (b) and (c) in

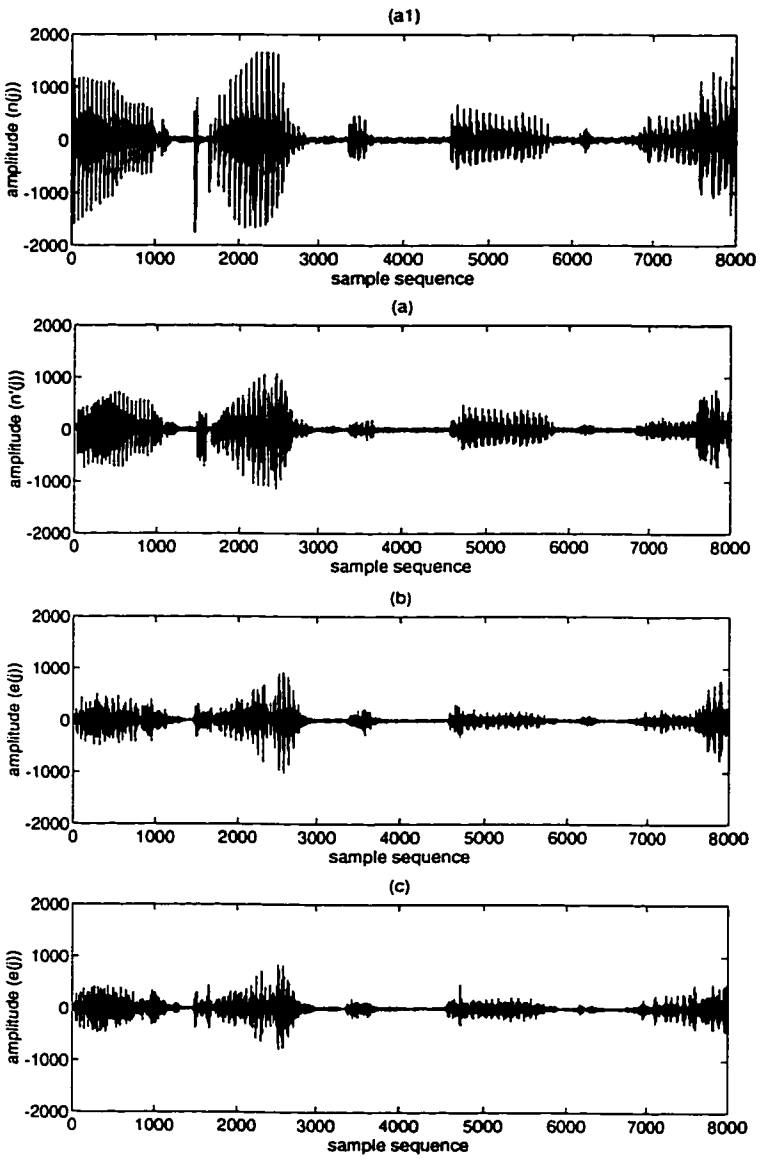


Figure 5.3: Performance comparison for the echo canceller using LOT and DLS transform codings : (8 basis functions and subsampling rate 8): (a1). reference signal $u(k)$; (a). Original echo $v(k)$; (b). Residual echo $\hat{x}(k)$ using DLS (c). Residual echo $\hat{x}(k)$ using LOT.

Figure 5.3 are still considerably large for both DLS and LOT transforms. The reasons are the same as described in [37] for wavelet transform and we describe this briefly later in the context of lapped transforms.

Usually, an echo can be considered as a combination of multi-delays of the reference signal. If the delay is just an integer times the downsampling rate and there are no other distortions, then such an echo can be removed perfectly by the multirate filter bank canceller; the results of such a canceller is shown in Figure 5.4, in which the reference signal is the same as the one used in Figure 5.3(a1). The echo path is a simple delay line with delay being equal to 8 sampling intervals and this number is the same as the downsampling rate M. For the general echo path, the echo cannot be completely cancelled in the reconstructed signal. The reason for the problem has been described in detail in [36] [37]. We discuss this for the lapped transforms in the following section, and then try to solve this problem by designing a suitable lapped transform.

5.3 Signal Distortion in Lapped Transform

The descriptions in [37] indicate that the subband system creates a time-variant echo path after using the downsampling operation. Consider the reference signal $\{u(k)\}$, where k represents the sampling point before the downsampling operation. Assume that the sampling period of the sequence $\{u(k)\}$ is T. The reference signal is transformed by a lapped transform with M basis functions. Let $\tilde{u}_r(j)$ be the transform coefficient of $\{u(k)\}$ with the r-th basis function. The new sampling period of the sequence $\{\tilde{u}_r(j)\}$ is then MT. In general, $u(k-N_r)$ results in $u(j-N_r')$ at output. For an integer N_r, N_r' may or may not be an integer. Now, let us consider an example when the echo $v(k)$ (before transformation) is a copy of the reference signal with a delay N_r; i.e., $v(k) = u(k - N_r)$.

Let $v_r(j)$ be the transform of $v(k)$; i.e., $u(k - N_r)$ with the r-th basis function. We then have

$$v_r(j) = \sum_{k=-\infty}^{\infty} u(k - N_r)h_r(jM - k),$$

where $h_r(j) = \phi_r(N - 1 - j)$. $v_r(j)$ can be further written as

$$v_r(j) = \sum_{k=-\infty}^{\infty} v(k)h_r(jM - k - N_r)$$

$$= \sum_{k=-\infty}^{\infty} v(k) h_r[M(j - \frac{N_r}{M}) - k]$$
$$= \tilde{u}_r(j - \frac{N_r}{M}), \tag{5.1}$$

(5.1) indicates that if $N_r = k \cdot M$ with k being an integer, $v_r(j) = \tilde{u}_r(j-k)$. In this case, we can apply the adaptive filter, and the echo can be cancelled properly. This result has been demonstrated in Figure 5.4. However, in a general situation when N_r is not equal to $k \cdot M$, we have $v_r(j) \neq \tilde{u}_r(j-k)$ for any integer k. Thus, if we apply the adaptive filters directly, the echo cannot be cancelled very well.

Now, let us consider a general echo path which consists of linear combinations of various delay lines. To simplify the discussion, we consider a two-band ($M = 2$) system. Referring to Figure 5.2, the transform coefficients of $u(k)$ (the reference signal) are given by

$$\begin{cases} \tilde{u}_0(j) = \sum_{k=-\infty}^{\infty} h_0(2j-k)u(k); \\ \tilde{u}_1(j) = \sum_{k=-\infty}^{\infty} h_1(2j-k)u(k); \end{cases}$$

where $h_r(j) = \phi_r(N-1-j)$. The transform coefficients of $v(k)$ are given by

$$\begin{cases} v_0(j) = \sum_{k=-\infty}^{\infty} h_0(2j-k)v(k); \\ v_1(j) = \sum_{k=-\infty}^{\infty} h_1(2j-k)v(k). \end{cases}$$

Taking Fourier transforms for both v_r and $\tilde{u}_r$ ($r = 0$ or 1), we obtain

$$\begin{aligned} \tilde{U}_0(\omega) &= \frac{1}{2}[U(\frac{\omega}{2})H_0(\frac{\omega}{2}) + U(\frac{\omega+2\pi}{2})H_0(\frac{\omega+2\pi}{2})], \\ \tilde{U}_1(\omega) &= \frac{1}{2}[U(\frac{\omega}{2})H_1(\frac{\omega}{2}) + U(\frac{\omega+2\pi}{2})H_1(\frac{\omega+2\pi}{2})], \\ V_0(\omega) &= \frac{1}{2}[V(\frac{\omega}{2})H_0(\frac{\omega}{2}) + V(\frac{\omega+2\pi}{2})H_0(\frac{\omega+2\pi}{2})], \\ V_1(\omega) &= \frac{1}{2}[V(\frac{\omega}{2})H_1(\frac{\omega}{2}) + V(\frac{\omega+2\pi}{2})H_1(\frac{\omega+2\pi}{2})], \end{aligned}$$

where $H_r(\omega)$, $\tilde{U}_r(\omega)$ and $V_r(\omega)$ are the Fourier transforms of $h_r(j)$, $\tilde{u}_r(j)$ and $v_r(j)$ respectively. Letting $h_c(k)$ be the impulse response of the echo path, we have

$$v(k) = h_c(k) * u(k);$$

and

$$V(\omega) = H_c(\omega)U(\omega);$$

where $*$ represents the convolution operation, and $H_c(\omega)$ is the Fourier transform of $h_c(k)$. To cancel the echo $\{v_r(j)\}$, the optimum linear filters $\{F_r(\omega)\}$ have to satisfy the following formulas

$$\begin{aligned} F_0(\omega) &= \frac{V_0(\omega)}{\tilde{U}_0(\omega)} \\ &= \frac{H_c(\frac{\omega}{2})U(\frac{\omega}{2})H_0(\frac{\omega}{2}) + H_c(\frac{\omega+2\pi}{2})U(\frac{\omega+2\pi}{2})H_0(\frac{\omega+2\pi}{2})}{U(\frac{\omega}{2})H_0(\frac{\omega}{2}) + U(\frac{\omega+2\pi}{2})H_0(\frac{\omega+2\pi}{2})}; \\ F_1(\omega) &= \frac{V_1(\omega)}{\tilde{U}_1(\omega)} \\ &= \frac{H_c(\frac{\omega}{2})U(\frac{\omega}{2})H_1(\frac{\omega}{2}) + H_c(\frac{\omega+2\pi}{2})U(\frac{\omega+2\pi}{2})H_1(\frac{\omega+2\pi}{2})}{U(\frac{\omega}{2})H_1(\frac{\omega}{2}) + U(\frac{\omega+2\pi}{2})H_1(\frac{\omega+2\pi}{2})}. \end{aligned}$$

The above equations indicate that both $F_0(\omega)$ and $F_1(\omega)$ are signal-dependent. Therefore, the echo cannot be cancelled completely[37]. To remove such signal dependence, we could choose $H_0(\omega)$ and $H_1(\omega)$ as the ideal low and high pass filters respectively [36, 37]; i.e.,

$$\begin{aligned} H_0(\omega) &= 0 \quad |\omega| > \tfrac{\pi}{2}; \\ H_1(\omega) &= 0 \quad |\omega| < \tfrac{\pi}{2}. \end{aligned}$$

Then we have

$$\begin{aligned} F_0(\omega) &= H_c(\tfrac{\omega}{2}); \\ F_1(\omega) &= H_c(\tfrac{\omega+2\pi}{2}). \end{aligned}$$

Summarizing the above results, we see that downsampling introduces distortion in echo cancellation. Such problems may be completely solved if either of the following two conditions is satisfied:

- delay is an integer-multiple of the downsampling rate;
- the filters are ideal bandpass filters.

Because of practical requirements, both of the above conditions are difficult to satisfy. Generally, the problem is caused by the frequency aliasing after the downsampling procedure when the filter banks are not exactly bandpass filters [37]. The well known sampling theorem states that the frequency aliasing problem is more serious when the sampling rate is lower. To reduce aliasing, we could either increase the sampling rate or add a set of filters before the subsampling operations. Figure 5.5 shows the simulation results with

a reduced sampling rate. In Figures 5.5 (a1) and (a), we show both the reference signal and the original echo signal. All the simulation set-ups are the same as in Figure 5.3. However, this time we choose the downsampling rate to be $M = 4$ instead of $M = 8$. It can be observed that the residual echoes are greatly suppressed in comparison with those in Figure 5.3. However, reducing the downsampling rate may create other problems. The main problem would be that the computation complexity would increase. Next, we will introduce an optimum lapped transform for echo cancellation while keeping the same downsampling rate.

5.4 Optimum Lapped Transform for Echo Canceller

In the previous section, we discussed the acoustic echo canceller with lapped transform. We demonstrated that frequency aliasing in the filter banks will cause incomplete echo suppression in the canceller. To reduce the aliasing in filter banks, two strategies are proposed. One is to reduce the downsampling rate, and the other is to develop a set of optimal filters which are close to ideal bandpass filters. In this section, we use the optimal criterion to design such an optimum transform. The optimum transform designing algorithm is described as follows. Let $\mathcal{M}$ be an arbitrary M-dimensional linear space spanned by the basis $\{h_0(n)\}$, $\{h_1(n)\}$, $\cdots$, $\{h_{M-1}(n)\}$, for $n \in [0, M+L-1]$. Any $[(M+L-1)\times 1]$ vector $\boldsymbol{x} \in \mathcal{M}$ can be expressed uniquely as

$$\boldsymbol{x} = \sum_{i=0}^{M-1} \alpha_i \boldsymbol{h}_i, \tag{5.2}$$

where the scalar $\{\alpha_i\}$ refers to transform coefficients and $\boldsymbol{h}_i = [h_i(0), h_i(1), \cdots, h_i(M+L-1)]^T$.

The corresponding discrete Fourier transform $H_r(\omega)$ of the r-th basis function $\boldsymbol{h}_r$ is given by

$$H_r(\omega) = \sum_{n=0}^{M+L-1} h_r(n) e^{-jn\omega}; \quad r \in [0, M-1]. \tag{5.3}$$

The energy of $H_r(\omega)$ in the r-th subband with the bandwidth B_r can be written as

$$\begin{aligned} E_r &= \int_{B_r} |H_r(\omega)|^2 \, d\omega \\ &= \sum_n h_r(n) \sum_m h_r(m) \int_{B_r} e^{-j(n-m)\omega} d\omega. \end{aligned} \tag{5.4}$$

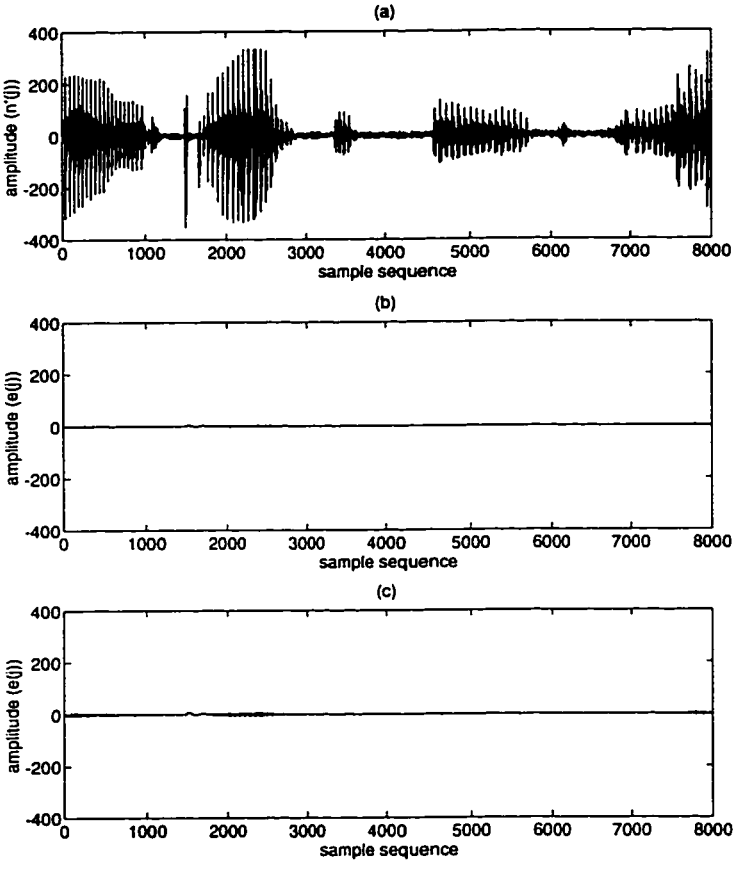


Figure 5.4: Performance comparison for the echo canceller using two transform codings: (delay 8, 8 basis functions and subsampling rate 8): (a). Original echo, in which the delay is just an integer times of the downsampling rate, and there are no other distortions. (b). Residual echo $\hat{x}(k)$ using DLS (c). Residual echo $\hat{x}(k)$ using LOT.

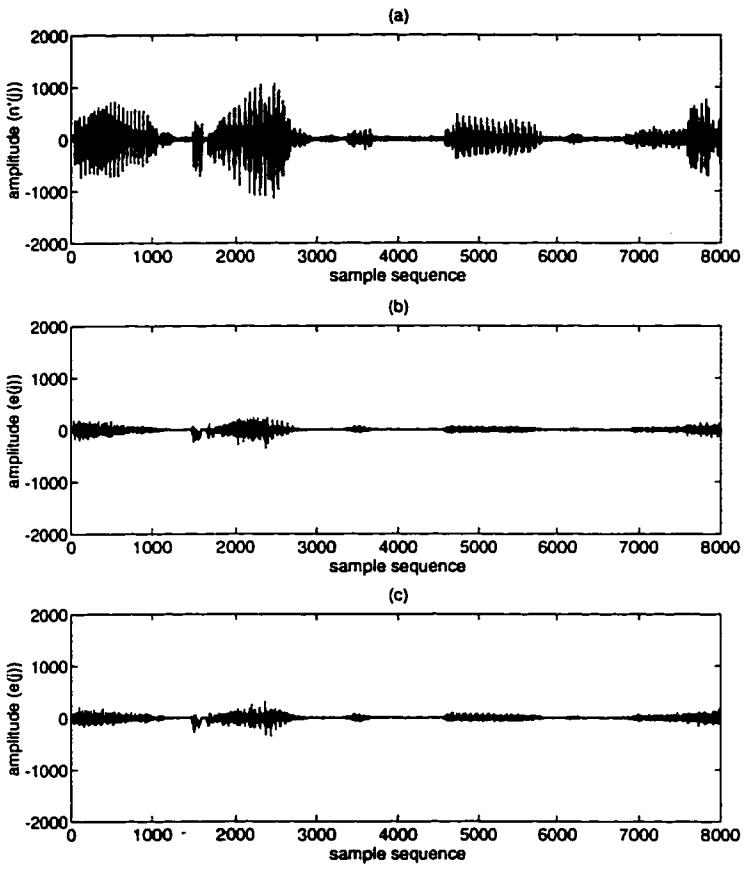


Figure 5.5: Performance comparison for the echo canceller using two transform codings (8 basis functions and subsampling by 4): (a). Original echo, an output of an echo path (with multi-delays and distortion). (b). Residual echo $\hat{x}(k)$ using DLS (c). Residual echo $\hat{x}(k)$ using LOT.

It is known that for a set of complete basis functions, they occupy the whole frequency bandwidth $[-\pi, \pi]$. The bandwidth B_r of the r-th basis function is choosen to be

$$B_r = \{-\frac{\pi(r+1)}{M}, -\frac{\pi r}{M}\} \quad \& \quad \{\frac{\pi r}{M}, \frac{\pi(r+1)}{M}\}.$$

Now let us calculate $\int_{B_r} e^{-j(n-m)\omega} d\omega$ by substituting B_r into the integral and we have

- for the case of $n = m$

$$\int_{B_r} e^{-j(n-m)\omega} d\omega = \frac{2\pi}{M}; \tag{5.5}$$

$$E_r = \frac{2\pi}{M} \sum_n h_r^2(n); \tag{5.6}$$

- for the case of $n \neq m$

$$\int_{B_r} e^{-j(n-m)\omega} d\omega = \frac{4}{n-m} \sin\frac{(n-m)\pi}{2M} \cos\frac{(n-m)\pi(r+\frac{1}{2})}{M}. \tag{5.7}$$

Substituting (5.7) into (5.4), we have

$$E_r = \sum_n h_r(n) \sum_m h_r(m) \frac{4}{n-m} \sin\frac{(n-m)\pi}{2M} \cos\frac{(n-m)\pi(r+\frac{1}{2})}{M}. \tag{5.8}$$

In matrix form, we have

$$E_r = \boldsymbol{h}_r^T \boldsymbol{A}_r \boldsymbol{h}_r, \tag{5.9}$$

where $\boldsymbol{h}_r = [h_r(0), h_r(1), \cdots, h_r(M+L-1)]^T$ and

$$[\boldsymbol{A}_r]_{nm} = \begin{cases} \frac{4}{n-m} \sin\frac{(n-m)\pi}{2M} \cos\frac{(n-m)\pi(r+\frac{1}{2})}{M}; & \text{for } n \neq m; \\ \frac{2\pi}{M}; & \text{for } n = m. \end{cases}$$

Thus, the optimum criterion based on maximizing the energy of $H_r(\omega)$ in the bandwidth B_r is described as

$$\max_{\boldsymbol{h}_r} \quad \boldsymbol{h}_r^T \boldsymbol{A}_r \boldsymbol{h}_r;$$

$$subject \quad to: \begin{cases} \boldsymbol{H}^T \boldsymbol{H} = \boldsymbol{I} \\ \boldsymbol{H}^T \boldsymbol{W} \boldsymbol{H} = \boldsymbol{O} \end{cases}; \tag{5.10}$$

where $\boldsymbol{H}$ is a transform matrix with each column corresponding to a basis function. $\boldsymbol{W}$ is a shift matrix defined in (2.41). The first constraint in (5.10) is for the orthogonality of the lapped transform, and the second is for the lapped orthogonality.

(5.10) indicates that A_r is dependent on r. This is a very complex optimization problem, and there is no simple method to derive the optimum solution for all r. We can solve this optimum problem only through a numerical algorithm. An efficient numerical algorithm is presented as follows:

Numerical Algorithm:

step 1: set $r = 0$ and find the optimal vector for the first basis function h_0 $(r = 0)$ by solving the following problem

$$\begin{aligned} \text{maximize} \quad & h_0^T A_0 h_0 \text{ with respect to } h_0 \\ \text{subject to} \quad & h_0^T h_0 = 1 \\ \text{and} \quad & h_0^T W h_0 = 0 \end{aligned}$$

step 2: set $r = r + 1$ and find the optimal vector for the r-th basis function: i.e.,

$$\begin{aligned} \text{maximize} \quad & h_r^T A_r h_r \text{ with respect to } h_r \\ \text{subject to} \quad & h_r^T h_r = 1 \\ & h_k^T h_r = 0; \quad k = 0, 1, \cdots, r-1 \\ \text{and} \quad & h_k^T W h_r = 0; \quad k = 0, 1, \cdots, r \end{aligned}$$

step 3: if $r < M - 1$, go to step 2. Otherwise, stop and store all h_r.

step 4: end.

In the above algorithm, instead of solving all optimal h_r simultaneously, we try to find the optimal vectors one by one. In this way, the optimization procedure is simplified, but the results we obtain are sub-optimal solutions.

Figure 5.6 shows the frequency response of the optimum basis obtained from the above algorithm for $M = 8$. Table 5.1 shows the corresponding normalized energy distribution in different subbands. The normalized energy is defined as the ratio of the energy in the choosen bandwidth of each basis function to its total energy. The normalized energy distributions of DLS and LOT are also shown in Table 5.1. It can be seen that the optimal bases have much higher energy concentration in the designated bandwidth. This means that by using the optimal lapped transform, less aliasing will occur in the downsampling procedure. In the next section, we will show the simulation results when the new optimal lapped transform is used in the subband echo cancellation.

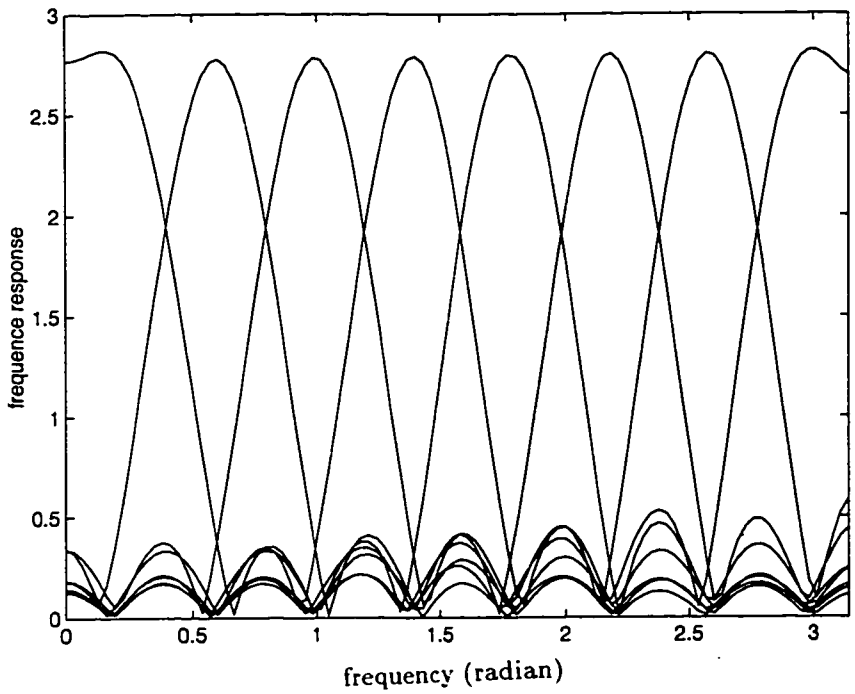


Figure 5.6: Frequency response of optimum lapped bases based on numerical algorithm($M = 8$ and $L = 8$). From left to right, the r-th main lobe corresponds to the r-th subband.

Table 5.1: **Normalized energy distribution of main lobe (E_r)**

	LOT	DLS	optimal lapped trans.
subband 1	0.8326	0.7874	0.8782
subband 2	0.1431	0.5990	0.7814
subband 3	0.1567	0.5953	0.7817
subband 4	0.4228	0.5953	0.7816
subband 5	0.6980	0.5953	0.7830
subband 6	0.1531	0.5953	0.7842
subband 7	0.1482	0.5990	0.7877
subband 8	0.6780	0.7874	0.8725

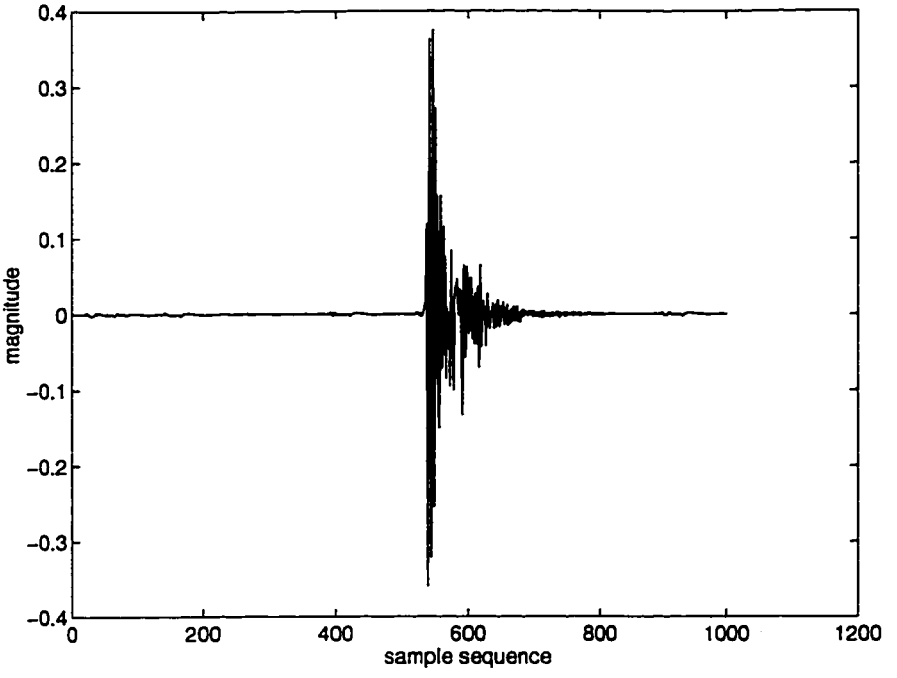


Figure 5.7: Type 3 channel

5.5 Simulation Results

In this section, we show some simulation results and compare subband echo cancellers with different transforms such as the DCT, the LOT, the MLT, the DLS, the Daubechies wavelet and the optimal wavelet designed for echo cancellation [37]. The reason for using the wavelet transforms here is that it was proved in [37] that an optimal wavelet can be designed such that the maximum echo suppression can be reached. In our simulations, we use three types of channels to generate echoes. The first type of channel is a single delay line with the delay being equal to 4 (the worst case [37]). The second is a linear combination of a few delays. The third channel (shown in Figure 5.7) is an impulse response collected in a room environment by a short clap. All transforms contain eight basis functions or subbands. Each basis function contains 16 elements for the DLS, the LOT, the MLT and the optimal lapped transform. However, the basis functions for both the Daubechies wavelet and the optimal wavelet contain 20 elements for each level of decomposition. Figures 5.8(a) and (b) show a reference signal and the echo signal obtained by Type 1 channel. Figures 5.8(c), (d) and 5.9 display the residual echoes

by using the LOT, the MLT, the DLS, the optimum lapped transform, the Daubechies wavelet and the optimum wavelet. Figures 5.10 to 5.13 show the same simulations as Figures 5.8 and 5.9 with channels Type 2 and Type 3 respectively. The corresponding echo suppressions, which are defined as the ratio of the original echo energy to the residual echo energy, are shown in Table 5.2. In this table, we find that the DLS has a slightly higher echo suppression than the LOT for all channel types. However, the optimal lapped transform yields much higher echo suppression than either the LOT, the MLT or the DLS, when considering 0.5dB pace. The MLT is not stable for all types of channels. Comparing with the LOT and DLS, the MLT is only better than them when we use the third channel, but much worse than them while using the first and second channel. Therefore, the MLT may not work well for echo cancellation because its transfer functions does not have linear phase. Comparing with the wavelet transforms, the optimal lapped transform is slightly better than the Daubechies wavelet with a filter length 20 and slightly worse than the optimum wavelet proposed in [37].

In wavelet transform, in order to generate 8 basis functions, we need to decompose both the reference signal and the echo signal into three levels with a set of quadrature mirror filter banks. The length of equivalent 8 band filter banks is 140 [68] which is much longer than lapped transforms. With longer filter length in the filter banks, we can design a better bandpass filter which results in less aliasing in the downsampling procedure. Therefore, much higher echo suppression can be expected. The optimum lapped transform keeps a small equivalent filter length (which is 16). With a shorter filter length, the lapped transform involves less computations, introduces fewer numerical errors, and has a smaller processing delay. Also, its echo suppression performance is close to the optimal wavelet with a filter length of 20. Another advantage of using lapped transform is that its sampling rate can be flexibly adjusted to achieve a desired echo suppression (as demonstrated in Section 5.3). The wavelet transform does not offer this kind of flexibility because it uses downsampling by two in each level of decomposition. This is the minimum downsampling rate and cannot be further reduced.

Table 5.2: **Echo suppression (dB) of echo cancellation**

channel	LOT	MLT	DLS	optimal lapped trans.	wavelet (Daub.)	optimum wavelet
type 1	2.96	2.10	2.98	3.32	2.72	3.50
type 2	8.34	6.25	8.36	10.98	10.34	11.73
type 3	4.79	5.78	4.95	8.95	8.00	9.12

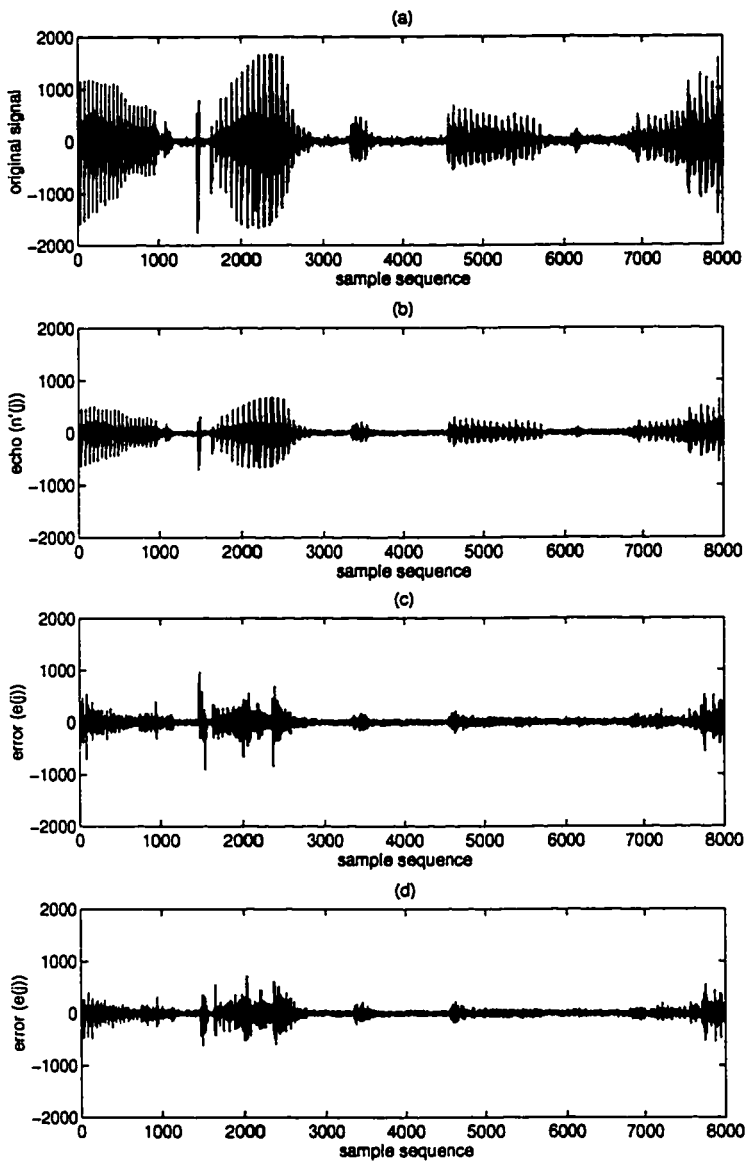


Figure 5.8: (a). Reference signal; (b) Echo with Type 1 channel; (c). Residual echo $\hat{x}(k)$ using LOT; (d). Residual echo $\hat{x}(k)$ using MLT

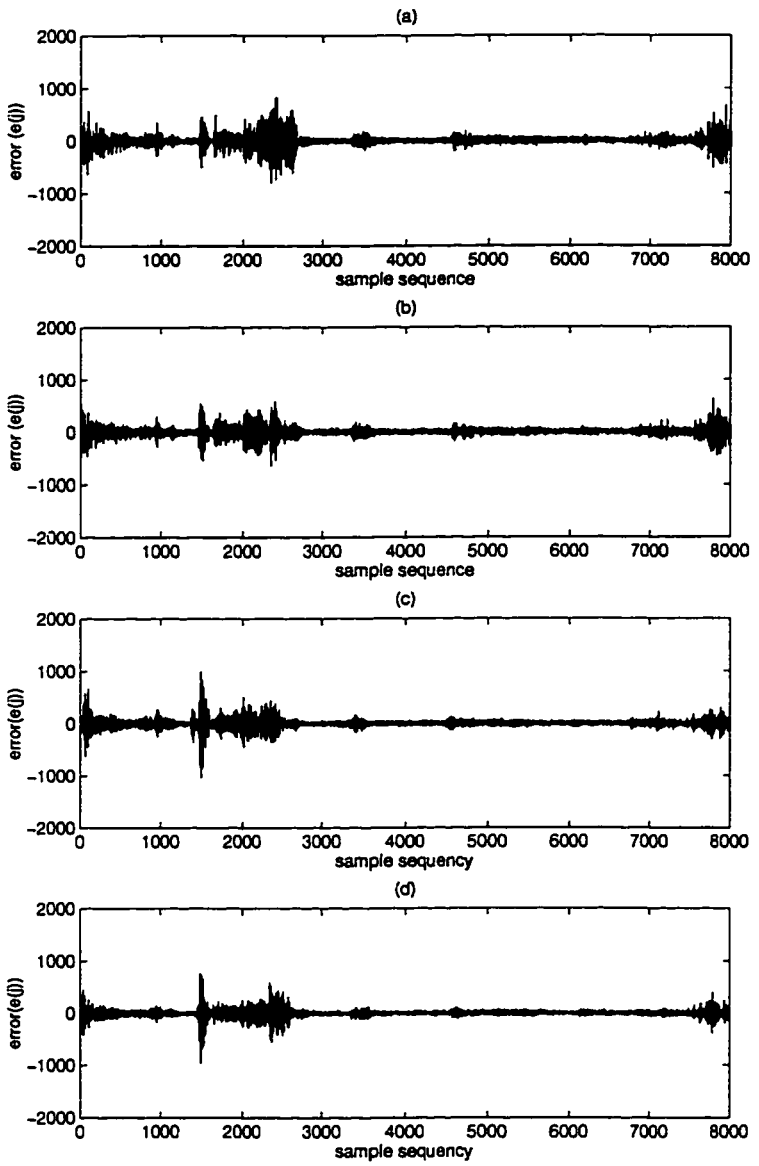


Figure 5.9: Residual echoes $\hat{x}(k)$ with (a). DLS; (b). optimum lapped transform with numerical algorithm; (c). wavelet (Daub.) [36]; (d). optimum wavelet [36]

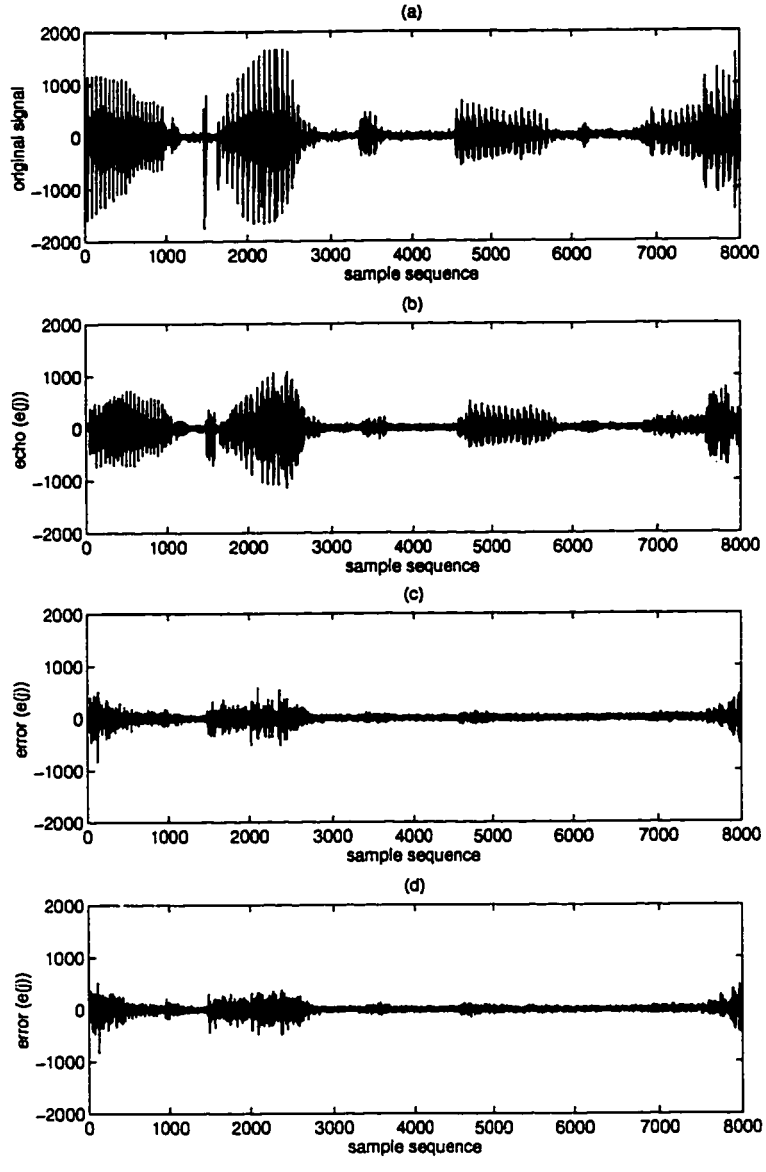


Figure 5.10: (a). Reference signal; (b). Echo with Type 2 channel; (c). Residual echo $\hat{x}(k)$ with LOT; (d). Residual echo $\hat{x}(k)$ with MLT

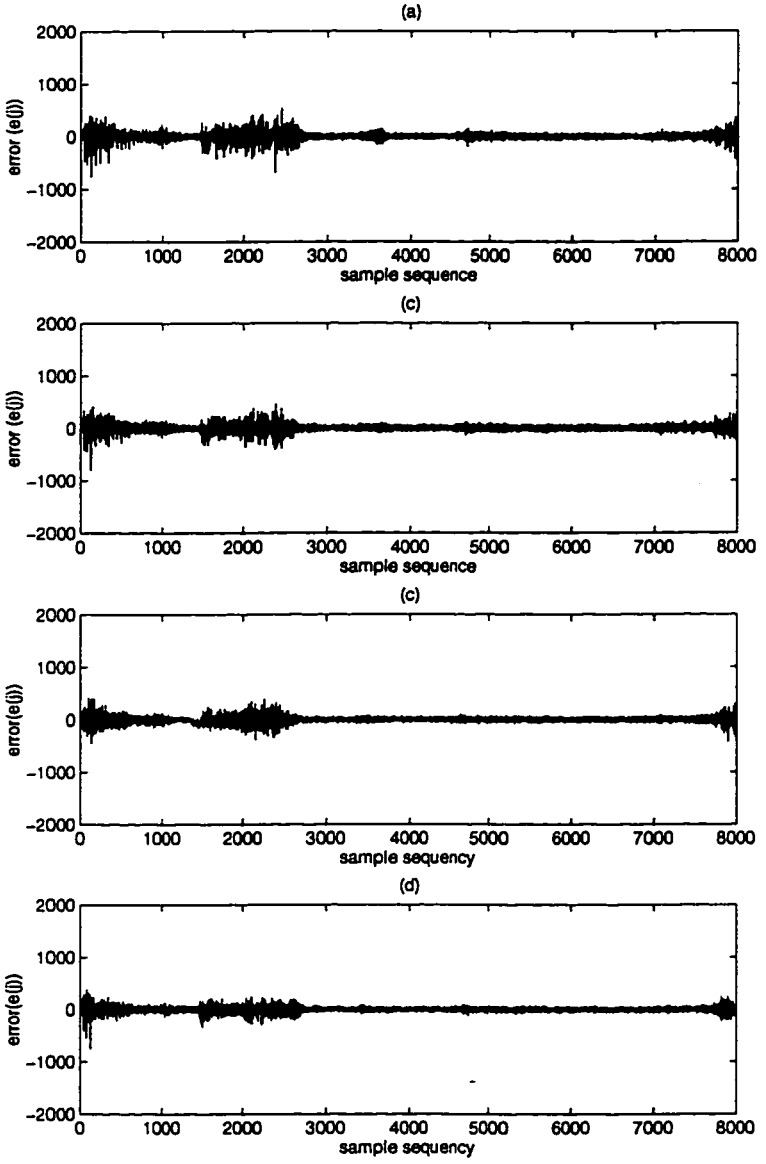


Figure 5.11: Continue for Type 2 channel, residual echoes $\hat{x}(k)$ with (a). DLS; (b). optimum lapped transform with numerical algorithm; (c). wavelet (Daub.) [36]; (d). optimum wavelet [36]

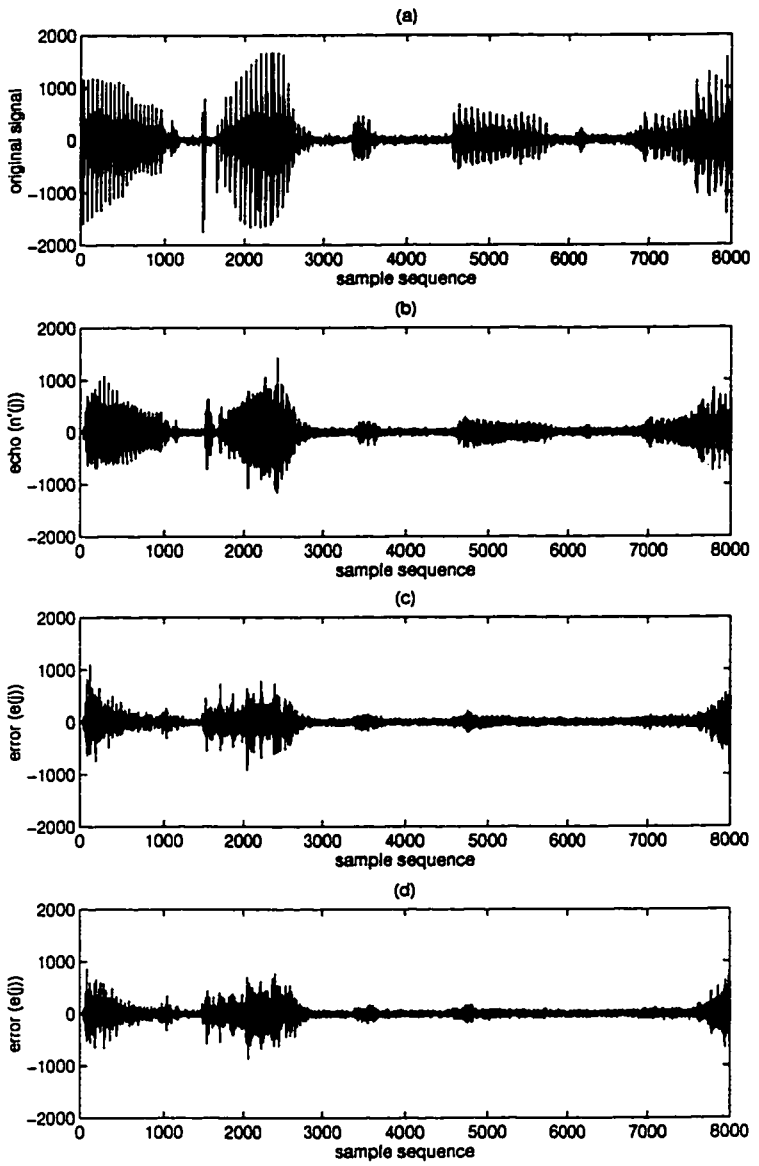


Figure 5.12: (a). Reference signal; (b). Echo with Type 3 channel; (c). Residual echo with LOT; (d). Residual echo with MLT

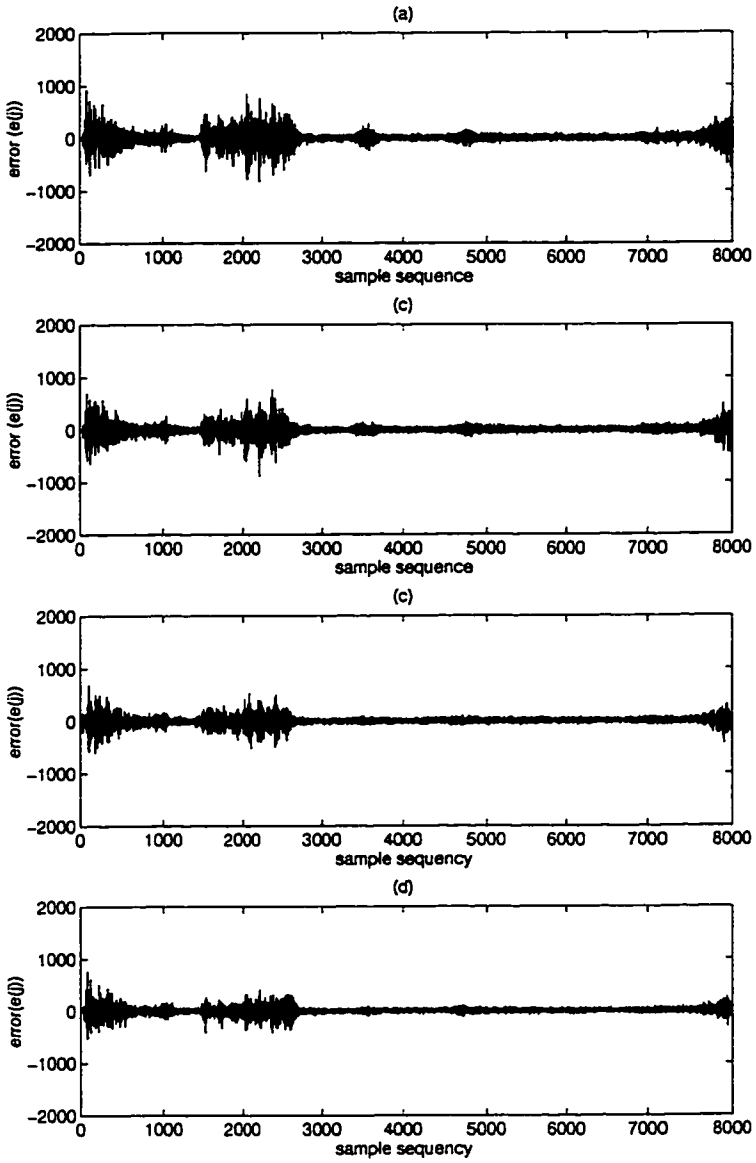


Figure 5.13: Continue for Type 3 channel, residual echoes $\hat{x}(k)$ with (a). DLS; (b). optimum lapped transform with numerical algorithm; (c). wavelet (Daub.) [36]; (d). optimum wavelet [36]

5.6 Complexity of Computation and Discussion

While applying the conventional adaptive algorithm without subband decomposition, for example, we can achieve 26.4dB echo suppression. The performance is better than the echo canceller based on the subband decomposition. This is bacause the aliasing introduced in the subband decomposition. However, using the subband echo canceller, the computation cost can be greatly reduced, which makes the echo canceller easier to be implemented for the long echo path [37].

Now, we will discuss why the computation load can be solved by using subband decomposition. For a traditional adaptive filter, if the length of filter is N, N multiplications are needed to compute the residual error in the cancellation system, and another $2N$ multiplications are required for updating the weights of adaptive filter. In total, we need to implement $3N$ multiplications in a single sampling period. Using subband echo cancellation with M filter bands, due to the downsampling operation, the length of the adaptive filter at each subband is reduced to $\frac{N}{M}$. Therefore in each subband, only $\frac{3N}{M}$ multiplications are required for computing residual errors and updating weights. Considering the total M subbands, the total multiplications are $\frac{3N}{M} \cdot M = 3N$ which is the same as the conventional adaptive filter without subband decomposition. However, those $3N$ multiplications can be implemented in M sampling periods due to the reduced sampling rate. As a result, the computation rate in terms of multiplications per second is greatly reduced in the subband echo cancellation system. Also, because a speech signal does not occupy the entire frequency bandwidth, some of the subbands contains very little energy. By discarding the bands with negligible signal energy, computation complexity can be further reduced. Detailed discussions can be found in [37].

The echo cancellers using lapped transforms and wavelets all belong to the same group — the subband echo canceller. The only difference between them is in the choice of filters for the filter banks. In Chapter 4, a fast algorithm for DLS/DLC is proposed, which has $\frac{M}{2}(\log_2 M + 2) + 3(L-2)$ multiplications in decomposition operations when the length of basis functions is $M + L$. For the optimal lapped transform, no fast algorithm has yet been developed and $M(M + L)$ multiplications are needed for signal decomposition. As for the wavelet packet decomposition, if the length of wavelet basis functions is N,

the number of multiplications required for signal decomposition (convolution operation) is $2(2^{m+1}-2)N$, where m is the level of the wavelet decomposition [68]. If we also consider M subbands, the number of multiplications for wavelet packet decomposition turns out to be $2(2M-1)N$. In general we have $N > M$, and the lapped transforms involve much less computation than the wavelet packet decompositions even without fast implementations.

Also, when the tap length is very long, the convergence of conventional adaptive filter with LMS implementation will be very slow. The numerical error will cause the algorithm not to converge. By breaking the signal into the subbands, the number of adaptive taps in each subband will be reduced by M times and the numerical error can be greatly reduced. As a result, the adaptive algorithm will be easy to converge for the long echo length.

In this chapter, we describe how to use block transforms to implement subband echo cancellation so that the computation load can be reduced while at the same time, the echo can be suppressed. Because most of the block transforms have fast algorithms, we can expect further computation load reduction by using lapped transform. We also show that by either choosing a suitable downsampling rate or designing an optimal transform, the aliasing distortion in the downsampling procedure can be reduced, and higher echo suppression can be achieved. Compared with echo canceller using filter banks, block transform systems are more flexible, easier to design, and requires less computations.

Chapter 6

Image Compression Using Lapped Transforms

Image compression [65], a tool for efficiently encoding a picture, reduces the number of bits required to represent an image. As a result, the required channel bandwidth for image transmission can be reduced. To realize image compression, many coding methods have been developed. Linear transform is one scheme used in image compression. Block transform is a sub-class of linear transforms which offer useful properties for applications. In this approach, the image is subjected to an invertible block transform with an aim to decorrelate the original signal. This decorrelation generally results in the signal energy being redistributed among a small number of transform coefficients. Other coefficients containing little energy are discarded. Block transforms can be divided into two groups: cutoff block transforms and lapped block transforms. Some typical examples of the cutoff block transforms are discrete Fourier transform, discrete cosine transform [66] and Karhunen-Loève transform. As mentioned in Chapters 1 and 2, there exists a problem in the cutoff block transform. This problem is known as the blocking effect, which leads to discontinuities (nonsmoothness) between two adjacent blocks in the reconstructed signal. In image compression, the blocking effects results in a reconstructed image that seems to be built up of small tiles. In many cases, the blocking effect is very pronounced when high image compression rates are used.

Lapped orthogonal transform [49], which was recently proposed, can be used to reduce the blocking effect in image compression. In this chapter, we use a smooth overlapped block transform, the DLS transform, for image compression. Compared with

lapped transform, an important characteristic of the DLS transform is that it involves an arbitrarily smooth cutoff through a bell function. In Chapters 3 and 4, we discuss the properties of the DLS transform and its fast implementations. By using a smooth bell function, a group of new orthogonal bases can be established. The transforms with the bell function not only reduce the blocking effect, but also improve the subband localization. In this chapter, we will first describe the block transform coding system; then we will discuss the optimal lapped transform with the maximum coding gain. A special optimal lapped transform with the highest decorrelation on AR-model is proposed. Finally, some simulation results are presented.

6.1 Block Transform Coding System for Image Compression

A general transform coding scheme involves dividing a $K \times K$ image into $M \times M$ blocks and performing a unitary transform on each sub-image. A unitary transform is an invertible linear transform whose kernel contains a set of complete, orthonormal discrete basis functions. A transform is referred to as one-dimensional (1-D) when it is performed along a single dimension of the image; i.e., along one row or one column of the image. A 1-D transform performed on M pixels along a single row or column is termed an M-point transform (a pixel is defined as a single discrete data point in the image). A transform is two-dimensional (2-D) when it is performed on a 2-D block of the image which contains several rows and columns of image pixels. All the 2-D transforms considered in this chapter are separable: the transform kernel can be decomposed into two 1-D kernels which are operated separately on the row pixels and the column pixels of the image. Thus, a separable transform on an $M \times M$ block of image pixels can be performed in two steps. First, a 1-D M-point transform is performed along each row of the block and then another 1-D M-point transform is performed along each column of the resulting output.

Consider a $K \times K$ image. For an $M \times M$ sub-block of pixels, $x(m,n)$ $(m,n = 0,\cdots,M-1)$, the forward 2-D block transform (concerning non-lapped transform) in

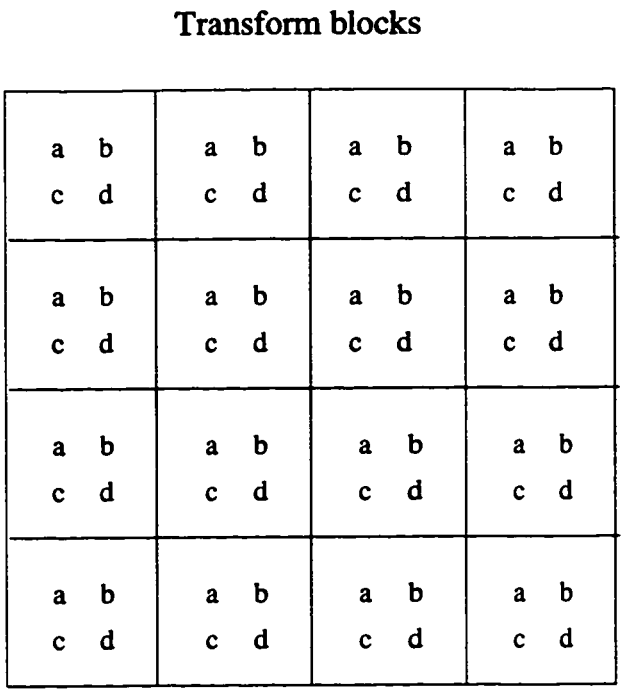


Figure 6.1: Decomposition representation of an image using block transforms

each block is described as [65]

$$X(u,v) = \sum_{n=0}^{M-1} \sum_{m=0}^{M-1} x(m,n)\phi_u(m)\phi_v(n). \tag{6.1}$$

In matrix form, we have

$$\boldsymbol{X} = \boldsymbol{\Phi}^T \boldsymbol{x} \boldsymbol{\Phi}, \tag{6.2}$$

where $\boldsymbol{\Phi}$ is defined as in (2.23). This process, described in Figure 6.1, shows that the image is a 4×4 block matrix. Each block contains four (a, b, c and d) transform coefficients. The inverse 2-D block transform is described as

$$x(n,m) = \sum_{v=0}^{M-1} \sum_{u=0}^{M-1} X(u,v)\phi_u(m)\phi_v(n), \tag{6.3}$$

and in matrix form, we have

$$\boldsymbol{x} = \boldsymbol{\Phi} \boldsymbol{X} \boldsymbol{\Phi}^T. \tag{6.4}$$

It is known that because image pixels are highly correlated, and the energy of an image signal mainly concentrates on a few transform coefficients; i.e., only a few transform coefficients have large variances, and most coefficients have negligible variances. For a

Markov signal, it is proved [66] that DCT has a very good energy concentration in its transform coefficients. This is because the DCT is asymptotically equivalent to KLT [66]. In the cutoff block transforms, when some transform coefficients are abandoned, the blocking effect will appear on the reconstructed image. This block effect can be seen clearly in Figures 6.2 and 6.3. These two figures show the comparisons of the original and the reconstructed compressed images at compression rates of 0.4 bits per pixel (bpp), 0.24 bits per pixel, and 0.16 bits per pixel, when using 16-point DCT. The compression rate, expressed as x bpp, is defined as the average number of bits used to represent one image pixel. The higher the compression rate, the smaller is the bits per pixel value. In Figures 6.2 and 6.3, the original images are represented by 8bpp. It can be observed that the higher the compression rate, the more obvious the blocking effects are. It is easily seen that when the compression rate reaches 0.16bpp, the tiles appear around face, hat and shoulder. If we do not compress the transform coefficients, the reconstructed image is perfect because block transforms are invertible.

The LOT [49], proposed by Malvar, is a new approach for improving the blocking effect in image compression. The LOT extends the cutoff blocks into overlapped blocks such that the boundaries between blocks are blurred, and the blocking effects can be reduced. The LOT satisfies the orthogonality and lapped orthogonality property: by manipulating the blocks, perfect reconstruction can be maintained even at high compression rates. Because of its smooth cutoff property, the DLS transform, another form of lapped block transform system, may also be helpful in reducing the block effect in image compression. The forward 2-D DLS transform formula has a form similar to (6.1), in which the range of summation is set at $[0, M+L-1]$ instead of at $[0, M-1]$. According to the unitary property we derive in Chapter 3, the perfect reconstruction of the DLS transform is guaranteed. The matrix form of the forward 2-D DLS transform in one block is described as

$$\boldsymbol{X} = \tilde{\boldsymbol{\Phi}}^T \boldsymbol{x} \tilde{\boldsymbol{\Phi}}, \tag{6.5}$$

where $\tilde{\boldsymbol{\Phi}}$ is defined by (3.33). The inverse transform is defined as

$$\boldsymbol{x} = \tilde{\boldsymbol{\Phi}} \boldsymbol{X} \tilde{\boldsymbol{\Phi}}^T. \tag{6.6}$$

In this chapter, we will compare the image compression results using the DCT, the LOT

and the DLS.

It is known that KLT is an optimal transform in the group of cutoff block transforms because it completely decorrelates a signal covariance matrix[66]. In the next section, we will introduce the corresponding optimal lapped transform with maximum coding gain which is equivalent to getting the maximum decorrelation of a signal covariance matrix.

6.2 Optimal Lapped Orthogonal Transforms

Our purpose in this section is to find a transform which allows the energy to be more concentrated in a few transform coefficients. To get such an optimal lapped block transform, one can maximize the transform coding gain G_{TC} defined in (3.73) [49]. The problem is formulated to,

$$\begin{aligned} \text{maximize the main diagonal elements (variances) of:} \quad & \boldsymbol{\Gamma}^T \boldsymbol{R}_{xx} \boldsymbol{\Gamma}, \\ \text{subject to } \textit{condition 1}: \quad & \boldsymbol{\Gamma}^T \boldsymbol{\Gamma} = \boldsymbol{I}_M, \\ \text{and } \textit{condition 2}: \quad & \boldsymbol{\Gamma}^T \boldsymbol{W} \boldsymbol{\Gamma} = \boldsymbol{O}_M, \end{aligned} \tag{6.7}$$

where $\boldsymbol{\Gamma}$ is the transform matrix in which each column $\boldsymbol{\gamma}_i$ represents a basis function. $\boldsymbol{R}_{xx}$ is auto-correlation matrix defined as $\boldsymbol{R}_{xx} = E[\boldsymbol{x}\boldsymbol{x}^T]$ and E represents the expectation operation. $\boldsymbol{W}$ is the shift matrix defined in (2.41). $\boldsymbol{I}$ and $\boldsymbol{O}$ are the identity matrix and the zero matrix, respectively. In [49], without any proof and based on maximum coding gain, a so-called quasi-optimal LOT algorithm is designed. In this section, we are first going to develop a general algorithm for the design of an optimal lapped transform and then prove the optimality of this algorithm.

It is well known [55] that the optimization of (6.7) without the condition 2 can be solved by using the eigendecomposition algorithm. To solve the maximization problem under condition 2, we can apply the Lagrangian multipliers on each diagonal element; i.e.,

$$\begin{aligned} \max_{\boldsymbol{\gamma}_i} \boldsymbol{\gamma}_i^T \boldsymbol{R}_{xx} \boldsymbol{\gamma}_i - \lambda_{1i}(\boldsymbol{\gamma}_i^T \boldsymbol{\gamma}_i - 1) - \lambda_{2i} \boldsymbol{\gamma}_i^T \boldsymbol{W} \boldsymbol{\gamma}_i, \\ \text{subject to } \boldsymbol{\gamma}_i^T \boldsymbol{\gamma}_i = 1 \text{ and } \boldsymbol{\gamma}_i^T \boldsymbol{W} \boldsymbol{\gamma}_i = 0, \end{aligned} \tag{6.8}$$

where, λ_{1i} and λ_{2i} denote the Lagrangian multipliers. Differentiating (6.8) with respect

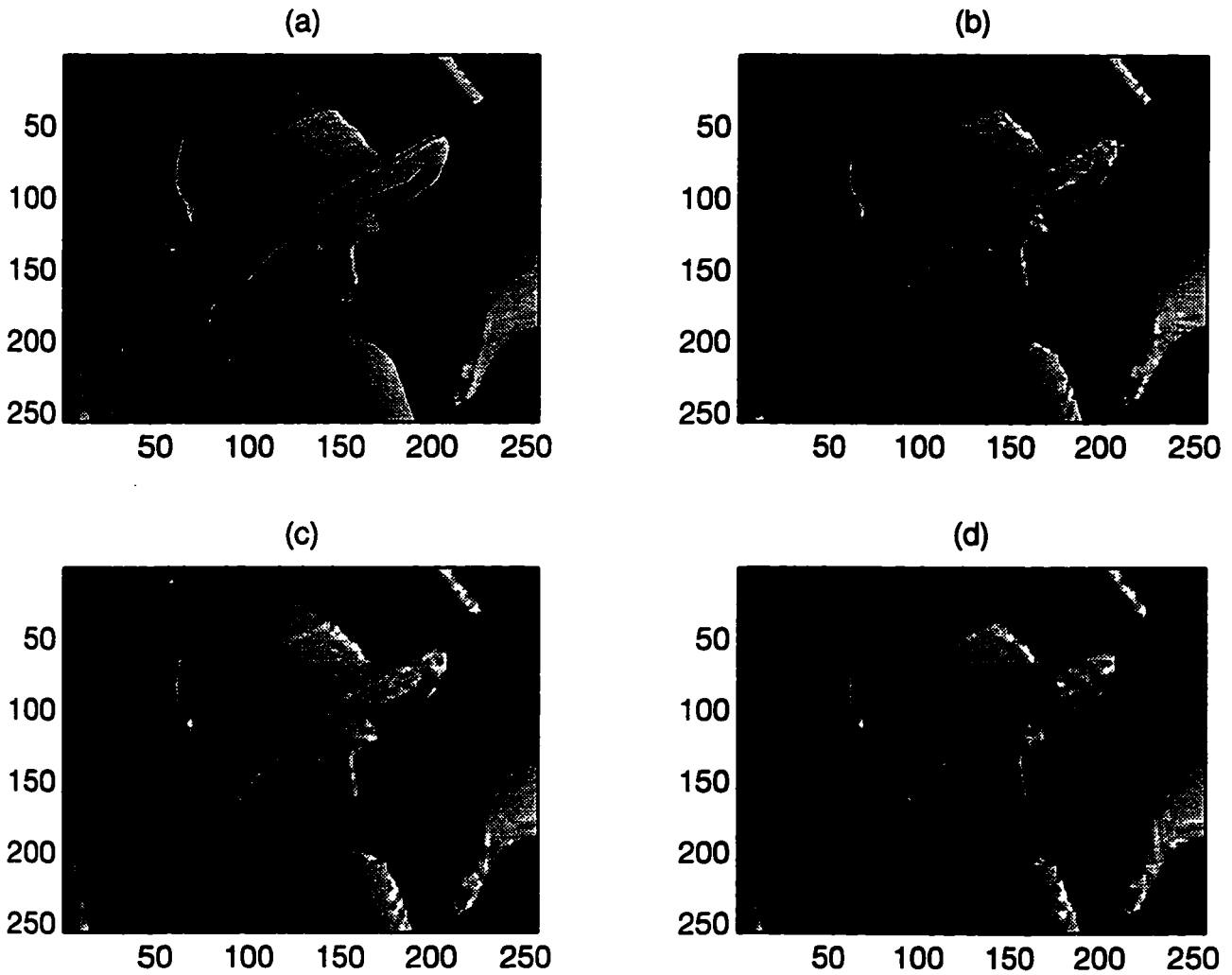


Figure 6.2: Comparisons of original and reconstructed signals with 0.4bpp; 0.24bpp and 0.16bpp using DCT. (a). original image, (b) 0.4bpp; (c). 0.24bpp; (d). 0.16bpp

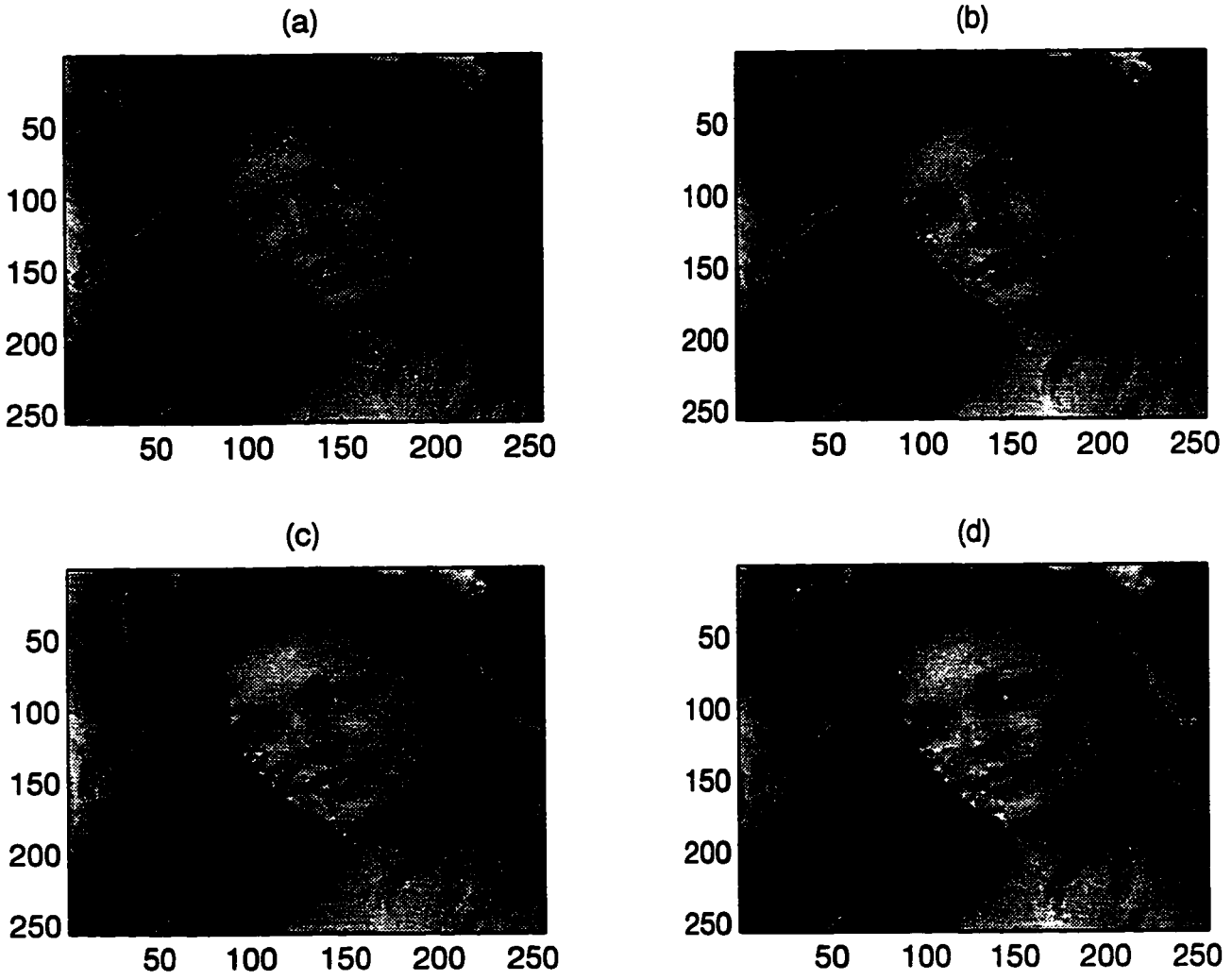


Figure 6.3: Comparisons of original and reconstructed signals with 0.4bpp; 0.24bpp and 0.16bpp using DCT. (a). original image, (b) 0.4bpp; (c). 0.24bpp; (d). 0.16bpp

to γ_i and equating the result to zero, we have

$$2\boldsymbol{R}_{xx}\gamma_i - 2\lambda_{1i}\gamma_i - 2\lambda_{2i}\boldsymbol{W}\gamma_i = 0.$$

Thus, γ_i must satisfy the following equation

$$\boldsymbol{R}_{xx}\gamma_i = \lambda_{1i}\gamma_i + \lambda_{2i}\boldsymbol{W}\gamma_i \tag{6.9}$$

Considering all γ_i in a matrix form, we have

$$\boldsymbol{R}_{xx}\boldsymbol{\Gamma} = \boldsymbol{\Gamma}\boldsymbol{\Lambda}_{1i} + \boldsymbol{W}\boldsymbol{\Gamma}\boldsymbol{\Lambda}_{2i}, \tag{6.10}$$

where

$$\boldsymbol{\Lambda}_1 = \begin{bmatrix} \lambda_{11} & & & 0 \\ & \lambda_{12} & & \\ & & \ddots & \\ 0 & & & \lambda_{1N} \end{bmatrix}$$

and

$$\boldsymbol{\Lambda}_2 = \begin{bmatrix} \lambda_{21} & & & 0 \\ & \lambda_{22} & & \\ & & \ddots & \\ 0 & & & \lambda_{2N} \end{bmatrix}.$$

Pre-multiplying (6.10) by $\boldsymbol{\Gamma}^T$, we obtain

$$\boldsymbol{\Gamma}^T\boldsymbol{R}_{xx}\boldsymbol{\Gamma} = \boldsymbol{\Gamma}^T\boldsymbol{\Gamma}\boldsymbol{\Lambda}_1 + \boldsymbol{\Gamma}^T\boldsymbol{W}\boldsymbol{\Gamma}\boldsymbol{\Lambda}_2. \tag{6.11}$$

Applying the constraint of (6.7), we obtain

$$\boldsymbol{\Gamma}^T\boldsymbol{R}_{xx}\boldsymbol{\Gamma} = \boldsymbol{\Lambda}_1. \tag{6.12}$$

Now the optimal lapped transform can be computed through (6.12) under the two conditions of (6.7). To solve for the optimal solution, we have the following theorem:

Theorem 6.1 *Let $\boldsymbol{\Phi}$ be an arbitrary lapped orthogonal transform matrix in one block (called pre-transform). There exists a matrix $\boldsymbol{V}$ such that the optimal lapped orthogonal basis functions (the solution of 6.7) can be expressed as*

$$\boldsymbol{\Gamma} = \boldsymbol{\Phi}\boldsymbol{V}, \tag{6.13}$$

where $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1\boldsymbol{\phi}_2\cdots\boldsymbol{\phi}_M]$, $\boldsymbol{\Gamma} = [\boldsymbol{\gamma}_1\boldsymbol{\gamma}_2\cdots\boldsymbol{\gamma}_M]$, $\boldsymbol{V} = [\boldsymbol{v}_1\boldsymbol{v}_2\cdots\boldsymbol{v}_M]$, *and* $\boldsymbol{v}_i$ *is the* i*-th eigenfunction of* $\boldsymbol{\Phi}^T\boldsymbol{R}_{xx}\boldsymbol{\Phi}$.

Proof: Substituting (6.13) into (6.12), we have

$$\boldsymbol{V}^T\boldsymbol{\Phi}^T\boldsymbol{R}_{xx}\boldsymbol{\Phi}\,\boldsymbol{V} = \boldsymbol{\Lambda}_1.$$

Let $\tilde{\boldsymbol{R}}_{xx} = \boldsymbol{\Phi}^T\boldsymbol{R}_{xx}\boldsymbol{\Phi}$, so that

$$\boldsymbol{V}^T\tilde{\boldsymbol{R}}_{xx}\boldsymbol{V} = \boldsymbol{\Lambda}_1.$$

To maximize the diagonal elements of $\boldsymbol{\Gamma}^T\boldsymbol{R}_{xx}\boldsymbol{\Gamma}$ is equivalent to maximizing those of $\boldsymbol{V}^T\tilde{\boldsymbol{R}}_{xx}\boldsymbol{V}$[55]. It is known that the solution of such optimization is the eigendecomposition of $\tilde{\boldsymbol{R}}_{xx}$. The remaining problem is to prove that the optimum transform matrix $\boldsymbol{\Gamma} = \boldsymbol{\Phi}\boldsymbol{V}$ satisfies the two conditions described in (6.7) and hence is a lapped transform matrix.

Let $\{\boldsymbol{v}_i\}$ be the eigenvector of $\tilde{\boldsymbol{R}}_{xx}$ corresponding to the i-th eigenvalue. From the property of eigendecomposition, we have

$$\boldsymbol{v}_i^T\boldsymbol{v}_j = \delta_{ij}.$$

Because $\boldsymbol{\Phi}$ is a lapped orthogonal transform matrix which satisfies $\boldsymbol{\Phi}^T\boldsymbol{\Phi} = \boldsymbol{I}_M$ and $\boldsymbol{\Phi}^T\boldsymbol{W}\boldsymbol{\Phi} = \boldsymbol{O}_M$, and $\boldsymbol{\Gamma}$ is related to $\boldsymbol{V}$ as in (6.13), we have

$$\boldsymbol{\Gamma}^T\boldsymbol{\Gamma} = \boldsymbol{V}^T\boldsymbol{\Phi}^T\boldsymbol{\Phi}\,\boldsymbol{V} = \boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}_M$$

$$\boldsymbol{\Gamma}^T\boldsymbol{W}\boldsymbol{\Gamma} = \boldsymbol{V}^T\boldsymbol{\Phi}^T\boldsymbol{W}\boldsymbol{\Phi}\,\boldsymbol{V} = \boldsymbol{O}_M$$

Now, we have proved that $\boldsymbol{\Gamma}$ satisfies both lapped orthogonal conditions in (6.7) and is optimal. □

From the proof of the above theorems, we make the following observations:

1. (6.13) provides a set of optimal lapped bases with a given $\boldsymbol{\Phi}$, the pre-transform;
2. the space spanned by $\boldsymbol{\Gamma} = [\boldsymbol{\gamma}_1\boldsymbol{\gamma}_2\cdots\boldsymbol{\gamma}_M]$ is the same as that spanned by $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1\boldsymbol{\phi}_2\cdots\boldsymbol{\phi}_M]$;
3. $\boldsymbol{v}_i$ are the eigenvectors of transformed matrix $\boldsymbol{\Phi}^T\boldsymbol{R}_{xx}\boldsymbol{\Phi}$;

4. using the optimal lapped transform $\boldsymbol{\Gamma}$, the coding gain can be maximized.

Theorem 6.1 also gives us a simple algorithm to obtain the optimal local sine/cosine transforms using (3.1) and (3.2). Similar to KLT, this optimal lapped transform is image dependent. In practical applications, an image dependent basis is not convenient to use, and it involves large amounts of computations. However, there is convicing evidence [66] that most images are close to an AR model. Therefore, if we can find an optimal lapped transform which can best decorrelate an AR-model signal, this transform which is signal independent can be easily applied to the compression of most images.

6.3 Sub-Optimal Lapped Transform based on AR-model Signal

In practical system design, we often compromise between performance and computations. The optimal lapped transform proposed in the last section may not be useful in applications because large amounts of computations are required to obtain a set of signal dependent bases. It is known that most signals, especially images, can be considered as AR-model-like signals. If we can find a transform which is optimal for an AR-model signal, it will be generally good for any image signals. Extending Theorem 6.1, we have the following corollary for the design of an optimal lapped transform with the highest decorrelation of AR-model .

Corollary 6.1 *Let $\boldsymbol{\Phi}$ be an arbitrary lapped orthogonal transform matrix in one block (called pre-transform), for any AR-model signal, there exists a matrix $\boldsymbol{V}$ such that the optimal lapped transform, which can highly decorrelate this AR-model signal, can be expressed as*

$$\boldsymbol{\Gamma} = \boldsymbol{\Phi V}, \tag{6.14}$$

where $\boldsymbol{\Phi} = [\phi_1 \phi_2 \cdots \phi_M]$, $\boldsymbol{\Gamma} = [\gamma_1 \gamma_2 \cdots \gamma_M]$, $\boldsymbol{V} = [v_1 v_2 \cdots v_M]$ and v_i is the eigenvector of $\boldsymbol{R}_A$ corresponding to the i-th largest eigenvalue of $\boldsymbol{R}_A$. $\boldsymbol{R}_A$ is the covariance matrix of the AR-model signal.

Proof: This Corollary is the simple extension of Theorem 6.1 with the signal being an AR-model signal. □

Now (6.14) provides a set of optimal lapped bases in the sense of complete decorrelation of AR-model image, and this set of transform bases is image independent. We call this set of bases as "sub-optimal bases" because it is optimal only for the AR-model signal. In the next section, we will show some simulation results of applying this "sub-optimal" bases to image compression.

6.4 Simulation Results

In image processing using transforms, the transformed data can be considered as the original image being filtered to create a set of signals, each of which contains the information of the original image in certain frequency bandwidths. Because most image signals do not occupy the whole frequency bandwidth, many coefficients can be discarded after quantization prior to encoding, resulting in image compression.

In simulation, we use the following image compression algorithm. Once an image is transformed and the energy in each band is computed, transform coefficients are quantized. Many coefficients which are less than a threshold are then discarded and only those holding larger energies are transferred through a channel. After that, the compressed coefficients are reconstructed by inverse transform. Here we use the simple quantization. The threshold is calculated by energies and compression rate given in the beginning.

Figures 6.4, 6.5, 6.6, 6.7, 6.8 and 6.9 show the comparisons of the original image and compressed images when using various transforms with $M = 16$. The original image is represented with 8 bits per pixel, and the compressed images are represented with 0.4 bits per pixel, 0.24 bits per pixel, and 0.16 bits per pixel, respectively. In our simulations, we use the standard Lena image of dimensions 256×256. Table 6.1 indicates the comparison of signal-to-noise ratio in dB for different transforms, in which OLT denotes the optimal lapped transform based on the first order AR-model signal with $\rho = 0.95$ [66]; i.e., the sub-optimal bases for image compression. For lapped transforms, the borders of each image is processed by padding zeros. The signal-to-noise ratio is defined as the ratio between the energy of the original image and that of the residual image which is the difference between the original image and reconstructed compressed

Table 6.1: **Comparison of signal-to-noise ratio (dB) (lena)**

	DLS	LOT	MLT	DCT	Sub-OLT based on AR-model.
0.4bpp	16.3	15.8	16.5	13.9	16.8
0.24bpp	13.8	13.6	14.3	12.2	14.5
0.16bpp	12.2	12.2	12.7	11.2	13.0

image. It can be observed that under different compression rates, the optimal lapped transform based on the AR-model signal always achieves the highest signal-to-noise ratio when it is compared with DCT, and with LOT, and with MLT, and finally with DLS. The signal-to-noise ratio using the DLS is higher than that using LOT. The MLT is even better than the DLS. When comparing the reconstructed images in Figures 6.4, 6.5, 6.6, 6.7, 6.8 and 6.9, it is found that there are obvious tiles at face, shoulder and hat in the compressed image with LOT, but for the DLS, the MLT and the OLT the tiles cannot be perceived. This phenomenon can be observed more clearly when using high resolution screens. Compared with the non-smooth cutoff system, the LOT improves the blocking effect only slightly. The local sine transform and the MLT, as a smooth block transform, achieves improvement on the blocking effect. Much more improvement can be realized with the "sub-optimal" lapped transform.

Figures 6.10, 6.11, 6.12, 6.13 6.14 and 6.15 show another example of comparing the compressed image with the original image when using various transforms with $M = 16$. Once again the compressed images are represented with 0.4 bits per pixel, 0.24 bits per pixel, and 0.16 bits per pixel, respectively. Comparing the compressed images with the original image, we may get the same conclusion as before. That is, there are obvious tiles on the face and shoulder with the LOT; with the DLS, with the MLT and the OLT the tiles are not so pronounced. Also the sub-optimal lapped transform based on the AR-model achieves significant improvement in reducing the blocking effect. Table 6.2 shows the comparison of signal-to-noise ratios in dB using DLS, DCT, LOT, MLT and the optimal lapped transform based on AR-model. All our simulation results indicate that the sub-optimal lapped transform consistently performs well for different images, and the signal-to-noise ratios using MLT are higher than those using LOT and DLS for this image.

In this chapter, block transform coding systems are described. One phenomenon

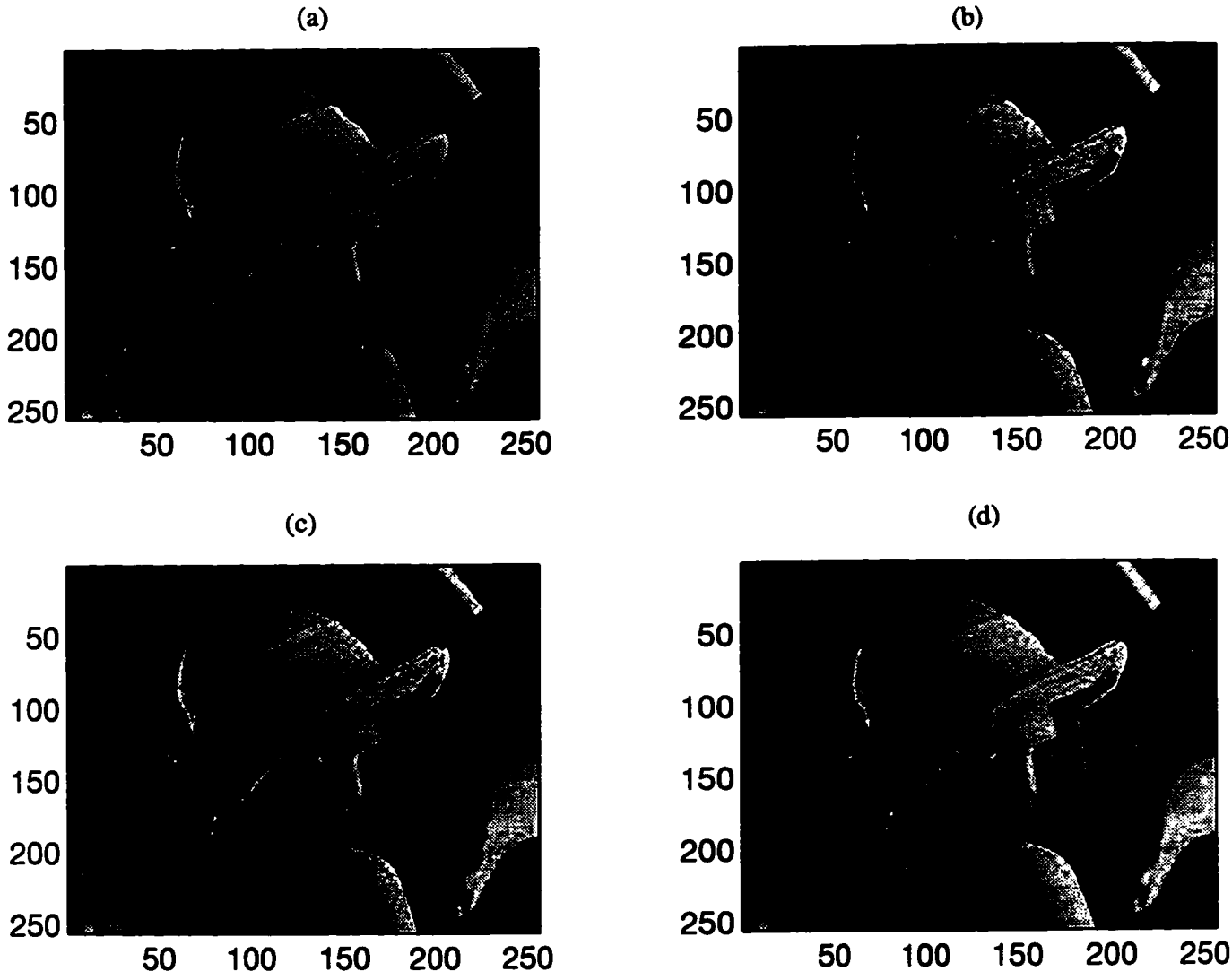


Figure 6.4: Comparisons of original and reconstructed signals with $M = 16$ and 0.4 bits per pixel. (a). original image, (b). DLS. (c). LOT (d). Sub-optimal lapped transform based on AR-model.

Table 6.2: **Comparison of signal-to-noise ratio (dB) (face)**

	DLS	LOT	MLT	DCT	Sub-OLT based on AR-model.
0.4bpp	16.8	16.2	17.0	14.9	17.2
0.24bpp	14.7	14.1	15.1	13.3	15.4
0.16bpp	13.3	13.2	13.7	12.4	14.0

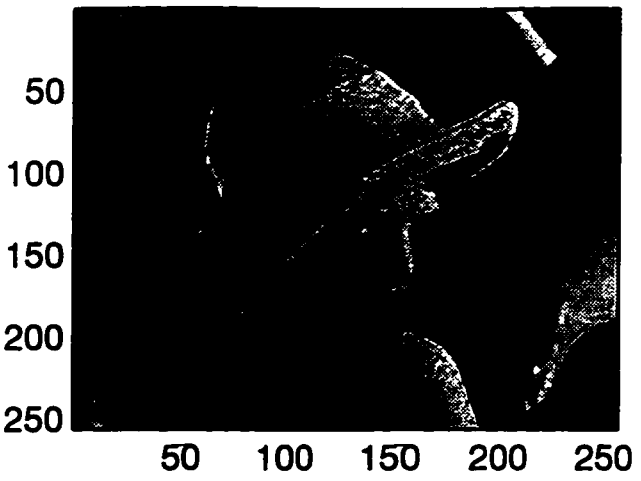


Figure 6.5: Continue ... Comparisons of original and reconstructed signals with $M = 16$ and 0.4 bits per pixel. (e). MLT

shows that DLS, as a smooth block system, may work well under a high compression system, when comparing with DCT and LOT. However, the MLT may achieve even better compression results. Also, because of the flexibility of length in the overlapped region in the smooth block transform, the performance can be improved further. The optimal lapped transform reaches the maximum coding gain, but leaves an open question for computing such data-dependent transforms. In Section 6.3, we introduce a sub-optimal lapped transform based on the AR-model. The simulation results shows that such a sub-optimal transform provides improved results for reducing the blocking effects.

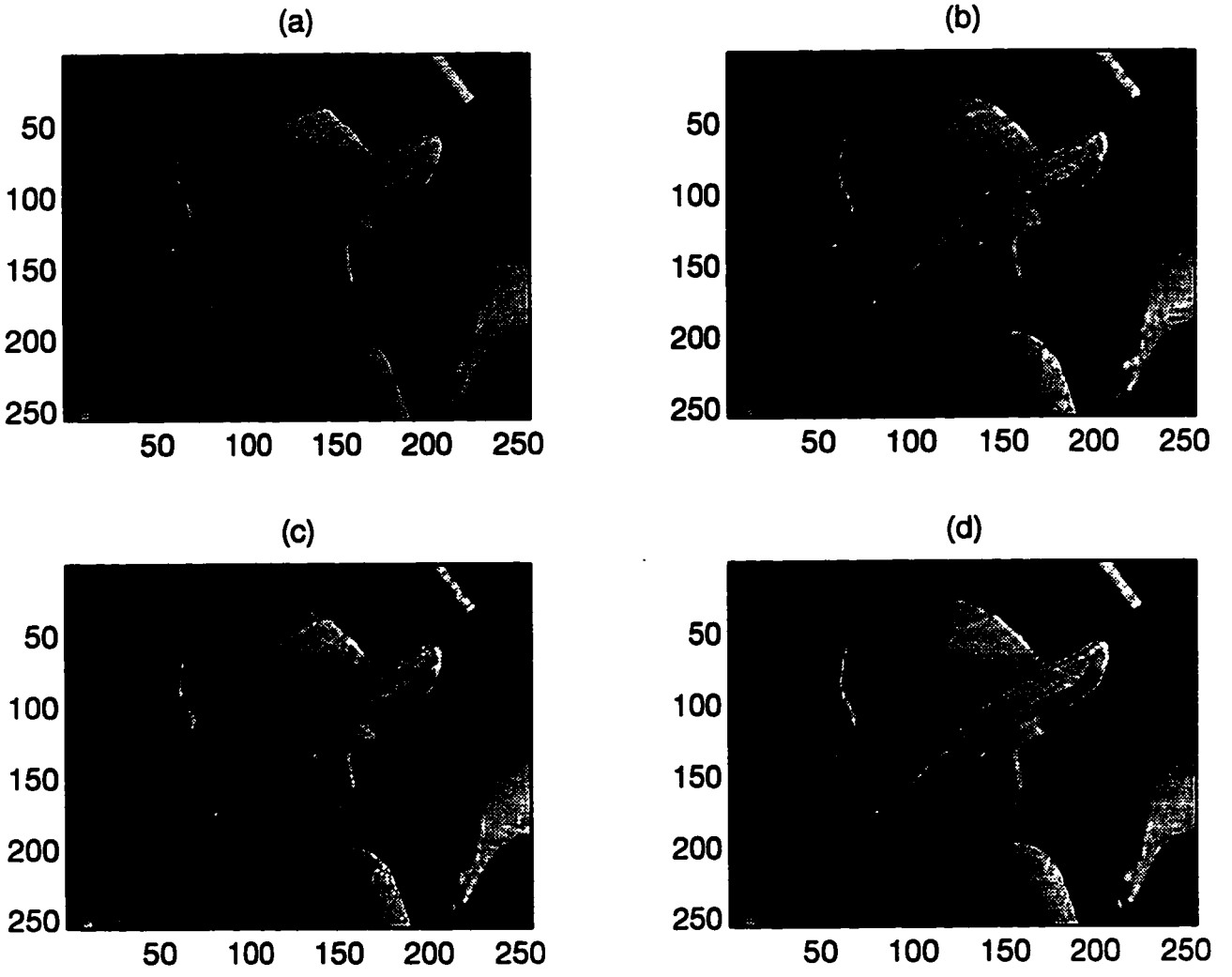


Figure 6.6: Comparisons of original and reconstructed signals with $M = 16$ and 0.24 bits per pixel. (a). original image, (b). DLS. (c). LOT (d). Sub-optimal lapped transform based on AR-model.

(e)

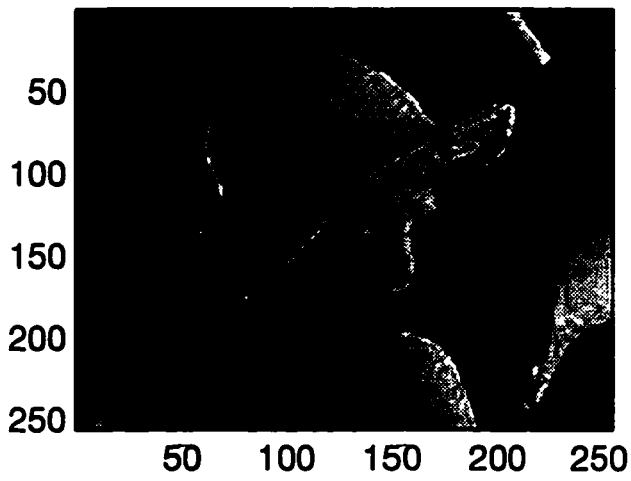


Figure 6.7: Continue ... Comparisons of original and reconstructed signals with $M = 16$ and 0.24 bits per pixel. (e). MLT

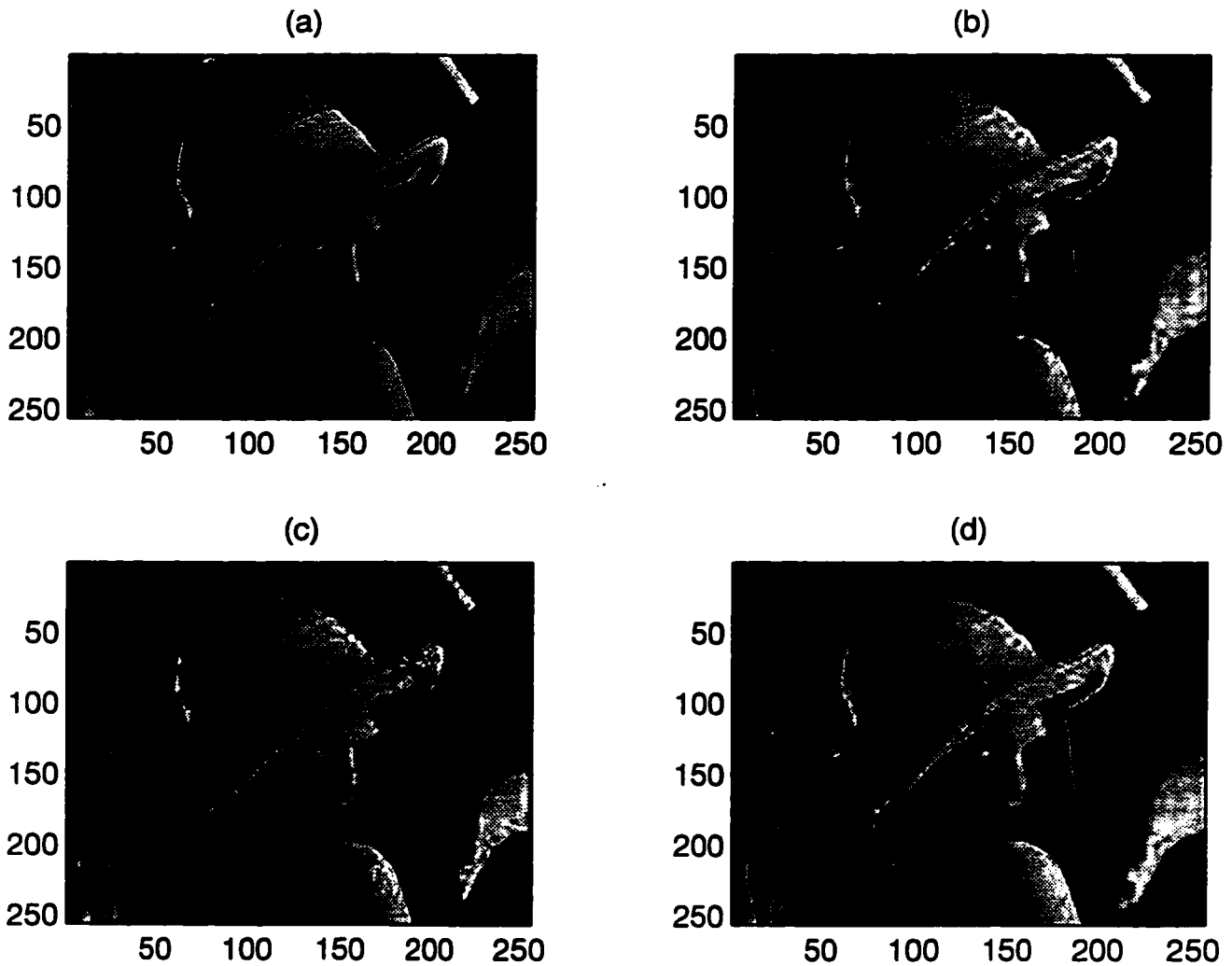


Figure 6.8: Comparisons of original and reconstructed signals with $M = 16$ and 0.16 bits/pixel. (a). original image, (b). DLS. (c). LOT (d). Sub-optimal lapped transform based on AR-model.

(e)

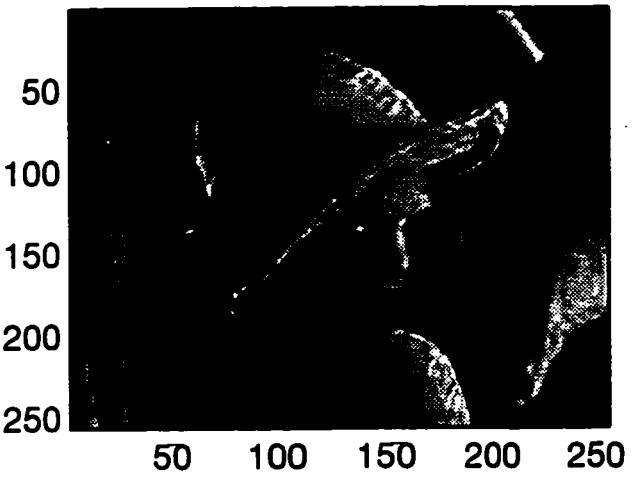


Figure 6.9: Continue ... Comparisons of original and reconstructed signals with $M = 16$ and 0.16 bits/pixel. (e). MLT

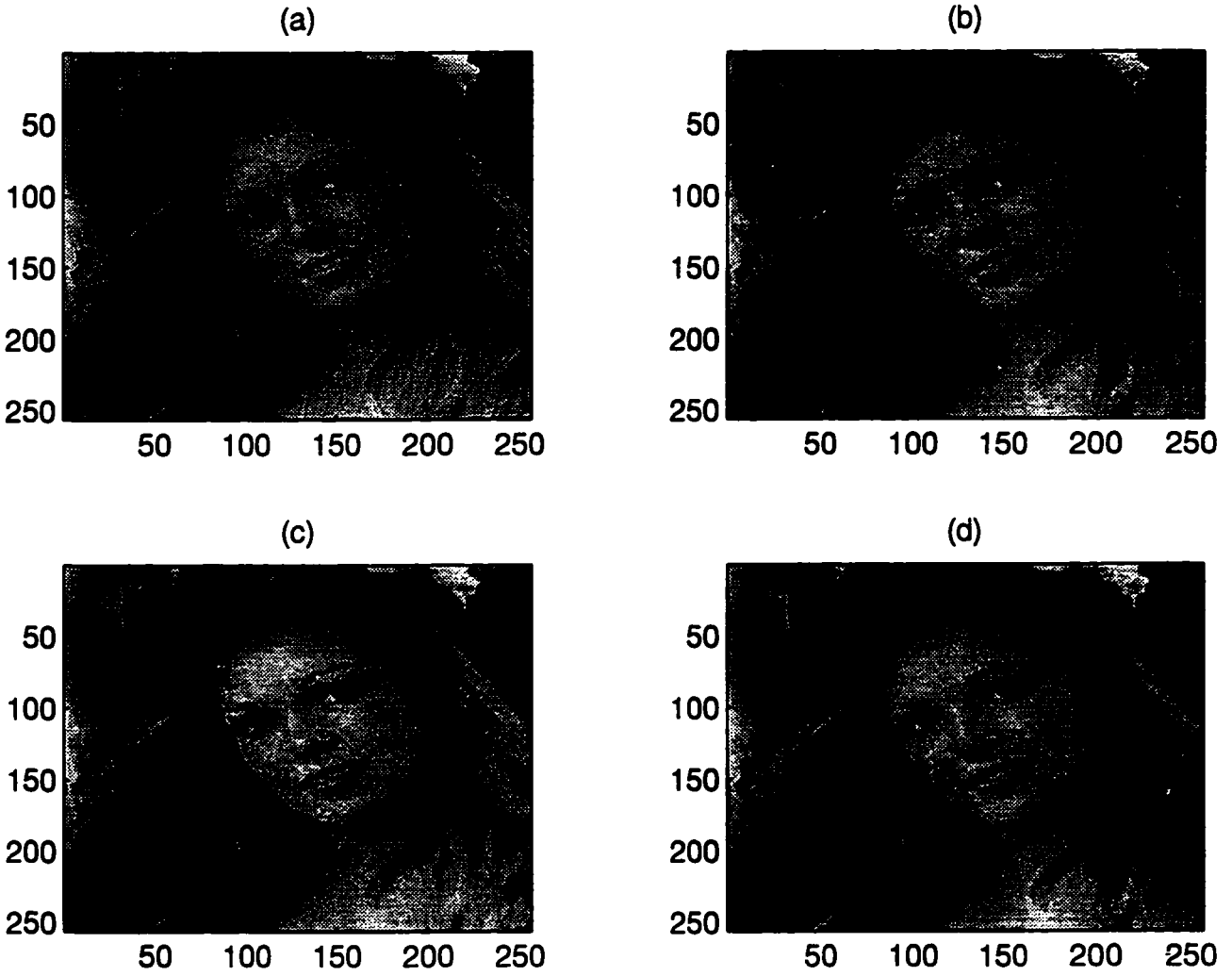


Figure 6.10: Comparisons of original and reconstructed signals with $M = 16$ and 0.4 bits per pixel. (a). original image, (b). DLS. (c). LOT (d). Sub-optimal lapped transform based on AR-model.

(e)

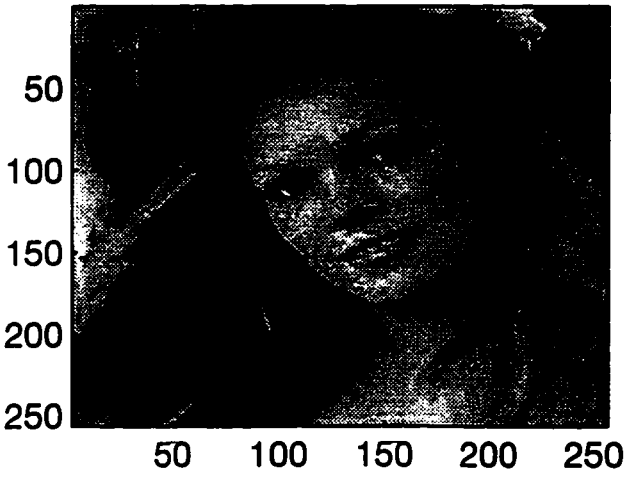


Figure 6.11: Continue ... Comparisons of original and reconstructed signals with $M = 16$ and 0.4 bits per pixel. (e). MLT

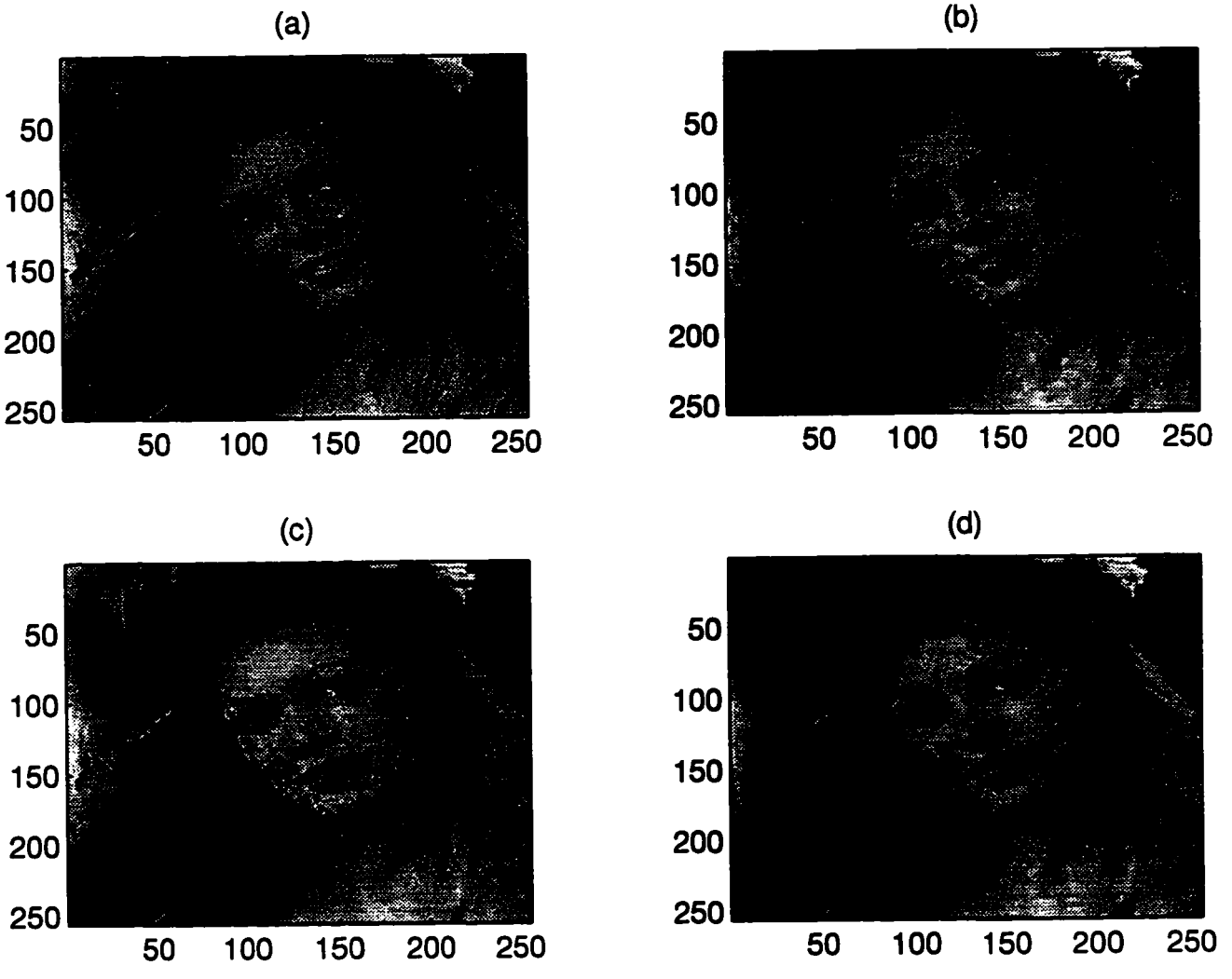


Figure 6.12: Comparisons of original and reconstructed signals with $M = 16$ and 0.24 bits per pixel. (a). original image, (b). DLS. (c). LOT (d). Sub-optimal lapped transform based on AR-model.

(e)

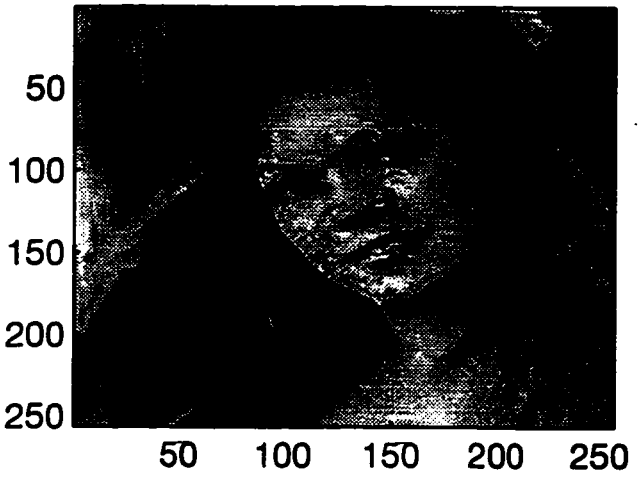


Figure 6.13: Continue ... Comparisons of original and reconstructed signals with $M = 16$ and 0.24 bits per pixel. (e). MLT

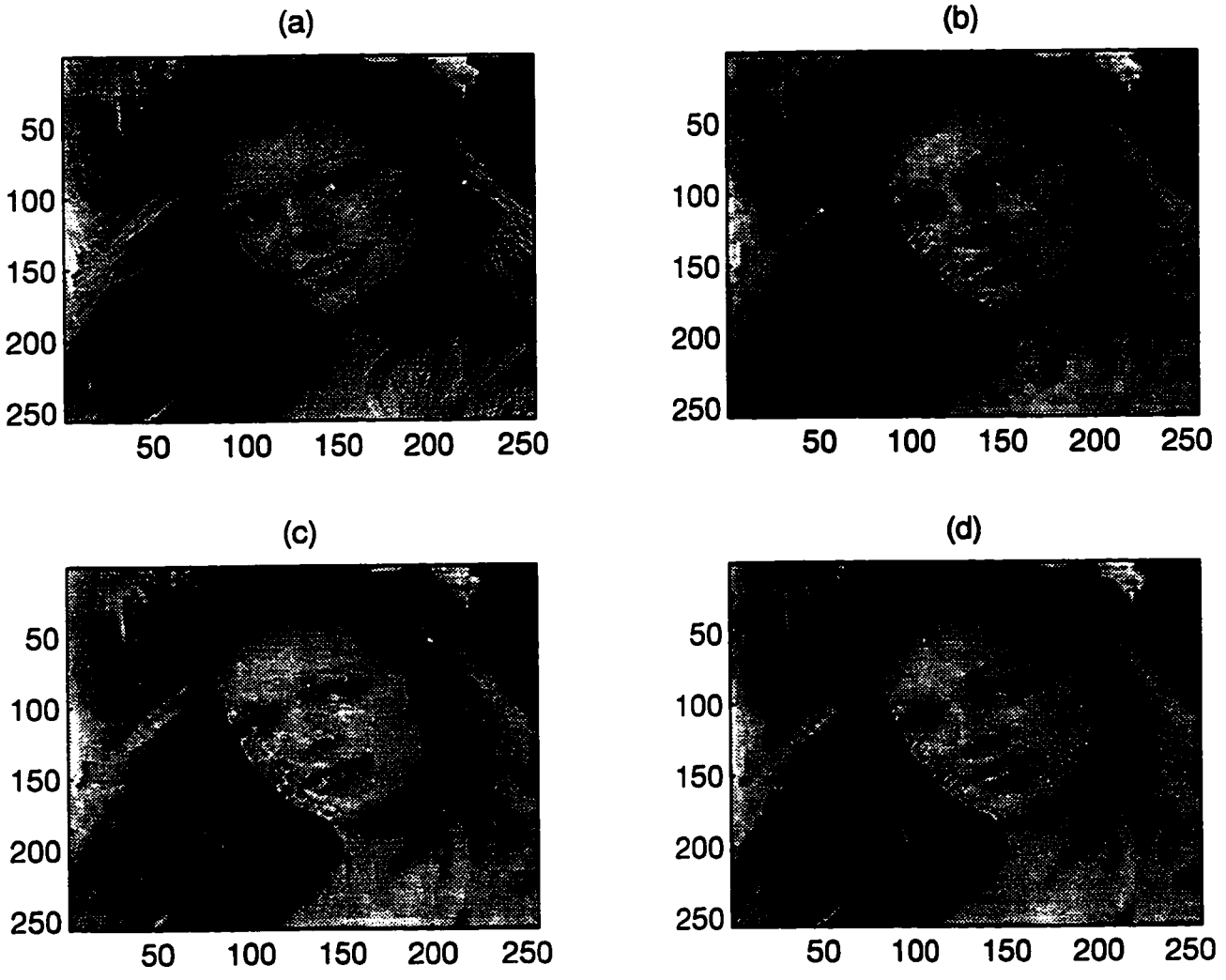


Figure 6.14: Comparisons of original and reconstructed signals with $M = 16$ and 0.16 bits per pixel. (a). original image, (b). DLS. (c). LOT (d). Sub-optimal lapped transform based on AR-model.

(e)

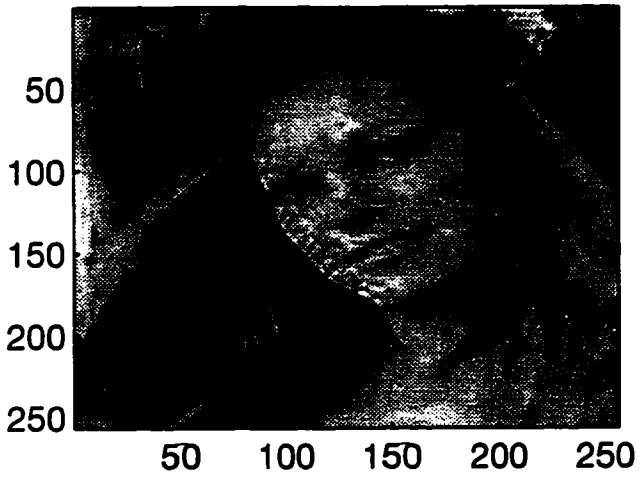


Figure 6.15: Continue ... Comparisons of original and reconstructed signals with $M = 16$ and 0.16 bits per pixel. (e). MLT

Chapter 7

Summary and Suggestions for Future Research

In this thesis, we examine the theory and properties of the lapped transforms based on the continuous DLS and DLC basis functions, and apply them to acoustic echo cancellation and image compression. Because of the smooth cutoff operation, DLS and DLC transforms have the potential to significantly reduce the blocking effect. Also, the performance analysis of DLS and DLC, such as variance distribution, coding gain and energy packing efficiency, are closer to ones of MLT, than ones of LOT. Even though MLT is slightly better than DLS in image compression, it does not perform better than DLS in echo cancellation. In this thesis, we have made the following contributions:

1. Operational Properties of DLS and DLC transforms

   A set of DLS and DLC transforms based on continuous local sine and cosine basis functions [17] is developed in Chapter 2. We examine the operational properties of the lapped transforms based on DLS and DLC basis functions in Chapter 3. DLS and DLC transforms which involve arbitrarily smooth cutoffs are shown to have an orthogonality property. We prove that the DLS and DLC belong to the family of lapped orthogonal transforms which satisfy the lapped orthogonality property. Despite the lack of unitarity for the transform matrix, the necessary and sufficient conditions for perfect reconstruction are derived. These properties allow for the design of the optimal lapped transforms in image and speech coding. Other properties, such as scale-in-time and shift-in-time properties, uniqueness property,

difference-in-time, and convolution properties are also examined. The performance analysis indicates that both the DLC and the DLS perform similarly.

2. Fast algorithm for computing DLS and DLC transforms

   By recognizing the particular structure of the transform matrix, we are able to significantly reduce the computational complexity required in the transformation. The computation reduction can be done by partitioning the transpose of the matrix into a set of sparse matrices. It is seen in Chapter 4 that the transform matrix can be represented by a cascade structure consisting of a set of Given's rotations and DST-IV matrix. The DST-IV matrix can also be factorized into a set of sparse matrices. The whole process allows us to partition the DLS transform matrix and implement it with a set of sparse matrices consisting of Given's rotations and butterfly operations. Such a fast algorithm for a general case of DLS transform yields $\frac{M}{2}(\log_2 M + 2) + 3(L-2)$ multiplications, and it is partially recursive. Also, this algorithm can easily be applied to computing the DLC transform. The number of computations required in DLC is the same as the number required in DLS. These algorithms make DLS and DLC suitable for parallel processing.

3. Lapped transforms in acoustic echo cancellation

   Based on the properties examined in Chapter 3 and the fast algorithms developed in Chapter 4, we apply the lapped transforms in the application of acoustic echo cancellation. Such a canceller can reduce the computation load with a subsampled signal. Compared to conventional methods, this scheme converges faster for tracking the changed echo path. Because of frequency aliasing, direct applications of the lapped transform in an echo canceller may not work well. To reduce frequency aliasing in the subsampling procedure, we propose an improved method by changing the subsampling rate in the block transforms. As a result, the echo residuals can be reduced significantly. Furthermore, we develop an optimal lapped transform that uses a criterion of maximum energy concentration in a given bandwidth. By this optimal lapped transform, an obvious improvement in the echo suppression is achieved in computer simulations. In comparison, the optimal lapped transform performs slightly worse than the optimal wavelet transform. However, it

is demonstrated in Chapter 5 that the optimal lapped transform involves much fewer computations and provides a much shorter processing delay than the optimal wavelet. Another advantage of using the lapped transform is that the sampling rate can be flexibly adjusted to achieve a desirable echo suppression, whereas the wavelet decomposition does not offer this kind of flexibility.

4. Lapped transforms in image compression

   For image compression, we use DLS, a smooth block transform. An important characteristic of such a transform is that it involves an arbitrarily smooth cutoff through a bell function. An optimal lapped transform which possesses the maximum coding gain is discussed. To satisfy the practical requirement — low computation load — we propose a special optimal lapped transform with the highest decorrelation of AR-model signal. Such an optimal transform is independent of the signal, and is sub-optimal for general image compression. In the simulation, the optimal lapped transform based on the AR-model is the most effective transform. It is shown in Chapter 6 that the DLS has a higher signal-to-noise ratio in image compression than either DCT or LOT. However, the results also shows that MLT is slightly better than DLS in image compression. The sub-optimal transform achieves the highest signal-to-noise ratio when compared with with DCT, LOT, MLT and DLS.

It is seen that our work in the lapped transforms based on DLS and DLC basis functions represents only a beginning; much work needs to be done. The following suggestions are for future research.

- Comparison with wavelet

  Both wavelet and lapped transform can be cast into a type of filter banks. Studying their relations can help us gain an insight into the design of an optimal discrete orthogonal transform for a particular application.

- Study of the fractional lapped transform

  In Chapter 5, it is shown that one advantage of lapped transform in echo cancellation is the flexibility of sampling rate variation. The echo suppression can be higher

when the sampling rate in lapped transform is reduced. An interesting question is how to develop a fractional lapped transform with arbitrary variation of the sampling rate so that, by considering both the echo suppression and computations, we may obtain an optimal sampling rate in the acoustic echo cancellation.

- Development of broad potential applications

  It is useful to find more applications of the lapped transforms based on the DLS and the DLC basis functions, particularly in the radar and microwave areas.

Investigations and development in the above areas should lead the lapped transforms — based on DLS and DLC — onto a new page. The lapped transforms will be widely applied in the areas which include signal processing, data compression and communications.

Appendix A

Solution of P_I, P_{II} and P_{III}

Rewrite (3.13), (3.14) and (3.15) as follows by letting $2k = r - t$

$$P_I = \sum_{n=0}^{M-1} \cos(r-t)\frac{\pi}{M}(n - \frac{L-1}{2}) \tag{A.1}$$

$$P_{II} = \sum_{n=0}^{L-1} [S_\varepsilon^2(n) - C_\varepsilon^2(n)] \cos[(r+t+1)\frac{\pi}{M}(n - \frac{L-1}{2})] \tag{A.2}$$

$$P_{III} = \sum_{n=L}^{M-1} \cos[(r+t+1)\frac{\pi}{M}(n - \frac{L-1}{2})]. \tag{A.3}$$

We will work on P_I, P_{II} and P_{III} separately.

(1). P_I : (A.1) can be written as

$$\begin{aligned} P_I &= \sum_{n=0}^{M-1} \cos[(r-t)\frac{\pi}{M}(n - \frac{L-1}{2})] \\ &= Re\left[\sum_{n=0}^{M-1} W_{2M}^{-(r-t)n} W_{2M}^{(r-t)(\frac{L-1}{2})}\right] \end{aligned}$$

where $W_{2M} = e^{-j\frac{\pi}{M}}$. Then, we have

$$P_I = Re\left[W_{2M}^{(r-t)(\frac{L-1}{2})} \cdot \frac{1 - W_{2M}^{-(r-t)M}}{1 - W_{2M}^{-(r-t)}}\right]$$

For $r - t = 2k$, $k = 0, 1, \cdots, \frac{M}{2}$, we have

$$P_I = \begin{cases} 0 & k \neq 0, \text{ i.e. } r \neq t \\ M & k = 0, \text{ i.e. } r = t. \end{cases} \tag{A.4}$$

which proves (3.16).

(2). P_{II} : Rewriting (A.2), we have

$$P_{II} = \sum_{n=0}^{L-1} [S_\varepsilon^2(n) - C_\varepsilon^2(n)] \cos[(r+t+1)\frac{\pi}{M}(n - \frac{L-1}{2})].$$

From the properties of $S_\varepsilon(n)$ and $C_\varepsilon(n)$, it is not hard to see that $[S_\varepsilon^2(n) - C_\varepsilon^2(n)]$ is anti-symmetric at $\frac{L}{2} - 1$ in $[0, L-1]$. Letting

$$f_{r,t}(n) = \cos[(r+t+1)\frac{\pi}{M}(n - \frac{L-1}{2})],$$

we have, when $(r-t)$ is an even number

$$\begin{aligned} f_{r,t}(L-1-n) &= \cos[(r+t+1)\frac{\pi}{M}(L-1-n-\frac{L-1}{2})] \\ &= \cos[(r+t+1)\frac{\pi}{M}(n-\frac{L-1}{2})] = f_{r,t}(n). \end{aligned} \tag{A.5}$$

It indicates that $f_{r,t}(n)$ is symmetric at $\frac{L}{2} - 1$ in $[0, L-1]$ when $(r-t)$ is an even number. Therefore, the inner product of P_{II} is anti-symmetric at $\frac{L}{2} - 1$ in $[0, L-1]$ when $(r-t)$ is even, and the summation is equal to zero; i.e.,

$$P_{II} = 0, \qquad \text{when } (r-t) \text{ is even}, \tag{A.6}$$

which proves (3.17).

(3). P_{III} : (A.3) can be written as

$$\begin{aligned} P_{III} &= \sum_{n=L}^{M-1} \cos[(r+t+1)\frac{\pi}{M}(n-\frac{L-1}{2})] \\ &= \sum_{n'=0}^{M-L-1} \cos[(r+t+1)\frac{\pi}{M}(n'+L-\frac{L-1}{2})], \quad (\text{let } n' = n-L) \\ &= \sum_{n=0}^{M-L-1} \cos[(r+t+1)\frac{\pi}{M}(n+\frac{L+1}{2})]. \end{aligned} \tag{A.7}$$

Let

$$g_{r,t}(n) = \cos[(r+t+1)\frac{\pi}{M}(n+\frac{L+1}{2})].$$

Then we have

$$g_{r,t}(M-L-1-n) = \cos[(r+t+1)\pi - (r+t+1)\frac{\pi}{M}(n+\frac{L+1}{2})].$$

Since $r - t = 2k$, we have $r + t + 1 = 2k + 2t + 1$ and thus

$$g_{r,t}(M - 1 - L - n) = -\cos[(r + t + 1)\frac{\pi}{M}(n + \frac{L+1}{2})] = -g_{r,t}(n).$$

Therefore, $g_{r,t}(n)$ is anti-symmetric at the centre of the interval $[0, M - L - 1]$ when $(r - t)$ is an even number and it satisfies

$$P_{III} = 0 \qquad \text{when } (r - t) \text{ is even,} \tag{A.8}$$

which proves (3.18).

Appendix B

Discussion of $< \psi_n, \psi_m >$

Considering (3.29), let

$$\begin{cases} T_I = Re \sum_{r=0}^{M-1} W_{2M}^{-\frac{n-m}{2}} W_{2M}^{-(n-m)r}, \\ T_{II} = Re \sum_{r=0}^{M-1} W_{2M}^{-\frac{n+m-L+1}{2}} W_{2M}^{-(n+m-L+1)r}. \end{cases} \tag{B.1}$$

It is obvious that the above equations are reduced to

$$\begin{cases} T_I = M, & \text{for } n = m, \\ T_{II} = M, & \text{for } n+m = 2k+L-1, k = 0 \text{ or } k = M. \end{cases} \tag{B.2}$$

For $m \neq n$, and $n + m \neq 2k + L - 1$, (B.1) is

$$\begin{cases} T_I = Re \left[W_{2M}^{-\frac{n-m}{2}} \frac{1-W_{2M}^{-(n-m)M}}{1-W_{2M}^{-(n-m)}} \right], \\ T_{II} = Re \left[W_{2M}^{-\frac{n+m-L+1}{2}} \frac{1-W_{2M}^{-(n+m-L+1)M}}{1-W_{2M}^{-(n+m-L+1)}} \right]. \end{cases} \tag{B.3}$$

By examining the numerators in (B.3), we find that the numerator in the expression of T_I is equal to zero if $(n-m)$ is even, and is equal to two if $(n-m)$ is odd. The numerator in T_{II} is equal to zeros if $(n+m-L+1)$ is even and is equal to two if $(n+m-L+1)$ is odd. For nonzero numerators, we manipulate T_I and T_{II} by multiplying $W_{2M}^{-\frac{n-m}{2}}$ and $W_{2M}^{-\frac{n+m-L+1}{2}}$ in the denominators, respectively. Then we have

$$\begin{cases} T_I = Re \frac{j}{\sin \frac{(n-m)\pi}{2M}} = 0, & \text{for } (n-m) \text{ odd}, \\ T_{II} = Re \frac{j}{\sin \frac{(n+m-L+1)\pi}{2M}} = 0, & \text{for } (n+m-L+1) \text{ odd}, \end{cases} \tag{B.4}$$

where j is the imaginary unit. Thus, considering all above results, we have,

$$T_I = \begin{cases} M & n = m \\ 0 & otherwise \end{cases} \tag{B.5}$$

$$T_{II} = \begin{cases} M & n+m = 2k+L-1, \quad k = 0 \text{ or } k = M \\ 0 & otherwise. \end{cases} \tag{B.6}$$

Combining (B.5) and (B.6), we have the following results.

1. When $n \neq m$ and $n+m \neq L-1$ and $n+m \neq 2M+L-1$, we have

$$T_I = 0,$$
$$T_{II} = 0,$$

which results in

$$< \psi_n, \psi_m >= 0.$$

2. When $n \neq m$ and $n+m = L-1$, we have

$$T_I = 0,$$
$$T_{II} = M,$$

resulting in

$$< \psi_n, \psi_m >= -b(n)b(m) = -b(n)b(L-1-n).$$

3. When $n \neq m$ and $n+m = 2M+L-1$, we have,

$$T_I = 0,$$
$$T_{II} = M,$$

resulting in

$$< \psi_n, \psi_m >= -b(n)b(m) = -b(n)b(2M+L-1-n).$$

4. When $n = m$ and $n+m = L-1$ or $n = m$ and $n+m = 2M+L-1$, we have

$$T_I = M,$$
$$T_{II} = M,$$

resulting in

$$< \psi_n, \psi_m >= 0.$$

5. When $n = m$ and $n + m \neq L - 1$ and $n + m \neq 2M + L - 1$, we have

$$T_I = M,$$
$$T_{II} = 0,$$

resulting in

$$< \psi_n, \psi_m >= b^2(n).$$

Our next problem is to determine the product of $b(n)$ and $b(m)$ for the cases 2, 3 and 5.

- For case 2, we have $n \neq m$ and $n+m = L-1$. This equation implies $0 \leq m \leq L-1$ and $0 \leq n \leq L-1$. Thus, we only need to consider the tail portion in Figure 3.1. We have

$$b(L-1-n) = \sin\left[\frac{(L-1-n)\pi}{2(L-1)} - \frac{1}{4}\sin\frac{2(L-1-n)\pi}{L-1}\right] = \cos[\frac{n\pi}{2(L-1)} - \frac{1}{4}\sin\frac{2n\pi}{L-1}],$$

and

$$b(n)b(L-1-n) = \frac{1}{2}\sin\left(\frac{n\pi}{L-1} - \frac{1}{2}\sin\frac{2n\pi}{L-1}\right).$$

- For case 3, we have $n \neq m$ and $n+m = 2M+L-1$, which indicates that $M \leq n, m$ because in the DLS transform $n, m \leq M + L - 1$. Thus, the bell function is in the region of head portion in Figure 3.1. Therefore, according to the third term in (2.46), we have

$$\begin{aligned} b(2M+L-1-n) &= \cos\left[\frac{(M+L-1-n)\pi}{2(L-1)} - \frac{1}{4}\sin\frac{2(M+L-1-n)\pi}{L-1}\right] \\ &= \sin\left[\frac{(n-M)\pi}{2(L-1)} - \frac{1}{4}\sin\frac{2(n-M)\pi}{L-1}\right]. \end{aligned}$$

Therefore,

$$b(n)b(2M+L-1-n) = \frac{1}{2}\sin\left(\frac{(n-M)\pi}{L-1} - \frac{1}{2}\sin\frac{2(n-M)\pi}{L-1}\right). \qquad \text{(B.7)}$$

- For case 5, we consider three situations. The first situation is $n = m \leq L-1$ and $n = m \neq (L-1)/2$. This means that n and m are both in the region of $[0, L-1]$. Then similar to case 2, the bell function is chosen in the range of the tail part, which corresponds to the sine function defined in the first term of (2.46). We have,

$$b^2(n) = \sin^2\left[\frac{n\pi}{2(L-1)} - \frac{1}{4}\sin\frac{2n\pi}{L-1}\right] = \frac{1}{2}\left[1 - \cos\left(\frac{n\pi}{L-1} - \frac{1}{2}\sin\frac{2n\pi}{L-1}\right)\right].$$

The second situation is for $M < n = m < M + L - 1$ and $n = m \neq 2M + L - 1$. Now, both n and m are in the region of $[M, M + L - 1]$. $b(n)$ corresponds to cosine function which is defined in the third term of (2.46), and we have

$$b^2(n) = \cos^2 \left[\frac{(n-M)\pi}{2(L-1)} - \frac{1}{4} \sin \frac{2(n-M)\pi}{L-1} \right] = \frac{1}{2} \left[1 + \cos \left(\frac{(n-M)\pi}{L-1} - \frac{1}{2} \sin \frac{2(n-M)\pi}{L-1} \right) \right]. \tag{B.8}$$

For the third situation when $L - 1 < n = m < M$, we have $2(L-1) < n+m < 2M$ and $b^2(n) = 1$.

Appendix C

Proof of Lemma 3.2

We prove Lemma 3.2 as follows.

Proof: Consider the $2M \times 2M$ matrix,

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{Q} & \boldsymbol{O} \\ \boldsymbol{B} & \boldsymbol{A} \\ \boldsymbol{O} & \boldsymbol{Q} \\ \boldsymbol{A} & \boldsymbol{B} \end{bmatrix},$$

where $\boldsymbol{O}$ is an $(M-L) \times M$ zero matrix. Then from the conditions (3.46) to (3.50), we have

$$\begin{aligned} \boldsymbol{R}\boldsymbol{R}^T &= \begin{bmatrix} \boldsymbol{Q} & \boldsymbol{O} \\ \boldsymbol{B} & \boldsymbol{A} \\ \boldsymbol{O} & \boldsymbol{Q} \\ \boldsymbol{A} & \boldsymbol{B} \end{bmatrix} \begin{bmatrix} \boldsymbol{Q}^T & \boldsymbol{B}^T & \boldsymbol{O}^T & \boldsymbol{A}^T \\ \boldsymbol{O}^T & \boldsymbol{A}^T & \boldsymbol{Q}^T & \boldsymbol{B}^T \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{Q}\boldsymbol{Q}^T & \boldsymbol{Q}\boldsymbol{B}^T & \boldsymbol{O}_{M-L} & \boldsymbol{Q}\boldsymbol{A}^T \\ \boldsymbol{B}\boldsymbol{Q}^T & \boldsymbol{A}\boldsymbol{A}^T + \boldsymbol{B}\boldsymbol{B}^T & \boldsymbol{A}\boldsymbol{Q}^T & \boldsymbol{B}\boldsymbol{A}^T + \boldsymbol{A}\boldsymbol{B}^T \\ \boldsymbol{O}_{M-L} & \boldsymbol{Q}\boldsymbol{A}^T & \boldsymbol{Q}\boldsymbol{Q}^T & \boldsymbol{Q}\boldsymbol{B}^T \\ \boldsymbol{A}\boldsymbol{Q}^T & \boldsymbol{A}\boldsymbol{B}^T + \boldsymbol{B}\boldsymbol{A}^T & \boldsymbol{B}\boldsymbol{Q}^T & \boldsymbol{A}\boldsymbol{A}^T + \boldsymbol{B}\boldsymbol{B}^T \end{bmatrix} = \boldsymbol{I}_{2M}. \end{aligned}$$

Because $\boldsymbol{R}$ is a square matrix, we have $\boldsymbol{R}^T = \boldsymbol{R}^{-1}$ and $\boldsymbol{R}\boldsymbol{R}^T = \boldsymbol{R}^T\boldsymbol{R} = \boldsymbol{I}_{2M}$. Now, let

us check the results of $\boldsymbol{R}^T\boldsymbol{R}$.

$$\begin{aligned}\boldsymbol{R}^T\boldsymbol{R} &= \begin{bmatrix} \boldsymbol{Q}^T & \boldsymbol{B}^T & \boldsymbol{O}^T & \boldsymbol{A}^T \\ \boldsymbol{O}^T & \boldsymbol{A}^T & \boldsymbol{Q}^T & \boldsymbol{B}^T \end{bmatrix} \begin{bmatrix} \boldsymbol{Q} & \boldsymbol{O} \\ \boldsymbol{B} & \boldsymbol{A} \\ \boldsymbol{O} & \boldsymbol{Q} \\ \boldsymbol{A} & \boldsymbol{B} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{Q}^T\boldsymbol{Q} + \boldsymbol{B}^T\boldsymbol{B} + \boldsymbol{A}^T\boldsymbol{A} & \boldsymbol{B}^T\boldsymbol{A} + \boldsymbol{A}^T\boldsymbol{B} \\ \boldsymbol{B}^T\boldsymbol{A} + \boldsymbol{A}^T\boldsymbol{B} & \boldsymbol{Q}^T\boldsymbol{Q} + \boldsymbol{B}^T\boldsymbol{B} + \boldsymbol{A}^T\boldsymbol{A} \end{bmatrix} = \boldsymbol{I}_{2M}.\end{aligned}$$

It is straightforward that

$$\boldsymbol{A}^T\boldsymbol{A} + \boldsymbol{B}^T\boldsymbol{B} + \boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{I}_M \tag{C.1}$$

$$\boldsymbol{A}^T\boldsymbol{B} + \boldsymbol{B}^T\boldsymbol{A} = \boldsymbol{O}_M. \tag{C.2}$$

Now the lemma is proven. □

Appendix D

Proof of Lemma 3.3

We prove Lemma 3.3 as follows.

Proof: From the first condition, we have

$$\boldsymbol{A}^T\boldsymbol{B} + \boldsymbol{B}^T\boldsymbol{A} = \boldsymbol{O}_M.$$

Pre-multiplying $\boldsymbol{B}$ at both sides of above formula, we obtain

$$\boldsymbol{B}\boldsymbol{A}^T\boldsymbol{B} + \boldsymbol{B}\boldsymbol{B}^T\boldsymbol{A} = \boldsymbol{O}_{L\times M}.$$

According to the second assumption, it is known that the first term of the above equation is a zero matrix. Thus we have,

$$\boldsymbol{B}\boldsymbol{B}^T\boldsymbol{A} = \boldsymbol{O}_{L\times M}.$$

Pre-multiplying $\boldsymbol{A}^T$ in the above equation, the above equation becomes

$$\boldsymbol{A}^T\boldsymbol{B}\boldsymbol{B}^T\boldsymbol{A} = \boldsymbol{O}_M, \tag{D.1}$$

from which, we can prove that

$$\boldsymbol{A}^T\boldsymbol{B} = \boldsymbol{B}^T\boldsymbol{A} = \boldsymbol{O}_M. \tag{D.2}$$

□

Appendix E

Proof of Theorem 3.3

Now with Lemmas 3.2 and 3.3, we are ready to prove Theorem 3.3.

Proof: Similar as Theorem 3.2, the necessary condition is the direct result of Theorem 3.1. To prove the sufficient condition, we may use the same procedures as in Theorem 3.2, Lemma 3.2, and Lemma 3.3. Rewrite $\tilde{\boldsymbol{\Phi}}$ as follows,

$$\tilde{\boldsymbol{\Phi}} = \begin{bmatrix} \boldsymbol{\Phi}^{(h)} & \boldsymbol{\Phi}^{(t)} & \boldsymbol{O}_{L\times M} \\ \boldsymbol{O}_{(M-L)\times M} & \boldsymbol{\Phi}^{(m)} & \boldsymbol{O}_{(M-L)\times M} \\ \boldsymbol{O}_{L\times M} & \boldsymbol{\Phi}^{(h)} & \boldsymbol{\Phi}^{(t)} \end{bmatrix}.$$

From $\tilde{\boldsymbol{\Phi}}\tilde{\boldsymbol{\Phi}}^T = \boldsymbol{I}_{M+L}$, we can obtain the conditions which are exactly the same as (3.46) to (3.50) in Lemma 3.2. Applying the results in Lemma 3.2, we obtain

$$[\boldsymbol{\Phi}^{(t)}]^T\boldsymbol{\Phi}^{(t)} + [\boldsymbol{\Phi}^{(m)}]^T\boldsymbol{\Phi}^{(m)} + [\boldsymbol{\Phi}^{(h)}]^T\boldsymbol{\Phi}^{(h)} = \boldsymbol{I}_M \tag{E.1}$$

$$[\boldsymbol{\Phi}^{(t)}]^T\boldsymbol{\Phi}^{(h)} + [\boldsymbol{\Phi}^{(h)}]^T\boldsymbol{\Phi}^{(t)} = \boldsymbol{O}_M. \tag{E.2}$$

Furthermore, we may apply Lemma 3.3 into (E.1) and obtain

$$[\boldsymbol{\Phi}^{(t)}]^T\boldsymbol{\Phi}^{(h)} = [\boldsymbol{\Phi}^{(h)}]^T\boldsymbol{\Phi}^{(t)} = \boldsymbol{O}_M. \tag{E.3}$$

(E.1) indicates the orthogonality property: i.e., $\boldsymbol{\Phi}^T\boldsymbol{\Phi} = \boldsymbol{I}_M$ and (E.3) shows the lapped orthogonality, i.e., $\boldsymbol{\Phi}^T\boldsymbol{W}\boldsymbol{\Phi} = \boldsymbol{O}_M$. Therefore, we have

$$\tilde{\boldsymbol{\Phi}}\tilde{\boldsymbol{\Phi}}^T = \boldsymbol{I}_{M+L}$$

if and only if $\boldsymbol{\Phi}^T\boldsymbol{\Phi} = \boldsymbol{I}_M$ and $\boldsymbol{\Phi}^T\boldsymbol{W}\boldsymbol{\Phi} = \boldsymbol{O}_M$. □

Bibliography

[1] N. Ahmed, T. Natarajan, and K.R. Rao, "Discrete cosine transform," IEEE Trans. on Comput., Vol.23, No.1, pp. 90-94, 1974.

[2] N. Ahmed and K.R. Rao, "Orthogonal transforms for digital signal processing," New York, NY: Springer, 1975.

[3] A.N. Akansu, R.A. Haddad and H. Caglar, "The binomial QMF-wavelet transform for multiresolution signal decomposition," IEEE Trans. Signal Processing, Vol.41, No.1, pp.13-19, 1993.

[4] G.B. Anderson and T.S. Huang, "Piecewise Fouier transform for picture bandwidth compression," IEEE Trans. on Commun. Technol. Vol.19, pp.133-140, 1971.

[5] H.C. Andrews, "Multidimensional rotation in feature selection," IEEE Trans. on Comput. Vol.20, No.9, pp. 1045-1051, 1971.

[6] P. Auscher, G.Weiss and M.V. Wickerhauser, "Local sine and cosine bases of Coifman and Meyer and the construction of smooth wavelets," section III in the book "Wavelets: a tutorial in theory and applications", edited by C.K. Cui, Academic Press, Inc., 1992.

[7] D.M. Baylon and J.S. Lim, "Transform/subband analysis and synthesis of signals," MIT Research Laboratory of Electronics Technical Report, June, 1990.

[8] A. Benallal and A. Gilloire, "Improvement of the tracking capability of the numerically stable FLS algorithms for adaptive filtering," Proc. of ICASSP, Vol.II, pp. 1031-1034, Glasgow, Scotland, May, 1989.

[9] R.E. Blahut, "Fast algorithms for digital signal processing," Addison-Wesley, 1985.

[10] P. Cassereau, "A new class of optimal unitary transform for image processing," Master's thesis, Mass. Inst. Tech., Cambridge, MA, 1985.

[11] A. Chan and C. Chui, "Applications of Spline-Wavelet Packets to Echo Cancellation and Other Signal Processing Problems," Canadian Conference on Electrical and Computer Engineering, pp. MM4.2.1-4.2.4, Toronto, Canada, 1992.

[12] N. Chelemal-D and K.R. Rao, "Fast computational algorithms for the discrete cosine transform," Proc. of 9th Asilomar Conf. on circuit, System and Computation, Pacific Grove, CA. November. 1985.

[13] J. Chen, H. Bes, J. Vàndewalle and P. Janssens, "A New Structure for Sub-band Acoustic Echo Canceller," Proc. of ICASSP, Vol.V, pp. 2574-2577, New York, April, 1988.

[14] W.H. Chen, C.H. Smith and S.C. Fralick, "A fast computational algorithm for the discrete cosine transform," IEEE Trans. on Commun. Technol. Vol.25, No.9, pp. 1004-1008, 1977.

[15] N.I. Cho and S.U. Lee, "Fast algorithm and implementation of 2-D discrete cosine transform," IEEE Trans. Circuits. Syst., Vol.38, No.3, pp. 297-305, 1991.

[16] R.J. Clarke, "Transform coding of images," London: Academic Press, 1985.

[17] R.R. Coifman and Y. Meyer, "Remarques sur l'analyse de Fourier à fenêtre, série I, C. R. Acad. Sci. Paris 312, 259-261, 1991.

[18] R.V. Cox and R.E. Crochiere, "Real-time simulation of adaptive transform coding," IEEE Trans. Acoust., Speech, Signal Processing, Vol.29, No.4, pp. 147-154, 1981.

[19] R.E. Crochiere and L.R. Rabiner, "Multirate digital signal processing," Prentice-Hall Inc. Englewood Cliffs, New Jersey, 1983.

[20] J.C. Darragh, "Subband and transform coding of images," Ph.D. Thesis Department of Electrical Engineering, UCLA, pp. 72-75, 1989.

[21] S.A. Dyer, N. Ahmed and D.R. Hummels, "Classification of vectorcardiograms using Walsh and cosine orthogonal transforms," IEEE Trans. Electromag., Compata., Vol.27, No.2, pp. 35-40, 1985.

[22] S.A. Dyer, N. Ahmed and D.R. Hummels, "vectorcardiographic data compression via Walsh and cosine transforms," IEEE Trans. Electromag., Compata., Vol.27, No. 2, pp. 24-34, 1985.

[23] E. Feig, "A fast scaled-DCT algorithm," Proc. SPIE Image Processing Algorithms and Techniques, 1244, pp. 2-13, 1990.

[24] A. Gilloire, "Experiments with Sub-band Echo Cancellers for Teleconferencing," Proc. of ICASSP, Vol.IV, pp. 2141-2144, Dallas, April, 1987.

[25] A. Gupta and K.R. Rao, "A fast recursive algorithm for the discrete sine transform," IEEE Trans. Acoust., Speech, Signal Processing, Vol.38, No.3, pp. 553-557, 1990.

[26] A. Gupta and K.R. Rao, "An efficient FFT algorithm based on the discrete sine transform," IEEE Trans. Acoust., Speech, Signal Processing, Vol.39, No.2, pp. 486-490, 1991.

[27] A. Habibi and P.A. Wintz, "Image coding by linear transformation and block quantization," IEEE Trans. Commun. Tech., Vol.19, No.1, pp. 50-63, 1977.

[28] S. Haykin, "Adaptive filter theory," Prentice-Hall, Inc. 1991.

[29] M. Hamidi and J. Pearl, "Comparison of the Cosine and Fourier Transforms of Markov-1 Signals", IEEE Trans. on Acoust., Speech, Signal Processing, Vol.24, No.10, pp. 428-429, 1976.

[30] R.M. Haralick, "A Storage Efficient Way to Implement the Discrete Cosine Transform", IEEE Trans. Comput., Vol.27, No.10, pp. 966-968, 1978.

[31] B. Hatty, "Recursive Least Squares Algorithms Using Multirate Systems for Cancellation of Acoustical Echoes," Proc. of ICASSP, Vol.II, pp. 1145-1148, New Mexico, April, 1990.

[32] H.S. Hou, "A Fast Recursive Algorithm for Computing the Discrete Cosince Transform", IEEE Trans. Acoust., Speech, Signal Processing, Vol.35, No.10, pp. 1455-1461, 1987.

[33] Y.S. Hsu, S. Prum, J.H. Kagel and H.C. Andrews, "Pattern recognition experiments in the M andala/cosine domain," IEEE Trans. Pattern Anal. Machine Intell, Vol.5, No.9, pp. 512-520, 1983.

[34] A.K. Jain, "Fundamentals of Digital Image Processing," Prentice-Hall, Englewood Cliffs, NJ, 1989.

[35] N.S. Jayant, "Digital coding of waveforms," Prentice-Hall, Englewood Cliffs, NJ, 1984.

[36] Q. Jin, K.M. Wong and Z.Q. Luo, "Design of an optimum wavelet for cancellation of long echoes in telephony," Proceeding of IEEE SP International Symposium on Time-Frequency and Time-Scale Analysis, Philadephia, pp. 488-491, October, 1994.

[37] Q. Jin, Z.Q. Luo and K.M. Wong, "Optimal filter banks for signal decompostion and its application in adaptive echo cancellation, " IEEE Trans. on Signal Processing, Vol.44, No.7, pp.1669-1680, July, 1996.

[38] H. Kitajima, "Energy packing efficiency of the Hadamard transform, " IEEE Trans. on Commun. Vol.24, pp.1256-1258, Nov. 1976.

[39] W. Kellermann, "Kompensation Akustischer Echos in Frequenzteilbandern," frequenz, vol. 39, No. 7/8, pp209-215, 1985.

[40] Jr. Kelly et al. "Self-adaptive Echo Canceller," U.S.patent, No.3,500,000, March 10, 1970.

[41] A.V. Oppenheim and R.W. Schafer, "Digital signal processing," Pretice-Hall International, Inc., London, 1975.

[42] R.P. Koilpillai and P.P. Vaidayanathan, "New results on cosine-modulated FIR filter banks satisfying perfect reconstruction," Proc. of ICASSP, Vol.III, pp.1793-1796, Toronto, Canada, May, 1991.

[43] R.P. Koilpillai and P.P. Vaidayanathan, "Cosine-modulated FIR filter banks satisfying perfect reconstruction," IEEE Trans. on Signal Processing, Vol.40, No.4, pp. 770-783, 1992.

[44] B.G. Lee, Kashef and A. Habibi, "Direct Computation of High- Order DCT Coefficients from Lower-Order DCT Coefficients", Proc. of SPIE 28th Annu. Int. Tech. Symp., San Diego, CA, August, 1984.

[45] B.G. Lee, "A new algorithm to compute the discrete cosine transform," IEEE Trans. Acoust., Speech, Signal Processing, Vol.32, No.6, pp. 1243-1245, 1984.

[46] J. Li and P.C. Yip, "The Given's rotation and sparse factorization of the DCT-II matrix," CRL report # 225, 1990.

[47] C. Loeffler, A. Lightenberg and G.S., Moschytz, "Practical Fast 1-D DCT Algorithm with 11 Multiplications", Proc. of ICASSP, Vol.II, pp. 988-991, Glasgow, Scotland, May, 1989.

[48] S. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," IEEE Trans. Patt. Anal. and Mach. Intell. Vol.11, pp. 674-693, 1989.

[49] H.S. Malvar, "Signal processing with lapped transforms," Artech House, Inc. , Norwood, MA, 1992.

[50] H.S. Malvar, "Lapped transforms for efficient transform/subband coding," IEEE Trans. Acoust. Speech, Signal Processing, Vol.38, No.6, pp. 969-978, 1990.

[51] H.S. Malvar and D.H. Staelin, "Reduction of blocking effects in image coding with a lapped orthogonal transform," Proc. of ICASSP, Vol.II, pp. 781-784, New York, April, 1988.

[52] H.S. Malvar, "Modulated QMF filter banks with perfect reconstruction," Electronics Letters, Vol. 26, pp.906-907, June, 1990.

[53] H.S. Malvar and D.H. Staelin, "The LOT : transform coding without blocking effects," IEEE Trans. Acoust. Speech, Signal Processing, Vol.37, No.4, pp.553-559, 1989.

[54] H.S. Malvar, "Extended lapped transforms : fast algorithms and application," Proc. of ICASSP, Vol.III, pp. 1797-1800, Toronto, Canada, May, 1991.

[55] G.H. Golub and C.F. Vān Loan, "Matrix Computation", Second Edition, The Johns Hopkins University Press, 1989.

[56] F.A. Graybill, "Matrices with Applications in Statistics", Second Edition, Wadsworth Publishing Company, Inc., 1983.

[57] S.H. Friedberg, A.J. Insel and L.E. Spence, " Linear Algebra", Second Edition, Prentice-Hall International Editions, 1989.

[58] O. R. Mitchell, E.J. Delp and S.G. Carlton, "Block truncation coding: a new approach to image compression," Proc. of ICC, pp. 12B.1.1 - 12B.1.4, 1978.

[59] K. Murano, S. Unagami and F. Amano, "Echo cancellation and application," IEEE Communication Magazine, pp. 49-55, January, 1990.

[60] M.J. Narashimha and A.M. Peterson, "On the computation of the discrete cosine transform", IEEE Trans. Commun., Vol.26, No.6, pp. 934- 936, 1987.

[61] J. Pearl, H.C. Andrews and W.K. Pratt, "Performance measures for transform data coding", IEEE Trans. Commun. Technol. Vol.20, No.6, pp.411-415, 1972.

[62] W.K. Pratt, J. Kane and H.C. Andrews, "Hadamard transform coding of images", IBID, Vol.57, pp. 58-68, 1969.

[63] W.K. Pratt, W. Chen and L.R. Weich, "Slant transform image coding", IBID, Vol.22, pp. 1075-1093, 1974.

[64] W.K. Pratt, "Digital image processing," Wiley-Interscience, New York, 1978.

[65] M. Rabbani and P.W. Hones, "Digital image compression techniques," SPIE Optical Engineering Press, Vol.TT7, 1991.

[66] K.R. Rao and P. Yip, "Discrete cosine transform — Algorithms, Advantages, Applications," Academic Press, Inc. 1990.

[67] R.L. deQueiroz and K.R. Rao, "Time-varying lapped transforms and wavelet packets," IEEE Trans. on Signal Processing, Vol.41, No.12, pp. 3293-3305, 1993.

[68] A.K. Soman and P.P. Vaidayanathan, "Paraunitary filter banks and wavelet packets," Proc. of ICASSP, Vol.IV, pp.397-400, San Francisco, CA, 1992.

[69] M. Sondi, "An adaptive echo canceller," BSTJ, March, 1967.

[70] R. Tolimieri, C. Lu and M. An, "Algorithms for discrete Fourier transform and convolutions," Springer-Verlag, 1989.

[71] P.P. Vaidayanathan, "Multirate digital filters, filter banks, polyphase networks and applications: A tutorial," Proc. IEEE, Vol.78, pp.56-93, 1990.

[72] P.P. Vaidayanathan, "Multirate systems and filter banks," PTR Prentice-Hall, Inc. Englewood Cliffs, New Jersey, 1993.

[73] P.P. Vaidayanathan, "Quadrature mirror filter banks, M-band extension and perfect-reconstruction techniques," IEEE ASSP Magazine, Vol.4, No.3, pp. 4-20, 1987.

[74] M. Vetterli, "Multirate filter banks for subband coding," Chapter 2, in the book of "Subband image coding," edited by J.W. Woods, Kluwer Academic Publishers, 1991.

[75] M. Vetterli, "Fast 2-D discrete cosine transform," Proc. of ICASSP, pp. 1538-1541, 1985.

[76] M. Vetterli and D.Le Gall, "Perfect reconstruction FIR filter banks: some properties and factorizations," IEEE Trans. Acoust. Speech, Signal Processing, Vol.37, No.7, pp. 1057-1071, 1989.

[77] M. Vetterli and D.Le Gall, "Perfect reconstruction FIR filter banks: theory and design," IEEE Trans. Acoust. Speech, Signal Processing, Vol.40, No.9, pp.2207-2232, 1992.

[78] M. Vetterli, "Fast 2-D discrete cosine transform," Proc. of ICASSP, pp.1538-1541, 1985.

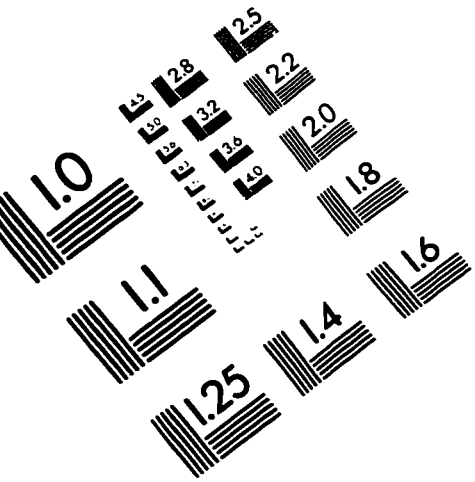
[79] Z. Wang, "Fast algorithm for the discrete W transform and for the discrete Fourier transform", IEEE Trans. Acoust., Speech, Signal Processing, vol.32, No.8, pp. 19-24, 1984.

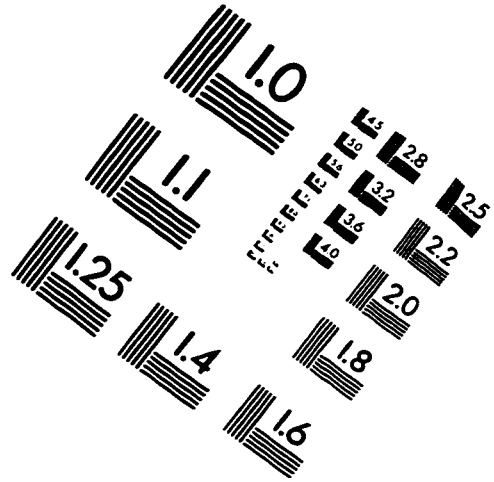
[80] M.V. Wickerhauser, "Acoustic signal compression with wavelet packets," Section VII in book "Wavelets: a tutorial in theory and applications," edited by C.K. Chui, Academic Press, Inc., 1992.

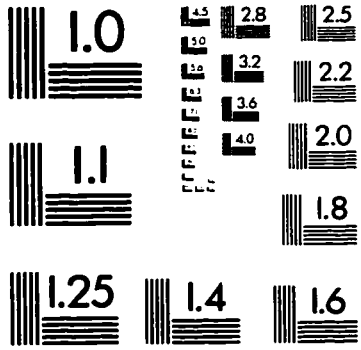
[81] S. Winograd, "On computing the discrete Fourier transform," Math Computation, Vol.32, No.141, pp. 175-199, Jan. 1978.

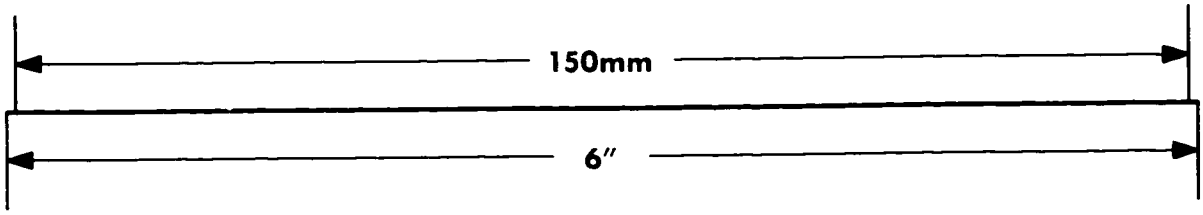
[82] J.W. Woods and S.D. O'Neil, "Subband coding of images," IEEE Trans. Acous. Speech, Signal Processing, Vol.34, No.5, pp. 1278-1288, 1986.

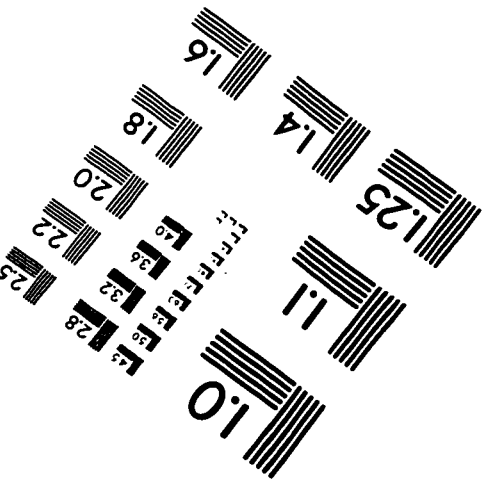
IMAGE EVALUATION
TEST TARGET (QA-3)

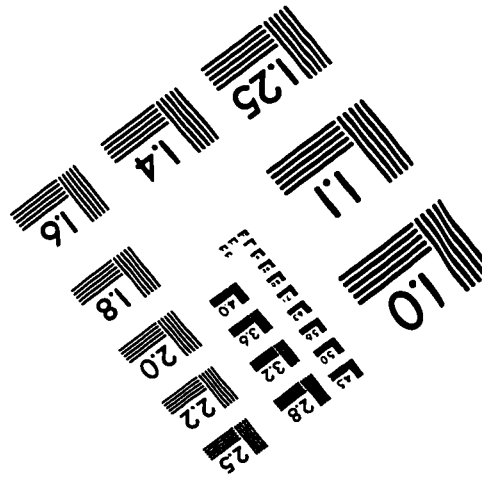












APPLIED IMAGE . Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989