

THE UNIVERSITY OF CALGARY

MULTIDIMENSIONAL CONTINUOUS-DISCRETE FREQUENCY

DOMAIN DIGITAL FILTERING

by

Ashfaq Choudhury

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF ELECTRICAL ENGINEERING

CALGARY, ALBERTA

April, 1989

© Ashfaq Choudhury 1989



National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

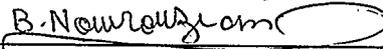
ISBN 0-315-54196-2

THE UNIVERSITY OF CALGARY
FACULTY OF GRADUATE STUDIES

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies for acceptance, a thesis entitled, "Multidimensional Continuous-Discrete Frequency Domain Digital Filtering", submitted by Ashfaq Choudhury in partial fulfillment of the requirements for the degree of Master of Science.



Supervisor, Dr. L.T. Bruton
Department of Electrical Engineering



Dr. B. Nowrouzian
Department of Electrical Engineering



Dr. R.M. Rangayyan
Department of Electrical Engineering



Dr. R.A. Stein
Department of Electrical Engineering



Dr. G.D. Lodwick
Department of Surveying Engineering

Date 28 June 89

ABSTRACT

A new multidimensional (MD) digital filtering technique is described in this thesis. The continuous-discrete frequency domain (CDFD) filtering technique employs continuous-frequency (e.g. infinite impulse response) filtering in N of the M dimensions ($N < M$) and discrete-frequency (e.g. discrete Fourier transform) filtering in the remaining $M-N$ dimensions. Notation is developed to describe CDFD filtering concisely. The advantages of MD CDFD filters over MD continuous-frequency and MD discrete-frequency filters are demonstrated in various signal processing examples and applications.

Some topics for further research are proposed, especially in the use of CDFD filters in three dimensions and in real-time MD filtering applications.

ACKNOWLEDGEMENT

I would like to thank Mr. Norm Bartley for his advice and helpful suggestions about this thesis. His patience in ploughing through innumerable equations is greatly appreciated. I am grateful to Ms. Ella Lok for typing this thesis (in spite of all the equations).

Most of all, I wish to thank my supervisor, Dr. L.T. Bruton for his advice, patience and understanding.

This research was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC), and through The University of Calgary research assistantships.

To my parents

TABLE OF CONTENTS

APPROVAL PAGE	ii
ABSTRACT	iii
ACKNOWLEDGEMENTS	iv
DEDICATION	v
TABLE OF CONTENTS	vi
LIST OF TABLES	x
LIST OF FIGURES	xi
1. SOME BASIC CONCEPTS IN MULTIDIMENSIONAL DIGITAL FILTERING	1
1.1. Introduction	1
1.2. MD Signals and Systems	2
1.2.1. MD Signals	3
1.2.2. Multidimensional Discrete Fourier Transform and Inverse Discrete Fourier Transform	7
1.2.3. MD Linear Shift Invariant Filters	9
1.2.3.1. Definition of a MD LSI Filter	9
1.2.3.2. Impulse Response of a MD LSI Filter	10
1.2.3.3. BIBO Stability of a MD LSI Filter	11

1.2.3.4. Frequency Response of a MD LSI Filter	11
1.2.3.5. Z-transform Transfer Function of a MD LSI Filter	12
1.3. MD Discrete-Frequency Filtering	14
1.4. MD Continuous-Frequency Filtering	20
1.4.1. MD Non-Recursive (FIR) Filtering	23
1.4.2. MD Recursive (IIR) Filtering	25
1.5. MD CDFD Filtering	29
1.6. Scope and Objective of Thesis	34
 2. NOTATION FOR CDFD FILTERING	 36
2.1. Introduction	36
2.2. Vector Representation of Frequency and Spatial Variables in MD signals	36
2.3. Alternative Array Representations of Scalar Functions in M Variables	40
2.4. Element-Wise Array Operations	45
2.5. The MD DFT Operator F []	50
2.6. The Partial MD DFT'	54
 3. CDFD FILTERING AND APPLICATIONS	 60
3.1. Introduction	60

3.2. Calculation of the Output Sequence $y(k)$ for CDFD Filters	60
3.2.1. Example: The 3D Case with $N = 1$	66
3.3. Steady-State Frequency Response of CDFD Filters	70
3.4. Design of CDFD Filters	77
3.4.1. Design Considerations	77
3.4.2. Design Procedure for MD CDFD Filters	82
3.4.3. Design Examples of CDFD Filters	83
3.4.3.1. CDFD Approximation of a 3D Cone-Stop Filter - A Design Example	83
3.4.3.2. CDFD Approximation of a 2D Circularly Symmetric Lowpass Filter - A Design Example	86
3.4.3.3. An Application of CDFD Filters in Seismic Image Processing - A Design Example	92
3.5. Comparison of CDFD Filters with IIR and DFT Filters in Seismic Signal Processing Applications	96
4. AN IMAGE PROCESSOR USING CDFD FILTERS	104
4.1. Introduction	104
4.2. Software Image Processing System	104
4.3. Proposed Scheme for Hardware Implementation of	

Image Processing System	111
4.3.1. Control Program	112
4.3.2. Memory	113
4.3.3. Processing Speed	116
5. AREAS FOR FURTHER RESEARCH	119
5.1. Introduction	119
5.2. Factors Affecting Real-Time CDFD Filters in Two and Three Dimensions	119
5.3. CDFD Filtering Using the Discrete Hartley Transform (DHT)	121
5.3.1. The Discrete Hartley Transform	121
5.3.2. MD CDFD Filtering Using the DHT Instead of the DFT on Real Signals	122
5.4. Conclusion	124
REFERENCES	125
APPENDIX A: COMPUTATION AND MEMORY REQUIREMENTS FOR 2D CDFD, IIR AND DFT FILTERS	126

LIST OF TABLES

Table 3.1. Comparison of computational efficiency and memory usage for CFDF, IIR and 2D-DFT filtering of seismic images.	98
--	----

LIST OF FIGURES

Fig. 1.1. A one dimensional continuous signal.	5
Fig. 1.2. A two dimensional continuous signal.	5
Fig. 1.3. Graphical representation of a 2D discrete signal.	5
Fig. 1.4. Graphical representation of a 2D mixed continuous-discrete signal.	6
Fig. 1.5. A rectangular region of support.	6
Fig. 1.6. Block diagram of z-transform transfer function.	13
Fig. 1.7. Frequency response of a 2D DFT circularly symmetric lowpass filter.	15
Fig. 1.8. Input and output masks for a 2D recursive filter.	27
Fig. 1.9. Frequency Response of 3D cone filter.	33
Fig. 3.1. Block diagram of 3D CDFD filter.	67
Fig. 3.2. Periodic extension of frequency response of 3D CDFD and DFT cone filters.	76
Fig. 3.3. Frequency response of circularly symmetric lowpass 2D CDFD filter.	79
Fig. 3.4. Symmetry in passband of circularly symmetric 2D CDFD lowpass filter.	81
Fig. 3.5. Design of CDFD 3D cone-stop filter.	85

Fig. 3.6. Design of CDFD 2D lowpass filter.	88
Fig. 3.7. Reduction of edge effects using CDFD filtering.	91
Fig. 3.8. Design of CDFD 2D fanstop filter.	93
Fig. 3.9. Low velocity ground roll removal from seismic images using a 2D CDFD fanstop filter.	97
Fig. 3.10. Number of additions necessary for ground roll removal from seismic images for 2D DFT, CDFD and IIR filters, varying with the number of temporal samples per trace.	99
Fig. 3.11. Number of multiplications necessary for ground roll removal from seismic images for 2D DFT, CDFD and IIR filters, varying with the number of temporal samples per trace.	100
Fig. 3.12. Number of additions necessary for ground roll removal from seismic images for 2D DFT, CDFD and IIR filters, varying with the number of traces.	102
Fig. 3.13. Number of multiplications necessary for ground roll removal from seismic images for 2D DFT, CDFD and IIR filters, varying with the number of traces.	103
Fig. 4.1. Output of software interactive image processing system.	106
Fig. 4.2. Arbitrary passband shapes of 2D CDFD filters obtained using image processing system.	110
Fig. 4.3. Memory diagram.	114
Fig. 4.4. Lookup table for coefficients required for 1D IIR bandpass	

filters.	115
---------------	-----

CHAPTER 1

SOME BASIC CONCEPTS IN MULTIDIMENSIONAL DIGITAL FILTERING

1.1. Introduction

Due to recent advances in high-speed computers and dedicated signal processing hardware, multidimensional (MD) digital filtering is being used in a wide variety of applications. MD digital filters are predominantly used to solve 2D and 3D signal processing problems.

Two-dimensional digital filters are used extensively in image processing applications such as biomedical image analysis and seismic signal processing. Typical biomedical image processing applications include the enhancement of CAT (Computer-Aided Tomography) scan and X-ray images [1]. In the field of seismic image processing, two-dimensional digital fan filtering techniques are used to selectively enhance and reject signals on the basis of their velocity, such as the removal of low velocity ground roll interference from seismic images [2].

Applications in three-dimensional signal processing are to be found in areas such as 3D seismic image processing and situations involving two dimensional time-varying images; for example, sonar, radar and television [3]. Three-dimensional frequency planar filters are used to track objects in time-varying radar or sonar images [4].

An exciting potential application of three-dimensional digital filtering is in the rapidly developing field of videophone technology. This involves transmitting and receiving video signals over currently-existing telephone networks which are subject to heavy interference.

There are two widely followed and, so far, distinct approaches to MD digital filtering. Firstly, *discrete frequency* methods, such as the MD discrete Fourier transform (MD DFT), have been employed using techniques that are essentially extensions of the 1D case [3]. Secondly, MD *continuous frequency* methods, such as finite impulse response (FIR) and infinite impulse response (IIR) techniques, have been employed [3].

Both these techniques have certain advantages and disadvantages [3]. The Continuous-Discrete Frequency Domain (CDFD) filtering technique presented in this thesis combines both continuous and discrete frequency filtering into a novel filtering technique that possesses some of the advantages of both continuous and discrete frequency filtering.

Some important fundamental concepts in MD digital filtering are presented in the next section.

1.2. MD Signals and Systems

This section covers some fundamental concepts in MD digital filtering. In addition to covering some basic theory on MD signals and systems, this section also defines the MD DFT and the MD IDFT (inverse discrete Fourier transform) -

two widely used transforms in MD digital filtering.

1.2.1. MD Signals

A signal is simply a means of conveying information; for example, a television picture, music or a sequence of numbers in the memory of a computer. Filters extract useful information and reject undesirable information from *input signals* to produce filtered *output signals*. For example, filters in a stereo system may be designed to remove unwanted high-frequency noise from music or remove D.C. or low-frequency signals to avoid damage to speaker systems.

A 1D signal (Fig. 1.1) is a function of *one* independent variable. For example, the sine wave $x(t)$ in Fig. 1.1 is a function of the independent variable t . The 2D sine wave $x(t_1, t_2) = \sin(\omega_1 t_1 + \omega_2 t_2)$ in Fig. 1.2 is a function of *two* independent variables t_1 and t_2 . Similarly, an MD signal $x(t_1, t_2, \dots, t_M)$ is a function of the M independent variables t_1, t_2, \dots, t_M . Thus, a multidimensional signal is any signal that can be expressed as a function of M independent variables ($M \geq 2$).

There are three types of MD signals - continuous, discrete and mixed continuous *and* discrete. A *continuous* MD signal can be expressed as a function of M independent variables where each variable is defined over a continuum of values; for example, the intensity $I(x, y)$ of a black and white photograph. The 2D sinusoidal signal $x(t_1, t_2)$ in Fig. 1.2 is also a continuous 2D signal.

A *discrete* MD signal is expressed as a function that is defined only on a set of points; for example, the intensity $I(k_1, k_2)$ of a digitized image. The 2D signal $x(k_1, k_2)$ in Fig. 1.3 is a discrete 2D signal. Discrete signals are often obtained from continuous signals by rectangular sampling. In the case of a digitized image, for example, $I(k_1, k_2)$ is obtained from $I(x, y)$ by periodic rectangular sampling. That is,

$$I(k_1, k_2) = I(x, y) \Big|_{\substack{x = k_1 T_1 \\ y = k_2 T_2}} \quad (1.1)$$

where T_1 and T_2 are the sampling periods in the x and y directions respectively.

A *mixed* continuous-discrete MD signal is expressed as a function of both discrete variables and continuous variables; for example, a seismic image where a fixed number of geophones at discrete locations measure waveforms that are continuous functions of time. The 2D signal $x(k_1, t_2)$ in Fig. 1.4 is continuous in t_2 and discrete in k_1 .

A *finite-extent* MD sequence is one which takes the value zero outside a finite region of support. The region of support can have any shape although typically it is rectangular. Thus, for a 2D finite-extent signal, the region of support is usually a rectangle (Fig. 1.5).

This thesis deals primarily with finite-extent MD discrete signals of the form $x(k_1, k_2, \dots, k_M)$, that have a region of support defined by $0 \leq k_i < L_i$;

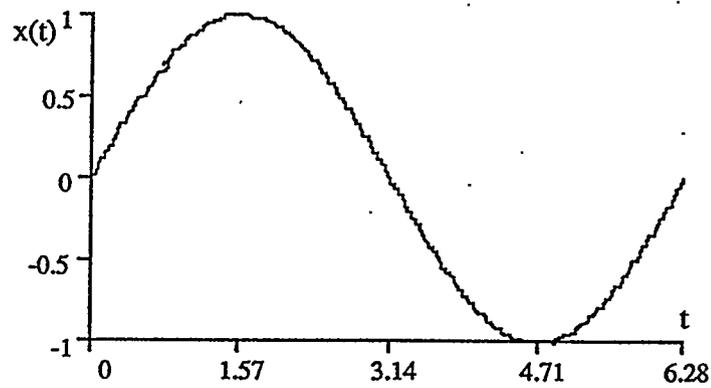


Fig. 1.1. A one-dimensional continuous signal $x(t) = \sin(t)$

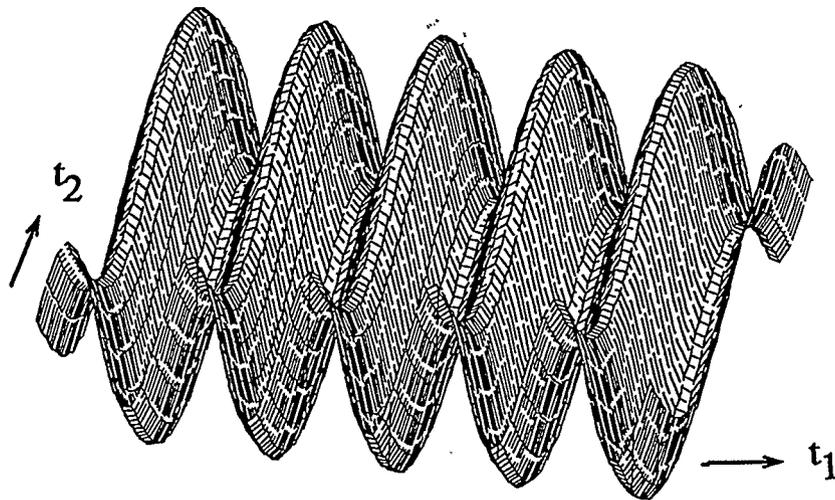


Fig. 1.2. A two-dimensional continuous signal $x(t_1, t_2) = \sin(\omega_1 t_1 + \omega_2 t_2)$

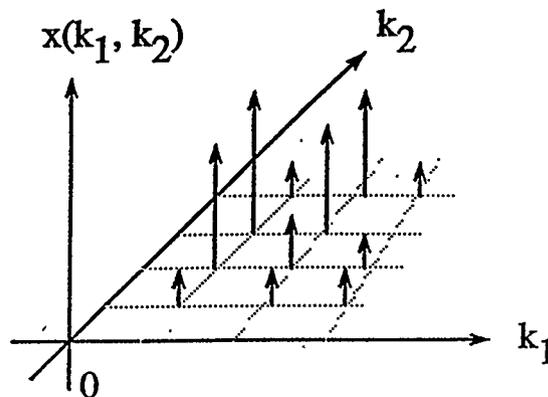


Fig. 1.3. Graphical representation of a 2D discrete signal $x(k_1, k_2)$.

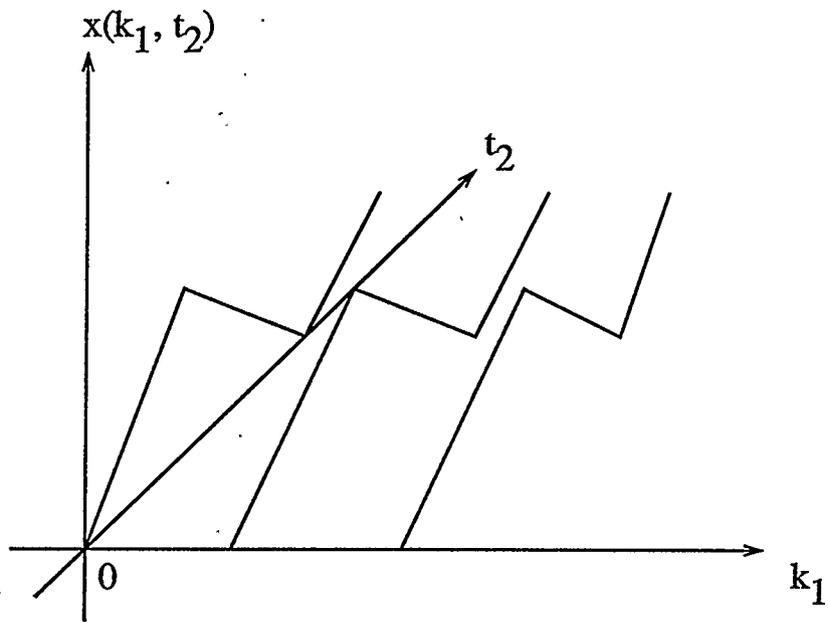


Fig. 1.4. Graphical representation of a 2D mixed continuous-discrete signal $x(k_1, t_2)$.

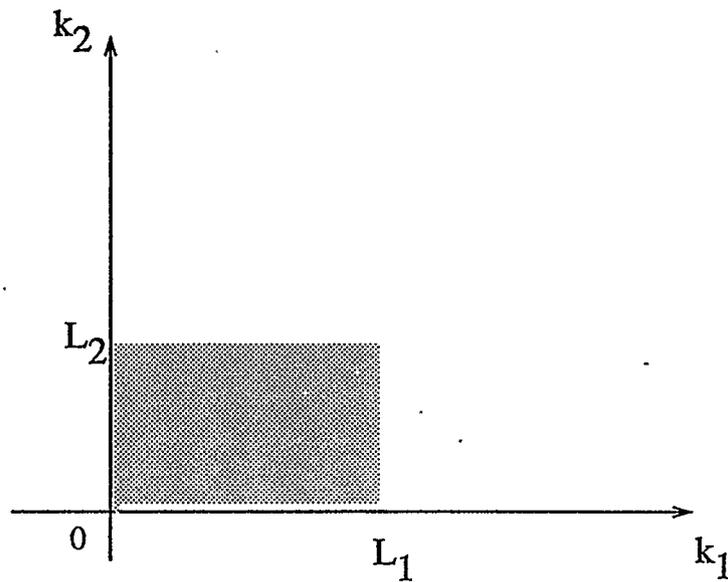


Fig. 1.5. A rectangular region of support (shaded area) for a 2D discrete signal (represented by dots) which is zero outside the region of support, $(0 \leq k_1 < L_1, 0 \leq k_2 < L_2)$.

k_i, L_i integer, $1 < L_i < \infty$ and $i = 1, 2, \dots, M$. Henceforth, unless otherwise specified, all MD finite-extent signals will be assumed to have a region of support as described above.

1.2.2. Multidimensional Discrete Fourier Transform and Inverse Discrete Fourier Transform

It is often convenient to represent and analyze temporal and spatial domain MD discrete finite-extent signals $x(k_1, k_2, \dots, k_M)$, henceforth referred to as MD spatial domain signals, in terms of their MD discrete frequency components. The MD Discrete Fourier Transform (MD DFT) transforms any MD spatial domain signal $x(k_1, k_2, \dots, k_M)$ into a frequency domain MD discrete finite-extent signal $x(\Omega_1, \Omega_2, \dots, \Omega_M)$, henceforth referred to as an MD frequency domain signal, where $\Omega_1, \Omega_2, \dots, \Omega_M$ are the discrete frequency variables corresponding to the variables k_1, k_2, \dots, k_M , and $0 < \Omega_i < L_i$.

The MD Discrete Fourier Transform $X(\Omega_1, \Omega_2, \dots, \Omega_M)$ of an MD spatial domain signal $x(k_1, k_2, \dots, k_M)$ is defined as

$$X(\Omega_1, \Omega_2, \dots, \Omega_M) \triangleq \sum_{k_1=0}^{L_1-1} \sum_{k_2=0}^{L_2-1} \dots \sum_{k_M=0}^{L_M-1} x(k_1, k_2, \dots, k_M) \cdot \exp \left[-j2\pi \left[\frac{k_1 \Omega_1}{L_1} + \frac{k_2 \Omega_2}{L_2} + \dots + \frac{k_M \Omega_M}{L_M} \right] \right] \quad (1.2)$$

where $0 \leq k_i \leq L_i - 1$, $0 \leq \Omega_i \leq L_i - 1$ and $1 < L_i < \infty$ for $i = 1, 2, \dots, M$.

For example, the 2D DFT of a 2D spatial domain signal $x(k_1, k_2)$ is given by

$$X(\Omega_1, \Omega_2) \triangleq \sum_{k_1=0}^{L_1-1} \sum_{k_2=0}^{L_2-1} x(k_1, k_2) e^{-\frac{j2\pi k_1 \Omega_1}{L_1}} e^{-\frac{j2\pi k_2 \Omega_2}{L_2}} \quad (1.3)$$

Just as the MD DFT transforms an MD spatial domain signal into an MD frequency domain signal, the multidimensional Inverse Discrete Fourier Transform (MD IDFT) transforms an MD frequency domain signal into an MD spatial domain signal. The MD IDFT is defined by

$$\begin{aligned} & x(k_1, k_2, \dots, k_M) \\ & \triangleq \frac{1}{L_1 L_2 \dots L_M} \sum_{\Omega_1=0}^{L_1-1} \sum_{\Omega_2=0}^{L_2-1} \dots \sum_{\Omega_M=0}^{L_M-1} X(\Omega_1, \Omega_2, \dots, \Omega_M) \cdot \\ & \exp \left[j2\pi \left[\frac{k_1 \Omega_1}{L_1} + \frac{k_2 \Omega_2}{L_2} + \dots + \frac{k_M \Omega_M}{L_M} \right] \right] \end{aligned} \quad (1.4)$$

In the 2D case given by (1.3), this becomes

$$x(k_1, k_2) \triangleq \frac{1}{L_1 L_2} \sum_{\Omega_1=0}^{L_1-1} \sum_{\Omega_2=0}^{L_2-1} X(\Omega_1, \Omega_2) e^{\frac{j2\pi k_1 \Omega_1}{L_1}} e^{\frac{j2\pi k_2 \Omega_2}{L_2}} \quad (1.5)$$

An important application of the MD DFT is in deriving the frequency responses of MD linear shift invariant digital filters which are described in the following

section.

1.2.3. MD Linear Shift Invariant Filters

Linear shift invariant (LSI) filters are the most widely used digital filters because they are comparatively easy to design and analyze, and sufficiently powerful to solve a variety of digital filtering problems.

1.2.3.1. Definition of an MD LSI Filter

Consider an MD discrete-domain filtering operator $P[\]$ that maps a set of MD discrete input signals $\{x_i(k_1, k_2, \dots, k_M)\}$ into a set of MD discrete output signals $\{y_i(k_1, k_2, \dots, k_M)\}$. This operation can be written as

$$P[\{x_i(k_1, k_2, \dots, k_M)\}] \triangleq \{y_i(k_1, k_2, \dots, k_M)\} \quad (1.6)$$

The filtering operator $P[\]$ is *linear* if and only if

$$P[\{\sum_i c_i x_i(k_1, k_2, \dots, k_M)\}] = \sum_i c_i \{y_i(k_1, k_2, \dots, k_M)\} \quad (1.7)$$

for any input $x_i(k_1, k_2, \dots, k_M)$ and any scalar constant c_i .

$P[\]$ is said to be *shift invariant* if and only if

$$P[\{x_i(k_1 - K_1, k_2 - K_2, \dots, k_M - K_M)\}] = \{y_i(k_1 - K_1, \\ k_2 - K_2, \dots, k_M - K_M)\} \quad (1.8)$$

where K_1, K_2, \dots, K_M are any constants.

The linearity condition implies that the response to a sum of scaled inputs is simply the sum of the scaled responses due to the inputs applied separately, while the shift invariance condition implies that a shift in the input produces a corresponding shift in the output. A filter satisfying both the linearity and shift invariance conditions is said to be a *linear shift invariant* (LSI) filter.

1.2.3.2. Impulse Response of an MD LSI Filter

The *impulse response* of a LSI filter, characterized by the filtering operator $P[]$, is defined as

$$h(k_1, k_2, \dots, k_M) \triangleq P[\delta(k_1, k_2, \dots, k_M)] \quad (1.9)$$

where $\delta(k_1, k_2, \dots, k_M)$ is the *discrete unit impulse* defined to be

$$\delta(k_1, k_2, \dots, k_M) = \begin{cases} 1, & k_1 = k_2 = \dots = k_M = 0 \\ 0, & \text{otherwise} \end{cases} \quad (1.10)$$

The output $y(k_1, k_2, \dots, k_M)$ of a MD LSI filter is calculated from the input $x(k_1, k_2, \dots, k_M)$ and the impulse response $h(k_1, k_2, \dots, k_M)$ using the MD convolution summation defined as

$$\begin{aligned}
& y(k_1, k_2, \dots, k_M) \\
& \triangleq \sum_{j_1=0}^{L_1-1} \sum_{j_2=0}^{L_2-1} \dots \sum_{j_M=0}^{L_M-1} h(k_1, k_2, \dots, k_M) x(k_1 - j_1, k_2 - j_2, \dots, k_M - j_M)
\end{aligned} \tag{1.11}$$

where each $j_i, i = 1, 2, \dots, M$, is an integer.

1.2.3.3. BIBO Stability of an MD LSI Filter

A MD LSI filter is BIBO (Bounded Input Bounded Output) stable if and only if, for any bounded input sequence, the output sequence remains bounded. A necessary and sufficient condition for an MD LSI filter to be BIBO stable is that its impulse response $h(k_1, k_2, \dots, k_M)$ be absolutely summable. That is

$$\sum_{k_1=0}^{L_1-1} \sum_{k_2=0}^{L_2-1} \dots \sum_{k_M=0}^{L_M-1} |h(k_1, k_2, \dots, k_M)| = S_1 < \infty \tag{1.12}$$

In general, it is considerably more difficult to test for MD stability than for 1D stability [3].

1.2.3.4. Frequency Response of an MD LSI Filter

The *steady state frequency response* $H(\Omega_1, \Omega_2, \dots, \Omega_M)$ of an MD LSI filter is the ratio of the response $\tilde{y}(k_1, k_2, \dots, k_M)$ to the input $\tilde{x}(k_1, k_2, \dots, k_M)$ when the input is an exponential of the form

$$\tilde{x}(k_1, k_2, \dots, k_M) = \exp \left[j2\pi \left[\sum_{i=1}^M \frac{k_i \Omega_i}{L_i} \right] \right]$$

and Ω_i, L_i are integer for all i . The "˜" above the x and y implies that these signals are periodically extended in all M dimensions with period L_i in the i th dimension [3].

The frequency response of an MD LSI filter can be obtained by applying the MD DFT to the impulse response of the filter and is given by

$$H(\Omega_1, \Omega_2, \dots, \Omega_M) \triangleq \sum_{k_1=0}^{L_1-1} \sum_{k_2=0}^{L_2-1} \dots \sum_{k_M=0}^{L_M-1} h(k_1, k_2, \dots, k_M) \cdot \exp \left[-j2\pi \left[\frac{k_1 \Omega_1}{L_1} + \frac{k_2 \Omega_2}{L_2} + \dots + \frac{k_M \Omega_M}{L_M} \right] \right] \quad (1.13)$$

The MD DFTs of the input and output sequences, $X(\Omega_1, \Omega_2, \dots, \Omega_M)$ and $Y(\Omega_1, \Omega_2, \dots, \Omega_M)$, respectively, are related through the frequency response $H(\Omega_1, \Omega_2, \dots, \Omega_M)$ as follows :

$$Y(\Omega_1, \Omega_2, \dots, \Omega_M) = H(\Omega_1, \Omega_2, \dots, \Omega_M) X(\Omega_1, \Omega_2, \dots, \Omega_M) \quad (1.14)$$

1.2.3.5. Z-transform Transfer Function of an MD LSI Filter

The z-transform transfer function $T(z_1, z_2, \dots, z_M)$ of an MD LSI filter is shown as a block diagram in Fig. 1.6. $T(z_1, z_2, \dots, z_M)$ relates the MD z-transforms of the input and output signals, $X(z_1, z_2, \dots, z_M)$ and $Y(z_1, z_2, \dots, z_M)$, respectively, in accordance with

$$T(z_1, z_2, \dots, z_M) = \frac{Y(z_1, z_2, \dots, z_M)}{X(z_1, z_2, \dots, z_M)} \quad (1.15)$$

where the z-transforms of the input and output signals are obtained using the definition of the MD z-transform :

$$X(z_1, z_2, \dots, z_M) \triangleq \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} \dots \sum_{k_M=-\infty}^{\infty} x(k_1, k_2, \dots, k_M) z_1^{-k_1} z_2^{-k_2} \dots z_M^{-k_M} \quad (1.16)$$

For LSI filters, (1.15) can also be represented as a ratio of polynomials so that

$$T(z_1, z_2, \dots, z_M) \triangleq \frac{N(z_1, z_2, \dots, z_M)}{D(z_1, z_2, \dots, z_M)} \triangleq \frac{\sum_{j_1=0}^{m_1} \sum_{j_2=0}^{m_2} \dots \sum_{j_M=0}^{m_M} n(j_1, j_2, \dots, j_M) z_1^{j_1} z_2^{j_2} \dots z_M^{j_M}}{\sum_{j_1=0}^{n_1} \sum_{j_2=0}^{n_2} \dots \sum_{j_M=0}^{n_M} d(j_1, j_2, \dots, j_M) z_1^{j_1} z_2^{j_2} \dots z_M^{j_M}} \quad (1.17)$$

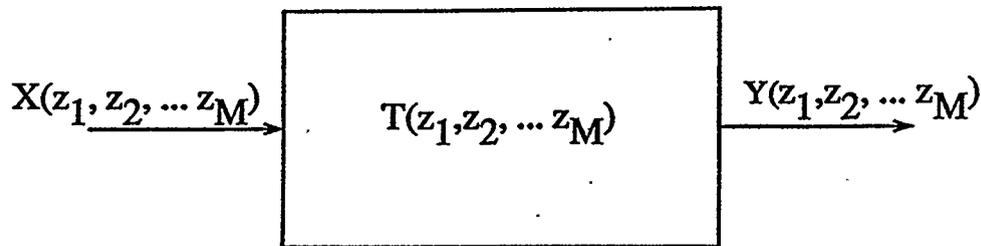


Fig. 1.6. Block diagram of z-transform transfer function, $T(z_1, z_2, \dots, z_M)$, of an MD LSI filter.

where $z_i = e^{j\omega_i}$ and $m_i \leq n_i$ for $i = 1, 2, \dots, M$. Each ω_i in (1.17) is a *continuous-frequency* variable and the complex variable z_i is the unit advance operator in the i^{th} direction.

The substitution $z_i = e^{j\omega_i}$ in (1.17) can be used to express $T(z_1, z_2, \dots, z_M)$ as a function of the *continuous-frequency* variables $\omega_1, \omega_2, \dots, \omega_M$. That is,

$$T'(\omega_1, \omega_2, \dots, \omega_M) = T(z_1, z_2, \dots, z_M) \Big|_{\substack{z_i = e^{j\omega_i} \\ i=1, 2, \dots, M}}$$

When the transfer function of a LSI filter can be expressed as a function of continuous-frequency variables, it is said to be a *continuous-frequency domain* filter.

Continuous and discrete frequency filtering are discussed in the next two sections.

1.3. MD Discrete-Frequency Filtering

An MD discrete-frequency filter has a frequency response that is a function of M *discrete* frequency variables $\Omega_1, \Omega_2, \dots, \Omega_M$. For example, consider the 2D discrete-frequency response $H(\Omega_1, \Omega_2)$ shown in Fig. 1.7 which is defined by

$$|H(\Omega_1, \Omega_2)| = \begin{cases} 1 & ; \Omega_1^2 + \Omega_2^2 \leq R^2 \\ 0 & ; \textit{elsewhere} \end{cases} \quad (1.18)$$

$H(\Omega_1, \Omega_2)$ takes the value 1 at all discrete frequency points (Ω_1, Ω_2) on or inside the shaded region on the $\Omega_1\Omega_2$ plane, and it takes the values 0 at all other discrete frequency points.

A MD discrete-frequency filter, such as the one shown in Fig. 1.7, can be implemented using MD DFT filtering which is a three step filtering process.

The first step in MD DFT filtering is to apply the MD DFT to a MD spatial domain signal $x(k_1, k_2, \dots, k_M)$ to obtain the MD frequency domain input signal $X(\Omega_1, \Omega_2, \dots, \Omega_M)$. That is

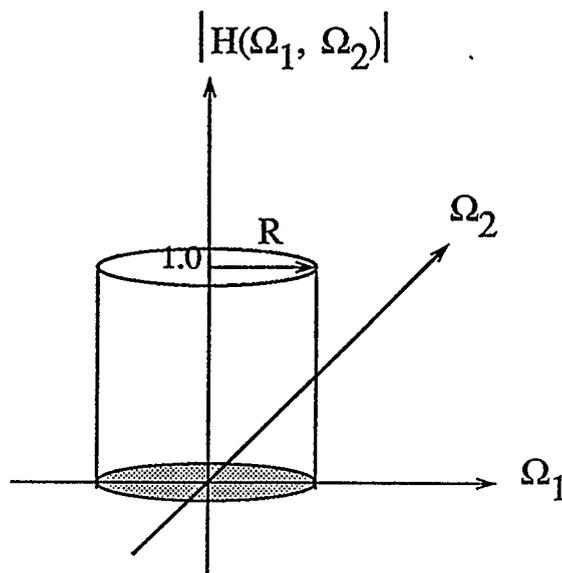


Fig. 1.7. Frequency response of a 2D DFT circularly symmetric lowpass filter.

$$\begin{aligned}
X(\Omega_1, \Omega_2, \dots, \Omega_M) &= \sum_{k_1=0}^{L_1-1} \sum_{k_2=0}^{L_2-1} \dots \sum_{k_M=0}^{L_M-1} x(k_1, k_2, \dots, k_M) \cdot \\
&\quad \exp \left[-j2\pi \left[\frac{k_1 \Omega_1}{L_1} + \frac{k_2 \Omega_2}{L_2} + \dots + \frac{k_M \Omega_M}{L_M} \right] \right] \quad (1.19)
\end{aligned}$$

The MD frequency domain output signal $Y(\Omega_1, \Omega_2, \dots, \Omega_M)$ is now obtained by multiplying $X(\Omega_1, \Omega_2, \dots, \Omega_M)$ by the MD frequency response of the filter $H(\Omega_1, \Omega_2, \dots, \Omega_M)$, so that

$$Y(\Omega_1, \Omega_2, \dots, \Omega_M) = H(\Omega_1, \Omega_2, \dots, \Omega_M) X(\Omega_1, \Omega_2, \dots, \Omega_M) \quad (1.20)$$

This constitutes the second step in the filtering process. The sequence $X(\Omega_1, \Omega_2, \dots, \Omega_M)$ is usually complex while $H(\Omega_1, \Omega_2, \dots, \Omega_M)$ is usually real. Therefore, the real and imaginary parts of $X(\Omega_1, \Omega_2, \dots, \Omega_M)$ are separately multiplied by $H(\Omega_1, \Omega_2, \dots, \Omega_M)$ and then combined to form $Y(\Omega_1, \Omega_2, \dots, \Omega_M)$. The third and final step in MD DFT filtering consists of applying the MD IDFT to $Y(\Omega_1, \Omega_2, \dots, \Omega_M)$ to obtain the required spatial domain output signal $\tilde{y}(k_1, k_2, \dots, k_M)$ which is periodically extended in M dimensions [3]. Thus,

$$\begin{aligned}
& \tilde{y}(k_1, k_2, \dots, k_M) \\
&= \frac{1}{L_1 L_2 \dots L_M} \sum_{\Omega_1=0}^{L_1-1} \sum_{\Omega_2=0}^{L_2-1} \dots \sum_{\Omega_M=0}^{L_M-1} Y(\Omega_1, \Omega_2, \dots, \Omega_M) \cdot \\
& \exp \left[2\pi \left[\frac{k_1 \Omega_1}{L_1} + \frac{k_2 \Omega_2}{L_2} + \dots + \frac{k_M \Omega_M}{L_M} \right] \right]
\end{aligned} \tag{1.21}$$

The required signal $y(k_1, k_2, \dots, k_M)$ is simply the part of $\tilde{y}(k_1, k_2, \dots, k_M)$ having support in $R^N \equiv \{k_i: 0 \leq k_i < L_i; k_i, L_i \text{ integer}, L_i < \infty, i = 1, 2, \dots, M\}$.

For example, consider a 2D signal $x(k_1, k_2)$ as the input to the filter whose frequency response is shown in Fig. 1.7. The 2D DFT filtering process carried out on $x(k_1, k_2)$ can be represented by the following sequence of equations:

Step 1:

$$X(\Omega_1, \Omega_2) = \sum_{k_1=0}^{L_1-1} \sum_{k_2=0}^{L_2-1} x(k_1, k_2) e^{-\frac{j2\pi k_1 \Omega_1}{L_1}} e^{-\frac{j2\pi k_2 \Omega_2}{L_2}} \tag{1.22}$$

Step 2:

$$Y(\Omega_1, \Omega_2) = H(\Omega_1, \Omega_2) X(\Omega_1, \Omega_2) \tag{1.23}$$

where $H(\Omega_1, \Omega_2)$ is defined in (1.18)

Step 3:

$$\tilde{y}(k_1, k_2) = \sum_{\Omega_1=0}^{L_1-1} \sum_{\Omega_2=0}^{L_2-1} Y(\Omega_1, \Omega_2) e^{\frac{j2\pi k_1 \Omega_1}{L_1}} e^{\frac{j2\pi k_2 \Omega_2}{L_2}} \quad (1.24)$$

One of the advantages of MD discrete-frequency filtering techniques, such as MD DFT filtering, is their simplicity. The design of MD DFT filters is straightforward and any required passband shape can be implemented.

However, a significant disadvantage of these methods is that they are computationally intensive and require large amounts of memory for data storage. In general, for an MD input signal of size $L_1 \times L_2 \times \cdots \times L_M$, where L_1, L_2, \cdots, L_M are all powers of 2, the number of floating-point operations required for MD DFT filtering is given by $2(L_1 L_2 \cdots L_M) [\log_2(L_1 L_2 \cdots L_M) + 1]$. For example, for a $256 \times 256 \times 256$ point 3D input signal ($L_1 = L_2 = L_3 = 256$), over 800 million floating-point operations will be required for 3D DFT filtering. Based solely on these computation requirements, such a filtering operation will take at least 25 seconds if the 3D DFT filter is implemented on a single Texas Instruments TMS-320C30 [5], one of the fastest signal processing chips currently available (speeds of up to 33 MFLOPS are possible). As a result, real-time video filtering cannot be achieved for a $256 \times 256 \times 256$ point image using a 3D DFT filter implemented on a single TMS-320C30.

Block filtering techniques [3], where large "blocks" of input data are processed simultaneously using parallel processing, are sometimes used to decrease

the time required for the filtering operation. However, since such blocks of data do not occur naturally in most real-time applications, these methods are typically used in off-line filtering applications where the data can conveniently be maintained and manipulated in computer memory.

DFT filtering techniques generally require significant amounts of memory for data storage. Typically, for a $L_1 \times L_2 \times \cdots \times L_M$ MD input signal, assuming the MD DFT filter has a purely real frequency response, $2(L_1 L_2 \cdots L_M)$ words of data must be stored in addition to the original input signal. For example, for a 3D $256 \times 256 \times 256$ image with a (relatively low) data wordlength of 16 bits, over 64 megabytes of memory are required. This is a considerable requirement, even on the most powerful of computer systems currently available.

The finite sequence length of the DFT and the periodic extension of the output signal produces undesirable "edge effects" in MD DFT filtering. In the case of 2D images for example, edge-effects produce "ringing" artifacts in the filtered output image. Ringing artifacts are also produced by Gibbs phenomena [6]. To reduce edge-effects the input signal is usually multiplied by a windowing function, such as Hamming [6], raised cosine [6], etc. Naturally, this increases the computational overhead for the MD DFT filtering process.

In spite of these disadvantages, MD DFT filtering is still widely used (for $M \leq 2$) because MD DFT filters are relatively easy to design and are inherently stable. MD continuous-frequency filtering techniques, described in the next

section, lend themselves to faster and much more compact filter algorithms than MD DFT filtering techniques, but the design process is much more complicated, as it is often very difficult to produce stable filters.

1.4. MD Continuous-Frequency Filtering

A MD continuous-frequency filter has a frequency response that is a function of M *continuous* frequency variables $\omega_1, \omega_2, \dots, \omega_M$. These filters are typically implemented using finite-order linear difference equations (LDE) henceforth referred to as simply LDEs. Transfer functions of the filters can be derived from these LDEs and it can be shown that they are functions of continuous-frequency variables.

Consider, for example, the 2D filtering operation represented by the first order 2D LDE that relates the input $x(k_1, k_2)$ to the output $y(k_1, k_2)$ as follows :

$$y(k_1, k_2) = x(k_1, k_2) + 2x(k_1 - 1, k_2 - 1) + 3x(k_1, k_2 - 1) + x(k_1 - 1, k_2) \quad (1.25)$$

In order to derive the z-domain transfer function $T(z_1, z_2)$, of the filter represented by (1.25), the 2D z-transform is applied to both sides of (1.25) resulting in

$$Y(z_1, z_2) = X(z_1, z_2)(1 + 2z_1^{-1}z_2^{-1} + 3z_2^{-1} + z_1^{-1}) \quad (1.26)$$

From the definition of the z-domain transfer function in (1.15),

$$T(z_1, z_2) \triangleq \frac{Y(z_1, z_2)}{X(z_1, z_2)} = 1 + 2z_1^{-1}z_2^{-1} + 3z_2^{-1} + z_1^{-1} \quad (1.27)$$

The frequency response $H(\omega_1, \omega_2)$ of the filter can be derived from (1.27) using the relationship in (1.18). That is,

$$H(\omega_1, \omega_2) = T(z_1, z_2) \Big|_{\substack{z_1=e^{j\omega_1} \\ z_2=e^{j\omega_2}}} \quad (1.28)$$

where ω_1, ω_2 are *continuous* frequency variables and T_1 and T_2 have been chosen as unity for simplicity.

$H(\omega_1, \omega_2)$ can now be written

$$H(\omega_1, \omega_2) = 1 + 2z_1^{-1}z_2^{-1} + 3z_2^{-1} + z_1^{-1} \Big|_{\substack{z_1=e^{j\omega_1} \\ z_2=e^{j\omega_2}}} \quad (1.29)$$

$$H(\omega_1, \omega_2) = 1 + 2e^{-j(\omega_1 + \omega_2)} + 3e^{-j\omega_2} + e^{-j\omega_1} \quad (1.30)$$

Using De Moivre's theorem to express the complex exponentials as a sum of sinusoids in (1.30) gives

$$\begin{aligned} H(\omega_1, \omega_2) = & 1 + 2\cos(\omega_1 + \omega_2) - j2\sin(\omega_1 + \omega_2) + 3\cos\omega_2 \\ & - j3\sin\omega_2 + \cos\omega_1 - j\sin\omega_1 . \end{aligned} \quad (1.31)$$

Collecting real and imaginary terms,

$$\begin{aligned}
H(\omega_1, \omega_2) &= [1 + 2\cos(\omega_1 + \omega_2) + 3\cos\omega_2 + \cos\omega_1] \\
&\quad - j[2\sin(\omega_1 + \omega_2) + 3\sin\omega_2 + \sin\omega_1]
\end{aligned} \tag{1.32}$$

Clearly, from (1.32), the 2D frequency response $H(\omega_1, \omega_2)$ corresponding to the first-order 2D LDE in (1.25) is a function of two *continuous* frequency variables, ω_1 and ω_2 .

In general, MD continuous-frequency filtering can be performed via MD LDEs of the form:

$$\begin{aligned}
&y(k_1, k_2, \dots, k_M) \\
&= \sum_{p_1=0}^{P_1-1} \sum_{p_2=0}^{P_2-1} \dots \sum_{p_M=0}^{P_M-1} a(p_1, p_2, \dots, p_M) x(k_1-p_1, k_2-p_2, \dots, k_M-p_M) \\
&\quad - \sum_{q_1=0}^{Q_1-1} \sum_{q_2=0}^{Q_2-1} \dots \sum_{q_M=0}^{Q_M-1} b(q_1, q_2, \dots, q_M) y(k_1-q_1, k_2-q_2, \dots, k_M-q_M)
\end{aligned} \tag{1.33}$$

where q_1, q_2, \dots, q_M cannot simultaneously equal zero and $P_i, Q_i < L_i ; i = 1, 2, \dots, M$.

Since the output MD spatial domain signal is calculated directly from the input MD spatial domain signal, only one stage of filtering is required. Continuous-frequency techniques are more suitable for real-time filtering because $y(k_1, k_2, \dots, k_M)$ is calculated *immediately* after $x(k_1, k_2, \dots, k_M)$ is

known, for all k_1, k_2, \dots, k_M . The edge effects inherent in discrete-frequency techniques can be eliminated if appropriate initial conditions are imposed at the boundaries of the input $x(k_1, k_2, \dots, k_M)$.

There are two types of continuous-frequency filters, namely, Infinite Impulse Response (IIR) filters, often referred to as *recursive* filters, and Finite Impulse Response (FIR) filters, often called *non-recursive* filters. These two filtering techniques are described in the next two sections.

1.4.1. MD Non-Recursive (FIR) Filtering

An MD non-recursive filter can be implemented by a LDE that is a special case of (1.33), that is, when $b(q_1, q_2, \dots, q_M) = 0$ for all q_1, q_2, \dots, q_M . Since $b(q_1, q_2, \dots, q_M) = 0$ for all q_1, q_2, \dots, q_M , each output sample $y(k_1, k_2, \dots, k_M)$ is a function of the previous and current input samples only. Previously calculated output samples are not used to calculate the next output sample - hence the term *non-recursive* filter. Such a filter may be implemented using an LDE of the form

$$\begin{aligned}
 & y(k_1, k_2, \dots, k_M) \\
 &= \sum_{p_1=0}^{P_1-1} \sum_{p_2=0}^{P_2-1} \dots \sum_{p_M=0}^{P_M-1} h(p_1, p_2, \dots, p_M) x(k_1-p_1, k_2-p_2, \dots, k_M-p_M)
 \end{aligned} \tag{1.34}$$

where $P_i < L_i$; $i = 1, 2, \dots, M$.

A comparison of (1.34) and (1.11) shows that $h(p_1, p_2, \dots, p_M)$ is the impulse response of the non-recursive filter represented by (1.34). Since $P_i < L_i$ for $i = 1, 2, \dots, M$, the impulse response $h(p_1, p_2, \dots, p_M)$ has a *finite number of non-zero samples*. Hence MD non-recursive filters are also called *finite impulse response (FIR) filters*.

A long impulse response is usually required to achieve high selectivity (i.e. sharp transitions from passband to stopband) in the frequency domain. Due to the necessarily finite extent of the impulse response, non-recursive filters satisfying given filter specifications often require a much higher order than recursive filters.

A convolution of the impulse response $h(p_1, p_2, \dots, p_M)$ with the MD input spatial domain signal $x(k_1, k_2, \dots, k_M)$ produces the MD output spatial domain signal $y(k_1, k_2, \dots, k_M)$ according to Equation (1.34). Thus, a higher order non-recursive filter requires a substantial number of computations and, if the filter order is sufficiently high, MD DFT filtering provides better computational efficiency. MD non-recursive filters are not used in this thesis.

One of the advantages of MD non-recursive filters is that they are always BIBO stable, since a finite length impulse response is always absolutely summable, according to (1.2). Consequently, FIR filter design algorithms are free to manipulate the filter coefficients to approximate a given specification without having to impose stability constraints.

Finally, non-recursive filters can be designed to have a purely real frequency response which implies that the frequency response possesses linear phase characteristics (zero phase). This eliminates distortion of lines and edges caused by non-zero phase frequency responses.

1.4.2. MD Recursive (IIR) Filtering

An MD recursive filter can be implemented by an LDE of the form (1.33) where $b(q_1, q_2, \dots, q_M) \neq 0$ for any q_1, q_2, \dots, q_M . Since each output value of $y(k_1, k_2, \dots, k_M)$ is a function of previously calculated output values, the term *recursive* filtering is used to describe this type of filtering operation. A recursive filter has an impulse response that has an infinite number of non-zero samples; therefore it is often called an infinite impulse response (IIR) filter. The long impulse responses that are a characteristic of recursive filters can produce a high degree of selectivity in the frequency domain.

Thus, for a given filter order, a recursive filter generally provides a higher degree of selectivity than a non-recursive filter. Alternatively, for a given filter specification, the required recursive filter usually has a much lower order than the corresponding non-recursive filter. This implies that fewer computations are required for a recursive filter than for the corresponding non-recursive filter. The computational efficiency of a recursive filter is a significant advantage and makes it more desirable than a non-recursive filter in real-time applications.

There are several methods for realizing MD recursive filters, the most common being the direct form method used in this thesis. As its name implies, the direct form method is a direct implementation of the recursive equation - the LDE for the recursive filter. The coefficients $a(p_1, p_2, \dots, p_M)$ are used to form an MD "input mask" of size $P_1 \times P_2 \times \dots \times P_M$ points, where each point corresponds to one of the coefficients $a(p_1, p_2, \dots, p_M)$. Similarly, the coefficients $b(q_1, q_2, \dots, q_M)$ are used to form a MD $Q_1 \times Q_2 \times \dots \times Q_M$ point "output mask".

Each input point overlapped by a point, or coefficient, on the mask is multiplied by that coefficient. All the weighted inputs are then added to form the input sum. Similarly, an output sum is created from the output mask and past outputs. The output sum is then subtracted from the input sum in accordance with the recursive equation to produce the next output point.

For example, for the 2D recursive filter shown in Fig. 1.8, the solid lines in Fig. 1.8a denote the boundary of the input mask and the solid lines in Fig. 1.8b represent the boundary of the output mask. Each point within the input mask corresponds to a coefficient $a(p_1, p_2)$, which multiplies the input value that it overlaps. These input values are added to form the input sum which is the first summation on the RHS of (1.33). Similarly, each point in the output mask corresponds to a coefficient $b(q_1, q_2)$, which multiplies the past output value that it overlaps. These weighted past output values are now added to form the output

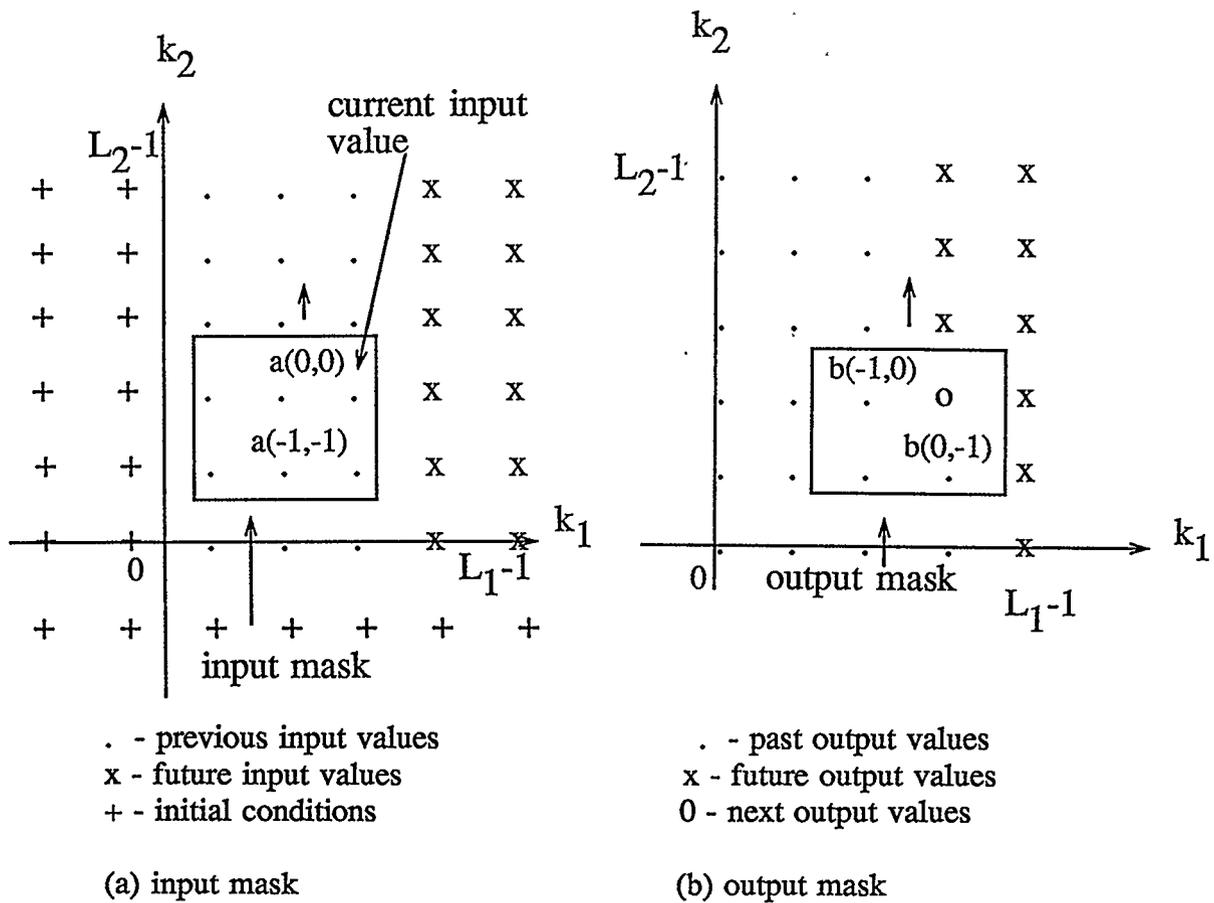


Fig. 1.8. Input and output masks for a 2D recursive filter.

sum - the second summation on the R.H.S. of (1.33). The next output value is now calculated by subtracting the output sum from the input sum in accordance with (1.33). The masks are then moved by one point to the next row or column, depending on the direction of recursion, and the process is repeated for the next output value.

From Fig. 1.8, it is clear that some initial conditions have to be set for the input mask. These are chosen to be zero to ensure stability.

There are some disadvantages associated with MD recursive filters. The determination of the coefficients $a(p_1, p_2, \dots, p_M)$ and $b(q_1, q_2, \dots, q_M)$ is a complicated problem that is the subject of continuing research [3]. The stability of MD recursive filters is non-trivial and not well understood at present. Approximation methods that are used to derive MD recursive filter coefficients often require sophisticated, computationally intensive, algebraic or numerical techniques in order to avoid instability [3]. MD recursive filter designs that guarantee stable MD filters are now available but complicated design procedures still limit their widespread use [3].

Another problem is that recursive filters are not zero-phase, that is, they have a non-linear phase response which can distort finely-detailed features in an image. This problem is usually overcome by resorting to "two-pass" filtering. This involves cascading two recursive filters having z-domain transfer functions $H(z_1, z_2, \dots, z_M)$ and $H(z_1^{-1}, z_2^{-1}, \dots, z_M^{-1})$ which produces the real non-negative frequency response $|H(\omega_1, \omega_2, \dots, \omega_M)|^2, z_i = e^{j\omega_i}$.

The next section briefly describes the proposed CDFD filtering technique that combines some of the advantages of continuous-frequency filtering and discrete-frequency filtering while circumventing some of the disadvantages inherent in each of these methods.

1.5. MD CDFD Filtering

The method of CDFD filtering applies discrete-frequency filtering to an MD input spatial domain signal $x(k_1, k_2, \dots, k_M)$, over less than all of the M dimensions and then applies continuous-frequency filtering over the remaining dimensions. Hence, if discrete-frequency filtering is first applied over $(M-N)$ of the M dimensions ($N < M$), then continuous frequency filtering will be applied over the remaining N dimensions.

MD CDFD filtering is a three-step process. First, the $(M-N)$ D DFT is applied to an input MD spatial domain signal $x(k_1, k_2, \dots, k_M)$ over the $M-N$ variables $k_{N+1}, k_{N+2}, \dots, k_M$, resulting in a *partially* DFT transformed *complex* signal $X(k_1, k_2, \dots, k_N, \Omega_{N+1}, \Omega_{N+2}, \dots, \Omega_M)$. This first step can be written as

$$X(k_1, k_2, \dots, k_N, \Omega_{N+2}, \dots, \Omega_M)$$

$$= \sum_{k_{N+1}=0}^{L_{N+1}-1} \sum_{k_{N+2}=0}^{L_{N+2}-1} \dots \sum_{k_M=0}^{L_M-1} x(k_1, k_2, \dots, k_N, k_{N+1}, k_{N+2}, \dots, k_M) \cdot \exp \left[-j2\pi \left(\frac{k_{N+1}\Omega_{N+1}}{L_{N+1}} + \frac{k_{N+2}\Omega_{N+2}}{L_{N+2}} + \dots + \frac{k_M\Omega_M}{L_M} \right) \right] \quad (1.35)$$

The second step in MD CDFD filtering consists of applying a continuous-frequency filtering operator, such as an ND recursive (IIR) filter, to the complex signal $X(k_1, k_2, \dots, k_N, \Omega_{N+1}, \Omega_{N+2}, \dots, \Omega_M)$ over the N variables k_1, k_2, \dots, k_N , to obtain the partially DFT transformed *complex* output signal

$Y(k_1, k_2, \dots, k_N, \Omega_{N+1}, \Omega_{N+2}, \dots, \Omega_M)$. Since $X(k_1, k_2, \dots, k_N, \Omega_{N+1}, \Omega_{N+2}, \Omega_M)$ is a *complex* valued signal, the ND recursive filter must operate separately on the real and imaginary parts of this signal. That is, a *complex filtering* operation is required. The filtered real and imaginary output signals are then combined to form the partially DFT transformed *complex* output signal $Y(k_1, k_2, \dots, k_N, \Omega_{N+1}, \Omega_{N+2}, \dots, \Omega_M)$. The second step can be written

$$Y(k_1, k_2, \dots, k_N, \Omega_{N+1}, \Omega_{N+2}, \dots, \Omega_M) \\ = \sum_{p_1=0}^{P_1-1} \sum_{p_2=0}^{P_2-1} \dots \sum_{p_N=0}^{P_N-1} a_{\Omega_{N+1}\Omega_{N+2} \dots, \Omega_M} (p_1, p_2, \dots, p_N) \cdot \\ X(k_1 - p_1, k_2 - p_2, \dots, k_N - p_N, \Omega_{N+1}, \Omega_{N+2}, \dots, \Omega_M) \quad (1.36)$$

$$- \sum_{q_1=0}^{Q_1-1} \sum_{q_2=0}^{Q_2-1} \dots \sum_{q_N=0}^{Q_N-1} b_{\Omega_{N+1}\Omega_{N+2} \dots, \Omega_M} (q_1, q_2, \dots, q_N) \cdot$$

$$Y(k_1 - q_1, k_2 - q_2, \dots, k_N - q_N, \Omega_{N+1}, \Omega_{N+2}, \dots, \Omega_M)$$

where q_1, q_2, \dots, q_N cannot be zero simultaneously and $P_i, Q_i < L_i$ for $i = 1, 2, \dots, N$. For each value of the discrete (M-N) tuple $\Omega_{N+1}, \Omega_{N+2}, \dots, \Omega_M$, the appropriate ND recursive filter has a different set of coefficients $a_{\Omega_{N+1}\Omega_{N+2} \dots, \Omega_M} (p_1, p_2, \dots, p_N)$ and $b_{\Omega_{N+1}\Omega_{N+2} \dots, \Omega_M} (q_1, q_2, \dots, q_N)$. Thus a total of $L_{N+1}L_{N+2} \dots L_M (L_1 \times L_2 \times \dots \times L_N)$ -point ND recursive filtering

operations are performed.

The third and final step in MD CDFD filtering is to apply the (M-N)D IDFT, over the M-N variables $\Omega_{N+1}, \Omega_{N+2}, \dots, \Omega_M$, to $Y(k_1, k_2, \dots, k_N, \Omega_{N+1}, \Omega_{N+2}, \dots, \Omega_M)$. This produces the MD spatial domain output signal $\tilde{y}(k_1, k_2, \dots, k_M)$. That is

$$\tilde{y}(k_1, k_2, \dots, k_M) = \frac{1}{L_{N+1}L_{N+2} \dots L_M} \sum_{\Omega_{N+1}=0}^{L_{N+1}-1} \sum_{\Omega_{N+2}=0}^{L_{N+2}-1} \dots \sum_{\Omega_M=0}^{L_M-1} Y(k_1, k_2, \dots, k_N, \Omega_{N+1}, \Omega_{N+2}, \dots, \Omega_M) \cdot \exp \left[j2\pi \left[\frac{\Omega_{N+1}k_{N+1}}{L_{N+1}} + \frac{\Omega_{N+2}k_{N+2}}{L_{N+2}} + \dots + \frac{\Omega_M k_M}{L_M} \right] \right] \quad (1.37)$$

where $\tilde{y}(k_1, k_2, \dots, k_M)$ is periodically extended in the dimensions over which the (M-N)D DFT is applied.

For example, a 3D cone filter might have a magnitude passband frequency response $|H_D(\Omega_1, \Omega_2, \Omega_3)|$ of unity at the discrete points shown in Fig. 1.9a where the Ω_3 axis has been enlarged for clarification. Consider the design of the 3D CDFD filter corresponding to the cone filter in Fig. 1.9a, using continuous-frequency filtering in one dimension ($N = 1$) and discrete frequency filtering in the

remaining two dimensions.

For a given input 3D signal $x(k_1, k_2, k_3)$ ($M = 3, N = 1$) the conventional *two-dimensional* DFT ($M-N = 2$) is applied to $x(k_1, k_2, k_3)$ over the dimensions k_2 and k_3 , leading to a partially DFT transformed complex signal $X(k_1, \Omega_2, \Omega_3)$, defined over a cubic grid. This step can be written as

$$X(k_1, \Omega_2, \Omega_3) = \sum_{k_2=0}^{L_2-1} \sum_{k_3=0}^{L_3-1} x(k_1, k_2, k_3) e^{-\frac{j2\pi k_2 \Omega_2}{L_2}} e^{-\frac{j2\pi k_3 \Omega_3}{L_3}} \quad (1.38)$$

This completes the first step in the CDFD filtering process.

Now, for *each* value of the discrete couple Ω_2, Ω_3 , the real and imaginary components of the corresponding complex sequence $X(k_1, \Omega_2, \Omega_3)$ are separately filtered by means of identical continuous-frequency filters (such as 1D recursive filters), leading to a filtered, complex output signal $Y(k_1, \Omega_2, \Omega_3)$. Each of the required $L_2 \times L_3$ filtering operations can be written

$$Y(k_1, \Omega_2, \Omega_3) = \sum_{p_1=0}^{P_1-1} a_{\Omega_2 \Omega_3}(p_1) X(k_1 - p_1, \Omega_2, \Omega_3) - \sum_{q_1=1}^{Q_1-1} b_{\Omega_2 \Omega_3}(q_1) Y(k_1 - q_1, \Omega_2, \Omega_3) \quad (1.39)$$

where $P_1, Q_1 < L_1$.

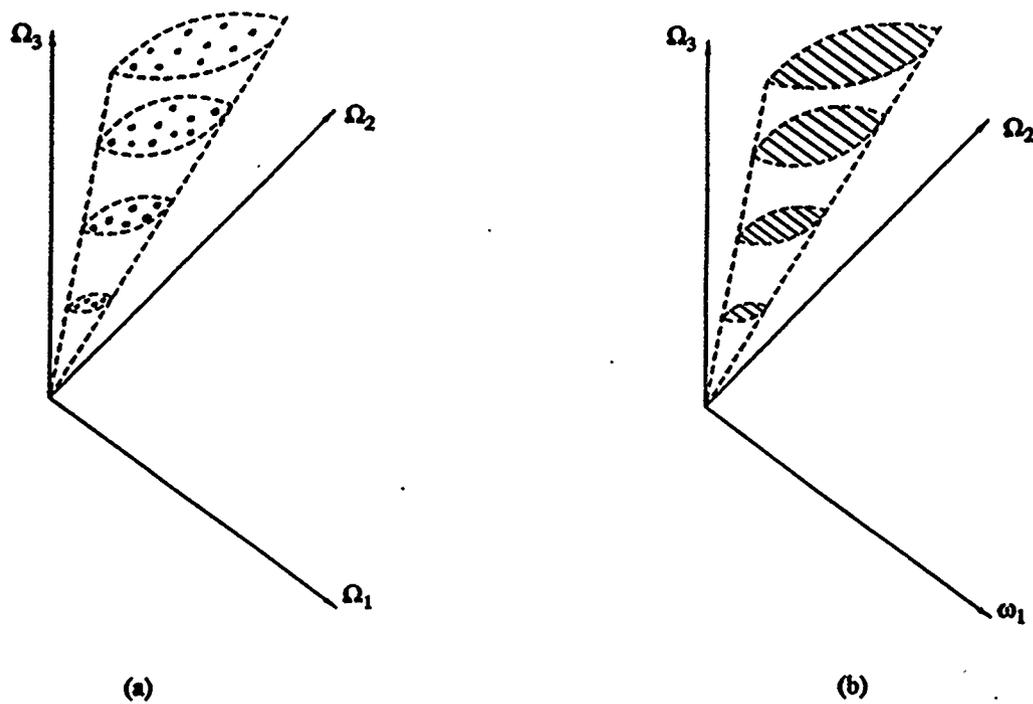


Fig. 1.9 (a) Frequency response $H_D(\Omega)$ of a 3D cone filter (dots represent passband gain of unity).
 (b) CDFD filter approximation of the frequency response in (a) (parallel lines represent passband gain of approximately unity).

The corresponding frequency response function $H(\omega_1, \Omega_2, \Omega_3)$ is continuous in the frequency variable ω_1 and discrete in the frequency variables Ω_2 and Ω_3 . The corresponding passband magnitude frequency response $|H(\omega_1, \Omega_2, \Omega_3)|$, shown in Fig. 1.9b, is simply a set of parallel straight lines, in the direction of the ω_1 axis, along which $|H(\omega_1, \Omega_2, \Omega_3)| \approx 1$. These lines pass through any plane $\omega_1 = \omega_{1_0}$ at the points $(\omega_{1_0}, \Omega_2, \Omega_3)$ where ω_{1_0} is a constant.

Each passband line in the CDFD cone filter in Fig. 1.9b corresponds to the passband of a separate 1D continuous-frequency bandpass filter, having upper and lower cutoff frequencies chosen to achieve the required cone shape by appropriately choosing the values of the coefficients $a_{\Omega_2\Omega_3}(p_1)$ and $b_{\Omega_2\Omega_3}(q_1)$.

In this thesis, applications of CDFD filtering are described, where recursive (IIR) filtering is employed in one frequency dimension and DFT filtering is employed in the remaining frequency dimensions. It is shown that such filters have some inherent practical advantages over MD filters that are either entirely discrete or entirely continuous in the frequency domain.

1.6. Scope and Objective of Thesis

The objective of this thesis is to introduce and employ MD CDFD filtering in a variety of signal processing problems and to identify promising potential applications for MD CDFD filters.

Some of the basic concepts and notation which are used to describe the MD CDFD filtering technique are discussed in Chapter 2. Two key concepts, namely alternate array representations of M-variate functions and the partial DFT, are also introduced in Chapter 2.

The CDFD filtering technique is described in Chapter 3. The frequency response of an MD CDFD filter is derived. Numerous design examples are provided to illustrate the CDFD filtering technique and the wide range of applications for MD CDFD filters. A comparison of a 2D CDFD fanstop filter with its IIR and 2D DFT counterparts is used to demonstrate the advantages of CDFD filtering over conventional filtering techniques.

A 2D image processing system using 2D CDFD filters is described in Chapter 4. Hardware schemes for hardware implementation of MD CDFD filters are also proposed.

Finally, conclusions and recommendations for further work are given in Chapter 5.

CHAPTER 2

NOTATION FOR CDFD FILTERING

2.1. Introduction

The equations describing the CDFD filtering technique in Section 1.5 are cumbersome. It is therefore desirable to express them in a more compact form as since equations of this type are used extensively in this thesis to describe CDFD filtering. To this end, notation that allows a more compact representation of MD functions is developed in this chapter. A more compact notation for the MD DFT and partial MD DFT is also presented.

In the next section, vector representation of MD frequency domain and spatial domain variables in MD signals is introduced.

2.2. Vector Representation of Frequency and Spatial Variables in MD Signals

Consider the real MD spatial domain signal $x(k_1, k_2, \dots, k_M) \in R$ which is a scalar real valued function (having region of support R) of the M spatial domain variables k_1, k_2, \dots, k_M . The set of M spatial domain variables k_1, k_2, \dots, k_M can be represented by a component column vector having elements that are discrete spatial variables. Thus,

$$\mathbf{k} \equiv \{k_1, k_2, \dots, k_M\}' \in R^M \quad (2.1)$$

where $0 \leq k_i < L_i$; with k_i, L_i being positive integers for $i = 1, 2, \dots, M$. The prime in (2.1) denotes transposition. The MD spatial domain signal $x(k_1, k_2, \dots, k_M)$ can now be written as a function of \mathbf{k} in the form

$$x(k_1, k_2, \dots, k_M) \equiv x(\mathbf{k}) \in R \quad (2.2)$$

Similarly, a complex MD frequency domain signal $Y(\Omega_1, \Omega_2, \dots, \Omega_M) \in C$ (the C indicates that $Y(\Omega_1, \Omega_2, \dots, \Omega_M)$ is a scalar *complex* valued function) can be written as a function of the MD column vector Ω having elements that are discrete-frequency variables, as follows :

$$\Omega \equiv \{\Omega_1, \Omega_2, \dots, \Omega_M\}' \in R^M \quad (2.3)$$

where $0 \leq \Omega_i < L_i$ with Ω_i, L_i being positive integers for $i = 1, 2, \dots, M$.

Thus, $Y(\Omega_1, \Omega_2, \dots, \Omega_M)$ can now be written as a function of Ω

$$Y(\Omega_1, \Omega_2, \dots, \Omega_M) \equiv Y(\Omega) \in C \quad (2.4)$$

The vector representation of a set of M variables can also be applied to continuous variables such as the set of M continuous-frequency variables $\omega_1, \omega_2, \dots, \omega_M$. These variables can be expressed in terms of the MD column vector ω , having elements that are continuous-frequency variables, as follows : :

$$\omega \equiv \{\omega_1, \omega_2, \dots, \omega_M\}' \in R^M \quad (2.5)$$

Thus, a function $x(\omega_1, \omega_2, \dots, \omega_M) \in R$, of the continuous frequency variables

$\omega_1, \omega_2, \dots, \omega_M$, can be written as a function of the MD column vector ω , that is

$$x(\omega_1, \omega_2, \dots, \omega_M) \equiv x(\omega) \in R \quad (2.6)$$

The representation of purely discrete or purely continuous signals as functions of MD column vectors (Equations (2.4) and (2.6), respectively) can be extended to mixed continuous-discrete signals. For example, consider the mixed continuous-discrete frequency response of an MD CDFD filter, $H(\omega_1, \omega_2, \dots, \omega_N, \Omega_{N+1}, \Omega_{N+2}, \dots, \Omega_M) \in C$. The N continuous frequency variables $\omega_1, \omega_2, \dots, \omega_N$, can be expressed as an ND column vector ω_c , where

$$\omega_c \equiv \{\omega_1, \omega_2, \dots, \omega_N\}' \in R^N \quad (2.7)$$

The subscript c on ω_c denotes that ω_c is a column vector associated with *continuous* frequency variables.

Similarly, the $M-N$ discrete frequency variables $\Omega_{N+1}, \Omega_{N+2}, \dots, \Omega_M$, can be written as a $(M-N)$ D column vector Ω_d , so that

$$\Omega_d \equiv \{\Omega_{N+1}, \Omega_{N+2}, \dots, \Omega_M\}' \in R^{M-N} \quad (2.8)$$

where $0 \leq \Omega_i < L_i$ with Ω_i, L_i being positive integer for $i = N + 1, N + 2, \dots, M$. The subscript d on Ω_d signifies that Ω_d is a column vector associated with *discrete* frequency variables. Henceforth, in this thesis, unless otherwise specified, the subscripts c and d will be used to denote vectors associated with continuous

and discrete frequency variables, respectively.

Thus, the MD mixed continuous-discrete frequency response $H(\omega_1, \omega_2, \dots, \omega_N, \Omega_{N+1}, \Omega_{N+2}, \dots, \Omega_M)$ can be written in terms of $\omega_{\mathbf{c}}$ and $\Omega_{\mathbf{d}}$.

That is

$$H(\omega_1, \omega_2, \dots, \omega_N, \Omega_{N+1}, \Omega_{N+2}, \dots, \Omega_M) \equiv H(\omega_{\mathbf{c}}, \Omega_{\mathbf{d}}) \in C \quad (2.9)$$

where the R.H.S. is clearly a more compact expression than the L.H.S.

MD signals that are a function of some frequency variables and some spatial variables, such as the partially DFT'ed MD signal $X(k_1, k_2, \dots, k_N, \Omega_{N+1}, \Omega_{N+2}, \dots, \Omega_M) \in C$ (Equation (1.35)), can also be expressed as functions of vectors. For the signal $X(k_1, k_2, \dots, k_N, \Omega_{N+1}, \Omega_{N+2}, \dots, \Omega_M)$, two vectors $\mathbf{k}_{\mathbf{c}}$ and $\Omega_{\mathbf{d}}$ are defined so that

$$\mathbf{k}_{\mathbf{c}} \equiv \{k_1, k_2, \dots, k_N\}' \in R^N \quad (2.10)$$

where $\mathbf{k}_{\mathbf{c}}$ is an ND column vector associated with continuous frequency variables and $\Omega_{\mathbf{d}}$ is the (M-N)D column vector defined in (2.8). This leads directly to a more compact expression for $X(k_1, k_2, \dots, k_N, \Omega_{N+1}, \Omega_{N+2}, \dots, \Omega_M)$, that is

$$X(k_1, k_2, \dots, k_N, \Omega_{N+1}, \Omega_{N+2}, \dots, \Omega_M) \equiv X(\mathbf{k}_{\mathbf{c}}, \Omega_{\mathbf{d}}) \in C \quad (2.11)$$

The vector representation of frequency and spatial variables described in this section is used throughout this thesis. In the next section, a system of notation that

represents MD signals using arrays is introduced. As a result, the CDFD filtering process can be described in a much more compact and convenient manner.

2.3. Alternate Array Representations of Scalar Functions in

M variables

In section 2.2, a system of notation that represented sets of M variables as vectors was used to write M-variate functions (i.e. functions of M variables) in a more compact form. This notation was then extended to represent functions of two types of variables, such as spatial and frequency variables, as functions of two types of vectors (Equation (2.11)). In this section, the system of notation introduced in section 2.2 is further extended so that M-variate functions can be represented by arrays.

Consider the MD complex signal $A(\mathbf{k}) \in C$, which is a scalar M-variate function and \mathbf{k} is the vector defined in (2.1). In the 3D case, for example, $\mathbf{k} \equiv \{k_1, k_2, k_3\}'$ and $A(\mathbf{k}) \equiv A(k_1, k_2, k_3)$. Now, if \mathbf{k} is divided into the vector \mathbf{k}_c and \mathbf{k}_d such that \mathbf{k}_c is defined by (2.10) and

$$\mathbf{k}_d \equiv \{k_{N+1}, k_{N+2}, \dots, k_M\}' \in R^{M-N} \quad (2.12)$$

then $A(\mathbf{k})$ can be written as follows :

$$A(\mathbf{k}) \equiv A(\mathbf{k}_c, \mathbf{k}_d) \in C \quad (2.13)$$

In the 3D example described above, let $\mathbf{k}_c = \{k_1\}$ and $\mathbf{k}_d = \{k_2, k_3\}'$ so that $A(\mathbf{k}_c, \mathbf{k}_d) \equiv A(k_1, k_2, k_3)$ as before.

Alternatively, $A(\mathbf{k}_c, \mathbf{k}_d)$ can be written as an (M-N)D array $\mathbf{A}_d(\mathbf{k}_c)$ having elements that are scalar complex functions of the N variables \mathbf{k}_c , that is

$$A(\mathbf{k}_c, \mathbf{k}_d) \triangleq \mathbf{A}_d(\mathbf{k}_c) \equiv \{A_{d_{\mathbf{k}_d}}(\mathbf{k}_c)\} \in C^{L_{N+1} \times L_{N+2} \times \cdots \times L_M} \quad (2.14)$$

In Equation (2.14) above, the brackets $\{ \}$ enclose the term describing each element of the array $\mathbf{A}_d(\mathbf{k}_c)$. The subscript \mathbf{k}_d on the expression for an element $A_{d_{\mathbf{k}_d}}(\mathbf{k}_c)$ implies that the array $\mathbf{A}_d(\mathbf{k}_c)$ has an element for *each value* of the vector \mathbf{k}_d . Thus, the array $\mathbf{A}_d(\mathbf{k}_c)$ is of size $L_{N+1} \times L_{N+2} \times \cdots \times L_M$ and each element $A_{d_{\mathbf{k}_d}}(\mathbf{k}_c)$ is a scalar complex function of the N variables \mathbf{k}_c .

The expression representing the array \mathbf{A}_d is written in bold type to distinguish it from the expression for an *element* of the array $A_{d_{\mathbf{k}_d}}$. The expression for an element of the array also contains an additional subscript vector (such as \mathbf{k}_d in $A_{d_{\mathbf{k}_d}}$) to distinguish it from the expression representing the array \mathbf{A}_d . This notation will be used throughout this thesis to represent (M-N)D arrays of N-variate scalar functions which may be either real or complex.

For the 3D example described above, Equation (2.14) can be written as follows :

$$\begin{aligned}
A(\mathbf{k}_c, \mathbf{k}_d) &= A(k_1, k_2, k_3) \triangleq \mathbf{A}_d(k_1) \equiv \{A_{d_{k_2, k_3}}(k_1)\} \\
&\equiv \begin{bmatrix} A_{d_{0,0}}(k_1) & A_{d_{0,1}}(k_1) & \cdots & A_{d_{0,L_3-1}}(k_1) \\ A_{d_{1,0}}(k_1) & A_{d_{1,1}}(k_1) & \cdots & A_{d_{1,L_3-1}}(k_1) \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ A_{d_{L_2-1,0}}(k_1) & A_{d_{L_2-1,1}}(k_1) & \cdots & A_{d_{L_2-1,L_3-1}}(k_1) \end{bmatrix} \quad (2.15)
\end{aligned}$$

where the 2D array (or matrix) in (2.15) is clearly of size $L_2 \times L_3$.

A scalar M-variate function such as $A(\mathbf{k}) \equiv A(\mathbf{k}_c, \mathbf{k}_d) \in C$, can also be written as an ND array $\mathbf{A}_c(\mathbf{k}_d)$ having elements that are scalar complex functions of the M-N variables \mathbf{k}_d , that is

$$A(\mathbf{k}_c, \mathbf{k}_d) \triangleq \mathbf{A}_c(\mathbf{k}_d) \equiv \{A_{c_{\mathbf{k}_c}}(\mathbf{k}_d)\} \in C^{L_1 \times L_2 \times \cdots \times L_N} \quad (2.16)$$

The array \mathbf{A}_c has one element for *each value* of the vector \mathbf{k}_c and is of size $L_1 \times L_2 \times \cdots \times L_N$. In the 3D case described in this section, equation (2.16) can now be written

$$\begin{aligned}
A(\mathbf{k}_c, \mathbf{k}_d) &= A(k_1, k_2, k_3) \triangleq A_c(k_2, k_3) \equiv \{A_{c_{k_1}}(k_2, k_3)\} \\
&\equiv \begin{bmatrix} A_{c_0}(k_2, k_3) \\ A_{c_1}(k_2, k_3) \\ \vdots \\ \vdots \\ A_{c_{L_1-1}}(k_2, k_3) \end{bmatrix} \in C^{L_1} \quad (2.17)
\end{aligned}$$

where the 1D array (or vector) in (2.17) has L_1 elements, each of which are complex scalar functions of the variables k_2, k_3 .

Thus, any of the following functional equivalences may be used to represent the scalar M-variate function $A(\mathbf{k}_c, \mathbf{k}_d) \equiv A(\mathbf{k})$:

$$A_c(\mathbf{k}_d) \equiv \{A_{c_{k_c}}(\mathbf{k}_d)\} \triangleq A(\mathbf{k}_c, \mathbf{k}_d) \triangleq A_d(\mathbf{k}_c) \equiv \{A_{d_{k_d}}(\mathbf{k}_c)\} \quad (2.18)$$

which, in the 3D case described above, may be written as

$$\begin{aligned}
& \begin{bmatrix} A_{c_0}(k_2, k_3) \\ A_{c_1}(k_2, k_3) \\ \vdots \\ A_{c_{L_1-1}}(k_2, k_3) \end{bmatrix} \\
& \cong A(k_1, k_2, k_3) \cong \begin{bmatrix} A_{d_{0,0}}(k_1) & A_{d_{0,1}}(k_1) & \cdots & A_{d_{0,L_3-1}}(k_1) \\ A_{d_{1,0}}(k_1) & A_{d_{1,1}}(k_1) & \cdots & A_{d_{1,L_3-1}}(k_1) \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ A_{d_{L_2-1,0}}(k_1) & A_{d_{L_2-1,1}}(k_1) & \cdots & A_{d_{L_2-1,L_3-1}}(k_1) \end{bmatrix} \quad (2.19)
\end{aligned}$$

It follows from (2.18) above that any scalar complex M-variate function $X(\mathbf{k})$ can be expressed as an MD array of complex elements as follows :

$$X(\mathbf{k}) \triangleq \mathbf{X}_{\mathbf{d}} \equiv \left\{ X_{d_{\mathbf{k}}} \right\} \in C^{L_1 \times L_2 \times \cdots \times L_M} \quad (2.20)$$

For example, a 2D real scalar function $x(k_1, k_2)$ can be written as

$$\begin{aligned}
x(k_1, k_2) \triangleq \mathbf{x}_d &\equiv \left\{ x_{d_{k_1, k_2}} \right\} \\
&\equiv \begin{bmatrix} x_{d_{0,0}} & x_{d_{0,1}} & \cdots & x_{d_{0,L_2-1}} \\ x_{d_{1,0}} & x_{d_{1,1}} & \cdots & x_{d_{1,L_2-1}} \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ x_{d_{L_1-1,0}} & x_{d_{L_1-1,1}} & \cdots & x_{d_{L_1-1,L_2-1}} \end{bmatrix} \quad (2.21) \\
&\in R^{L_1 \times L_2}
\end{aligned}$$

where the value of each real element $x_{d_{k_1, k_2}}$ is simply the value of the function

$x(k_1, k_2)$ evaluated with the appropriate values of k_1 and k_2 .

In the next section, some element-wise array operations that facilitate the manipulation of MD arrays used to describe CDFD filtering are defined.

2.4. Element-Wise Array Operations

Three element-wise array operators are described in this section. They operate on MD arrays and are useful in describing both CDFD and MD DFT filtering. First, let $\mathbf{A} \equiv \{A_{\mathbf{k}}\}$, $\mathbf{B} \equiv \{B_{\mathbf{k}}\}$ and $\mathbf{C} \equiv \{C_{\mathbf{k}}\}$ be MD real arrays of size

$L_1 \times L_2 \times \cdots \times L_M$ so that $\mathbf{A}, \mathbf{B}, \mathbf{C} \in R^{L_1 \times L_2 \times \cdots \times L_M}$.

Multiplication and Division

The M -dimensional element-wise array multiplication operator \cdot_M is defined such that

$$\mathbf{C} = \mathbf{A} \cdot_M \mathbf{B} \quad (2.22)$$

implies that

$$\mathbf{C} \equiv \{C_{\mathbf{k}}\} = \{A_{\mathbf{k}} B_{\mathbf{k}}\} \quad (2.23)$$

and also that

$$\mathbf{C} = \{B_{\mathbf{k}} A_{\mathbf{k}}\} = \mathbf{B} \cdot_M \mathbf{A} \quad (2.24)$$

For example, for two 2D arrays of size 3×3 , $\mathbf{X} \equiv \{X_{k_1, k_2}\}$ and $\mathbf{H} \equiv \{H_{k_1, k_2}\}$ where $0 \leq k_1, k_2 < 3$, the 2D element-wise array multiplication operator \cdot_2 can be used to derive a third array $\mathbf{Y} \equiv \{Y_{k_1, k_2}\}$ such that

$$\mathbf{Y} = \mathbf{H} \cdot_2 \mathbf{X} \quad (2.25)$$

This can be written in full as

$$\begin{aligned}
\mathbf{Y} \equiv \left\{ Y_{k_1, k_2} \right\} &\equiv \begin{bmatrix} Y_{0,0} & Y_{0,1} & Y_{0,2} \\ Y_{1,0} & Y_{1,1} & Y_{1,2} \\ Y_{2,0} & Y_{2,1} & Y_{2,2} \end{bmatrix} \\
&= \begin{bmatrix} H_{0,0} & H_{0,1} & H_{0,2} \\ H_{1,0} & H_{1,1} & H_{1,2} \\ H_{2,0} & H_{2,1} & H_{2,2} \end{bmatrix} \cdot \begin{bmatrix} X_{0,0} & X_{0,1} & X_{0,2} \\ X_{1,0} & X_{1,1} & X_{1,2} \\ X_{2,0} & X_{2,1} & X_{2,2} \end{bmatrix} \\
&= \begin{bmatrix} H_{0,0}X_{0,0} & H_{0,1}X_{0,1} & H_{0,2}X_{0,2} \\ H_{1,0}X_{1,0} & H_{1,1}X_{1,1} & H_{1,2}X_{1,2} \\ H_{2,0}X_{2,0} & H_{2,1}X_{2,1} & H_{2,2}X_{2,2} \end{bmatrix}
\end{aligned}$$

(2.26)

Thus, each element of \mathbf{Y} is the product of the corresponding elements in \mathbf{H} and \mathbf{X} .

From (2.26), it is clear that all three arrays must have the same size and dimension.

Similarly, the M -dimensional element-wise array division operator $/_M$ is defined for $\mathbf{B} \neq 0$, such that

$$\mathbf{C} = \mathbf{A} /_M \mathbf{B} \quad (2.27)$$

implies that

$$\mathbf{C} = \{A_{\mathbf{k}}/B_{\mathbf{k}}\} \quad (2.28)$$

In the 2D example described above, this becomes

$$\mathbf{H} = \mathbf{Y} /_2 \mathbf{X} \quad (2.29)$$

which can be written in full as

$$\begin{aligned} \mathbf{H} &\equiv \left\{ H_{k_1, k_2} \right\} \equiv \begin{bmatrix} H_{0,0} & H_{0,1} & H_{0,2} \\ H_{1,0} & H_{1,1} & H_{1,2} \\ H_{2,0} & H_{2,1} & H_{2,2} \end{bmatrix} \\ &= \begin{bmatrix} Y_{0,0} & Y_{0,1} & Y_{0,2} \\ Y_{1,0} & Y_{1,1} & Y_{1,2} \\ Y_{2,0} & Y_{2,1} & Y_{2,2} \end{bmatrix} /_2 \begin{bmatrix} X_{0,0} & X_{0,1} & X_{0,2} \\ X_{1,0} & X_{1,1} & X_{1,2} \\ X_{2,0} & X_{2,1} & X_{2,2} \end{bmatrix} \quad (2.30) \\ &= \begin{bmatrix} Y_{0,0}/X_{0,0} & Y_{0,1}/X_{0,1} & Y_{0,2}/X_{0,2} \\ Y_{1,0}/X_{1,0} & Y_{1,1}/X_{1,1} & Y_{1,2}/X_{1,2} \\ Y_{2,0}/X_{2,0} & Y_{2,1}/X_{2,1} & Y_{2,2}/X_{2,2} \end{bmatrix} \end{aligned}$$

provided $\mathbf{X} \neq 0$.

The two operators \cdot_M and $/_M$ defined above provide a useful relationship as follows :

Given

$$\mathbf{C} = \mathbf{A} \cdot_M \mathbf{B} \quad (2.31)$$

then if $\mathbf{B} \neq 0$

$$\mathbf{A} = \mathbf{C} /_M \mathbf{B} \quad (2.32)$$

Convolution

Now assume that $A_{\mathbf{k}}$ and $B_{\mathbf{k}}$ are scalar functions of N variables. The M -dimensional element-wise array convolution operator $*_{\frac{N}{M}}$ is defined such that

$$\mathbf{C} = \mathbf{A} *_{\frac{N}{M}} \mathbf{B} \quad (2.33)$$

implies that

$$\mathbf{C} = \{A_{\mathbf{k}} * B_{\mathbf{k}}\} \quad (2.34)$$

where $*$ is the N -dimensional circular convolution operator.

For example, for two 2D arrays of size 3×3 , $\mathbf{x} \equiv \{x_{k_1, k_2}^{(n_1)}\}$ and $\mathbf{h} \equiv \{h_{k_1, k_2}^{(n_1)}\}$ where $0 \leq k_1, k_2 < 3$, a third array $\mathbf{y} \equiv \{y_{k_1, k_2}^{(n_1)}\}$ can be defined such that

$$\mathbf{y} = \mathbf{h} *_{\frac{1}{2}} \mathbf{x} \quad (2.35)$$

implies that

$$\begin{aligned}
\mathbf{y} &\equiv \begin{bmatrix} y_{0,0}(n_1) & y_{0,1}(n_1) & y_{0,2}(n_1) \\ y_{1,0}(n_1) & y_{1,1}(n_1) & y_{1,2}(n_1) \\ y_{2,0}(n_1) & y_{2,1}(n_1) & y_{2,2}(n_1) \end{bmatrix} \\
&= \begin{bmatrix} h_{0,0}(n_1) & h_{0,1}(n_1) & h_{0,2}(n_1) \\ h_{1,0}(n_1) & h_{1,1}(n_1) & h_{1,2}(n_1) \\ h_{2,0}(n_1) & h_{2,1}(n_1) & h_{2,2}(n_1) \end{bmatrix} * \frac{1}{2} \begin{bmatrix} x_{0,0}(n_1) & x_{0,1}(n_1) & x_{0,2}(n_1) \\ x_{1,0}(n_1) & x_{1,1}(n_1) & x_{1,2}(n_1) \\ x_{2,0}(n_1) & x_{2,1}(n_1) & x_{2,2}(n_1) \end{bmatrix} \quad (2.36) \\
&= \begin{bmatrix} h_{0,0}(n_1) * x_{0,0}(n_1) & h_{0,1}(n_1) * x_{0,1}(n_1) & h_{0,2}(n_1) * x_{0,2}(n_1) \\ h_{1,0}(n_1) * x_{1,0}(n_1) & h_{1,1}(n_1) * x_{1,1}(n_1) & h_{1,2}(n_1) * x_{1,2}(n_1) \\ h_{2,0}(n_1) * x_{2,0}(n_1) & h_{2,1}(n_1) * x_{2,1}(n_1) & h_{2,2}(n_1) * x_{2,2}(n_1) \end{bmatrix}
\end{aligned}$$

where $*$ is the 1D circular convolution operator [4]. In (2.36) above, each element of \mathbf{y} is the result of a 1D circular convolution of the corresponding elements of \mathbf{h} and \mathbf{x} .

In the next section, the MD DFT operator $F_{1,M}^M []$ is defined. The equations for MD DFT filtering are then rewritten using the notation described thus far in this chapter.

2.5. The MD DFT Operator $F_{1,M}^M []$

The MD operator $F_{1,M}^M []$ operates on an MD spatial domain signal $x(\mathbf{k}) \equiv x(k_1, k_2, \dots, k_M)$ and transforms it into an MD frequency domain signal $X(\Omega) \equiv X(\Omega_1, \Omega_2, \dots, \Omega_M)$ as follows :

$$X(\Omega) \triangleq F_{1,M}^M[x(\mathbf{k})] \quad (2.37)$$

and

$$F_{1,M}^M[x(\mathbf{k})] \triangleq \sum_{k_1=0}^{L_1-1} \sum_{k_2=0}^{L_2-1} \cdots \sum_{k_M=0}^{L_M-1} x(\mathbf{k}) \exp \left[j2\pi \left[-\frac{k_1\Omega_1}{L_1} - \frac{k_2\Omega_2}{L_2} - \cdots - \frac{k_M\Omega_M}{L_M} \right] \right] \quad (2.38)$$

In Equation (2.38) above, even though $x(\mathbf{k})$ has been written in the notation developed in this chapter, the multiple-summation and exponential terms on the R.H.S. are still cumbersome. In order to express the exponential term in a more compact form, the diagonal matrix \mathbf{L} is defined such that

$$\mathbf{L} \equiv \begin{bmatrix} L_1 & 0 & \cdots & 0 \\ 0 & L_2 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ 0 & 0 & \cdots & L_M \end{bmatrix} \quad (2.39)$$

where L_i is the finite sequence length in the i^{th} dimension. The multiple-summation in (2.38) will be written, for brevity, as follows :

$$\sum_{k_1=0}^{L_1-1} \sum_{k_2=0}^{L_2-1} \cdots \sum_{k_M=0}^{L_M-1} \triangleq \sum_{\mathbf{k}=0}^{\mathbf{L}-1} \quad (2.40)$$

Thus, Equation (2.38) can now be written

$$F_{1,M}^M[x(\mathbf{k})] \triangleq \sum_{\mathbf{k}=0}^{\mathbf{L}-1} x(\mathbf{k}) \exp[-j\Omega'(2\pi\mathbf{L}^{-1})\mathbf{k}] \quad (2.41)$$

where Ω and \mathbf{k} are defined by (2.3) and (2.1) respectively. The superscript M on $F_{1,M}^M$ denotes that an MD sequence is being transformed while the subscript $1,M$ indicates that the transformation is applied over all M variables k_1, k_2, \dots, k_M .

The significance of the subscript $1,M$ becomes apparent when the DFT operation is not performed on all M variables, such as during a partial DFT operation. For example, the partial DFT operation in Equation (1.35) in Chapter 1 can be represented by the operator $F_{N+1,M}^M$, since the DFT is applied only to the variables $k_{N+1}, k_{N+2}, \dots, k_M$. It will be shown in the next section that the notation that allows the MD DFT operation to be written compactly (Equation (2.41)), can also be applied to the partial DFT.

An operator $F_{1,M}^{-M} []$ representing the MD IDFT is defined as follows :

$$x(\mathbf{k}) \triangleq F_{1,M}^{-M}[X(\Omega)] \triangleq \frac{1}{\det(\mathbf{L})} \sum_{\Omega=0}^{\mathbf{L}-1} X(\Omega) \exp[j\Omega'(2\pi\mathbf{L}^{-1})\mathbf{k}] \quad (2.42)$$

where the negative sign on the superscript of $F_{1,M}^{-M}$ signifies that the inverse transform is applied. As an example of the notation described in this chapter, the MD DFT filtering process, represented by equations (1.19), (1.20) and (1.21) in

Chapter 1, is now described using this new notation.

Equation (1.19) can be written as follows :

$$X(\Omega) = F_{1,M}^M[x(\mathbf{k})] \triangleq \sum_{\mathbf{k}=0}^{L-1} x(\mathbf{k}) \exp[-j\Omega'(2\pi L^{-1})\mathbf{k}] \quad (2.43)$$

Equation (1.20) can be written as

$$Y(\Omega) = H(\Omega)X(\Omega) \quad (2.44)$$

If $Y(\Omega)$, $H(\Omega)$ and $X(\Omega)$ are expressed as MD arrays having real or complex elements as in (2.20), then (2.44) can also be written as

$$Y(\Omega) \triangleq \mathbf{Y}_d \equiv \{Y_{d_\Omega}\} = \mathbf{H}_d \cdot_M \mathbf{X}_d \equiv \{H_{d_\Omega} X_{d_\Omega}\} \quad (2.45)$$

The MD frequency response of the filter can be derived from (2.45) using (2.31) and (2.32), so that

$$H(\Omega) \triangleq \mathbf{H}_d \equiv \{H_{d_\Omega}\} = \mathbf{Y}_d /_M \mathbf{X}_d \equiv \{Y_{d_\Omega} / X_{d_\Omega}\} \quad (2.46)$$

The final step in the MD DFT filtering process, represented by (1.21) can be written as follows :

$$\tilde{y}(\mathbf{k}) = F_{1,M}^{-M}[Y(\Omega)] \triangleq \frac{1}{\det(\mathbf{L})} \sum_{\Omega=0}^{L-1} Y(\Omega) \exp[j\Omega'(2\pi L^{-1})\mathbf{k}] \quad (2.47)$$

In the next section, the partial MD DFT is described using the notation developed thus far in this chapter.

2.6. The Partial MD DFT

The partial MD DFT operation on an MD signal $x(\mathbf{k})$; $\mathbf{k} \equiv \{k_1, k_2, \dots, k_M\}'$, consists of applying the DFT to not all of the M variables \mathbf{k} .

Let \mathbf{k} be partitioned as follows :

$$\mathbf{k} \equiv \{\mathbf{k}_c, \mathbf{k}_d\}' \in R^M \quad (2.48)$$

where

$$\mathbf{k}_c \equiv \{k_1, k_2, \dots, k_N\}' \in R^N \quad (2.49)$$

$$\mathbf{k}_d \equiv \{k_{N+1}, k_{N+2}, \dots, k_M\}' \in R^{M-N} \quad (2.50)$$

and $N < M$. Define the diagonal matrix

$$\mathbf{L}_d \equiv \begin{bmatrix} L_{N+1} & 0 & \dots & 0 \\ 0 & L_{N+2} & \dots & 0 \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ 0 & 0 & \dots & L_M \end{bmatrix} \quad (2.51)$$

where L_i are the sequence lengths in the i^{th} dimension for $i = N + 1, N + 2, \dots, M$.

The partial MD DFT of $x(\mathbf{k})$, denoted by the operator $F_{N+1, M}^M []$ is defined by

$$F_{N+1,M}^M[x(\mathbf{k})] \triangleq \sum_{k_{N+1}=0}^{L_{N+1}-1} \sum_{k_{N+2}=0}^{L_{N+2}-1} \cdots \sum_{k_M=0}^{L_M-1} x(\mathbf{k}_c, \mathbf{k}_d) \quad (2.52)$$

$$\exp\left[-j\boldsymbol{\Omega}'_d(2\pi\mathbf{L}_d^{-1})\mathbf{k}_d\right] \in C$$

where $\boldsymbol{\Omega}_d$ is given by (2.8). The multiple-summation in (2.52) will be written, for brevity, as follows :

$$\sum_{k_{N+1}=0}^{L_{N+1}-1} \sum_{k_{N+2}=0}^{L_{N+2}-1} \cdots \sum_{k_M=0}^{L_M-1} \triangleq \sum_{\mathbf{k}_d=0}^{\mathbf{L}_d-1} \quad (2.53)$$

using the notation introduced in (2.40).

The above summations are over the variables \mathbf{k}_d and do not include the variables \mathbf{k}_c . It is therefore more convenient to write the *scalar function* $x(\mathbf{k}_c, \mathbf{k}_d)$ as an array $\mathbf{x}_c(\mathbf{k}_d) \in R^{L_1 \times L_2 \times \cdots \times L_N}$, having elements that are scalar functions of the M-N variables \mathbf{k}_d . This is done using the functional equivalences in (2.18), so that Equation (2.52) now becomes

$$F_{N+1,M}^M[x(\mathbf{k})] \triangleq \sum_{\mathbf{k}_d=0}^{\mathbf{L}_d-1} \mathbf{x}_c(\mathbf{k}_d) \exp[-j\boldsymbol{\Omega}'_d(2\pi\mathbf{L}_d^{-1})\mathbf{k}_d] \quad (2.54)$$

The expression on the RHS of (2.54) is evaluated by applying the (M-N)D DFT to each element $x_{c_{\mathbf{k}_c}}(\mathbf{k}_d)$ of the ND array $\mathbf{x}_c(\mathbf{k}_d)$, over the M-N variables \mathbf{k}_d . The

resulting ND array of (M-N)D DFT'ed sequences $X_{c_{\mathbf{k}_c}}(\Omega_{\mathbf{d}})$ can be written

$$\mathbf{X}_{\mathbf{c}}(\Omega_{\mathbf{d}}) \equiv \left\{ X_{c_{\mathbf{k}_c}}(\Omega_{\mathbf{d}}) \right\} = \left\{ F_{N+1,M}^M \left[x_{c_{\mathbf{k}_c}}(\mathbf{k}_{\mathbf{d}}) \right] \right\} \in C^{L_1 \times L_2 \times \cdots \times L_N} \quad (2.55)$$

Combining Equations (2.54) and (2.55) leads directly to

$$\begin{aligned} F_{N+1,M}^M[x(\mathbf{k})] &\triangleq \sum_{\mathbf{k}_{\mathbf{d}}=0}^{L_{\mathbf{d}}-1} x_{\mathbf{c}}(\mathbf{k}_{\mathbf{d}}) \exp[-j\Omega_{\mathbf{d}}'(2\pi L_{\mathbf{d}}^{-1})\mathbf{k}_{\mathbf{d}}] \triangleq \mathbf{X}_{\mathbf{c}}(\Omega_{\mathbf{d}}) \\ &= \left\{ F_{N+1,M}^M \left[x_{c_{\mathbf{k}_c}}(\mathbf{k}_{\mathbf{d}}) \right] \right\} \in C^{L_1 \times L_2 \times \cdots \times L_N} \end{aligned} \quad (2.56)$$

The partial MD DFT $\mathbf{X}_{\mathbf{c}}(\Omega_{\mathbf{d}})$ has the same dimensions as $x_{\mathbf{c}}(\mathbf{k}_{\mathbf{d}})$. Therefore, $\mathbf{X}_{\mathbf{c}}$ is an array in $C^{L_1 \times L_2 \times \cdots \times L_N}$ having complex elements and of size $L_1 \times L_2 \times \cdots \times L_N$.

The partial MD IDFT denoted by $F_{N+1,M}^{-M}[\]$ can be written as

$$\begin{aligned}
& F_{N+1,M}^{-M} \left[X(\mathbf{k}_c, \Omega_d) \right] \\
& \triangleq \frac{1}{\det(\mathbf{L}_d)} \sum_{\Omega_d=0}^{\mathbf{L}_d-1} \mathbf{X}_c(\Omega_d) \exp[j\Omega_d'(2\pi\mathbf{L}_d^{-1})\mathbf{k}_d] \\
& = \left\{ F_{N+1,M}^{-M} \left[X_{c_{\mathbf{k}_c}}(\Omega_d) \right] \right\} = \mathbf{x}_c(\mathbf{k}_d) \triangleq x(\mathbf{k})
\end{aligned} \tag{2.57}$$

In (2.57), $X(\mathbf{k}_c, \Omega_d)$ is a partially MD DFT'ed signal that can be expressed as the ND array $\mathbf{X}_c(\Omega_d)$ having elements that are scalar complex functions of Ω_d . The application of the partial MD IDFT $F_{N+1,M}^{-M} [\]$ to $\mathbf{X}_c(\Omega_d)$ results in the ND array $\mathbf{x}_c(\mathbf{k}_d)$ having elements that are scalar complex functions of \mathbf{k}_d . $\mathbf{x}_c(\mathbf{k}_d)$ can then be expressed as a scalar complex function $x(\mathbf{k})$ using the functional equivalences in (2.18).

As an example, consider the 3D scalar real function $x(k_1, k_2, k_3)$; $0 \leq k_i < 3$ for $i = 1, 2, 3$ (i.e. $\mathbf{k} = \{k_1, k_2, k_3\}'$). Let $\mathbf{k}_c = \{k_1, k_2\}'$ and $\mathbf{k}_d = \{k_3\}$. The function $x(k_1, k_2, k_3) \equiv x(\mathbf{k}_c, \mathbf{k}_d)$ can be expressed as a 2D array (or matrix) $\mathbf{x}_c(\mathbf{k}_d)$ using the functional equivalences in (2.18). That is,

$$x(\mathbf{k}) \equiv x(\mathbf{k}_c, \mathbf{k}_d) = x(k_1, k_2, k_3) \triangleq \mathbf{x}_c(k_3) \equiv \left\{ x_{c_{k_1, k_2}}(k_3) \right\} \tag{2.58}$$

This can be written in full as

$$x(\mathbf{k}) \equiv x(k_1, k_2, k_3) \triangleq \mathbf{x}_c(k_3) \equiv \begin{bmatrix} x_{c,0,0}(k_3) & x_{c,0,1}(k_3) & x_{c,0,2}(k_3) \\ x_{c,1,0}(k_3) & x_{c,1,1}(k_3) & x_{c,1,2}(k_3) \\ x_{c,2,0}(k_3) & x_{c,2,1}(k_3) & x_{c,2,2}(k_3) \end{bmatrix} \quad (2.59)$$

Now, the partial MD DFT, which in this case is simply the 1D DFT in the variable k_3 , is applied to each element of $\mathbf{x}_c(k_3)$ so that

$$F_{N+1,M}^M[x(\mathbf{k})] = F_{3,3}^3[x(k_1, k_2, k_3)] \triangleq \mathbf{X}_c(\Omega_3) \quad (2.60)$$

$$\equiv \begin{bmatrix} F_{3,3}^3[x_{c,0,0}(k_3)] & F_{3,3}^3[x_{c,0,1}(k_3)] & F_{3,3}^3[x_{c,0,2}(k_3)] \\ F_{3,3}^3[x_{c,1,0}(k_3)] & F_{3,3}^3[x_{c,1,1}(k_3)] & F_{3,3}^3[x_{c,1,2}(k_3)] \\ F_{3,3}^3[x_{c,2,0}(k_3)] & F_{3,3}^3[x_{c,2,1}(k_3)] & F_{3,3}^3[x_{c,2,2}(k_3)] \end{bmatrix}$$

where $F_{3,3}^3[\]$ is simply a 1D DFT on the variable k_3 . Thus, for the top left hand element of the matrix in (2.60),

$$F_{3,3}^3[x_{c,0,0}(k_3)] \triangleq \sum_{k_3=0}^2 x_{c,0,0}(k_3) e^{-j2\pi \frac{k_3 \Omega_3}{3}} = X_{c,0,0}(\Omega_3) \quad (2.61)$$

Similar operations are performed on the other elements of $\mathbf{X}_c(\Omega_3)$ leading directly to

$$\mathbf{X}_c(\Omega_3) \equiv \begin{bmatrix} X_{c,0,0}(\Omega_3) & X_{c,0,1}(\Omega_3) & X_{c,0,2}(\Omega_3) \\ X_{c,1,0}(\Omega_3) & X_{c,1,1}(\Omega_3) & X_{c,1,2}(\Omega_3) \\ X_{c,2,0}(\Omega_3) & X_{c,2,1}(\Omega_3) & X_{c,2,2}(\Omega_3) \end{bmatrix} \quad (2.62)$$

$$= F_{3,3}^3[x(k_1, k_2, k_3)]$$

which is the partial 3D DFT of $x(k_1, k_2, k_3)$ performed on the variable k_3 . Using the functional equivalences in (2.18), $\mathbf{X}_c(\Omega_3)$ can also be written as

$$\mathbf{X}_c(\Omega_3) \triangleq X(k_1, k_2, \Omega_3) \quad (2.63)$$

where $X(k_1, k_2, \Omega_3)$ is a scalar complex function of the spatial variables k_1, k_2 and the frequency variable Ω_3 .

In the next chapter, the CDFD filtering process is described using the notation developed in this chapter.

CHAPTER 3

CDFD FILTERING AND APPLICATIONS

3.1. Introduction

The notation developed in Chapter 2 is used to provide a detailed description of MD CDFD filtering in this chapter. The procedure used to calculate the output sequence $y(k)$ for an MD CDFD filter is described in Section 3.2. The frequency response of MD CDFD filters is derived in Section 3.3 and compared with the frequency response of MD DFT filters. The design of CDFD filters is described in Section 3.4. Various design considerations are examined and discussed. Finally, the advantages and disadvantages of CDFD filtering compared to conventional filtering techniques are discussed in Section 3.5.

3.2. Calculation of the Output Sequence $y(k)$ for CDFD filters

The first step in MD CDFD filtering is to obtain the partial MD DFT of the MD input signal $x(k)$, as given by (2.56). The expression $X_c(\Omega_d)$ in (2.56) is the partial MD DFT of $x(k)$ where the DFT is applied over the $(M-N)$ variables $k_d \equiv \{k_{N+1}, k_{N+2}, \dots, k_M\}'$. Using the functional equivalences in (2.18), $X_c(\Omega_d)$, which is an ND array of $(M-N)$ -variate complex sequences $X_{c, k_c}(\Omega_d)$, can be written as an $(M-N)D$ array $X_d(k_c)$ of N -variate complex sequences $X_{d, \Omega_d}(k_c)$, as

follows :

$$\begin{aligned} \mathbf{X}_{\mathbf{c}}(\boldsymbol{\Omega}_{\mathbf{d}}) &\equiv \left\{ X_{\mathbf{c}}(\boldsymbol{\Omega}_{\mathbf{d}}) \right\}_{\mathbf{k}_{\mathbf{c}}} \triangleq X(\mathbf{k}_{\mathbf{c}}, \boldsymbol{\Omega}_{\mathbf{d}}) \triangleq \mathbf{X}_{\mathbf{d}}(\mathbf{k}_{\mathbf{c}}) \equiv \left\{ X_{\mathbf{d}}(\mathbf{k}_{\mathbf{c}}) \right\}_{\boldsymbol{\Omega}_{\mathbf{d}}} \\ &= F_{N+1, M}^M[x(\mathbf{k})] \end{aligned} \quad (3.1)$$

The signals $\mathbf{X}_{\mathbf{c}}(\boldsymbol{\Omega}_{\mathbf{d}})$, $\mathbf{X}_{\mathbf{d}}(\mathbf{k}_{\mathbf{c}})$ and $X(\mathbf{k}_{\mathbf{c}}, \boldsymbol{\Omega}_{\mathbf{d}})$ in (3.1) are all periodically extended in the $(M-N)$ dimensions $\boldsymbol{\Omega}_{\mathbf{d}}$ because the partial MD DFT is applied over these dimensions. In the general MD DFT case, periodic extension is present in all M dimensions and gives rise to "edge effects" in the filtered output MD signal [3].

The second step in MD CDFD filtering is to apply an array filtering operator to the partial MD DFT transformed array $\mathbf{X}_{\mathbf{d}}(\mathbf{k}_{\mathbf{c}})$. This array filtering operator Φ consists of a set of LSI continuous-frequency ND filter operators $\phi_{\boldsymbol{\Omega}_{\mathbf{d}}}$, so that

$$\Phi \equiv \left\{ \phi_{\boldsymbol{\Omega}_{\mathbf{d}}} \right\} \equiv \left\{ \phi_{\Omega_{N+1}, \Omega_{N+2}, \dots, \Omega_M} \right\} \quad (3.2)$$

In (3.2), the array operator Φ must be of dimension $(M-N)$, the same as $\mathbf{X}_{\mathbf{d}}(\mathbf{k}_{\mathbf{c}})$.

The elemental LSI filter operators $\phi_{\boldsymbol{\Omega}_{\mathbf{d}}}$ are applied to each sequence in the N variables $\mathbf{k}_{\mathbf{c}}, X_{\mathbf{d}}(\mathbf{k}_{\mathbf{c}}, \Omega_{N+1}, \Omega_{N+2}, \dots, \Omega_M)$, over $\mathbf{k}_{\mathbf{c}}$. Typically,

$\phi_{\boldsymbol{\Omega}_{\mathbf{d}}}$ corresponds to an IIR or FIR filter having a corresponding ND impulse response $h_{\boldsymbol{\Omega}_{\mathbf{d}}}(k_1, k_2, \dots, k_N) \equiv h_{\boldsymbol{\Omega}_{\mathbf{d}}}(\mathbf{k}_{\mathbf{c}})$. The corresponding output of such

a filter is given by

$$Y_{d_{\Omega_d}}(\mathbf{k}_c) = h_{\Omega_d}(\mathbf{k}_c) * X_{d_{\Omega_d}}(\mathbf{k}_c) \quad (3.3)$$

where $*$ implies N-dimensional convolution over the N variables \mathbf{k}_c . In general,

$X_{d_{\Omega_d}}(\mathbf{k}_c)$ is complex, implying that the real and imaginary parts are filtered

separately to obtain the complex filtered output sequence $Y_{d_{\Omega_d}}(\mathbf{k}_c)$. The

complete (M-N)-dimensional array $\mathbf{Y}_d(\mathbf{k}_c) \equiv \{Y_{d_{\Omega_d}}(\mathbf{k}_c)\}$ is therefore given by

$$\mathbf{Y}_d(\mathbf{k}_c) = \left\{ h_{\Omega_d}(\mathbf{k}_c) * X_{d_{\Omega_d}}(\mathbf{k}_c) \right\} \quad (3.4)$$

which, for brevity, can be written in terms of the element-wise array convolution operator as

$$\mathbf{Y}_d(\mathbf{k}_c) = \mathbf{h}_d(\mathbf{k}_c) *_{M-N}^N X_d(\mathbf{k}_c) \quad (3.5)$$

where

$$\mathbf{h}_d(\mathbf{k}_c) \equiv \left\{ h_{\Omega_d}(\mathbf{k}_c) \right\}$$

This describes the (M-N)-dimensional set of ND convolution operations, where \mathbf{Y}_d ,

$\mathbf{X}_d \in C^{L_{N+1} \times L_{N+2} \times \cdots \times L_M}$, $\mathbf{h}_d \in R^{L_{N+1} \times L_{N+2} \times \cdots \times L_M}$ and \mathbf{Y}_d , \mathbf{X}_d ,

\mathbf{h}_d are all periodically extended in the (M-N) variables Ω_d . Following the

notation in (2.18), $Y_{\mathbf{d}}(\mathbf{k}_{\mathbf{c}})$ can be written in alternate array representations as

$$Y_{\mathbf{d}}(\mathbf{k}_{\mathbf{c}}) \triangleq Y(\Omega_{\mathbf{d}}, \mathbf{k}_{\mathbf{c}}) \triangleq Y_{\mathbf{c}}(\Omega_{\mathbf{d}}) \equiv \left\{ Y_{c_{\mathbf{k}_{\mathbf{c}}}}(\Omega_{\mathbf{d}}) \right\} \quad (3.6)$$

where Y is a complex M -variate scalar function and $Y_{\mathbf{c}} \in C^{L_1 \times L_2 \times \cdots \times L_N}$ is an N -dimensional array having elements $Y_{c_{\mathbf{k}_{\mathbf{c}}}}(\Omega_{\mathbf{d}})$, that are scalar complex functions of the $(M-N)$ variables $\Omega_{\mathbf{d}}$.

The third and final step of MD CDFD filtering is to obtain the inverse partial DFT of $Y_{\mathbf{c}}(\Omega_{\mathbf{d}})$ over the frequencies $\Omega_{\mathbf{d}}$, given by

$$\begin{aligned} & F_{N+1, M}^{-M} \left[Y_{\mathbf{c}}(\Omega_{\mathbf{d}}) \right] \\ & \triangleq \frac{1}{|\mathbf{L}_{\mathbf{d}}|} \sum_{\Omega_{\mathbf{d}}=0}^{\mathbf{L}_{\mathbf{d}}-1} Y_{\mathbf{c}}(\Omega_{\mathbf{d}}) \exp[j \Omega_{\mathbf{d}}' (2\pi \mathbf{L}_{\mathbf{d}}^{-1}) \mathbf{k}_{\mathbf{d}}] \in C^{L_1 \times L_2 \times \cdots \times L_N} \\ & = \left\{ F_{N+1, M}^{-M} \left[Y_{c_{\mathbf{k}_{\mathbf{c}}}}(\Omega_{\mathbf{d}}) \right] \right\} \quad (3.7) \\ & = y_{\mathbf{c}}(\mathbf{k}_{\mathbf{d}}) \equiv \left\{ y_{c_{\mathbf{k}_{\mathbf{c}}}}(\mathbf{k}_{\mathbf{d}}) \right\} \in C^{L_1 \times L_2 \times \cdots \times L_N} \end{aligned}$$

[This operation corresponds to the *conventional* M dimensional inverse DFT for $N=0$.] Thus, using the notation in (2.18), the final output sequence is defined as

$$\tilde{y}(\mathbf{k}) \triangleq y_{\mathbf{c}}(\mathbf{k}_{\mathbf{d}}) = F_{N+1,M}^{-M} \left[Y_{\mathbf{c}}(\Omega_{\mathbf{d}}) \right] \quad (3.8)$$

The three steps for CDFD filtering, as described in equations (3.1), (3.5) and (3.7) may be combined to give

$$\tilde{y}(\mathbf{k}) \triangleq y_{\mathbf{c}}(\mathbf{k}_{\mathbf{d}}) = F_{N+1,M}^{-M} \left[\mathbf{h}(\mathbf{k}_{\mathbf{c}}) *_{M-N}^N \left[F_{N+1,M}^M [x(\mathbf{k})] \right] \right] \quad (3.9)$$

which is the CDFD input/output equation, and can be written in full as

$$\begin{aligned} \tilde{y}(\mathbf{k}) &\triangleq y_{\mathbf{c}}(\mathbf{k}_{\mathbf{d}}) \\ &= \frac{1}{|\mathbf{L}_{\mathbf{d}}|} \sum_{\Omega_{\mathbf{d}}=0}^{\mathbf{L}_{\mathbf{d}}-1} \left[\mathbf{h}(\mathbf{k}_{\mathbf{c}}) *_{M-N}^N \left[\sum_{\mathbf{k}_{\mathbf{d}}=0}^{\mathbf{L}_{\mathbf{d}}-1} x(\mathbf{k}_{\mathbf{c}}, \mathbf{k}_{\mathbf{d}}) \exp[-j \Omega'_{\mathbf{d}} (2\pi \mathbf{L}_{\mathbf{d}}^{-1}) \mathbf{k}_{\mathbf{d}}] \right] \right] \\ &\quad \cdot \exp[j \Omega'_{\mathbf{d}} (2\pi \mathbf{L}_{\mathbf{d}}^{-1}) \mathbf{k}_{\mathbf{d}}] \end{aligned} \quad (3.10)$$

The sequence $\tilde{y}(\mathbf{k})$ is the MD periodically extended output signal of the CDFD filter. It is important to note that $\tilde{y}(\mathbf{k})$ is periodically extended *only in the dimensions* $k_{N+1}, k_{N+2}, \dots, k_M$ *over which the DFT is applied.* The final output sequence $y(\mathbf{k})$ is that part of $\tilde{y}(\mathbf{k})$ having support in \mathcal{R}^M where

$$\begin{aligned} \mathcal{R}^M &= \{ \mathbf{k} : 0 \leq k_i \leq L_i - 1, i = 1, 2, \dots, M ; L_i \text{ integer}, L_i < \infty \\ &\quad \text{for } i = N + 1, N + 2, \dots, M \} \end{aligned} \quad (3.11)$$

\mathcal{R}^M is similar to the region of support \mathcal{R}^M for the conventional DFT case.

However, for CDFD filtering, the region of support \mathcal{R}^M does not require that L_i be *integer* for $i \leq N$ because the partial (M-N)D DFT is not evaluated over the dimensions k_i for $i \leq N$. This more relaxed region of support \mathcal{R}^M eliminates troublesome "edge effects" in the N dimensions \mathbf{k}_c unlike the conventional DFT case where edge effects are present in all M dimensions.

An interesting special case occurs for $N = 0$, that is, for discrete-frequency filtering of *all* M variables. In this case, each element of the array $\mathbf{h}(\mathbf{k}_c)$ in (3.10) corresponds to $h_{\Omega_d}(\mathbf{k}_c)$ in (3.3) and is equal to constants p_{Ω_d} where p_{Ω_d} is a real number. Then, from (3.3)

$$Y_{d_{\Omega_d}}(\mathbf{k}_c) = p_{\Omega_d} X_{d_{\Omega_d}}(\mathbf{k}_c) \quad (3.12)$$

Therefore, with $\mathbf{p} \equiv \{p_{\Omega_d}\} \in R^{L_1 \times L_2 \times \dots \times L_M}$ and $\mathbf{k}_c \equiv 0$, it follows that

$$\mathbf{Y}_d = \mathbf{p} \cdot_M \mathbf{X}_d \quad (3.13)$$

From Equations (2.32) and (2.33) it is easily shown that (3.13) is the same as Equation (2.44), implying that CDFD filtering is equivalent to conventional MD DFT filtering for the special case when $N = 0$.

3.2.1. Example: The 3D Case with $N = 1$

The 2D partial DFT is applied to the input signal $x(\mathbf{k}) = x(k_1, k_2, k_3)$ with $\mathbf{k}_c = \{k_1\}'$ and $\mathbf{k}_d = \{k_2, k_3\}'$, so that the output of the DFT process is given by

$$\begin{aligned}
X(\mathbf{k}_c, \Omega_d) &= X(k_1, \Omega_2, \Omega_3) \\
&= F_{2,3}^3 [x(k_1, k_2, k_3)] \\
&= \sum_{k_2=0}^{L_2-1} \sum_{k_3=0}^{L_3-1} x(k_1, k_2, k_3) \exp \left[-j2\pi \left(\frac{k_2 \Omega_2}{L_2} + \frac{k_3 \Omega_3}{L_3} \right) \right]
\end{aligned} \tag{3.14}$$

where $\Omega_d = \{\Omega_2, \Omega_3\}'$. This is represented by the first of the three blocks in Fig. 3.1. The signal $X(\mathbf{k}_c, \Omega_d) = X(k_1, \Omega_2, \Omega_3)$ may be thought of as a set of one-dimensional ($N = 1$) complex sequences where each sequence corresponds to a particular discrete pair Ω_2, Ω_3 . Thus, there are $L_2 L_3$ such complex sequences having support in R^3 . $X(\mathbf{k}_c, \Omega_d)$ can be expressed in its alternate array representations, according to (3.1), as follows :

$$\mathbf{X}_c(\Omega_2, \Omega_3) \equiv \left\{ X_{c_{k_1}}(\Omega_2, \Omega_3) \right\} \triangleq X(k_1, \Omega_2, \Omega_3) \triangleq \mathbf{X}_d(k_1) \equiv \left\{ X_{d_{\Omega_2 \Omega_3}}(k_1) \right\} \tag{3.15}$$

The second step in the filtering process is to apply a two-dimensional ($M-N = 2$) array filtering operator $\Phi[] \equiv \{\phi_{\Omega_2 \Omega_3}[]\}$ to the 2D partial DFT array $\mathbf{X}_d(k_1)$. Each element $\phi_{\Omega_2 \Omega_3}[]$ of $\Phi[]$ is an LSI continuous-frequency one-dimensional ($N = 1$) filter operator which, in this example, is a 1D IIR filter operation on the variable k_1 of each elemental sequence $X_{d_{\Omega_2 \Omega_3}}(k_1)$ of $\mathbf{X}_d(k_1)$. The second step can therefore be written in terms of the filtering operator $\Phi[]$ as follows :

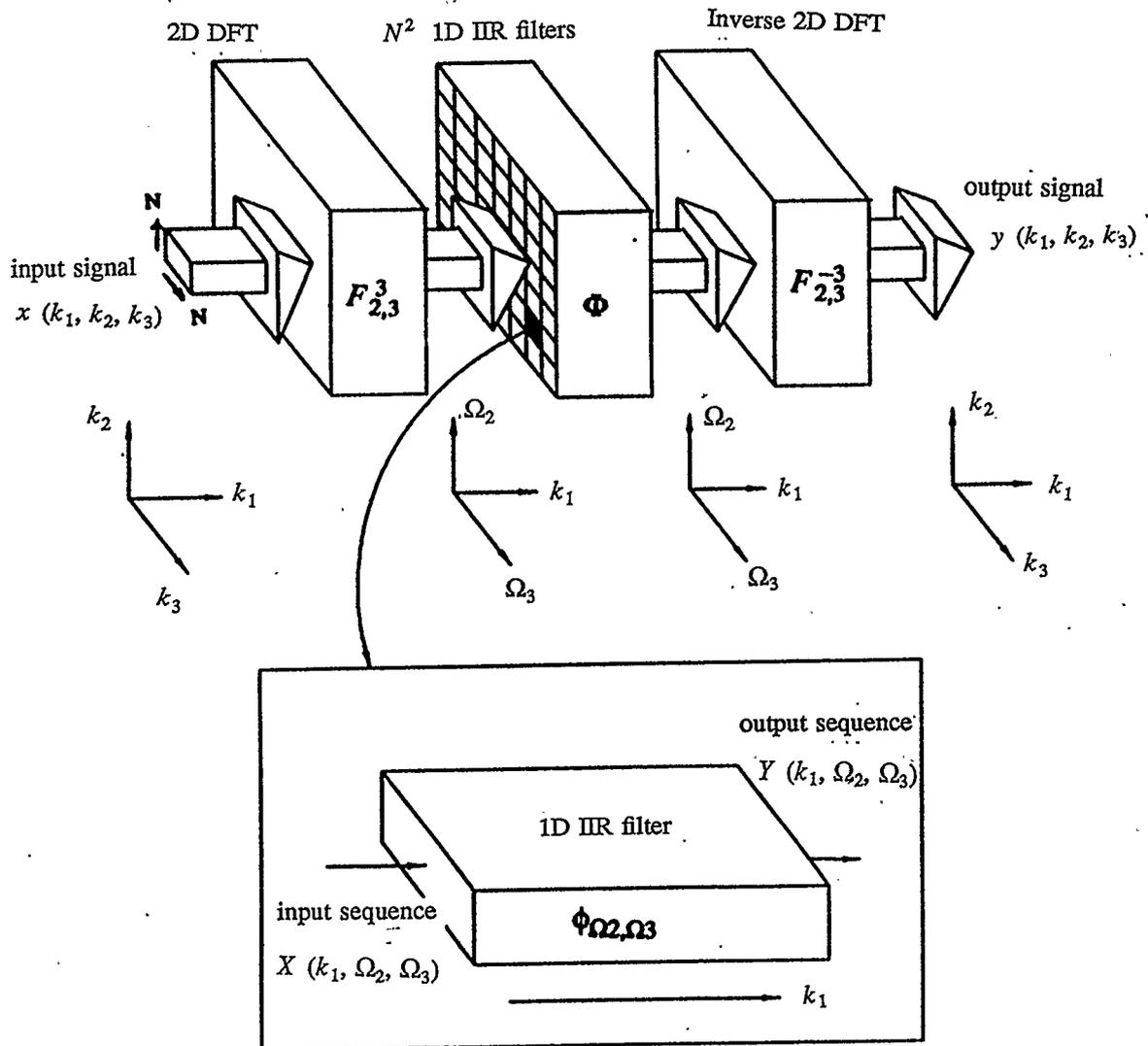


Fig. 3.1. Block diagram of a 3D CDFD filter.

$$Y_{\mathbf{d}}(k_1) \equiv \left\{ Y_{d_{\Omega_2\Omega_3}}(k_1) \right\} = \Phi[X_{\mathbf{d}}(k_1)] \equiv \left\{ \phi_{\Omega_2\Omega_3} \left[X_{d_{\Omega_2\Omega_3}}(k_1) \right] \right\} \quad (3.16)$$

Each elemental filtering operation, denoted by the $\phi_{\Omega_2\Omega_3} [X_{d_{\Omega_2\Omega_3}}(k_1)]$ term of

(3.16), can be written as a linear difference equation (LDE) of the form

$$Y_{d_{\Omega_2\Omega_3}}(k_1) = \sum_{p=0}^{P-1} a_{p_{\Omega_2\Omega_3}} X_{d_{\Omega_2\Omega_3}}(k_1 - p) - \sum_{q=1}^{Q-1} b_{q_{\Omega_2\Omega_3}} Y_{d_{\Omega_2\Omega_3}}(k_1 - q) \quad (3.17)$$

The coefficients $a_{p_{\Omega_2\Omega_3}}$ and $b_{q_{\Omega_2\Omega_3}}$ are functions of Ω_2, Ω_3 . The frequency

response corresponding to (3.17) is given by

$$H_{d_{\Omega_2\Omega_3}}(\Omega_1) = \frac{\sum_{p=0}^{P-1} a_{p_{\Omega_2\Omega_3}} \exp \left[-j2\pi \left[\frac{p\Omega_1}{L_1} \right] \right]}{1 + \sum_{q=1}^{Q-1} b_{q_{\Omega_2\Omega_3}} \exp \left[-j2\pi \left[\frac{q\Omega_1}{L_1} \right] \right]} \quad (3.18)$$

having a corresponding unit impulse response $h_{d_{\Omega_2\Omega_3}}(k_1)$. Equation (3.17) may

therefore be written in the alternate form

$$Y_{d_{\Omega_2\Omega_3}}(k_1) = X_{d_{\Omega_2\Omega_3}}(k_1) * h_{d_{\Omega_2\Omega_3}}(k_1) \quad (3.19)$$

where * represents 1D convolution in the variable k_1 . Equation (3.19) represents

an elemental convolution that can be written in the array notation of (2.33) so that

$$\begin{aligned} \mathbf{Y}_{\mathbf{d}}(\mathbf{k}) &\equiv \left\{ Y_{d_{\Omega_2 \Omega_3}}(k_1) \right\} = \mathbf{X}_{\mathbf{d}}(k_1) * \frac{1}{3} \mathbf{h}_{\mathbf{d}}(k_1) \\ &\equiv \left\{ X_{d_{\Omega_2 \Omega_3}}(k_1) * h_{d_{\Omega_2 \Omega_3}}(k_1) \right\} \end{aligned} \quad (3.20)$$

where $*$ is the 1D convolution operator in k_1 . The second step is represented by the second block and the inset in Fig. 3.1.

Using the functional equivalences in (3.1), $\mathbf{Y}_{\mathbf{d}}(k_1)$ can be written as

$$\begin{aligned} \mathbf{Y}_{\mathbf{d}} &\equiv \left\{ Y_{d_{\Omega_2 \Omega_3}}(k_1) \right\} \triangleq \mathbf{Y}_{\mathbf{c}}(\Omega_2, \Omega_3) \equiv \left\{ Y_{c_{k_1}}(\Omega_2, \Omega_3) \right\} \\ &\triangleq Y(k_1, \Omega_2, \Omega_3) \end{aligned} \quad (3.21)$$

The third and final step is to obtain the partial 3D IDFT of $\mathbf{Y}_{\mathbf{c}}(\Omega_2, \Omega_3)$ with respect to the variables Ω_2, Ω_3 and is represented by the third block in Fig. 3.1.

The output of this operation is given by

$$\begin{aligned} \hat{y}(\mathbf{k}_{\mathbf{c}}, \mathbf{k}_{\mathbf{d}}) &= \hat{y}(k_1, k_2, k_3) \\ &= F_{2,3}^{-3} [Y(k_1, \Omega_2, \Omega_3)] \\ &= \frac{1}{L_2 L_3} \sum_{\Omega_2=0}^{L_2-1} \sum_{\Omega_3=0}^{L_3-1} Y(k_1, \Omega_2, \Omega_3) \exp \left[j 2\pi \left[\frac{k_2 \Omega_2}{L_2} + \frac{k_3 \Omega_3}{L_3} \right] \right] \end{aligned} \quad (3.22)$$

This signal is periodically extended in the dimensions k_2 and k_3 , with periods L_2 and L_3 , respectively. The output $y(\mathbf{k}) = y(k_1, k_2, k_3)$ is that part of $\tilde{y}(k_1, k_2, k_3)$ having support in

$$\mathbb{R}^3 = \{\mathbf{k} : 0 \leq k_i \leq L_i - 1, i = 1, 2, 3; L_i \text{ integer for } i = 2,3\}$$

3.3. Steady-State Frequency Response of CDFD Filters

In order to derive an expression for the steady-state frequency response of a CDFD filter, it is useful to express MD signals that are functions of both frequency *and* spatial variables as M-variate functions (e.g. $x(\mathbf{k}_c, \mathbf{\Omega}_d)$) where N of the M variables are spatial variables (\mathbf{k}_c) and the rest of the variables are frequency variables ($\mathbf{\Omega}_d$). Thus, the alternate array representation of MD signals will not be used in this section.

The calculation of the overall steady-state input-output frequency response of MD CDFD filters requires that the input signal $x(\mathbf{k})$ be bounded in only those dimensions $k_{N+1}, k_{N+2}, \dots, k_M$ over which the partial MD DFT and MD IDFT are applied. (A similar assumption must be made for all M dimensions to calculate the frequency response for conventional MD DFT filters). However, $x(\mathbf{k})$ may be spatially unbounded for the dimensions k_1, k_2, \dots, k_N over which continuous-frequency filtering occurs.

The impulse response of an MD CDFD filter is expressed as an (M-N)D array of N-variate functions in Equation (3.5). Using the functional equivalences in

(3.1), $\mathbf{h}_d(\mathbf{k}_c)$ can also be expressed as an M-variate function $h(\mathbf{k}_c, \Omega_d)$ of N spatial variables \mathbf{k}_c and (M-N) frequency variables Ω_d . Thus,

$$\mathbf{h}_d(\mathbf{k}_c) \equiv \left\{ h_d \left(\mathbf{k}_c \right) \right\}_{\Omega_d} \triangleq h(\mathbf{k}_c, \Omega_d) \quad (3.23)$$

The steady-state frequency response of an MD CDFD filter $H(\Omega_c, \Omega_d)$, can be obtained by applying the partial MD DFT to the impulse response of the filter $h(\mathbf{k}_c, \Omega_d)$, on the variables \mathbf{k}_c , as follows :

$$\begin{aligned} H(\Omega_c, \Omega_d) &= F_{1,N}^M[h(\mathbf{l}_c, \Omega_d)] \\ &= \sum_{l_1=0}^{L_1-1} \sum_{l_2=0}^{L_2-1} \cdots \sum_{l_N=0}^{L_N-1} h(\mathbf{l}_c, \Omega_d) \exp \left[-j 2\pi \left[\sum_{i=1}^N \frac{\Omega_i l_i}{L_i} \right] \right] \end{aligned} \quad (3.24)$$

where $\mathbf{l}_c = \{l_1, l_2, \dots, l_N\}'$ is used instead of \mathbf{k}_c for convenience.

The steady state frequency response $H(\Omega_c, \Omega_d)$ is the ratio of the response $\tilde{y}(\mathbf{k})$ to the input $\tilde{x}(\mathbf{k})$, where the input is an exponential of the form

$$\tilde{x}(\mathbf{k}) \equiv \exp \left[j 2\pi \left[\sum_{i=1}^M \frac{k_i \Omega_i}{L_i} \right] \right] \quad (3.25)$$

and Ω_i is an integer for all i and L_i is an integer for $i = N + 1, N + 2, \dots, M$.

The above constraints on L_i are required to ensure that $x(\mathbf{k})$, which is defined as the part of $\tilde{x}(\mathbf{k})$ in \mathcal{R}^M , has no discontinuities at the boundaries of \mathcal{R}^M , thereby ensuring (as in the conventional MD DFT case) that $X(\mathbf{k}_c, \Omega_d)$, $Y(\mathbf{k}_c, \Omega_d)$ and

$\tilde{y}(\mathbf{k})$ are periodic. If L_i , $i = N + 1, N + 2, \dots, M$ are permitted to be non-integers, then spectral leakage effects will occur in $X(\mathbf{k}_c, \mathbf{\Omega}_d)$, $Y(\mathbf{k}_c, \mathbf{\Omega}_d)$ and $\tilde{y}(\mathbf{k})$ due to discontinuities in $x(\mathbf{k})$ at the boundaries of \mathbf{R}^M , *exactly as for the conventional MD DFT case.*

After the first step in the MD CDFD filtering technique, the input to the IIR filter is the partial MD DFT of $x(\mathbf{k})$ given by

$$X(\mathbf{k}_c, \mathbf{\Omega}_d) = F_{N+1, M}^M[x(\mathbf{k})] = \delta(\mathbf{\Omega}_d - \mathbf{\Omega}_{d_0}) |\mathbf{L}_d| \exp \left[j 2\pi \left\{ \sum_{i=1}^N \frac{k_i \Omega_i}{L_i} \right\} \right] \quad (3.26)$$

which is an *unbounded* N-dimensional complex exponential sequence. The (M-N)D delta function $\delta(\mathbf{\Omega}_d - \mathbf{\Omega}_{d_0})$ indicates that the energy of the transformed signal is present at the point $\mathbf{\Omega}_d = \mathbf{\Omega}_{d_0} \in \mathbf{R}^M$. The (M-N)D delta function is defined as follows :

$$\delta(\mathbf{\Omega}_d - \mathbf{\Omega}_{d_0}) \equiv \delta(\Omega_{N+1} - \Omega_{N+1_0}, \Omega_{N+2} - \Omega_{N+2_0}, \dots, \Omega_M - \Omega_{M_0}) \quad (3.27)$$

The second step in the filtering process is to filter each $X(\mathbf{k}_c, \mathbf{\Omega}_d)$ using an N-dimensional IIR filter. The output of each IIR filter $Y(\mathbf{k}_c, \mathbf{\Omega}_d)$ can be expressed as the N dimensional convolution of the input sequence $X(\mathbf{k}_c, \mathbf{\Omega}_d)$ with the unit impulse response of the IIR filter $h(\mathbf{k}_c, \mathbf{\Omega}_d)$ as follows :

$$\begin{aligned}
Y(\mathbf{k}_c, \Omega_d) &= X(\mathbf{k}_c, \Omega_d) * h(\mathbf{k}_c, \Omega_d) \\
&= \sum_{l_1=0}^{L_1-1} \sum_{l_2=0}^{L_2-1} \cdots \sum_{l_N=0}^{L_N-1} X(\mathbf{k}_c - \mathbf{l}_c, \Omega_d) h(\mathbf{l}_c, \Omega_d) \quad (3.28)
\end{aligned}$$

where * represents ND convolution over the variables \mathbf{k}_c ,

$$\mathbf{l}_c \equiv \{l_1, l_2, \dots, l_N\}$$

and

$$\mathbf{k}_c - \mathbf{l}_c \equiv \{k_1 - l_1, k_2 - l_2, \dots, k_N - l_N\}$$

Substituting (3.26) in (3.28) gives

$$\begin{aligned}
Y(\mathbf{k}_c, \Omega_d) &= \sum_{l_1=0}^{L_1-1} \sum_{l_2=0}^{L_2-1} \cdots \sum_{l_N=0}^{L_N-1} \delta(\Omega_d - \Omega_{d_0}) |\mathbf{L}_d| \exp \left[j 2\pi \left[\sum_{i=1}^N \frac{k_i \Omega_i}{L_i} \right] \right] \\
&\quad \cdot \exp \left[-j 2\pi \left[\sum_{i=1}^N \frac{\Omega_i l_i}{L_i} \right] \right] h(\mathbf{l}_c, \Omega_d) \\
&= \delta(\Omega_d - \Omega_{d_0}) |\mathbf{L}_d| \exp \left[j 2\pi \left[\sum_{i=1}^N \frac{k_i \Omega_i}{L_i} \right] \right] \\
&\quad \cdot \sum_{l_1=0}^{L_1-1} \sum_{l_2=0}^{L_2-1} \cdots \sum_{l_N=0}^{L_N-1} h(\mathbf{l}_c, \Omega_d) \exp \left[-j 2\pi \left[\sum_{i=1}^N \frac{\Omega_i l_i}{L_i} \right] \right] \quad (3.29)
\end{aligned}$$

Substituting (3.24) and (3.26) in (3.29) gives

$$Y(\mathbf{k}_c, \Omega_d) = X(\mathbf{k}_c, \Omega_d)H(\Omega_c, \Omega_d) \quad (3.30)$$

The third and final step in the filtering process is to obtain the partial MD IDFT of (3.30) which leads directly to

$$\begin{aligned} \tilde{y}(\mathbf{k}) = & \frac{1}{|\mathbf{L}_d|} \sum_{\Omega_{N+1}=0}^{L_{N+1}-1} \sum_{\Omega_{N+2}=0}^{L_{N+2}-1} \cdots \sum_{\Omega_M=0}^{L_M-1} |\mathbf{L}_d| \exp \left[j2\pi \left[\sum_{i=1}^N \frac{k_i \Omega_i}{L_i} \right] \right] \\ & \cdot \delta(\Omega_d - \Omega_{d_0}) H(\Omega_c, \Omega_d) \exp \left[j2\pi \left[\sum_{i=N+1}^M \frac{k_i \Omega_i}{L_i} \right] \right] \end{aligned} \quad (3.31)$$

Rearranging terms in (3.31) and substituting (3.25) gives

$$\tilde{y}(\mathbf{k}) = H(\Omega_c, \Omega_{d_0}) \tilde{x}(\mathbf{k}) \quad (3.32)$$

Thus, for any given value of $\Omega_d \in \mathbb{R}^M$, Equation (3.32) can be written as

$$\tilde{y}(\mathbf{k}) = H(\Omega_c, \Omega_d) \tilde{x}(\mathbf{k}) . \quad (3.33)$$

Rearranging terms in (3.33) leads directly to the following expression for the steady-state frequency response of an MD CDFD filter. That is, with $\tilde{x}(\mathbf{k})$ as given in Equation (3.25),

$$H(\Omega_c, \Omega_d) = \frac{\tilde{y}(\mathbf{k})}{\tilde{x}(\mathbf{k})} \quad (3.34)$$

Consequently, the *frequency responses of the set of IIR filters uniquely determine the overall input-output frequency response of the CDFD filter.*

The above frequency response $H(\Omega_c, \Omega_d)$ is periodically extended only in the dimensions Ω_d because the DFTs are applied over these dimensions. It should be noted that the troublesome "edge effects" that can occur at the boundaries of \mathcal{R}^M (due to discontinuities of $\tilde{x}(\mathbf{k})$) do not occur in the dimensions k_1, k_2, \dots, k_N over which continuous-frequency filtering is applied.

The periodic extension of the frequency response of DFT and CDFD filters is illustrated in Fig. 3.2. The periodic extension of the frequency response of a 3D DFT cone filter is shown in Fig. 3.2a. The Ω_3 axis has been enlarged for clarification and the dots describe the region in which the gain of the ideal cone filter is unity. The frequency response is repeated in all three directions $\Omega_1, \Omega_2, \Omega_3$ with periods L_1, L_2, L_3 respectively, because the DFT is applied in all three dimensions $\Omega_1, \Omega_2, \Omega_3$.

The corresponding 3D CDFD cone filter, employing continuous-frequency (IIR) filtering in the ω_1 dimension and discrete-frequency (DFT) filtering in the Ω_2 and Ω_3 dimensions, is shown in Fig. 3.2b. No periodic extension is present in ω_1 direction because continuous frequency (e.g. IIR) filtering is applied in this dimension. The parallel lines represent a passband gain of approximately unity. In this case, the continuous-discrete frequency response is repeated only in the Ω_2 and Ω_3 dimensions, with periods L_2 and L_3 respectively, because DFT filtering is applied only over these two dimensions. The absence of periodic extension in the

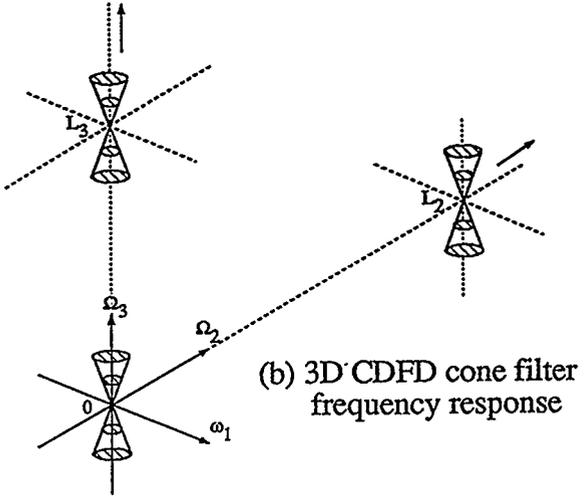
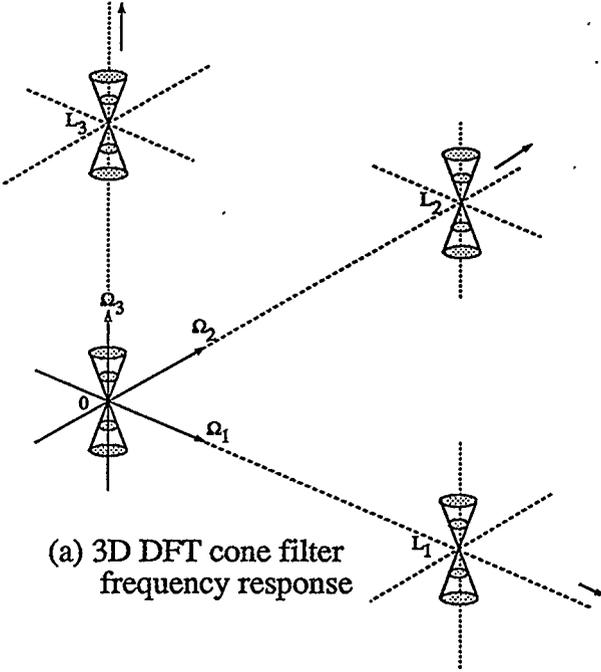


Fig. 3.2. Periodic extension of frequency response

ω_1 dimension eliminates "edge effects" in this dimension. This constitutes one of the advantages inherent in CDFD filtering.

3.4. Design of CDFD Filters

The design of CDFD filters from a given set of specifications is described here with the aid of illustrative examples. The design procedure and the examples are based on CDFD filters that employ continuous-frequency filtering in one dimension and discrete-frequency filtering in the remaining dimensions. This type of CDFD filter can be designed easily because the one-dimensional IIR filters employed are easy to design and stability issues are well understood. It is shown that this approach has the added advantage that any passband shape, limited by the frequency responses of the 1D IIR filter sections, can be implemented.

3.4.1. Design Considerations

The design of CDFD filters involves finding a suitable approximation to a specified "ideal" frequency response. This approach is somewhat similar to the design of MD IIR filters because the one-dimensional continuous-frequency filters used in the CDFD technique are 1D IIR filters.

The required filter order for the 1D IIR filters is a function of the specified passband shape and the transition region from the passband to stopband. In general, MD CDFD filters employ lower filter orders than their MD IIR counterparts, which results in computational savings. When choosing the filter

order and the passband and stopband widths, it is important to ensure that desired signal frequencies lie well within the passband of the 1D IIR filters. If the desired signal frequencies lie in the transition region of the 1D IIR filters, then the output of the CDFD filter will have a large imaginary component because the maximum phase shift occurs in the transition region of an IIR filter. This is undesirable because the output of the CDFD filter has an imaginary component that must be eliminated by techniques, such as two-pass filtering, which result in high computational overheads. An appropriate choice of filter order and passband width for the 1D IIR filters allows the imaginary component of the output of the CDFD filter to be neglected.

For example, one quadrant of the frequency response of a 2D CDFD circularly symmetric lowpass filter is shown in Fig. 3.3. Discrete-frequency filtering (DFT) is applied in the Ω_1 dimension and continuous-frequency filtering is applied in the ω_2 dimension. This implies that a set of 1D IIR filters, one for each value of Ω_1 , is used to approximate the 2D frequency response. From straightforward geometric considerations, it is clear that these 1D IIR filters must be lowpass filters.

The difference between a good design (Fig. 3.3a) and a poor design (Fig. 3.3b) is illustrated in Fig. 3.3. The dots represent a desired set of frequencies. In Fig. 3.3a, the desired signal lies well within the passband of 2D CDFD filter and hence, in the passband of the 1D IIR filter used to implement it (inset). In Fig.

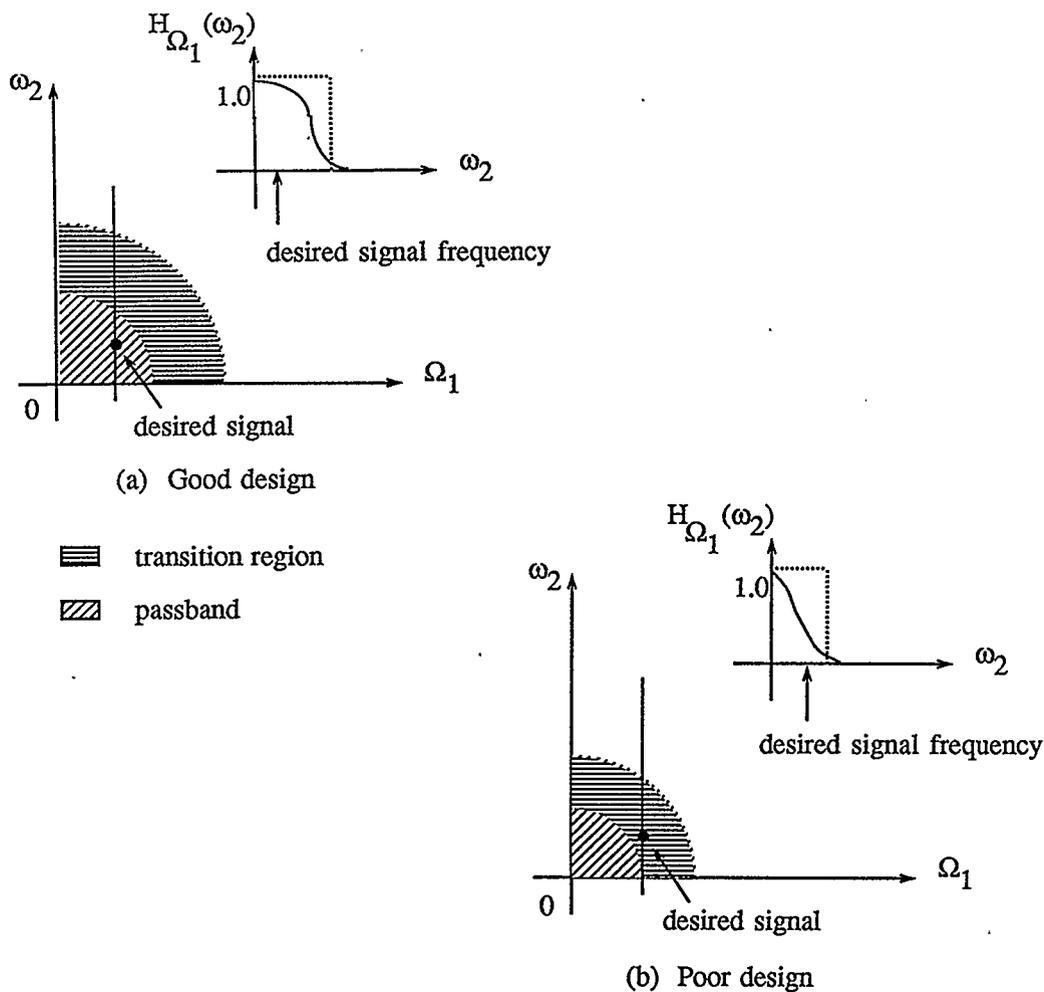


Fig. 3.3. One quadrant of the frequency response of a circularly symmetric lowpass 2D CDFD filter with continuous frequency (IIR) filtering in the ω_2 dimension. The insets show the 1D IIR lowpass filter frequency response at the cross-section.

3.3b, the desired signal lies in the transition region of the CDFD filter giving rise to undesirable phase shifts and, consequently, a large undesirable imaginary component at the output of the CDFD filter.

It is generally desirable to use the same filter order for the set of 1D IIR filters because this simplifies hardware implementation of the resulting CDFD filters. An advantage of the CDFD technique that makes it suitable for hardware implementation is the fact that in some cases the same 1D IIR filter may be used more than once to implement the CDFD filter. Furthermore, for some values of the discrete-frequency variables, no 1D IIR filter need be implemented when the gain of the filter is required to be zero, resulting in computational savings and reduced hardware complexity.

These design considerations are demonstrated in Fig. 3.4 which shows the circularly symmetric 2D CDFD lowpass filter in 2D frequency space. It is clear from symmetry, that the set of 1D IIR filters used at $\Omega_1 \leq \Omega_{1_0}$ are the same filters required at $\Omega_1 \geq (L_1 - \Omega_{1_0} - 1)$.

It is also clear that for $\Omega_{1_0} < \Omega_1 < (L_1 - \Omega_{1_0} - 1)$, no 1D IIR filters are required due to the zero passband specified for these frequencies. In the next section, a summary of the design procedure is described for MD CDFD filters employing continuous-frequency IIR filtering in one signal dimension and discrete-frequency filtering in the remaining signal dimensions.

3.4.2. Design Procedure for MD CDFD Filters

The design procedure for MD CDFD filters is listed below. It is assumed that continuous-frequency filtering is employed in only one dimension and discrete-

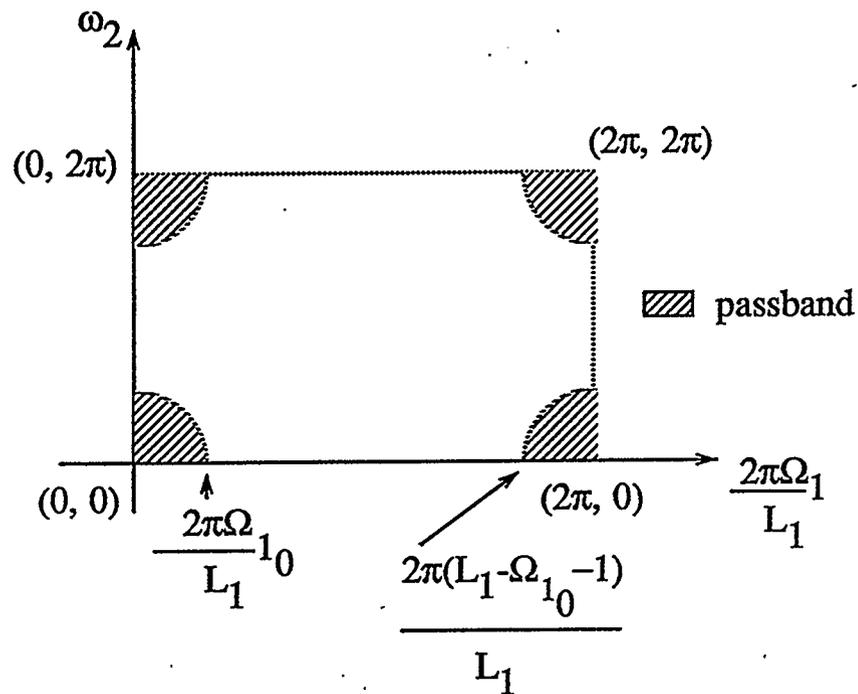


Fig. 3.4. Symmetry in passband of circularly symmetric 2D CDFD lowpass filter.

frequency filtering is employed in the remaining $(M-1)$ dimensions.

1. *The specified MD frequency response is examined for symmetry.* By choosing the proper dimension for continuous-frequency IIR filtering, utilising the symmetry of the MD frequency response, the overall hardware requirements for the IIR filters can be reduced to a minimum.

2. After dimension ω_1 for IIR filtering has been chosen, the *specified MD frequency response is projected on to this dimension for each value of the discrete-frequency variables $\Omega_2, \Omega_3, \dots, \Omega_M$* . This results in a set of filter specifications for a total of $L_2 \times L_3 \times \dots \times L_M$ 1D IIR filters.
3. *The required $L_2 \times L_3 \times \dots \times L_M$ 1D IIR filters are designed based on the specifications obtained in (2).*
 - a) It is desirable to use the same filter order for all 1D IIR filters.
 - b) 1D IIR filters may not be required for all values of $\Omega_2, \Omega_3, \dots, \Omega_M$. At some values an "all go" function may be required while at other values an "all stop" function may be required. The set of filters may therefore be completely bypassed (all go) or the output for a given value of $\Omega_2, \Omega_3, \dots, \Omega_M$ may simply be 0 ("all stop").
 - c) The frequency response of the MD CDFD filter will then be completely determined by the frequency responses of the 1D IIR filters.
 - d) Upon completion of these steps, there will be up to $L_2 \times L_3 \times \dots \times L_M$ LDEs to be implemented.
 - e) The filtering process can now be carried out according to the procedure outlined in Section 3.2.

3.4.3. Design Examples of CDFD Filters

3.4.3.1. CDFD Approximation of a 3D Cone-Stop Filter - A Design Example

The following example illustrates how the CDFD filtering technique can be used to approximate and implement a 3D digital filter having a cone-stop magnitude frequency response, as shown in Fig. 3.5a. The filter has an input $x(\mathbf{k})$ and output $y(\mathbf{k})$ where $\mathbf{k} = \{k_1, k_2, k_3\}'$. The shaded region represents the stopband of the filter in which the value of the magnitude frequency response is ideally 0 at all discrete-frequency points $\Omega = \{\Omega_1, \Omega_2, \Omega_3\}'$. The ideal magnitude frequency response $|H(\Omega)|$ is unity outside the shaded cone region.

In this example, the magnitude frequency response $|H(\Omega)|$ in Fig. 3.5a is approximated using a 3D CDFD filter having IIR filtering in one signal dimension ω_1 and DFT filtering in the remaining two signal dimensions Ω_2, Ω_3 .

In the case of conventional discrete-frequency filtering, $|H(\Omega)|$ is defined on a 3D rectangular grid, as indicated in the $\Omega_3 = 0$ section shown in Fig. 3.5b. However, for CDFD filtering, the corresponding frequency-response is *continuous* in the ω_1 dimension due to continuous frequency filtering along k_1 . $|H(\omega_1, \Omega_2, \Omega_3)|$ is defined as the vertical lines shown for $\Omega_3 = 0$ in Fig. 3.5c. The full set of vertical lines over the Ω_2, Ω_3 plane defines the passbands of the required set of lowpass filters which, from simple geometry, must have bandwidths $B(\Omega_2, \Omega_3)$

that are proportional to the distance $\sqrt{\Omega_2^2 + \Omega_3^2}$ of the lines from the origin in the

$\omega_1 = 0$ plane. Thus,

$$B(\Omega_2, \Omega_3) = B_0 \sqrt{\Omega_2^2 + \Omega_3^2} \quad (3.35)$$

where B_0 is a constant that determines the width of the cone.

The impulse response of the ideal lowpass filter, having the frequency response shown in Fig. 3.5d, is given by

$$h_{d_{\Omega_2 \Omega_3}}(k_1) = F_{1,1}^{-3} [H(\Omega_1, \Omega_2, \Omega_3)] \quad (3.36)$$

which is the impulse response of the required 3D cone-stop filter at specific values of Ω_2 and Ω_3 .

Let the sequence lengths of $x(k_1, k_2, k_3)$ in the k_2 and k_3 directions be 64 and an unspecified length in k_1 ; that is, $L_2 = L_3 = 64$ and $\mathbf{L}_d = \text{diag}\{L_2, L_3\}$.

Using the expression for the partial MD IDFT from Equation (2.57) in Equation (3.36) gives

$$h_{d_{\Omega_2 \Omega_3}}(k_1) = \frac{1}{L_1} \sum_{\Omega_1=0}^{L_1-1} H(\Omega_1, \Omega_2, \Omega_3) \exp\left[\frac{j2\pi k_1 \Omega_1}{L_1}\right] \quad (3.37)$$

If the sequence length L_1 in Ω_1 is sufficiently large and $k_1 \ll L_1/2$, then Equation

(3.37) can be written as

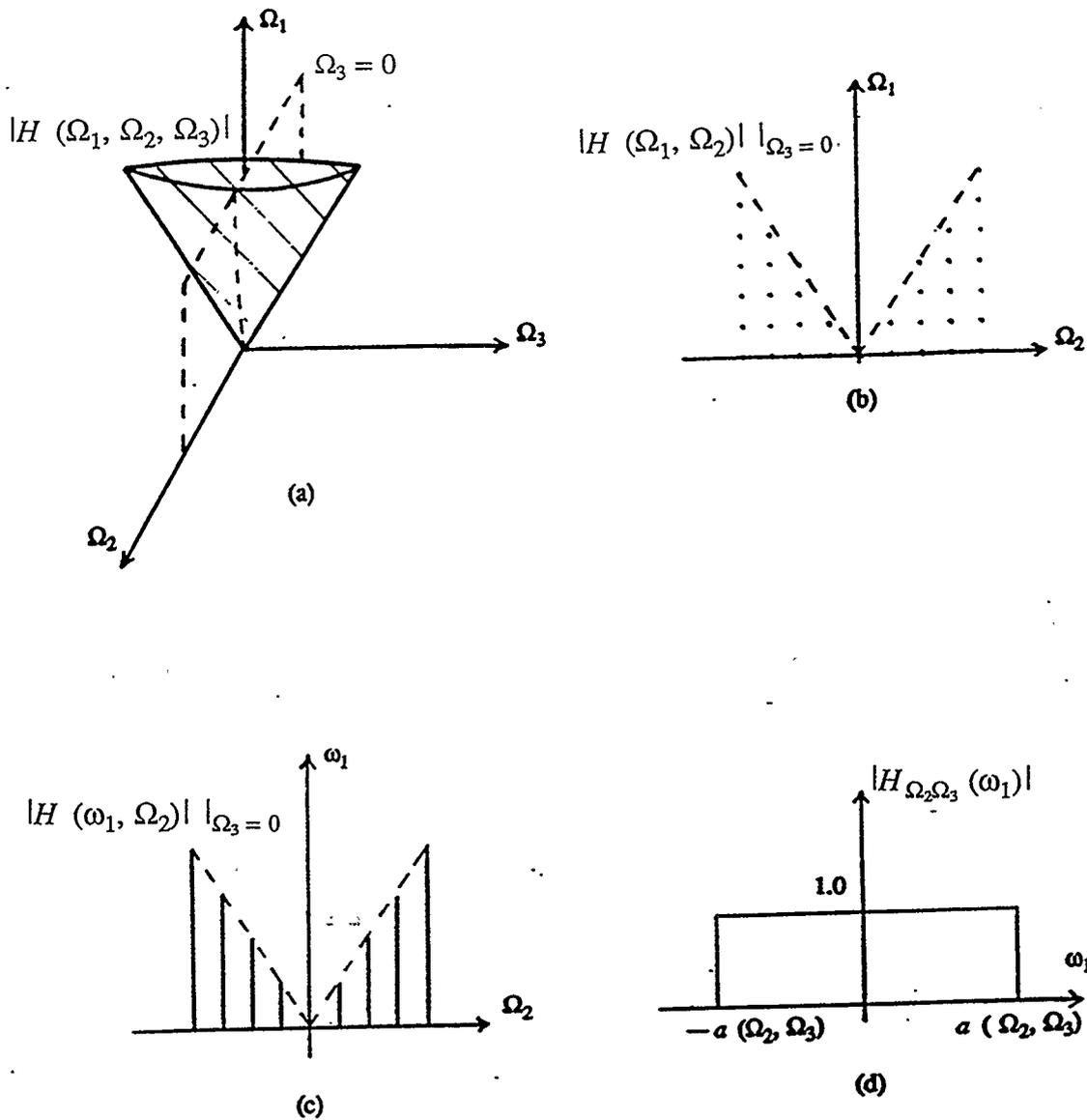


Fig. 3.5. Design of 3D CDFD cone-stop filter.

(a) Ideal magnitude frequency response of a 3D cone-stop filter.

(b) Section through $\Omega_3 = 0$ plane of frequency response in (a).

Dots represent passband gain of unity.

(c) Frequency response of CDFD filter approximation to (a) at $\Omega_3 = 0$ plane.

Vertical lines represent passband gain of approximately unity.

(d) Lowpass filter frequency response in ω_1 required to produce the frequency response in (a).

$$h_{d_{\Omega_2\Omega_3}}(k_1) \approx 2B_0 \sqrt{\Omega_2^2 + \Omega_3^2} \operatorname{sinc} \left[\frac{2\pi k_1}{L_1} B_0 \sqrt{\Omega_2^2 + \Omega_3^2} \right] \quad (3.38)$$

which is the required impulse response for the CDFD filter to approximate the ideal cone-stop filter for given values of Ω_2 and Ω_3 . The impulse response $\mathbf{h}_d(k_1)$ of the CDFD filter for all values of $(k_1, \Omega_2, \Omega_3)$ can therefore be written in matrix form

$$\mathbf{h}_d(k_1) = \begin{bmatrix} h_{0,0}(k_1) & h_{0,1}(k_1) & \cdots & h_{0,63}(k_1) \\ h_{1,0}(k_1) & h_{1,1}(k_1) & \cdots & h_{1,63}(k_1) \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ h_{63,0}(k_1) & h_{63,1}(k_1) & \cdots & h_{63,63}(k_1) \end{bmatrix} \quad (3.39)$$

where each element of the impulse response matrix $\mathbf{h}_d(k_1) \equiv \{h_{d_{\Omega_2\Omega_3}}(k_1)\}$ is given by Equation (3.38). Clearly, this design example requires $64 \times 64 (= 4096)$ different lowpass filtering operations, in addition to two 2D-DFT operations on 64×64 data points for each value of k_1 .

3.4.3.2. CDFD Approximation of a 2D Circularly Symmetric Lowpass Filter -

A Design Example

The magnitude frequency response of an ideal circularly symmetric 2D lowpass filter is shown in Fig. 3.6a. The shaded region represents the passband of the filter where the magnitude frequency response $|H(\Omega_1, \Omega_2)|$ is ideally unity,

$|H(\Omega_1, \Omega_2)|$ is assumed to be ideally 0 in the stopband of the filter (unshaded region). The filter has input $x(\mathbf{k})$ and output $y(\mathbf{k})$ where $\mathbf{k} \equiv \{k_1, k_2\}'$.

In the case of conventional discrete-frequency filtering, $H(\Omega_1, \Omega_2)$ is defined on a grid of discrete-frequency points, as shown in the $\Omega_1 = \Omega_1$ cross section in Fig. 3.6b. A set of lowpass filters, such as the one in Fig. 3.6b, will approximate the frequency response in Fig. 3.6a. From simple geometry, the bandwidth of these filters at each value of Ω_1 is given by

$$\begin{aligned} B(\Omega_1) &= \sqrt{R^2 - \Omega_1^2}, \quad -R \leq \Omega_1 \leq R \\ &= 0, \quad |\Omega_1| > R \end{aligned} \tag{3.40}$$

The discrete-frequency response in Fig. 3.6b can be *approximated* by a continuous-frequency 1D IIR lowpass filter in ω_2 having the frequency response $H_{\Omega_1}(\omega_2)$ in Fig. 3.6c.

For this example, a 3rd order 1D IIR lowpass Butterworth filter in ω_2 has been used to implement $H_{d_{\Omega_1}}(\omega_2)$. The filter has a cutoff frequency $B(\Omega_1)$, where $B(\Omega_1)$ is given by Equation (3.40). Thus, the frequency response $H_{d_{\Omega_1}}(\omega_2)$ of each filter (one for each value of Ω_1) is given by

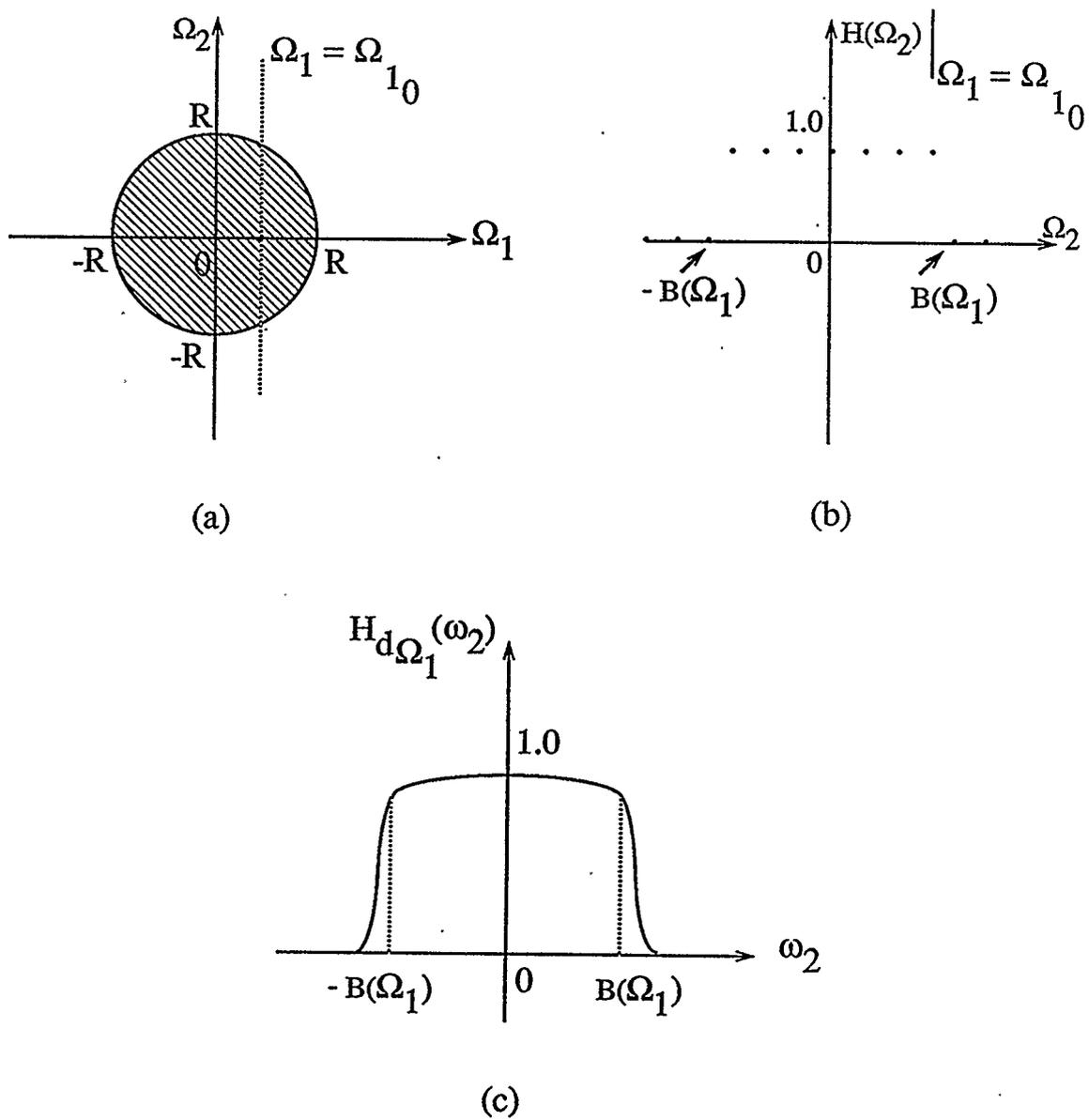


Fig. 3.6. Design of 2D CDFD lowpass filter.

(a) Ideal frequency response of a 2D lowpass filter.

(b) Section through $\Omega_1 = \Omega_{10}$ line of the frequency response in (a).

(c) Lowpass filter frequency response in ω_2 required to approximate the frequency response in (b).

$$H_{d_{\Omega_1}}(\omega_2) = \frac{1}{\sqrt{1 + \left(\frac{\omega_2}{B(\Omega_1)}\right)^6}} \quad (3.41)$$

It is important to note that the 1D IIR filter used for $\Omega_1 = \Omega_{1_0}$ is the same as the filter required for $\Omega_1 = -\Omega_{1_0}$. Therefore for $-R \leq \Omega_1 \leq R$, the number of filters required is half the total number of discrete-frequency points Ω_1 in this range. This represents a substantial saving in hardware. Furthermore, from (3.40), $B(\Omega_1) = 0$ when $|\Omega_1| > R$, implying that an all stop filtering operation is required for this range of values of Ω_1 . In other words, *no 1D IIR filtering operations are required for $|\Omega_1| > R$* . This results in a further significant decrease in computation and hardware because after obtaining the L_2 L_1 -point row-DFTs (sequence lengths L_1 and L_2 are assumed for Ω_1 and Ω_2 , respectively), only $2R$ 1D IIR lowpass filtering operations are required. Finally, by taking the symmetry of the required frequency response into account, a total of *only R distinct 1D IIR filters are required*.

The overall input/output frequency response $\mathbf{H}_d(\omega_2)$ of the 2D CDFD lowpass filter can be written as a 1D array as follows :

$$\mathbf{H}_d(\omega_2) \equiv \begin{bmatrix} H_{d_0}(\omega_2) \\ H_{d_1}(\omega_2) \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ H_{d_{L_2-1}}(\omega_2) \end{bmatrix} \equiv \left\{ H_{d_{\Omega_1}}(\omega_2) \right\} \quad (3.42)$$

where each $H_{d_{\Omega_1}}(\omega_2)$ is calculated from Equation (3.41) for a given value of Ω_1 .

The reduction of "edge-effects" in at least one dimension in CDFD filtering is illustrated in Fig. 3.7. The input signal is shown in Fig. 3.7a. The input signal has a sharp discontinuity at the rear edge in the k_2 direction, swinging from a magnitude of 1 to zero. This edge discontinuity will give rise to "edge-effects" in the k_2 direction if 2D DFT filtering is applied to the input image. The transitions of magnitude from 1 to 0 in the input image in the k_1 direction occur away from the edges of the image. This reduces the "edge-effects" in the k_1 direction and allows the edge-effects in the k_2 direction to be observed more easily.

The output of a 2D DFT lowpass filter, having the frequency response in Fig. 3.6a, is shown in Fig. 3.7b. The 2D DFT filter has a 3rd order lowpass Butterworth cross-section in the Ω_2 direction with the cutoff frequencies varying as

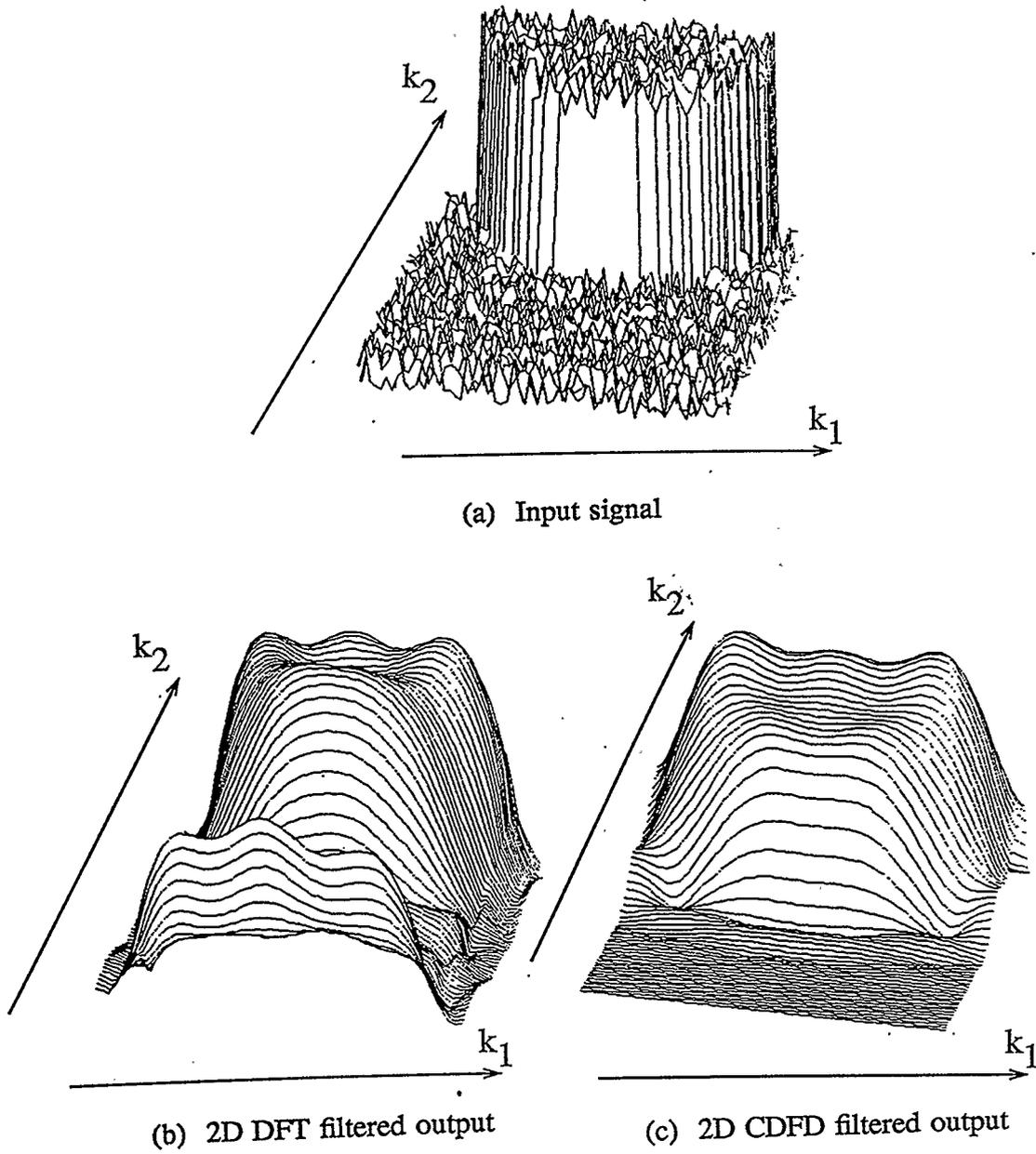


Fig. 3.7. Reduction of edge effects using CDFD filtering.

a function of Ω_1 , as given by Equation (3.40). Fig. 3.7b clearly shows the edge-effects due to DFT filtering in the k_2 direction. A spurious edge is present at the front of the signal and "ringing" artifacts can be observed. In contrast, the output of the 2D CDFD filter described in this section shows no edge-effects in the k_2 direction because IIR filtering was applied in this direction. This demonstrates one of the advantages inherent in CDFD filtering, namely, that edge-effects are eliminated in the signal dimensions in which continuous-frequency filtering is applied. However, ringing are present in the direction k_1 in which DFT filtering is applied.

3.4.3.3. An Application of CDFD Filters in Seismic Image Processing -

A Design Example

A 2D fanstop filter, having the ideal magnitude frequency response $|H(\Omega_1, \Omega_2)|$ shown in Fig. 3.8(a) can be used to eliminate low-velocity ground roll interference from 2D seismic images [2]. This fanstop filter has a total angular stopband width of $\theta_s = 10^\circ$ and may be implemented using the CDFD filtering technique. A cross-section of the frequency response in Fig. 3.8(a), along the $\Omega_2 = \Omega_{20}$ line is shown in Fig. 3.8(b). The frequency response $|H(\Omega_1)|_{\Omega_2 = \Omega_{20}}$ is clearly a discrete-frequency lowpass filter frequency response in Ω_1 . This has been approximated using a 3rd order 1D IIR lowpass Butterworth filter as shown in Fig. 3.8c. From simple geometry, the cutoff frequency $B(\Omega_2)$

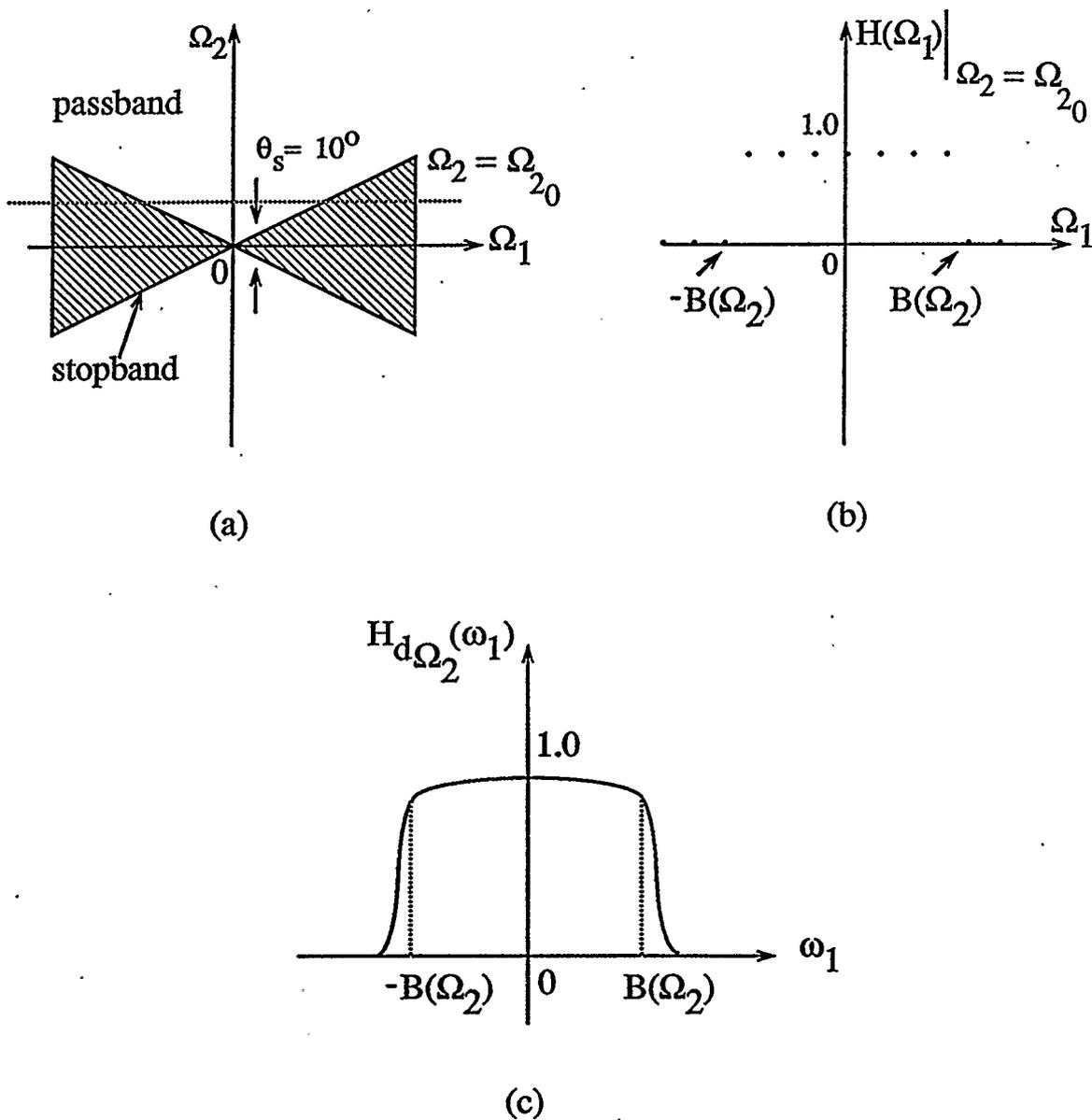


Fig. 3.8. Design of fanstop filter required for seismic image processing.
 (a) Magnitude frequency response $|H(\Omega_1, \Omega_2)|$ of a fanstop filter required for ground-roll removal from seismic images.
 (b) Section through $\Omega_2 = \Omega_{20}$ plane of frequency response in (a).
 (c) Frequency response of 1D IIR filter used to approximate (b).

for the 1D IIR lowpass filter at a given value of Ω_2 is given by

$$B(\Omega_2) = \frac{\Omega_2}{\tan\left[\frac{\theta_s}{2}\right]} \quad (3.43)$$

Therefore, the frequency response $H_{d_{\Omega_2}}(\omega_1)$ of each 1D IIR filter is given by

$$H_{d_{\Omega_2}}(\omega_1) = \frac{1}{\sqrt{1 + \left(\frac{\omega_1}{\Omega_2} \tan\left(\frac{\theta_s}{2}\right)\right)^6}} \quad (3.44)$$

The frequency response of the 2D CDFD fanstop filter can be written as a 1D array (or vector) of functions of ω_1 as follows :

$$\mathbf{H}_d(\omega_1) \equiv \{H_{d_{\Omega_2}}(\omega_1)\} = \begin{bmatrix} H_{d_0}(\omega_1) \\ H_{d_1}(\omega_1) \\ \cdot \\ \cdot \\ \cdot \\ H_{d_{63}}(\omega_1) \end{bmatrix} \quad (3.45)$$

where the sequence length L_2 in Ω_2 is 64 and each element $H_{d_{\Omega_2}}(\omega_1)$ is given

by Equation (3.44).

The input to the CDFD filter is a 1024×64 (k_1, k_2) 2D seismic image consisting of 64 traces ($L_2 = 64$) having 1024 temporal samples per trace

($L_1 = 1024$). The first step in the CDFD filtering process is to perform 1024 64-point FFTs on the input image according to Equation (2.56). The required fanstop filtering operation is then approximated using 64 1D IIR filter operators, one for each value of Ω_2 . Due to the symmetry of the fanstop filter, only 32 different filters are required, since the filter for $\Omega_2 = \Omega_{2_0}$ is the same as the filter for $\Omega_2 = -\Omega_{2_0}$. Each filter operates on the real and imaginary components of a 1024-point trace (row) of the partial discrete Fourier transformed input image according to Equation (3.4). Third-order lowpass Butterworth filters whose passband widths vary linearly with trace number Ω_2 (as given by Equation (3.43)), have been employed. The recursive (IIR) filtering has been performed in the ω_1 dimension because the larger sequence length ($L_1 = 1024$) allows the startup transients of the 1D IIR filters to decay to zero, resulting in less distortion in the output image. In addition, applying 1024 64-point FFTs to the input image requires 60 percent of the complex multiplications required for 64 1024-point FFTs. Finally, 1024 64-point inverse FFTs are performed on the complex outputs of the IIR filtering operations to produce the complex output of the combined CDFD filter, according to Equation (3.7).

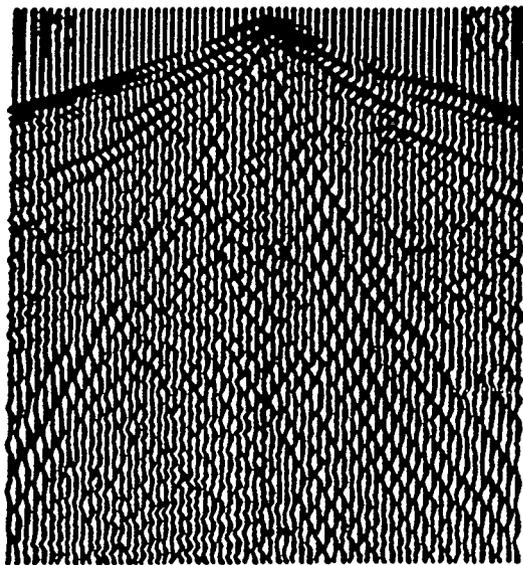
For this application, the magnitude of the imaginary component of $y(\mathbf{k})$ of the CDFD filter is found to be less than 3 percent of the magnitude of the real component. Therefore, the imaginary component of the output of the CDFD filter is neglected for this application.

The input image $x(\mathbf{k})$ and the real and imaginary components of the output $y(\mathbf{k})$ of the CDFD filter are shown in Figs. 3.9a, b and c, respectively. The ground roll, which corresponds to the angular 'ripples' in Fig. 3.9a, has clearly been reduced by the CDFD filter, as required. A comparison of Figs. 3.9b and 3.9c confirms that the magnitude of the imaginary component of the CDFD filter output is negligible compared to the magnitude of the real component of the output.

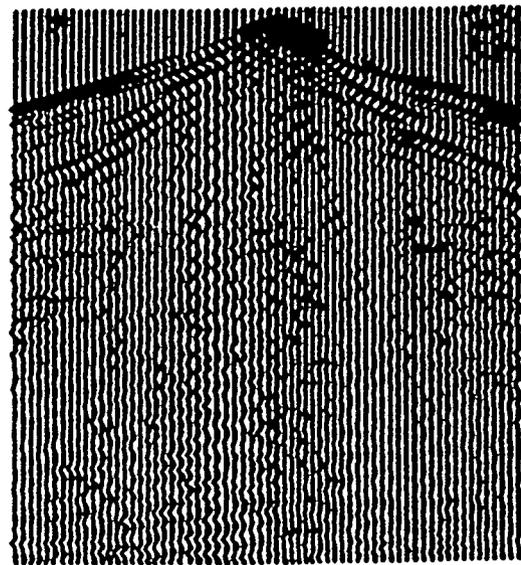
The corresponding output of a 2×5 IIR 10 degree 2D fanstop filter, designed by Bruton, Bartley and Stein [2], is shown for comparative purposes in Fig. 3.9d. A comparison of Figs. 3.9b and 3.9d shows that the input/output performance of the CDFD filter is similar to that of the 2×5 IIR filter.

3.5. Comparison of CDFD Filters with IIR and DFT Filters in Seismic Signal Processing Applications

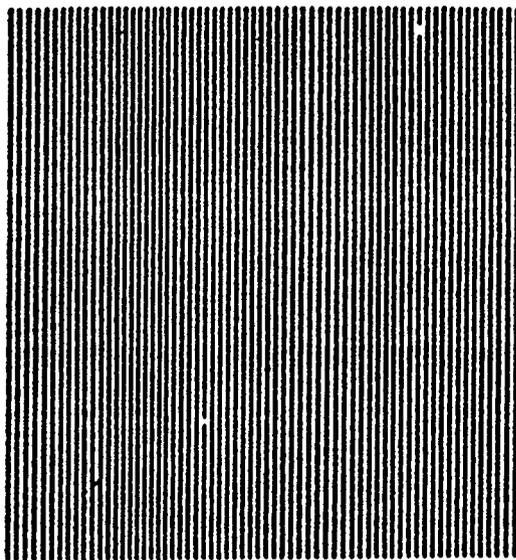
Some simple calculations have been performed to compare the complexity of the CDFD fanstop filter with its IIR [2] and 2D DFT counterparts, for the application described in Section 3.4.3.3. The criteria chosen for this comparison are the approximate number of multiplications N_{MULT} , the approximate number of additions N_{ADD} , and the approximate amount of memory storage required for each of the three filtering methods for a 64×1024 seismic image. The results of this comparison [Appendix A], for the seismic filtering application, are shown in Table 3.1.



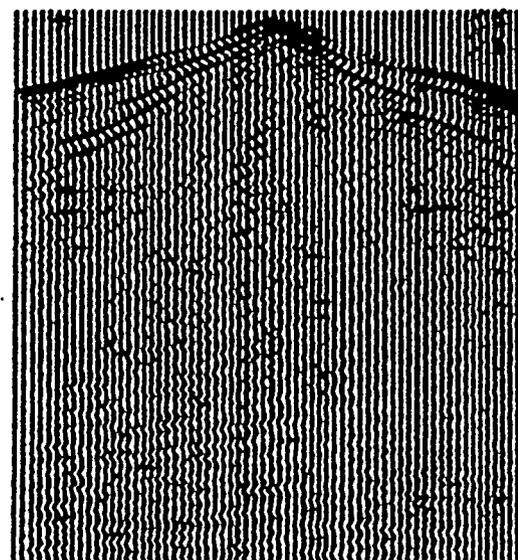
(a) Input seismic image



(b) CDFD-filtered output (real)



(c) CDFD-filtered output (imaginary)



(d) IIR-filtered output

Fig. 3.9. Low velocity ground roll removal from seismic images using a 2D CDFD fanstop filter. The output of a 2D IIR filter is shown in (d) for comparison.

Types of Filters	N_{MULT}	N_{ADD}	Bytes of Memory (8 bit data)
CDFD	4.06×10^6	2.36×10^6	1.62×10^5
IIR	4.59×10^6	4.46×10^6	6.55×10^4
2D-DFT	8.52×10^6	4.19×10^6	1.97×10^5

Table 3.1. Comparison of computational efficiency and memory usage for CDFD, IIR and 2D DFT filtering of seismic images.

The CDFD filter requires fewer additions and multiplications than the IIR and 2D DFT methods. As expected, the 2D DFT method is computationally least efficient and is the most memory intensive. The IIR method requires the smallest amount of memory. However, the CDFD method requires fewer computations than the IIR method.

The approximate number of additions and multiplications required for each of the three methods, for varying image sizes, has been calculated [Appendix A]. Calculations have been performed for a constant number of spatial seismic traces (64) with the number of temporal samples per trace L_1 varying from 2 to 512. The graphs in Figs. 3.10 and 3.11 show the number of additions and multiplications, respectively, for all three methods. From these data it is clear that the CDFD method requires fewer additions and multiplications than the other two methods, for the range of image sizes considered here.

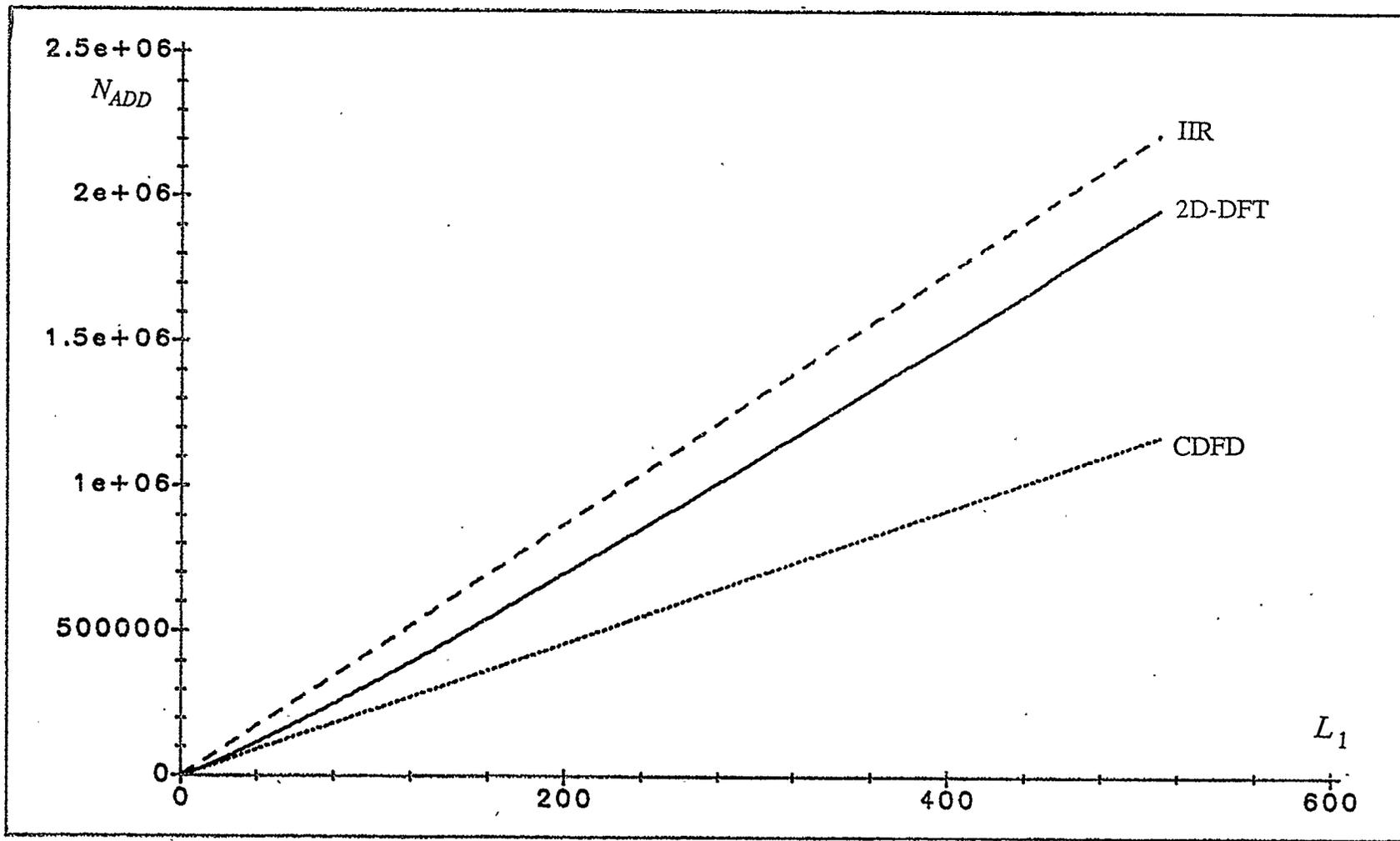


Fig. 3.10. Number of additions necessary for removing ground roll from a $64 \times L_1$ seismic image using 2D-DFT, IIR and CDFD fan stop filters. L_1 represents the number of temporal samples per seismic trace.

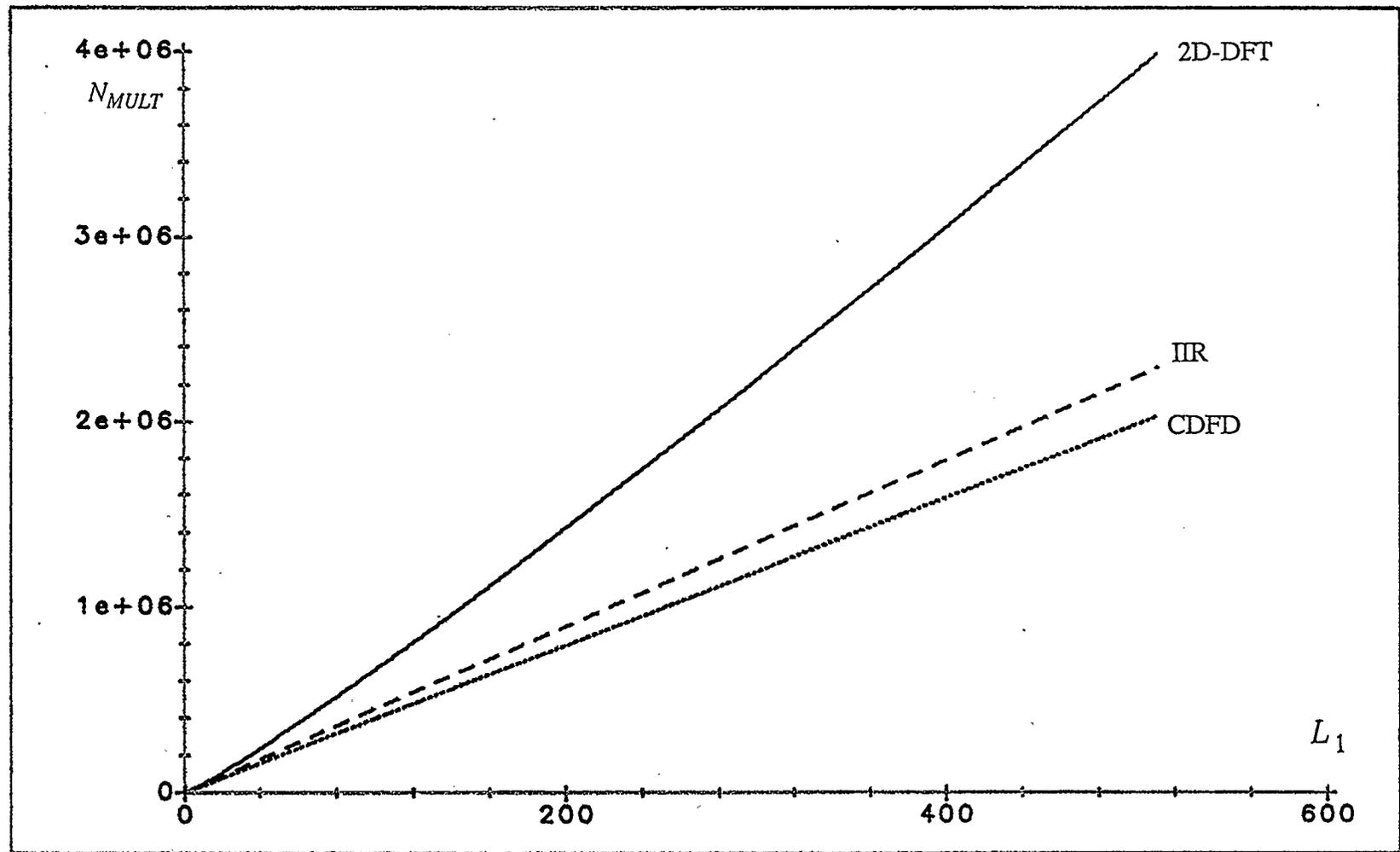


Fig. 3.11. Number of multiplications necessary for removing ground roll from a $64 \times L_1$ seismic image using 2D-DFT, IIR and CDFD fan stop filters. L_1 represents the number of temporal samples per seismic trace.

Another set of calculations has been performed, for a constant number of temporal samples (64) per seismic trace, with the number of traces L_2 varying from 2 to 512. The number of additions and multiplications, respectively, for all three methods is shown in Figs. 3.12 and 3.13. It is evident from Fig. 3.13 that if the image size exceeds 128 traces, the IIR method requires fewer multiplications than the CDFD method. For an image containing fewer than 128 spatial seismic traces, 64 in this case, the CDFD method requires fewer multiplications than the IIR method. These results agree with the results in Table 3.1 and demonstrate that for the application described in this section, CDFD filtering requires fewer computations than both IIR and 2D DFT filtering.

CDFD filtering has two other significant advantages. Firstly, edge effects are eliminated in the dimensions in which continuous-frequency filtering is applied, unlike MD DFT filters. Secondly, MD filter passbands of arbitrary shape can be designed easily unlike MD IIR filters. This is demonstrated for the 2D case in Chapter 4 which describes a software image processing system based on CDFD filters. Some important considerations in implementing this system in hardware are also discussed.

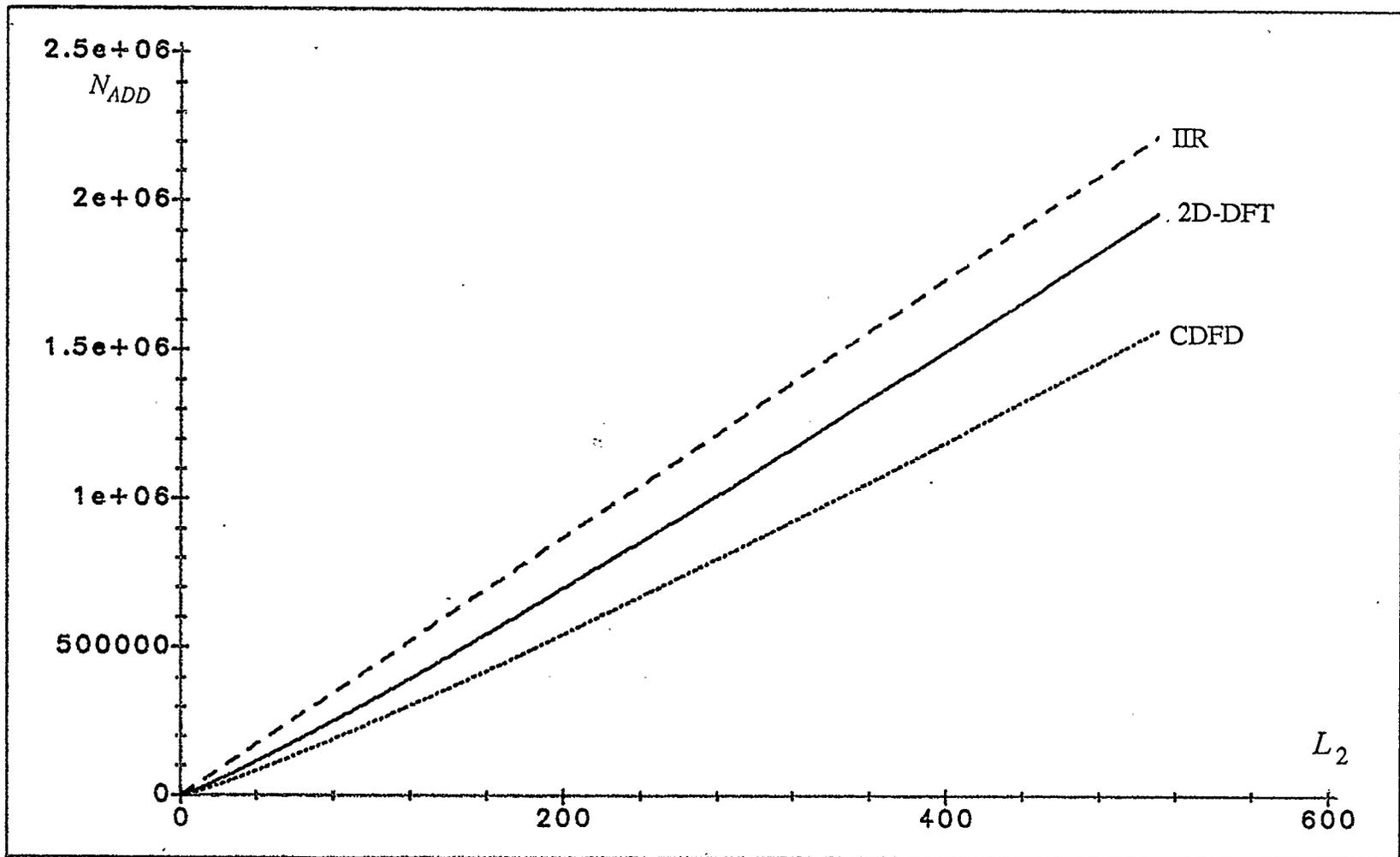


Fig. 3.12. Number of additions necessary for removing ground roll from a $64 \times L_2$ seismic image using 2D DFT, IIR and CDFD fan stop filters. L_2 represents the number of seismic traces.

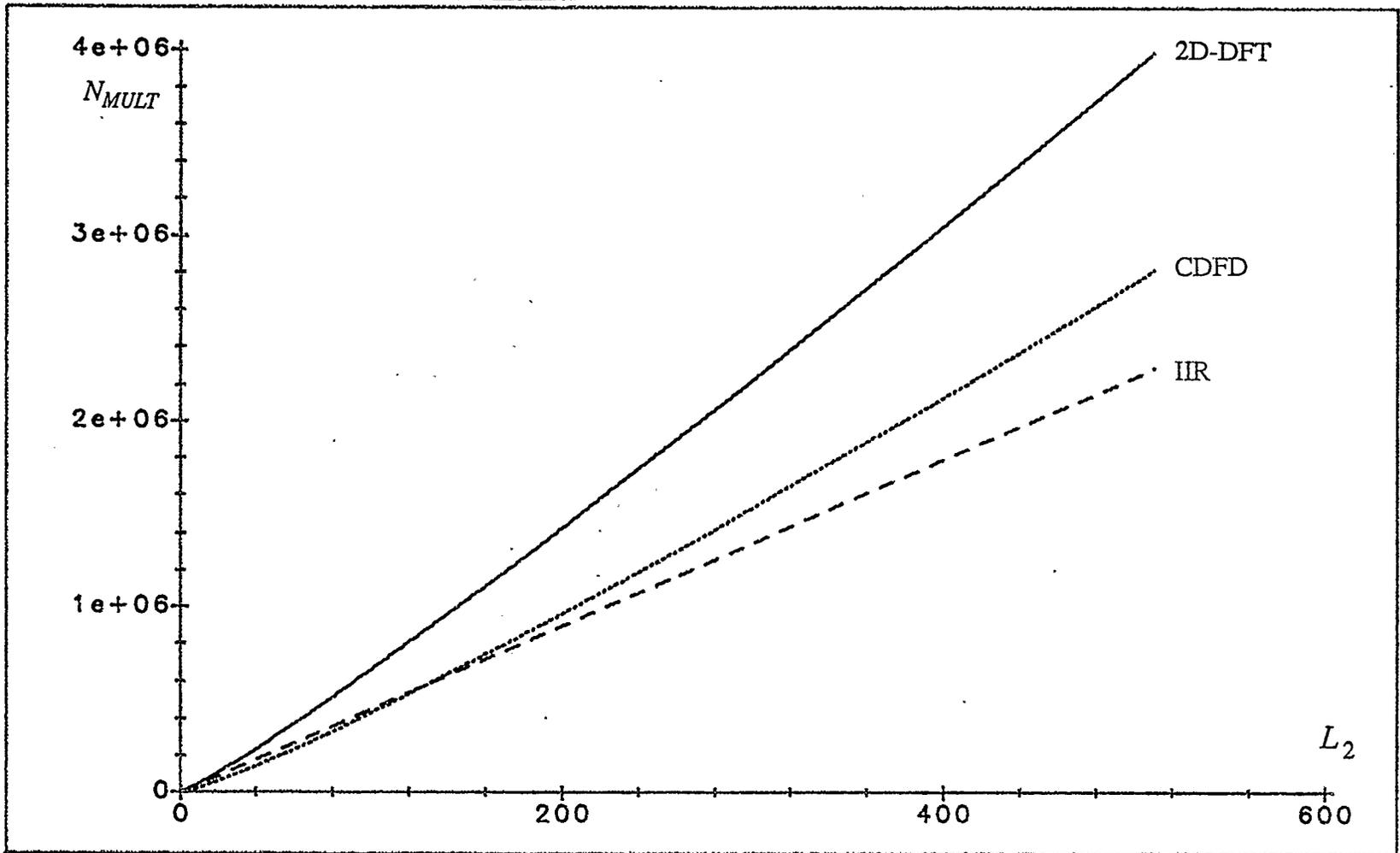


Fig. 3.13. Number of multiplications necessary for removing ground roll from a $64 \times L_2$ seismic image using 2D DFT, IIR and CDFD fan stop filters. L_2 represents the number of seismic traces.

CHAPTER 4

AN IMAGE PROCESSOR USING CDFD FILTERS

4.1. Introduction

In many image processing applications, such as biomedical and seismic image processing, it is desirable to have an image processing system that allows the user to interactively filter images. For example, in order to enhance an X-ray image, a radiologist may require selective filtering of certain features of the image. An interactive image processing system, designed for this type of application, is described in this chapter. The system, based on a software implementation of 2D CDFD filters, is described in Section 4.2. Some hardware schemes, which may be used to implement this system, are described in Section 4.3.

4.2. Software Image Processing System

The interactive software image processing system described in this section runs on a SUN-3 computer. The program used to implement the image processing system is written in 'C'. An FFT program and some display routines written by N.R. Bartley are included in the package. Apart from some sections of the FFT program that are written in assembly language, all other programs are in 'C'. The image processing program requires input from the terminal and a mouse (used for drawing filter passbands on the display terminal).

The image processing program has been written primarily to demonstrate the advantages of CDFD filters in fast, interactive image processing. The image processing system will first be described from the user's perspective. The images processed by the system are 128×128 pixels in size and are stored in binary files. A sample run of the program on a file called "testfile" will be used to demonstrate the user interface.

The image in 'testfile' is shown in Fig. 4.1a. It consists of the superposition of a very low frequency 2D sinusoidal waveform parallel to the edges of the image and a high frequency 2D sinusoidal waveform at 45° to the edges of the image. It is required to remove this high frequency waveform from the image. The display in Fig. 4.1a, that is the operation of viewing an image file, is obtained by the keyboard command "dspl testfile". Since the 2D CDFD filter required to remove the high frequency component from the test image has to be designed in the frequency domain, a 2D FFT operation is applied to the test image. This is accomplished by the command "fft testfile". The output of the 2D FFT operation is written into a file called "fft_out". This file is now displayed using the "dspl" command and is shown in Fig. 4.1b. By inspection of the Fourier transformed image, the user must determine which areas of the 2D frequency space constitute the required signal. This requires a background in 2D frequency domain filtering and the user must be properly trained to make use of the system.

The required passband (or stopband) of the 2D CDFD filter is drawn on the Fourier transformed image. This involves using the mouse to outline the region

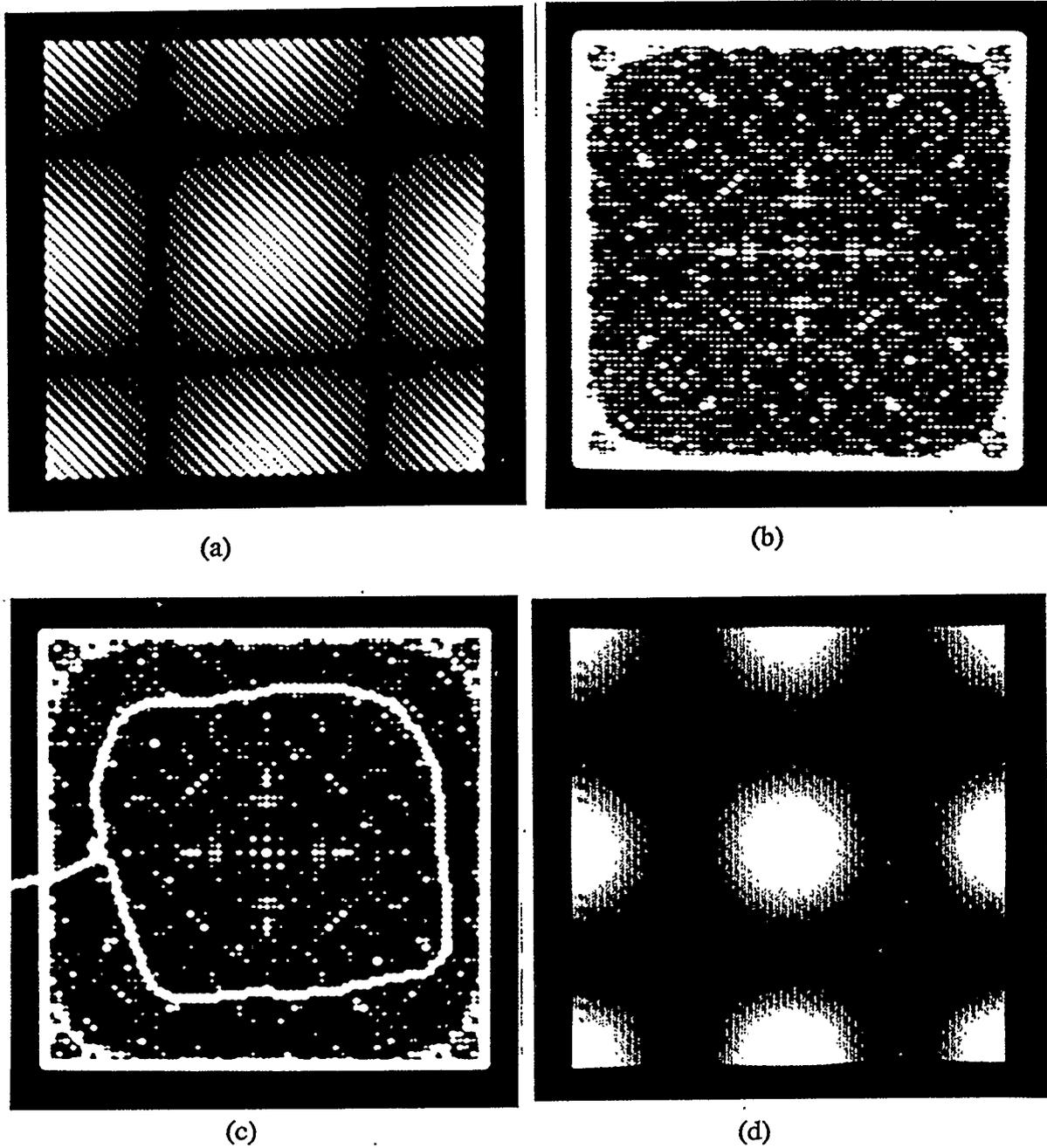


Fig. 4.1. Output of software interactive image processing system.

- a) input test image
- b) 2D FFT of test image
- c) stopband of required CDFD filter (encircled)
- d) 2D CDFD filter output

containing the required signal frequencies. Alternatively, the region containing undesirable frequencies may be outlined as the stopband if convenient. It should

be noted that the frequency domain signal is displayed from $(0,0)$ to $(2\pi, 2\pi)$ in normalized 2D frequency space instead of $(-\pi, -\pi)$ to (π, π) . The higher frequencies lie at the center of the Fourier transformed image instead of at the edges and this must be taken into account when drawing the CDFD filter passband.

The actual procedure of drawing the passband or stopband shape using the mouse is straightforward. The drawing procedure is invoked by the command "mouse". This configures the right button of the mouse as "draw", the center button as "redraw" and the left button as "quit". Pressing and holding down the draw button causes a dotted line, starting at the bottom left corner $(0,0)$, to be drawn on the Fourier transformed image. This line is used to encircle the required region for the passband (or stopband) of the CDFD filter. *A closed contour is required for the passband (or stopband).* This is shown in Fig. 4.1c. If an error is made while drawing, the "redraw" (middle) button on the mouse will clear the dotted line and a new contour can be drawn using the "draw" button. Once a satisfactory contour has been drawn, the "quit" (left) button is used to exit the drawing procedure.

If the top or bottom edge of the Fourier transformed image is part of the contour, then a lowpass or highpass 2D CDFD filter is required. Otherwise a 2D bandpass or bandstop filtering operation is necessary. The user must choose any one of four types of filtering operation by using one of the commands: "lowpass", "highpass", "bandpass" and "bandstop". Upon invoking one of these commands, the input image is filtered using the required type of 1D filters in the CDFD

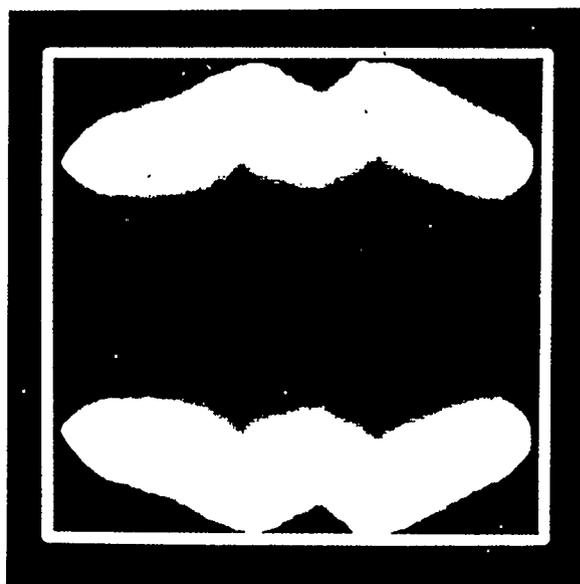
method. This normally takes between 8 to 18 CPU seconds on a SUN-3 computer for image sizes of 128×128 pixels. The filtered image is written into a file called "output" which can be displayed using the "dspl" command. The output file usually contains only the real part of the output image. However, the program can easily be modified to display the imaginary part of the output image. It is evident from the filtered image in Fig. 4.1d that the high frequency sinusoidal waveform has been almost removed as required.

The lowpass, highpass, bandpass and bandstop 1D IIR filters used in the system are all derived from 3rd order Butterworth filters. The selectivity obtained using these filters was found to be satisfactory for the test images used. Higher order filters may be used if required. During the CDFD filtering process, a set of 128, 128-point row FFTs is applied to the input image. A 1D IIR filtering operation is then performed along every column of the partially Fourier transformed image (if necessary). For columns which lie in the 2D CDFD filter stopband, no filtering is performed and a column of zeros is written into the real and imaginary parts of the IIR filter output. If an entire column lies in the 2D CDFD filter passband, once again no IIR filtering is performed and the real and imaginary parts of the column are simply copied into the corresponding columns of the IIR filter output. This results in substantial savings in computation and processing time. The required bandwidths and cutoff frequencies for the IIR filters are generated by the "mouse" program based on the filter passband drawn on the screen.

It is also possible to view the magnitude frequency response of a CDFD filter designed using this system. This is achieved by using a unit impulse (a Kronecker delta function) at the origin as the input image. The filter passband is drawn as before on the Fourier transformed input image. The Fourier transform (2D FFT) of the output is obtained using the "fft" command. The output of this 2D FFT is the frequency response of the 2D CDFD filter. Two such frequency responses are shown in Fig. 4.2, demonstrating the flexibility of CDFD filtering in designing filter passbands of arbitrary shape.

The software image processing system allows the user to easily design 2D CDFD image processing filters of arbitrary passband shape. However, some knowledge of 2D filtering is required before the user can efficiently use the system. The system can be made more user friendly so that all technical decisions (e.g. the type of filtering required) can be presented to the user in layman's terms. An alternative approach is to incorporate the image processing system in an expert system so that user decisions are reduced to a minimum. A threshold level can be set to determine the desirable portion of the image and the system can then design the necessary filters. A more sophisticated approach might involve pattern recognition in the Fourier transformed image to determine desirable and undesirable signal frequencies and patterns.

This type of system can be used in medical image processing because, with proper training, the user can selectively enhance features in the image. The next section outlines some ideas for implementing this system in hardware.



(a) "W" shaped passband



(b) Fan shaped passband

Fig. 4.2. Arbitrary passband shapes of 2D CDFD filters obtained using the image processing system.

4.3. Proposed Scheme for Hardware Implementation of the Image Processing System

The image processing system described in Section 4.2 can be implemented in hardware to increase the processing speed, possibly to real-time. A brief description of the hardware that can be used for this purpose is given in this section based on the TMS320C30 signal processor, one of the fastest digital signal processors available [5]. The current cost of this chip is approximately \$300 and therefore, in order to keep the cost of the proposed system to a minimum, it is assumed that a single TMS320C30 is used. It is possible to use a number of TMS320C30s in parallel to increase processing speeds but this can be very expensive.

The hardware scheme outlined here utilizes some of the special features inherent to CDFD filters and therefore can be applied to general MD CDFD filters. It is assumed that the 1D IIR filters required for the signal processor will be programmed in direct form in the TMS320C30 software. A control program resides in the TMS320C30 ROM and performs some special functions aside from the usual control functions such as input/output (I/O), timing etc. The input data (size $L_1 \times L_2$) is read in row by row. The next section describes some of the special functions of the control program.

4.3.1. Control Program

Aside from the usual control functions such as timing operations, I/O and control of arithmetic operations that are required in conventional digital signal processors, the control program has certain functions that are unique to CDFD filtering. The control program checks bandwidths and cutoff frequencies of the filters required for successive columns of the partially DFTed images to see if different 1D IIR filters are required for successive columns. If not, then the coefficients used for the previous filter are simply copied and used for the next filter.

The control program also checks for special cases, such as columns where "all go" or "all stop" functions are required. In the former case, the input signal to the bank of filters is simply copied into the appropriate memory locations for the output of the filters. For an "all stop" operation, that is, when the entire column lies in the stopband of the 2D CDFD filter, the output memory locations for the column are cleared to zero.

The control program also controls the input and output of data row by row. A row FFT is performed on each row and each of the L_1 samples is channeled into the input of the appropriate 1D IIR filter. Two filtering operations must be performed for each input point because the input to the filters is usually complex. The previous m rows (m being the order of the IIR filters) are stored in memory and updated by the program after calculation of each row of output. The control program also initiates the calculation of appropriate filter coefficients based on

bandwidths and cutoff frequencies obtained from the user-specified frequency response. Alternatively, a "lookup table" approach can be used to obtain the coefficients and this is discussed in Section 4.3.2 which describes some features of the memory storage used.

4.3.2. Memory

A schematic memory diagram is shown in Fig. 4.3. The top of the diagram shows the ROM for storing the coefficients required for the FFT operations in the form of a lookup table. This technique speeds up the FFT operation because the coefficients need not be calculated every time the FFT is applied.

In order to minimize computation, a lookup table approach can be used to determine filter coefficients for the 1D IIR filters. This implies that a finite number of bandwidths and cutoff frequencies can be achieved for any type of 1D IIR filter (i.e. lowpass, highpass, bandpass and bandstop). For example, if the sequence length for each column is L_2 (filtering performed along k_2), then the following cutoff frequencies for lowpass filters are allowed: $0 < \Omega_{co} < L_2/2$, Ω_{co} being an integer. This constraint is not very restrictive because specifications for the 1D IIR filters provided by the user are only accurate to the smallest increment of k_2 which is 1. A schematic lookup table for bandpass filter coefficients is shown in Fig. 4.4. Each set of upper and lower cutoff frequencies (Ω_{co_U} and Ω_{co_L}) produces an address $ADDR_{\Omega_{co_L}, \Omega_{co_U}}$ of the memory location containing the coefficients for

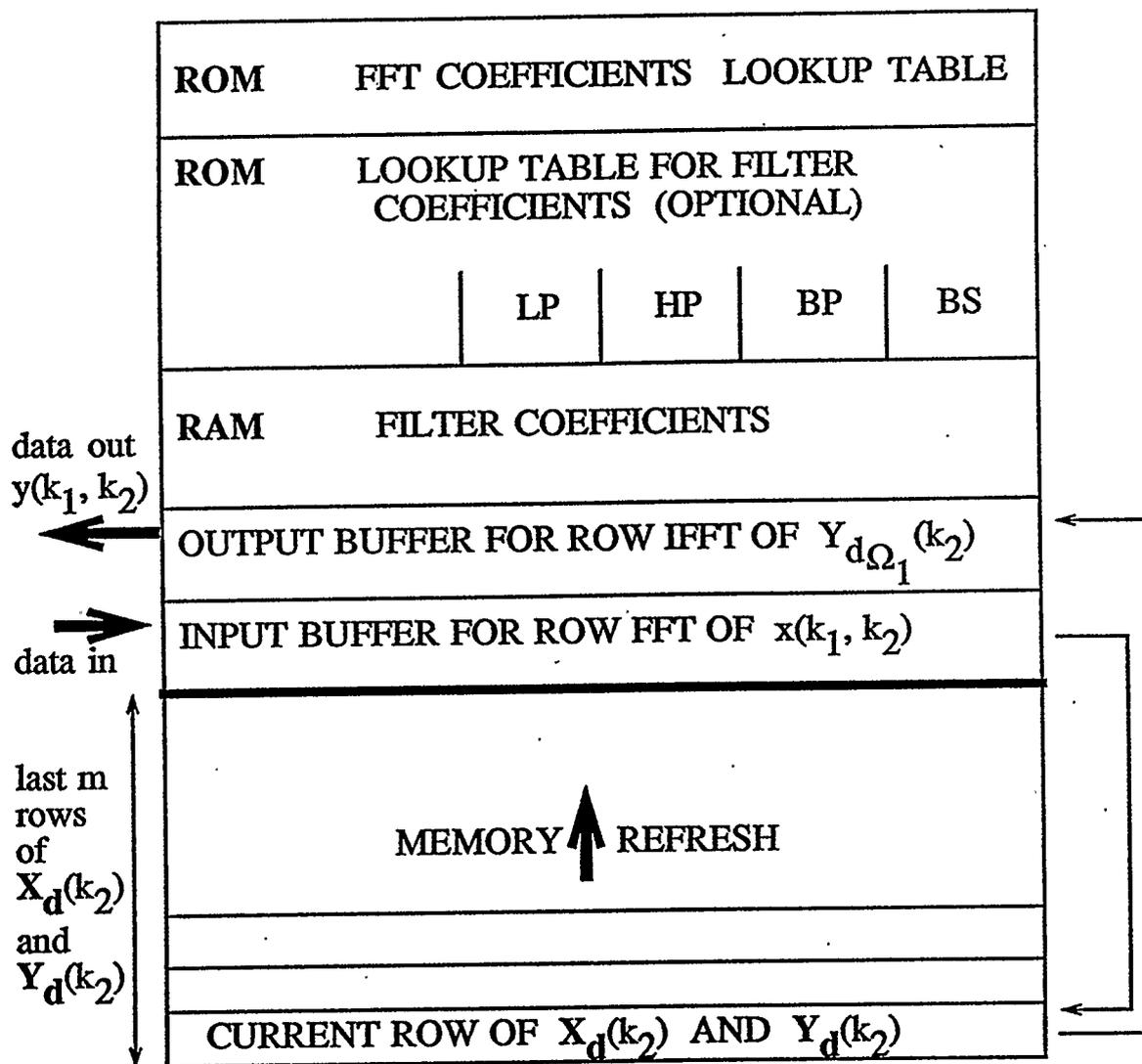


Fig. 4.3. Memory diagram.

the 1D IIR bandpass filter. Lookup tables can be implemented for all four types of filters as shown in Fig. 4.3 (LP, HP, BP, BS). The lookup tables are situated in ROM. Different filter orders may be implemented by changing the coefficient values which are stored in RAM.

$\Omega_{coL} \backslash \Omega_{coU}$	1	2	...	L_2-1
1	ADDR _{1,1}	ADDR _{1,2}
2	ADDR _{2,1}	ADDR _{2,2}
...
L_2-1

Fig. 4.4. Lookup table for coefficients of the required 1D IIR bandpass filters. Each combination Ω_{co} , BW produces the address (ADDR) in memory where the required coefficients are stored. Ω_{coL} is the lower cutoff frequency and Ω_{coU} is the upper cutoff frequency.

The RAM also contains input and output buffer space for the FFT and inverse FFT operation. If the size of the image is $L_1 \times L_2$, (L_1 columns and L_2 rows) then an L_1 -point FFT is required on each row. Since the output of the FFT is complex, $2L_1$ data points must be stored. The input row $x(k_1, k_2)$ is read into the

buffer and a 1D FFT is applied. The output of the FFT $X(\Omega_1, k_2)$ is written into the buffer, overwriting $x(k_1, k_2)$. $X(\Omega_1, k_2)$ can be written in its alternate array representation as $\mathbf{X}_d(k_2) \equiv \{X_{d_{\Omega_1}}(k_2)\}$. Each complex sequence $X_{d_{\Omega_1}}(k_2)$ is used as the input to the 1D IIR filter for the appropriate value of Ω_1 . The last m values, where m is the order of the 1D IIR filters, of $X_{d_{\Omega_1}}(k_2)$ and $Y_{d_{\Omega_1}}(k_2)$ are stored in memory (RAM) and refreshed as each new value of these sequences becomes available. These values are accessed directly by the 1D IIR filters. The filter output for an entire row is written into the output buffer and the inverse FFT is applied to this complex signal. The contents of the buffer make up one row of the CDFD filter output. The advantage of the lookup table approach to obtaining the filter coefficients lies in the fact that the pipelining available in the TMS320C30 can be used to advantage. The TMS320C30 has a 4-level deep pipeline [5] so that as one arithmetic operation is being carried out using one coefficient the next three coefficients are already in the pipelines, instead of having to be calculated. This results in improved speed. The next section briefly discusses the speeds obtainable using the scheme outlined in this section.

4.3.3. Processing Speed

Some estimates of the processing speed of the hardware signal processor, described in the previous sections, are given in this section. These estimates do not take into account the fact that the TMS320C30 must also be programmed to

handle the I/O and user interface. They are simply an estimate of the maximum speed at which the 2D CDFD filtering can be carried out on a single TMS320C30 signal processor.

Due to the extensive pipelining available in the TMS320C30 [5], a "multiply add and store" instruction is executed in 100 ns. Calculations have been performed for an input image of size 1024×1024 pixels. Each row of 1024 pixels will require $4L_1 \log_2 L_1$ ($L_1 = 1024$) "multiply add and store" operations for a 1024-point row FFT. This first step in CDFD filtering requires a processing time of approximately 4.10 ms (per row). The second stage in CDFD filtering consists of applying $L_1(1024)$ 1D IIR filtering operations for both the real and imaginary parts of the row DFT performed in the first step. Assuming approximately 10 "multiply add and store" operations for each output point for the 1D IIR filtering operation, approximately $10 L_1 \times 2$ (because real and imaginary points are filtered separately) "multiply add and store" operations are required. Assuming an execution time of 100 ns, this requires approximately 2.05 ms. Finally, the inverse row DFT on the output from the filters requires another 4.1 ms. The total processing time for a 1024-point row is therefore approximately 10.25 ms. For real-time video applications based on a frame update rate of 30 Hz, the maximum allowable time for processing a row is 0.032 ms. Clearly, the processing speed obtained does not meet this criterion.

Some techniques for improving the processing speed are proposed in the next chapter. In addition, some areas for further research are presented.

CHAPTER 5

AREAS FOR FURTHER RESEARCH

5.1. Introduction

The hardware signal processor, based on the TMS320C30 signal processor, described in Chapter 4 does not achieve real-time video filtering speeds. However, it is important to note that this signal processor filters an input image row by row. This filtering scheme can be readily adapted to real-time filtering. Factors that should be taken into account in order to achieve real-time speeds are discussed briefly in the next section. The extension of real-time CDFD filtering to 3D is also described. A CDFD filtering technique using the Discrete Hartley Transform (DHT) instead of the DFT is described in section 5.3. Finally, some concluding remarks are given in Section 5.4.

5.2. Factors Affecting Real-Time CDFD Filters in Two and Three

Dimensions

The signal processor described in Section 4.3 is ideally suited to real-time processing because it processes an input image row by row. The input image is often available in this row format, such as in a raster display. The ability to process such an image, without requiring one entire frame of the image to be read and stored by the system, represents a substantial saving in processing time. The

two major factors affecting the processing time are the time taken to perform a L_1 -point row FFT on the input row (of L_1 pixels) of the image and the time taken by the L_1 1D IIR filters to generate the next row of output samples.

One approach to these problems is a hardware intensive solution. Dedicated FFT hardware can be constructed for very fast row FFT operations based on bit-slice designs or a custom VLSI design. The IIR filtering operations can also be made faster by using L_1 parallel 1D IIR filters instead of the L_1 consecutive IIR filtering operations described in Section 4.3. This approach will prove to be very expensive especially when extended to 3D CDFD filtering.

The "row by row" approach to 2D CDFD real-time filtering can be extended to 3D CDFD filtering in real-time. One of the most common applications of 3D CDFD real-time filters would be in the processing of video signals. This usually involves processing a two-dimensional time-varying image. One approach to real-time 3D CDFD filtering is to perform a 2D FFT operation on each frame (size $L_1 \times L_2$) of the input signal and then apply $L_1 \times L_2$ IIR filtering operations in the remaining dimension, time. Finally, the output frame from the $L_1 \times L_2$ IIR filtering operations, one per pixel, is inverse FFT transformed to produce the output image. This method has the advantage that a time-varying image of specified size ($L_1 \times L_2$) but having an unspecified number of frames can be filtered as the input becomes available, frame by frame.

However, the above method clearly requires even more computations per output pixel than the "row by row" scheme for 2D real-time CDFD filters. An alternative approach to both 2D and 3D CDFD filtering is based on the fact that the input signal often has no imaginary component. In such cases, it is desirable to use the discrete Hartley transform (DHT), which transforms a real sequence into another real sequence, rather than the DFT which transforms a real sequence into a complex one. The next section outlines some of the advantages of using the DHT in CDFD filtering.

5.3. CDFD Filtering Using the Discrete Hartley Transform (DHT)

5.3.1. The Discrete Hartley Transform (DHT)

The MD discrete Hartley transform (DHT) $H_{1,M}^M[\]$ of a MD signal $x(\mathbf{k})$ is defined as follows

$$\begin{aligned} H_{1,M}^M [x(\mathbf{k})] &= X_H(\Omega) \\ &\triangleq \sum_{\mathbf{k}=0}^{\mathbf{L}-1} x(\mathbf{k}) \text{cas} [2\pi \mathbf{k}' \mathbf{L}^{-1} \Omega] \end{aligned} \quad (5.1)$$

where $\mathbf{L} = \text{diag} \{L_1, L_2, \dots, L_M\}$, $\sum_{\mathbf{k}=0}^{\mathbf{L}-1} \equiv \sum_{k_1=0}^{L_1-1} \sum_{k_2=0}^{L_2-1} \dots \sum_{k_M=0}^{L_M-1}$ and

$\text{cas}(x) \equiv \cos(x) + \sin(x)$. The "real-to-real" nature of the DHT is evident from equation (5.1). The MD inverse discrete Hartley transform (IDHT) is defined as

$$\begin{aligned}
H_{1,M}^{-M} [X_H(\Omega)] &= x(\mathbf{k}) \\
&\triangleq \frac{1}{|\mathbf{L}|} \sum_{\Omega=0}^{L-1} X_H(\Omega) \text{cas} \left[2\pi \mathbf{k}' \mathbf{L}^{-1} \Omega \right]
\end{aligned} \tag{5.2}$$

The partial MD DHT $H_{N+1,M}^M$ and IDHT $H_{N+1,M}^{-M}$ are defined as follows:

$$\begin{aligned}
H_{N+1,M}^M [x(\mathbf{k})] &= X_H(\mathbf{k}_c, \Omega_d) \\
&\triangleq \sum_{\mathbf{k}_d=0}^{L_d-1} x(\mathbf{k}_c, \mathbf{k}_d) \text{cas} \left[2\pi \mathbf{k}_d' \mathbf{L}_d^{-1} \Omega_d \right]
\end{aligned} \tag{5.3a}$$

and

$$\begin{aligned}
H_{N+1,M}^{-M} [X_H(\mathbf{k}_c, \Omega_d)] &= x(\mathbf{k}) \\
&\triangleq \frac{1}{|\mathbf{L}_d|} \sum_{\Omega_d=0}^{L_d-1} X_H(\mathbf{k}_c, \Omega_d) \text{cas} \left[2\pi \mathbf{k}_d' \mathbf{L}_d^{-1} \Omega_d \right]
\end{aligned} \tag{5.3b}$$

5.3.2. MD CDFD Filtering Using the DHT Instead of the DFT on Real Signals

The procedure for MD CDFD filtering, using DHT filtering in (M-N) dimensions and IIR filtering in the remaining N dimensions, is very similar to the CDFD filtering process described in Chapter 3. In fact, the partial MD DHT and MD IDHT are substituted for the partial MD DFT and IDFT in the filtering process. Thus, the following three step process is used:

1. Partial MD DHT is applied to input signal $x(\mathbf{k})$ resulting in the *real* signal $X_H(\mathbf{k}_c, \Omega_d)$.
2. ND recursive (IIR) filtering operations are performed on $X_H(\mathbf{k}_c, \Omega_d)$ over the N variables \mathbf{k}_c resulting in the MD *real* output $Y_H(\mathbf{k}_c, \Omega_d)$.
3. Partial MD IDFT is applied to $Y_H(\mathbf{k}_c, \Omega_d)$ resulting in the *real* periodically extended (in the $(M-N)$ dimensions Ω_d) output sequence $\tilde{y}(\mathbf{k})$. The output of the CDFD filter $y(\mathbf{k})$ is the part of $\tilde{y}(\mathbf{k})$ having support in \mathbb{R}^M .

The DHT and IDHT can be computed using fast Hartley transform (FHT) algorithms [7]. These algorithms require approximately the same amount of computation and memory as real-valued FFT algorithms [7]. This, in itself, does not represent a significant advantage over using the DFT. However, major computational savings are obtained because, unlike the DFT case, *complex filtering is not required* since the DHT transforms a *real* sequence into another *real* sequence. Thus, the required number of filtering operations is halved.

Furthermore, unlike the DFT case where the imaginary output of the CDFD filter is often ignored, no such approximation is necessary for the DHT. In real-time applications in 2D and 3D CDFD filtering, the DHT can significantly reduce the amount of computation required if the input signal is real.

5.4. Conclusion

A new hybrid MD filtering technique is described in this thesis. Continuous-frequency filtering is performed in less than all M dimensions of an MD signal while discrete-frequency filtering is performed over the remaining dimensions. This results in a powerful and flexible filtering technique - continuous-discrete frequency domain (CDFD) filtering.

A system of notation that allows the equations and symbols for CDFD filtering to be written concisely, is introduced. Some examples and applications of CDFD filters in 2D and 3D are described. Some important factors affecting the implementation of real-time CDFD filters in hardware are discussed.

Further work is required in this area and in the utilization of CDFD filters in 3D and higher dimensions. The flexibility and computational advantage of CDFD filters make them well suited to multidimensional digital filtering.

REFERENCES

- [1] R.C. Gonzalez and P. Wintz, "*Digital Image Processing*", Chapter 1, Addison-Wesley (1987).
- [2] L.T. Bruton, N.R. Bartley and R.A. Stein, "*The Design of Stable High-Quality Two-Dimensional Recursive Filters for Seismic Signal Processing*", Advances in Geophysical Data Processing, volume 2, pp. 223-261 (1985).
- [3] D.E. Dudgeon and R.M. Mersereau, "*Multidimensional Digital Signal Processing*", Chapters 1-5, Prentice-Hall (1984).
- [4] A.V. Oppenheim and R.W. Schaffer, "*Digital Signal Processing*", Chapter 3, pp. 87-171, Prentice-Hall (1975).
- [5] K.S. Lin, G.A. Frantz and R. Simar, "*The TMS320 Family of Digital Signal Processors*", Proceedings of the IEEE, pp. 1143-1159 (September 1987).
- [6] A. Antoniou, "*Digital Filters: Analysis and Design*", Chapters 1-5, 7-8, McGraw-Hill (1979).
- [7] H.V. Saveersen, D.L. Jones, C.S. Burns and M.T. Heideman, "*On computing the Discrete Hartley Transform*", IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. ASSP-33, No. 4 (October 1989).
- [8] G.T. Herman and A. Lent, "*Iterative Reconstruction Algorithms*", Computers in Biology and Medicine, 6 (1976), 273-294.

APPENDIX A

COMPUTATION AND MEMORY REQUIREMENTS FOR 2D CDFD, IIR AND DFT FILTERS

I) Memory Requirements (M)

Assume for simplicity each data value is stored in 1 byte of memory. Let the image size (signal size) be $L_1 \times L_2$ data points. Assume mask sizes for recursive filtering are small compared to image size. The following approximate values for memory required by each of the three methods are obtained:

IIR METHOD

Input-image size = $L_1 L_2$ bytes. Output image can be overlapped over input image if filter mask is small. Therefore

$$M_{IIR} \approx L_1 L_2 \text{ bytes .}$$

2D DFT METHOD

Input-image/output-image size = $L_1 L_2$ bytes. Real and Imaginary DFT transformed images require $2L_1 L_2$ bytes. Real windowing function requires $L_1 L_2$ bytes. Therefore,

$$M_{DFT} \approx 4L_1 L_2 \text{ bytes .}$$

CDFD METHOD

Input-image/output-image size = L_1L_2 bytes. Real and Imaginary DFT transformed images require $2L_1L_2$ bytes. Assume recursive filter mask is small and output can be overlapped over input. Then

$$M_{CDFD} \approx 3L_1L_2 \text{ bytes .}$$

II) Number of Additions Required (N_{ADD})

A row FFT of sequence length L requires $2L \log_2 L$ additions [2]. The input image is assumed to be of size $L_1 \times L_2$, where the FFT is applied to the columns (L_2 points) during CDFD filtering.

IIR METHOD

Assume filter mask is of size $m_o n_o$. Then

$$\begin{aligned} N_{ADD} &= 2[(2m_o n_o - 1)] L_1L_2 \\ &= (4m_o n_o - 2) L_1L_2 \end{aligned}$$

assuming two-pass filtering.

2D DFT METHOD

$$\begin{aligned} N_{ADD} &= 2(2L_1L_2 \log_2 L_2) + 2(2L_2L_1 \log_2 L_1) \\ &= 4L_1L_2(\log_2 L_1 + \log_2 L_2) \end{aligned}$$

because of forward and inverse DFT.

CDFD METHOD

Assume filter mask is of size n_o . Then,

$$\begin{aligned} N_{ADD} &= 2(2L_1L_2\log_2L_2) + 2(2n_o - 1)L_1L_2 \\ &= 2L_1L_2[2\log_2L_2 + (2n_o - 1)] \end{aligned}$$

because the real and imaginary parts of the partially DFT transformed image are filtered separately, each requiring $(2n_o - 1)$ additions per data point.

III) Number of Multiplications Required (N_{MULT})

A row FFT of sequence length L requires $4L \log_2 L$ multiplications [2]. During CDFD filtering, the 1D FFT is applied to each column (length L_2) of the input image.

IIR METHOD

Assume filter mask is of size $m_o n_o$ and two-pass filtering is performed. Then

$$N_{MULT} = 2(2m_o n_o - 1)L_1L_2$$

2D DFT METHOD

$$\begin{aligned} N_{MULT} &= 2(4L_1L_2\log_2L_2) + 2(4L_2L_1\log_2L_1) + 2L_1L_2 \\ &= 2L_1L_2(4\log_2L_1 + 4\log_2L_2 + 1) \end{aligned}$$

because of forward and inverse DFTs and the multiplication of the real and imaginary parts of the 2D DFTed image by a real window function.

CDFD METHOD

Assume filter mask is of size n_o and real and imaginary parts of partially DFT transformed image are filtered separately, each requiring $(2n_o - 1)$ multiplications per data point. Then, if column FFTs are applied,

$$\begin{aligned} N_{MULT} &= 2(4L_1L_2\log_2L_2) + 2(2n_o - 1)L_1L_2 \\ &= 2L_1L_2[4\log_2L_2 + (2n_o - 1)] \end{aligned}$$