

Department of Electrical  
and  
Computer Systems Engineering

Technical Report  
MECSE-1-2005

Multi-Dimensional Photonic Signal Processing

Le Nguyen Binh

**MONASH**  
UNIVERSITY

# MULTI-DIMENSIONAL PHOTONIC SIGNAL PROCESSING

Le Nguyen Binh

Department of Electrical and Computer Systems Engineering  
Monash university, Clayton Victoria 3168 Australia  
e-mail: [le.nguyen.binh@eng.monash.edu.au](mailto:le.nguyen.binh@eng.monash.edu.au)

## Summary

*Photonic communication systems and networks have progressed significantly over the last decade. Their operational regions have expanded in temporal, spatial and spectral domains, thus demand innovative techniques for processing of communication signals and spectral channels as well as spatially routing them. The information transport infrastructure is currently based and directionally developed towards super-dense wavelength division multiplexing with provisional routing and switching in spatial (e.g. routing to different sub-network elements) and spectral (e.g. multiplexing/demultiplexing, filtering, adding/dropping of wavelength channels) domain. The processing of photonic signals is becoming very important in these diverse domains.*

*Ultra high bandwidth properties of fiber-optic signal processing systems could provide the necessary processing power for computationally demanding two-dimensional signal processing applications. The techniques of fiber-optic signal processing so far have not been applied to the area of two-dimensional signal processing. Further the matured fields of integrated optics and integrated photonics as well as recent developments of nano-photonics allow innovative structures for processing of lightwaves in photonic domain.*

*This report/monograph has sought to integrate the fields of discrete signal processing and fiber-optic signal processing, integrated photonics and nano-photonics to establish a methodology based on which physical systems can be implemented. Because fiber-optics is essentially one-dimensional planar medium, the methodology has been proposed in order to implement 2-D signal processing using 1-D sources and processors.*

*A number of 2-D filter design algorithms are implemented. The algorithms are applicable to photonic filters that perform 2-D processing. The developed 2-D filter design methods are generic. Several photonic signal processing (PSP) architectures are proposed to enable efficient coherent lightwave signal processing. Although the structures are originally developed for 2-D processing, they are also applicable for 1-D structures. Using a combination of one-dimensional filter structures, 2-D fiber-optic filters are constructed. The relationship between the fiber-optic model and the mathematical model has been linked to allow quick implementation. Using the developed methodologies, multi-dimensional coherent photonic signal processors can be designed.*

*The technologies to support physical realization of such systems are not yet matured, therefore it is difficult to estimate the exact capabilities of such systems. However,*

*theoretically the architectures provide the potential for all-optical signal processing and it is envisaged that the processing bandwidth can reach the Tera-Hz region. We are proposing a number of multi-dimensional structures that would be realizable in the near future.*

## **TABLE OF CONTENTS**

<b>1.</b>	<b><u>INTRODUCTION TO PHOTONIC SIGNAL PROCESSING</u></b>	<b>7</b>
1.1	PHOTONIC SIGNAL PROCESSING: A BRIEF OVERVIEW	7
1.2	SPATIAL AND TEMPORAL APPROACH	8
1.3	FIBER-OPTIC DELAY LINE APPROACH	10
1.4	MOTIVATION	11
1.5	SUMMARY OF SECTION 1	11
<b>2.</b>	<b><u>MULTIDIMENSIONAL SIGNAL PROCESSING</u></b>	<b>12</b>
2.1	MULTI-DIMENSIONAL SIGNAL	12
2.2	DISCRETE DOMAIN SIGNALS	13
2.3	MULTI-DIMENSIONAL DISCRETE SIGNAL PROCESSING	15
2.4	SEPARABILITY OF 2-D SIGNALS	16
2.5	SEPARABILITY OF 2-D SIGNAL PROCESSING OPERATIONS	16
2.6	SUMMARY OF SECTION 2	18
2.7	THE OBJECTIVE	19
<b>3.</b>	<b><u>DEVELOPMENT OF FILTER DESIGN METHODS FOR 2-D PSP</u></b>	<b>19</b>
3.1	2-D FILTER SPECIFICATIONS	19
3.2	MATHEMATICAL MODEL OF 2-D DISCRETE PHOTONIC SYSTEMS	20
3.2.1	TRANSFER FUNCTION DESCRIPTION	20
3.2.2	STATE-SPACE EQUATION DESCRIPTION	23
3.3	FILTER DESIGN METHODS	26
3.3.1	DIRECT DESIGN METHODS	26
3.3.2	USE OF MATRIX DECOMPOSITION	27
3.4	SUMMARY OF SECTION 3	27
<b>4.</b>	<b><u>DIRECT 2-D FILTER DESIGN METHODS</u></b>	<b>28</b>
4.1	FIR AND IIR STRUCTURES IN 2-D SIGNAL PROCESSING	28
4.2	FREQUENCY SAMPLING METHOD	30
4.3	WINDOWING METHOD	32
4.4	McCLELLAN TRANSFORMATION METHOD	32
4.5	SUMMARY OF SECTION 4	36
<b>5.</b>	<b><u>MATRIX DECOMPOSITION METHODS</u></b>	<b>36</b>
5.1	SINGLE-STAGE SINGULAR VALUE DECOMPOSITION	36
5.2	MULTIPLE-STAGE SINGULAR VALUE DECOMPOSITION	41
5.3	ITERATIVE SINGULAR VALUE DECOMPOSITION	43
5.4	OPTIMAL DECOMPOSITION	45
5.5	OTHER 2-D FILTER DESIGN METHODS BASED ON MATRIX DECOMPOSITION	47
5.6	SUMMARY OF SECTION 5	48

<b>6.</b>	<b><u>2-D FILTER ORDER REDUCTION USING BALANCED APPROXIMATION THEORY</u></b>	<b>48</b>
6.1	MOTIVATION FOR LOWER ORDER PHOTONIC FILTERS	48
6.2	DESCRIPTION OF 2-D SYSTEM IN STATE-SPACE FORMAT	49
6.3	BALANCED APPROXIMATION METHOD	50
6.4	FILTER ORDER REDUCTION USING BALANCED APPROXIMATION: AN EXAMPLE	53
6.5	SUMMARY OF SECTION 6	56
<b>7.</b>	<b><u>FIBER-OPTIC DELAY LINE FILTERS</u></b>	<b>56</b>
7.1	COHERENT AND INCOHERENT OPERATION OF PHOTONIC FILTERS	56
7.2	USING OPTICAL FIBERS TO REALIZE DELAYED LINE FILTER	58
7.2.1	PHOTONIC REALIZATION OF DELAY	58
7.2.2	PHOTONIC REALIZATION OF TAB COEFFICIENTS	59
7.2.3	PHOTONIC REALIZATION OF SUMMER/SPLITTER	60
7.3	GRAPHICAL REPRESENTATION OF PHOTONIC CIRCUITS	62
7.4	SUMMARY OF SECTION 7	64
<b>8.</b>	<b><u>PHOTONIC IMPLEMENTATION OF 2-D FILTERS</u></b>	<b>65</b>
8.1	PHOTONIC FILTER STRUCTURES	65
8.2	COHERENT SYSTEM	66
8.3	2-D DIRECT STRUCTURE FILTER	66
8.4	2-D SEPARABLE STRUCTURE FILTER	68
8.5	BINARY TREE FILTER	69
8.6	PHOTONIC TRANSVERSAL FILTER	71
8.7	1-D DIRECT STRUCTURE PHOTONIC FILTER	75
8.8	PARALLEL STRUCTURE FILTERS	78
8.9	OTHER 1-D FILTER STRUCTURES	80
8.10	REALIZATION OF POLES	81
8.11	SUMMARY OF SECTION 8	83
<b>9.</b>	<b><u>DISCUSSIONS AND ANALYSIS</u></b>	<b>84</b>
9.1	SUMMARY OF THE REPORT	84
9.2	2-D PHOTONIC FILTER DESIGN FLOWCHART	84
9.3	EXAMPLES OF PHOTONIC 2-D PSP IMPLEMENTATION	86
9.3.1	THE SPECIFICATION	86
9.3.2	CHOICE OF A DESIGN METHODOLOGY	87
9.3.3	SEPARABLE IMPLEMENTATION USING MATRIX DECOMPOSITION METHODS	87
9.3.4	NON-SEPARABLE IMPLEMENTATION USING DIRECT METHODS	91
9.3.5	COMPARISON OF MATRIX DECOMPOSITION METHOD DESIGN AND DIRECT METHOD DESIGN	94
9.4	POSSIBLE AREAS OF APPLICATIONS	95
9.5	CONCLUDING REMARKS	96

## TABLE OF FIGURES

Figure 1.1: Spatial Fourier optical signal processor. The photonic active components are the acousto-optic diffractors.....	9
Figure 1.2: Fiber-optic delay line processor. The coupler can be replaced by a 3-port optical circulator. The reflector can be a fiber Bragg grating.....	10
Figure 2.1 (a) Infinite extent 1-D signal.(b) 2-D signal with finite predefined limit of $7 \times 7$ (each index refers to the crossing at the bottom-left corner of the grid it belongs to) and (c) The signal in (b) fed into 1-D signal processor.....	14
Figure 2.2: A separable 2-D signal processing operation.....	17
Figure 3.1: Illustration of the difference equation Eq. 3.2.1 .....	21
Figure 3.2: 1-D trapezoidal integrator signal flow diagram .....	21
Figure 3.3: 2-D trapezoidal integrator SFG diagram.....	22
Figure 3.4: General form of FIR filter signal flow diagram [15] .....	27
Figure 3.5: A subsection of filter in [16].....	27
Figure 4.1: Magnitude response of a filter designed using frequency sampling method.....	31
Figure 4.2: Magnitude response of the filter designed with transition band consideration.....	31
Figure 4.3: Magnitude response of filter of order $20 \times 20$ .....	32
Figure 4.4: Desired 2-D filter magnitude specification and the required 1-D filter specification.....	33
Figure 4.5: Transformation sequence used [10].....	34
(a) (b) Figure 4.6: (a) Frequency response of a filter of order $13 \times 13$ designed using transformation method (b) Impulse response (filter coefficients) of the filter designed by transformation method .....	34
Figure 4.7: 2-D filter designed using a different transform function .....	35
Figure 5.1: (a) Magnitude specification of low-pass filter (b) Magnitude response of $15 \times 15$ 2-D filter designed using single stage singular value decomposition .....	40
Figure 5.2: Separable implementation of 2-D filter using single-stage singular value decomposition .....	41
Figure 5.3 (a) Magnitude specification of $90^\circ$ fan filter (b) Magnitude response of $32 \times 32$ 2-D filter designed using multiple-stage singular value decomposition. ....	42
Figure 5.4: Errors and magnitudes of singular values .....	42
Figure 5.5: Structure of 2-D $90^\circ$ fan filter.....	43
(a) (b) Figure 6.1: (a) Ideal magnitude response of the filter and (b) Actual filter response of the $16 \times 16$ order filter .....	54
Figure 6.2: Hankel singular values of the filter.....	54
Figure 6.3: Reduced order( $10 \times 10$ ) filter (a) Ideal characteristic of the filter (b) Actual characteristic of the filter.....	55
Figure 7.1: Signal flow diagram of discrete-time tab filter, the unit delay is the traveling time of lightwaves over a distance equivalent to the unit sampling time . The coefficients $h_s$ are the transmittances over the specific path. ....	58
Figure 7.2: Schematic diagram of an optical coupler.....	60
Figure 7.3:(a) Optical coupler as a splitter (b) Optical coupler as a summer.....	61
Figure 7.4: Graphical representation of optical coupler.....	62
Figure 7.5: Schematic diagram of DCFBOR.....	63
Figure 7.6: Signal flow diagram representation of DCFBOR.....	64
Figure 8.1: Fiber/integrated optic implementation of direct structure.....	67
Figure 8.2: 2-D fiber-optic filter structure.....	68
Figure 8.3: Binary tree structure [1].....	70
Figure 8.4 (a) The original transversal structure [5] and (b) The proposed structure .....	72
Figure 8.5: Signal flow diagram of 1-D direct structure .....	76
Figure 8.6: Photonic implementation of direct 1-D structure.....	77
Figure 8.7: An arrangement of two $2 \times 2$ couplers to form a $3 \times 3$ coupler.....	77
Figure 8.8: Lattice form signal flow diagram.....	81
Figure 8.9: Denominator realization using feed forward structure .....	81
Figure 8.10: IIR filter structure using FIR subsections .....	82

Figure 8.11: A 1-D IIR filter subsection example .....	83
Figure 9.1: Magnitude specification.....	87
Figure 9.2: Schematic diagram of the separable 2-D filter.....	90
Figure 9.3: 1-D filter stage 1 .....	90
Figure 9.4: 1-D filter stage 2 .....	90
Figure 9.5: Magnitude response of the designed 2-D filter.....	91
Figure 9.6: Normalized filter magnitude response .....	94

## 1. INTRODUCTION TO PHOTONIC SIGNAL PROCESSING

---

---

*In recent years, there has been a notable increase in the number of applications that require an extremely fast signal processing speed that cannot be met by current all-electronic technology. Photonic signal processing (PSP) opens the possibilities for meeting the demands of such high-speed processing by exploiting the ultra high bandwidth capability of lightwave signals with specific applications in the field of photonic communications and fiber optic sensor networks.*

### 1.1 Photonic Signal Processing: A brief overview

---

The field of signal processing is concerned with the conditioning of a signal to fit certain required characteristics such as bandwidth, amplitude, and phase. Conventional techniques of signal processing make direct or indirect use of electronics. For example, frequency filtering, a most important signal processing procedure, can be performed through direct electronic means such as tunable IC filters or indirectly by digitizing the input for subsequent processing by computers or special purpose digital signal processing chips. Although a high performance can be obtained using either of the techniques, electronic methods suffer from physical limitations that govern the maximum processing speed. The demands for high performance beyond that achievable by electronic means have been increasing recently due to the increase in computationally demanding real-time processing applications.

Using lightwaves instead of electronic signals as the information carrier in signal processing is an appealing concept. The full potential of the technology has been accelerated in recent years due to the invention and discovery of photonic crystals. Several important advances have been made in utilizing light as the information carrier including real-time spatial-light modulators and electro-optic devices, micro-ring resonators, photonic crystal fibers, guided wave crystal photonics, super-prisms [8]. Another incentive for using light as the information carrier is the superiority of fiber-

optic communication systems which offer the wide bandwidth properties of photonic fiber medium. To fully exploit the capability of photonic systems, PSP is very essential.

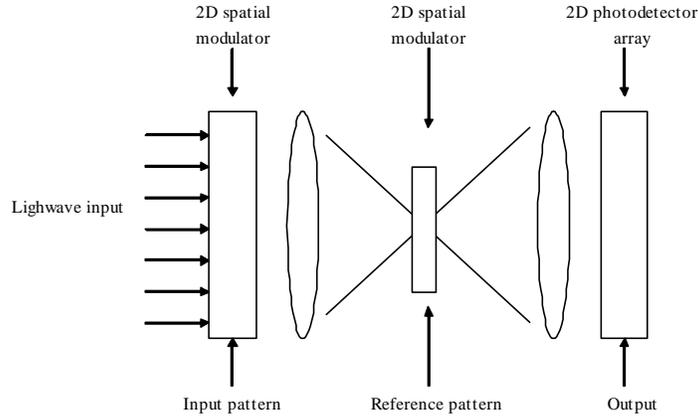
The field of PSP can be divided into two distinctive approaches which are outlined in the following sections.

## **1.2 Spatial and Temporal Approach**

---

The first use of lightwaves for signal processing applications was developed as early as in 1968 when an “integrated photonic correlator” [1] consisting of spatial light modulators and lenses in a planar waveguide was suggested. Further developments along this line were made and several experimental devices including acousto-optic spectrum analyser, a time-integrating acousto-optic correlator, a hybrid electro-optic/acousto-optic vector multiplier, a high-speed electro-optic analog-to-digital converter, and several fiber delay-line processors [2] were demonstrated.

The advantage of the spatial and temporal approach over the conventional electronic approach can be seen in light of the fact that lenses which are 2-D devices, have Fourier transform properties and can therefore act as a massively parallel Fourier transform processor. Taking advantage of the massive parallelism can mean the removal of the Von-Neumann bottleneck of present-day digital computers. Although an all-photonic computer does not seem feasible in the near future, a hybrid photonic-electronic computer offering ultra-high speed processing capability that could be realized by combining photonic information processing for some specific functions and electronics for general operation [2].



*Figure 1.1: Spatial Fourier optical signal processor. The photonic active components are the acousto-optic diffractors.*

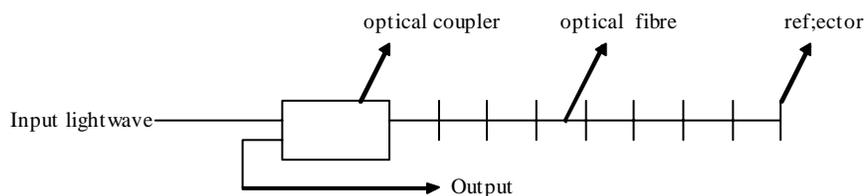
The drawback with spatial and temporal approach is the fact that the signal processing is performed in analogue manner. As shown in *Figure 1.1*, lightwaves carrying different signals must travel through different media therefore suffering acoustic diffraction resulting in crosstalk [2]. It is interesting to note that using holographic techniques, several layers of neural nets can be implemented with each layer in parallel format making spatial and temporal approach a suitable technique for neural network implementation. Although this technique may be useful in implementation of opto-electronic computer, the approach is not suitable for signals that have been transmitted through photonic fiber communication networks. Such signals are sequentially linear and to be processed by a spatial and temporal processor, a conversion into a suitable 2-D format using demultiplexing devices and laser arrays will be required. The following section introduces a technique which is ideal for lightwave signals from guided media such as photonic fibers and photonic crystals.

The spatial structures can be translated in fiber and integrated photonic forms using planar lightwave circuit (PLC) using silica-on-silicon technology, for example the array waveguide filters acting as wavelength muxes and demuxes and spatial separators.

### 1.3 Fiber-Optic Delay Line Approach

---

Guided-wave photonics and fiber optics provide alternative architectures for PSP to the classic spatial or time integrating architecture introduced in *Section 1.2*. The main advantage of guided-wave systems over spatial and temporal system is the wide bandwidth property available with photonic fiber transmission medium. For example, a silica fiber with a nominal  $5\mu\text{s}$  delay can store 1 GHz bandwidth signals for time periods less than one millisecond[2]. Another advantage of guided wave optics can be stated as the elimination of acoustic diffraction. However, since photonic fiber is essentially a 1-D medium (signal propagates along one axis - that of the fiber), this architecture sacrifices the 2-D nature of light that is utilized in time and space integrating architectures. In effect, in guided-wave systems, the advantage of massive 2-D parallel processing capability of light is sacrificed for the wide bandwidth of guided wave optics which enables high speed processing. Despite this limitation which confines the use of fiber-optic technology to signals from guided lightwave transmission medium, the simple fact that the current major usage of photonic systems is in communication systems makes the technology useful as it presents the possibilities of removing the bottleneck caused by opto-electronic conversion and therefore ensuring full utilization of fiber bandwidth. So far various uses have been found for fiber-optic signal processors as frequency filters, matched filters, correlators, and waveform and sequence generators [35].



*Figure 1.2: Fiber-optic delay line processor. The coupler can be replaced by a 3-port optical circulator. The reflector can be a fiber Bragg grating..*

*Figure 1.2* shows one possible configuration of fiber-optic processor. Although filter coefficients were realized using reflectors in *Figure 1.2*, other in-line components such as photonic attenuator/amplifier can also be used for implementing filter coefficients. It is evident that the operation of fiber-optic delay line filters is similar to that of digital

filters. In fact, the correct term to describe the fiber-optic signal processing would be ‘discrete-time PSP’ rather than digital signal processing as the range of the input or output signal is not digital at all. In any case, the discrete-time property makes it possible to apply the well developed z-transform techniques to filter design. In Section 2, the application of the z-transform techniques for analysis and design of fiber-optic systems is discussed in detail.

#### **1.4 Motivation**

---

The demand for multi-dimensional photonic signal processing (M-D PSP) can be attributed to various factors due the growing feasibility of high-capacity digital transmission networks capable of transmitting ultra-high bit rate and time division multiplexing up to 160 Gb/s as well as fiber optical sensor networks.

A problem with the implementation of such systems is the lack of devices that are capable of processing an enormous amount of data associated with multi-dimensional signals. With photonic transmission networks becoming the transport infrastructure, PSP technique has become increasingly more desirable compared to O/E and E/O conversion techniques. As discussed in Section 1.3, fiber-optic signal processing systems are ideal for such processing demands for several reasons: all-optical (or photonic) processing of photonic information of optical communication systems are possible using fiber-optic signal processing; 2-D signals usually require much higher bandwidth than 1-D signals and therefore must be processed by a high bandwidth system to allow real-time performance; it is likely that future telecommunication networks would be all fiber-optic.

#### **1.5 Summary of Section 1**

---

- The differences between electronic system and photonic system described.
- The basic operating principles of PSP using spatial and temporal architecture described.

- Fiber-optic delay line signal processing as an alternative to spatial and temporal approach introduced.

Table 1-1 outlines the differences between spatial-temporal approach and fiber-optic delay line approach.

	<b>Spatial and Temporal</b>	<b>Fiber-optic</b>
<b>Principle operating mode</b>	unguided	guided
<b>Components used</b>	lenses, light modulators, mirrors, masks, LED or laser arrays, slits	lasers or LED's, optical fibers, optical amplifiers(attenuators, reflectors)
<b>Time mode</b>	continuous-time	discrete-time
<b>Flexibility</b>	hard to change configuration once developed	easy to adjust the function using different tab values
<b>Analysis method</b>	difficult (some Fourier transforms)	well known z-transform method
<b>Accuracy</b>	low	high
<b>Cross-talk</b>	yes	no
<b>Major use</b>	Photonic computing	Communication signal processing
<b>Parallel processing capability</b>	massive parallel processing	limited parallel processing

Table 1-1: Outline of the two different approaches to PSP

## 2. MULTIDIMENSIONAL SIGNAL PROCESSING

*Multidimensional signal processing enables processing of signals that depend on more than one co-ordinate. Although many concepts of multidimensional signal processing are straightforward extensions of 1-D signal processing theory, there are also significant differences that need to be clarified, particularly when referred to photonics. Discussions of multidimensional signal processing in this report is limited to 2-D signal processing applicable to photonics that is by far the most important class of multidimensional signal processing.*

### 2.1 Multi-dimensional Signal

One may define multidimensional signals as signals whose values at a certain instance of time, space, or other coordinates depend on more than one variable. In 2-D signal processing, each of the properties depends on both x and y direction and therefore the

concepts of spatial signal, and therefore spatial frequency must be introduced. Spatial frequency does not depend on time, but rather depends on the spatial variations of the 2-D signal. There are two distinct spatial frequencies, one in x direction and one in y direction. 2-D signals form the most important class of multidimensional signals and methods developed for 2-D signal can be generalized to signals of larger dimensions. This report concentrates on developing filter design methods for two-dimensional signals.

## 2.2 Discrete Domain Signals

---

A signal domain can be either continuous or discrete. For digital signal processing purposes however, it is convenient to ‘sample’ continuous domain signals at a discrete interval so that in effect it has a discrete domain. In 1D, signal to be processed or stored in a sequential manner can be sampled at discrete intervals of time or direction. Put into an equation form, 1D signal can be represented by a train of scaled impulses as in Eq (2.2.1)

$$a(t) = \lim_{\Delta \rightarrow 0} \sum_{k=-\infty}^{\infty} a(k\Delta)\delta(t - k\Delta)\Delta \quad \text{Equation 2.2.1: 1D signal}$$

where  $\Delta$  is the sampling period and n denotes the sequence number. The sampled signal can be infinite in extent and reflects this accordingly. If  $\Delta$  is infinitely short, then above expression reduces to the representation of a continuous signal as expected.

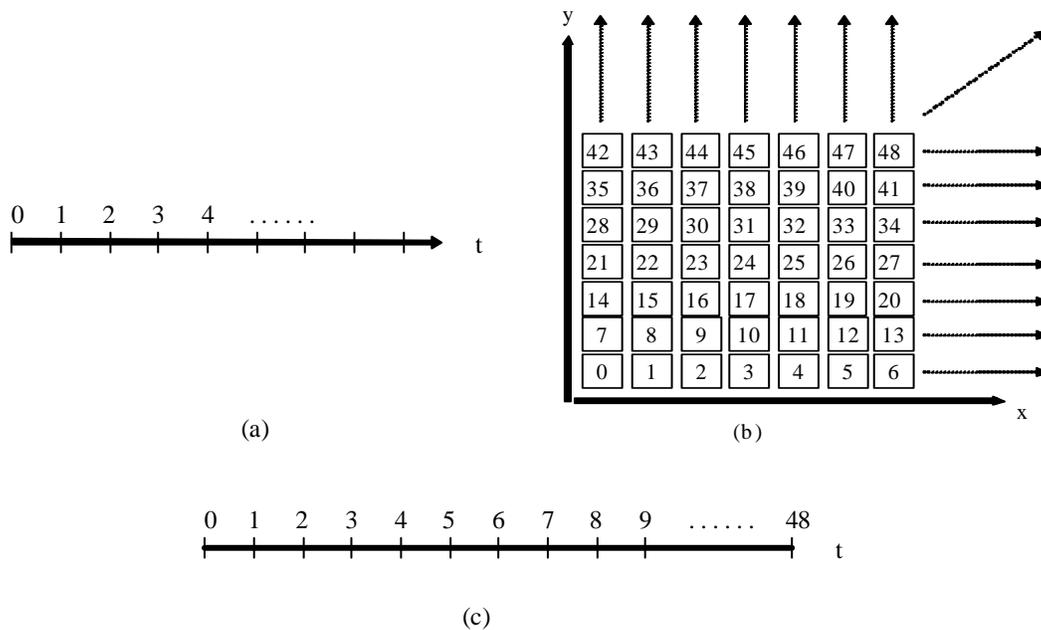
In 2-D, a natural extension to (2.2.1) can be made as shown in Eq(2.2.2).

$$a(x, y) = \lim_{\Delta_1 \rightarrow 0} \lim_{\Delta_2 \rightarrow 0} \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} a(k_1\Delta_1, k_2\Delta_2)\delta(x - k_1\Delta_1, y - k_2\Delta_2)\Delta_1\Delta_2 \quad \text{Eq. 2.2.2:}$$

*Sampled 2-D signal*

There is an important difference between sampling of 1D signals and 2-D signals in practice. Assuming there is only one sampling device, 1D signal such as the one shown in *Figure 2.1(a)* can be sampled by taking values at discrete intervals. If the signal duration is infinite in extent, no truncation is needed as the transfer function defines the limit even

if the signal is not periodic. For 2-D signals with infinite duration, this is not the case. As it can be seen in *Figure 2.1(b)*, if the 2-D signal was sampled infinitely in one dimension, the part of the 2-D signal which extends in the other dimension will never be sampled. For 2-D signals, there is always a predefined limit on how many samples are taken in each dimension. After reaching the limit in one dimension, the coordinate on the other dimension is incremented by one sampling period and the sampling process continues until the number of samples in the first dimension again reaches the limit. The process is repeated until the pre-defined number of samples in the second dimension is reached. The consequence is that the process gives a train of sampled 2-D signals stretched out in 1D as shown in *Figure 2.1(c)*. For 1-D discrete time processing of 2-D signals, the signal must be sampled in this way so that the processor can implement the delays  $z_1^{-1}$  and  $z_2^{-1}$  using only one dimensional delay photonic element. The limiting of sample space is similar to windowing or truncation performed on 1-D signals for some signal processing operations such as discrete Fourier transform (DFT).



*Figure 2.1 (a) Infinite extent 1-D signal.(b) 2-D signal with finite predefined limit of  $7 \times 7$ (each index refer to the crossing at the bottom-left corner of the grid it belongs to) and (c) The signal in (b) fed into 1-D signal processor*

Discrete space form of 2-D signal with predefined limits can be expressed by Eq.

2.2.3.

$$a[n_1, n_2] = \sum_{k_1=0}^{n_1} \sum_{k_2=0}^{n_2} a[k_1, k_2] \delta[n_1 - k_1, n_2 - k_2] \quad \text{Eq. 2.2.3: Discrete-space sampled 2-D}$$

*signal*

Using the above form of coding 2-D signals in a linear sequence, 2-D signal processing using 1-D medium such as optical fiber can be made possible.

### 2.3 Multi-dimensional Discrete Signal Processing

---

Having made a reasonable compromise in the size of the predefined limit, i.e. truncation window size, the Nyquist rate can be applied to 2-D signals to determine the sampling rate. In 1D, the Nyquist rate is twice the highest frequency component of the sampled signal and defines the sampling rate necessary to preserve the entire bandwidth of the signal.

In 2-D, the direction in which the Nyquist rate is applied must be made clear as sampling in one dimension at the Nyquist rate may not guarantee the preservation of the 2-D signal if the signal varies faster with respect to the other dimension. To preserve the entire 2-D signal bandwidth, sampling must be performed at twice the highest spatial frequency component of the 2-D signal in ANY direction in the sample space. For example, consider a signal which has a 20 GHz component in  $n_1$ -axis but has a 60 GHz component at  $70^\circ$  from  $n_1$ -axis. In this case, the sampling rate of 40 GHz in both dimensions is not adequate as the signal has a frequency component of  $60 \text{ GHz} \times \sin(70^\circ) = 56 \text{ GHz}$  along  $n_2$ -axis. Since the sampling rates in the both dimensions are usually kept the same, sampling rate of  $56 \times 2 = 112 \text{ GHz}$  in both dimensions will preserve the entire 2-D signal bandwidth.

In discrete-time signal processing, the term *normalized frequency* is used to describe a frequency independent of the system sampling frequency. The concept is applied in 2-D processing with a straightforward extension to spatial frequency.

## 2.4 Separability of 2-D Signals

---

A 2-D sequence is *separable* if it can be represented by a product of two 1-D sequences as shown in (2.4.1). Separable sequences form an important and special, but limited class of 2-D sequences. Many results in 1-D theory has a simple extension for separable 2-D sequences whereas for non-separable sequences such extensions often do not exist. If a 2-D sequence is separable, the separability can be exploited to reduce the processing requirements resulting in considerably less amount of computation. Unfortunately, most 2-D sequences are not separable.

$$s[n_1, n_2] = f[n_1]g[n_2] \qquad \text{Eq. 2.4.1: A separable 2-D sequence}$$

An example of a separable sequence is the unit sample sequence  $\delta(n_1, n_2)$  shown in (2.4.2) (a) [10]. Other examples of separable sequences include the unit step sequence  $u(n_1, n_2)$  and the example in (2.4.2) (b).

$$\delta[n_1, n_2] = \delta[n_1]\delta[n_2] \dots \dots \dots (a) \qquad a^{n_1}b^{n_2} + b^{n_1+n_2} = (a^{n_1} + b^{n_1})b^{n_2} \dots \dots \dots (b)$$

Eq. 2.4.2: Examples of 2-D sequences

## 2.5 Separability of 2-D Signal Processing Operations

---

Similar to 2-D sequences, a 2-D signal processing operation can be classified as separable or non-separable. The consequence of an operation being separable is that the operation yields the correct answer when it is performed in two independent cascade stages with each stage performing the operation with respect to only one of the independent variables. The situation is illustrated in *Figure 2.2*.

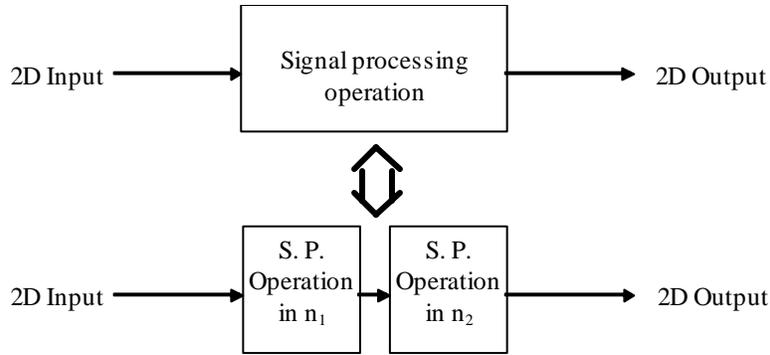


Figure 2.2: A separable 2-D signal processing operation

An example of a 2-D separable signal processing operation is double integration. A double integration procedure can be expressed by (2.5.1).

$$F(n_1, n_2) = \int_{n_2=-\infty}^{\infty} \int_{n_1=-\infty}^{\infty} f(n_1, n_2) dn_1 dn_2 \quad \text{Eq. 2.5.1: Double integration}$$

The 2-D sequence  $f(n_1, n_2)$  is integrated with respect to  $n_1$  first, and then with respect to  $n_2$ . The two procedures can be put in cascade and thus double integration operation is classified as a *separable operation*. Note that the separability of the 2-D input sequence  $f(n_1, n_2)$  is not a pre-requisite for the success of the operation. In addition, separable signal processing operations have separable impulse responses. The 2-D signal processing can be performed by convoluting the 2-D input with the 1-D filter impulse response in one dimension, and the operation can be completed by convoluting the result of the first convolution with the filter impulse response in the other dimension. It is therefore clear that the operations can be performed using a cascade stage of two filters. As with the case of separable sequences, separable operations form a special class of 2-D signal processing operations. Most signal processing operations are not separable.

In discrete domain, separable operations can be expressed in terms of a product of two z-transform transfer functions. For example, Eq. 2.5.2 is double integration using Simpson's rule for digital integration [11].

$$H(z_1, z_2) = \frac{T_1}{2} \left[ \frac{1 + z_1^{-1}}{1 - z_1^{-1}} \right] \cdot \frac{T_2}{2} \left[ \frac{1 + z_2^{-1}}{1 - z_2^{-1}} \right] \quad \text{Eq. 2.5.2: Digital double integration}$$

*transfer function using Simpson's rule*

It is clear that  $H(z_1, z_2)$  is separable. For other functions, the separability is often not the case. A circularly symmetric 2-D digital low pass filter for example (see Eq. 2.5.3) cannot be separated into a product of two functions each dealing with only one kind of delays ( $z_1$  or  $z_2$ ). In such cases, a way of dealing with non-separability must be found as cascade stages will no longer work. It is the non-separability of most 2-D signal processing functions that makes implementation of 2-D filters a difficult task.

$$H(z_1, z_2) = 1 + 0.9z_1^{-1} + 0.9z_2^{-1} + 0.8z_1^{-1}z_2^{-1} - 0.1z_1^{-2} - 0.05z_1^{-1}z_2^{-2} - 0.05z_1^{-2}z_2^{-1} - 0.1z_2^{-2} + 0.1z_1^{-3} + 0.1z_1^{-3}z_2^{-1} + 0.07z_1^{-3}z_2^{-2} + 0.07z_1^{-2}z_2^{-3} - 0.05z_1^{-3}z_2^{-3}$$

*Eq. 2.5.3: A version of 2-D digital low pass filter*

Another advantage of having a separable implementation is the issue of stability. The stability analysis of non-separable filters is very difficult and there are no known simple methods of checking the stability of 2-D filters directly from the transfer function or from pole-zero plots as is the case with 1-D systems<sup>1</sup>. However, with separable filters if 1-D sub-sections are stable, then the overall stability is guaranteed. Stability of 1-D filters can be guaranteed by having all system poles inside the unit circle.

## 2.6 Summary of Section 2

---

Although *Sections 2.4* and *2.5* dealt with 2-D signals, the result is extendible to systems with larger dimensions.

- The concept of multidimensional signal, and the term spatial frequency are introduced.

---

<sup>1</sup> For more detailed discussion on stability checking using position of poles, refer to Section 4, [25].

- Discrete-domain signal processing results are extended to 2-D systems.
- Separability of sequences and signal processing operations are outlined and their significances illustrated.

## 2.7 The Objective

---

*The objective of this section is the design methodology for developing a 2-D signal filter suitable for PSP using the fiber-optic and PLC signal processing architecture.*

## 3. DEVELOPMENT OF FILTER DESIGN METHODS FOR 2-D PSP

---

---

*In Section 2, the concepts of 2-D signal processing have been introduced. Out of many possible mathematical models for 2-D systems, the model best suited to fiber-optic signal processing must be found. In this section, two different mathematical models of representing 2-D systems are presented and a brief introduction to 2-D filter design methods is given.*

### 3.1 2-D Filter Specifications

---

To specify a filter, two approaches can be adopted. One approach specifies a filter in mathematical form by specifying the transfer function or the state-space equations of the filter. This method can specify the exact behavior of the filter. The other approach specifies a filter by its transfer characteristics of magnitude and phase response or impulse response of the filter. This later approach is more intuitive than the former because it is easy to see how the filter would behave in practical implementation. However the accuracy of the filter then depends on the accuracy of the specification therefore can sometimes be inadequate. In any case, the later approach must go through the mathematical description before implementation. Developing a method for designing and implementing a filter from its dynamic characteristics therefore encompasses the mathematical description.

The method developed in this section assumes the spatial frequency responses of the filter to be specified. The 2-D photonic filter design process can be as follows: Specification of magnitude or impulse response the desired 2-D filter; Development of transfer function or state-space description of the 2-D filter; Development of signal flow diagram of the 2-D filter and Development of photonic implementation of the 2-D filter.

To specify a filter using its frequency response, both magnitude and the phase responses need to be supplied. However, designing a filter with a certain phase response is a very difficult task. In many cases of interest, a condition of linear phase is all that is required and in this report, the condition is adhered to. The reason for requiring a linear phase can be explained by the Fourier transform of a 1-D linear phase filter, Eq. 3.1.1.

$$V(f)e^{-j\phi\omega} \Leftrightarrow v(t - \phi) \quad \text{Eq. 3.1.1: Fourier transform of linear phase filter}$$

In Eq. 3.1.1, the phase  $\phi\omega$  is proportional to frequency. The Fourier transform (FT) of linear phase on the right of Eq. 3.1.1 shows that a linear phase corresponds to pure time delay. The result is extendible to 2-D simply by substituting frequency by spatial frequency and time delay by spatial delay. A non-linear phase response leads to uneven delays and therefore inter-symbol interference (ISI).

## 3.2 Mathematical Model of 2-D Discrete Photonic Systems

---

### 3.2.1 Transfer Function Description

2-D transfer function description of the filter can also be explained by using a 2-D difference equation. As in 1-D, 2-D transfer functions can readily be turned into 2-D difference equations (see Eq. 3.2.1).

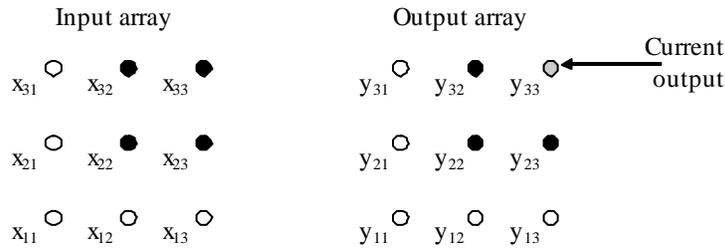
$$H(z_1, z_2) = \frac{T_1}{2} \left[ \frac{1 + z_1^{-1}}{1 - z_1^{-1}} \right] \cdot \frac{T_2}{2} \left[ \frac{1 + z_2^{-1}}{1 - z_2^{-1}} \right]$$

The equivalent difference equation is given by

$$y(n_1, n_2) = \frac{T_1 T_2}{4} [x(n_1, n_2) + x(n_1 - 1, n_2) + x(n_1, n_2 - 1) + x(n_1 - 1, n_2 - 1)] \\ + y(n_1 - 1, n_2) + y(n_1, n_2 - 1) - y(n_1 - 1, n_2 - 1)$$

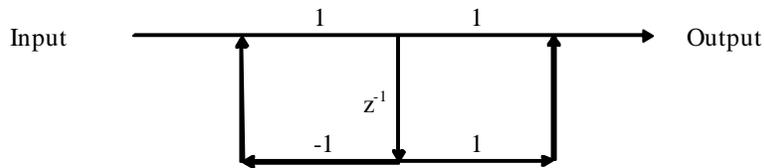
*Eq. 3.2.1: 2-D trapezoidal integrator difference equation.*

Figure 3.1 illustrates the sample points which are summed in  $y(n_1, n_2)$  of Eq. 3.2.1. Because we are dealing with spatial delay and not time delay, the actual implementation of delay depends on the signal transmission format. If all points of the 2-D signal are transmitted in parallel, then the delays need not be time delay raising the possibility of parallel processing similar to that of spatial and temporal architecture described in Section 1.2. Further discussion on the issue is presented in Section 9.



*Figure 3.1: Illustration of the difference equation Eq. 3.2.1*

A 1-D integrator transfer function amounts to just the first half or the second half of  $H(z_1, z_2)$  in Eq. 3-2 and can readily be turned into a signal flow graph (SFG) as shown in Figure 3.2.



*Figure 3.2: 1-D trapezoidal integrator signal flow diagram*

The SFG of 2-D version of the trapezoidal integrator transfer function  $H$  is shown in Figure 3.3. A notable difference is the presence of two different delay elements. In spatial terms, one represents the vertical delay and the other represents the horizontal delay.

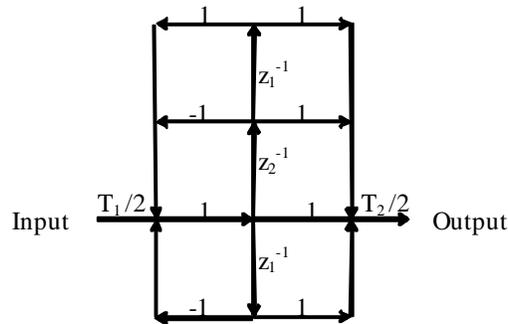


Figure 3.3: 2-D trapezoidal integrator SFG diagram

Whether a SFG can readily be turned into photonic domain depends on its structure. A major obstacle that prevents a direct translation of SFGs into photonic circuits is the number of complex interconnections. Too many complex interconnections can result in the loss of modularity making the photonic implementation of the transfer function difficult.

Transfer functions can be manipulated into potentially useful forms for photonic implementation. One such manipulation technique is called *continued fraction expansion realization (CFER)* (see *Box 3-1*). Although the method results in reduced number of filter components, not all transfer function can be expanded easily. A method of designing a filter transfer function that can be expanded using continued fraction expansion is therefore required. Such a design method does not exist currently and as a consequence the use of CFER is confined to only transfer functions that are expandable.

***Box 3-1: Continued Fraction Expansion***

Given a transfer function, the numerator of the transfer function is recursively long-divided by the denominator until the remainder is only a simple fraction. A possible form for a fraction which has been expanded using continued fraction expansion is shown in Eq. 3-3.

$$H(z_1, z_2) = C_1 + \frac{1}{A_1 z_1 + \frac{1}{C_2 + \frac{1}{B_1 z_2 + \frac{1}{\ddots}}}}$$

*Eq. 3.2.2: An example of continued fraction expansion format [16]*

It is intuitively obvious that such expansions do not exist for all polynomial fractions. A method of checking the existence of such expansion is given in [16].

### 3.2.2 State-Space Equation Description

State-space description can be seen as an alternative description method to the transfer function description. The advantages offered by state-space description include the notion of observability and controllability. Although such concepts are useful in 2-D dynamic control system, the applications of the concepts are not obvious in 2-D signal processing. At best, the main advantage of using state-space approach can be stated as the established techniques of 1-D state-space theory such as algorithms to manipulate state-space matrices to obtain a reduced order system (see *Section 6*).

#### 3.2.2.1 An Algorithm for Conversion of a 2-D Transfer Function into 2-D State-space Equation

In [14], a 2-D state-space description is formulated from a 2-D FIR transfer function description using the following method. Ref.[12] has given a more generalized case of transfer functions of 2-D IIR filters.

#### **Box 3-2: Formulation of state-space Eq.s from transfer function[14]**

A 2-D FIR transfer function can be expressed by

$$H(z_1, z_2) = \sum_{n_1=0}^{N_1} \sum_{n_2=0}^{N_2} h(n_1, n_2) z_1^{-n_1} z_2^{-n_2} \quad \text{Eq. 3.2.3: 2-D FIR transfer function}$$

State-space form can be expressed as

$$\begin{aligned} \begin{bmatrix} x^h(i+1, j) \\ x^v(i, j+1) \end{bmatrix} &= \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ \mathbf{A}_3 & \mathbf{A}_4 \end{bmatrix} \begin{bmatrix} x^h(i, j) \\ x^v(i, j) \end{bmatrix} + \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix} u(i, j) \\ &\equiv \mathbf{Ax} + \mathbf{bu} \end{aligned} \quad \text{(a)}$$

$$y(i, j) = \begin{bmatrix} \mathbf{c}_1 & \mathbf{c}_2 \end{bmatrix} \begin{bmatrix} \mathbf{x}^h(i, j) \\ \mathbf{x}^v(i, j) \end{bmatrix} + du(i, j)$$

$$\equiv \mathbf{c}\mathbf{x} + du$$

(b)

where

$$\mathbf{A}_1 = \begin{bmatrix} \mathbf{0} & \mathbf{I}_{N_1-1} \\ 0 & \mathbf{0} \end{bmatrix}, \quad \mathbf{A}_2 = \begin{bmatrix} h_{1N_2} & \cdots & h_{11} \\ \vdots & \ddots & \vdots \\ h_{N_1N_2} & \cdots & h_{N_11} \end{bmatrix}$$

$$\mathbf{A}_3 = \mathbf{0}, \quad \mathbf{A}_4 = \begin{bmatrix} \mathbf{0} & \mathbf{I}_{N_2-1} \\ 0 & \mathbf{0} \end{bmatrix}$$

$$\mathbf{b} = [h_{10} \quad \cdots \quad h_{N_10} \mid 0 \quad \cdots \quad 0 \quad 1]^T$$

$$\mathbf{c} = [1 \quad 0 \quad \cdots \quad 0 \mid h_{0N_2} \quad \cdots \quad h_{01}]$$

$$d = h_{00}$$

*Eq. 3.2.4: 2-D state-space representation*

### 3.2.2.2 An Algorithm to Convert a 2-D State-space equations into a 2-D Transfer Function

For state-space approach to be useful, there must be a method of converting 2-D state-space Eqs into 2-D transfer function form. An algorithm to perform such a task could not be found in standard text books of digital signal processing, and therefore it had to be devised independently. The algorithm given in this section performs conversion from a 2-D state-space equations specified in *Box 3-2* to a 2-D transfer function description.

By rearranging Eq. 3.2.5, we can obtain the transfer function form of the same system with input denoted by  $\mathbf{x}$  and output denoted by  $\mathbf{y}$  as in *Eq. 3.2.6*.

$$H(z_1, z_2) = \mathbf{C} \left[ \begin{pmatrix} z_1 \mathbf{I}_m & \\ & z_2 \mathbf{I}_n \end{pmatrix} - \mathbf{A} \right]^{-1} \mathbf{B} \quad \text{Eq. 3.2.5: Transfer function from state-space}$$

*description*

What makes the implementation of above equation difficult is the presence of matrix inverse. Because the matrix inside the bracket in

$$H(z_1, z_2) = \sum_{n_1=0}^{N_1} \sum_{n_2=0}^{N_2} h(n_1, n_2) z_1^{-n_1} z_2^{-n_2}$$

is expressed in matrix form as Eq. 3.2.7

$$Z = \begin{bmatrix} h_{00} & h_{01} & \cdots & h_{0n} \\ h_{10} & h_{11} & \cdots & h_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ h_{m0} & h_{m1} & \cdots & h_{mn} \end{bmatrix}$$

Eq. 3.2.6

describes a 2-D system, the determinant of the matrix contains two independent variables  $z_1$  and  $z_2$  and cannot therefore be solved by simply obtaining the eigenvalues of the matrix and cross-multiplying to get coefficients of variables as in 1-D determinant calculation.

**Box 3-2: An Algorithm to obtain the characteristic polynomial of a matrix describing a 2-D system**

1. Let A be the matrix of size  $m \times n$  describing a 2-D system and Z be a zero matrix of size  $m \times n$ .
2. Let  $z_1 = 0, z_2 = 0$ .
3. Let A' be a matrix formed by eliminating  $m - z_1$  rows and  $n - z_2$  columns from matrix A.
4. Let  $Z_{z_1, z_2} = Z_{z_1, z_2} + \Delta(A')$ .  $\Delta$  is 1-D determinant operator.
5. Repeat step 3 and 4 until all combinations of  $z_1$  rows and  $z_2$  columns are tried.
6. Repeat step 3 to 5 with different value of  $z_1$  and  $z_2$  until all coefficients of the characteristic polynomial Z are found.
7. Reverse the signs of elements of Z whose indices sum to an odd number.

The resulting matrix Z contains the coefficients of 2-D characteristic polynomial in format shown in Eq. 3.2.8.

$$H(z_1, z_2) = \sum_{n_1=0}^{N_1} \sum_{n_2=0}^{N_2} h(n_1, n_2) z_1^{-n_1} z_2^{-n_2}$$

is expressed in matrix form as Eq. 3.2.7: 2-D characteristic polynomial

$$Z = \begin{bmatrix} h_{00} & h_{01} & \cdots & h_{0n} \\ h_{10} & h_{11} & \cdots & h_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ h_{m0} & h_{m1} & \cdots & h_{mn} \end{bmatrix}$$

For the 2-D transfer function description, denominator and the numerator can be calculated by

$$\begin{aligned} \mathbf{h}_a &= \det(\mathbf{A}) \\ \mathbf{h}_b &= \det(\mathbf{A} - \mathbf{B} \times \mathbf{C}) - \det(\mathbf{A}) \times (D - 1) \end{aligned} \quad \text{Eq. 3.2.8: Transfer function numerator and denominator}$$

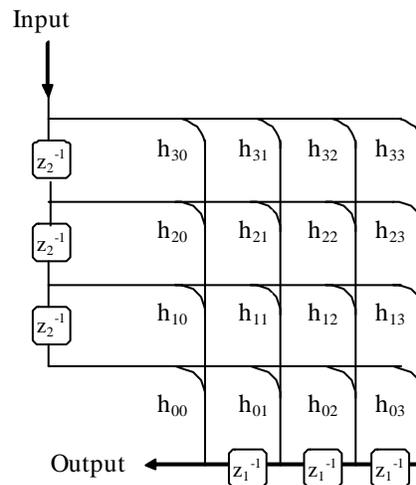
### 3.3 Filter Design Methods

---

#### 3.3.1 Direct Design Methods

Direct design methods include *window method*, *frequency sampling method*, *transformation method* for FIR filter implementations, and *impulse response method* for IIR filters. All of the design methods listed are similar to the 1-D methods of the same name involving some extensions of the concepts into 2-D and they all result in non-separable transfer functions.

The general format of non-separable filter transfer functions generated by FIR filter design methods is given in Eq. 3.2.4 and the most general form of 2-D FIR filter signal flow diagram is shown in Figure 3.4. In *Section 4*, the details of the algorithms and the implementations of 2-D direct design FIR filters are discussed. Direct design methods for IIR filters are also discussed in *Section 4.1* and *Section 8.10*.





- The outline of filter design steps is presented.
- Algorithms to convert between state-space description and transfer function description are given.
- Two approaches of 2-D filter design procedure, namely direct design and matrix decomposition design are introduced.

#### 4. DIRECT 2-D FILTER DESIGN METHODS

---

---

*As the first of two streams of 2-D filter design methods, direct 2-D filter design methods are introduced. This class of 2-D filter design methods does not use matrix decomposition and instead uses 2-D magnitude specifications directly to produce non-separable designs.*

##### 4.1 FIR and IIR Structures in 2-D Signal Processing

---

A digital filter can be divided into two broad classes, FIR (Finite Impulse Response) and IIR (Infinite Impulse Response). FIR filters only use feed forward structure and therefore are non-recursive, whereas IIR filters use feedback as well as feed forward structure and therefore are recursive. The impulse response of an IIR filter is infinite in duration, therefore the name ‘infinite impulse response filter’.

Given a 2-D frequency response specification, one can either try to formulate a FIR filter transfer function or an IIR filter transfer function. There are several factors which must be considered when deciding which structure to implement for a given frequency response specification.

- *Linearity of the phase response of the filter*
- *Stability of the filter*
- *Order of the implemented filter*

**Linearity** of the designed filter's phase response is very important as explained previously in *Section 3.1*. Linear phase FIR filters are very easy to design as the condition for the linear phase is simply a symmetric impulse response which in turn is guaranteed if the magnitude response of the 2-D filter is symmetric about the two axis. For 2-D IIR filters, phase linearity is much more difficult to guarantee. Often IIR filters are specified only with magnitude characteristics and the phase response is generally accepted for what it is (which is non-linear). The lack of control over phase response of IIR filters limits its usefulness in many applications [10].

**Stability** is a very important issue in designing of any dynamic system which requires no explanations. The advantage of 2-D FIR filter over 2-D IIR filter regarding the issue of stability is that for 2-D FIR filters, stability is inherent in its definition. Since the impulse response of FIR filter is finite in duration, bounded input results in bounded output and the filter is therefore always stable. Although 2-D IIR filters can be designed to be stable, as mentioned in *Section 2.5* there is no simple algorithm for checking the condition for stability of a 2-D IIR filter. A mathematical theory involving complex cepstrum to check for the stability condition of 2-D filters is quite involved and most algorithms for checking the 2-D stability simply repeat 1-D stability condition over the 2-D space many times over which can be computationally very inefficient [10]. As a consequence of the lack of usable algorithms or simple method for stability testing, there is no known method of designing stable 2-D IIR filters [10]. In practice, 2-D FIR filters are therefore much more preferred to 2-D IIR filters.

**Order** of the filter refers to the number of delay elements in the numerator (or the denominator if the filter is IIR). Order of the filter has a direct consequence in the final implementation of the filter as the determining factor of number of processing elements. Higher order filters require more processing elements than lower order filters which makes low order filters more desirable. One distinctive advantage of IIR filters over FIR filters is that the order of the filter required for a given magnitude specification is smaller. FIR filters can sometimes require an excessive order (the definition of 'excessive' depends on the implementation medium - for example in software implementation of a digital filter, a 1000th order filter might be quite acceptable but with fiber-optic delay

line filters, the maximum order is well below 50). For example an integrator, which has an infinite impulse response by nature can be described using a first order filter within 12.5% error [11] whereas achieving the same error with 1st order FIR filters would be impossible. In addition, 2-D filters usually require much higher order filters than 1-D filters of similar transition band requirements and it is therefore important that the 2-D filter design methods keep the order of the filter to an acceptable level.

## 4.2 Frequency Sampling Method

---

As the first of the direct 2-D filter design methods, 2-D frequency sampling method produces an FIR filter with the minimum of fuss. Using the fact that the transfer function of a FIR filter is same as the impulse response of the filter, 2-D frequency sampling method takes discrete Fourier transform of even-spaced samples of 2-D frequency response and uses the result as the coefficients for 2-D transfer function. It is noted that the procedure is identical to that of 1-D frequency sampling method.

It is observed in [10] that the filter designed using frequency sampling method is not optimal as far as number of delay elements is concerned. Also, the frequency response of the filter is controlled only by the sampling rate of the frequency sampling and the nature of the frequency samples. For example, increasing the frequency sampling rate will increase the number of discrete sampling points of the impulse response and hence will result in a filter with a better frequency response but with a larger order. It is also found that frequency response can be improved considerably especially around the transition band if the ideal frequency response takes account of the transition frequency values.

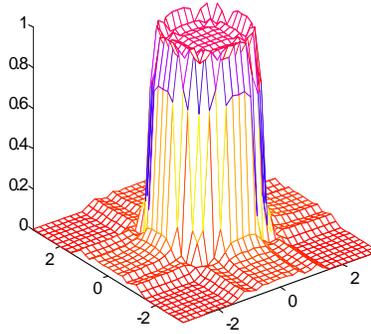
### ***Example 4-1: A filter design example using frequency sampling method***

**Design Aim:** A low pass filter with normalized cut off frequency of 0.5 in both dimensions.

**Method:** Frequency sampling method

**Program used:** FIR2-DFS.m

**Result:** The magnitude response of the designed filter is shown below. Filter is of  $33 \times 33^{\text{rd}}$  order and the error is 3.63%.

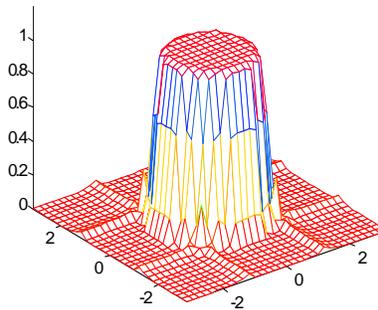


*Figure 4.1: Magnitude response of a filter designed using frequency sampling method*  
 The filter error is calculated using Eq. 4-1 where  $2N_1+1$  and  $2N_2+1$  are order of the filter in  $n_1$  and  $n_2$  dimension, respectively.  $H_d$  is the ideal frequency response,  $H_f$  is the actual filter response, and  $\Omega_1$  and  $\Omega_2$  are the frequency sampling rates.

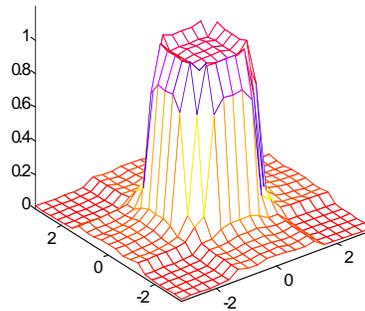
$$e = \frac{\sum_{n_1=-N_1}^{N_1} \sum_{n_2=-N_2}^{N_2} |H_d[n_1\Omega_1, n_2\Omega_2] - H_f[n_1\Omega_1, n_2\Omega_2]|}{(2N_1 - 1)(2N_2 - 1)}$$

*Eq. 4.2.1: Filter error calculation formula*

The magnitude response shown below is obtained by incorporating the transition band values into the ideal frequency response parameter. It can be seen that the ripple in the transition band has disappeared and the error is found to be only 2.74% which is nearly 1% less than that obtained without any transition band consideration.



*Figure 4.2: Magnitude response of the filter designed with transition band consideration*  
 The response shown below is obtained with a filter of order  $20 \times 20$ . The error is found to be 5.85% which compares unfavorably with 3.63% obtained with the first design. Clearly lower order filters result in considerably worse performance.



*Figure 4.3: Magnitude response of filter of order 20×20*

### 4.3 Windowing Method

---

The window method for 2-D filters use a 2-D window instead of 1-D window to achieve a finite impulse response sequence in 2-D. As with 1-D windowing method, the 2-D windowing method begins by performing a Fourier transform on the desired frequency response expression. The impulse response of the filter in 2-D is then multiplied by the expression for window. The purpose of multiplying by a windowing function is to reduce the effect of sharp transitions in the transition band and also to make the impulse response finite through truncation. The windowing function is chosen so that the frequency response is least affected and the impulse response is as short as possible. It is noted in [10] that the performances of window method and the frequency sampling method are similar and an example is therefore omitted.

### 4.4 McClellan Transformation Method

---

McClellan transformation method takes an entirely different approach to the design process of 2-D filters. The idea is to transform a 1-D FIR filter into a 2-D filter of the desired characteristic. The 1-D filter can be designed using any 1-D filter design method so that its frequency response is a cross-section of the desired 2-D filter response (see Figure 4.4).

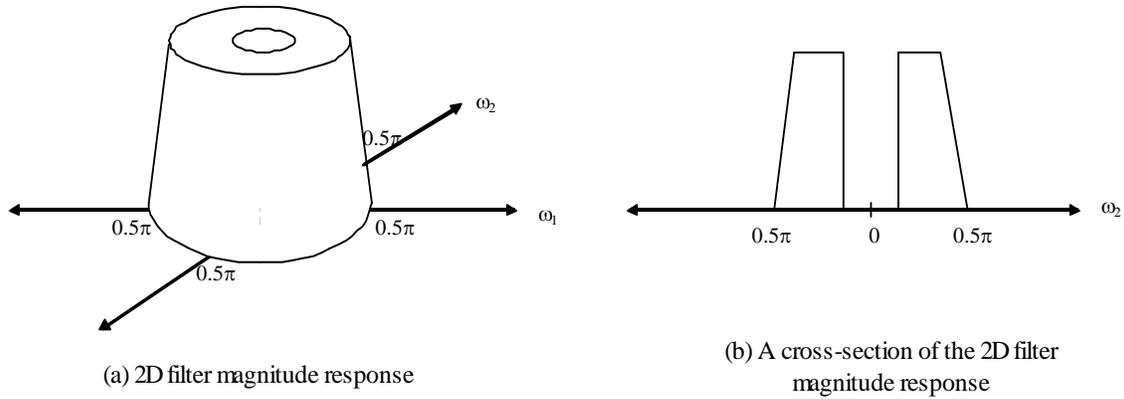


Figure 4.4: Desired 2-D filter magnitude specification and the required 1-D filter specification

The transformation procedure can be described mathematically by Eq. 4-2. Given the transfer function of 1-D FIR filter, each coefficient is multiplied by the transformation function  $T$  which is a function of  $\omega_1$  and  $\omega_2$ . The resulting transfer function is also a function of  $\omega_1$  and  $\omega_2$  and describes a 2-D filter with the desired magnitude response.

$$H(\omega_1, \omega_2) = h(0) + \sum_{n=1}^N 2h(n) \cdot [T(\omega_1, \omega_2)]^n$$

where

Eq. 4.4.1: 2-D filter transformation

$$T(\omega_1, \omega_2) = \sum_{(n_1, n_2) \in R_T} t(n_1, n_2) \cdot e^{-j\omega_1 n_1} \cdot e^{-j\omega_2 n_2}$$

In Eq. 4-2,  $R_T$  is the region of support of  $t(n_1, n_2)$  which describes the transformation function. It should be noted that using different transformation functions, many 2-D filters can be designed from a single 1-D filter. It is also important to note that as long as the 1-D filter is a linear phase filter, the transformed 2-D filter is also a linear phase filter as long as phase of the transformation function  $T$  is linear (i.e. the transfer function is symmetric about the zero delay point - see Eq. 4-3 in Example 4-2) since multiplying a linear phase function by another linear phase function does not affect the linearity of the phase of the resultant function.

**Example 4-2: 2-D filter design using transformation method**

Design Aim: 2-D low pass filter with frequency cut off at 0.5 in both dimensions

Method: McClellan transformation method. The filter order is 13×13 and the transformation function used is given by

$$T(\omega_1, \omega_2) = -\frac{1}{2} + \frac{1}{4}e^{-j\omega_1} + \frac{1}{4}e^{-j\omega_2} + \frac{1}{4}e^{j\omega_1} + \frac{1}{4}e^{j\omega_2} \\ + \frac{1}{8}e^{-j\omega_1} \cdot e^{-j\omega_2} + \frac{1}{8}e^{j\omega_1} \cdot e^{j\omega_2} + \frac{1}{8}e^{j\omega_1} \cdot e^{-j\omega_2} + \frac{1}{8}e^{-j\omega_1} \cdot e^{j\omega_2}$$

Eq. 4.4.2: Transformation function used in this example

Graphically, above transformation function can be expressed as in Figure 4.5.

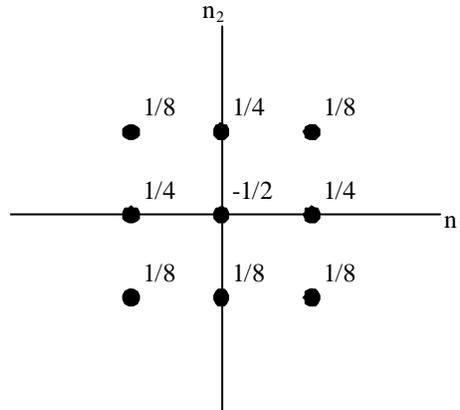


Figure 4.5: Transformation sequence used [10]

Program used: FIR2-DTF.m

Result: The error calculated = 13.71%.

Figure 4.6 shows the frequency response of the designed filter

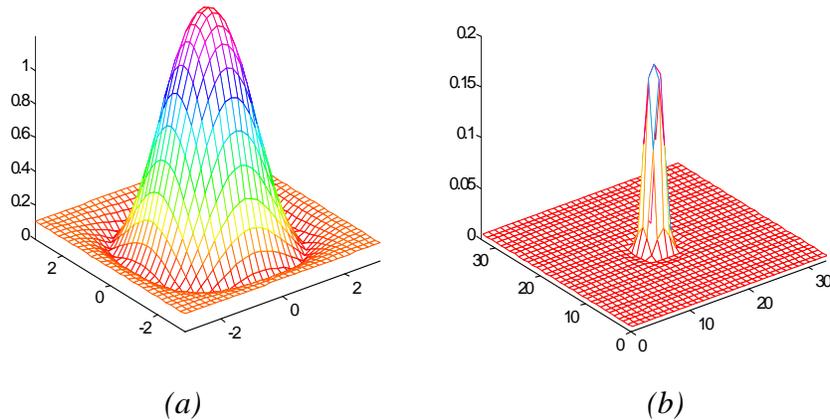


Figure 4.6:

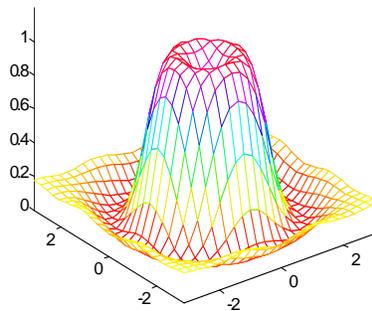
(a) Frequency response of a filter of order 13×13 designed using transformation method

(b) Impulse response(filter coefficients) of the filter designed by transformation method

It is clear that the performance of the filter designed using the transformation method is somewhat worse than that designed using frequency sampling method. A factor which

should be taken into consideration is the order of the filter. The order of the filter in *Example 4-2* is only  $13 \times 13$  whereas in *Example 4-1* it is  $33 \times 33$ . The reason for such large difference lies with the coefficients of the resulting filter of transformation method. In *Figure 4.6(b)*, the coefficients of the filter as originally designed by the transformation method is shown as the form of impulse response of the filter. Clearly, it is a  $33 \times 33$  order filter, however the surrounding 10 rows and columns do not contribute to the filter response at all and therefore can be removed without affecting the response of the filter and the remaining coefficients constitute a  $13 \times 13$  order filter.

The transform function has a large bearing on the eventual filter transfer function. The filter shown in *Figure 4-7* is of the same order as the filter in *Figure 4.6(a)* and 1-D prototype function is identical. However, the transformation function is now a  $7 \times 7$  function<sup>2</sup>. It can be seen that the transition band is much more distinct whereas the stopband is not as well attenuated as the filter response in *Figure 4.6(a)*.



*Figure 4.7: 2-D filter designed using a different transform function*

The overall error of the filter whose frequency response is shown in *Figure 4.7* is found to be 12.84%. It is clear that the good performance in passband and the transition band is offset by the poor stopband attenuation. Finally, it is noted that the resulting transfer function is a non-separable FIR transfer function.

---

<sup>2</sup> [13] discusses methods of designing such transformation functions in great detail

## 4.5 Summary of Section 4

---

- The differences between FIR structure and IIR structure are discussed with reference to the implications to 2-D filters.
- Frequency sampling method is implemented and an example of filter design using frequency sampling method is shown.
- Windowing method in 2-D is briefly discussed.
- McClellan transformation is shown and implemented. Two filters are designed using two different transformation functions and the performances are analyzed.

## 5. MATRIX DECOMPOSITION METHODS

---

---

*As the second of the two streams of 2-D filter design methods, matrix decomposition methods are introduced. Matrix decomposition methods result in a set of separable 1-D magnitude responses which can be implemented using any 1-D filter design methods. Using either this approach or direct approach of Section 4, a transfer function of the desired 2-D filter can be obtained which can then be implemented by the photonic implementation methods discussed in Section 8.*

### 5.1 Single-Stage Singular Value Decomposition

---

In Section 3.3.2, the application of matrix decomposition to 2-D filter design is briefly introduced. Matrix decomposition is a mathematical procedure where a matrix is split into a sum of products of vectors as shown in

$$\mathbf{H} = \sum_{i=1}^n \lambda^{1/2} u_i v_i \quad \text{Eq. 5.1.1: Matrix decomposition}$$

where  $\lambda_i$  is  $i^{\text{th}}$  eigenvalue of  $\mathbf{H}$  and  $u_i$  and  $v_i$  are the decomposed vectors. An example of matrix decomposition is the well-known LU decomposition which splits a matrix into a lower triangular matrix and an upper triangular matrix. *Example 5-1* shows how LU decomposition can be used to express a matrix as sum of products of vectors

***Example 5-1: LU decomposition***

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} = \mathbf{P} \cdot \mathbf{L} \cdot \mathbf{U} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0.1429 & 1 & 0 \\ 0.5714 & 0.5000 & 1 \end{bmatrix} \begin{bmatrix} 7 & 8 & 9 \\ 0 & 0.8517 & 1.1743 \\ 0 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0.1429 \\ 0.5714 \\ 1 \end{bmatrix} \begin{bmatrix} 7 & 8 & 9 \end{bmatrix} + \begin{bmatrix} 1 \\ 0.5 \\ 0 \end{bmatrix} \begin{bmatrix} 0 & 0.8517 & 1.1743 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1.1432 & 1.4867 \\ 4 & 4.5712 & 5.2429 \\ 7 & 8 & 9 \end{bmatrix} + \begin{bmatrix} 0 & 0.8517 & 1.1743 \\ 0 & 0.4258 & 0.5871 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

$$\mathbf{A} = \sum_{i=1, x=1, 2, 3}^3 \mathbf{P} \cdot \mathbf{L}_{xi} \cdot \mathbf{U}_{ix}$$

*Eq. 5-2: LU decomposition expressed as sum of products*

A decomposition method particularly suited to 2-D filter design needs is the singular value decomposition (SVD). The SVD reduces a 2-D matrix into two matrices  $\mathbf{U}$  and  $\mathbf{V}$ , and a diagonal matrix  $\mathbf{S}$  of singular values of the original matrix. Singular values are related to the eigenvalues of the matrix and the result has the form

$$\mathbf{A} = \mathbf{U} \cdot \mathbf{S} \cdot \mathbf{B}$$

$$= \sum_{i=1, x=1, 2, \dots, N}^N \mathbf{U}_{xi} \mathbf{S}_{ii} \mathbf{B}_{xi}$$

*Eq. 5.1.2: Singular value decomposition*

The unique feature of SVD is that the matrix ‘power’ is distributed to the singular values of the matrix in decreasing order of the position of the singular value in the matrix  $\mathbf{S}$  starting from the top left corner. Consequently, to approximate the matrix  $\mathbf{A}$  by the

product of just one set of two vectors  $\mathbf{U}_{xi}$  and  $\mathbf{B}_{xi}$ , the best approximation will be made by the product of first set of vectors that result from SVD of the matrix. Mathematically, the property can be described as shown in Eq. 5-3. As an example, if we wanted to

approximate the matrix  $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$  by the product of a set of vectors, SVD would be

performed on the matrix resulting in  $\mathbf{U} = \begin{bmatrix} 0.4046 & 0.9145 \\ 0.9145 & -0.4046 \end{bmatrix}$ ,  $\mathbf{V} = \begin{bmatrix} 0.5760 & 0.8174 \\ -0.8174 & 0.5760 \end{bmatrix}$ ,

and  $\mathbf{S} = \begin{bmatrix} 5.4650 & 0 \\ 0 & 0.3660 \end{bmatrix}$ .  $\mathbf{U}_{x1} = [0.4046 \ 0.9145]$ ,  $\mathbf{S}_{11} = 5.4650$ , and  $\mathbf{V}_{x1} =$

$[0.5760 \ 0.8174]$  form the first set of vectors. The resulting approximation would then

be  $\mathbf{U}_{x1}\mathbf{S}_{11}\mathbf{V}_{x1}^T = \begin{bmatrix} 1.2736 & 1.8072 \\ 2.8790 & 4.0853 \end{bmatrix}$ . To obtain a better approximation of  $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ , the

product of the second set of vectors could be added.

$$\left\| \mathbf{A} - \sum_{i=1, x=1, 2 \dots N}^N \mathbf{U}_{xi} \mathbf{S}_{ii} \mathbf{B}_{xi}^T \right\| = \min_{\mathbf{U}_{xi}, \mathbf{S}_{ii}, \mathbf{B}_{xi}} \left\| \mathbf{A} - \sum_{i=1, x=1, 2 \dots N}^N \hat{\mathbf{U}}_{xi} \hat{\mathbf{S}}_{ii} \hat{\mathbf{B}}_{xi}^T \right\|$$

where

$\hat{\mathbf{U}}_{xi}$  and  $\hat{\mathbf{B}}_{xi}$  are subsets of  $\mathbf{U}_{xi}$  and  $\mathbf{B}_{xi}$ , and  $\hat{\mathbf{S}}_{ii}$  is the corresponding singular value

*Eq. 5.1.3: Minimal square error approximation property of SVD [14]*

In [17], the SVD is used to decompose a 2-D magnitude specification matrix into two 1-D magnitude specification. The result is a design procedure in which a 2-D filter design becomes a set of 1-D filter design. The matrix decomposition methods have several advantages over the direct methods of Section 4.

1. The resulting 1-D magnitude specifications can be met by any of the standard algorithms for 1-D filter design such as least-squares method or Parks-McClellan algorithms available in many computer simulation packages.
2. As long as 1-D filter sections are stable, the overall 2-D filter is also stable. The stability of final 2-D design can therefore be guaranteed easily without

the need for heavy mathematical analysis or computationally expensive algorithms involved with 2-D filter designs.

3. The filter designer is given the flexibility to decide how many sets of 1-D filter sections are included in the system.
4. The resulting 2-D filter is parallel in structure and therefore does not introduce unnecessary processing delays.

All 2-D filters designed using matrix decomposition can be described by Eq. 5.1.4 where  $F_i(z_1)$  and  $G_i(z_2)$  are 1-D filter sections and  $K$  is the number of singular values included in the system.

$$H(z_1, z_2) = \sum_{i=1}^K F_i(z_1) \cdot G_i(z_2) \quad \text{Eq. 5.1.4: 2-D filter transfer function designed using matrix decomposition method}$$

A simple example of 2-D filter design using only one filter section as in [17] is given in Example 5-2. The example chosen is deliberately simple to show the fundamental concepts involved in 2-D filter design using matrix decomposition methods.

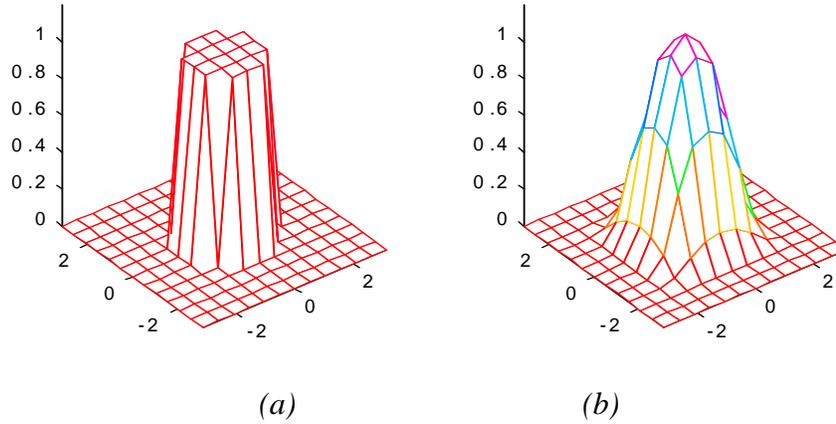
**Example 5-2: A 2-D filter design using SVD with single parallel section**

**Design Aim:** A low-pass filter with normalized cut-off frequency of 0.5 in both dimensions.

**Method:** Single-stage singular value decomposition

**Program used:** SVDFIR2-D.M

**Result:** With single stage, the error of frequency response of the designed filter is 6.663%. Although this is quite low, it may not be acceptable in some cases for which an extension of single-stage singular value decomposition may be employed as shown in Section 5.2.



*Figure 5.1:*  
 (a) *Magnitude specification of low-pass filter*  
 (b) *Magnitude response of 15×15 2-D filter designed using single stage singular value decomposition*

The 1-D filters designed are FIR filters and the actual filter coefficients are given in a table format as shown below. It is noted that the first and the second filter sections are identical since the frequency specification is symmetric about the origin. Consequently for symmetric filters, only one 1-D filter needs to be designed to complete the design for a 2-D filter implying significant simplification in the filter design procedure.

Coefficient order	$b_1$	$b_2$
<b>0</b>	-0.0007	-0.0007
<b>1</b>	0.0010	0.0010
<b>2</b>	0.0025	0.0025
<b>3</b>	-0.0090	-0.0090
<b>4</b>	-0.0273	-0.0273
<b>5</b>	0.0197	0.0197
<b>6</b>	0.1837	0.1837
<b>7</b>	0.3553	0.3553
<b>8</b>	0.3553	0.3553
<b>9</b>	0.1837	0.1837
<b>10</b>	0.0197	0.0197
<b>11</b>	-0.0273	-0.0273
<b>12</b>	-0.0090	-0.0090
<b>13</b>	0.0025	0.0025
<b>14</b>	0.0010	0.0010
<b>15</b>	-0.0007	-0.0007

*Table 5-1: Coefficients of the FIR filter designed using single-stage singular value decomposition*

The filter designed in *Example 5-2*, when implemented takes on the structure shown in *Figure 5-2*. In case of a multiple-stage implementation, several of the structure shown below would be connected in parallel to form the 2-D filter (see *Figure 5.5*).

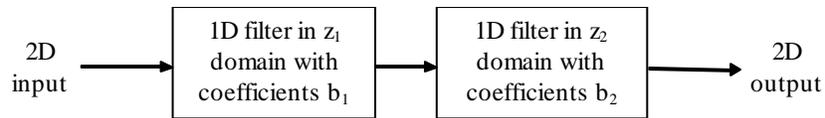


Figure 5.2: Separable implementation of 2-D filter using single-stage singular value decomposition

## 5.2 Multiple-Stage Singular Value Decomposition

Multiple-stage SVD takes the leap forward from single-stage SVD method and includes stages that belong to second largest singular values and smaller. Depending on the relative magnitude of singular values, the inclusion of extra stages can result in a sizable reduction in error, or sometimes it has no effect at all. *Example 5-3* shows a case where inclusion of multiple stages results in more than 30% reduction in the error of the single stage implementation.

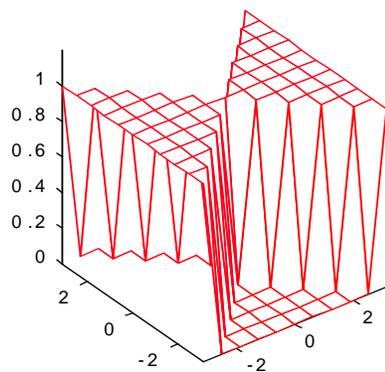
### ***Example 5-3: A 2-D filter design using SVD with multiple parallel sections.***

Design Aim: A  $90^\circ$  fan filter of order of  $32 \times 32$

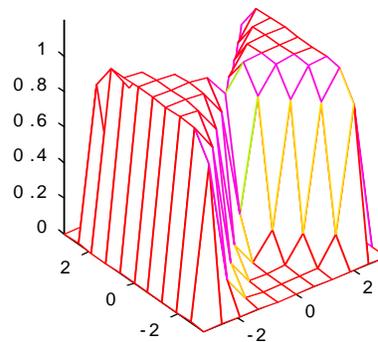
Method: Multiple stage singular value decomposition with 1-D FIR filters

Program used: SVDFIR2-D.m

Result: The error of frequency response of the filter is given as  $18.65\%$ <sup>3</sup>.



(a)



(b)

<sup>3</sup> Although this value is quite large compared to the single digit figure we have been obtaining so far, it should be kept in mind that the error largely depends on the specification. Therefore it is only meaningful to compare error between different implementations of the same magnitude specification.

Figure 5.3

(a) Magnitude specification of  $90^\circ$  fan filter

(b) Magnitude response of  $32 \times 32$  2-D filter designed using multiple-stage singular value decomposition.

The filter is obtained after six parallel stages. Figure 5.4 shows the error and magnitudes of the singular values against the number of included parallel stages. Figure 5.4 appears to show that there is roughly a linear relationship between the error curve and the singular values curve. The relationship is actually more subtle than this. A little thought will reveal that greater the gradient of singular value curve is, flatter the error curve will be. This is because if there is a large difference between two singular values, adding the stage which belongs to the smaller singular value will have little effect on the overall performance. As a rule of thumb, if the singular value of a parallel stage is less than one-tenth of the first singular value, then it is probably not worth including.

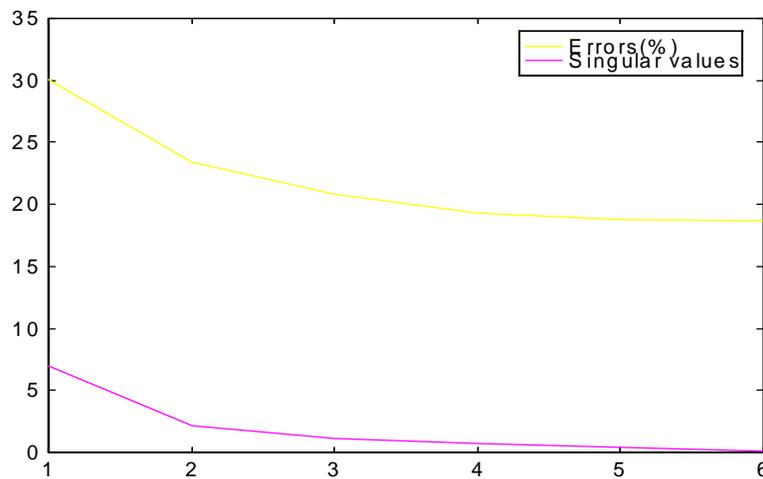


Figure 5.4: Errors and magnitudes of singular values

As it can be observed in Example 5-3, quite complex magnitude specifications can be achieved by filters designed using singular value decomposition method. The filter of Example 5-3 required 12 1-D filter designs overall. With the current computer technology, calculations for around ten 32nd order 1-D filters can be done virtually in real time and therefore the SVD method is practical even for adaptive filtering. The resulting filter structure is shown in Figure 5.5.

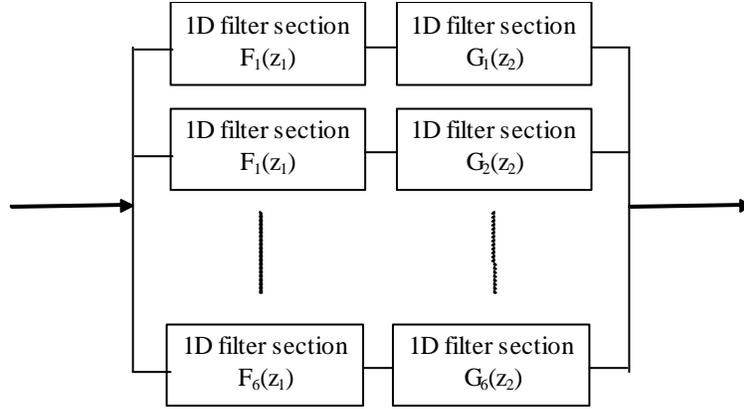


Figure 5.5: Structure of 2-D 90° fan filter

### 5.3 Iterative Singular Value Decomposition

There are many variations on the theme of matrix decomposition, in particular the SVD. The iterative singular value decomposition (ISVD)[18] is devised in order to avoid ‘negative’ magnitude definitions that arises from the plain SVD procedure of the previous section. By keeping the 1-D magnitude positive, the paper claims that the 1-D filter design procedure becomes less intricate. The ISVD is described in *Box 5-1*.

#### ***Box 5-1: Iterative singular value decomposition [19]***

1. Let the 2-D magnitude specification be  $\mathbf{A}$ . Let  $\mathbf{A}^+_1 = \mathbf{A}$  and  $\mathbf{A}^-_1 = \mathbf{0}$ .
2. Perform singular value decomposition on  $\mathbf{A}^+$ .  $\lambda_{1i}$  are the singular values of  $\mathbf{A}^+$ . By the definition of SVD given in Eq. 5.1.3,  $\lambda_{11}$  is larger than any other  $\lambda_s$ .

$$\begin{aligned} \mathbf{A}^+ &= \sum_{i=1}^{\eta_1} \lambda_{1i} \mathbf{u}_{1i} \mathbf{v}_{1i}^t \\ &= \sum_{i=1}^{\eta_1} \mathbf{u}_{1i} \lambda_{1i}^{1/2} \cdot \lambda_{1i}^{1/2} \mathbf{v}_{1i}^t \end{aligned}$$

3. Because of Perron’s result on non-negative matrices[18], the vectors  $\mathbf{u}_{11}$  and  $\mathbf{v}_{11}$  are also non-negative. It is then possible to estimate  $\mathbf{A}^+$  by  $\mathbf{u}_{11} \lambda_{11}^{1/2} \cdot \lambda_{11}^{1/2} \mathbf{v}_{11}^t$ . Assigning the first of the pair as  $\mathbf{F}^+_1$  and  $\mathbf{G}^+_1$  gives the first non-negative 1-D magnitude specifications.  $\mathbf{F}_1$  and  $\mathbf{G}_1$  are assigned  $\mathbf{F}^+_1$  and  $\mathbf{G}^+_1$

$$S_1 = 1$$

$$\mathbf{F}_1 = \mathbf{F}^+_1$$

$$\mathbf{G}_1 = \mathbf{G}^+_1$$

4.  $\mathbf{A}_2$  can now be calculated using Eq. 5-5. This matrix can now be separated into  $\mathbf{A}^+_2$  and  $\mathbf{A}^-_2$  sum of which make up the error matrix  $\mathbf{A}_2$ .

$$\begin{aligned}\mathbf{A}_2 &= \mathbf{A} - S_1 \mathbf{F}_1 \mathbf{G}_1 \\ &= \mathbf{A}_2^+ + \mathbf{A}_2^-\end{aligned}$$

Eq. 5.3.1: Error matrix

where

$$\begin{aligned}A_2^+(m,n) &= A_2(m,n) \quad \text{if } A_2(m,n) \geq 0 \\ &= 0 \quad \text{if } A_2(m,n) < 0 \\ A_2^-(m,n) &= A_2(m,n) \quad \text{if } A_2(m,n) < 0 \\ &= 0 \quad \text{if } A_2(m,n) \geq 0\end{aligned}$$

5. Singular value decomposition is performed on both matrices resulting in two sets of vectors  $S_2, \mathbf{F}_2^+, \mathbf{G}_2^+$  and  $S_2, \mathbf{F}_2^-, \mathbf{G}_2^-$ .  $S_2$  is 1 for the vectors resulting from decomposition of  $\mathbf{A}^+$ , and -1 for the vectors from  $\mathbf{A}^-$ .
6. Euclidean norms are calculated for resulting error matrix defined in Eq. 5-6. The same operation is performed with  $S_2, \mathbf{F}_2^-, \mathbf{G}_2^-$  in place of  $S_2, \mathbf{F}_2^+, \mathbf{G}_2^+$  in Eq. 5-6 with  $E_2^-$  as the result.

$$\mathbf{E}_2^+ = \mathbf{A} - \sum_{i=1}^2 S_i \mathbf{F}_i^+ \mathbf{G}_i^+$$

$$\|E_2^+\|_2 = \left\{ \sum_{m=0}^M \sum_{n=0}^N [E_2(m,n)]^2 \right\}^{1/2}$$

Eq. 5.3.2: Euclidean norm

7.  $E_2^+$  and  $E_2^-$  are compared. Since smaller error means closer approximation to the original matrix, the set of vectors that results in the lower Euclidean norm is chosen as  $\mathbf{F}_2$  and  $\mathbf{G}_2$ .
8.  $\mathbf{A}_3$  is assigned the error matrix  $\mathbf{A}_2 - \mathbf{A}^+$  or  $\mathbf{A}_2 - \mathbf{A}^-$  depending on whether  $E_2^+$  is greater or smaller than  $E_2^-$ . Steps 4,5,6 and 7 are then repeated with appropriate substitution.

The procedure is repeated until a satisfactory approximation of the original matrix is obtained.

Compared to plain singular value decomposition algorithm, ISVD algorithm converges more slowly because adding an extra stage does less to compensate for the error than plain SVD algorithm since only a part of the error is compensated. An example of a filter designed using the iterative singular-value decomposition is given in *Example 5-4*.

***Example 5-4: 2-D Filter design example using iterative singular value decomposition***

Design aim: A bandpass filter of order of 32×32 with normalized passband between 0.3 and 0.6.

Method: Iterative singular value decomposition algorithm

Program used: ISVDFIR2-D.m

Result:

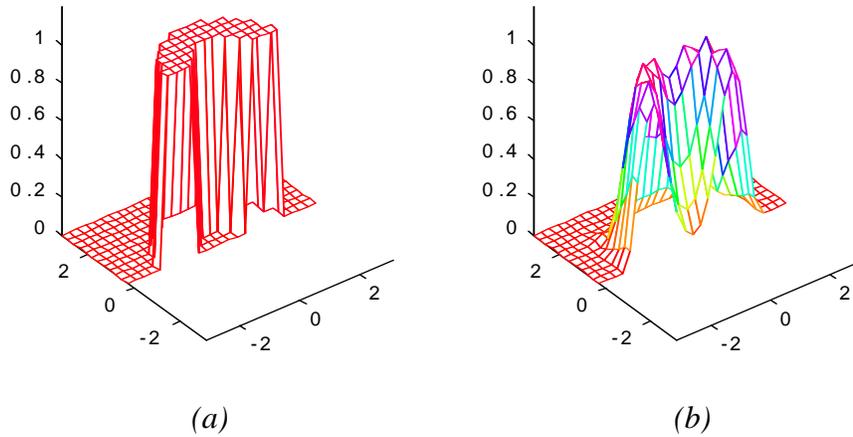


Figure 56: Iterative singular value decomposition (a) Ideal magnitude response (b) Actual filter magnitude response

The filter error is reasonably low at 9.88% after seven approximation stages. Overall, the filter requires seven 1-D FIR filter design stages as the magnitude specification is symmetric about the two axis. Because of the complexity of the magnitude specification, the designed filter does not perform as well as one might expect. This can be corrected to some extent using better 1-D filter design procedures such as the Parks-McClellan algorithm.

## 5.4 Optimal Decomposition

Optimal decomposition is an improvement on ISVD. The algorithm given in *Box 5-2* is based on optimization of the 1-D magnitude vectors so that the Euclidean error is minimized. The error is found to be 9.68%, which is only slightly better than that of ISVD.

### ***Box 5-2: Optimal decomposition[19]***

In *Eq. 5-6*, the definition of the Euclidean norm is defined. In the optimal decomposition, the objective is to minimize the value of this error estimate to provide the best set of vectors that will make up the original specification matrix.

Continuing with the constraint that magnitude vectors must all be positive, we then perform exponential mapping to  $\mathbf{F}_i$  and  $\mathbf{G}_i$  (see *Eq. 5-7*).

$$\mathbf{F}_i = \begin{bmatrix} e^{x_{i0}} & e^{x_{i1}} & \dots & e^{x_{iM}} \end{bmatrix}$$

$$\mathbf{G}_i = \begin{bmatrix} e^{y_{i0}} & e^{y_{i1}} & \dots & e^{y_{iN}} \end{bmatrix}$$

#### *Eq. 5.4.1 : Magnitude exponential mapping*

The purpose of exponential transformation is so that the optimizing variables  $x_{ij}$  and  $y_{ij}$  are not constrained to be positive. However the condition of positive magnitude is

retained as all values of  $\mathbf{F}_i$  and  $\mathbf{G}_i$  will be positive no matter what the values of  $x_{ij}$  and  $y_{ij}$  are. Non-linear optimizing technique must be applied since this problem is non-linear. Numerous algorithms exist for non-linear optimization of several variables and any technique can be used to obtain the answer.

Choosing bad starting points for optimization routines results in local minima or bad convergence points and it is recommended that iterative singular value decomposition algorithm be used to provide the initial points for optimization.

The algorithm can be applied to the same design problem as in *Example 5-4* and the result is given in *Example 5-5*.

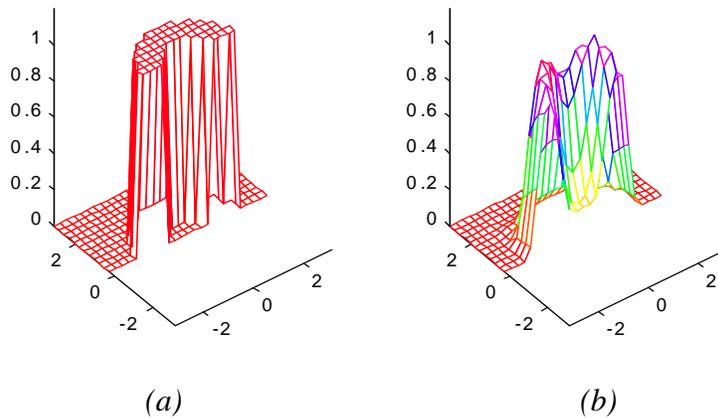
***Example 5-5: 2-D filter design example using optimal decomposition***

**Design aim:** A 2-D bandpass filter with normalized passband frequencies of 0.33 and 0.66 in both dimensions (same as *Example 5-4*).

**Method:** Optimal decomposition [19]

**Program used:** ODFIR2-D.m

**Result:** The error is 9.65% compared to 9.88% for ISVD algorithm after seven stages. The number of filter designs required is seven (same as ISVD), however each filter stage requires a great deal more computational effort than the ISVD method as it requires non-linear optimization to be performed on quite a large number of variables.



*Figure 5-7: Optimal decomposition (a) Ideal magnitude response (b) Actual filter magnitude response*

Due to the computational constraints, full optimization is not performed. Even then the optimization routine took a very long time to perform and the reason is attributed to the number of variables to be optimized being so large (20 to 30 variables depending on the order of the filter transfer function).

## 5.5 Other 2-D Filter Design Methods Based on Matrix Decomposition

There are many other 2-D filter design methods that are based on the idea of matrix decomposition. So far, all the methods discussed decompose the 2-D magnitude specification into a set of two 1-D magnitude specifications so that the 2-D filter design procedure is essentially reduced to that of 1-D. It is shown that using this approach, the design problem is reduced significantly, but it is also shown that since the approach produces only an approximation to the 2-D transfer function the methods based on magnitude decomposition does not perform as well as direct methods.

One notable 2-D filter design method uses matrix decomposition but is not based on magnitude decomposition is by Shaw and Mistra [20]. In this approach, it is assumed that the 2-D transfer function is already obtained using some 2-D filter design methods (such as direct methods of *Section4*).

A 2-D transfer function can be represented by matrices as shown in *Eq. 5.5.1*.

$$H(z_1, z_2) = \frac{h_b(z_1, z_2)}{h_a(z_1, z_2)}$$

$$= \frac{\mathbf{z}_{b1}^T \begin{bmatrix} h_b(n_{b1}-1, m_{b1}-1) & h_b(n_{b1}-1, m_{b1}-2) & \cdots & h_b(n_{b1}-1, 0) \\ \vdots & \vdots & \ddots & \vdots \\ h_b(1, m_{b1}-1) & h_b(1, m_{b1}-2) & \cdots & h_b(1, 0) \\ h_b(0, m_{b1}-1) & h_b(0, m_{b1}-2) & \cdots & h_b(0, 0) \end{bmatrix} \mathbf{z}_{b2}}{\mathbf{z}_{a1}^T \begin{bmatrix} h_a(n_{a1}-1, m_{a1}-1) & h_a(n_{a1}-1, m_{a1}-2) & \cdots & h_a(n_{a1}-1, 0) \\ \vdots & \vdots & \ddots & \vdots \\ h_a(1, m_{a1}-1) & h_a(1, m_{a1}-2) & \cdots & h_a(1, 0) \\ h_a(0, m_{a1}-1) & h_a(0, m_{a1}-2) & \cdots & h_a(0, 0) \end{bmatrix} \mathbf{z}_{a2}}$$

where

$$\mathbf{z}_{b1} \triangleq [z_1^{bm_1-1} \quad z_1^{bm_1-2} \quad \cdots \quad z_1 \quad 1]^T$$

$$\mathbf{z}_{b2} \triangleq [z_2^{bm_1-1} \quad z_2^{bm_1-2} \quad \cdots \quad z_2 \quad 1]^T$$

$$\mathbf{z}_{a1} \triangleq [z_1^{am_1-1} \quad z_1^{am_1-2} \quad \cdots \quad z_1 \quad 1]^T$$

$$\mathbf{z}_{a2} \triangleq [z_2^{am_1-1} \quad z_2^{am_1-2} \quad \cdots \quad z_2 \quad 1]^T$$

*Eq. 5.5.1: 2-D transfer function in matrix form*

In Shaw's method, the decomposition is performed on 2-D transfer function matrices. The method can result in efficient filters in terms of the required elements, however it results in 1-D filter sections with different orders and thus does not offer the modularity of the other decomposition methods [20]. Other methods exist for yet more efficient filter design and a 2-D filter order reduction method is described in Section 6.

## **5.6 Summary of Section 5**

---

- Matrix decomposition as a method of reducing a 2-D filter design procedure into a set of 1-D filter design procedures is described.
- Singular value decomposition method is introduced with an example.
- The iterative singular value decomposition is introduced with an example.
- Optimal decomposition is introduced with an example.

## **6. 2-D FILTER ORDER REDUCTION USING BALANCED APPROXIMATION THEORY**

---

---

*Keeping the filter order to the minimum is important for several reasons and more so for photonic implementations as coupling losses of higher order filters may render the actual implementation impossible. To achieve lower order filters with good performance, the balanced approximation method used in control systems theory is applied to the order reduction of 2-D digital filters.*

### **6.1 Motivation for Lower Order Photonic Filters**

---

For an adequate filter performance, that is satisfying the rolloff factor and the passband ripple, the order of the filter must be appropriately chosen. In general, increasing the order of the filter can significantly reduce the error. However, higher filter order directly

translates to extra filter components, noise, and higher attenuation which are unacceptable in many cases including fiber-optic implementations. It is therefore critical that the order of the filter remains low without sacrificing overall performance measures such as error response.

The motivation for keeping filter order low is greater for fiber-optic systems than in other filter implementations for the reason that higher order filters cause large coupling loss which must be compensated by a pre-amplifier which in turn introduces noise when the amplification factor is large. It is generally accepted that filters with order greater than 16 start becoming difficult to realize in practice with the current technology. However, as we have seen in *Sections 5 and 6*, 2-D filters with orders of around  $30 \times 30$  are quite common.

Balanced approximation, derived from control theory, is a model reduction technique for 1-D systems. As 2-D filters usually have high orders, application of the filter order reduction method to 2-D filters may prove to be very rewarding especially for fiber-optic filters which must have low orders for feasibility.

## **6.2 Description of 2-D System in State-Space Format**

---

As balanced approximation is originally developed for model reduction of dynamic systems, it uses the state-space model of digital systems. The implication is that 2-D systems, which we have been representing using transfer functions must now be represented in state-space format.

The representation of 2-D systems in a state-space format has been a topic for research for a number of years and several models have been proposed [22,23]. It is noted in [12] that the model in [22](see *Box 3-2*) is most general and the model proposed in [23] can actually be embedded into the model in [22].

Although converting from 2-D transfer function description to 2-D state-space description involves only plug-in formulae, converting from 2-D state-space description to 2-D transfer function is much more involved and a novel algorithm is described in

Section 3.2.2.2. Using the two algorithms, the balanced approximation method can be applied to 2-D systems.

### 6.3 Balanced Approximation Method

---

Using a known 2-D filter transfer function, the balanced approximation method (BAM) finds the *balancing transformation* matrix  $T$  which ‘balances’ the system. The order reduction is subsequently performed by removing states that do not contribute substantially to the system behavior.

The first task is to find the generalized reachability and observability gramians defined as

$$K = \frac{1}{(2\pi j)^2} \oint_{|z_1|=1} \oint_{|z_2|=1} f(z_1, z_2) f^*(z_1, z_2) \frac{dz_1}{z_1} \frac{dz_2}{z_2}$$

$$W = \frac{1}{(2\pi j)^2} \oint_{|z_1|=1} \oint_{|z_2|=1} g^*(z_1, z_2) g(z_1, z_2) \frac{dz_1}{z_1} \frac{dz_2}{z_2}$$

where

$$f(z_1, z_2) = [\mathbf{I}(z_1, z_2) - \mathbf{A}]^{-1} \mathbf{b}$$

$$g(z_1, z_2) = \mathbf{c}[\mathbf{I}(z_1, z_2) - \mathbf{A}]^{-1}$$

Eq. 6.3.1: Reachability and observability gramians

Fortunately, the double integrations do not have to be solved directly and can be partially solved (as distinct from partial integration) using the Lyapunov approach [21]. If  $K_{11}$  denotes the upper left upper block of  $K$  and  $K_{22}$  denotes the lower right block of  $K$ , then  $K_{11}$  and  $K_{22}$  can be obtained using Eq. 6.3.1. The same notations apply to the observability gramian,  $W$ .

$$K_{11} = \sum_{i=0}^{N_1-1} \mathbf{A}_1^i \mathbf{P} (\mathbf{A}_1^T)^i \quad W_{22} = \sum_{i=0}^{N_2-1} \mathbf{A}_4^i \mathbf{Q} (\mathbf{A}_4^T)^i$$

where where

$$\mathbf{P} = \mathbf{H}_b \mathbf{H}_b^T \quad \mathbf{Q} = \mathbf{H}_c^T \mathbf{H}_c$$

$$\mathbf{H}_b = \begin{bmatrix} h_{10} & \cdots & h_{1N_2} \\ \vdots & \ddots & \vdots \\ h_{N_10} & \cdots & h_{N_1N_2} \end{bmatrix} \quad \mathbf{H}_c = \begin{bmatrix} h_{01} & \cdots & h_{0N_2} \\ \vdots & \ddots & \vdots \\ h_{N_11} & \cdots & h_{N_1N_2} \end{bmatrix}$$

$$W_{11} = \mathbf{I}_{N_1} \quad K_{22} = \mathbf{I}_{N_2}$$

*Eq. 6.3.2: Calculations of reachability and observability gramians*

A system is said to be balanced if its gramians satisfy the following condition where  $\sigma_{ij}$  are the Hankel singular value of the system.

$$K_{11} = W_{11} = \begin{bmatrix} \sigma_{11} & 0 & \cdots & 0 \\ 0 & \sigma_{12} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \sigma_{1N_1} \end{bmatrix}$$

$$K_{22} = W_{22} = \begin{bmatrix} \sigma_{21} & 0 & \cdots & 0 \\ 0 & \sigma_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \sigma_{2N_2} \end{bmatrix}$$

*Eq. 6.3.3: The condition for a balanced system*

To balance a system, the similarity transform  $T$  that will achieve the above condition must be found. Applying the balancing similarity transform  $T$  to the subsections of gramians by *Eq. 6.3.4* will result in the condition in *Eq. 6.3.3* being satisfied.

$$\hat{K} = T^{-1} K T^{-T}$$

$$\hat{W} = T^T W T$$

*Eq. 6.3.4: Similarity transformations of gramians*

The balancing transformation  $T$  can be found by applying the algorithm given in [24].

**Box 6-1: Determination of the balancing transformation  $T$  [24].**

1. Cholesky factorization of  $K_{11}$ : The resulting lower triangular matrix is assigned  $L_c$ .
2. Formation of  $L_c^T W_{11} L_c$
3. Symmetric eigenvalue/eigenvector problem,  $U_{11}^T (L_{11c}^T W_{11} L_{11c}) U_{11} = \Lambda_{11}^T$ .
4. Formation of  $T_{11}$ :  $T_{11} = L_c U_1 \Lambda_{11}^{-1/2}$

The same procedure with appropriate subscript substitutions can be used to find  $T_{22}$ . Once both  $T_{11}$  and  $T_{22}$  are found, the overall transformation matrix  $T$  can be found by performing an operation denoted by  $\oplus$  in [21].

$$\begin{aligned} \mathbf{T} &= \mathbf{T}_{11} \oplus \mathbf{T}_{22} \\ &= \begin{bmatrix} \mathbf{T}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_{22} \end{bmatrix} \end{aligned}$$

*Eq. 6.3.5: Overall balancing transformation*

Using the balancing matrix  $T$ , a balanced realization of the system can be found by similarity transformation of state-space matrices as shown in Eq. 6-6.

$$\begin{aligned} \hat{\mathbf{A}} &= \mathbf{T}^{-1} \mathbf{A} \mathbf{T} \\ \hat{\mathbf{b}} &= \mathbf{T}^{-1} \mathbf{b} \\ \hat{\mathbf{c}} &= \mathbf{c} \mathbf{T} \\ \hat{d} &= d \end{aligned}$$

*Eq. 6-6: Balanced realization of 2-D system using the balancing transformation matrix  $T$ .*

By observing how many significant Hankel singular values exist, one can make the decision on how many states should be preserved thereby determining the order  $r_1$  and  $r_2$ . The state-space matrices can then be partitioned using the following scheme.

$$\begin{aligned} \hat{\mathbf{A}} &= \begin{bmatrix} \hat{\mathbf{A}}_1 & \hat{\mathbf{A}}_2 \\ \hat{\mathbf{A}}_3 & \hat{\mathbf{A}}_4 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{1r} & * & \mathbf{A}_{2r} & * \\ * & * & * & * \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_{4r} & * \\ \mathbf{0} & \mathbf{0} & * & * \end{bmatrix} \\ \hat{\mathbf{b}} &= \begin{bmatrix} \hat{\mathbf{b}}_1 \\ \hat{\mathbf{b}}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{b}_{1r} \\ * \\ \mathbf{b}_{2r} \\ * \end{bmatrix} \\ \hat{\mathbf{c}} &= [\hat{\mathbf{c}}_1 \mid \hat{\mathbf{c}}_2] = [\mathbf{c}_{1r} \quad * \mid \mathbf{c}_{2r} \quad *] \end{aligned}$$

*Eq. 6.3.6: Matrix partitioning of balanced system*

$\mathbf{A}_{1r}$  is a  $[r_1 \times r_1]$  matrix if  $r_1$  is greater than  $r_2$  and a  $r_1 \times r_2$  matrix if  $r_2$  is greater than  $r_1$ . On the other hand  $\mathbf{A}_{2r}$  is a  $r_2 \times r_1$  matrix if  $r_1$  is greater than  $r_2$  and a  $[r_2 \times r_2]$  matrix if  $r_2$  is greater than  $r_1$ . The dimensions of  $\mathbf{A}_{3r}$  and  $\mathbf{A}_{4r}$  are the same of that of  $\mathbf{A}_{1r}$  and  $\mathbf{A}_{2r}$ , respectively. Finally, the reduced system is obtained by forming new system matrices by

$$\begin{aligned} \mathbf{A}_r &= \left[ \begin{array}{c|c} \mathbf{A}_{1r} & \mathbf{A}_{2r} \\ \hline \mathbf{A}_{3r} & \mathbf{A}_{4r} \end{array} \right] \\ \mathbf{b}_r &= \left[ \begin{array}{c} \mathbf{b}_{1r} \\ \mathbf{b}_{2r} \end{array} \right] \\ \mathbf{c}_r &= [\mathbf{c}_{1r} \mid \mathbf{c}_{2r}] \end{aligned} \quad \text{Eq. 6.3.7: Reduced system matrices}$$

The resulting system which is described by the matrices  $\mathbf{A}_r$ ,  $\mathbf{b}_r$ , and  $\mathbf{c}_r$  is of order of  $[r_1 \times r_2]$ . From the new 2-D state-space description of the reduced system, one can obtain the 2-D system transfer function of lower order using the algorithm given in Section 3.2.2.2.

#### 6.4 Filter Order Reduction Using Balanced Approximation: An Example

---

In this section, a  $15 \times 15$  order bandpass filter is designed using optimal decomposition method, and the balanced approximation method is applied to reduce the filter order.

***Example 6-1: Application of balanced approximation method for 2-D filter order reduction***

**Design Aim:** 2-D bandpass filter with normalized passband frequency between 0.33 and 0.66 with lowest order acceptable.

**Method Used:** Optimal decomposition for filter design, and balanced approximation for order reduction.

**Programs Used:** ODFIR2-D.m, BA.m

**Results:** Using optimal decomposition method, a filter with specifications shown below is designed. The error is approximated at 11% after 6 stages of approximations.

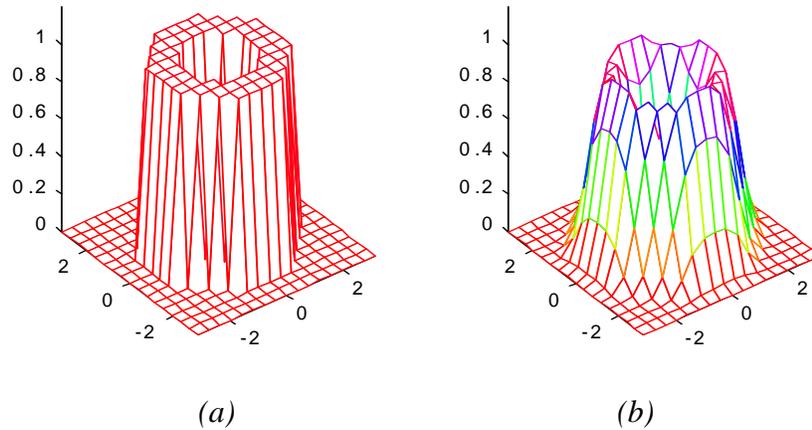


Figure 6.1: (a) Ideal magnitude response of the filter and (b) Actual filter response of the 16x16 order filter

Balanced approximation is then applied to the filter design. To apply the order reduction however, a new reduced order had to be chosen and the choice is made based on the Hankel singular values of the system plotted in Figure 6.2. Clearly, it seems reasonable to retain only up to 10th order in both dimensions since from 11th order onwards, the Hankel singular values become very small indeed.

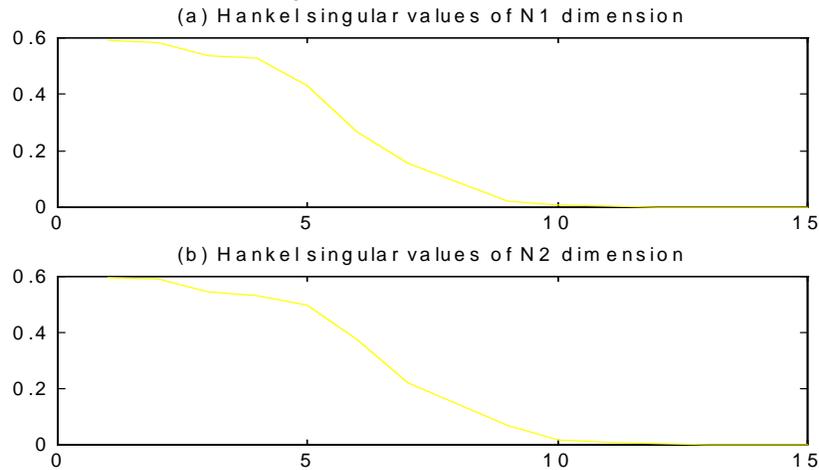
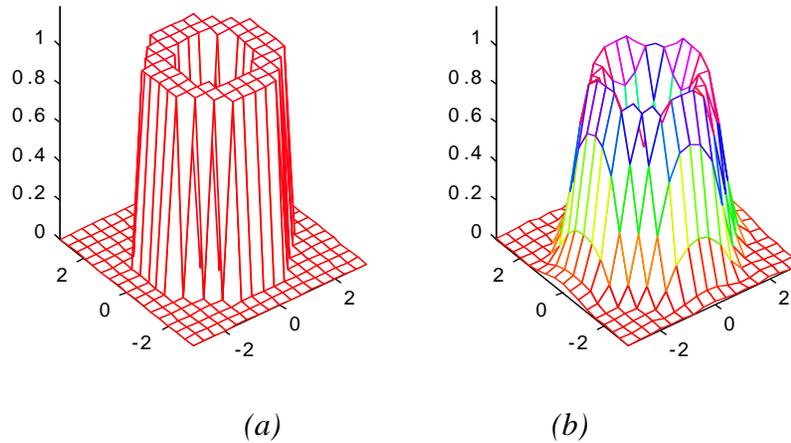


Figure 6.2: Hankel singular values of the filter

Choosing the new order of the filter as 10x10, the BAM is applied with the following excellent results.



(a) (b)  
 Figure 6.3: Reduced order(10×10) filter (a) Ideal characteristic of the filter (b) Actual characteristic of the filter

As the plots of magnitude characteristic shows, there is hardly any difference between the original design and the reduced order design. The error estimate of 11.46% compared to 11% of the original 15×15 order design confirms this point and shows that balanced approximation indeed produces filters of significantly lower order with very little sacrifice in performance.

The result of application of BAM to a 2-D filter transfer function can be summarized as follows:

1. Reduced order filter.
2. Usually IIR structure.
3. If the original filter has a separable denominator, then the reduced filter also has a separable denominator[21] allowing a separable implementation.
4. Little sacrifice in performance(magnitude error and phase linearity). The phase remains nearly linear for the resulting IIR structure as well which is a feature difficult to achieve with other 2-D IIR filter design methods. The proof of the linearity is given in [21].

All in all, the BAM provides an excellent method of reducing filter order to a realizable level without a large deterioration in performance and should therefore be given a consideration before implementation of 2-D filters.

## 6.5 Summary of Section 6

---

- The problem with ordinary 2-D filter design methods when implementing fiber optic filters is stated as the high order of the designed filters.
- A method of the filter order reduction called ‘Balanced Approximation’ is presented.
- An example of application of the balanced approximation to 2-D filter order reduction is given.

## 7. FIBER-OPTIC DELAY LINE FILTERS

---

---

*As described in Section 1.3, fiber optic delay line architecture is an alternative architecture to spatial and temporal architecture. The fiber optic delay line architecture used in this report to implement 2-D filters is described in further detail with a mathematical analysis.*

### 7.1 Coherent and Incoherent Operation of Photonic Filters

---

When the advances in laser technology first made guided wave photonic systems possible, most pioneering photonic systems used multi-mode propagation of light as the main mode of signal transmission. However with the advent of lasers with narrower linewidths, it has become possible to operate lightwave systems in single mode resulting in greater bandwidth-distance product. Single mode systems are becoming increasingly popular and the trend towards single mode systems is set to continue.

Aside from the mode of propagation, another factor that determines the characteristic of a lightwave system is whether system is operating in coherent or in incoherent modes. The differences between the two operations can be summarized as follows: in a coherent

system, the light source can be regarded as operating in a single wavelength (although in reality, no matter how small the linewidth is, the emitted lightwave is certain to contain more than one wavelength component). The use of coherent light as the signal carrier simply means that the phase as well as the amplitude of the lightwaves must be regarded as a part of the information being carried by the lightwave. In incoherent systems, the information is carried only by the intensity of the lightwave. One may therefore consider incoherent systems as the amplitude modulated system with intensity modulation instead of amplitude modulation. It is obvious that negative range cannot be expressed by intensity-based incoherent systems unless one biases the light intensity to a predefined level. The receiver can then detect negative range by comparing the received intensity value to the predefined level.

The differences between coherent systems and incoherent systems are shown in *Table 7-1*. For signal processing purposes, incoherent operation implies that the modulating frequency of the source must be much lower than the photonic frequency implying that the full bandwidth of the laser cannot be used. Coherent operation on the other hand allows utilization of the full bandwidth of the system resulting in greater processing speed. However, because coherent systems tend to be more vulnerable to environmental effects such as phase jitter, some shielding must be used to reduce the adverse effects to a negligible level [3]. Another important advantage coherent systems have over incoherent systems is the flexibility in system design. Incoherent systems fall into the category of so called *positive systems* and have restrictions on quantities such as number and positions of system poles and zeros [9]. In spite of such constraints, most of the research works reported so far on fiber-optic delay line signal processing have been using incoherent systems [3-7]. The reasons for avoiding coherent systems have been that 'coherent systems are more difficult to implement in practice and are usually more complicated than incoherent systems because of the stringent requirements on the stability of the source and photonic delay paths' [4]. In future however, it is likely that lasers capable of coherent operation over longer distances as well as better techniques for controlling the delay paths will be available. Coherent systems thus may yet represent the possibility for full bandwidth all-photonic processing.

	COHERENT SYSTEM	INCOHERENT SYSTEM
<b>Information carrier</b>	amplitude, phase	intensity
<b>Bandwidth</b>	very wide	wide
<b>Required linewidth of the source</b>	very narrow	narrow
<b>Negative range</b>	amplitude and phase can combine to express a negative value	predetermined bias value is necessary

Table 7-1: Differences between coherent and incoherent lightwave system

## 7.2 Using Optical Fibers to Realize Delayed Line Filter

Three main components are required in most forms of discrete-time filters: delay, coefficient, and summer/splitter. To illustrate how the components are realized in photonic domain, discrete-time tab filter shown in *Figure 7-1* is used as an example. As the signal flow diagram for a discrete time tab filter is general, the photonic components used to realize a discrete-time tab-filter can be used in other filter structures.

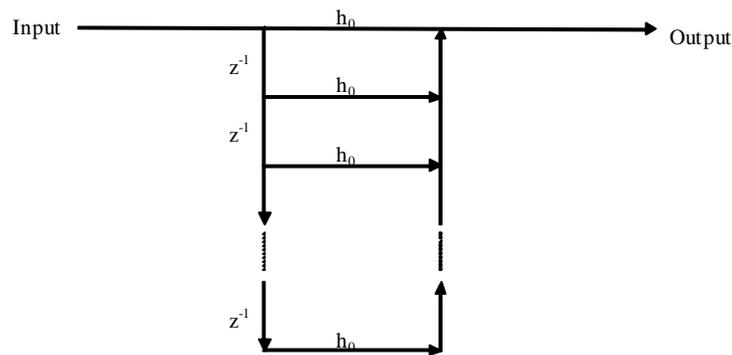


Figure 7.1: Signal flow diagram of discrete-time tab filter, the unit delay is the traveling time of lightwaves over a distance equivalent to the unit sampling time . The coefficients  $h_s$  are the transmittances over the specific path.

### 7.2.1 Photonic Realization of Delay

In fiber-optic delay line filters, photonic fibers are used as delay elements as signal propagation time can be controlled using the length of the fiber. The transfer function of optical fiber, ignoring the fiber signal dispersion and the fiber intensity loss, can be expressed mathematically by *Eq. 7-1*.

$$H(\omega) = e^{-j\beta L}$$

*Eq. 7.2.1: Transfer function of the fiber optic delay line*

where  $L$  is the length of the delay lines,  $\beta$  is the propagation constant of the guided fundamental mode. The propagation constant  $\beta$  is defined by  $\beta = \omega n_{eff}/c$  where  $n_{eff}$  is the effective refractive index of the guided mode in the fiber or optical planar channel waveguide,  $\omega$  is the operating optical frequency in radians, and  $c$  is the speed of light. The inverse of the time delay  $T$  is  $n_{eff}f/c$  and equals to the sampling frequency of the filter. Choosing a reference length of the optical delay as  $L_d$ , if  $L$  is a integer multiple of  $L_d$  the transfer function can be expressed as shown in *Eq. 7.2.2*.

$$H(\omega) = e^{-jd(L\frac{2\pi f n_{eff}}{c})} \dots\dots(a)$$

$$z^{-d} = e^{-jd(\omega T)} \dots\dots(b)$$

*Eq. 7.2.2 (a) Transfer function of fiber optic delay line with multiple delays (b) z-transform parameter*

Upon comparing *Eq. 7.2.2(a)* and *Eq. 7.2.2(b)*, it can be observed that the two equations are very similar and in fact, if  $\omega T$  in *Eq. 7.2.2(b)* is replaced by  $(2\pi f)(n_{eff}L_d/c)$ , then the two equations are identical. It is therefore clear that fiber can act as a delay whose length is controlled by  $n_{eff}$  and  $L$ .

Optical fiber has several properties which enable it to be an ideal delay line medium: flexibility which enables a relatively compact implementation of the system, the accuracy of time interval between tabs that can be produced, insensitivity to electromagnetic interference which is useful when used in electro-magnetic environments - quite often the case with signal processing equipment.

### **7.2.2 Photonic Realization of Tab Coefficients**

General form of a feed forward transfer function in z-domain can be expressed by

*Eq. 7.2.3*. In the previous section, it is shown that fiber can implement  $z^{-d}$  part of the transfer function.

$$H(z) = \sum_{d=0}^n h_d z^{-d}$$

Eq. 7.2.3: General form of a discrete-time transfer function

in z-domain(with no feedback)

There are several ways to realize the coefficients magnitude  $|h_d|$  such as optical amplifiers /attenuators[6](earlier methods of achieving filter coefficients included reflectors, radiation due to bending, and evanescent coupling by polishing the cladding down very close to the fiber core [8]). However the negative sign can be difficult to realize as it represents a negative intensity in an incoherent system! The inability to represent negative quantities effectively is a major limitation of incoherent systems. On coherent systems a multiplication by a negative coefficient represents a phase shift of  $180^\circ$ .

### 7.2.3 Photonic Realization of Summer/Splitter

In photonic domain, summing/splitting of signals can be performed by optical couplers (see Figure 7.2).

$$\begin{bmatrix} E_3 \\ E_4 \end{bmatrix} = \begin{bmatrix} \sqrt{1-k_1} & -j\sqrt{k_1} \\ -j\sqrt{k_2} & \sqrt{1-k_2} \end{bmatrix} \begin{bmatrix} E_1 \\ E_2 \end{bmatrix}$$

Eq. 7.2.4: 2x2 Optical coupler transfer matrix

Splitting of signal can be performed by using just one of the input terminals ( $E_3, E_4$ ) and both output terminals ( $E_1, E_2$ ) - see Figure 7.3(a). Summing can be achieved by using just one output signal port and both input ports - see Figure 7.3(b).

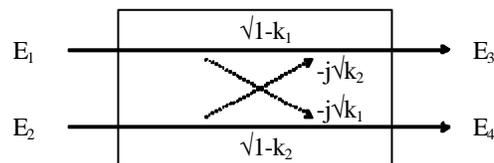


Figure 7.2: Schematic diagram of an optical coupler

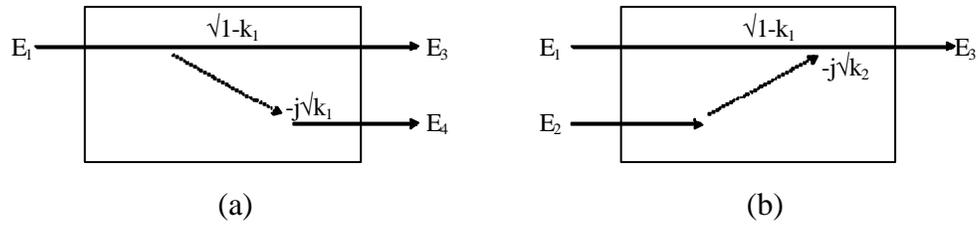


Figure 7.3:(a) Optical coupler as a splitter (b) Optical coupler as a summer

When using an optical coupler as a summer/splitter, there are two undesirable properties that must be taken into account. Firstly, as can be seen from the transfer matrix, when photonic signal goes through a coupler the signal amplitude (and therefore intensity) is attenuated by the coupling factor of the optical coupler which for a half intensity splitter is  $1/\sqrt{2}$ . A photonic filter is likely to have cascaded stages of optical couplers and the combined coupling coefficients cause quite substantial attenuation of the original input. Optical amplifiers are therefore usually necessary to compensate for the amplitude attenuation [6]. Second problem arising from the use of optical couplers as splitter/summer is the phase shift of  $-90^\circ$  associated with cross-coupling of photonic signals. The phase shift is not an issue for concern in an intensity-based system (incoherent system), however it can cause difficulty in coherent systems, especially when coupler is being used as a summer as the signals that are being added must be in the same phase at the output of the coupler. To illustrate this problem, consider adding two signals  $E_1$  and  $E_2$  that are in phase before they enter the coupler ( Eq. 7.2.5). At the output of the coupler, only one output is cross-coupled and therefore phase shifted whereas the other output retains the phase of the input signal. The added signal is therefore an inaccurate representation of the summing operation.

$$\begin{bmatrix} E_{out} \end{bmatrix} = \begin{bmatrix} \sqrt{1-k_1} & -j\sqrt{k_1} \end{bmatrix} \begin{bmatrix} E_1 \\ E_2 \end{bmatrix} \quad \text{Eq. 7.2.5: Optical coupler as a summer}$$

$$E_{out} = \sqrt{1-k_1} E_1 - j\sqrt{k_1} E_2$$

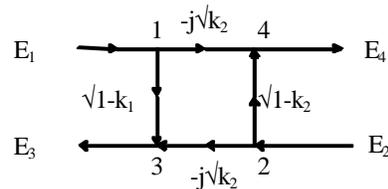
However, the phase shifting property of couplers, if manipulated well can act be used as perfect phase-shifters necessary in implementing negative coefficient taps. It is therefore conceivable that with the right choice of input and output terminals, a coherent signal

processing system that represents negativity without biasing (as in incoherent systems) is feasible.

### 7.3 Graphical Representation of Photonic Circuits

---

An photonic circuit can be translated directly into a signal flow diagram (SFG) as the elements in an photonic circuit and the elements in its SFG have a direct one-to-one correspondence. To effectively utilize the SFG representation in analyzing photonic circuits, the well known Mason's rule<sup>4</sup> of analyzing the SFGs is applied to the photonic circuits [9]. The key to the application of the rule is the planar SFG representation of optical coupler as shown in *Figure 7.4* [9].



*Figure 7.4: Graphical representation of optical coupler*

Photonic components other than couplers such as fiber delay lines and amplifier/attenuators have a straightforward representation in the SFG. Using the above representation for optical couplers, photonic circuits can be analyzed systematically. The result is a very powerful technique that enables a systematic mathematical analysis of photonic lumped circuits. Using this technique, the z-transfer function of a system from any one node to another (instead of just from one preset input node to a preset output node) can be calculated allowing the system designer more degrees of freedom in designing and using photonic circuits. Once the transfer function in z-domain has been obtained, as z-transform theory is very well developed one can simply apply the conventional analysis to the photonic circuits.

---

<sup>4</sup> Mason's rule can be found in many digital signal processing textbooks. The rule is applied without modifications to SFG representations of optical circuits.

Alternative to this method of analyzing photonic circuit is the matrix-method which attempts to analyze photonic circuit by direct manipulation of the coupler transfer matrix (Eq. 7.2.4). The disadvantage of such approach is that when the photonic circuit consists of more than a few photonic elements, it becomes extremely difficult to recognize what effect each element is having on the overall function of the system. Graphical approach allows direct manipulation of the photonic circuit as the correlation between a SFG and the photonic circuit it represents is very high.

The graphical method is best suited to analyzing a lumped photonic system, most likely to confirm the operations of an photonic circuit or to find new functions of an photonic circuit configuration. For further discussion on the uses of the graphical method, see Section 8.10.

***Example 7-1: Graphical method of analysis of double-coupler feedback photonic resonator[9]***

**Introduction:** Double coupler feedback photonic resonator (DCFBOR) is a configuration which results in one optical energy storage element through feedback and one interferometer through different path lengths in the feed forward path. The resulting transfer function contains one pole and one zero at the origin, and therefore the configuration can be used to realize all pole IIR filters.

**Method:** Graphical technique for photonic circuit analysis [9]

**Result:** The photonic circuit of double coupler feedback optical resonator is shown in Figure 7-5. The SFG of DCFBOR is shown in Figure 7.6.

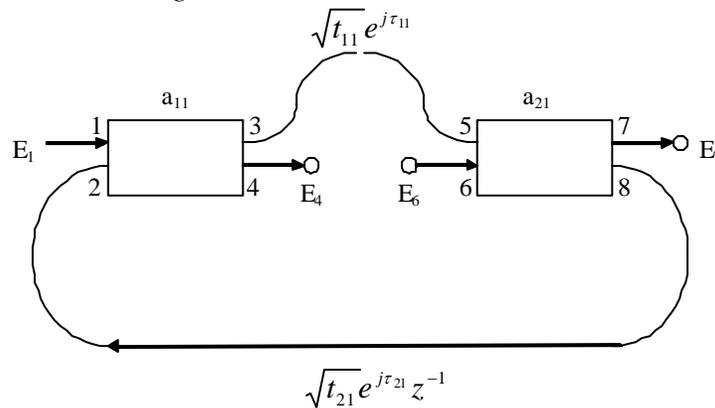


Figure 7.5: Schematic diagram of DCFBOR

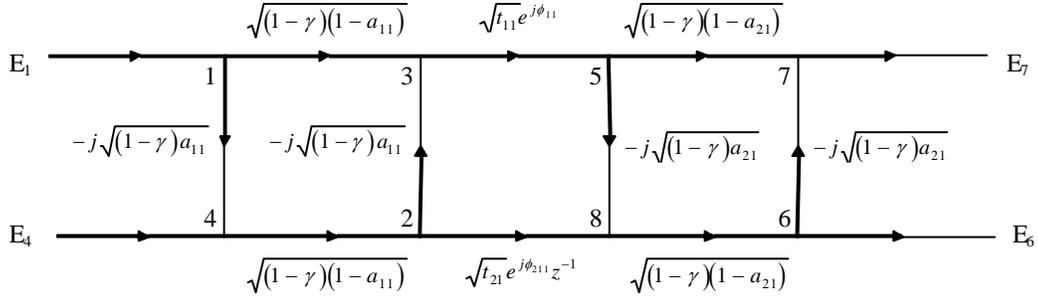


Figure 7.6: Signal flow diagram representation of DCFBOR

The details of application of Mason's rule of determining the signal flow diagram transfer functions can be found in [36]. The resulting transfer function, as expected has one zero at the origin and one pole at a location in z-plane determined by the circuit parameters.

$$\begin{aligned}
 H(z) &= \frac{E_7}{E_1} \\
 &= \frac{(1-\gamma)\sqrt{(1-a_{11})(1-a_{21})}t_{11}e^{j\phi_{11}}}{1+(1-\gamma)\sqrt{a_{11}a_{21}t_{11}t_{21}}e^{j(\phi_{11}+\phi_{21})}z^{-1}}
 \end{aligned}$$

Eq. 7.3.1: DCFBOR transfer function as determined by the graphical technique

In conclusion, the graphical technique presents a previously unavailable systematic method of analysing photonic circuits. The greatest potential will be realized when the technique is implemented in a software form as the technique can be time-consuming to apply manually if there are more than two feedback loops.

#### 7.4 Summary of Section 7

- The differences between coherent and incoherent operation of lightwave systems are described.
- The architecture and components of delayed line filters are described.
- Table 7-2 shows components that make up a photonic digital filter with corresponding element in a SFG.

Photonic implementation	Signal flow diagram element
-------------------------	-----------------------------

unit length fiber	delay line
optical amplifier, optical attenuator, etc.	multiplicative coefficient
coupler	summing/splitting point

*Table 7-2: Comparison of photonic and signal flow diagram elements*

- A method of representing photonic circuits in signal flow diagram is introduced and its advantages are outlined. It is stated that using Mason's rule, the transfer function of an photonic circuit can be obtained directly from signal flow diagrams. An example of application of the graphical method is also given.

## **8. PHOTONIC IMPLEMENTATION OF 2-D FILTERS**

---

*In Section 7, the fiber-optic signal processing technique is introduced. In this section, various structures of fiber-optic signal processing systems, some of them novel, are shown. Using these structures along with the 2-D filter design methods given in the Sections 45 and 6, 2-D filters can be implemented in photonic domain.*

### **8.1 Photonic Filter Structures**

---

In Sections 45 and 6, various methods of developing 2-D transfer function of the filter with desired characteristics have been developed. To implement a 2-D transfer function in photonic domain, as with other implementations of digital filtering systems, one must consider what kind of structure the fiber-optic filter should have in order to reduce error and at the same time, be economical.

Many of the structures used here are similar to 1-D fiber-optic filter structures and simply require cascading of the structures to construct a 2-D fiber-optic filter. However, for transfer functions not derived using matrix decomposition methods, novel structures must be devised to accommodate the requirements of non-separable transfer functions. Possible filter structures include binary tree structure, direct structure, lattice structure, parallel structure, and transversal structure all of which are described in the following sections.

## 8.2 Coherent System

---

The filters implemented in this report assume coherent operation of the laser. As explained in *Section 7.1*, this implies that the phase of incoming signal cannot be discarded and the constructive and destructive interference of signals from different paths combining must be taken into account when designing the filter. Although the restriction is somewhat harsh, it is noted in *Section 7* that there are important advantages of using coherent operation over incoherent operation. Furthermore, unless specifically noted otherwise, all filters implement 2-D FIR structures. The implementation methods for IIR filters are discussed in *Section 8.10*

## 8.3 2-D Direct Structure Filter

---

We first consider a filter structure that is suited to direct design methods of *Section 4*, i.e. no matrix decomposition. None of binary tree structure, transversal filter structure, or other 1-D structures which are introduced later this section is suitable for implementation of 2-D filters designed without using matrix decomposition. 2-D frequency sampling method for example does not generate a set of separable transfer functions and therefore cannot be implemented using 1-D structures. In such cases, two options exist for photonic implementation: (i) Break down the 2-D transfer function into a set of 1-D transfer functions by decomposing the filter transfer function using singular value decomposition and (ii) Use a 2-D direct filter structure.

2-D direct structure filter is derived from the signal flow diagram shown *Figure 3.4* in *Section 3.3.2*. Translating the signal flow diagram into photonic domain is a relatively easy task in this case and the result is shown in *Figure 8.1*.

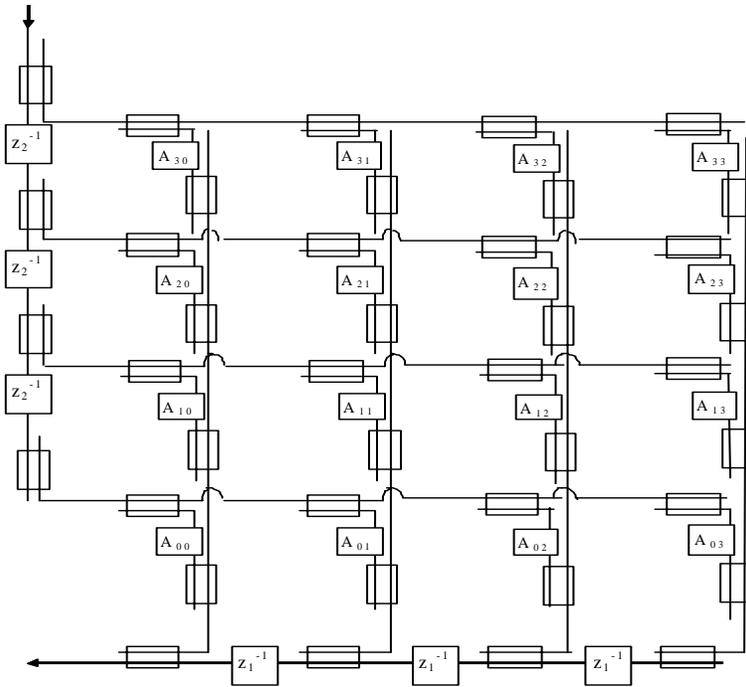


Figure 8.1: Fiber/integrated optic implementation of direct structure

In Figure 8.1, each box labeled  $A_{mn}$  represents a coefficient module which contains a one photonic attenuator (or some form of attenuation mechanism) and possibly a phase modulator. The coupler ports are arranged so that the number of cross coupling a signal path contains is four for all signal paths therefore ensuring that the phase of the signal entering each coefficient module is consistent over the whole network. The negative coefficients can then be realized by including a  $180^\circ$  phase shifter in each negative coefficient module.

It should be noted that there are two different kinds of delays. The implementation of the delays is quite simple if we are considering signal input of the form described in *Figure 2.1(c)*. For the 2-D sequence of *Figure 2.1(b)*,  $z_1^{-1}$  can be implemented as just one unit delay, and  $z_2^{-1}$  can be implemented as 7 unit delays as this is the number elements in a row of the signal.

The quantities of photonic elements required for 2-D direct photonic implementation are given by the following Eq.s.

$$\begin{aligned}
C_{2D} &= 2N_1N_2 + N_1 + N_2 \\
N_{PM2D} &= N_{OA2D} = N_1 \times N_2 \\
N_{OA2D} &= 1
\end{aligned}
\tag{Eq. 8.3.1}$$

where  $C_{2-D}$  = number of optical couplers required;  $N_{PM2-D}$  = number of optical phase modulators required;  $N_{OA2-D}$  = number of optical attenuators required;  $N_1$  = filter order in  $n_1$  dimension;  $N_2$  = filter order in  $n_2$  dimension

#### 8.4 2-D Separable Structure Filter

In the previous section, it is made clear that matrix decomposition methods of *Section 5* can be implemented in the photonic domain by cascading of 1-D fiber-optic structures since matrix decomposition methods basically generate sets of 1-D transfer functions. The 1-D structures however must be combined in a way so that the filter performs 2-D signal processing operation as intended. Essentially, the combined structure must implement a product of sum of 1-D transfer functions. Shown in Figure 8.2 is a fiber-optic structure that implements the required function.

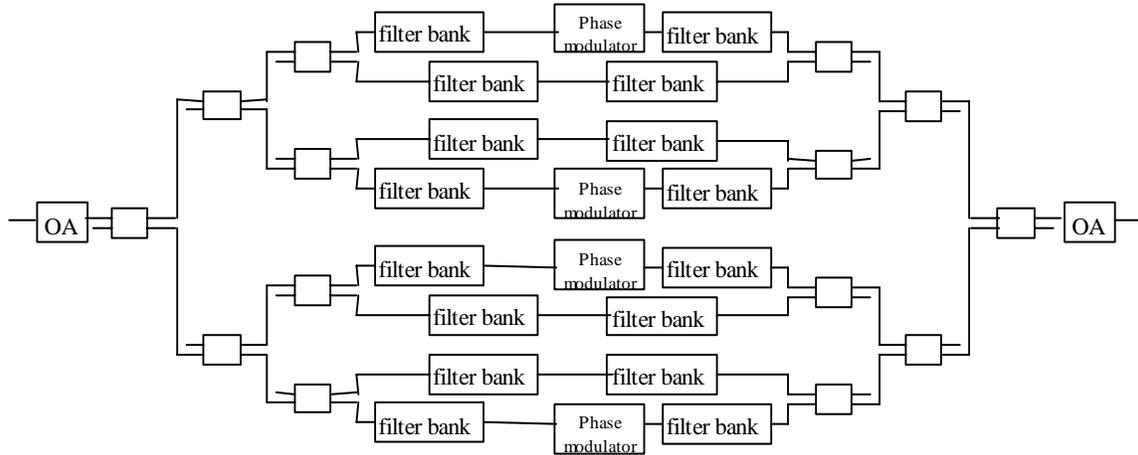


Figure 8.2: 2-D fiber-optic filter structure

Any 1-D filter implementation can be substituted for the filter banks as long as it is modular (the signal in and the signal out must have the same zero reference point). One can therefore regard the above structure as the general structure for a separable 2-D photonic filter. Specifically, Figure 8.2 implements a 2-D filter generated by matrix

decomposition with eight significant singular values. The order of the resultant 2-D filter is defined by the order of 1-D filters in the filter banks. Even though the diagram does not make any distinctions, it should be kept in mind that the length of the delays  $z_1^{-1}$  and  $z_2^{-1}$  are also different for the two filter banks in each parallel branch.

In the following sections, a number of 1-D sub-structures that can be used to implement the filter banks are described. It is up to the filter designer to choose which structure is most suitable for the particular application as any one of the structures can be substituted into the filter banks.

The number of photonic elements required for the separable 2-D structure is given by

$$\begin{aligned}
 C &= \sum_{i=0}^{\log_2 N_s} 2^{i+1} \\
 N_{PM} &= \frac{N_s}{2} \\
 N_{OA} &= 2
 \end{aligned}
 \tag{Eq. 8.4.1}$$

where  $C$  = number of optical couplers required;  $N_{PM}$  = number of optical phase modulators required(maximum);  $N_{OA}$  = number of optical amplifiers required(maximum);  $N_s$  = number of parallel stages included

## 8.5 Binary Tree Filter

---

A 1-D FIR filter transfer function can be implemented in fiber-optic format by arranging filter elements in binary tree structure as shown in Figure 8.3. The particular filter shown in the figure is a 1-D 3rd order FIR filter. The extensions to higher order filters are obvious.

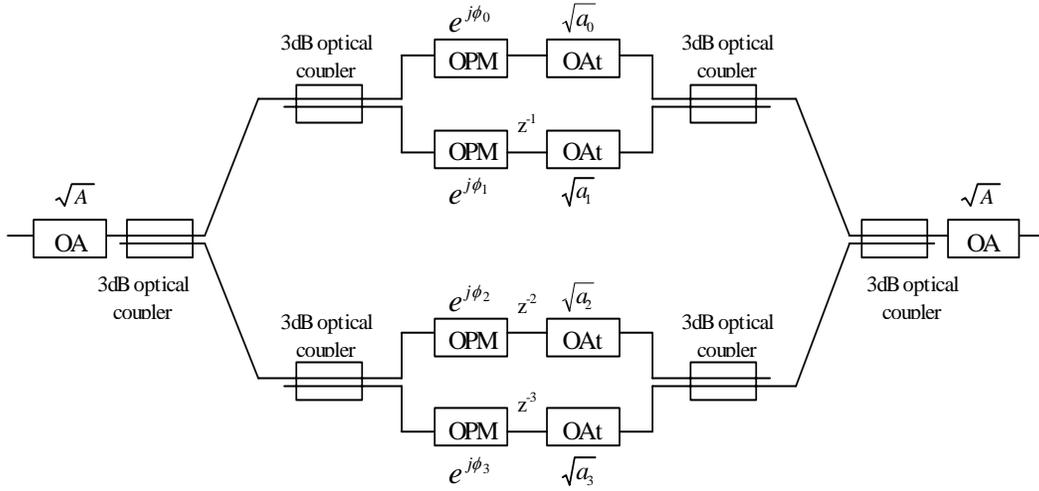


Figure 8.3: Binary tree structure [1]

The transfer function of the filter implemented with the structure can be expressed as

$$H_{az}(z) = \frac{E_{o,az}}{E_{i,az}} = (1 - \gamma)^{\log_2(k+1)} (k + 1)^{-1} A \sum_{d=0}^k (-1)^d \exp(j\phi_d) \sqrt{a_d} z^{-d} \quad \text{Eq. 8.5.1}$$

$(1 - \gamma)^{\log_2(k+1)}$  = common excess loss factor

$(k + 1)^{-1}$  = common 3 dB loss factor of the couplers

where  $A, a_d$  = intensity attenuation of optical attenuators

$\phi_d$  = optical phase shift of the optical phase modulator

$d$  = delay order

Eq. 8.5.2: Transfer function of the binary tree structure filter[1]

Because the system is not intensity based, the signal is assumed to have the form  $E_o e^{j(\omega_o t + \phi(t) + \varepsilon)}$  which includes the optical phase  $\phi$ . Negative coefficients of the transfer function can therefore be achieved by shifting the phase of the signal by  $180^\circ$  through phase modulators [1]. With appropriate manipulation of the optical coupler characteristic as a perfect  $-90^\circ$  phase shifter, the use of optical modulators to achieve negative coefficients can be reduced or even completely eliminated as with transversal filter structure shown in the next section.

The advantage of the binary structure is that the number of 3 dB splitting stages the signal must travel through is the minimum achievable. The loss due to splitting is thus minimum and the effect of noise due to splitting is also minimum. The structure is used in [1] to implement a fiber-optic integrator in which the input and output terminals of couplers are selected so that the sign alternation in the numerator that is the characteristic of Newton's family of digital integrators is achieved without phase modulators. However, this (alternating sign) is not true for an arbitrary filter transfer function and therefore most filter transfer functions would necessitate the use of optical phase modulators (OPMs) as shown in Figure 8.3.

The number of photonic elements required for the binary tree structure is given by

$$\begin{aligned}
 C_{BT} &= \sum_{i=0}^{\log_2 N} 2^{i+1} \\
 N_{PM_{BT}} &= N \\
 N_{At_{BT}} &= N \\
 N_{A_{BT}} &= 1
 \end{aligned}
 \tag{Eq. 8.5.3}$$

where  $C_{BT}$  = number of optical couplers required;  $N_{PM_{BT}}$  = number of optical phase modulators required(maximum);  $N_{At_{BT}}$  = number of optical attenuators required;  $N_{A_{BT}}$  = number of optical amplifiers required(minimum) and  $N$  = order of the 1-D filter sections

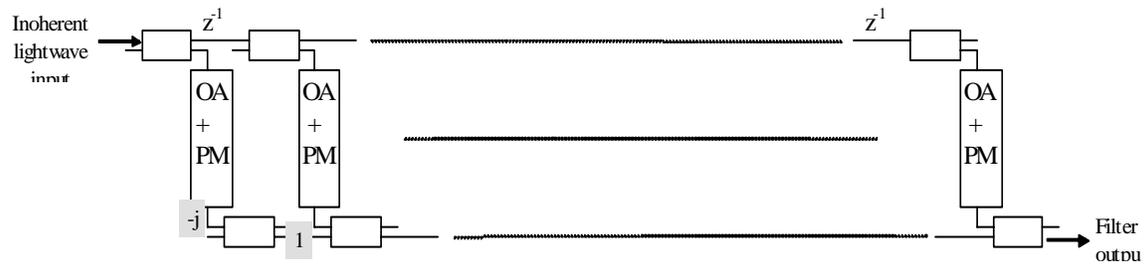
## 8.6 Photonic Transversal Filter

---

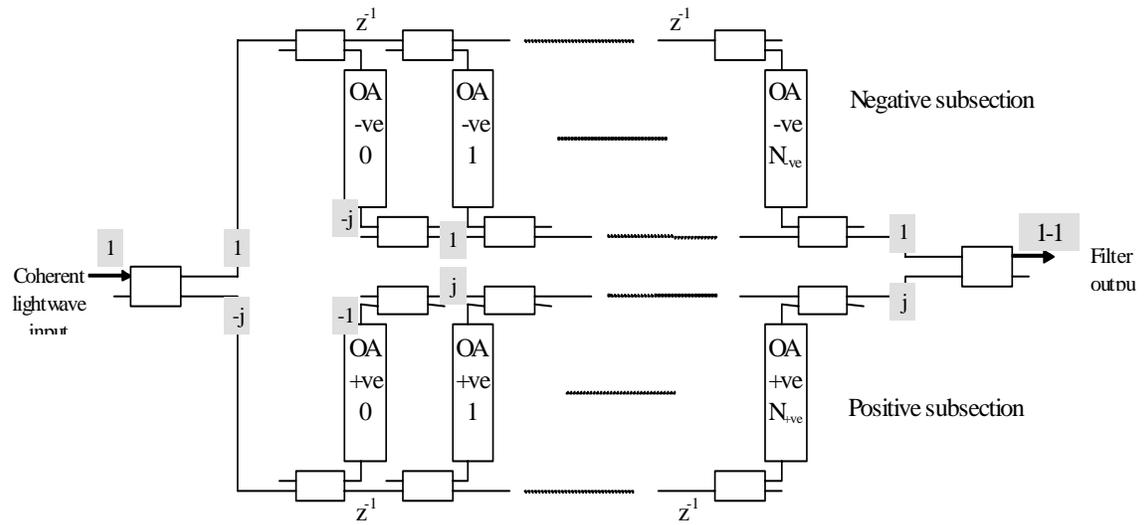
In the previous section, it is shown that the binary tree structure requires phase modulators except in cases when the signs are alternating or when there is equal number of positive coefficients as negative coefficients in the transfer function. Such cases are rare, and in most cases phase modulators will be needed to achieve the negativity. In this section, a structure that does not require any phase modulators is proposed. This structure, based on the transversal filter structure in [5](shown in Figure 8.4(a)) achieves 180° phase shift required by negative coefficients through appropriate arrangement of coupler ports. The signal paths for positive coefficients contain four cross couplings resulting in  $-j \times -j \times -j \times -j = 1$ , i.e. no phase shift, and all signal paths for negative

coefficients contain two cross couplings resulting in  $-j \times j = -1$ , i.e.  $180^\circ$  phase shift. Shown in *Figure 8-4(b)* is the proposed transversal filter structure that does not require phase modulators to achieve negative coefficients. The phase of the signal at various points of the filter is shown in gray scale.

The proposed structure is particularly suited to the transfer functions generated by matrix decomposition as it can implement a 1-D FIR transfer function with very little modifications (needed for the coupling attenuations) to its filter coefficients. How the additional sections can be accommodated to form a 2-D filter is shown later in this section.



(a)



(b)

*Figure 8.4 (a) The original transversal structure [5] and (b) The proposed structure*

The advantages of the proposed filter structure over the simple transversal filter structure in [5] are threefold. First, as mentioned previously, is the fact that no phase modulators are required to achieve negative coefficient. The second advantage is that no signal biasing or differential detection is required. The implication of the second advantage is that the filter structure is modular and can therefore be connected in parallel or in series without any modification to the filter module. Another advantage of the proposed structure is that the signal goes through less number of couplers than the implementation of [5]. This property can be observed if one considers the fact that the signal only goes through in average  $N/2$  couplers for the proposed structure (since the coefficients are split into two groups) whereas for the traditional transversal filter the number of couplers the signal must travel through is always  $N$ , where  $N$  is the number of coefficients of the filter transfer function which may or may not be the same as the order of the filter.

The proposed structure can also be used for adaptive filtering as only the coefficients of the optical amplifier/attenuator need to be changed to modify the filtering operation. However for adaptive operation, the filter coefficients that are zero and therefore not included in the original structure may have to be included in case the adapting algorithm changes them to non-zero values. The situation is slightly disadvantageous for the proposed structure compared to the simple transversal structure of [5], because the number of positive and negative coefficients ( $N_{-ve}$  and  $N_{+ve}$ ) of an  $N$ th order filter can range from 0 to  $N+1$  and both sets of coefficients need to be fully implemented. Fortunately the situation is not as bad as one might expect. At first, it appears that the number of optical attenuators (OAs) needed is  $2N$  since both negative and positive sections should implement the full order of the filter transfer function. For the proposed structure, it can be shown that the number of OAs required is approximately  $(N+1) \times 1.5$  ( $N+1$  components are required for the simple transversal filter structure of [5] for adaptive operation). The reason for requiring only  $(N+1) \times 1.5$  instead of  $2N$  can be explained as follows.

Assume that we can exchange the input connections to two parallel sections using optical switches so that the signs of the two parallel sections can be reversed. First consider the

case where the number of positive coefficients and negative coefficients of the filter transform function are exactly the same, i.e.,  $N_{+ve}=N_{-ve}=(N+1)/2$ . Clearly  $(N+1)/2$  OAs are needed for each parallel subsection and therefore total of  $N+1$  OAs are required. Now consider the case where one sign is completely dominant over the other, i.e.  $N_{+ve}=N+1$  and  $N_{-ve}=0$ , or  $N_{-ve}=N+1$  and  $N_{+ve}=0$ . In this case,  $N+1$  OAs are necessary in at least one of the parallel subsections even though the other subsection does not need any. To accommodate for both situations with one filter structure, it is clear that  $(N+1) + (N+1)/2 = 1.5(N+1)$  OAs are necessary. The result for optical attenuators can be extended to phase modulators and couplers with a few simple modifications.

In the case where adaptive filtering is not required, total number of components required is minimized by the use of the proposed structure. For 1-D digital filtering purposes also, this structure is economical, modular and easy to implement. *Example 8-1* is a simple exemplar of a fiber-optic filter realized from a transfer function using the parallel structure.

***Example 8-1: Photonic filter implementation using the proposed transversal structure***

**Design Aim:** A sample implementation of a simple transfer function

**Method:** The proposed transversal structure. Fiber losses and other non-linear effects are ignored.

**Results:**

$$H(z^{-1}) = 1 + 0.2z^{-1} - 0.3z^{-2} - 0.2z^{-3} + 0.4z^{-4} + 0.6z^{-5} - 0.5z^{-6}$$

*Eq. 8.6.1: Filter transfer function to be realized using the proposed transversal structure*

The first step is to divide the transfer function into two parts, namely the positive coefficients and the negative coefficients, denoted by  $\mathbf{k}_+$  and  $\mathbf{k}_-$ . It is assumed that all the coefficients are realized using optical attenuators and not amplifiers (optical attenuator is known to produce less noise than does the optical amplifier).

An observation of the filter structure shown previously in *Figure 8-4* will reveal that all streams of lightwaves in each parallel section pass through the same number of couplers and therefore suffers the same attenuation from coupling. In this example, the positive coefficients suffer  $(\sqrt{2})^7$  attenuation whereas the negative coefficients suffer a  $(\sqrt{2})^6$  attenuation. Optical preamplifier gain is therefore set at  $(\sqrt{2})^7$  and the negative coefficient attenuators are set at  $h_n \div \sqrt{2}$  where  $h_n$  denotes the negative coefficient values. This factor of  $\sqrt{2}$  compensates for the effect of the extra coupler positive coefficient signals go through. Optical preamplifier is set at  $(\sqrt{2})^k$  where  $k$  is the greater of  $k_+$  and  $k_-$  to compensate for the coupling losses.

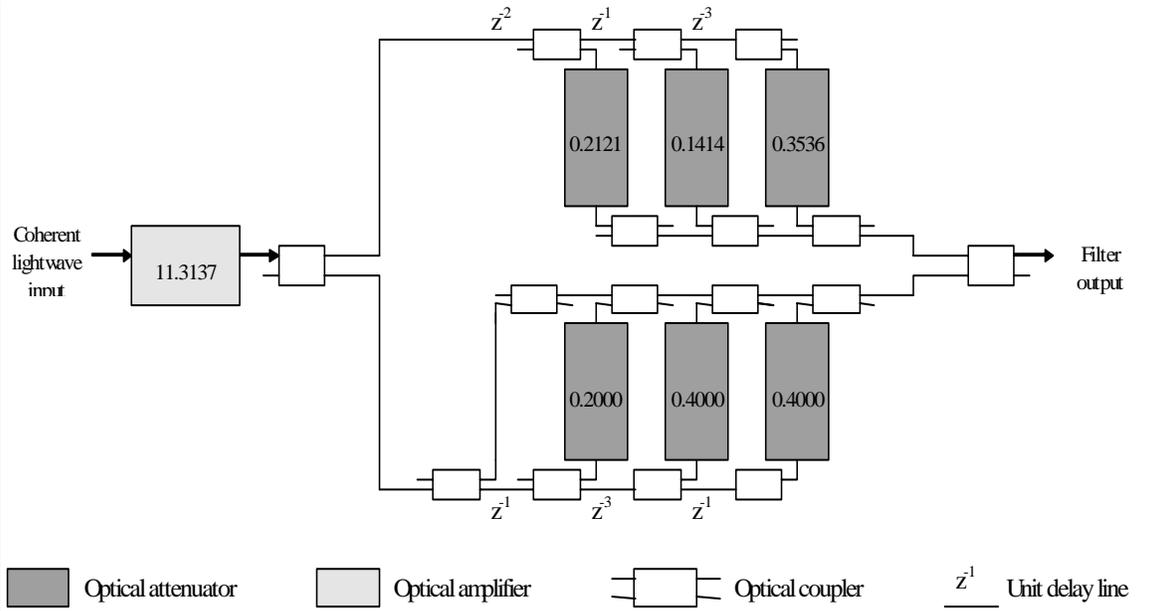


Figure 8-5: Fiber optic implementation of Eq. 8-2 using the proposed transversal filter structure

Because the binary tree splitting stages of Figure 8.3 add further attenuation to the lightwave when the transversal structure is incorporated into a 2-D system, the filter amplifier and the attenuator settings must be adjusted accordingly.

The number of optical elements required for the transversal structure is given by

$$\begin{aligned}
 C_{TF} &= 2N + 2 \\
 N_{PMTF} &= 0 \\
 N_{ATF} &= N \\
 N_{ATF} &= 1
 \end{aligned}
 \tag{Eq. 8.6.2}$$

where  $C_{TF}$  = number of optical couplers required;  $N_{PMTF}$  = number of optical phase modulators required;  $N_{ATF}$  = number of optical attenuators required;  $N_{ATF}$  = number of optical amplifiers required (minimum) and  $N$  = order of the 1-D filter sections

## 8.7 1-D Direct Structure Photonic Filter

1-D direct structure provides the classic alternative to the binary tree structure and transversal filter structure for implementation of 1-D filters. Unlike the structures mentioned, direct structure can also implement IIR filters very easily. The direct structure

signal flow diagrams can be found in most signal processing books and is quite straightforward. However, photonic implementation of the direct structure has not been developed as yet and this section shows an implementation of the structure using fiber-optic elements.

$$H(z) = \frac{\sum_{k=0}^L b_k z^{-k}}{1 - \sum_{k=1}^N a_k z^{-k}} \quad \text{Eq. 8.7.1}$$

The transfer function shown in Eq. 8.7.1 can be represented in signal flow diagram format as shown in Figure 8-6 and the photonic implementation is shown in Figure 8.5.

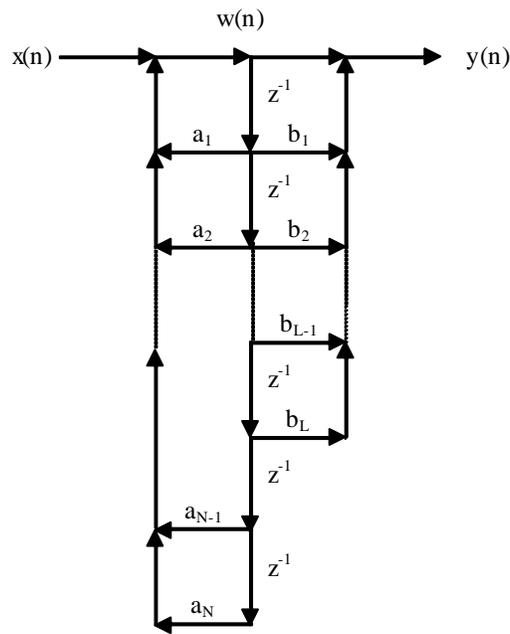


Figure 8.5: Signal flow diagram of 1-D direct structure

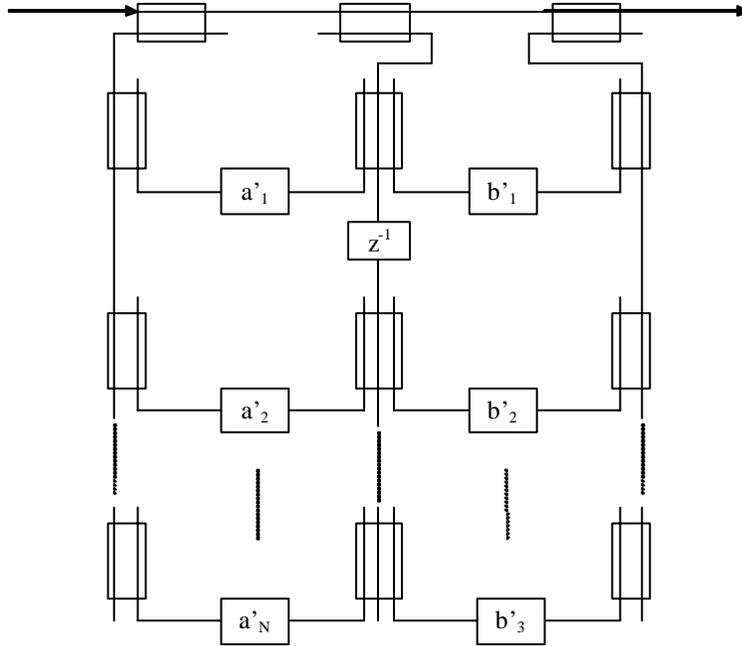


Figure 8.6: Photonic implementation of direct 1-D structure

As with the 2-D direct structure, each signal path contains four cross-couplings and therefore suffers no overall phase shift ( $-j \times -j \times -j \times -j = 1$ ). The boxes in the horizontal branches represent coefficient module identical to that in 2-D direct structure and the boxes in the vertical branches represent the delay elements. The structure is optimal in the sense that the number of delay elements is minimal and can be used to implement 1-D IIR functions in the photonic domain. The couplers used in the middle section of the implementation represent either 3x3 couplers or a cascade arrangement of two 2x2 couplers shown in Figure 8.7.

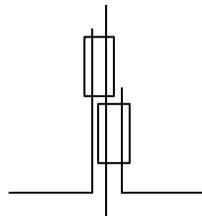


Figure 8.7: An arrangement of two 2x2 couplers to form a 3x3 coupler

The number of photonic elements required for the direct structure is given by

$$\begin{aligned}
C_{DF} &= 3N + N_{num} + 3 \\
N_{PMDF} &= N_{AtDF} = N_{num} + N_{den} \\
N_{ADF} &= 1
\end{aligned}
\tag{Eq. 8.7.2}$$

where  $C_{DF}$  = number of optical couplers required;  $N_{PMDF}$  = number of optical phase modulators required;  $N_{AtDF}$  = number of optical attenuators required;  $N_{ADF}$  = number of optical amplifiers required;  $N_{num}$  = order of the 1-D filter numerator and  $N_{den}$  = order of the 1-D filter section denominator

## 8.8 Parallel Structure Filters

---

Parallel structure is a simple variation on the theme on the direct structure. This structure is slightly different from the direct structure in that signals pass through less number of couplers and therefore the signals are not attenuated as much.

Basically the structure can be thought of as a parallel arrangement of 2nd order direct structures. The signals are split into several branches at the beginning and after the signals have traveled through the parallel sub-structures, they are merged back into one signal path. To implement the parallel structure, the filter transfer function must be factorized into a sum of several sub-transfer functions. If each sub-transfer function is in the form of 2<sup>nd</sup> order substructure, it is then possible to turn the sub-transfer function into signal flow diagram form which in turn can be realized in the optical domain.

### ***Example 8-2: Filter design using parallel structure filter***

Design Aim: A 1-D low-pass filter of 7th order with normalized cut-off frequency at 0.3

Method: Parallel factorization

Program used: PARALLEL.m

Results: The transfer function designed using Chebychev approximation routine in MATLAB simulation package is decomposed as shown below in signal flow diagram format. It is found that the performance of the two implementations are nearly identical (see *Figure 8-9*).

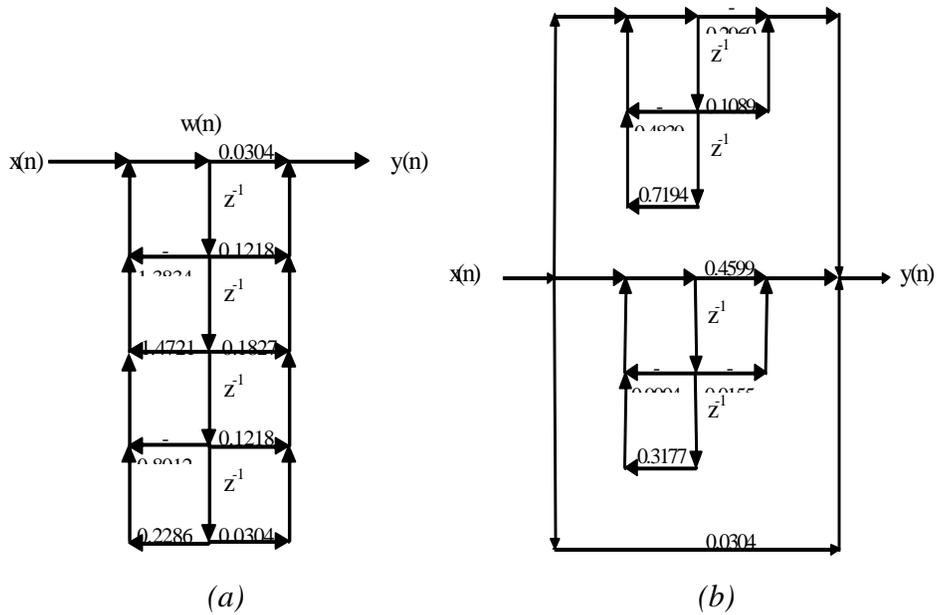


Figure 8-9: Parallel structure example signal flow diagram (a) Direct structure (b) Parallel structure

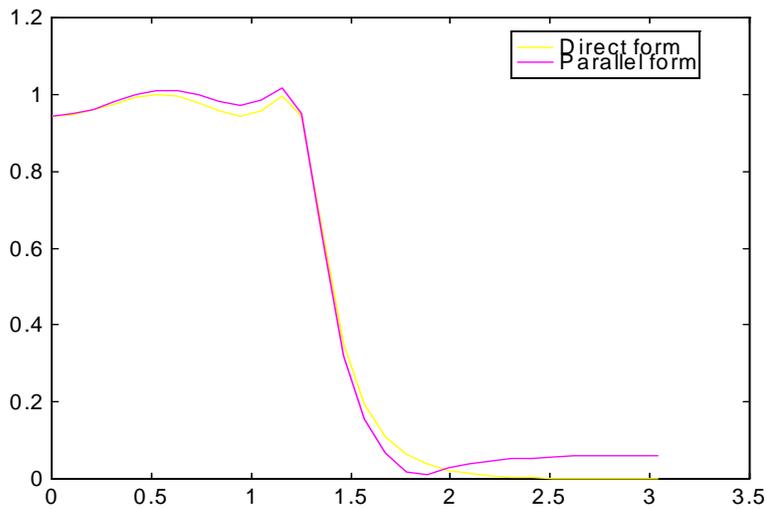


Figure 8-10: Frequency response of direct and parallel structures

For photonic implementations of parallel structures, the splitting of signal at the beginning of the structure can be performed using a binary tree of couplers in which case the amplitude and the phase change caused by cross-coupling must be considered. The task is very similar to that performed for binary filter structure. Essentially, the phase changes can be corrected by including a  $180^\circ$  shift every second parallel branch, and

amplitude compensation is equal to 2 raised to  $\log_2 P$  where P is the number of parallel stages (which must be a power of 2).

The performance of the parallel structure is similar to that of the direct structure as can be seen in *Figure 8-10*, and the number of delays as well as the number of amplifiers/attenuators required are also the same. However, the number of summer/splitters in parallel structure is greatly increased and in *Section 9.3.3.2*, it will be shown that the number of optical couplers needed for the implementation of parallel structure is also very much larger than that required for other 1-D structures.

The number of photonic elements required for the parallel structure is given by

$$\begin{aligned}
 C_{PF} &= \frac{N_{den}}{2} \times 7 + \sum_{i=0}^{\log_2 \frac{N_{den}}{2}} 2^{i+1} \\
 N_{PMPF} &= 2N_{den} + \frac{\log_2 \frac{N_{den}}{2}}{2} \\
 N_{AtPF} &= 2N_{den} \\
 N_{APF} &= 1
 \end{aligned}
 \tag{Eq. 8.8.1}$$

where  $C_{PF}$  = number of optical couplers required;  $N_{PMPF}$  = number of optical phase modulators required;  $N_{AtPF}$  = number of optical attenuators required;  $N_{APF}$  = number of optical amplifiers required (minimum);  $N_{num}$  = order of the 1-D filter numerator and  $N_{den}$  = order of the 1-D filter section denominator

## 8.9 Other 1-D Filter Structures

---

Several other filter structures exist for filter realization in optical domain. Lattice form(see *Figure 8-11*) is a structure particularly suited to incoherent systems since each section of the signal flow diagram is identical to the optical coupler signal flow diagram shown in *Figure 7.3(a)*. The coefficients k can be implemented using optical amplifier/attenuators. The lattice structure is not suitable for coherent signal processing as coherent systems must take phase into account and since the lattice structure assumes

that the cross coupling does not cause any phase shift, it will be inefficient and difficult to implement coherent systems using the lattice structure.

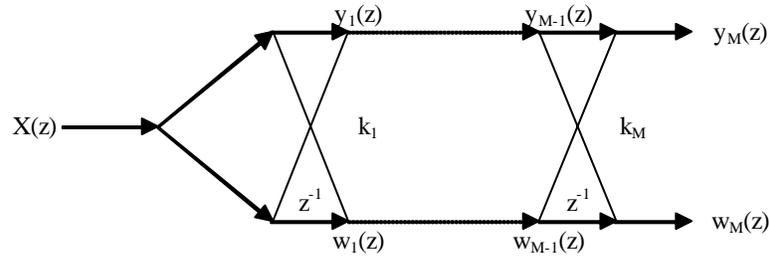


Figure 8.8: Lattice form signal flow diagram

## 8.10 Realization of Poles

In discussing feed forward structures such as the binary tree structure or the direct structure, the realization of the denominator part of filter transfer functions are ignored. This can be justified in the case of finite impulse response filters (FIR) where the denominator is simply 1. In cases of IIR filters, the denominator must be incorporated into the system as in 1-D direct structure. An alternative is to form a pure pole filter by using a feed forward structure in a feedback loop as shown in Figure 8.9.

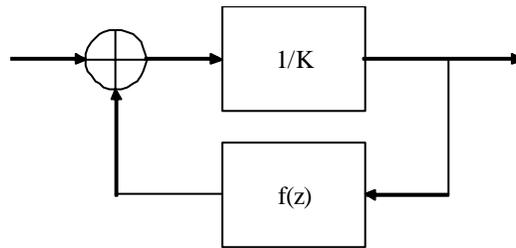


Figure 8.9: Denominator realization using feed forward structure

Assuming that the denominator of transfer function can be written as

$$p(z) = K + f(z) \quad \text{Eq. 8.10.1}$$

where  $f(z)$  has no constant terms, applying the feedback structure results in the following transfer function:

$$H_p(z) = \frac{\frac{1}{K}}{1 + \frac{1}{K} f(z)} \quad \text{Eq. 8.10.2}$$

It is therefore possible to use the feed forward structure such as transversal structure without modifications in feedback branch to form the denominator of an IIR filter transfer function. The overall filter structure incorporating numerator  $q(z)$  and denominator  $K+f(z)$  is shown in Figure 8.10.

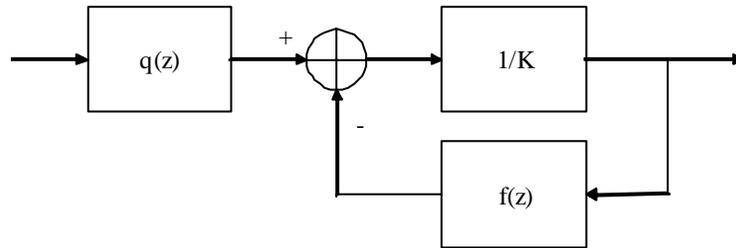


Figure 8.10: IIR filter structure using FIR subsections

Note that  $q(z)$  and  $f(z)$  can be replaced by  $q(z_1, z_2)$  and  $f(z_1, z_2)$ . The structure in Figure 8.10 can therefore be used to implement a subsection of separable filters of Section 5 or a complete 2-D direct filter implementation of Section 4. Example 8-3 implements a case of the former.

**Example 8-3:** A 2-D recursive filter subsection realization

Design Aim: The recursive transfer function given in Eq. 8-3.

$$H(z^{-1}) = \frac{1 + 0.2z^{-1} - 0.3z^{-2}}{1 - 0.2z^{-3} + 0.4z^{-4} + 0.6z^{-5} - 0.5z^{-6}}$$

Eq. 8.10.3: A 2-D filter subsection transfer function

The filter subsection can be implemented in photonic domain as shown in Figure 8.11.

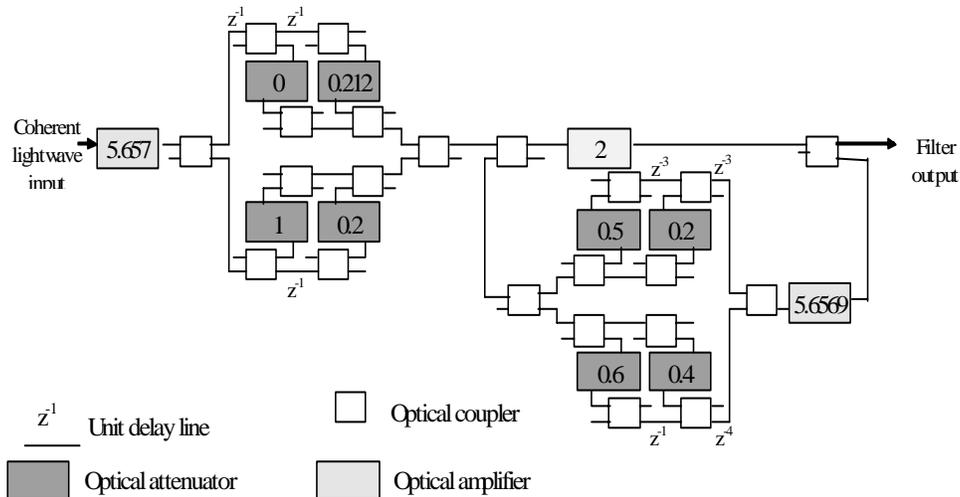


Figure 8.11: A 1-D IIR filter subsection example

Finally, when the SFG is ready, the graphical technique introduced in *Section 7.3* can be used to check the validity of the design. However, applying the technique by hand to photonic circuits shown in this section would be time-consuming and error prone as there is too many photonic elements to consider. In fact, applying any graphical technique by hand to the photonic circuits in this section would be impractical. Due to the impracticality of performing a graphical analysis by hand, the graphical analysis of the photonic circuits is omitted here. Readers are referred to the recent report [26]

## 8.11 Summary of Section 8

- The significance of using coherent systems for PSP stated.
- 2-D direct structure illustrated.
- Various matrix decomposition realizations using 1-D filter structures shown.
- Realizations of IIR filters using FIR filter sections shown.

## 9. DISCUSSIONS AND ANALYSIS

---

---

### 9.1 Summary of the Report

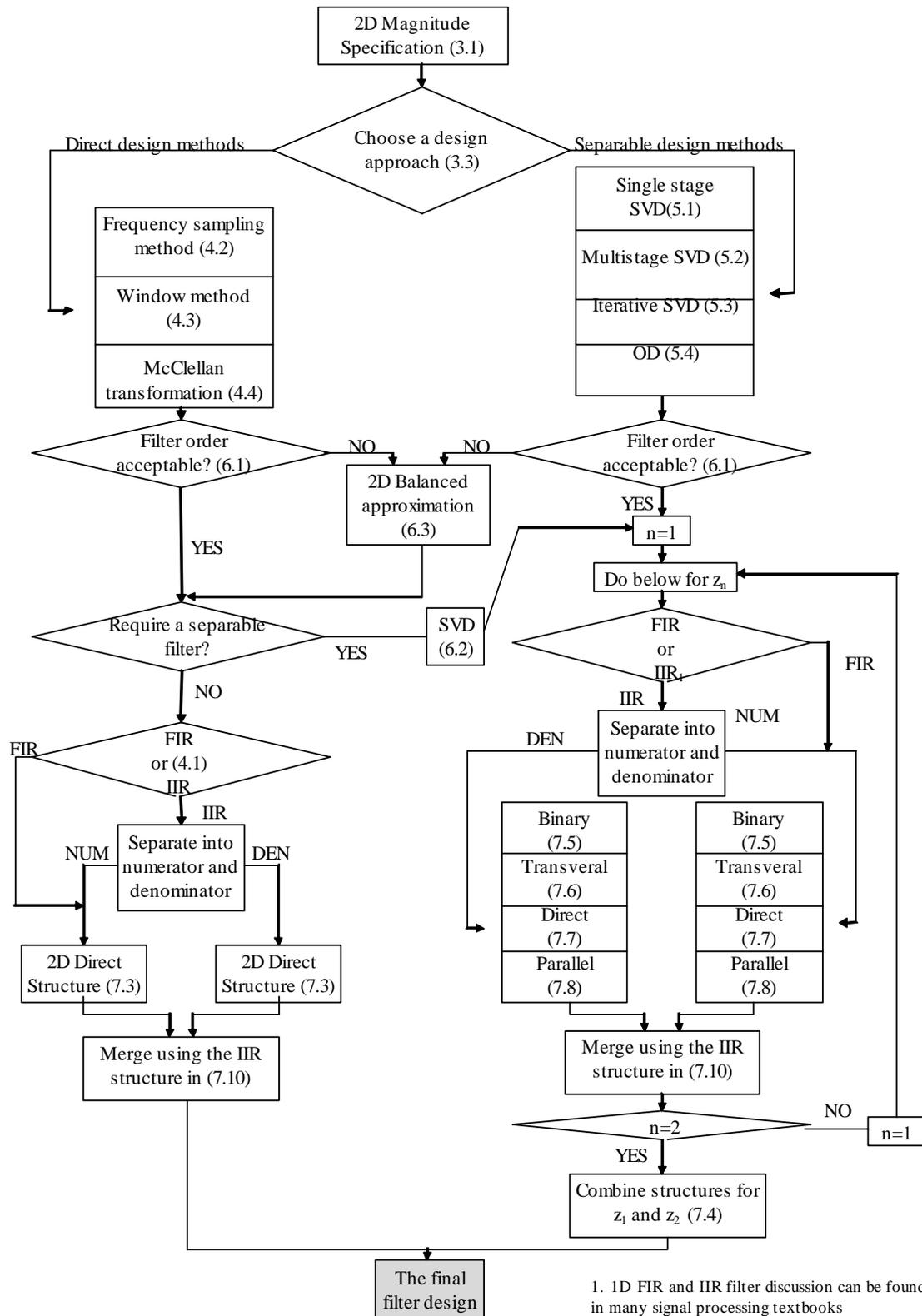
---

- The issues in multidimensional signal processing are introduced (Section 2)
- Mathematical models for 2-D filters are given (Section 3).
- Alternative streams of 2-D filter design methods are introduced and implemented (Sections 4 and 5).
- A 2-D filter order reduction method is implemented (Section 6).
- The issues in 2-D signal processing using fiber-optic techniques are discussed (Section 7).
- Structures for photonic implementation of 2-D filters are designed or introduced (Section 8).

### 9.2 2-D Photonic Filter Design Flowchart

---

In designing and implementing a 2-D filter, several design decisions must be made. The following flowchart illustrates the possible paths that may be taken when designing and implementing a 2-D filter in photonic domain. Tables in the chart show various design methods any one of which may be substituted for another within the same table. For example for separable design methods path (the branch on the right side of the flowchart), any one of single stage SVD, multistage SVD, iterative SVD, or OD may be used to calculate the appropriate 2-D transfer function. The numbers inside brackets show the relevant section which to be referred to.



1. 1D FIR and IIR filter discussion can be found in many signal processing textbooks

Obviously, if a design result is not satisfactory, some of the procedures must be repeated

to find a better solution. The iteration paths are not shown in the flowchart as they are reasonably obvious.

### 9.3 Examples of Photonic 2-D PSP Implementation

---

In this section, 2-D filter implementations of the same frequency specification are performed from scratch first using a matrix decomposition method, and then a direct method. The design methods used in the example are by no means ‘ideal’. As shown in the flowchart show in the previous page, there are many possible paths and ultimately, the choices lie with the filter designer.

#### 9.3.1 The Specification

The filter to be designed is a low-pass filter with normalized spatial cut off frequency of 0.7 in x and y directions (see Figure 9.1). The overall filter magnitude response error defined by Eq. 4.2.1 is to be less than 10%. The 2-D signal is transmitted coherently through an optical fiber medium using linear sequencing of a  $256 \times 256$  pixel frame starting from  $x=0$  and  $y=0$ . The frequency response of the filter is to be circularly symmetric. The order of the filter should be sufficiently low to enable it to be implemented in photonic domain. The phase response of the filter should be linear. The filter is intended to be a noise remover as most noise components are in the high frequency band.

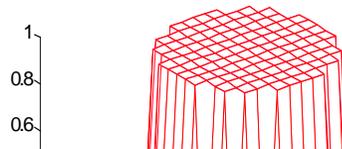


Figure 9.1: Magnitude specification

### 9.3.2 Choice of a Design Methodology

As the first decision to be made, a choice between direct design methods and separable design methods must be made. The following table shows the differences between separable and direct design methods.

Property	Direct design methods	Separable design methods
<b>Number of parallel stages</b>	1	$\geq 1$
<b>Filter structure</b>	FIR or IIR	FIR or IIR
<b>Filter design procedure</b>	One 2-D filter design	Many 1-D filter designs
<b>Coherent processing</b>	Yes	Yes
<b>Performance</b>	Depends on the order of the filter and the 2-D filter design method	Depends on number of additional stages and the 1-D filter design method
<b>Fiber-optic implementation</b>	Direct structure only	Binary, parallel, transversal, direct

Table 9-1: Comparison of direct structure and separable structure

Although the two methodologies result in very different designs, there are no apparent advantages to be gained from preferring one structure over another. There are multitude of factors contributing to the actual performance and the economy of the designed filter and the best design will be obtained by using both methodologies and comparing various statistics to find out which design method gives better results for the particular application.

### 9.3.3 Separable Implementation Using Matrix Decomposition Methods

#### 9.3.3.1 Choice of a Decomposition Method

There are four decomposition methods to consider. All methods are tried to compare the performances.

	Single stage SVD	Multiple stage SVD	Iterative SVD	Optimal decomposition
<b>Number of parallel stages</b>	1	2	2	2
<b>Resulting design</b>	FIR (or IIR) <sup>5</sup>	FIR (or IIR)	FIR (or IIR)	FIR (or IIR)

<sup>5</sup> ( ) contains the possible alternatives.

<b>Computational requirements</b>	Minimal	Intermediate	Intermediate	Very heavy
<b>Major claim for advantage</b>	Simple and quick	Accurate	Easy 1-D filter design	Accurate and easy 1-D filter design
<b>Major disadvantage</b>	Too rough (large error)	Phase must be considered when designing 1-D filter sections	Not accurate enough	Not accurate enough & heavy computational load
<b>Filter Error(%)</b>	10.7262	8.2740	9.9603	9.8521
<b>Order</b>	16	16	16	16
<b>Total no of multiplications required</b>	32	64	64	64

*Table 9-2: Comparison of various decomposition methods*

The comparison is made on the basis of the least number of parallel stages that will satisfy the specified 10% overall error. Also, the 1-D filter design methods used are identical for the four different decomposition methods (note that perhaps this is not quite fair on methods such as ISVD algorithm or OD algorithm which put their strengths on making filter design task easier by keeping the 1-D magnitudes all positive). As can be seen in the table, the best performance can be obtained by multiple stage SVD algorithm that, with two parallel stages, results in a filter error of 8.274%.

As multiple stage SVD algorithm produces four sets of 1-D magnitude responses, there are four 1-D filters to be implemented and their coefficients are shown in *Table 9-3*. The 1-D filter design method used is least-squares algorithm.

Order	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$z_1$ stage 1	0.0004	-	-	0.0114	0.0083	-	0.0819	0.5015	0.5015	0.0819	-	0.0083	0.0114	-	-	0.0004
$z_1$ stage 2	0.0022	0.0055	-	0.0037	0.0406	-	-	0.0082	0.0082	-	-	0.0406	0.0037	-	0.0055	0.0022
$z_2$ stage 1	-	0.0028	-	-	0.0600	0.0849	0.0071	0.0139	0.0139	0.0071	0.0849	0.0600	-	-	0.0028	-
$z_2$ stage 2	-	0.0080	0.0130	-	-	-	0.0028	0.0050	0.0050	0.0028	-	-	-	0.0130	0.0080	-
	0.0025			0.0165	0.0338	0.0109					0.0109	0.0338	0.0165			0.0025

*Table 9-3: Coefficients of the designed filter*

The filters shown above are symmetric about the centre (which is formed by the coefficients for 7th order and 8th order), therefore the phase response is guaranteed to be linear. The repeated coefficients also raise the possibility of saving of components through exploitation of symmetry.

The filter order of 15, although slightly too high for photonic implementation, will be retained for the example purpose. If the filter order is to be reduced at this stage, then one would merge the separable expressions into a single expression, then use the balanced approximation method illustrated in *Section 6*. Such a procedure would result in an IIR filter design of reduced order.

### 9.3.3.2 Photonic Implementation of the Separable 2-D Filter

Because we are dealing with 2-D separable structure, we need to decide which 1-D photonic filter structure should be used to implement the four 1-D filter stages. *Table 9-4* is a comparison of the structures on the number of components required.

	Binary	Transversal	Direct	Parallel
<b>No of splitting stages per amplifier</b>	10	19	30	10
<b>Total no. of optical phase modulators</b>	34	0	64	96
<b>Total no. of optical attenuators</b>	64	64	64	96
<b>Total no. of optical couplers</b>	126	142	136	184
<b>Total no. of optical amplifiers</b>	4	4	4	4
<b>Total total no. of components</b>	228	210	268	380
<b>Remarks</b>	Low attenuation	No phase modulators required	Simple, but inefficient	Inefficient, but low attenuation

*Table 9-4: Comparison of different fiber-optic filter structures*

In Table 9-4, the number of splitting stages per OA is used to indicate how many signal splitting stages the optical pre-amplifier must compensate for. If 30 stages must be compensated for as in the direct structure, the amplification factor would need to be  $(\sqrt{2})^{30}=32768$  which is clearly unacceptable for optical amplifiers. A solution such as including several OAs in cascade would be required in such situations provided the accumulated ASE noises do not surpass the signal level. For the example, transversal filter structure is chosen as the implementation as it appears to be the most efficient structure. Below diagram shows the schematic diagram for the photonic implementation.

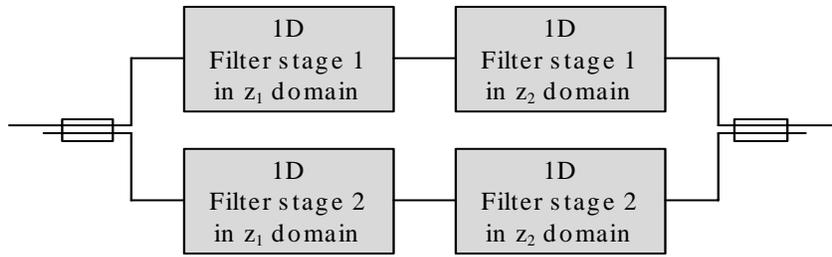


Figure 9.2: Schematic diagram of the separable 2-D filter

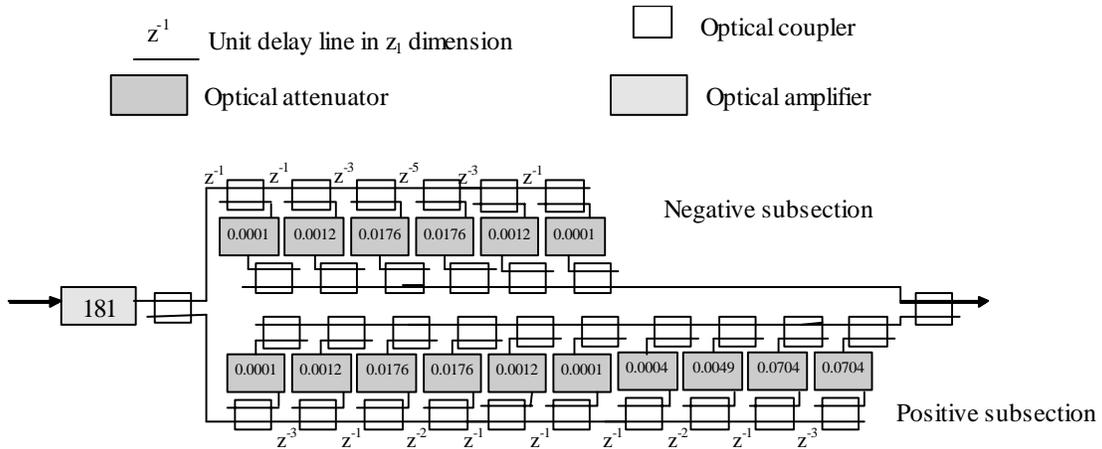


Figure 9.3: 1-D filter stage 1

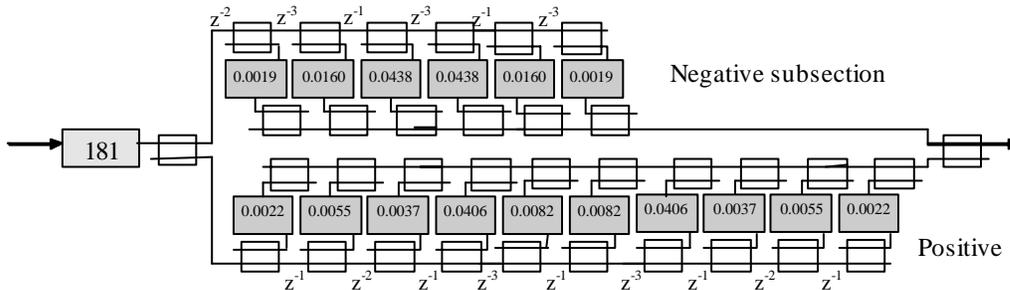


Figure 9.4: 1-D filter stage 2

Because the filter is circularly symmetric, the filter structure and attenuator/amplifier coefficients are the same for the second dimension  $z_2$  except for the delay which must be replaced by  $z_2^{-1}$ . Physically, this delay would correspond to  $(z_1^{-1})^{255}$  as one line delay is same as the entire row of pixels. For the bottom parallel section of Figure 9.3, the positive subsections and the negative subsections of the two 1-D subsections will need to

be exchanged. This is because the binary splitting stages at the start and end of the structure shown in *Figure 9.2* introduce another phase shift of  $180^\circ$  for the lower stage. Exchanging of subsections can compensate for the phase shift. The filter coefficients are related to the attenuator settings by

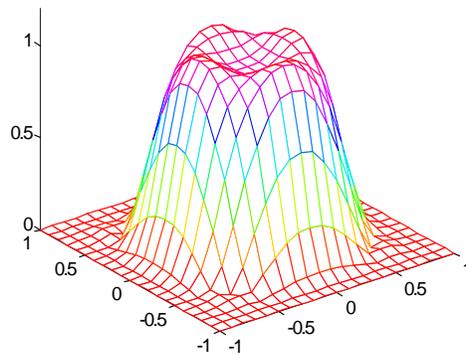
$$a_{+i} = \frac{h_i}{\sqrt{2}^{k+2}}, a_{-i} = -h_i$$

where

$$k = |\text{No of + ve terms} - \text{No of - ve terms}|$$

*Eq. 9.3.1: Relationship between the filter coefficients and the attenuator settings*

where the number of positive terms is greater than the number of negative terms in the transfer function.  $a_{+i}$  and  $a_{-i}$  can be exchanged and multiplied by -1 to obtain the settings if the reverse is true. The magnitude response of the filter is plotted in *Figure 9.5*.



*Figure 9.5: Magnitude response of the designed 2-D filter*

### **9.3.4 Non-Separable Implementation Using Direct Methods**

#### **9.3.4.1 Choice of a Design Method**

There are two methods to choose from: frequency sampling method and McClellan transformation method. The designs results are shown in *Table 9-5*.

Property	Frequency sampling method	McClellan Transformation Method
<b>Resulting design</b>	FIR	FIR or IIR
<b>Requirements</b>	Magnitude specification	1-D Magnitude specifications and a transformation function
<b>Remarks</b>	The filter accuracy depends heavily on the accuracy of frequency sampling	Better transformation functions may result in much improved error
<b>Filter order</b>	20×20	12×12
<b>Magnitude spec. error(%)</b>	7.61	22.68%

*Table 9-5: Comparison of direct design methods*

The poor performance of McClellan transformation method can be attributed to its low attenuation in the stopband. It is envisaged that with a better transformation function, McClellan transformation method will perform better. For the current design task however, frequency sampling method is chosen. The resultant filter coefficients are not shown due to the number of coefficients being too large to show here.

#### 9.3.4.2 Reduction of Filter Order Using Balanced Approximation Method

The filter design in the current form requires a 20×20 order filter. This implies that for some signal paths, the signal must go through 20+20 = 40 power splitting stages. A huge pre-amplification factor will therefore be required. A solution to this problem is to use the filter order reduction method of *Section 6*. In this case, the reduced filter is 10×10 resulting in 11+11 = 22 splitting stages which is still large, but again for example purpose, is retained.

Order	0	1	2	3	4	5	6	7	8	9	10
0	0.0015	-0.0046	0.0120	-0.0239	0.0290	-0.0258	0.0246	-0.0250	0.0262	-0.0197	0.0092
1	-0.0045	0.0098	-0.0251	0.0628	-0.0895	0.0991	-0.1155	0.1212	-0.1174	0.0816	-0.0373
2	0.0128	-0.0325	0.0702	-0.1456	0.1945	-0.2091	0.2341	-0.2318	0.2181	-0.1479	0.0706
3	-0.0273	0.0852	-0.1757	0.2946	-0.3387	0.3073	-0.2844	0.2461	-0.2319	0.1602	-0.0844
4	0.0370	-0.1279	0.2614	-0.3985	0.4165	-0.3249	0.2366	-0.1628	0.1567	-0.1155	0.0711
5	-0.0437	0.1660	-0.3410	0.4815	-0.4550	0.2844	-0.1124	0.0031	-0.0112	0.0317	-0.0394
6	0.0520	-0.2103	0.4309	-0.5751	0.5001	-0.2461	-0.0139	0.1646	-0.1466	0.0590	0.0059
7	-0.0531	0.2156	-0.4349	0.5640	-0.4755	0.2151	0.0571	-0.2204	0.2045	-0.1033	0.0123
8	0.0469	-0.1903	0.3841	-0.4995	0.4244	-0.1944	-0.0508	0.1957	-0.1731	0.0941	-0.0124
9	-0.0291	0.1171	-0.2344	0.3020	-0.2527	0.1154	0.0245	-0.1134	0.1145	-0.0534	0.0051
10	0.0117	-0.0487	0.1017	-0.1362	0.1171	-0.0529	-0.0127	0.0468	-0.0388	0.0156	0.0003

*Table 9-6: Denominator coefficients*

Order	0	1	2	3	4	5	6	7	8	9	10
0	1.0000	-3.3531	6.6740	-9.7374	11.1831	-10.3924	7.8531	-4.7539	2.2164	-0.7249	0.1283
1	-2.9261	9.8116	-19.5290	28.4926	-32.7230	30.4093	-22.9790	13.9103	-6.4854	2.1211	-0.3753
2	5.4318	-18.2134	36.2521	-52.8915	60.7444	-56.4494	42.6564	-25.8220	12.0390	-3.9375	0.6967

3	-7.5706	25.3851	-50.5266	73.7179	-84.6629	78.6767	-59.4526	35.9896	-16.7795	5.4880	-0.9710
4	8.4280	-28.2599	56.2486	-82.0662	94.2508	-87.5866	66.1854	-40.0654	18.6797	-6.1095	1.0809
5	-7.6733	25.7295	-51.2120	74.7179	-85.8115	79.7440	-60.2591	36.4779	-17.0071	5.5624	-0.9841
6	5.7326	-19.2220	38.2595	-55.8202	64.1080	-59.5752	45.0184	-27.2519	12.7057	-4.1556	0.7352
7	-3.4579	11.5948	-23.0783	33.6710	-38.6703	35.9360	-27.1553	16.4385	-7.6641	2.5067	-0.4435
8	1.6190	-5.4287	10.8053	-15.7649	18.1056	-16.8254	12.7142	-7.6965	3.5884	-1.1736	0.2076
9	-0.5366	1.7992	-3.5812	5.2250	-6.0008	5.5765	-4.2139	2.5509	-1.1893	0.3890	-0.0688
10	0.0965	-0.3236	0.6441	-0.9398	1.0793	-1.0030	0.7579	-0.4588	0.2139	-0.0700	0.0124

Table 9-7: Numerator coefficients

The filter error is found to be 6.57% which is actually less than the original design! It has therefore been shown that the application of filter order reduction did not result in any notable degradation in the performance. As can be deduced from the fact that there are two sets of filter coefficients, the resulting design is an infinite impulse response design. Another factor which must be taken into account is the fact that neither numerator nor the denominator is symmetrical. No components are duplicated therefore removing the possibility for saving of components by exploitation of symmetry.

#### 9.3.4.3 Photonic Implementation of Non-Separable Filters

As the filter structure is in form of IIR filter, the structure proposed in *Section 8* must be employed in realizing the fiber-optic filter. The filter shown in Figure 8.1 will form the subsection of the IIR structure shown in Figure 8.11. The actual filter schematic diagram is omitted as the diagram will be too cluttered to make any clear statement of the structure of the filter. The attenuator settings can be calculated by considering how many splitting stages the signal being attenuated by the coefficient module must go through. The relationship between the attenuator settings  $a_{ij}$  and the filter coefficients  $h_{ij}$  is given by

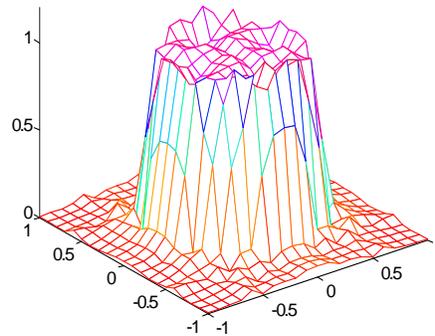
$$a_{ij} = \frac{h_{ij}}{\sqrt{2}^{k_i+2j+2}} \quad \text{Eq. 9.3.2}$$

where  $k_i$  is the order of the filter in  $n_i$  dimension. The filter statistics are given in *Table 9-8*. The filter is shown to be very inefficient compared to the separable implementations as the number of components required is very much higher. Clearly, the trade off for the good error response is the large number of components required to implement the design in practice.

Factor	2-D Direct Form
No of splitting stages per amplifier	33 (worst case)
No. of OPMs	242
No. of optical attenuators	242
No. of optical couplers	528
No. of optical Amplifiers	2
Total no. of components	1014
Remarks	Very small error

*Table 9-8: Direct structure filter statistics*

The filter magnitude response is shown below in *Figure 9.6*. Although the passband contains some irregularities, the phase response of IIR filters designed using balanced approximations remains approximately linear (refer to *Section 6*) therefore satisfying the phase requirement of the specification.



*Figure 9.6: Normalized filter magnitude response*

### **9.3.5 Comparison of Matrix Decomposition Method Design and Direct Method Design**

Although we have started out with the exactly the same specification, the two design results are strikingly different. Direct method results in a filter with over 1000 elements whereas matrix decomposition method results in a filter with around 200 elements. The error is also different. Various factors such as performance and economy must be taken into account when deciding which structure to implement. The table below shows the comparison of the two methods using the results obtained for this example. For the

example, the better implementation technique would be matrix decomposition method as it offers a filter of reasonable performance with smaller number of components.

Property	Direct design method	Matrix decomposition method
<b>No of splitting stages per OA</b>	33(worst case)	19
<b>No of OAs</b>	2	4
<b>No of attenuators</b>	242	64
<b>No of couplers</b>	528	142
<b>No of phase modulators</b>	242	64
<b>Total no of components</b>	1014	210
<b>Error(%)</b>	6.57	8.27
<b>Remark</b>	Very good transition band and low error	Economical

*Table 9-9: Comparison of direct method and matrix decomposition method*

#### **9.4 Possible Areas of Applications**

The fiber-optic and integrated photonic signal processing has found applications in areas such as optical matrix multiplications, convolutions and broadband signal processing, time division multiplexing in the femtosecond time resolution, high order optical correlation, higher order spectrum analysis for amplitude and phase, spatial and time photonic Fourier transformation. Because of the wideband properties of optical fibers and integrated photonic waveguides, PSP technique makes a natural processing architecture for fiber transmission systems carrying wide bandwidth data. An application of 2-D fiber-optic signal processing system would be the signal processing of ultra-fast TDM information that is the photonic packet switching technique. Novel time division coding used as photonic headers is also another potential application of M-D PSP. 2-D fiber-optic signal processing architecture can be therefore a natural candidate for the processing of ultra-fast signals transmitted over optical guided media.

Further capabilities of 2-D fiber-optic signal processing can be explored by considering multiplexing of signals by some form of wavelength division multiplexing. By transmitting the entire frame of the ultra-fast signals at once, it becomes possible to process the signal without using delays and by just using optical attenuators and OAs. However there are many practicalities to be overcome in realizing such as system and further works is required.

## 9.5 Concluding remarks

---

The objective of the research in this report is to explore the possible ways of realizing a 2-D signal processing system using fiber-optic signal processing architecture. A general technique for designing a 2-D filter is illustrated and numerous examples of utilization of the technique are given. Although the discussion is focused on fiber-optic systems, the design procedure for 2-D filters are just as applicable to any other signal processing architectures. For example, the 2-D filter order reduction method given in *Section 6* can be used to simplify 2-D lightwaves systems which may or may not be fiber-optic systems.

The design of 2-D filters is classified into two different classes. One class used matrix decomposition to reduce the design of 2-D filters into a set of 1-D filter design procedures. The other class used direct extensions of 1-D filter design methods. It is found that neither has a distinctive superiority over another and that the designer has to choose what is the best for the particular application, most likely by designing both and comparing the performances. All of the design procedures are implemented using the MATLAB™ programming language.

Among the matrix decomposition methods, the multiple stage singular value decomposition method of *Section 6.2* performed the best whereas for direct methods, frequency sampling method produced filters with smallest errors. However, the result should be taken with caution as there are many factors to be considered before declaring one method superior over another. The differences between the various methods are outlined in *Table 9-2* and *Table 9-5*.

A 2-D Filter order reduction method is applied to make fiber- and integrated optic signal processing more feasible. The technology allows the filter designer to produce filters of orders that are implementable in practice without sacrifices in performance.

Different possible filter structures are proposed and illustrated for photonic implementation of 2-D filters. Most of the filter structures discussed can be used in 1-D coherent fiber-optic signal processing and are not limited to 2-D coherent fiber-optic

signal processing. Some of the proposed structures such as transversal structure are extremely efficient in the number of components used to achieve a certain performance requirement. To make the efficient structures possible, the phase shifting property of optical couplers when the incoming lightwaves is cross-coupled is utilized. Filter structures for FIR and IIR filters are also shown and examples are given in *Section 8.10*.

It is evident that the fiber-optic signal processing technology presents a new direction in the usage of optical fiber, lasers, and photonics technologies which are evolving very fast. In [4] an incoherent signal processing system operating at 100 MHz is demonstrated. The authors note that the raising this capability to over 10 GHz is a relatively straightforward procedure involving shorter fiber lengths and lasers and detectors with faster rise and fall time. They also note that although conventional digital signal processing and analog signal processing techniques are limited in their usefulness for signal bandwidths exceeding one or two GHz. Current research efforts on fiber-optic signal processing on lightwaves of millimeter wavelength region will allow signal processing at bandwidths of up to 100GHz even to THz region if parametric amplification is employed. The field of 2-D signal processing which requires ultra-fast processing capability has a great deal to gain from the usage of the high speed processing capability of fiber-optic architectures. In particular especially with the fast pace of research and inventions of photonic circuits reaching the nano-scale employing photonic crystal wave guiding techniques will allow multi-dimensional processing in the photonic domain flourishing in the near future.

## References

- [1] Ngo N.Q. and L.N. Binh, “*Programmable incoherent Newton-Cotes Optical Integrator*”, *Optics Communications*, full paper, vol. 119, 1995, pp. 390-402.
- [2] H. F. Taylor, “*Application of Guided-Wave Optics in Signal Processing and Sensing*,” *Proc. IEEE*, vol 75, No 11, pp1524-1535, November 1987
- [3] See for example B. Moslehi, J. W. Goodman, M. Tur, and H. J. Shaw, “Fiber-Optic Lattice Signal Processing,” *Proc. IEEE*, Vol 72 No 7, July 1984;
- [4] C.K. Marsden and J.H. Zhao “*Optical filter analysis and design: A digital signal processing approach*”, J Wiley, NY 1999.
- [5] J. Capmany, and J. Cascon, “Optical programmable transversal filters using fiber-amplifiers,” *Electronic Letters*, vol 28, pp 1245-1246, 1992

- [6] B. Moslehi, "Fiber-optic filters employing optical amplifiers to provide design flexibility," *Electronic Letters*, Vol 28, pp 226-228, 1992
- [7] E. Heyde and R. A. Minasian, "Photonic signal processing of microwave signals using an active fibre Bragg grating pair" *IEEE Trans. Microwave Theory tech* , pp 1463 -1466, 1997.
- [8] F.T.S. Yu and I.C. Khoo, *Principles of Optical Engineering*, Section 9.
- [9] L. N. Binh, N. Q. Ngo, and S. F. Luk, "Graphical representation and analysis of the Z-shaped double-coupler optical resonator," *IEEE J. Lightwave Technol.*, vol 11, pp 1782-1792, 1993
- [10] J. S. Lim, *Two-Dimensional Signal and Image Processing*, Englewood Cliffs, NJ, Prentice Hall, 1990
- [11] Binh L.N., "Generalized analysis of a double-coupler doubly-resonant optical circuit", *IEE Proceedings Optoelectronics*, vol. 142, Dec.1995, pp. 296-304.
- [12] S. Kung, B. C. Levy, M. Morf, T. Kailath, "New results in 2-D systems theory, Part II: 2-D state-space models-realization and the notions of controllability, observability, and minimality," *Proc. IEEE*, VOL. 65, NO. 6, June 1977, pp 945-961
- [13] R. Mersereau, W. F. G. Mecklenbrauker, and T.F. Quatieri, Jr., "McClellan Transformations for Two-Dimensional Digital Filtering: I-Design", *IEEE Trans. Circuits and Systems.*, VOL. CAS-23, No. 7, July 1976, pp 405-422
- [14] A. Antoniou and W Lu, "Design of two-dimensional digital filters by using the singular value decomposition," *IEEE Trans. Circ. Systems*, VOL. CAS-34, NO. 10, Oct 1987, pp1191-1198
- [15] Z-J Mou, "Efficient 2-D and Multi-D Symmetric FIR Filter Structures," *IEEE Trans. Signal Procesing*, Vol. 42, No. 3, March 1994
- [16] S. Mitra, A. Sagar, and N Pendergrass, "Realizations of two-dimensional recursive digital filters," *IEEE Trans. Circuits Syst. Cas-22(3)*, 1975, pp177-184
- [17] R. E. Twogood, S. K. Mitra, "Computer-aided design of separable two-dimensional digital filters," *IEEE Trans. Acoustic Speech. Signal Proc.*, VOL ASSP-25, NO. 2, April 1977, pp 165-169
- [18] T.B. Deng, M. Kawamata, and T. Higuchi, "Design of 2-D digital filters based on the optimal decomposition of magnitude specifications," *IEEE J. Circuits, Systems, and Computers*, Vol. 3, No. 3, 1993.
- [19] T Deng, M. Kawamata, "Frequency-Domain Design of 2-D digital filters using the iterative singular value decomposition," *IEEE Trans. Circ. Systems*, Vol.. 38, NO.10, Oct 1991, pp 1225-1228.
- [20] A.K. Shaw and P. Mitra, "An exact realization of 2-D IIR filters using separable 1-D modules," *IEEE Trans. Circuits Syst.*, vol. 37, 1990.
- [21] W. Lu, H. Wang, and A. Antoniou, "Design of two-dimensional digital filters using singular-value decomposition and balanced approximation method," *IEEE Trans. Sig. Proc.*, VOL. 39, No. 10, Oct 1991, pp 2253-2262.
- [22] R.P. Roesser, "A discrete state-space model for linear image processing," *IEEE Trans. Automatic Control*, Vol.AC-20, No.1, Feb 1975, pp 1-10.
- [23] E. Fornasini and G. Marchesini, "State-space realization theory of two-dimensional filters," *IEEE Trans. on Automatic Control*, VOL. AC-21, NO. 4, August 1976, pp 484-492.

- [24] A.J. Laub, “*Computation of ‘balancing’ transformations,*” Proc. 1980 Joint Ameri. Contr. Conf., FA8-E, 1980
- [25] J.S. Lim, *Two-Dimensional Signal and Image Processing*, Englewood Cliffs, NJ, Prentice Hall, 1990
- [26] LN Binh and D. Trowwer “*OPTMASON*”: *A program for AUTOMATIC DERIVATION of the optical transfer functions of PHOTONIC CIRCUITS from their connection graphs*, Report No.MECSE-34-2004, 2004, <http://www.ds.eng.monash.edu.au/techrep/reports/post2003index.html>.

## INDEX

---

### 2

- 2-D difference Eq., 24
- 2-D direct structure filter, 82
- 2-D frequency sampling method, 36
- 2-D separable structure Filter, 84
- 2-D transfer function description, 24
- 2-D windowing method, 39

---

### B

- Balanced approximation, 62
- Balancing transformation, 64
- Binary tree structure, 86

---

### C

- Coefficient, 72

---

### D

- Delay, 72
- Discrete-time filter, 72

---

### F

- Frequency filtering, 14
- Frequency sampling method, 31

---

### I

- Impulse response method, 31
- Iterative singular value decomposition, 53

---

### L

- Linear phase, 24
- Linearity, 34
- LU decomposition, 45

---

### M

- McClellan transformation method, 39
- Minimal square error approximation property, 47

---

### N

- Normalized frequency, 18
- Nyquist rate, 17

---

***O***

Observability gramians, 63

Optimal decomposition, 56

Order, 35

---

***R***

Reachability grammians, 63

---

***S***

Separable operation, 20

Separable sequences, 19

Simpson's rule, 21

Singular value decomposition, 46

Spatial frequency, 14

Spatial signal, 14

Splitter, 72

Stability, 21, 35

Summer, 72

---

***T***

Transformation method, 31

Transversal filter structure, 88

---

***W***

Window method, 31