2010

An Efficient Design of 2-D Digital Filters Using Singular Value Decomposition and Genetic Algorithm with Canonical Signed Digit (CSD) Coefficients

Bashier Elkarami
University of Windsor

An Efficient Design of 2-D Digital Filters Using Singular Value Decomposition and Genetic Algorithm with Canonical Signed Digit (CSD) Coefficients.

by

BASHIER ELKARAMI

A Thesis

Submitted to the Faculty of Graduate Studies

through Electrical and Computer Engineering

in Partial Fulfillment of the Requirements for

the Degree of Master of Applied Science at the

University of Windsor

Windsor, Ontario, Canada

2010

Your file Votre référence
ISBN: 978-0-494-80231-1
Our file Notre référence
ISBN: 978-0-494-80231-1

DECLARATION OF ORIGINALITY

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

In this thesis, the design of 2-D filters by SVD is proposed. This technique reduces the complexity of the designed 2-D digital filters by decomposing it into a set of 1-D digital filters in z1 and z2 connected in cascade.

The design by SVD can be improved by varying the order of 1-D digital filters in each section based on their corresponding singular values. It is shown that by assigning higher order filters to the sections with greater singular values (SVs), and lower order filters to the sections with lower SVs, a sizable reduction in the total number of required multiplications is achieved.

A Genetic Algorithm (GA) is used to design each of the 1-D filters instead of classical optimization.

Canonical signed digit system is used to represent filters' coefficients. CSD helps to improve the efficiency of multiplications and thus increase the throughput rate.

Examples are provided to demonstrate the effectiveness and usefulness of the proposed technique.

DEDICATION

To my family

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to Dr. M. Ahmadi for his guidness through out my thesis. His advices, patience, and encouragment helped me to successfully complete my research.

I thank my wife, parent, and brothers for their support and help.

I thank my friends for their support during my difficult times.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1 Introduction

Digital signal processing is one of the most powerful techniques concerned with transformation and manipulation of signals and their information. Signals are represented as sequence of numbers and they are processed by using digital means in accordance with specific computational algorithms that can be implemented on computers. Since the implementation cost of DSP has decreased significantly, the applications of DSP have vastly increased in many diverse fields. DSP can be found in mobile phones, multimedia computers, video recorders, CD players, hard disk driver controllers, and modems. Digital signal processing can deal with one dimensional signals or multidimensional signals represented as multidimensional arrays such as sampled images. The processing of 1-D DSP and 2-D DSP are conceptually similar where many operations performed on 1-D DSP are the same as those performed on 2-D DSP. Except that the amount of data involved in 2-D DSP are higher than these in 1-D DSP.

Digital filters are a very important part of DSP. They are used either to separate combined signals or to restore signals that have been distorted. Digital filters may be implemented in software by general purpose computer or implemented in hardware on DSP chip or ASIC processor. Digital filters can be in one dimensional or n-dimensional filters where $n \geq 2$. All of them have been utilized in many fields, for example 1-D digital filters are used in Biomedical engineering and speech communications. On the other hand, 2-D digital filters have been used in X-ray, satellite weather, and TV transmission [1, 2, 3, and 4]. To reduce the power consumption and implementation cost of digital filters, the complexity of multipliers has to be reduced. One way of doing that is by using CSD representation system to represent filter coefficients which reduces the number of partial products being summed [5, 6, and 7]. Generally, digital filters are divided into two basic types: Recursive digital filters known as Infinite impulse response IIR and Finite impulse response nonrecursive digital filters [8, 9, 10, 11, and 12].

1.2 Characterization of 1-D FIR and IIR filters

1.2.1 1-D FIR filters

In FIR digital filters, one whose impulse response is of finite duration, the current output, $y(n)$, is calculated from the current and previous input value as shown bellow;

$$y(n) = \sum_{i=0}^{N} a_i x(n-i) \qquad (1.1)$$

The z-transfer function of FIR of order N is given as the following:

$$H(z) = \sum_{i=0}^{N} a_i z^{-i} \qquad (1.2)$$

The magnitude and phase response of the filter at a particular frequency ω is given by replacing Z with $e^{j\omega T}$ in the transfer function.

$$H(z) = \sum_{i=0}^{N} a_i e^{-j\omega i T} \qquad (1.3)$$

a_i are the filter coefficients.

1.2.2 1- D IIR filters

In IIR digital filters, whose impulse response continues forever, the current output ,y(n), is calculated from the current and previous input values as well as the previous output and therefore,

$$y(n) = \sum_{i=0}^{N} b_i x(n-i) - \sum_{j=0}^{M} a_j x(n-j) \qquad (1.4)$$

If $M = N$ the z-transfer function is:

$$H(z) = \frac{\sum_{i=0}^{N} a_i z^{-i}}{\sum_{j=0}^{N} b_j z^{-j}} = \frac{A(z)}{B(z)} \qquad (1.5)$$

$B(z) \neq 0 \ and \quad |z| \geq 1$

The magnitude and phase of the filter at a particular frequency ω is given by replacing Z with $e^{j\omega T}$ in the transfer function.

$$H(z) = \frac{\sum_{i=0}^{N} a_i \ e^{-j\omega i T}}{\sum_{j=0}^{N} b_j \ e^{-j\omega j T}} \qquad (1.6)$$

a_i *and* b_i *are filter coefficients*

1.3 Design Technique for 1-D FIR filters

One of the most common design methods for 1-D FIR is the Fourier series in conjunction with windows. The most common used windows are Hann, Hamming, and Kaiser [2, 12]. Linear and nonlinear programming is another method which includes optimization technique such as least square error.

1.4 Design Technique for 1-D IIR filters

The most traditional method is transforming from analog filters such as Butterworth, Chebysheve, elliptical, and Bessel filters, to digital filters [13]. The most commonly used transformation is the Bilinear transformation. Similar to 1-D FIR, optimization technique can be used to design IIR filters.

Stability is a major concern in designing digital filters. Unstable filters will produce output that is so large to become not possible to compute. The most commonly used stability criterion is the bounded-input the bounded output rule (BIBO). A system is stable in BIBO sense if for any bounded input sequence, the output sequence is bounded [14, 15, 17, and 18]. IIR filters are cheaper to implement and have faster response time comparing to FIR filters because they require less multiplications than FIR filters. These advantages come from the fact that the feedback loop is used to modify the input with the weighted samples of previous output. This yields an infinite impulse response with the requirement of only a finite number of computational steps per output point. However, the feedback loop causes the inherent stability problem. On other hand, FIR generates the output from weighted samples of input only which makes FIR filters always stable but require higher order and vast amount of computations comparing to IIR to satisfy the same specifications.

1.5 Two Dimensional Digital Filters

2-D digital filters are computational algorithms that transform a 2-D input sequence of numbers into a 2-D output sequence according to pre-specified rules. Similar to 1-D digital filters, 2-D digital filters can be divided to two basic types: Finite impulse response (FIR) and Infinite impulse response (IIR). The objective of 2-D digital filters is usually being either enhancement of an image to make it more acceptable to human eye, or removal of the effect of some degradation mechanisms, or separation of features to facilitate identifications and measurement by human or machine.

1.6 Applications of 2-D digital filters

2-D digital filters have wide applications in the field of image processing such as video coding, medical image, enhancement and analysis of area photographs and analysis of satellite weather photos. 2-D digital filters are used in HDTV to help reducing video noise, create higher picture clarity and bright color transitions in each video frame. 2-D digital filters can improve picture resolution and minimize distortion by separating chrominance from luminance in the video signals. 2-D digital filters are used in x-ray to remove noise from image data for more accuracy. 2-D digital filters play important role to improve seismic discontinuity data for interpretation by separating the reflection from the surface and reflection from ground surface [19, 20]. 2-D digital filters remove geometric distortion and radiometric errors from the sonar image in order to analyze the image [21]. Also, 2-D digital filters remove the noise comes with real data because of vehicle instability or in the water [22]. 2-D digital filters are used to remove interference in radio astronomy signals so they can obtain much clearer image of the stars and galaxies [23]. 2-D digital filters are used for efficient image encoding specially when digital transmissions and storage of image is needed and the amount of bits required is huge like in broadcast TV teleconference. 2-D digital filters are used in satellite images to getred of various image degradation occure because of random noise, interference geographical distortion.

1.7 Characterization of 2-D FIR and IIR filters

1.7.1 2-D FIR filters

Similar to 1-D FIR filters, 2-D FIR can be written as

$$y(m,n)=\sum_{i=0}^{N}\sum_{j=0}^{N}a(i,j)\,x(m-i,n-j) \qquad (1.7)$$

The z-transfer function of 2-D FIR digital filters is as follows:

$$H(z_1,z_2)=\sum_{i=0}^{N}\sum_{j=0}^{N}a(i,j)\,z_1^{-i}z_2^{-j} \qquad (1.8)$$

Where $a(i,j)$ is the filter coefficients and N is the filter order. The corresponding frequency response is obtained by substituting $z_1=e^{j\omega_1 T}$ and $z_2=e^{j\omega_2 T}$

$$H(z_1,z_2)=\sum_{i=0}^{N}\sum_{j=0}^{N}a(i,j)\ e^{-ji\omega_1 T}e^{-ji\omega_2 T} \tag{1.9}$$

1.7.2 2-D IIR filters

Similar to 1-D IIR filters, 2-D IIR can be written as

$$y(m,n)=\sum_{i=0}^{N}\sum_{j=0}^{N}a(i,j)x(m-i,n-j)-\sum_{i=0}^{N}\sum_{j=0}^{N}b(i,j)x(m-i,n-j) \tag{1.10}$$

The z-transfer function of 2-D IIR digital filters is as follows:

$$H(z_1,z_2)=\frac{\sum_{i=0}^{N}\sum_{j=0}^{N}a(i,j)\ z_1^{-i}z_2^{-j}}{\sum_{i=0}^{N}\sum_{j=0}^{N}b(i,j)\ z_1^{-i}z_2^{-j}} \tag{1.11}$$

$$B(z_1,z_2)\neq 0 \quad \bigcap_{i=1}^{2} |z_i|\geq 1$$

And the frequency response is:

$$H(z_1,z_2)=\frac{\sum_{i=0}^{N}\sum_{j=0}^{N}a(i,j)e^{-ji\omega_1 T}e^{-ji\omega_2 T}}{\sum_{i=0}^{N}\sum_{j=0}^{N}b(i,j)e^{-ji\omega_1 T}e^{-ji\omega_2 T}} \tag{1.12}$$

Where a (i , j) and b (i , j) are the filter coefficients and N is the filter order.

There are some subclass IIR filters that should be introduced as follows:

1.7.3 Separable product 2-D IIR filters

The filter of this sub-class is a cascade of 1-D filters [24, 25, 26, and 27].The advantages of this sub-class are reducing the problem from 2- D to 1-D, and reducing the stability problem to 1-D. However, the shape of the cutoff boundary has to be rectangular.

$$H(z_1,z_2) = H_1(z_1)\, H_2(z_2)$$

$$H_k(z_k) = \frac{\sum_{i=0}^{N} a_k\, z_k^{-i}}{\sum_{j=0}^{N} b_k\, z_k^{-j}} \tag{1.13}$$

1.7.4 Separable denominator non-separable numerator 2-D IIR filters

This filter has the advantages of separable product filters but not their disadvantages. Circular symmetric, quadrantal, and octagonal symmetric magnitude response is achievable.

$$H(z_1,z_2) = \frac{\sum_{i=0}^{N}\sum_{j=0}^{N} a(i,j)\, z_1^{-i} z_2^{-j}}{\left(\sum_{i=0}^{N} b_{1_i}(i,j)\, z_1^{-i}\right)\left(\sum_{j=0}^{N} b_{2_j}(i,j)\, z_2^{-j}\right)} \tag{1.14}$$

1.7.5 Symmetry property of 2-D filters with separable denominator non-separable numerator.

Some filters have some desirable properties as follows:

i-Central Symmetry

In this sub-class, the magnitude response is equal in first and third quadrants and in second and fourth. [28, 29, 30, 31]

ii-Quadrantal Symmetric

In this sub-class, the magnitude response is equal in all four quadrants [28,32].

$$a_{1j} = a_{m-1j} = a_{1m-j} = a_{m-1m-j}$$
$$b_{1j} = b_{2j}$$

iii-Octagonal Symmetric

It is realized when

$$a_{ij} = a_{ji}$$

Filter class	Multiplications	Reduction
General class	$2(m+1)^2$	0
Central class	$(m+1)(m+3)$	M^2-1
Quadrantal class	$M^2/4+3m+3$ M^2/4+3m+3	$7M^2/4+m-1$
Octagonal class	$(M^2+22m+24)/8$	$(15M^2+10m-8)/8$

Table (1.1) shows how much reduction we can get of each class

1.8 Design techniques

1.8.1 2-D FIR digital filters

One of the easiest methods is Window technique in conjunction with Fourier series which can be used to design filters of any order but it is not optimal in any sense [33]. The window technique could be done directly in two dimensions or by designing 1-D filters then transforming them to 2-D filters using one of transformation techniques. The transformation techniques transform from 1-D filters to 2-D filters and it could be analog to analog, digital to digital or analog to digital. One of the most powerful transformations is McClellan [3, 34, and 35]. The frequency sampling technique uses to design high order filtersbut its design time is excessive. One of the most powerful techniques is the optimization technique which depends on minimizing the error between the ideal filter and the designed filter.

1.8.2 2-D IIR digital filters

Frequency – domain technique offers flexibility in design and implementation. The optimization technique depends on minimizing the error between the ideal filter and the designed filter [5]. The transformation technique is the simplest and fastest technique, though filters designed by it are not optimum. One of the most power transformations is double bilinear. Separable technique is simple and economical to implement because it reduces the design problem from 2-D to 1-D filters. Filters designed by this technique do not satisfy the specifications as closely as the non-separable filters.

Two difficulties are associated with all of previous design techniques: first, the stability is difficult to guarantee in two dimensions. Second, the design in two dimensions requires a vast amount of computations which makes the design very complex. In this thesis, we solve these problems by using singular value decomposition. The design by SVD reduces the design problem from two dimensional to one dimensional. In this way, we can overcome these problems easily because in 1-D filters the stability is easy to guarantee and the computational amount is much less in 1-D filters. To get better result, we use the combination of Genetic Algorithm and canonical signed digit to design 1-D filters. Genetic algorithm is more effective in escaping the local optimum than classical optimization algorithms. By using CSD to represent filters coefficients, we reduce the number of non zero digits which reduces the multipliers and as result reduce complicity and hardware cost.

1.9 Number Systems

In digital computer system, different number system may represent the same value of data in different form. Number systems can be classified into three types; weighted number system, unweighted number system, and the homomorphic number system. Weighted number system can be represented in fixed point or floating point based format. This type includes decimal, binary, octal, hexadecimal, one's complement, and two's complement. Unweighted number system is based on using different radix to scale a number like in residue number system. Homomorphic number system is rooted from the idea of abstract algebra. Fixed point and floating point of weighted number system are commonly used in designing digital filters and the following is the brief description of some of them.

1.9.1 Two's complement system

This system uses the left most bit to determine whether the number is positive or negative. For positive number, zero is assigned to the left most bits. For negative number, one is assigned to the left most bit. To convert to two's complement from decimal, first convert to binary then complement all bits then add one.

1.9.2 Signed Magnitude Number System

Similar to two's complement system, this system uses the left most bit for determining positivity or negativity of a number

$$A = \sum_{i=0}^{N-1} b_i \times 2^{i-1} \qquad (1.15)$$

1.9.3 Signed digit number system

In this system, each bit could take 1,-1, or 0

$$A = \sum_{i=0}^{N-1} b_i \times 2^{-i} \qquad (1.16)$$

1.9.4 Canonical Signed Digit Coefficients

The canonical signed-digit (CSD) number system is based on the signed digit number system which allows individual digits to have a sign as well as a value.

$$Digit = \left\{ -\left\lfloor \frac{r}{2} \right\rfloor, .,., -1, 0, 1, .,., \left\lfloor \frac{r}{2} \right\rfloor \right\} \qquad (1.17)$$

The ternary number system where r=2 is used which allows the digits to have values of $1, 0, \overline{1}$. Where $\overline{1}$ denotes -1.

Therefore, CSD system is considered as an extension of the ternary number system. The signed digit number system is considered a redundant number system because for a given value may be represented by more than one sequence of digits. For example,

$$0.01 = 1 \times 2^{-2} = 0.25$$
$$and$$
$$0.\bar{1}1 = -1 \times 2^{-1} + 1 \times 2^{-2} = -0.25$$

0.01 and $0.\bar{1}1$ are two different representations with the same value. However, for any given value of two or more redundant representations in signed digit system, there will be only one CSD representations. Two restrictions must be met in CSD system, first no two non-zero digit to be adjacent. Second, there is a limit on the number of non-zero digits that may be presented in a CSD number. A number in CSD number system is represented as a sum and difference of power of two as the following:

$$x = \sum_{k=1}^{L} S_k \times 2^{-P_k} \qquad (1.18)$$

Where

$k = 1 : M$

$P_k \in \{0, 1, 2, ..., M - 1\}$

S_k is ternary digits S_k $1, 0, \bar{1}$

M is a pre-specified word length.

L is the number of non-zero digits

$S_k \times S_{k+1} = 0$ for all k

For example, the following three numbers are converted to CSD system with M=12 and L=3

$$0.8594 = 2^{\bar{0}} - 2^{-3} - 2^{-6} = 00001.00\bar{1}00\bar{1}0$$

$$1.6875 = 2^{1} - 2^{-2} - 2^{-4} = 00010.0\bar{1}0\bar{1}000$$

$$50 = 2^{6} - 2^{-2} + 2^{1} = 10\bar{1}0010.00000$$

$$63 = 2^{6} - 2^{0} = 100000\bar{1}.00000$$

The example clearly shows that all the non-zero digits are separated by at least one zero digit.

The fewer non-zero digits give the CSD system an advantage over binary system where the fewer non-zero digits mean fewer products in multipliers. Multipliers in digital filters are performed with shifters, adders and subtractions. The multiplier repeatedly shifts one or more bit position and adds to a partial product according to the bit pattern of the multiplicand. While shift operations execute quickly, addition operations slower and comprise the bulk of multiplications time.

To demonstrate that fact, let's multiply 14 and 13 in both binary system and CSD system.

In binary system:

$$
\begin{array}{r l}
0\ 0\ 0\ 0\ 1\ 1\ 0\ 1 & 13 \\
0\ 0\ 0\ 0\ 1\ 1\ 1\ 0 & 14 \\
\hline
0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 & \\
0\ 0\ 0\ 0\ 1\ 1\ 0\ 1 & \textit{add} \\
0\ 0\ 0\ 0\ 1\ 1\ 0\ 1 & \textit{add} \\
0\ 0\ 0\ 0\ 1\ 1\ 0\ 1 & \textit{add} \\
\hline
0\ 0\ 0\ 0\ 1\ 0\ 1\ 1\ 0\ 1\ 1\ 0 & 182
\end{array}
$$

In CSD system:

$$
\begin{array}{r l}
0\ 0\ 0\ 1\ 0\ \bar{1}\ 0\ 1 & 13 \\
0\ 0\ 0\ 1\ 0\ 0\ \bar{1}\ 0 & 14 \\
\hline
0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 & \\
0\ 0\ 0\ \bar{1}\ 0\ 1\ 0\ \bar{1} & \textit{subtract} \\
0\ 0\ 0\ 1\ 0\ \bar{1}\ 0\ 1 & \text{shift 2 times} \\
\hline
0\ 0\ 0\ 0\ 1\ 0\ 1\ 1\ 0\ 1\ 1\ 0 & 182
\end{array}
$$

This example shows that the binary system needs four additions but the CSD system needs one subtraction which is the same as additions, thus reducing the complexity of multipliers. Therefore CSD system reduces the cost of designing digital filters since multiplications consume most of the power and implementations cost.

1.10 Conversion from 2's – complement number to CSD number

To convert a 2's-complement number x that has a j digit where

$X = x_1 x_2 \ldots\ldots x_j$

We start from the least significant bit LSB (the rightmost) of x to the most significant bit MSB of x and we deal with two bits at the time.

1-let $x_j = x_{j-1}$, $C_0 = 0$

2-for i=0.........j-1, let $C_{i+1} = x_i x_{i+1} + x_i C_i + x_{i+1} C_{i+1}$

Where

C_{i+1} is the carry of the step i+1

C_i is the carry of the step i

The'+' is the logical OR

3-for i=0......,j-1, let $CSD_i = x_i + C_i - 2C_{i+1}$

Example:

We convert 441 to 2's-complements then to CSD

$441 = 0110111001(2\text{'s}) = 100\bar{1}00\bar{1}001(CSD)$

Table 2 and flowchart 1 illustrate the conversion's steps

Table 3 shows comparison between 2's-comlement and CSD in non-zero digits

i	x_i	x_{i+1}	C_i	C_{i+1}	CSD_i
0	1	0	0	1	1
1	0	0	1	1	0
2	0	1	1	1	0
3	1	1	1	1	-1
4	1	1	1	1	0
5	1	0	1	1	0
6	0	1	1	1	-1
7	1	1	1	1	0
8	1	0	1	1	0
9	0	1	1	1	1

Table (1.2) Illustrate the conversion's steps of 441

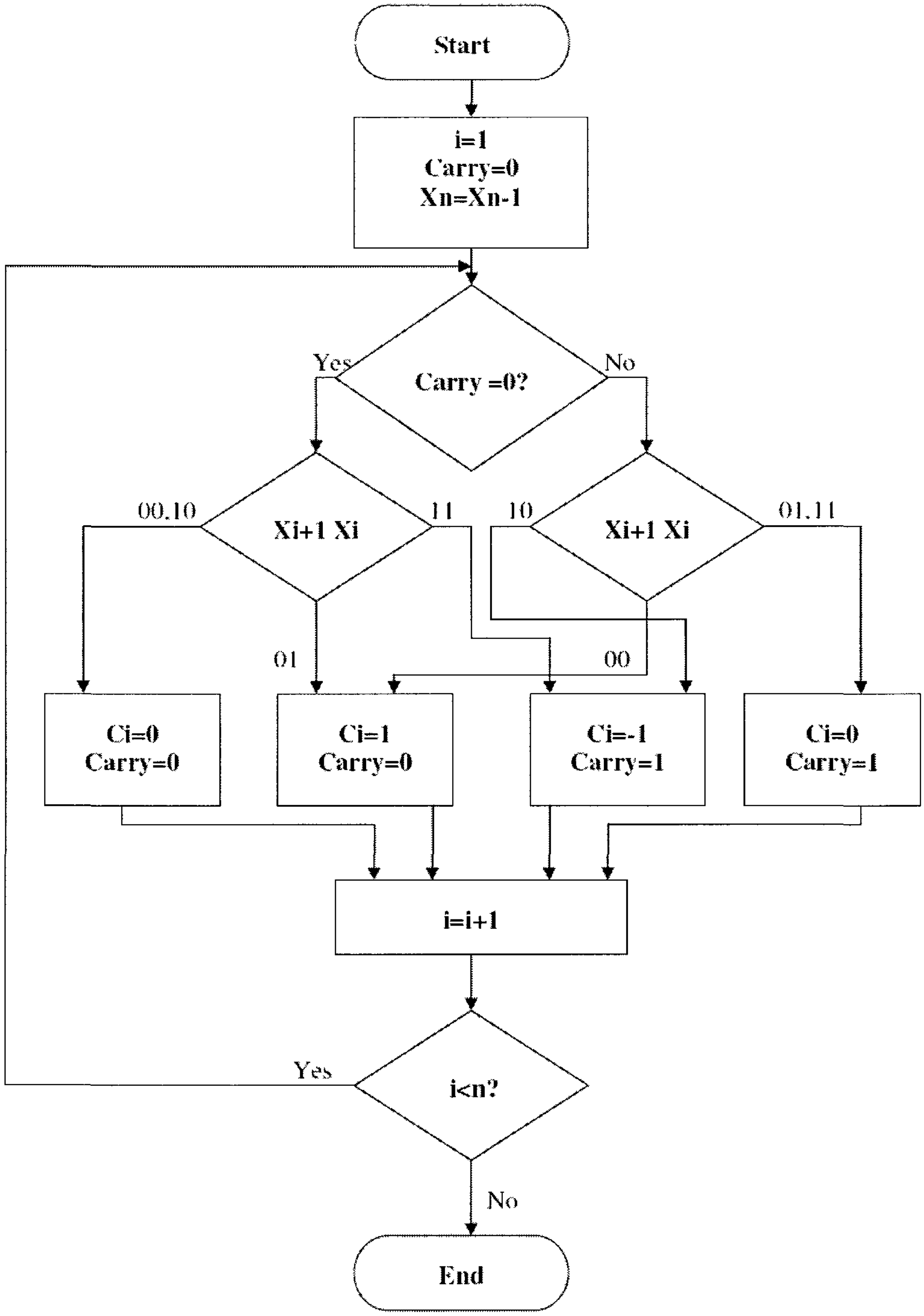


Figure (1.1).a flowchart can be used to convert 2's-complements then to CSD

To show the significant of CSD over 2's-complements, we compare between them by converting same number from decimal to 2's-complements then to CSD and calculate the non-zero digits in both of them.

DEC	2's complement	Non-zero	CSD	Non-zero	Difference
7	0111	3	$100\bar{1}$	2	1
13	01101	3	$10\bar{1}01$	3	0
14	01110	3	$100\bar{1}0$	2	1
15	01111	4	$1000\bar{1}$	2	2
63	0111111	6	$0100000\bar{1}$	2	4
-127	01111111	7	$1000000\bar{1}$	2	5
374.25	1010001001.11	6	$1010001010.0\bar{1}$	5	1
.12345	.000111111001	7	$.00100000\bar{1}001$	3	4

Table (1.3) shows comparison between 2's-comlement and CSD in non-zero digits

There is another way to convert to CSD system from 2's-complements.

In this method, each group of consecutive 1s is changed to a ternary representation from binary representation. This is done starting from the rightmost 1 and proceeding left until the last 1.

The first 1 is to be changed to -1 and the rest of 1's is to be changed to 0's, and then add 1 at the end of the group. Each group of 1's is to be modified separately from the rest of 1's.

Let's take 441=0110111001 as an example

We will start at LSB (rightmost):

The first 1 stays without modification because it has no adjusting 1

The 111 group: first change the first 1 to -1 and the rest to zeros. After that we place 1 to the let of the original sequence.

$$011\ \ 0\ \ 111\ \ 001$$

$$011\ \ 1\ \ 00\bar{1}\ \ 001$$

Then repeat the same steps to the other groups of 1's

$$0111\ \ \ 00\bar{1}001$$

$$100\bar{1}\ \ \ 00\bar{1}001$$

$$100\bar{1}00\bar{1}001$$

And this is the same number we had before.

1.11 Organization of this thesis

In chapter two, we will provide a brief background on SVD. In chapter three, introduction to GA with details. Chapter four, we provide details on designing 2-D digital filters by using singular value decomposition and an improvement that can be made to SVD to get more efficient design. In chapter five, we introduce genetic algorithm with details. In chapter five, we provide examples on designing 2-D digital filters by SVD. In chapter six, provide examples on designing 2-D digital filters by combining genetic algorithm with SVD and CSD. Chapter seven is the conclusion.

CHAPTER II

SINGULAR VALUE DECOMPOSITION

2.1 Introduction to Singular Value Decomposition (SVD)

Singular Value Decomposition (SVD) is a matrix factorization technique which decomposes an m × n matrix A, with rank r, into three orthogonal matrices

$$SVD(A) = U_{mxm} S_{mxn} V'_{nxn} \qquad (2.1)$$

U and V are orthogonal matrices because the columns of U are orthogonal to each other and the rows of V' are orthogonal as well. The matrix S is a diagonal matrix containing non-negative singular values in descending order.

$$S = diag\{\sigma_1, \sigma_2, ..., \sigma_r\}$$
$$where$$
$$\sigma_1 \geq \sigma_2 \geq ...... \geq \sigma_r > 0$$

The columns of U are called the left singular vectors of A while the columns of V are called the right singular vectors of A. if $U = [u_1, u_2, ..., u_m]$ and $V = [v_1, v_2, ..., v_n]$

$$AA' = U \begin{bmatrix} S^2 & 0 \\ 0 & 0 \end{bmatrix}_{m \times m} U'$$

and

$$AA' = V \begin{bmatrix} S^2 & 0 \\ 0 & 0 \end{bmatrix}_{n \times n} V'$$

Therefore, the singular values of A are the positive square roots of the nonzero eigenvalues of AA' *or* $A'A$. The left singular vector of A (u_i) is the eigenvector of AA', and the right singular vector of A (v_i) is the eigenvector of $A'A$.

SVD has some applications as following:

1-Find the L_2 norm and Frobenius norm of a matrix

$$\|A\|_2 = \sigma_1$$
$$and \qquad (2.2)$$
$$\|A\|_F = \left(\sum_{i=1}^{r} \sigma_i^2 \right)^{1/2}$$

2-The condition number of a nonsingular matrix $A \in C^{n \times n}$ is defined as

$$cond(A) = \|A\|_2 \|A^{-1}\|_2 = \frac{\sigma_1}{\sigma_n} \qquad (2.3)$$

3-The range and null space of a matrix $A \in C^{m \times n}$ of rank *r* assume the forms

$$\Re(A) = span\{u_1, u_2, \ldots.., u_r\}$$
$$N(A) = span\{v_{r+1}, v_{r+2}, \ldots.., v_n\}$$

4-Compute Moore-Penrose pseudo-inverse:

The Moore-Penrose pseudo-inverse of a matrix $A \in C^{m \times n}$ is defined as the matrix $A^{+} \in C^{n \times m}$ that satisfies the following four conditions

(i) $AA^{+}A = A$

(ii) $A^{+}AA^{+} = A^{+}$

(iii) $(AA^{+})^{H} = AA^{+}$

(iv) $(A^{+}A)^{H} = A^{+}A$

The Moore-Penrose pseudo-inverse of **A** can be obtained using SVD as

$$A^{+} = U \;\; S^{+} \;\; V^{H}$$

$$S^{+} = \begin{bmatrix} S^{-1} & 0 \\ 0 & 0 \end{bmatrix}_{n \times m}$$

$$S^{-1} = diag\left\{\sigma_1^{-1}, \sigma_2^{-1}, \ldots, \sigma_r^{-1}\right\}$$

So we have

$$A^{+} = \sum_{i=1}^{r} \frac{v_i u_i^H}{\sigma_i} \qquad (2.4)$$

2.2 Example

Compute the SVD for the following matrix:

$$A = \begin{pmatrix} 4 & 0 \\ 3 & -5 \end{pmatrix}$$

Step 1 Compute A' and A'A

$$A' = \begin{pmatrix} 4 & 3 \\ 0 & -5 \end{pmatrix}$$

$$A'A = \begin{pmatrix} 4 & 3 \\ 0 & -5 \end{pmatrix} \begin{pmatrix} 4 & 0 \\ 3 & -5 \end{pmatrix}$$

$$A'A = \begin{pmatrix} 25 & -15 \\ -15 & 25 \end{pmatrix}$$

Step 2 Determine the eigenvalues of A'A in descending order and in absolute sense. Then square roots these to obtain the singular values of A.

$$A'A - cI = \begin{pmatrix} 25-c & -15 \\ -15 & 25-c \end{pmatrix}$$

$$|A'A - cI| = (25-c)(25-c) - (-15)(-15) = 0$$

$$c^2 - 50c + 400 = 0$$

Now, we compute the singular values as following:

$$c_1 = 40$$

$$c_2 = 10$$

$$S_1 = \sqrt{40} = 6.3245$$

$$S_2 = \sqrt{10} = 3.1622$$

Step 3

Construct diagonal matrix S.

$$S = \begin{pmatrix} 4.16 & 0 \\ 0 & 1.924 \end{pmatrix}$$

$$S^{-1} = \begin{pmatrix} 6.3245 & 0 \\ 0 & 3.1622 \end{pmatrix}$$

Step 4

Compute the eigenvectors of A'A, and then place these eigenvectors along the columns of V and compute V'.

$for\ c_1 = 40$

$$A'A - cI = \begin{pmatrix} 25-40 & -15 \\ -15 & 25-40 \end{pmatrix} = \begin{pmatrix} -15 & -15 \\ -15 & -15 \end{pmatrix}$$

$$(A'A - cI)x_1 = 0$$

$$\begin{pmatrix} -15 & -15 \\ -15 & -15 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$-15x_1 - 15x_2 = 0$$

$$-15x_1 - 15x_2 = 0$$

$let \quad x_2 = -x_1$

$$X_1 = \begin{pmatrix} x_1 \\ -x_1 \end{pmatrix}$$

Dividing by its length.

$$L = \sqrt{x_1^2 + x_2^2} = x_1\sqrt{2}$$

$$X_1 = \begin{pmatrix} x_1/L \\ -x_1/L \end{pmatrix} = \begin{pmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{pmatrix} = \begin{pmatrix} 0.7071 \\ -0.7071 \end{pmatrix}$$

$for\ c_1 = 10$

$$A'A - cI = \begin{pmatrix} 25-10 & -15 \\ -15 & 25-10 \end{pmatrix} = \begin{pmatrix} 15 & -15 \\ -15 & 15 \end{pmatrix}$$

$$(A'A - cI)x_1 = 0$$

$$\begin{pmatrix} 15 & -15 \\ -15 & 15 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$15x_1 - 15x_2 = 0$$

$$-15x_1 + 15x_2 = 0$$

$let\quad x_2 = x_1$

$$X_2 = \begin{pmatrix} x_1 \\ x_1 \end{pmatrix}$$

Dividing by its length.

$$L = \sqrt{x_1^2 + x_2^2} = x_1\sqrt{2}$$

$$X_2 = \begin{pmatrix} x_1/L \\ -x_1/L \end{pmatrix} = \begin{pmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{pmatrix} = \begin{pmatrix} 0.7071 \\ 0.7071 \end{pmatrix}$$

$$V = [X_1 \;\; X_2] = \begin{pmatrix} 0.7071 & 0.7071 \\ -0.7071 & 0.7071 \end{pmatrix}$$

$$V' = \begin{pmatrix} 0.7071 & -0.7071 \\ 0.7071 & 0.7071 \end{pmatrix}$$

Step 5

Compute U as $U = AVS^{-1}$

$$U = AVS^{-1} = \begin{pmatrix} 4 & 0 \\ 3 & -5 \end{pmatrix} \begin{pmatrix} 0.7071 & 0.7071 \\ -0.7071 & 0.7071 \end{pmatrix} \begin{pmatrix} 0.1581 & 0 \\ 0 & 0.3162 \end{pmatrix}$$

$$U = AVS^{-1} = \begin{pmatrix} 4 & 0 \\ 3 & -5 \end{pmatrix} \begin{pmatrix} 0.1118 & 0.2236 \\ -0.1118 & 0.2236 \end{pmatrix}$$

$$U = AVS^{-1} = \begin{pmatrix} 0.4472 & 0.8944 \\ 0.8944 & -0.4472 \end{pmatrix}$$

It is easy to compute SVD using Matlab function

[U, S, V]=svd (A)

U = 0.4472 0.8944

0.8944 - 0.4472

S = 6.3246 0

0 3.1623

v = 0.7071 -0.7071

0.7071 0.7071

CHAPTER III

GENETIC ALGORITHM

3.1 Introduction

Optimization algorithms have been used effectively in the design of digital filters for their abilities to provide the desired characteristics. They are fast, efficient and found to work reasonably well for designing digital filters. However, optimization problems for designing digital filters are often complex, highly nonlinear and multimodal in nature. These methods are very good in locating local minima but unfortunately, they are not designed to discard inferior local solutions in favor of better ones. Therefore, they tend to locate minima in the locale of the initialization point.

GA is stochastic search algorithm that was originally motivated by the mechanisms of natural selection, natural genetics and the principle of survival of the fittest. GA was developed by Howlland in 1975 and it was further improved by Goldberg [34]. GA provides an efficient searching capability for the optimal solution to the objective function of an optimization problem without being stuck in local minima. GA has some advantages over classical optimization and search methods:

1-Instead of operating on a single solution, GA operates on group of initial solutions in parallel, thus reducing the possibility of reaching local optimum.

2- Direct manipulation of the encoded representation of parameters, rather than the parameters themselves.

3-Use probabilistic transition rules rather than deterministic operations.

4- Does not use derivative information or other auxiliary information.

3.2 GA Cycle

GA manipulates a collection of individuals named population where each individual (known as chromosome) represents one candidate solution to the problem. In each generation, GA eliminates some individuals and only fit individuals survive to reproduce and to recombine their genetic materials to produce new individuals as offspring.

Each individual is associated with a fitness value that reflects how good it is comparing with other solutions in the population. The crossover mechanism simulates the recombination process by exchanging portions of data string between chromosomes. The mutation mechanism causes random alterations of the strings. The selection, crossover and mutation processes constitute the basic GA cycle or generation, which repeated until some predetermined criteria are reached. A basic GA cycle operates as following:

1- Initialize a population randomly.

2- Evaluate the fitness of each chromosome.

3- Apply crossover and mutation operation on selected parents to create offspring chromosomes.

4- Setup a new population of offspring chromosomes using a certain replacement strategy.

5- Check the termination criteria, if it is not reached repeat steps 2-5, otherwise stop and return the best chromosomes. Figure 3.1 illustrate the basic GA cycle.

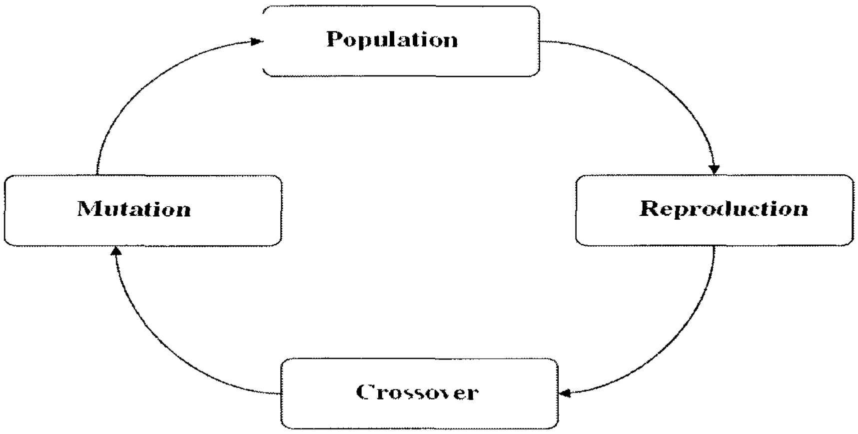


Figure (3.1) GA cycle

3.3 Population initialization

GA usually generates a random initial population P_0 where the size of population depends on the complicity of the problem to be solved. The initialization does not need to be purely random where prior knowledge of the problem domain is sometimes invoked to seed P_0 with good chromosomes. These chromosomes can be represented by binary, decimal or alphabetical data with fixed-length strings. Bit-string encoding is the most classical approach in GA because of its simplicity. Once an initial population P_0 is created, the main GA cycle can begin.

3.4 Fitness functions

The fitness function, known as objective or cost function, is used to provide a measure of how chromosomes have performed in the problem domain. Fitness function should avoid being extremely rugged which will lead to slow or poor convergence of the GA.

Given a population P_t at generation t, the GA iteration starts by evaluated the set:

$F_t = F_t(1), F_t(2), \ldots\ldots, F_t(N_p)$

Of objective function values associated with the chromosomes $X_t(k)$ Where $k = 1, 2, \ldots, N_p$

The GA applies the genetic operators and selection to produce population P_{t+1} for the next generation. Although the objective function for GA is formulated as in classical optimization algorithm, the GA does not need gradient information. Therefore, the mathematical structure of these algorithms is simple and flexible.

3.5 Reproduction

Reproduction operation selects chromosomes based on their fitness. The fitter chromosomes have higher chance to be selected and survive to the next generation. The selection mechanism can be done using different schemes such as rank selection, fitness proportionate selection (FPS) and tournament selection. One of The most common schemes implementing FPS is the roulette wheel selection [3]. In this method, the wheel is divided to non-equal spaces with respect to chromosomes fitness. The chromosome with highest fitness gets the largest space and greatest chance to be chosen. Let's say we have a population of four chromosomes with fitness value of 25%, 21%, 15%, and 39% of the total fitness. Since we have population of four, the Roulette wheel will spin four times and choose one each time. Since the chromosome with fitness of 39% has the highest fitness and the largest space on the wheel, it has the greatest chance to be selected. The Roulette wheel scheme can be executed as the following steps:

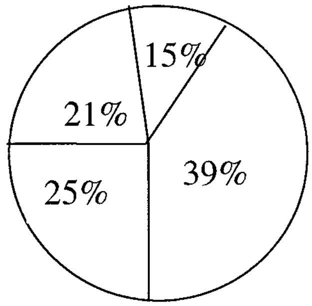


Figure (3.2) the roulette wheel

1- Calculate the total fitness by summing the fitness of all chromosomes.

2- Generate a random number between 0 and the total fitness.

3- Return the first chromosome whose fitness added to the fitness of the preceding chromosomes is greater than or equal to the random number.

4- Repeat step 1 to step 3 until the population size is reached.

3.6 Crossover

The goal of crossover is to generate new chromosomes (offspring) that are fitter than their parents and contain both parents' genes. Crossover probability P_c, defined as the ratio of the number of offspring produced in each generation to the population size, is set to control the crossover operation rate. A higher crossover probability reduces the chance of being stuck in local optimum. But if P_c is too high, the chromosomes with high fitness may be destroyed. Some ways of performing crossover operation are one-point crossover and two –point crossover. One-point crossover is done by choosing two chromosomes as parents then choose a crossover point to make the exchange. The crossover point divides each chromosome into two halves and the exchange is executed by swapping one half of one chromosome with one half of the second chromosome. In two-point crossover, two crossover points are chosen then we exchange the two ends of one chromosome with the two ends of other chromosome. One point crossover can be executed as following:

1- Randomly select two chromosomes as parents

2- Generate a random number between 0 and 1

3- If the number is less than P_c, select a random crossover point and exchange the chromosomes beyond this point between the parents.

4- If the number is greater than P_c, the two parents are cloned to the next generation.

5- Repeat step 1 to step 5 until the whole population is reached.

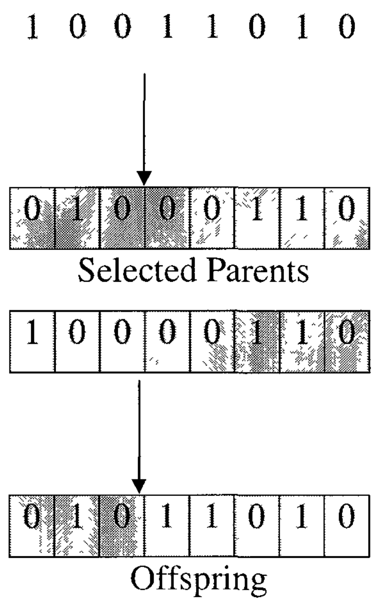


Figure (3.3) One-point Crossover

3.7 Mutation

Mutation operation randomly changes an offspring after crossover operation and it occurs at low probability rate named mutation probability P_m. Mutation operation is the key operation to maintain diversity in genetic algorithm and prevents GA from being stuck at local optimal solution. Bit-flip mutation is the most common mutation operation in binary encoding. It is done by inverting '1' to'0' and '0' to '1' after passing a probability test on their position. In case they do not pass the bits remain unchanged.

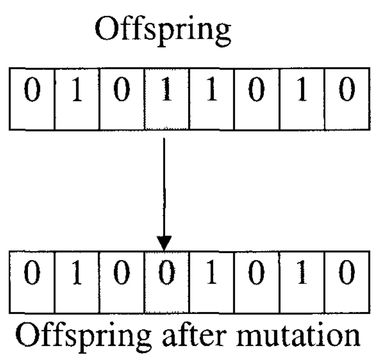


Figure (3.4) Mutation operation

3.8 Example

To demonstrate how GA works, we will try to find the maximum of

$F(x) = x_1 + x_2 + x_3$

We will follow the steps illustrated in figure (3.5)

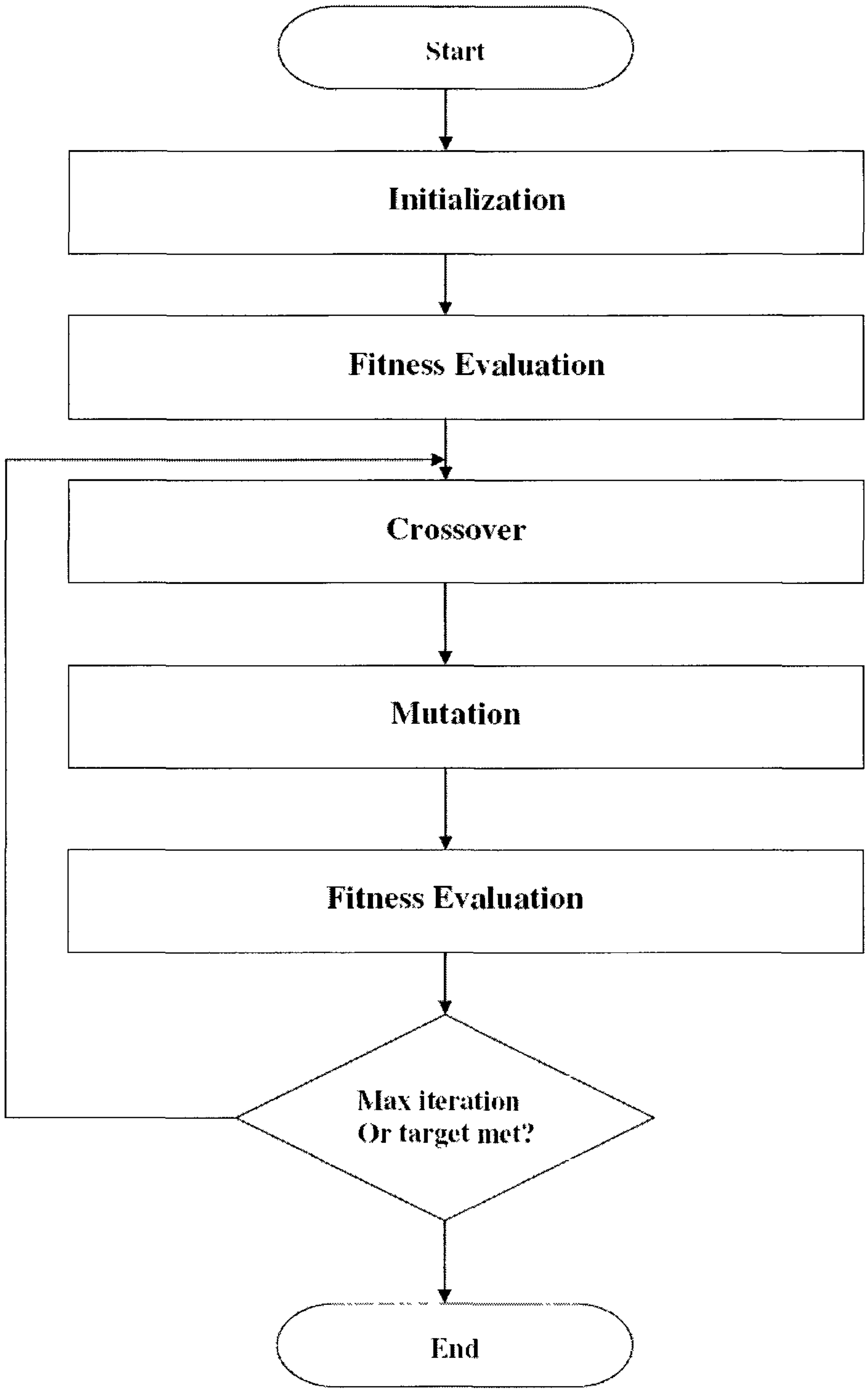


Figure (3.5) flow chart of GA

Step one. Initialization

In this step, we will initialize a random population of four chromosomes. Each chromosome contains the initial value of the three variables in the objective function $x_1, x_2,$ *and* x_3. For simplicity, we will encode the three variables in binary of four bits wordlength.

$x_1 = 1 \quad 0\ 0\ 0\ 1$

$x_2 = 3 \quad 0\ 0\ 1\ 1$

$x_3 = 0 \quad 0\ 0\ 0\ 0$

To construct a chromosome, all three variables are concatenated

[0 0 0 1 0 0 1 1 0 0 0 0]

Now, we generate the full generation

Chromosomes													Decimal values		
													X1	X2	X3
Ch1	0	0	0	1	0	0	1	1	0	0	0	0	1	3	0
Ch2	0	0	1	0	0	1	0	0	0	1	0	1	2	4	5
Ch3	0	1	1	0	0	0	0	1	0	0	1	0	6	1	2
Ch4	1	0	0	1	0	1	0	0	1	0	1	0	9	4	10

Initial population

Step Two. Fitness Evaluation

To get the fitness value of each chromosome, we decoded them to decimal then apply the fitness function as following:

Chromosomes													Fitness Values
Ch1	0	0	0	1	0	0	1	1	0	0	0	0	4
Ch2	0	0	1	0	0	1	0	0	0	1	0	1	11
Ch3	0	1	1	0	0	0	0	1	0	0	1	0	9
Ch4	1	0	0	1	0	1	0	0	1	0	1	0	23
Total fitness													47

The result of fitness evaluation

Step Three. Reproduction

In this example, we will use Roulette Wheel to select parents as following:

1-Sum up all fitness value 47.

2-Choose a random number between 1and 47 let us say 25.

3-Return the first chromosome whose fitness added to the fitness of the preceding chromosomes is greater than or equal to the randomly selected number as shown in the following.

Chromosomes													Sum of the preceding Fitness
Ch1	0	0	0	1	0	0	1	1	0	0	0	0	0+4=4
Ch2	0	0	1	0	0	1	0	0	0	1	0	1	4+11=15
Ch3	0	1	1	0	0	0	0	1	0	0	1	0	15+9=24
Ch4	1	0	0	1	0	1	0	0	1	0	1	0	24+23=47

Sum of preceding fitness values

Since 47 is larger than the randomly selected number 24 and the fourth chromosome becomes the first chosen parent

Ch1	1	0	0	1	0	1	0	0	1	0	1	0

The first selected parent

We repeat steps 1-3 until the full population is reached.

The second random number is 15

Chromosomes													Sum of the preceding Fitness
Ch1	0	0	0	1	0	0	1	1	0	0	0	0	0+4=4
Ch2	0	0	1	0	0	1	0	0	0	1	0	1	4+11=15
Ch3	0	1	1	0	0	0	0	1	0	0	1	0	15+9=24
Ch4	1	0	0	1	0	1	0	0	1	0	1	0	24+23=47

So chromosome number two is chosen and becomes the second parent

Ch2	0	0	1	0	0	1	0	0	0	1	0	1

The second selected parent

The third random number is 2

Chromosomes													Sum of the preceding Fitness
Ch1	0	0	0	1	0	0	1	1	0	0	0	0	0+4=4
Ch2	0	0	1	0	0	1	0	0	0	1	0	1	4+11=15
Ch3	0	1	1	0	0	0	0	1	0	0	1	0	15+9=24
Ch4	1	0	0	1	0	1	0	0	1	0	1	0	24+23=47

So chromosome number one chosen and becomes the third parent

Ch3	0	0	0	1	0	0	1	1	0	0	0	0

The third selected parent

The fourth random number is 10

Chromosomes													Sum of the preceding Fitness
Ch1	0	0	0	1	0	0	1	1	0	0	0	0	0+4=4
Ch2	0	0	1	0	0	1	0	0	0	1	0	1	4+11=15
Ch3	0	1	1	0	0	0	0	1	0	0	1	0	15+9=24
Ch4	1	0	0	1	0	1	0	0	1	0	1	0	24+23=47

So chromosome number two chosen and becomes the fourth parent

Ch4	0	0	1	0	0	1	0	0	0	1	0	1

The fourth selected parent

Step Four. Crossover

One-point crossover is used in this example as following:

1- Randomly select a pair of chromosomes to make the crossover operation. So we divide the population into pairs.

2- Choose a crossover probability Pc say 90% then generate a random number between [0 1] for each pair say [.6 .3]

3- Since .6 and .3 less than Pc, we select a random crossover point for each pair. These points should be between 1 and 12, say 3, 8.

Ch1	0	0	0	1	0	0	1	1	0	0	0	0

Ch2	0	0	1	0	0	1	0	0	0	1	0	1

Crossover for the first and second chromosomes

Offspring1	0	0	0	0	0	1	0	0	0	1	0	1

First offspring

Offspring2	0	0	1	1	0	0	1	1	0	0	0	0

Second offspring

Ch3	0	1	1	0	0	0	0	1	0	0	1	0

Ch4	1	0	0	1	0	1	0	0	1	0	1	0

Crossover for the third and fourth chromosomes

Offspring3	0	1	1	0	0	0	0	1	1	0	1	0

Third offspring

Offspring4	1	0	0	1	0	1	0	0	0	0	1	0

Fourth offspring

So the new population is as following:

New population of offspring													Decimal values		
													X1	X2	X3
Offspring1	0	0	0	0	0	1	0	0	0	1	0	1	0	4	5
Offspring2	0	0	1	1	0	0	1	1	0	0	0	0	3	3	0
Offspring3	0	1	1	0	0	0	0	1	1	0	1	0	6	1	10
Offspring4	1	0	0	1	0	1	0	0	0	0	1	0	9	4	2

The resulting offspring

Step Five. Mutation

Each bit of each chromosome is examined with the mutation probability Pm and if a random generated number less than Pm, mutation is applied.

Offspring1	0	0	0	0	0	1	0	0	0	1	0	1	8	4	5
Offspring2	0	0	1	1	0	0	1	1	0	0	0	0	3	3	0
Offspring3	0	1	1	0	0	0	0	1	1	0	1	0	6	0	10
Offspring4	1	0	0	1	0	1	0	0	0	0	1	0	9	12	2

$\Downarrow$

Offspring1	1	0	0	0	0	1	0	0	0	1	0	1
Offspring2	0	0	1	1	0	0	1	1	0	0	0	0
Offspring3	0	1	1	0	0	0	0	0	1	0	1	0
Offspring4	1	0	0	1	1	1	0	0	0	0	1	0

Offspring after mutation

Step Six. Fitness Evaluation

The new population is decoded to decimal then evaluated by applying the fitness function as the following:

New population of offspring													Decimal values		
													X1	X2	X3
Offspring1	1	0	0	0	0	1	0	0	0	1	0	1	8	4	5
Offspring2	0	0	1	1	0	0	1	1	0	0	0	0	3	3	0
Offspring3	0	1	1	0	0	0	0	0	1	0	1	0	6	0	10
Offspring4	1	0	0	1	1	1	0	0	0	0	1	0	9	12	2

The decoded offspring population

Chromosomes													Fitness function
Offspring1	1	0	0	0	0	1	0	0	0	1	0	1	17
Offspring2	0	0	1	1	0	0	1	1	0	0	0	0	6
Offspring3	0	1	1	0	0	0	0	0	1	0	1	0	16
Offspring4	1	0	0	1	1	1	0	0	0	0	1	0	23
Total fitness													62

Fitness evaluation of offspring

Now, we have completed the first generation and the next generation will use the current population with its offspring chromosomes and fitness values and apply the roulette Wheel again. The maximum solution for this problem is [111111111111111] where each variable equals to 15.

3.9 GA Analysis

To understand the effects of GA operations, the schema notion has to be defined. A schema is a combination of a coding system of chromosomes and a 'do not care symbol' *. The do not care symbol * may take any value of the coding system for example let the binary system be the coding system So * can take 0 or 1. Let us have a four bit chromosome *1*0, this chromosome could be 0100, 0110, 1100, or 1110.

3.9.1 The Effect of Reproduction

Let $H(S_i,t)$ represents the number of chromosomes of schema S_i. If Roulette Wheel is used, the number of chromosomes of the next schema can be estimated

$$\mathrm{H}(S_i,t+1)=\mathrm{H}(\mathrm{S}_i,\mathrm{t})\times\frac{f(S_i,t)}{f(t)}$$

$f(S_i,t)$ is the average fitness of schema i .

$f(t)$ is the average of the current schema.

If $f(S_i,t)>1$ that means the schema has an above average fitness in the generation i and receive an increasing number of offspring in the next generation.

If $f(S_i,t)<1$that means the schema has a below average fitness in the generation i and receive a decreasing number of offspring in the next generation.

3.9.2 The Effect of Crossover Operation

A schema survives the crossover operation if the crossover point falls outside the schema length $d(s)$. If one-point crossover is applied, the probability of a schema being destroyed is

$$P_d(s) = \frac{d(s)}{L-1}$$

L is the length of the chromosome

So the probability of being survived is

$$P_s(s) = 1 - P_d(s)$$

And by using the crossover probability P_c is

$$P_{sc}(s) \geq 1 - P_c \times P_d(s)$$

As result, the longer the schema is, the higher chance it will be destroyed.

3.9.3 The Effect of Mutation

A schema survives from distortion only if all positions in the schema remain unchanged. The survive probability is

$$P(s) = P_b^{O(s)}$$

$O(s)$ is the order of the schema

P_b is the probability of a single bit to survive

$$P_b = 1 - P_m$$

P_m is the mutation probability.

Since $P_m \ll 1$

$$P(s)=1-P_m \times O(s)$$

So the higher the order is, the higher the chance of a schema being destroyed.

CHAPTER IV

DESIGN 2-D DIGITAL FILTERS BY SVD

4.1 Design recursive digital filters

The transfer function of a quadrantally symmetric filter has a separable denominator $H(z_1,z_2)$ can be expressed as

$$H(z_1,z_2)=\sum_{i=1}^{k} f_i(z_1)g_i(z_2) \qquad (4.1)$$

Let $A=(a_{lm})$ be a desired amplitude response where

$$a_{lm}=\left|H(e^{j\pi u_l},e^{j\pi v_m})\right|, \qquad 1\le l\le \mathrm{L}, \quad 1\le m\le M$$

and let u_l *and* v_m, be normalized frequencies such that

$$u_l=\frac{l-1}{L-1} \quad and \quad \mathrm{v}_m=\frac{m-1}{M-1}$$

$$0\le u_l\le 1, \qquad 0\le v_m\le 1$$

The singular value of A

$$A=\sum_{i=1}^{r}\sigma_i u_i v_i' \qquad (4.2)$$

Where σ_i are the singular values of A.

u_i is the ith eigenvector of AA^T associated with the ith eigenvalue σ_i^2.

v_i is the ith eigenvector of $A^T A$ associated with σ_i^2.

r is the rank of A, and v_i' denotes the transpose of v_i.

If we let $\phi_i=\sigma_i^2 u_i$ *and* $\gamma_i=\sigma_i^2 v_i$, then can be written as

$$A = \sum_{i=1}^{r} \phi_i \gamma_i' \qquad (4.3)$$

Where ϕ_i *and* γ_i are sets of orthogonal L-dimensional and M-dimensional vectors, respectively.

Now, by comparing (4.1 and 4.3) and assuming that k = 1, r=1 and that ϕ_1 *and* γ_1 are the desired amplitude responses for the 1-D filters characterized by $f_1(z_1)$ and $g_1(z_2)$ respectively, a 2-D digital filter can be designed as the following steps:

1- Design 1-D filters f_1 and g_1, characterized by $f_1(z_1)$ and $g_1(z_2)$ by using one of the many available optimization methods.

2- Connec t filters f_i and g_i, in cascade.

The transfer function of the cascade filter obtained is given by

$$H_1(z_1,z_2)=f_1(z_1)g_1(z_2) \qquad (4.4)$$

An important property of the SVD can be stated as

$$\left\| A - \sum_{i=1}^{k} \phi_i \gamma_i' \right\| = \min_{\phi_i^{\wedge}, \gamma_i^{\wedge}} \left\| A - \sum_{i=1}^{k} \phi_i^{\wedge} \gamma_i^{\wedge'} \right\| \quad \text{for } 1 \le k \le r$$

$$where\ \phi_i^{\wedge} \in R^L, \gamma_i^{\wedge} \in R^L, and \qquad (4.5)$$

$$\|X\| = \left[\sum_{l=1}^{L} \sum_{m=1}^{M} x_{lm}^2 \right]^{1/2}$$

is the Frobenius norm of a matrix $X = (x_{lm}) \in R^{L \times M}$. The above relation shows that for any fixed k (1<=k <= r), $\sum_{i=1}^{k} \phi_i \gamma_i^{'}$ is a minimal mean-square-error approximation to A. Since r >1, from Equation (4.5) can be written

$$\left\| A - \left| H_1(e^{j\pi u_l}, e^{j\pi v_m}) \right| \right\| \approx \left\| A - \phi_1 \gamma_1^{'} \right\| = \varepsilon_1 \qquad (4.6)$$

From (4.6) A can be written as

$$A = \phi_1 \gamma_1^{'} + \varepsilon_1 \qquad (4.7)$$

If we compare (4.3) and (4.7), we get

$$\varepsilon_1 = \sum_{i=2}^{r} \phi_i \gamma_i^{'}$$

$$A = \phi_1 \gamma_1^{'} + \sum_{i=2}^{r} \phi_i \gamma_i^{'}$$

The approximation error ε_1 is associated with the number of sections k that have been realized. So ε_1 may be reduced by finding a way of realizing more of the terms in (4.3) by means of parallel filter sections so that we can write

$$\begin{aligned} & A = \phi_1 \gamma_1^{'} + \phi_2 \gamma_2^{'} + \varepsilon_2 \\ & where \qquad (4.8) \\ & \varepsilon_2 = \sum_{i=3}^{r} \phi_i \gamma_i^{'} \end{aligned}$$

Since all entries of A are nonnegative, it follows that all entries of ϕ_1 and γ_1 are nonnegative. Nevertheless, the elements of ϕ_i and γ_i for i >= 2 may assume negative values so a careful treatment of ϕ_2 and γ_2 is necessary to get red of their negative components. Let $\overline{\phi}_2$ and $\overline{\gamma}_2$ be the absolute values of the most negative components of ϕ_2 and γ_2 respectively. If

$e_\phi = [1\ 1\ .\ .\ .\ 1]' \in \mathrm{R}^L$ and $e_\gamma = [1\ 1\ .\ .\ .\ 1]' \in \mathrm{R}^M$

Then all components of

$$\tilde{\phi}_2 = \phi_2 + \overline{\phi}_2\ \mathrm{e}_\phi \quad \text{and} \quad \tilde{\gamma}_2 = \gamma_2 + \overline{\gamma}_2\ \mathrm{e}_\gamma \qquad (4.9)$$

are nonnegative. Let us assume that it is possible to design 1-D linear-phase or zero-phase filters characterized by

$$\tilde{f}_1(z_1), \tilde{g}_1(z_2), \tilde{f}_2(z_1), \textit{ and } \tilde{g}_2(z_2) \textit{ such that}$$

$$\tilde{f}_i(e^{j\pi u_l}) = \left|\tilde{f}_i(e^{j\pi u_l})\right| e^{j\alpha_1 u_l} \qquad 1 \le l \le L \quad i=1,2$$

$$\tilde{g}_i(e^{j\pi v_m}) = \left|\tilde{g}_i(e^{j\pi v_m})\right| e^{j\alpha_2 v_l} \qquad 1 \le m \le M \quad i=1,2$$

$$\textit{where} \qquad (4.10)$$

$$\left|\tilde{f}_1(e^{j\pi u_l})\right| \approx \phi_{1l} \ , \ \left|\tilde{f}_2(e^{j\pi u_l})\right| \approx \tilde{\phi}_{2l}$$

$$\left|\tilde{g}_1(e^{j\pi v_m})\right| \approx \gamma_{1m} \ , \ \left|\tilde{g}_2(e^{j\pi v_m})\right| \approx \tilde{\gamma}_{2m}$$

Here $\tilde{\phi}_{2l}$ and $\tilde{\gamma}_{2m}$ are the Lth component of $\tilde{\phi}_2$ and the mth component of $\tilde{\gamma}_2$ respectively, and α_1 , α_2 are constants which are equal to zero if zero-phase filters are to be employed. Now let

$\alpha_1 = -\pi n_1$ and $\alpha_2 = -\pi n_2$

where n_1 , n_2 are nonnegative integers, and define

$$f_2(z_1) = \tilde{f}_2(z_1) - \bar{\phi}_2 z_1^{-n1}$$

and

$$g_2(z_1) = \tilde{g}_2(z_2) - \bar{\gamma}_2 z_2^{-n2}$$

if we form

$$H_2(z_1,z_2) = f_1(z_1)g_1(z_2) + f_2(z_1)g_2(z_2) \qquad (4.11)$$

$$\begin{aligned}\left|H_2(e^{j\pi u_l}, e^{j\pi v_m})\right| \\ &= \left|f_1(e^{j\pi u_l})g_1(e^{j\pi v_m}) + f_2(e^{j\pi u_l})g_2(e^{j\pi v_m})\right| \\ &\approx \left|\phi_{1l}\gamma_{1m} + \phi_{2l}\gamma_{2m}\right| \quad 1 \le l \le L,\ 1 \le m \le M\end{aligned}$$

$$(4.12)$$

which in conjunction with (4.8) implies that

$$\begin{aligned}\left\|A - \left|H_2(e^{j\pi u_l}, e^{j\pi v_m})\right|\right\| \approx \left\|A - \left|\phi_1\gamma_1' + \phi_2\gamma_2'\right|\right\| \\ \le \left\|A - (\phi_1\gamma_1' + \phi_2\gamma_2')\right\| = \varepsilon_2 = \min_{\hat{\phi}_1, \hat{\gamma}_1} \left\|A - (\hat{\phi}_1\hat{\gamma}_1' + \hat{\phi}_2\hat{\gamma}_2')\right\|\end{aligned}$$

$$(4.13)$$

Evidently, the approximation error has been reduced from ε_1 to ε_2 by means of a parallel subfilter. According to (4.13), the two-section 2-D digital filter obtained has an amplitude response which is a minimal mean-square-error approximation to the desired amplitude response.

Since $f_1(z_1)$ and $g_1(z_2)$ corresponds to the largest singular value σ_{1m}, the quantity $\left| f_1(e^{j\pi u_l}) g_1(e^{j\pi v_m}) \right|$ represents the main contribution to the amplitude response of the 2-D filter. For this reason, the subfilter characterize $f_1(z_1)$ and $g_1(z_2)$ is said to be the main section of the 2-D filter. On the other hand, $\left| f_2(e^{j\pi u_l}) g_2(e^{j\pi v_m}) \right|$

represents a correction to the amplitude response, and the subfilter characterized by $f_1(z_1)$ and $g_1(z_2)$ is said to represent a correction section.

Other correction sections characterized by $f_i(z_1)$ and $g_i(z_2)$ can be designed using ϕ_i and γ_i (i=3,....,k,k ≤ r) in similar manner. When k sections are designed, including the main section, $H_k(z_1,z_2)$ can be formed as

$$H_k(z_1,z_2) = \sum_{i=1}^{k} f_i(z_1)\, g_i(z_2)$$

Then we have

$$\left\| A - \left| H_k(e^{j\pi u_l}, e^{j\pi v_m}) \right| \right\| \approx \left\| A - \left| \sum_{i=1}^{k} \phi_i \gamma_i' \right| \right\| \leq \varepsilon_k = \min_{\hat{\phi}_i, \hat{\gamma}_i} \left\| A - \sum_{i=1}^{k} \hat{\phi}_i \hat{\gamma}_i' \right\|$$

(4.14)

4.2 ERROR COMPENSATION

A further improvement is possible through the use of error compensation. When the main section and the correction sections are designed by using an optimization method, approximation errors will inevitably occur which will accumulate and manifest themselves as the overall approximation error in the design of the 2-D filter. Fortunately, it is possible to prevent the accumulation of error through compensation. When the design of the main section is complete, define an error matrix

$$E_1 = A - \left| f_1(e^{j\pi u_l}) g_1(e^{j\pi v_m}) \right| \qquad (4.15)$$

and then perform SVD on E1, to obtain

$$E_1 = S_{22}\phi_{22}\gamma_{22}' + \ldots + S_{r2}\phi_{r2}\gamma_{r2}' \qquad (4.16)$$

Data ϕ_{22} and γ_{22}' can be used to deduce $f_2(z_1)\, g_2(z_2)$ thus, the first correction section can be designed. Now form error matrix E, as

$$\begin{aligned} E_2 &= E_1 - \left| S_{22} f_2(e^{j\pi u_l}) g_2(e^{j\pi v_m}) \right| \\ &= A - \left| f_1(e^{j\pi u_l}) g_1(e^{j\pi v_m}) + S_{22} f_2(e^{j\pi u_l}) g_2(e^{j\pi v_m}) \right| \end{aligned} \qquad (4.17)$$

and perform SVD on E, to obtain

$$E_2 = S_{33}\phi_{33}\gamma_{33}' + \ldots + S_{r3}\phi_{r3}\gamma_{r3}'$$

As before, data ϕ_{33} and γ_{33}' can be used to design the second correction section. This procedure is continued until the elements of the error matrix become sufficiently small for the application at hand.

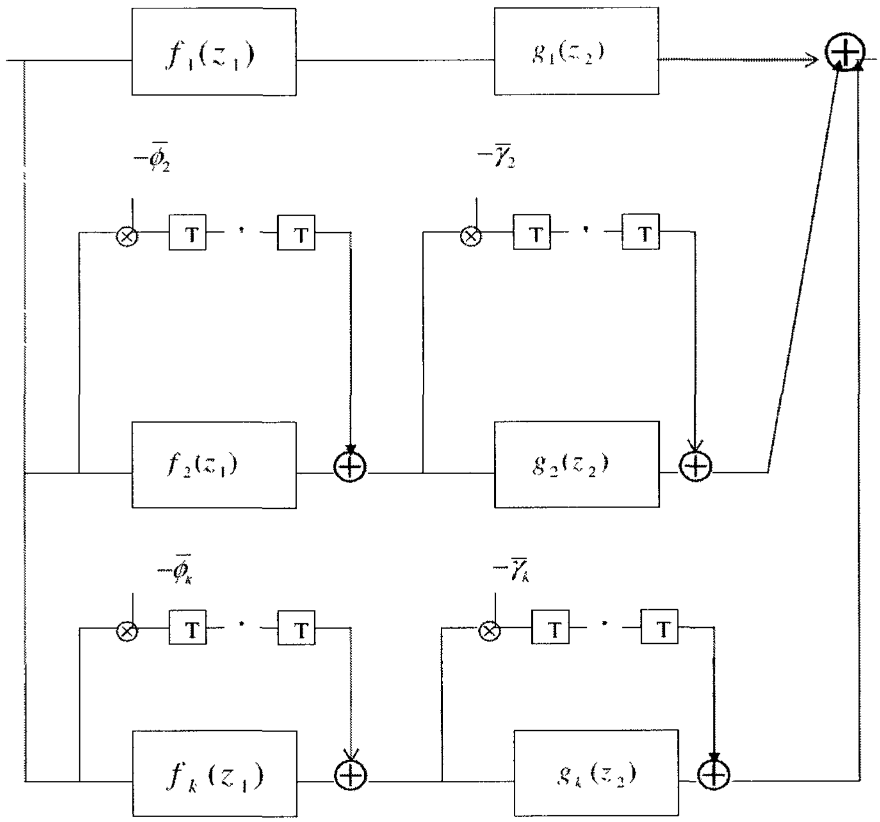


Figure (4.1) Realization of quadrantally symmetric2-D filter

Example (1)

Design 4th order 2-D IIR low pass filter specified by:

$$\left|H\left(e^{j\omega_1 T_1}, e^{j\omega_2 T_2}\right)\right| = \begin{cases} 1 & 0 \le \sqrt{\omega_1^2+\omega_2^2} \le 0.45\pi \\ 0 & 0.45\pi < \sqrt{\omega_1^2+\omega_2^2} \le \pi \end{cases}$$

$$where \quad T_1 = T_2 = 1$$

M=L=41

Five sections are used for this example.

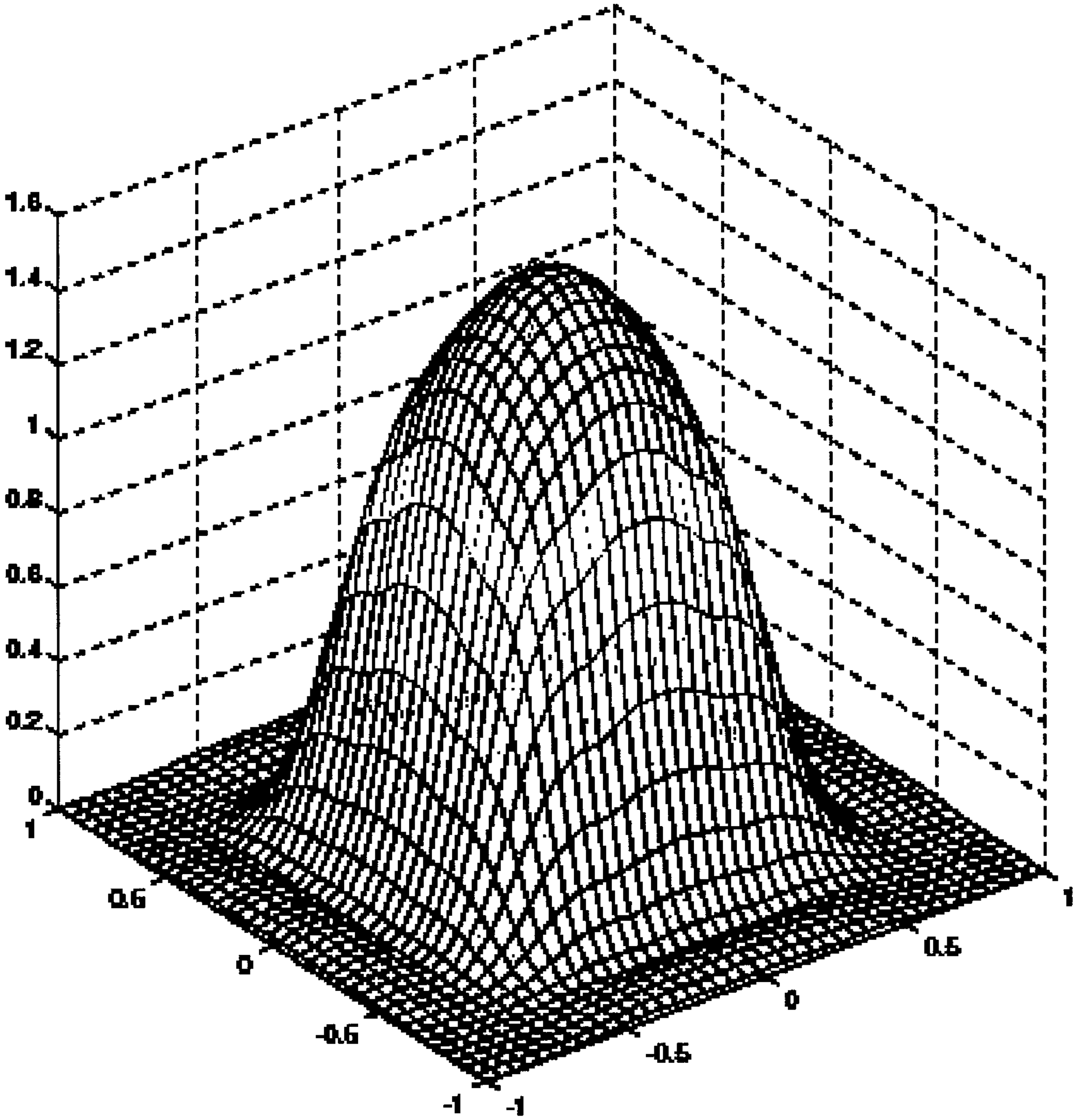


Figure (4.2) 4^{th} order low pass IIR

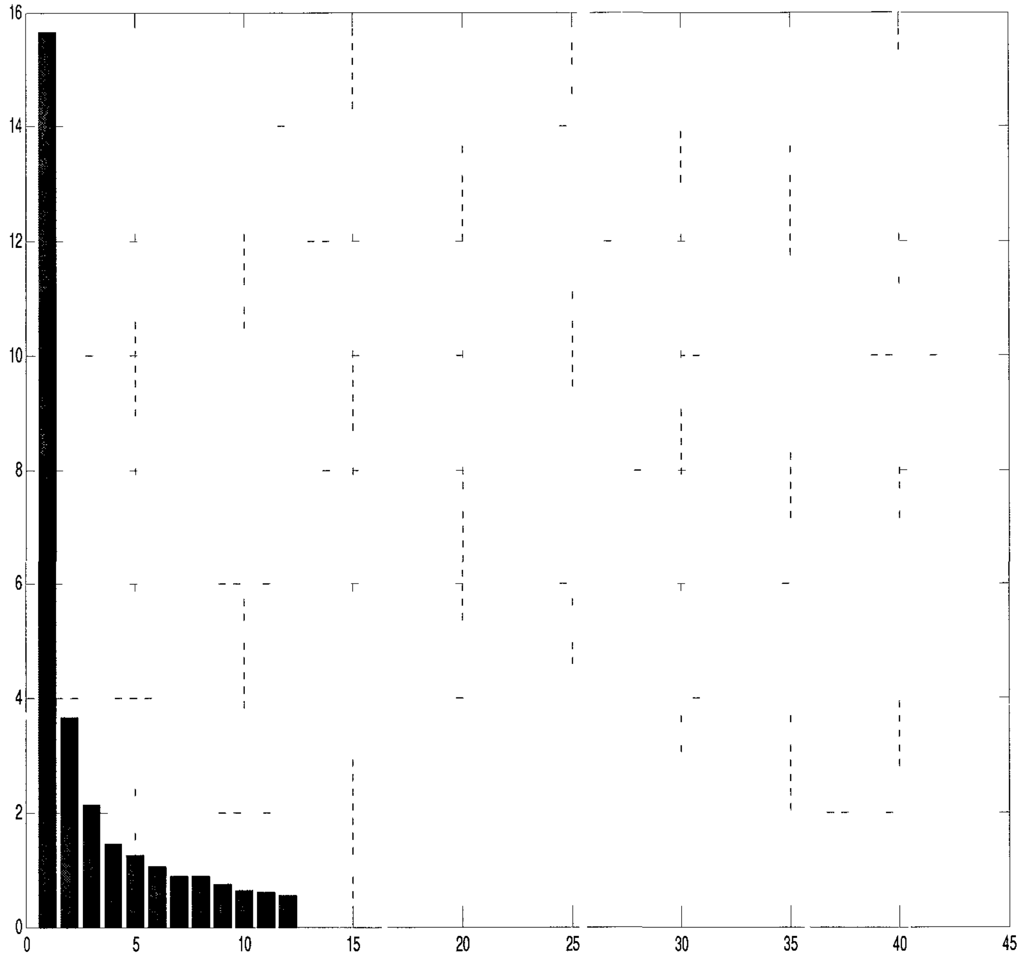


Figure (4.3) the singular values of the 4th order low pass IIR

4.3 Design nonrecursive digital filters

The transfer function of a 2-D FIR filter with support in the rectangle defined by $-\frac{N_i}{2} \le n_i \le \frac{N_i}{2}, i = 1, 2$ can be written as

$$H(z_1, z_2) = \sum_{n_1 = -N_1/2}^{N_1/2} \sum_{n_2 = -N_2/2}^{N_2/2} h(n_1, n_2)\, z_1^{-n_1} z_2^{-n_2} \qquad (4.18)$$

Where $h(n_1, n_2)$ is the impulse response. If $h(n_1, n_2)$ is real and

$h(n_1, n_2) = h(-n_1, -n_2)$

Then the frequency response of the filter given by

$$H(z_1, z_2) = \sum_{n_1 = -N_1/2}^{N_1/2} \sum_{n_2 = -N_2/2}^{N_2/2} h(n_1, n_2)\, e^{-j\omega_1 n_1 T_1}\ e^{-j\omega_2 n_2 T_2}$$

$$= X\,(\omega_1, \omega_2)$$

Is symmetrical with respect to the origin of (ω_1, ω_2) plane such that

$$X\,(\omega_1, \omega_2) = X\,(-\omega_1, -\omega_2) \qquad (4.19)$$

$$where \quad -\pi \le \omega_1, \omega_2 \ge \pi$$

Assume the desired arbitrary frequency response A satisfies equation (4.19) that is:

$$X\,(\pi\, u_l, \pi\, v_m) = X\,(-\pi\, u_l, -\pi\, v_m)$$

$$1 \le l \le L \quad and \quad 1 \le m \le M$$

u_l *and* v_m are normalized frequencies

$$u_l = -1 + 2\left(\frac{l-1}{L-1}\right) \quad \text{and} \quad v_m = -1 + 2\left(\frac{m-1}{M-1}\right)$$

$$-l \le u_l \le l, \quad -l \le v_m \le l$$

The SVD of A gives

$$A = \sum_{i=1}^{r} \sigma_i u_i v_i^{'}$$

Where σ_i are the singular values of A,

u_i is the ith eigenvector of AA^T associated with the ith eigenvalue σ_i^2

v_i is the ith eigenvector of $A^T A$ associated with σ_i^2.

r is the rank of A, and $v_i^{'}$ denotes the transpose of v_i.

If we let $\phi_i = \sigma_i^2 u_i$ and $\gamma_i = \sigma_i^2 v_i$, then can be written as

$$A = \sum_{i=1}^{r} \phi_i \gamma_i^{'}$$

Since the frequency response satisfies the equation 4.19 then vectors ϕ_i *and* γ_i are either mirror image symmetric or antisymmetric simultaneously for i=1,2,....r

The transfer function of (4.18) can be rewritten as

$$H(z_1, z_2) = \sum_{i=1}^{k} F_i(z_1)\, G_i(z_2)$$

This means that a 2-D FIR filter can be designed as a cascade of k section of 1-D subfilters. $F_i(z_1)$ and $G_i(z_2)$ are the transfer function of two cascaded 1-D subfilters. Since these subfilters are FIR filters with support in the rectangle defined by $-\frac{N_i}{2} \le n_i \le \frac{N_i}{2}, i = 1, 2$ we have

$$F_i(z_1) = \sum_{n1=-N_1/2}^{n1=N_1/2} f_i(n_1) z_1^{-n_1} \qquad (4.20)$$

$$G_i(z_2) = \sum_{n1=-N_1/2}^{n1=N_1/2} g_i(n_2) z_2^{-n_2} \qquad (4.21)$$

Assume $F_i(z_1)$ and $G_i(z_2)$ represent zero-phase or $\pi/2$ phase filters, then their frequency responses are given by

$$F_i(ze^{-j\omega_1 T_1}) = \sum_{n1=-N_1/2}^{n1=N_1/2} f_i(n_1) e^{-j\omega_1 n_1 T_1}$$

$$= \Phi_i(\omega_1) e^{j\theta_i} \qquad (4.22)$$

$$G_i(z_2) = \sum_{n1=-N_1/2}^{n1=N_1/2} g_i(n_2) e^{-j\omega_2 n_2 T_2}$$

$$= \Gamma_i(\omega_2) e^{j\theta_i} \qquad (4.23)$$

In case $f_i(n_1)$ and $g_i(n_1)$ are mirror-image symmetric

$\theta_i = 0 \rightarrow \Phi_i(\omega_1)$ *and* $\Gamma_i(\omega_2)$ are real function that are even with respect to ω_1 *and* ω_2 respectively.

In case $f_i(n_1)$ and $g_i(n_1)$ are mirror-image antisymmetric

$\theta_i = \frac{\pi}{2} \rightarrow \Phi_i(\omega_1)$ *and* $\Gamma_i(\omega_2)$ are real function that are odd with respect to ω_1 *and* ω_2.

So a zero-phase 2-D filter can be obtained as:

$$H(e^{j\omega_1 T_1}, e^{j\omega_2 T_2}) = \sum_{i=1}^{k} F_i(e^{j\omega_1 T_1}) G_i(e^{j\omega_2 T_2})$$
$$= \sum_{i=1}^{k} \pm \Phi_i(\omega_1) \Gamma_i(\omega_2) \qquad (4.24)$$

Where

$$+ \rightarrow \theta_i = 0$$
$$- \rightarrow \theta_i = \frac{\pi}{2}$$

From 4.24 and 4.18

$$X(\omega_1, \omega_2) = \sum_{i=1}^{k} \pm \Phi_i(\omega_1)\, \Gamma_i(\omega_2) \qquad (4.25)$$

So to obtain a zero-phase 2-D FIR filter, we design 1-D FIR subfilters characterized by $F_i(z_1)$ and $G_i(z_2)$ as zero-phase or $\pi/2$ phase filter then connect them.

The impulse response of the resulting filter is given by

$$H(n_1, n_2) = \sum_{i=1}^{k} f_i(n_1) g_i(n_1) \qquad (4.26)$$

Example (2)

Design 17th order 2-D FIR low pass filter specified by:

$$\left|H\left(e^{j\omega_1 T_1}, e^{j\omega_2 T_2}\right)\right| = \begin{cases} 1 & 0 \le \sqrt{\omega_1^2 + \omega_2^2} \le 0.35\pi \\ 0 & 0.65\pi \le \sqrt{\omega_1^2 + \omega_2^2} \le \pi \end{cases}$$

$$where \quad T_1 = T_2 = 1$$

M=L=21

Five sections are used

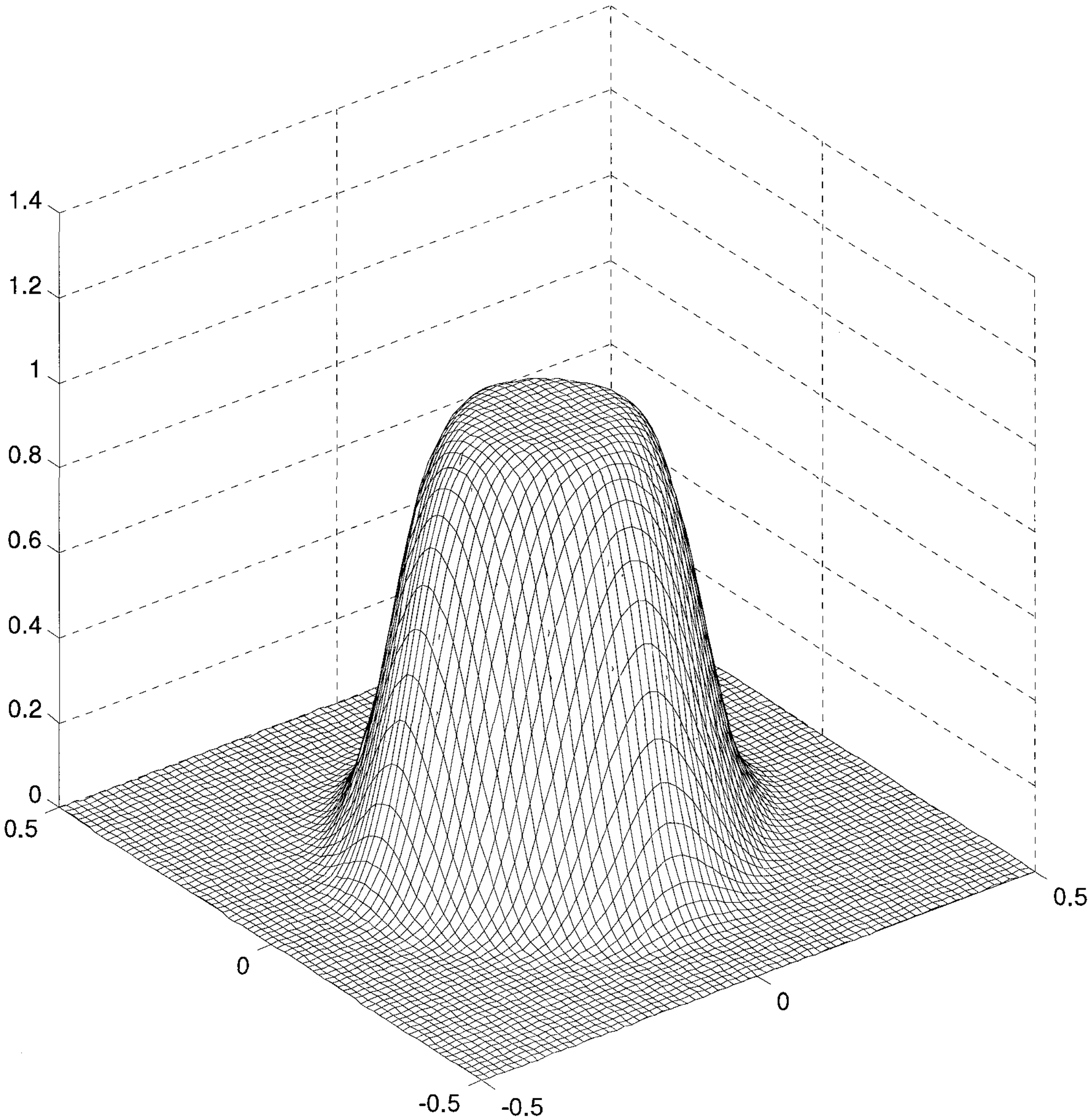


Figure (4.4) the 17th order low pass FIR

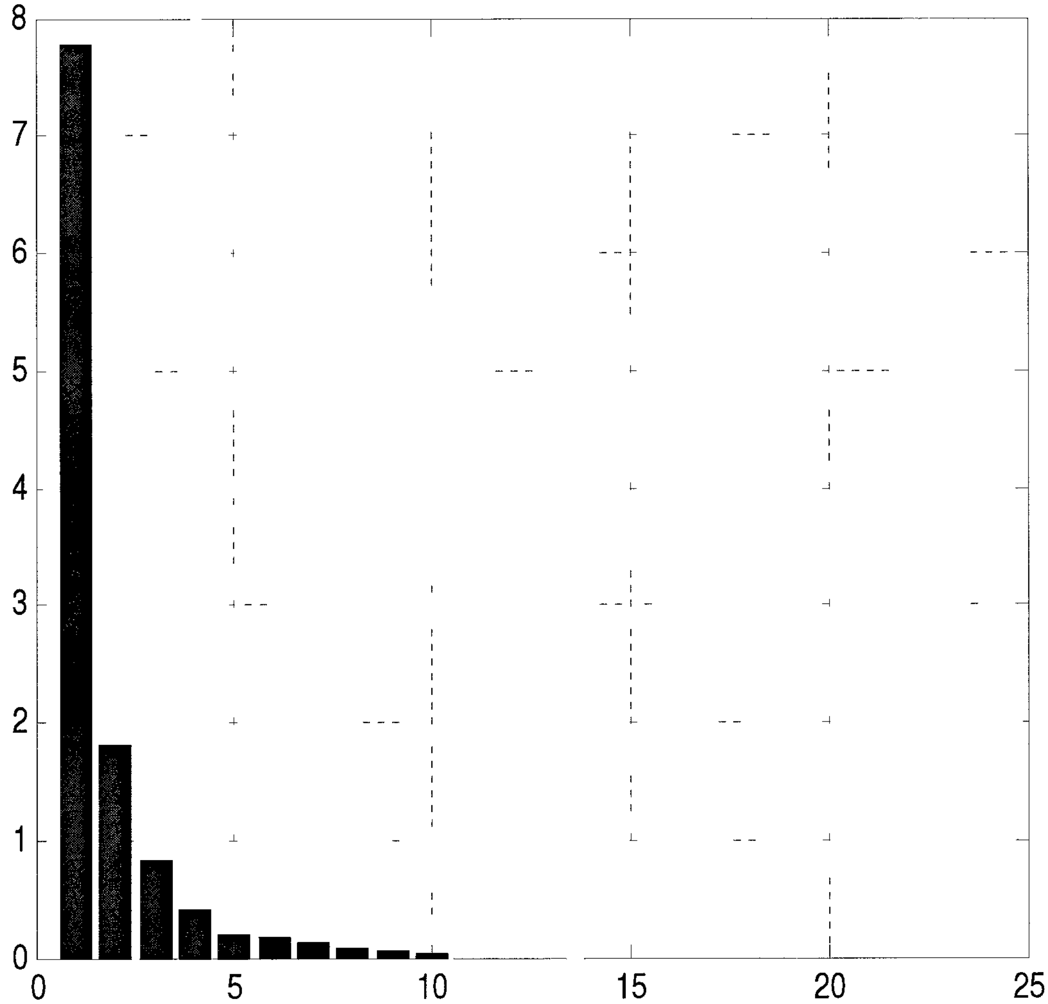


Figure (4.5) the singular values of the 17^{th} order low pass FIR

4.4 Advantages of SVD

1-The stability of 2-D filter is guaranteed if the 1-D subfilters are stable.

2-The design can be accomplished by designing a set of 1-D subfilters and, therefore, the many well-established techniques for the design of 1-D filters can be employed.

3-The sensitivity of the structure to coefficient quantization is expected to be low and as result reduction is introduced in the amplitude and phase response of the filter.

4-Computations in the various parallel subfilters can be carried out simultaneously and they less complex than in other 2-d filters design method.

5-The number of multipliers (or multiplications per output sample) is small. For example, if there are k parallel subfilters and each was of order N x N, then the upper bound on the number of multipliers would be 4k(N + 1) as opposed to (kN + 1)2 + 2 kN in the case of a corresponding state-space implementation.

6-The SVD of the matrix representation of a digital filter usually decrease rapidly, therefore, matrix can be represented with less subsections than its order.

7-Subsections corresponding to lower SV's value are eliminated.

4.5 Improved design

Assume A is a symmetric, and assume the error in all sections is zero except in section k so the mean square error will be

$$E_1 = \sigma_k \left\| \phi_k \phi_k^{'} - \gamma_k \gamma_k^{'} \right\| \qquad (4.27)$$

Now, let us assume the error in all sections is zero except in section j so the mean square error will be

$$E_2 = \sigma_j \left\| \phi_j \phi_j^{'} - \gamma_j \gamma_j^{'} \right\| \qquad (4.28)$$

Assume that $E_1 = E_2$ then we can write

$$\frac{\sigma_j}{\sigma_k} = \frac{\left\| \phi_k \phi_k^{'} - \gamma_k \gamma_k^{'} \right\|}{\left\| \phi_j \phi_j^{'} - \gamma_j \gamma_j^{'} \right\|} \qquad (4.29)$$

If $\sigma_j > \sigma_k$ then in order to have equal error in both sections, the error in section k has to be greater than in section j by the factor σ_j / σ_k . Therefore, the overall error is more sensitive to sections with greater singular values. As a result, an improvement can be achieved by varying the orders of 1-D filters in the sections in accordance with the significance of their corresponding singular values. Higher order subfilters are chosen for larger singular values and lower order subfilters are associated with singular values with smaller magnitude. To make the overall delays are equals for all sections, additional delay elements are added in each section. Therefore, the number of filter coefficients is reduced.

4.6 Examples

Example (3)

Design 4^{th} order 2-D IIR low pass filter specified by:

$$\left|H(e^{j\omega_1 T_1}, e^{j\omega_2 T_2})\right| = \begin{cases} 1 & 0 \le \sqrt{\omega_1^2+\omega_2^2} \le 0.45\pi \\ 0 & 0.45\pi < \sqrt{\omega_1^2+\omega_2^2} \le \pi \end{cases}$$

$$where \quad T_1 = T_2 = 1$$

M=L=41

Five sections are used

For this example, we will do the following:

Increase the order of the first section by 2 and become 6.

Decrease the order of the second section by 2 and become 4.

Decrease the order of the third, the fourth and the fifth section by 2 and they become 2.

Now, we add delay elements to each section as following:

First section we add nothing

Second section, the order 4 so we add the total 4.

Third, fourth, and fifth section, the order 2 so we add the total of 8.

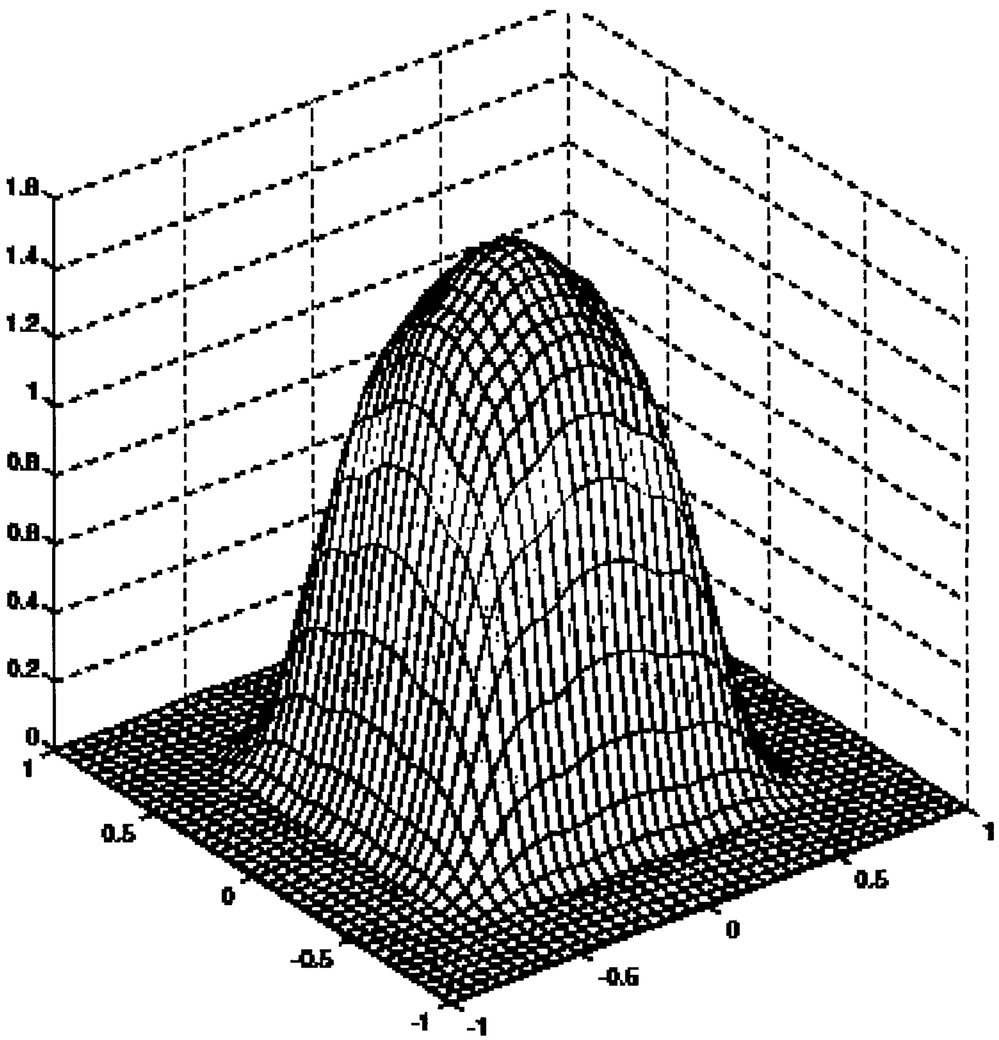


Figure (4.6) the 4th order low pass IIR by improved method

IIR	S1	S2	S3	S4	S5	Total	Reduction
Regular method	20	20	20	20	20	100	16
Improved method 6,4,2,2,2	28	20	12	12	12	84	16%

Table (4.1) coefficients comparison between regular SVD and improved SVD of 4^{th} order IIR

MSE IIR	Regular method	Improved method
Order of subfilters	4-4-4-4-4	6-4-2-2-2
PASSBAND	.0061	.00509
STOPBAND	.0042	.00419

Table (4.2) error comparison between improved and regular SVD of 4^{th} order IIR

Table (4.1) and table (4.2) show that a reduction of 16% in coefficients achieved by improved method with small error.

Example (4)

Design 17^{th} order 2-D FIR low pass filter specified by:

$$\left|H(e^{j\omega_1 T_1}, e^{j\omega_2 T_2})\right| = \begin{cases} 1 & 0 \le \sqrt{\omega_1^2+\omega_2^2} \le 0.35\pi \\ 0 & 0.65\pi \le \sqrt{\omega_1^2+\omega_2^2} \le \pi \end{cases}$$

$where \quad T_1 = T_2 = 1$

M=L=21

Five sections are used.

For this example, we will do the following:

Increase the order of the first section by 2 and become 19.

Decrease the order of the second section by 4 and become 15.

Decrease the order of the third section by 6 and become 13.

Decrease the order of the fourth section by 8 and become 11.

Decrease the order of the fourth section by 10 and become 9.

Now, we add delay elements to each section as following:

First section we add nothing.

Second section, we have two 1-D filters of order 15 so we add four for each 1-D filters and the total of eight in the section.

Third section, the order is 13 so we add the total of 12.

Fourth section, the order is 11 so we add the total of 16.

Fourth section, the order is 9 so we add the total of 20.

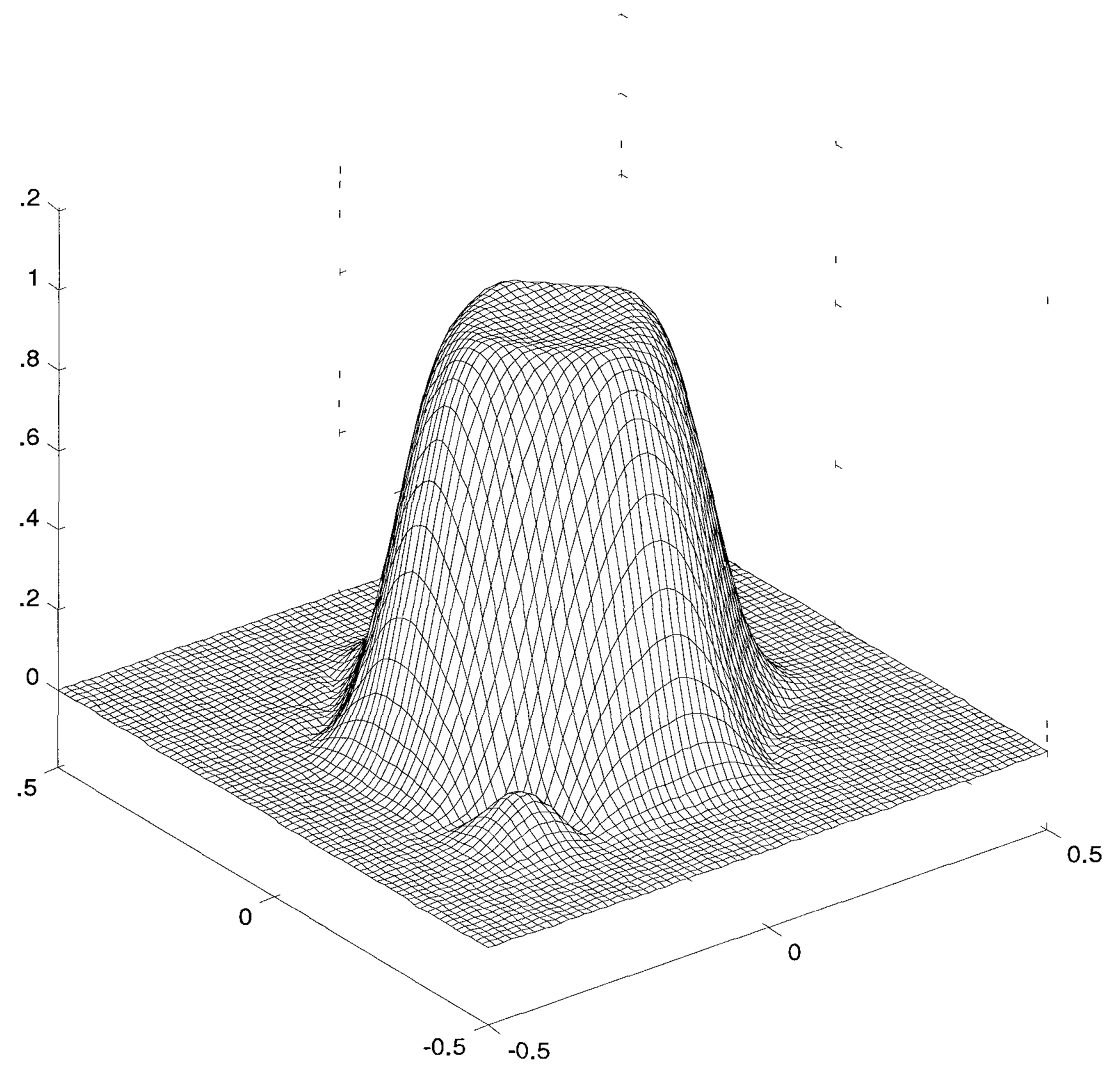


Figure (4.7) the 17^{th} order low pass FIR by improved method

FIR	S1	S2	S3	S4	S5	total	Reduction
Regular method	36	36	36	36	36	180	36
Improved method 19,15,13,11,9	40	32	28	24	20	144	20%

Table (4.3) coefficients comparison between improved and regular SVD of 17^{th} order FIR

MSE FIR	Regular method	Improved method
Order of subfilters	17-17-17-17-17	19-15-13-11-9
PASSBAND	.0032	.003175
STOPBAND	.00054	.0005102

Table (4.4) error comparison between improved and regular SVD of 17^{th} order FIR

Table (4.3) and table (4.4) show that a reduction of 20% in coefficients achieved by improved method with small error.

Example (5)

Design 15th order 2-D FIR Band pass filter specified by:

$$\left|H(e^{j\omega_1 T_1}, e^{j\omega_2 T_2})\right| = \begin{cases} 0 & 0 \le \sqrt{\omega_1^2 + \omega_2^2} \le 0.24\pi \\ 1 & 0.36\pi \le \sqrt{\omega_1^2 + \omega_2^2} \le 0.64\pi \\ 0 & 0.76\pi \le \sqrt{\omega_1^2 + \omega_2^2} \le \pi \end{cases}$$

$where \quad T_1 = T_2 = 1$

M=L=36

Seven sections are used

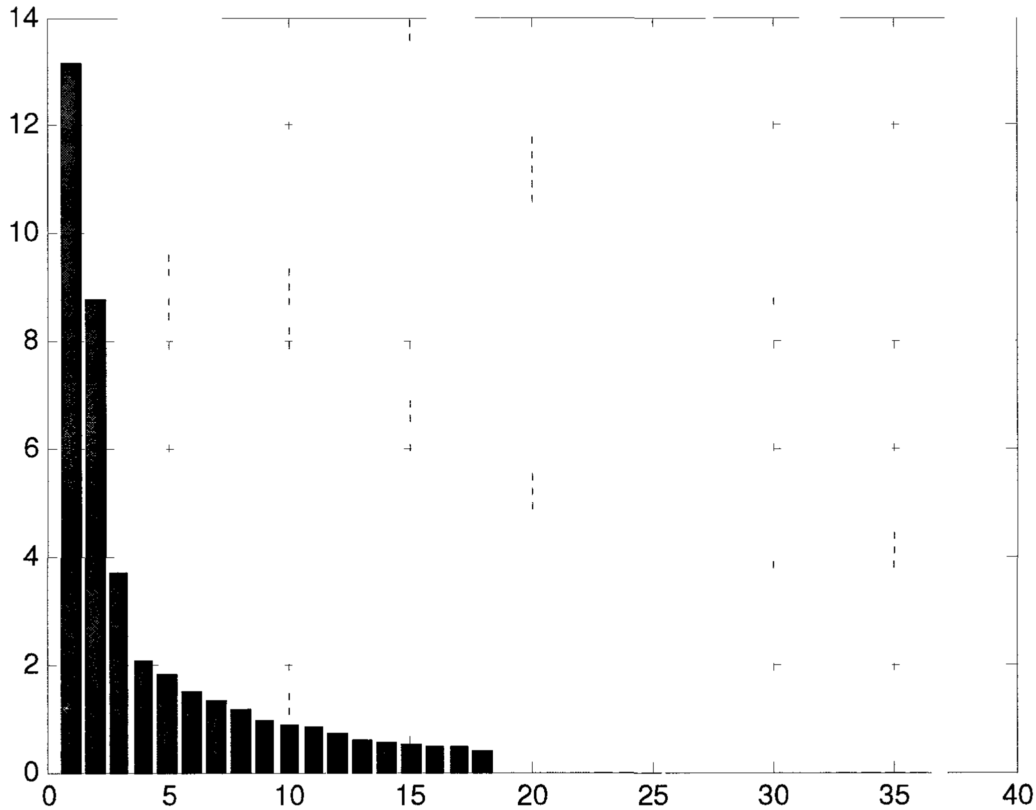


Figure (4.8) the singular values of the 15^{th} order band pass FIR

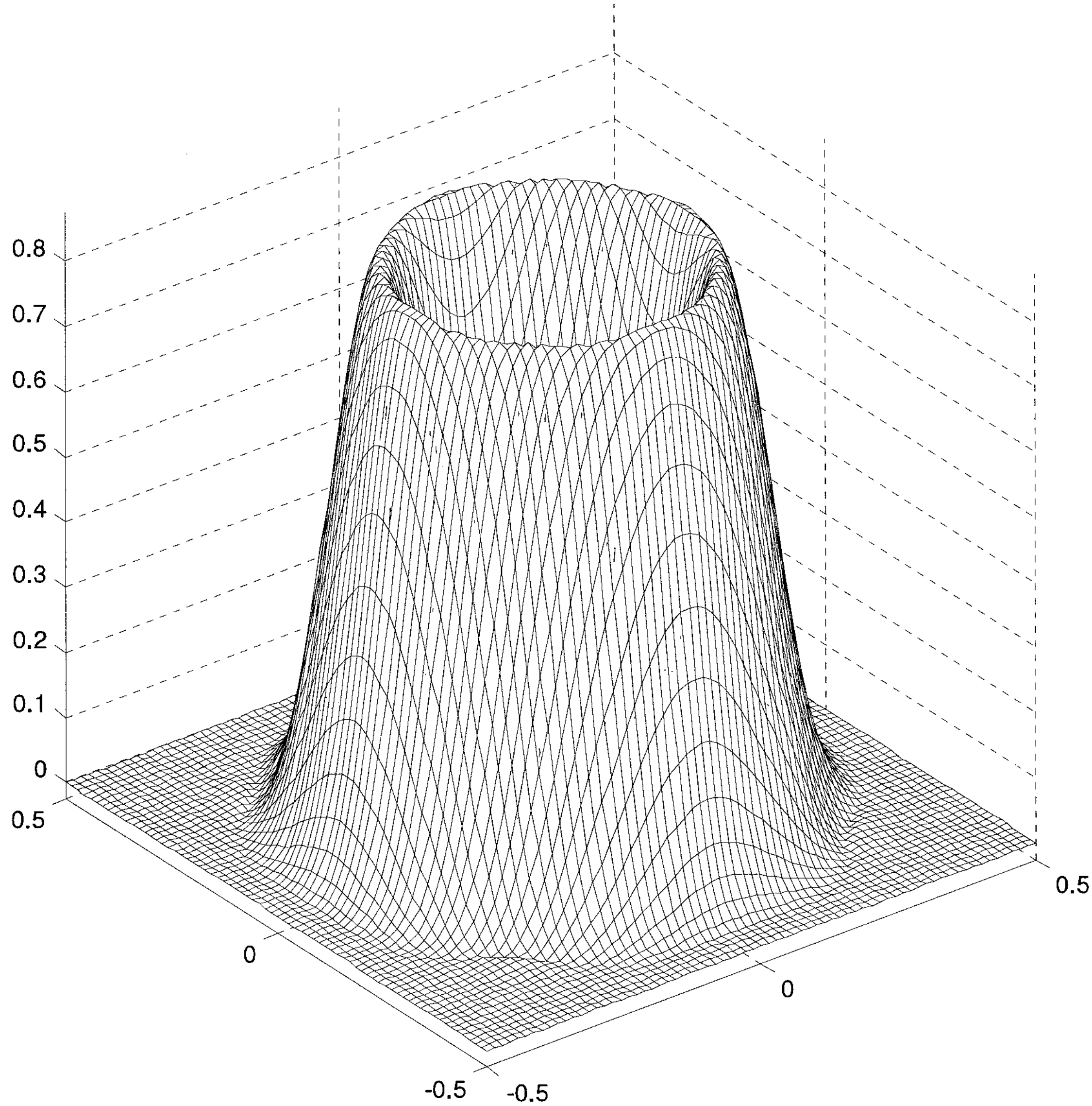


Figure (4.9) the 15^{th} order band pass FIR by regular method

Design by improved method

For this example, we will do the following:

Increase the order of the first section and second section by 2 and become 17 because the singular values corresponding to the first and second section are high.

Decrease the order of the third section by 4 and become 13.

Decrease the order of the fourth section by 6 and become 11.

Decrease the order of the fifth section by 8 and become 9.

Decrease the order of the sixth section by 10 and become 7.

Decrease the order of the seventh section by 12 and become 5.

Now, we add delay elements to each section as following:

First and second section we add nothing.

Third section, we add the total of eight in the section.

Fourth section, the order is 11 so we add the total of 12.

Fifth section, the order is 9 so we add the total of 16.

Sixth section, the order is 7 so we add the total of 20.

Seventh section, the order is 5 so we add the total of 24.

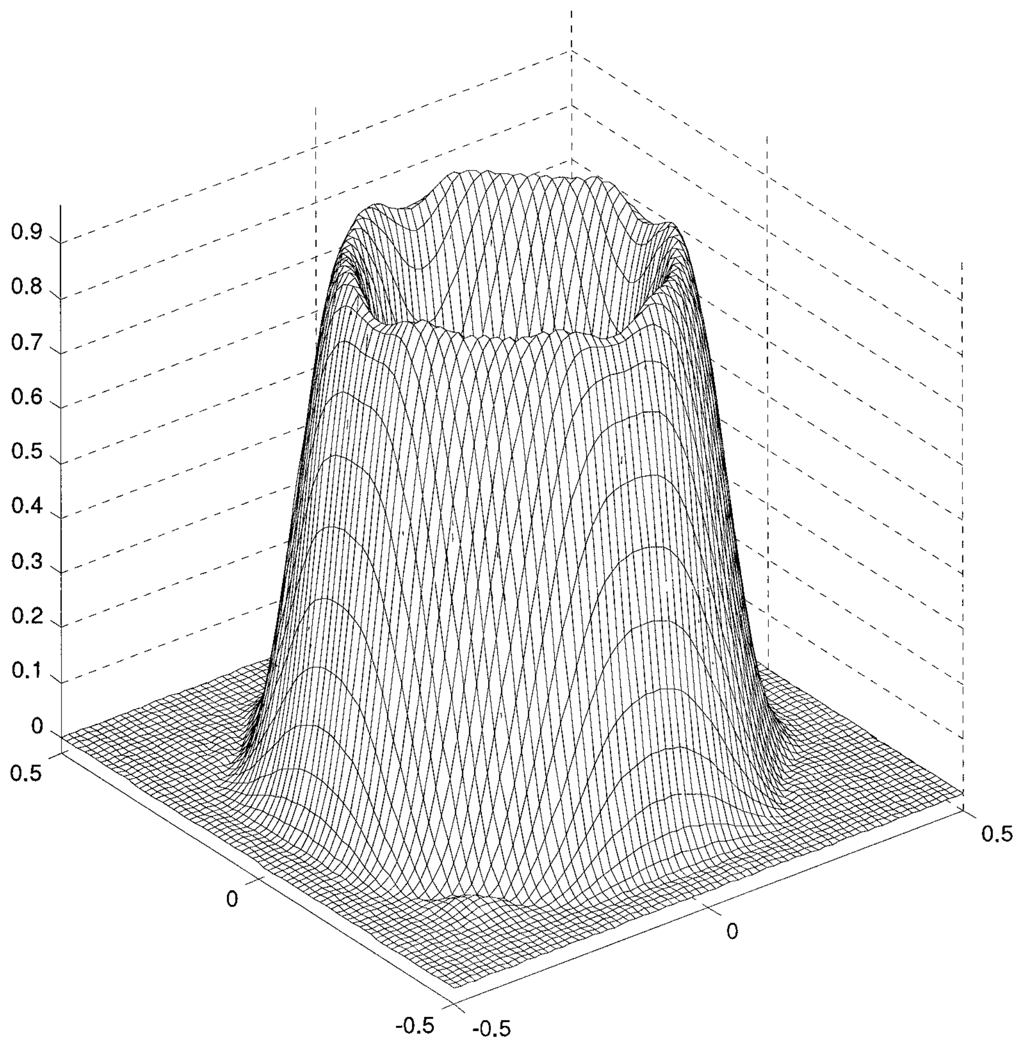


Figure (4.10) the 15th order band pass FIR by improved method

15^{th} FIR	S1	S2	S3	S4	S5	S6	S7	total	Reduction
Regular method	32	32	32	32	32	32	32	224	52
Improved method 17,17,13,11,9,7,5	36	36	28	24	20	16	12	172	23.21%

Table (4.5) coefficients comparison between improved and regular SVD of 15^{th} order band pass FIR

MSE FIR	Regular method	Improved method
Order of subfilters	15-15-15-15-15-15-15	17-17-13-11-9-7-5
PASSBAND	1.0672e-003	1.69e-003
STOPBAND	1.6e-003	1.023e-003

Table (4.6) error comparison between improved and regular SVD of 15^{th} order band pass FIR

Table (4.3) and table (4.4) show that a reduction of 23.21% in coefficients achieved by improved method with a slightly higher error.

Example (6)

Design 31^th^ order 2-D FIR High pass filter specified by:

$$\left|H\left(e^{j\omega_1 T_1}, e^{j\omega_2 T_2}\right)\right| = \begin{cases} 0 & 0 \le \sqrt{\omega_1^2 + \omega_2^2} \le 0.4\pi \\ 1 & 0.6\pi \le \sqrt{\omega_1^2 + \omega_2^2} \le \pi \end{cases}$$

$$where \quad T_1 = T_2 = 1$$

M=L=21

Seven sections are used

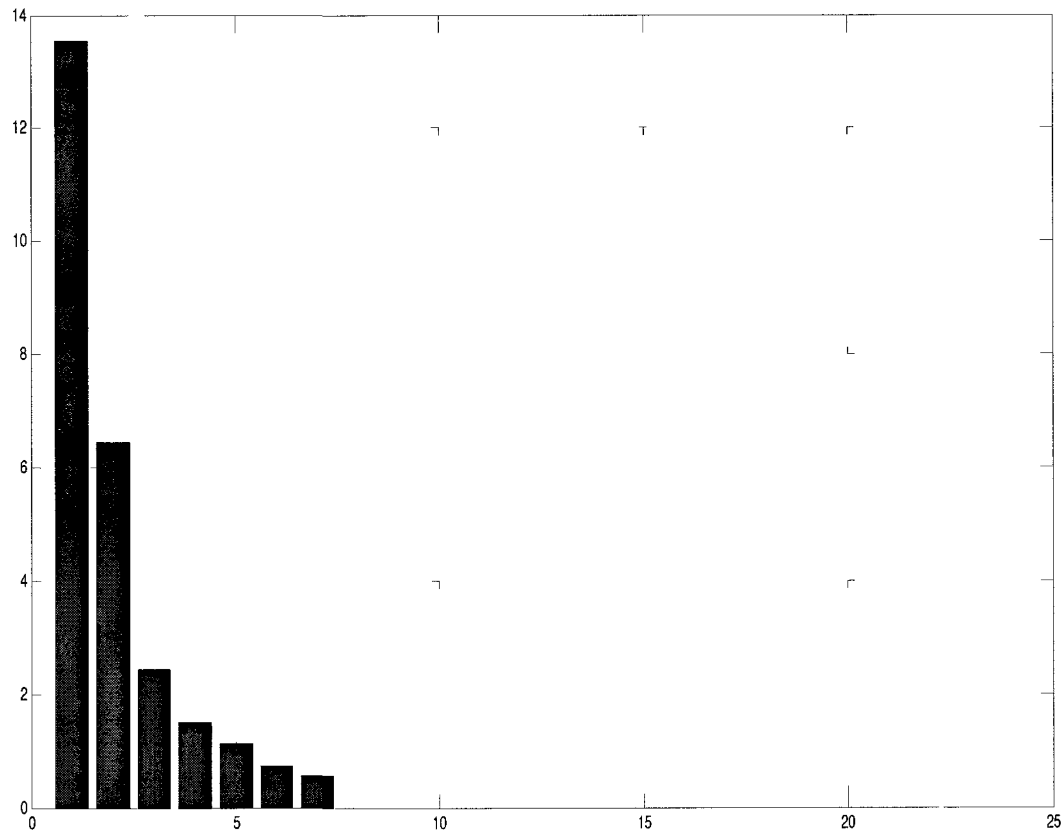


Fıgure (4.11) the sıngular values of the 31st order hıgh pass FIR

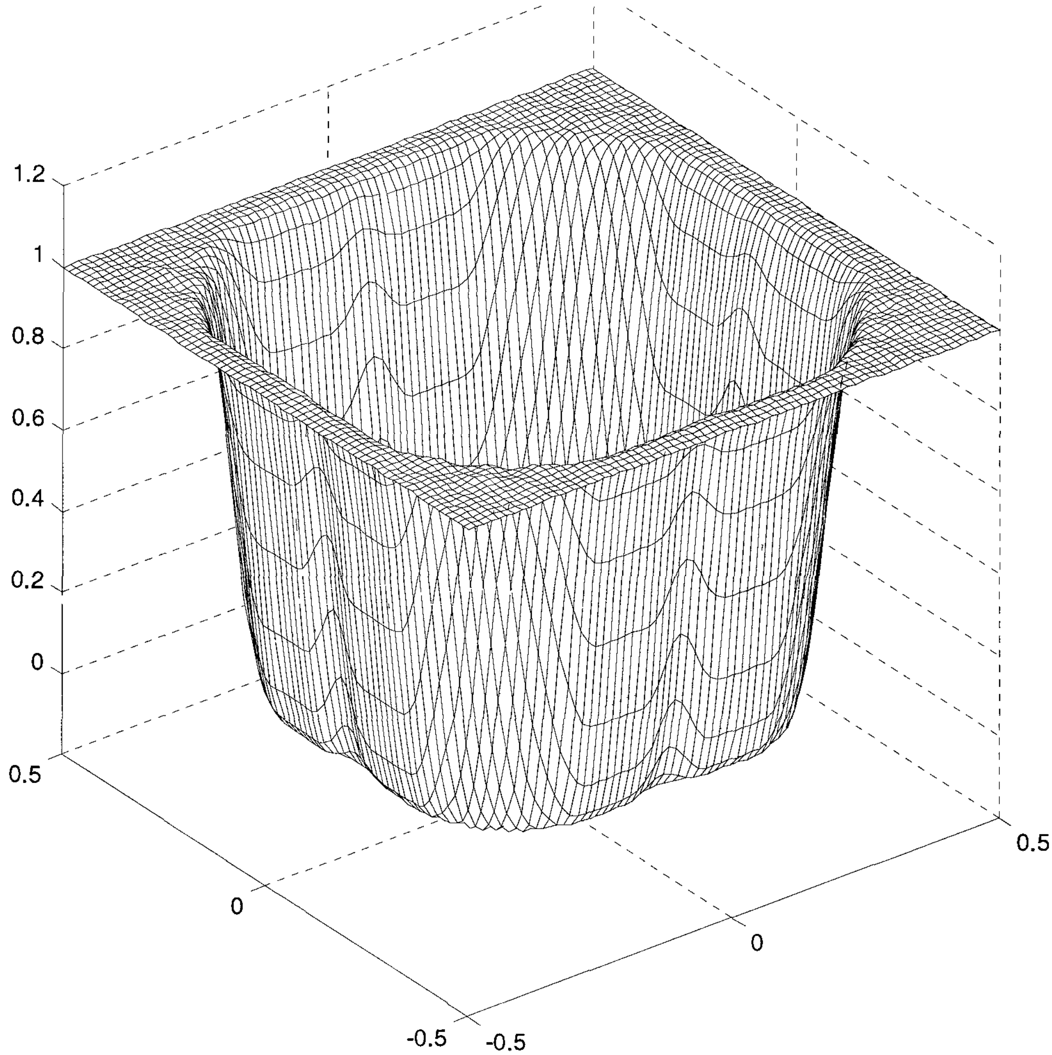


Figure (4.12) The 31s order high pass FIR by regular method

Design by improved method

For this example, we will do the following:

Increase the order of the first section and second section by 2 and become 33 because the singular values corresponding to the first and second section are high.

Decrease the order of the third section by 4 and become 29.

Decrease the order of the fourth section by 6 and become 27.

Decrease the order of the fifth section by 8 and become 25.

Decrease the order of the sixth section by 10 and become 23.

Decrease the order of the seventh section by 12 and become 21.

Now, we add delay elements to each section as following:

First and second section we add nothing.

Third section, we add the total of eight in the section.

Fourth section, the order is 27 so we add the total of 12.

Fifth section, the order is 25 so we add the total of 16.

Sixth section, the order is 23 so we add the total of 20.

Seventh section, the order is 21 so we add the total of 24.

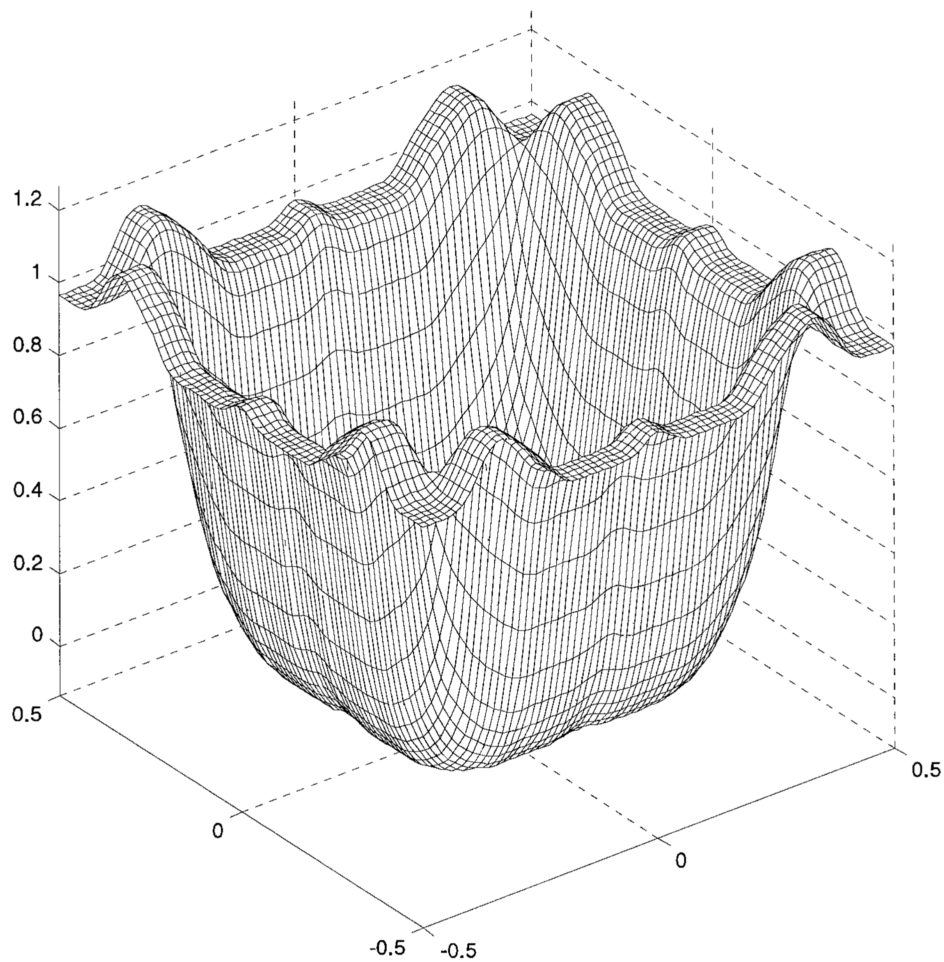


Figure (4.13) The 31s order high pass FIR by improved method

31st FIR	S1	S2	S3	S4	S5	S6	S7	Total	Reduction
Regular method	64	64	64	64	64	64	64	448	52
Improved method 33,33,29,27, 25,23,21	68	68	60	56	52	48	44	396	11,61%

Table (4.7) error comparison between improved and regular SVD of 31st order high pass FIR

MSE FIR	Regular method	Improved method
Order of subfilters	31X7	33-33-29-27-25-23-21
PASSBAND	.0017	.0035
STOPBAND	.0044	.0091

Table (4.8) error comparison between improved and regular SVD of 31^{st} order high pass FIR

Table (4.7) and table (4.8) show that a reduction of 11.61% in coefficients achieved by improved method with a slightly higher error.

Example (7)

Design 37th order low pass FIR filter

$$\left|H\left(e^{j\omega_1 T_1}, e^{j\omega_2 T_2}\right)\right| = \begin{cases} 1 & 0 \le \sqrt{\omega_1^2+\omega_2^2} \le 0.4\pi \\ 0 & 0.4\pi < \sqrt{\omega_1^2+\omega_2^2} \le \pi \end{cases}$$

In this example, we will design the filter with five different orders as following:

First 37-37-37-37

Second 39-29-21-13

Third 41-29-21-13

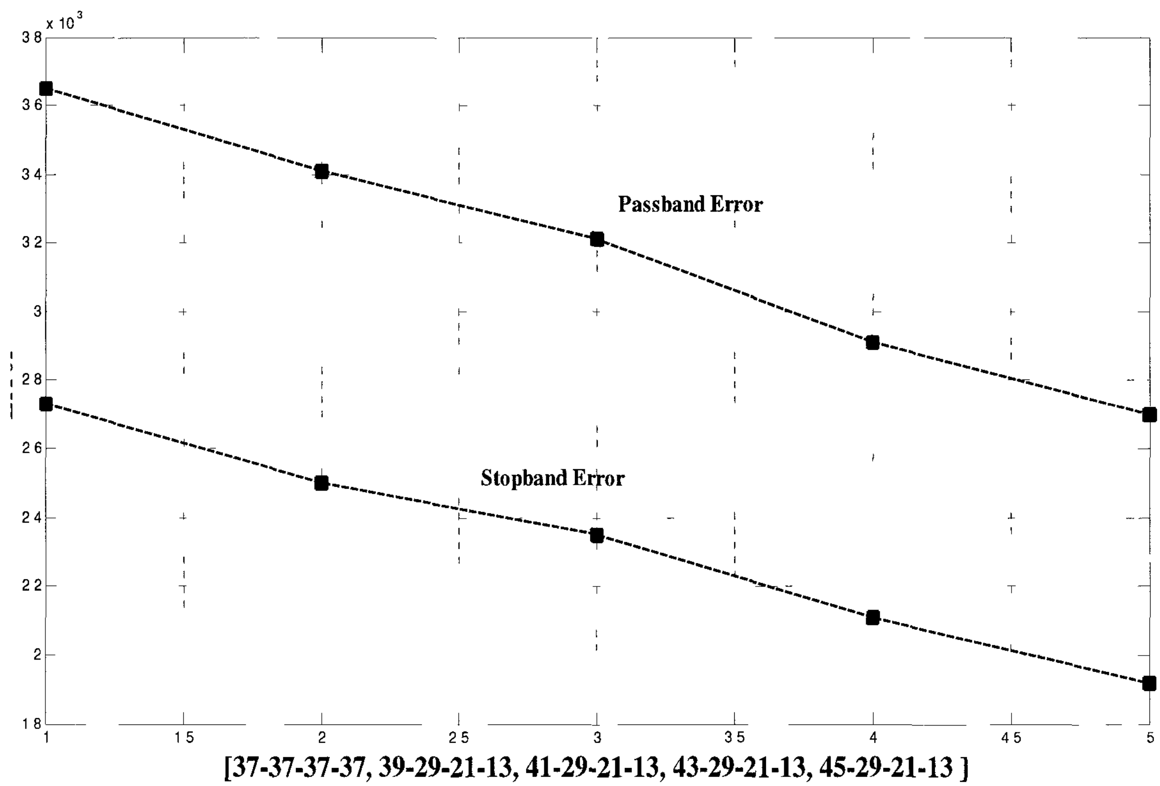
Fourth 43-29-21-13

Fifth 45-29-21-13

The following figure shows comparision between the five filters in passband and stopband

37-37-37-37	304	Reduction	
39-29-21-13	212	92	30.26 %
41-29-21-13	216	88	28.95 %
43-29-21-13	220	84	27.63 %
45-29-21-13	224	80	26.32 %

Comparision between the five filters design and the number of reduction atcheived


x 10^3
3 8
3 6
3 4
3 2
3
2 8
2 6
2 4
2 2
2
1 8
1
1 5
2
2 5
3
3 5
4
4 5
5
Passband Error
Stopband Error
[37-37-37-37, 39-29-21-13, 41-29-21-13, 43-29-21-13, 45-29-21-13]

4.7 Conclusion

In this chapter, 2-D digital filters have been designed by regular and improved SVD method. A reduction in coefficients 11% to 23 % was obtained with acceptable error.

DESIGN 2-D DIGITAL FILTERS BY GENETIC ALGORITHM

5.1 Design digital filters by genetic algorithm

In this chapter, we will use the combination of SVD and GA to design 2-D digital filters. The design technique is the same as in chapter four but when it comes to design 1-d subfilters, we will use GA instead of classical optimization methods to design these 1-d subfilters. Therefore, GA is only used to design 1-D digital filters.

5.2 Modification of GA

Some modifications have to be made to make GA more suitable for our design needs. These modifications as following:

5.2.1 Initial population

GA starts by initializing a random population whose chromosomes represents the coefficients of a filter transfer function and these coefficients are encoded to CSD format. Each chromosome is constructed by concatenating all the filter coefficients in a transfer function and each chromosome has a 16 bit word length and maximum of four non-zero digits. For N order IIR filter, each chromosome presents the coefficients of IIR transfer function as following:

$$a_0 \; a_1 \; a_2 \ldots\ldots a_n, b_0 \; b_1 \; b_2, \ldots. b_n$$

For N order FIR filter, each chromosome presents coefficients of FIR transfer function.

$$a_0 \;\; a_1 \;\; a_2 \ldots\ldots a_n$$

Throughout this thesis, the population size is 80 nd the maximum iterations is 100.

5.2.2 Fitness function

The least mean square is used to form the objective function. The magnitude response of a filter function is determined by evaluating the transfer function with $z = e^{jwt}$ over a specific frequency range. The LMS error function is formulated by comparing the magnitude response of the transfer function at each frequency to the desired magnitude response at that frequency then the resulted error value is squared and summed with the square of the error value at other frequency. The fitness value is calculated by inversing the LMS error value.

$$E_{mg} = \left|H_I(e^{j\omega})\right| - \left|H_D(e^{j\omega})\right|$$

$$E = \sum E_{mg}^2(jwt)$$

$$fitness = \frac{1}{E}$$

5.2.3 Reproduction, Crossover and mutation operations

The Roulette Wheel is used to execute reproduction operation.

To execute the crossover operation, the one-point crossover is used with crossover probability of 90%. The mutation probability is 1%.

5.2.4 Elitist operation

The idea of crossover and mutation operations is to create new chromosomes (offspring) fitter than their parents. However, there is no guarantee that will happen all time. Crossover and mutation operations can produce offspring that are highly unfit. These unfit chromosomes may make GA get worse as it progress. Elitism is a technique implemented to prevent the loss of the fittest chromosomes due to crossover and mutation operations. Elitism ensures that the best chromosomes are always kept. So if crossover and mutation operations yield less fit offspring, their parents are held over to the next generation.

5.2.5 CSD restoration techniques

The offspring created by crossover and mutation operations might violate the CSD constraints. Therefore, after each crossover and mutation operation, the new chromosomes (offspring) will be checked for violations and in case there are some, chromosomes will be restored to their nearest CSD number. One way to restore the violated chromosomes is by decoding these chromosomes to their decimal number then converts them back to their nearest CSD representation.

$$
\begin{array}{ll}
parent_1 & 0\ 1\ 0\ 1\ 0\ 1 \\
parent_2 & 1\ 0\ 1\ 0\ 0\ 1 \\
offspring_1 & 0\ 1\ 1\ 0\ 0\ 1
\end{array}
$$

$$
\begin{array}{ll}
offspring & 0\ 1\ 0\ 1\ 0\ 1 \\
mutation & 0\ 1\ 0\ 1\ 1\ 1
\end{array}
$$

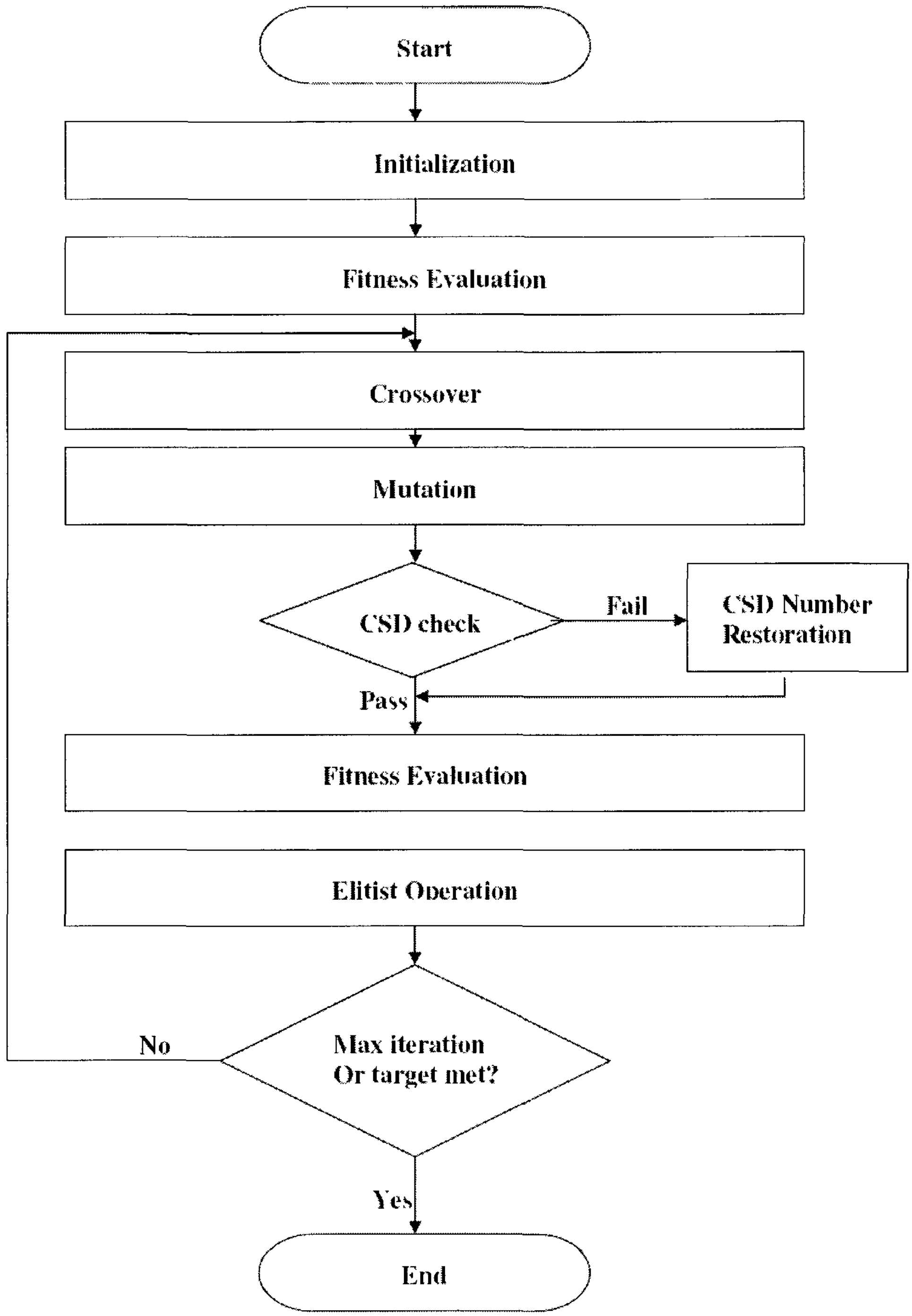


Figure (5.1) flow chart of modified GA

5.3 Examples

Example (1)

Design 4th order 2-D IIR low pass filter specified by:

$$\left|H\left(e^{j\omega_1 T_1}, e^{j\omega_2 T_2}\right)\right| = \begin{cases} 1 & 0 \le \sqrt{\omega_1^2+\omega_2^2} \le 0.45\pi \\ 0 & 0.45\pi < \sqrt{\omega_1^2+\omega_2^2} \le \pi \end{cases}$$

$$where \quad T_1 = T_2 = 1$$

M=L=41

Five sections are used

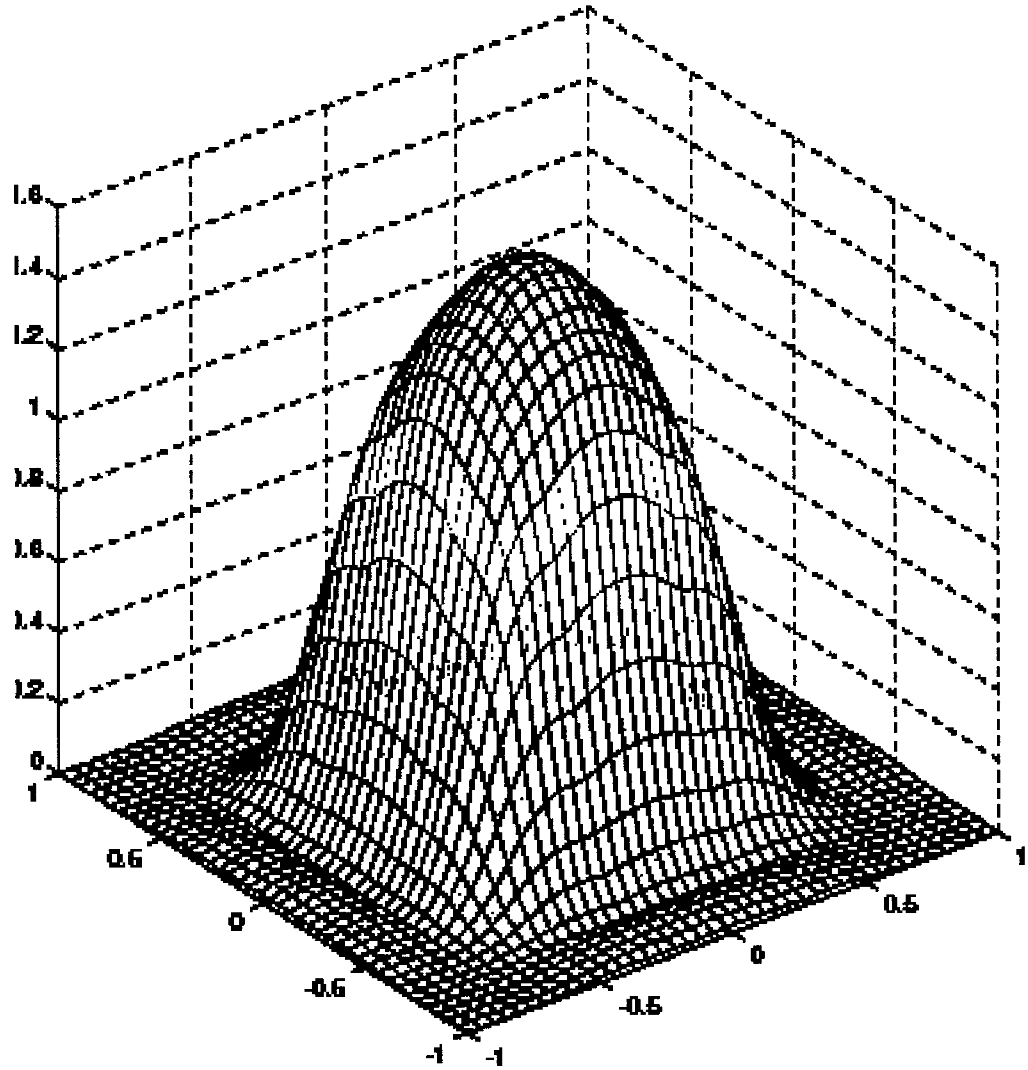


Figure (5.2) the 4^{th} order low pass IIR by regular method

The numerator coefficients of 1-D filter (f1)of the first section in decimal and CSD																
0.2354	0	1	0	0	0	-1	0	0	0	1	0	0	0	1	0	0
0.0093	0	0	0	0	0	0	1	0	1	0	-1	0	0	1	0	0
-0.3603	0	0	1	0	0	1	0	0	0	-1	0	0	0	0	1	0
0.0093	0	0	0	0	0	0	1	0	1	0	-1	0	0	0	0	1
0.2354	0	1	0	0	0	-1	0	0	0	1	0	0	0	1	0	0

The denominator coefficients of 1-D filter (f1)of the first section in decimal and CSD																
1.0000	1.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.1012	0	0	1	0	-1	0	1	0	0	0	0	-1	0	0	0	0
0.3928	0	0	-1	0	0	1	0	0	1	0	0	1	0	0	0	0
0.0377	0	0	0	0	1	0	1	0	-1	0	1	0	0	0	0	0
0.1831	0	1	0	-1	0	0	0	-1	0	0	-1	0	0	0	0	0

Table (5.1) The numerator and denominator coefficients of 1-D filter (f1) of the first section of the 4^{th} order IIR regular method

Design by Improved method

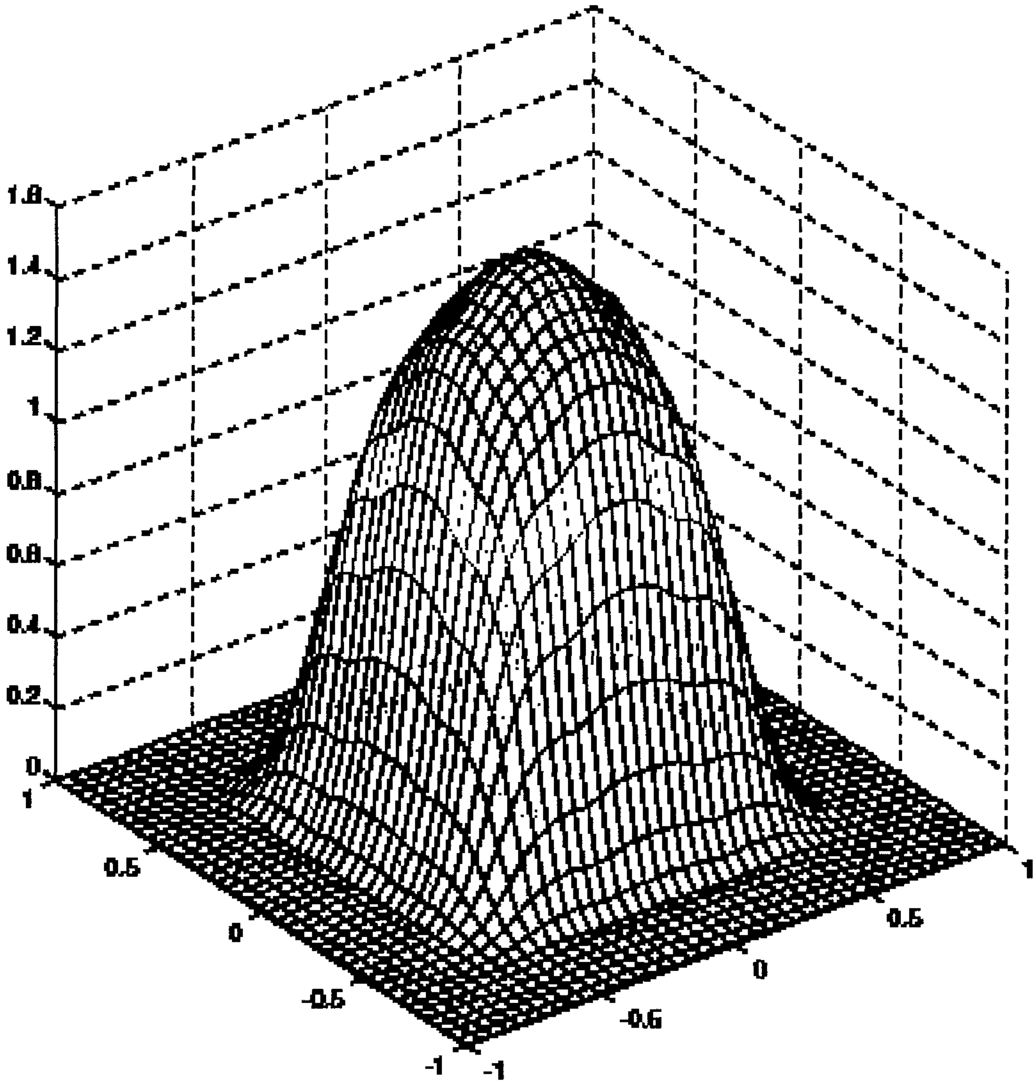


Figure (5.3) the 4th order low pass IIR by improed method

The numerator coefficients of 1-D filter (f1) of the first section in decimal and CSD																
0.1685	**0**	**1**	**0**	**-1**	**0**	**-1**	**0**	**-1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
0.0121	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**0**	**1**	**0**	**-1**	**0**	**0**	**0**
-0.3553	**0**	**0**	**1**	**0**	**0**	**1**	**0**	**1**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**
-0.00067	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**-1**	**0**	**1**	**0**	**1**	**0**	**1**
0.3546	**0**	**0**	**-1**	**0**	**0**	**-1**	**0**	**-1**	**0**	**-1**	**0**	**0**	**0**	**0**	**0**	**0**
-0.0115	**0**	**0**	**0**	**0**	**0**	**-1**	**0**	**1**	**0**	**0**	**0**	**1**	**0**	**0**	**-1**	**0**
-0.1685	**0**	**-1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

The denominator coefficients of 1-D filter (f1)of the first section in decimal and CSD																
1.0000	**1.**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
0.1281	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**0**	**1**	**0**	**0**	**0**
0.2591	**0**	**1**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**1**	**0**	**1**	**0**	**0**	**0**	**0**
0.0456	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**0**	**0**	**-1**	**0**	**-1**	**0**	**0**	**0**	**0**
0.2204	**0**	**1**	**0**	**0**	**-1**	**0**	**0**	**0**	**1**	**0**	**0**	**-1**	**0**	**0**	**0**	**0**
0.0031	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**1**	**0**	**-1**	**0**	**0**
-0.0295	**0**	**0**	**0**	**0**	**-1**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**-1**	**0**	**-1**	**0**

Table (5.2) The numerator and denominator coefficients of 1-D filter (f1) of the first section of the 4^{th} order IIR regular method

MSE IIR	Regular method	Improved method
Order of subfilters	4-4-4-4-4	6-4-2-2-2
PASSBAND	.0056	.00501
STOPBAND	.0042	.004147

Table (5.3) error comparison between improved and regular SVD of 4^{th} order IIR

Example (2)

Design 17^{th} order 2-D FIR low pass filter specified by:

$$\left|H(e^{j\omega_1 T_1}, e^{j\omega_2 T_2})\right| = \begin{cases} 1 & 0 \le \sqrt{\omega_1^2 + \omega_2^2} \le 0.35\pi \\ 0 & 0.65\pi \le \sqrt{\omega_1^2 + \omega_2^2} \le \pi \end{cases}$$

$$where \quad T_1 = T_2 = 1$$

M=L=21

Five sections are used

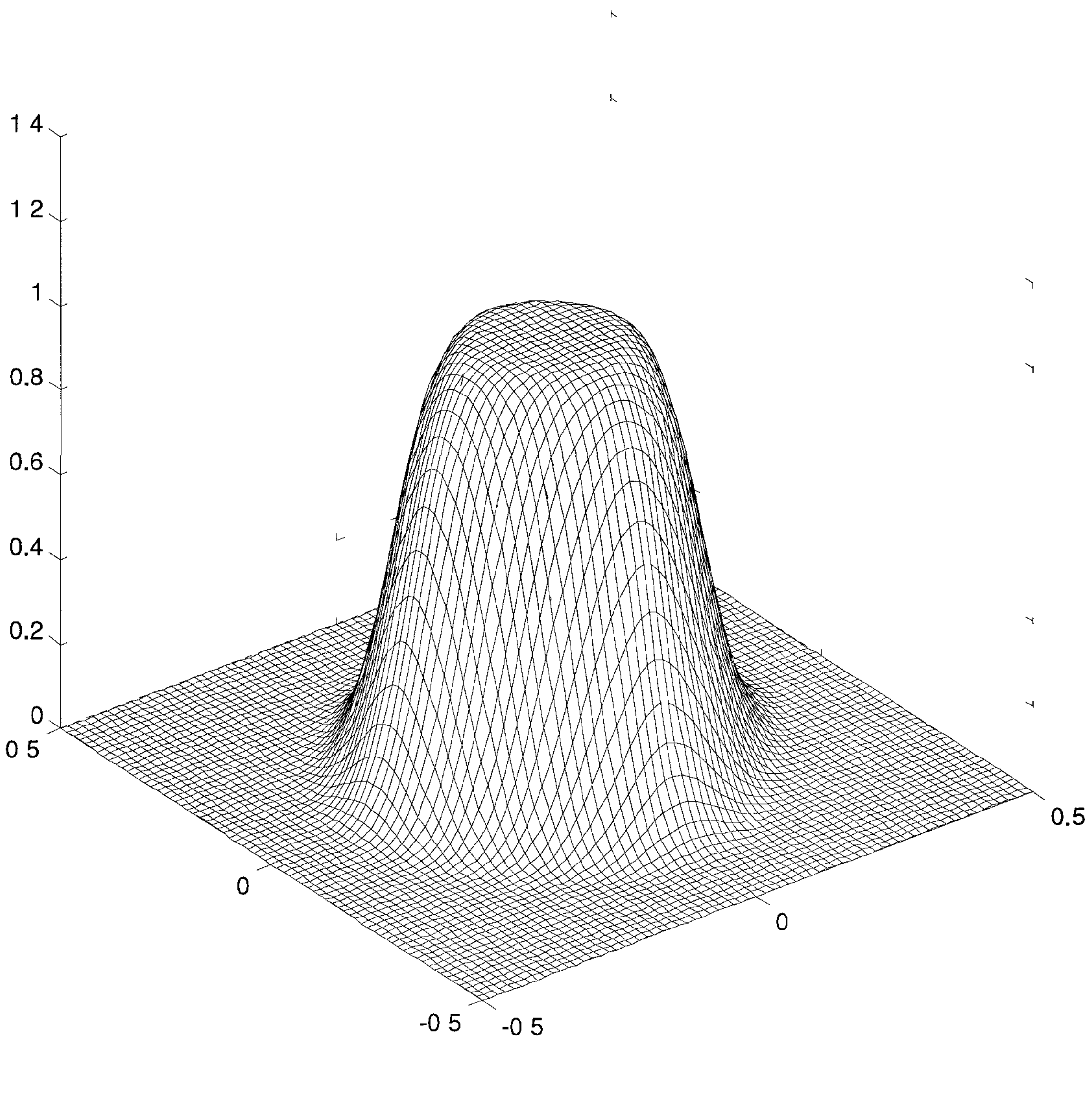


Figure (5 4) The 17th order low pass FIR by regular method

The coefficients of 1-D filter (f1)of the first section in decimal and CSD																
-0.0006	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**-1**	**0**	**1**	**0**	**1**	**0**	**1**
0.00067	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**-1**	**0**	**-1**
-0.00067	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**-1**	**0**	**1**	**0**	**1**	**0**	**1**
-0.0040	**0**	**0**	**0**	**0**	**-1**	**0**	**1**	**0**	**1**	**0**	**0**	**-1**	**0**	**0**	**0**	**0**
0.00152	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**0**	**1**	**0**	**-1**
0.0287	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**0**	**1**	**0**	**0**	**0**	**1**	**0**
0.0055	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**-1**	**0**	**0**	**0**	**1**
-0.1707	**0**	**0**	**0**	**0**	**0**	**-1**	**0**	**1**	**0**	**1**	**0**	**-1**	**0**	**0**	**0**	**0**
-0.3880	**0**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**0**	**0**	**0**	**-1**	**0**	**-1**	**0**	**0**
-0.3880	**0**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**0**	**0**	**0**	**-1**	**0**	**-1**	**0**	**0**
-0.1707	**0**	**0**	**0**	**0**	**0**	**-1**	**0**	**1**	**0**	**1**	**0**	**-1**	**0**	**0**	**0**	**0**
0.0055	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**-1**	**0**	**0**	**0**	**1**
0.0287	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**0**	**1**	**0**	**0**	**0**	**1**	**0**
0.00152	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**0**	**1**	**0**	**-1**
-0.0040	**0**	**0**	**0**	**0**	**-1**	**0**	**1**	**0**	**1**	**0**	**0**	**-1**	**0**	**0**	**0**	**0**
-0.00067	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**-1**	**0**	**1**	**0**	**1**	**0**	**1**
0.00067	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**-1**	**0**	**-1**
-0.00067	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**-1**	**0**	**1**	**0**	**1**	**0**	**1**

Table (5.4) The coefficients of 1-D filter (f1) of the first section of the 17^{th} order FIR regular method

Design by Improved method

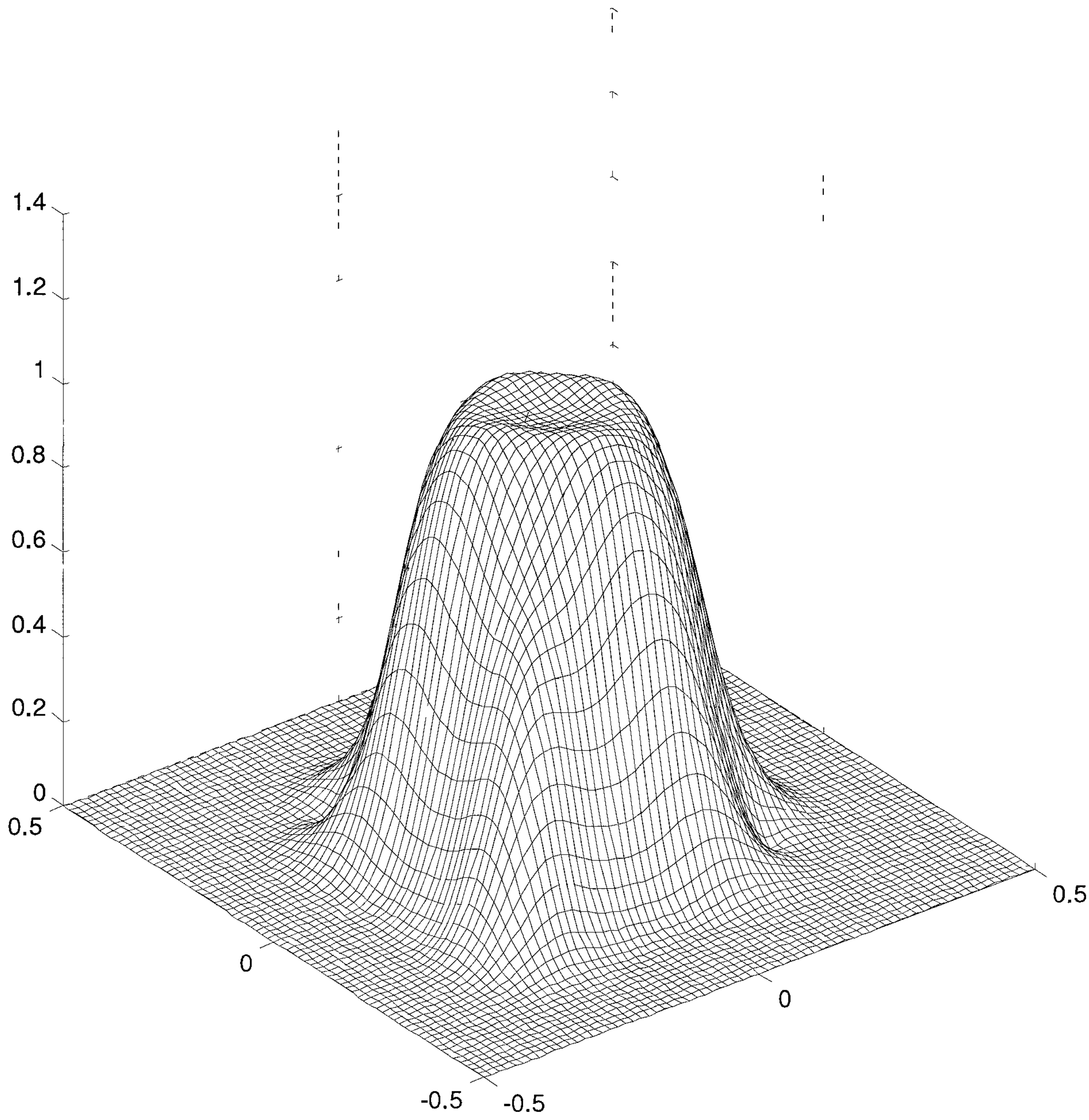


Figure (5.5) The 17^{th} order low pass FIR by improved method

The coefficients of 1-D filter (f1)of the first section in decimal and CSD																
-0.00067	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**-1**	**0**	**1**	**0**	**1**	**0**	**1**
-0.00067	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**-1**	**0**	**1**	**0**	**1**	**0**	**1**
0.00067	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**-1**	**0**	**-1**
-0.00065	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**-1**	**0**	**1**	**0**	**1**	**0**	**1**
-0.0052	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**-1**	**0**	**-1**	**0**	**-1**	**0**	**0**	**-1**	**0**
0.0016	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**1**	**0**	**0**	**-1**
0.0324	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**-1**	**0**	**0**
0.0057	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**0**	**0**	**-1**	**0**	**0**	**-1**
-0.1732	**0**	**-1**	**0**	**1**	**0**	**1**	**0**	**0**	**0**	**-1**	**0**	**0**	**0**	**0**	**0**	**0**
-0.3886	**0**	**0**	**1**	**0**	**0**	**-1**	**0**	**0**	**1**	**0**	**0**	**0**	**1**	**0**	**0**	**0**
-0.3886	**0**	**0**	**1**	**0**	**0**	**-1**	**0**	**0**	**1**	**0**	**0**	**0**	**1**	**0**	**0**	**0**
-0.1732	**0**	**-1**	**0**	**1**	**0**	**1**	**0**	**0**	**0**	**-1**	**0**	**0**	**0**	**0**	**0**	**0**
0.0057	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**0**	**0**	**-1**	**0**	**0**	**-1**
0.0324	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**-1**	**0**	**0**
0.0016	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**1**	**0**	**0**	**-1**
-0.0052	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**-1**	**0**	**-1**	**0**	**-1**	**0**	**0**	**-1**	**0**
-0.00065	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**-1**	**0**	**1**	**0**	**1**	**0**	**1**
0.00067	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**-1**	**0**	**-1**
-0.00067	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**-1**	**0**	**1**	**0**	**1**	**0**	**1**
-0.00067	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**-1**	**0**	**1**	**0**	**1**	**0**	**1**

Table (5.5) The coefficients of 1-D filter (f1) of the first section of the 17th order FIR imroved method

MSE FIR	Regular method	Improved method
Order of subfilters	17-17-17-17-17	19-15-13-11-9
PASSBAND	.0032	.003148
STOPBAND	.00054	.000501

Table (5.6) error comparison between improved and regular SVD of 17^{th} order FIR

Example (3)

Design 15th order 2-D FIR Band pass filter specified by:

$$\left|H(e^{j\omega_1 T_1}, e^{j\omega_2 T_2})\right| = \begin{cases} 0 & 0 \le \sqrt{\omega_1^2 + \omega_2^2} \le 0.24\pi \\ 1 & 0.36\pi \le \sqrt{\omega_1^2 + \omega_2^2} \le 0.64\pi \\ 0 & 0.76\pi \le \sqrt{\omega_1^2 + \omega_2^2} \le \pi \end{cases}$$

$where \quad T_1 = T_2 = 1$

M=L=36

Seven sections are used

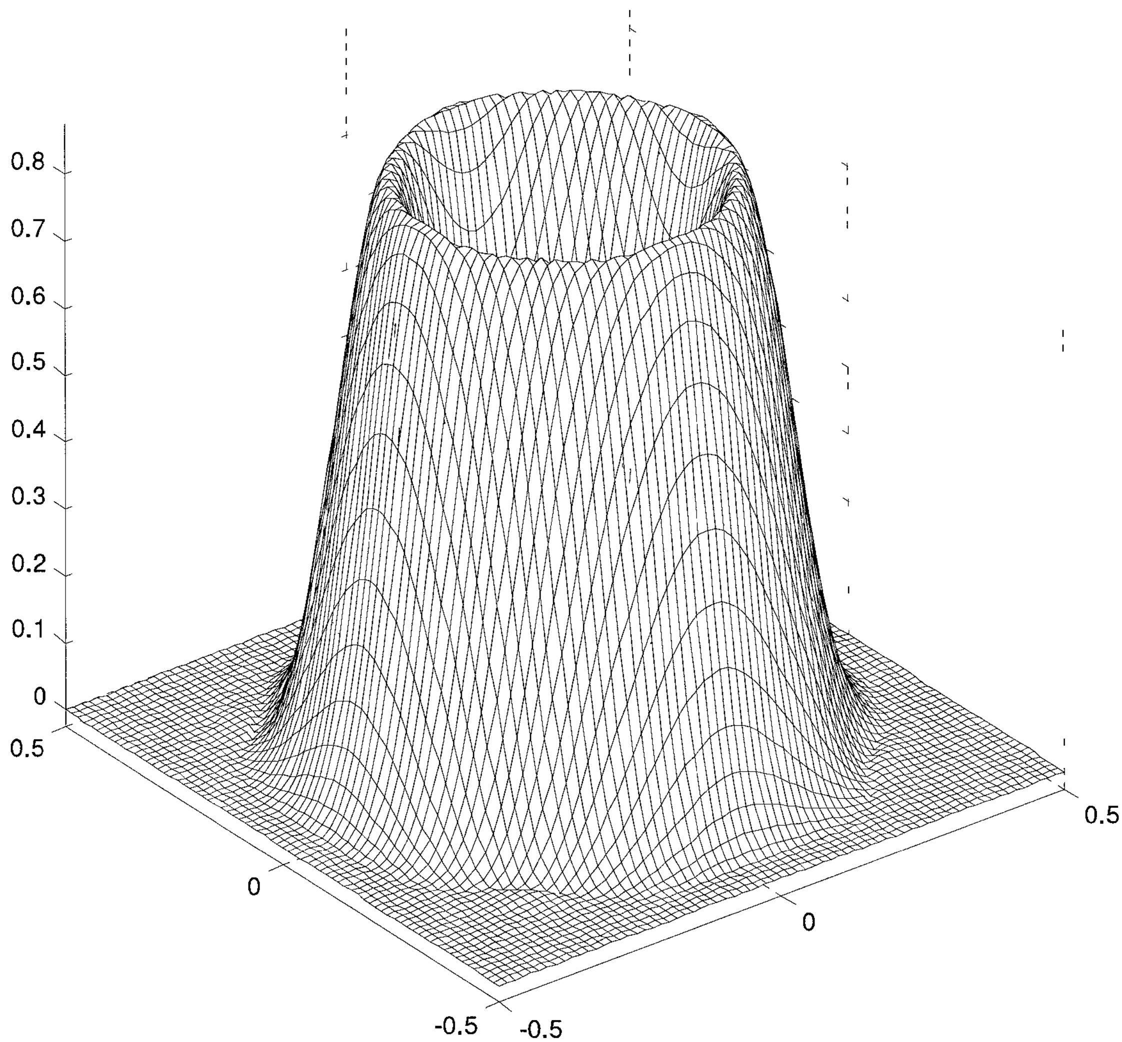


Figure (5.6) The 15^{th} order band pass FIR by regular method

The coefficients of 1-D filter (f1)of the first section in decimal and CSD																
0.0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.0018	0	0	0	0	0	0	0	0	1	0	0	-1	0	1	0	1
-0.0071	0	0	0	0	0	0	-1	0	0	1	0	-1	0	0	0	-1
-0.0200	0	0	0	0	0	-1	0	-1	0	0	-1	0	0	1	0	0
0.0637	0	0	0	1	0	0	0	0	0	1	0	1	0	0	0	-1
0.1021	0	0	1	0	-1	0	1	0	0	0	1	0	0	0	0	0
-0.2063	0	-1	0	1	0	-1	0	-1	0	0	0	0	0	0	0	0
-0.4682	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	1
-0.2063	0	-1	0	1	0	-1	0	-1	0	0	0	0	0	0	0	0
0.1021	0	0	1	0	-1	0	1	0	0	0	1	0	0	0	0	0
0.0637	0	0	0	1	0	0	0	0	0	1	0	1	0	0	0	-1
-0.0200	0	0	0	0	0	-1	0	-1	0	0	-1	0	0	1	0	0
-0.0071	0	0	0	0	0	0	-1	0	0	1	0	-1	0	0	0	-1
0.0018	0	0	0	0	0	0	0	0	1	0	0	-1	0	1	0	1
0.0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table (5.7) The coefficients of 1-D filter (f1) of the first section of the 15^{th} order band pass FIR regular method

Design by Improved method

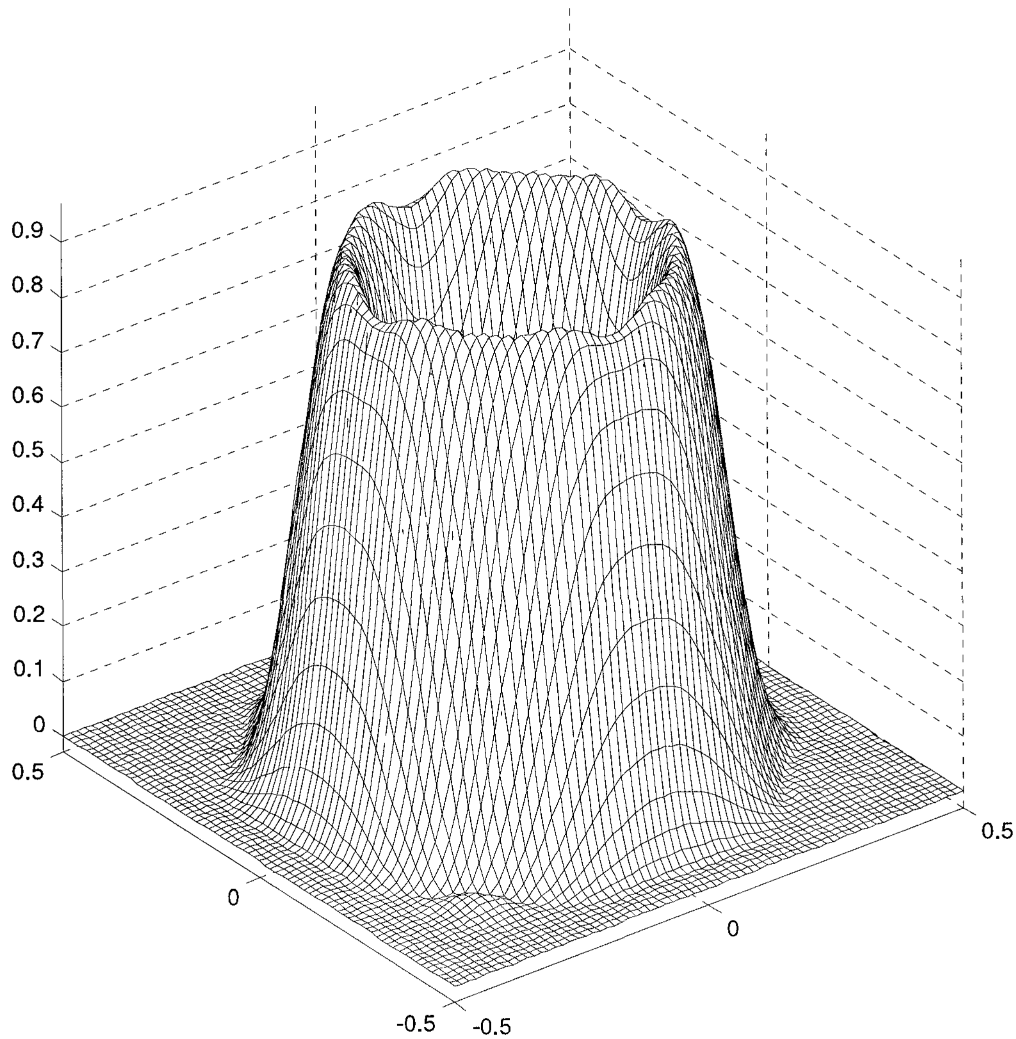


Figure (5.7) The 15^{th} order band pass FIR by improved method

The coefficients of 1-D filter (f1)of the first section in decimal and CSD																
0.00082	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**1**	**0**	**1**
0.0000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
0.00354	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**-1**	**0**	**1**	**0**	**0**	**-1**
0.0024	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**0**	**-1**	**0**	**1**
-0.0250	**0**	**0**	**0**	**0**	**-1**	**0**	**1**	**0**	**-1**	**0**	**1**	**0**	**0**	**0**	**0**	**0**
0.0128	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**1**	**0**	**0**	**0**	**1**	**0**	**0**
0.1273	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**-1**	**0**	**0**	**0**
-0.0145	**0**	**0**	**0**	**0**	**0**	**-1**	**0**	**0**	**0**	**1**	**0**	**0**	**1**	**0**	**1**	**0**
-0.3906	**0**	**0**	**1**	**0**	**0**	**-1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**-1**
-0.3906	**0**	**0**	**1**	**0**	**0**	**-1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**-1**
-0.0145	**0**	**0**	**0**	**0**	**0**	**-1**	**0**	**0**	**0**	**1**	**0**	**0**	**1**	**0**	**1**	**0**
0.1273	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**-1**	**0**	**0**	**0**
0.0128	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**1**	**0**	**0**	**0**	**1**	**0**	**0**
-0.0250	**0**	**0**	**0**	**0**	**-1**	**0**	**1**	**0**	**-1**	**0**	**1**	**0**	**0**	**0**	**0**	**0**
0.0024	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**0**	**-1**	**0**	**1**
0.00354	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**-1**	**0**	**1**	**0**	**0**	**-1**
0.0000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
0.00082	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**1**	**0**	**1**

Table (5.8) The coefficients of 1-D filter (f1) of the first section of the 15^{th} order band pass FIR improved method

MSE FIR	Regular method	Improved method
Order of subfilters	15-15-15-15-15-15-15	17-17-13-11-9-7-5
PASSBAND	1.0689e-003	1.69e-003
STOPBAND	1.65e-003	1.023e-003

Table (5.9) error comparison between improved and regular SVD of 15^{th} order band pass FIR

Table (4.3) and table (4.4) show that a reduction of 23.21% in coefficients achieved by improved method with a slightly higher error.

Example (4)

Design 31th order 2-D FIR High pass filter specified by:

$$\left|H(e^{j\omega_1 T_1}, e^{j\omega_2 T_2})\right| = \begin{cases} 0 & 0 \le \sqrt{\omega_1^2 + \omega_2^2} \le 0.4\pi \\ 1 & 0.6\pi \le \sqrt{\omega_1^2 + \omega_2^2} \le \pi \end{cases}$$

$$where \quad T_1 = T_2 = 1$$

M=L=21

Seven sections are used

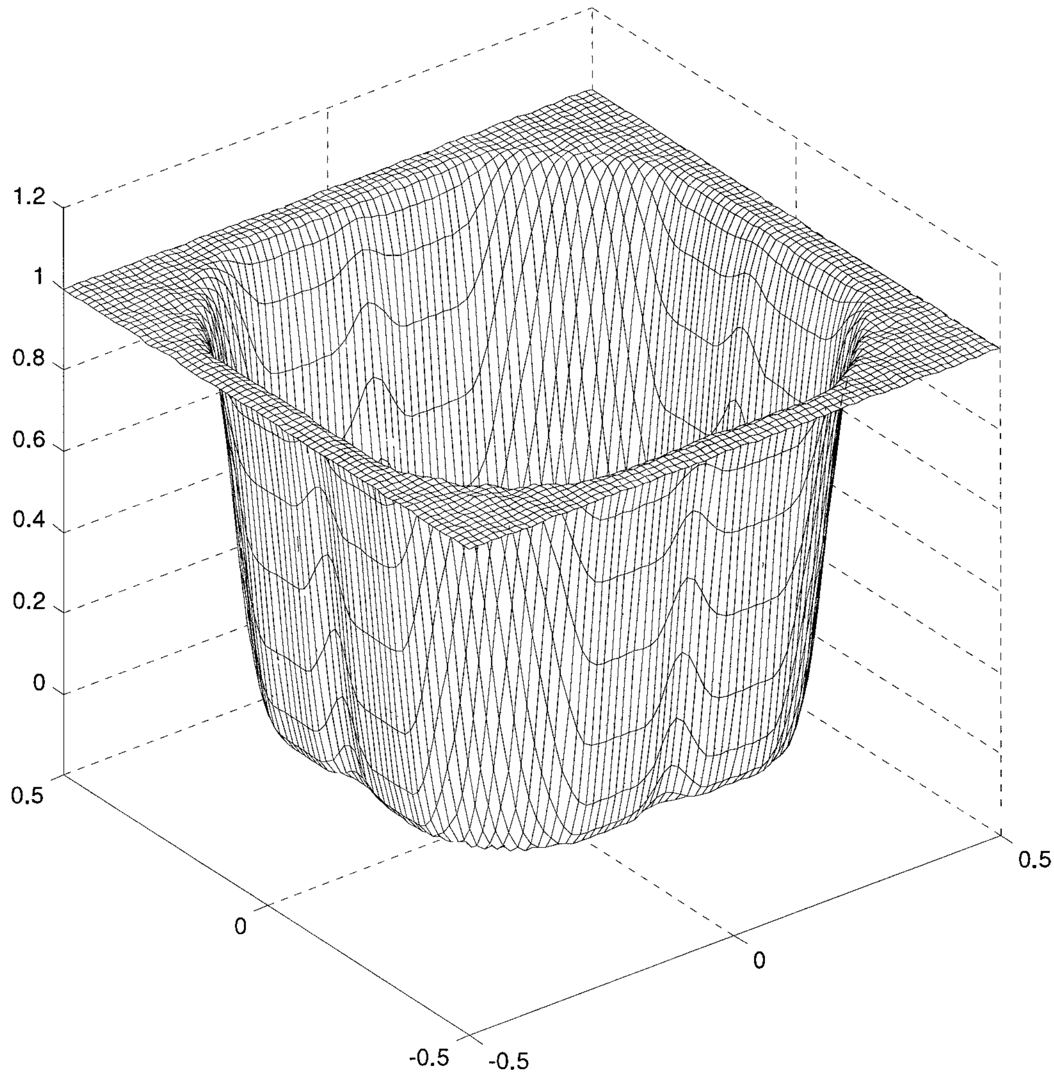


Figure (5.8) The 31st order high pass FIR by regular method

The coefficients of 1-D filter (f1)of the first section in decimal and CSD																
0.00067	0	0	0	0	0	0	0	0	0	1	0	-1	0	-1	0	-1
-0.00067	0	0	0	0	0	0	0	0	0	-1	0	1	0	1	0	1
-0.00081	0	0	0	0	0	0	0	0	0	-1	0	1	0	-1	0	-1
0.00082	0	0	0	0	0	0	0	0	0	1	0	-1	0	1	0	1
-0.0007	0	0	0	0	0	0	0	0	0	-1	0	1	0	1	0	-1
-0.0014	0	0	0	0	0	0	0	0	-1	0	1	0	0	1	0	1
0.0030	0	0	0	0	0	0	0	1	0	-1	0	0	1	0	0	-1
-0.005	0	0	0	0	0	0	0	-1	0	-1	0	0	-1	0	0	1
0.00067	0	0	0	0	0	0	0	0	0	1	0	-1	0	-1	0	-1
0.0075	0	0	0	0	0	0	1	0	0	0	0	-1	0	-1	0	-1
-0.0167	0	0	0	0	0	-1	0	0	0	-1	0	0	-1	0	0	-1
0.0165	0	0	0	0	0	1	0	0	0	1	0	0	-1	0	0	1
-0.00081	0	0	0	0	0	0	0	0	0	-1	0	1	0	-1	0	-1
-0.0703	0	0	0	-1	0	0	-1	0	0	0	0	0	0	1	0	-1
0.1724	0	1	0	-1	0	-1	0	0	0	0	1	0	0	0	0	0
-0.7481	-1.	1	0	0	0	0	0	0	1	0	0	0	0	-1	0	0
0.1724	0	1	0	-1	0	-1	0	0	0	0	1	0	0	0	0	0
-0.0703	0	0	0	-1	0	0	-1	0	0	0	0	0	0	1	0	-1
-0.00081	0	0	0	0	0	0	0	0	0	-1	0	1	0	-1	0	-1
0.0165	0	0	0	0	0	1	0	0	0	1	0	0	-1	0	0	1
-0.0167	0	0	0	0	0	-1	0	0	0	-1	0	0	-1	0	0	-1
0.0072	0	0	0	0	0	0	1	0	0	0	0	-1	0	-1	0	-1
0.00067	0	0	0	0	0	0	0	0	0	1	0	-1	0	-1	0	-1
-0.005	0	0	0	0	0	0	0	-1	0	-1	0	0	-1	0	0	1
0.0030	0	0	0	0	0	0	0	1	0	-1	0	0	1	0	0	-1
-0.0014	0	0	0	0	0	0	0	0	-1	0	1	0	0	1	0	1
-0.0007	0	0	0	0	0	0	0	0	0	-1	0	1	0	1	0	-1
0.00082	0	0	0	0	0	0	0	0	0	1	0	-1	0	1	0	1
-0.00081	0	0	0	0	0	0	0	0	0	-1	0	1	0	-1	0	-1
-0.00067	0	0	0	0	0	0	0	0	0	-1	0	1	0	1	0	1
0.00067	0	0	0	0	0	0	0	0	0	1	0	-1	0	-1	0	-1

Table (5.10) The coefficients of 1-D filter (f1) of the first section of the 31^{st} order high pass FIR regular method

Design by Improved method

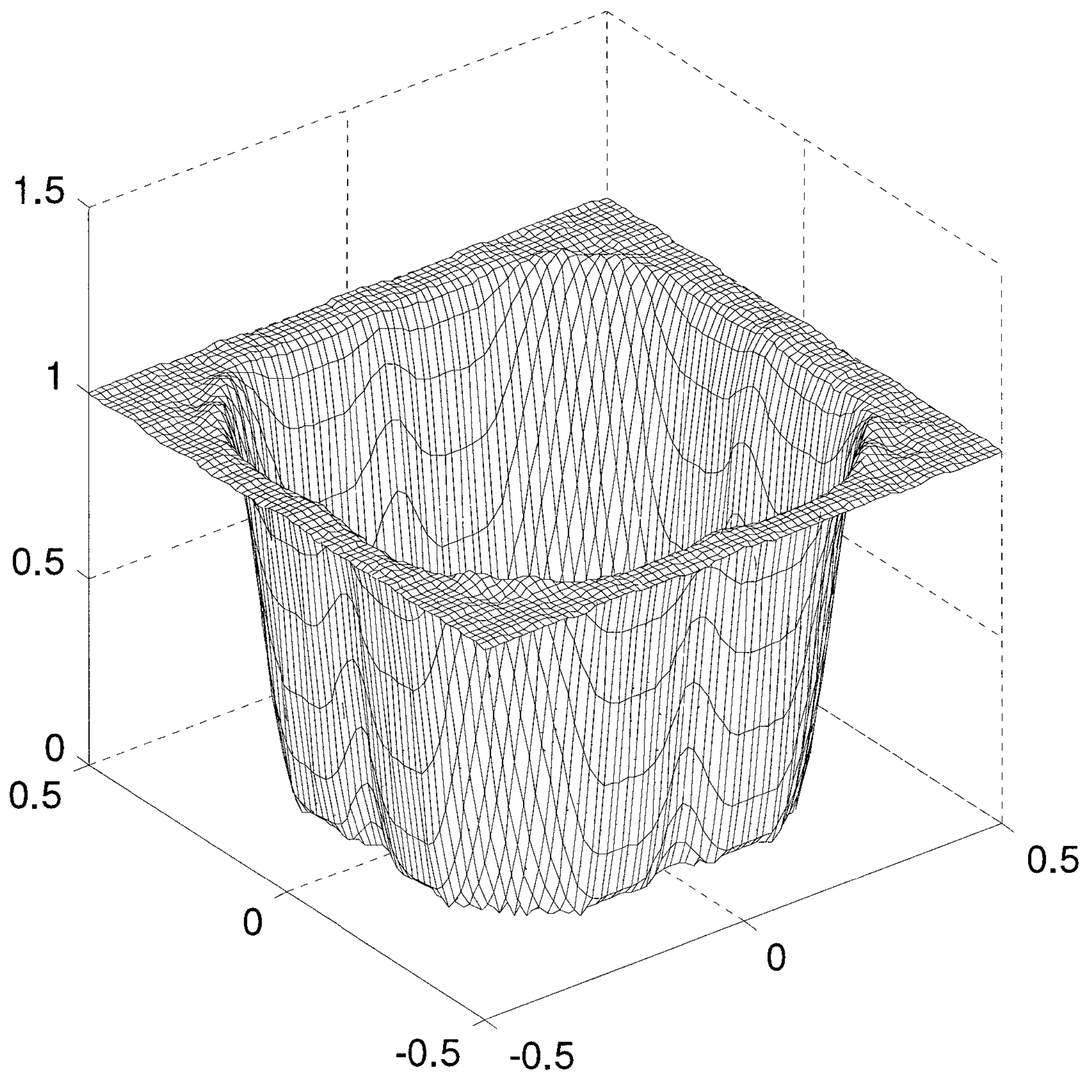


Figure (5.9) The 31st order hig pass FIR by improved method

The coefficients of 1-D filter (f1)of the first section in decimal and CSD																
-0.00067	0	0	0	0	0	0	0	0	0	-1	0	1	0	1	0	1
0.00067	0	0	0	0	0	0	0	0	0	1	0	-1	0	-1	0	-1
-0.00067	0	0	0	0	0	0	0	0	0	-1	0	1	0	1	0	1
-0.0013	0	0	0	0	0	0	0	0	0	-1	0	-1	0	-1	0	-1
0.0012	0	0	0	0	0	0	0	0	0	1	0	1	0	-1	0	1
-0.00081	0	0	0	0	0	0	0	0	0	-1	0	1	0	-1	0	-1
-0.0014	0	0	0	0	0	0	0	0	-1	0	1	0	0	1	0	1
0.0035	0	0	0	0	0	0	0	1	0	0	-1	0	0	1	0	1
-0.0065	0	0	0	0	0	0	-1	0	1	0	-1	0	-1	0	0	0
0.00067	0	0	0	0	0	0	0	0	0	1	0	-1	0	-1	0	-1
0.0082	0	0	0	0	0	0	0	0	0	1	0	-1	0	1	0	1
-0.0173	0	0	0	0	0	-1	0	0	-1	0	0	1	0	1	0	0
0.0158	0	0	0	0	0	1	0	0	0	0	0	1	0	-1	0	-1
-0.00067	0	0	0	0	0	0	0	0	0	-1	0	1	0	1	0	1
-0.0706	0	0	0	-1	0	0	-1	0	0	0	0	-1	0	-1	0	0
0.1726	0	1	0	-1	0	-1	0	0	0	1	0	0	0	0	0	0
-0.7481	-1.	1	0	0	0	0	0	0	1	0	0	0	0	-1	0	0
0.1726	0	1	0	-1	0	-1	0	0	0	1	0	0	0	0	0	0
-0.0706	0	0	0	-1	0	0	-1	0	0	0	0	-1	0	-1	0	0
-0.00067	0	0	0	0	0	0	0	0	0	-1	0	1	0	1	0	1
0.0158	0	0	0	0	0	1	0	0	0	0	0	1	0	-1	0	-1
-0.0173	0	0	0	0	0	-1	0	0	-1	0	0	1	0	1	0	0
0.0082	0	0	0	0	0	0	0	0	0	1	0	-1	0	1	0	1
0.00067	0	0	0	0	0	0	0	0	0	1	0	-1	0	-1	0	-1
-0.0065	0	0	0	0	0	0	-1	0	1	0	-1	0	-1	0	0	0
0.0034	0	0	0	0	0	0	0	1	0	0	-1	0	0	1	0	1
-0.0014	0	0	0	0	0	0	0	0	-1	0	1	0	0	1	0	1
-0.00081	0	0	0	0	0	0	0	0	0	-1	0	1	0	-1	0	-1
0.0012	0	0	0	0	0	0	0	0	0	1	0	1	0	-1	0	1
-0.0013	0	0	0	0	0	0	0	0	0	-1	0	-1	0	-1	0	-1
-0.00067	0	0	0	0	0	0	0	0	0	-1	0	1	0	1	0	1
0.00067	0	0	0	0	0	0	0	0	0	1	0	-1	0	-1	0	-1
-0.00067	0	0	0	0	0	0	0	0	0	-1	0	1	0	1	0	1

Table (5.11) The coefficients of 1-D filter (f1) of the first section of the 31^{st} order high pass FIR improved method

MSE FIR	Regular method	Improved method
Order of subfilters	31-31-31-31-31-31-31	33-33-29-27-25-23-21
PASSBAND	.0017	.003503
STOPBAND	.0044	.0091

Table (5.12) error comparison between improved and regular SVD of 31st order high pass FIR

5.4 Design with CSD=5

5.4.1 low pass FIR

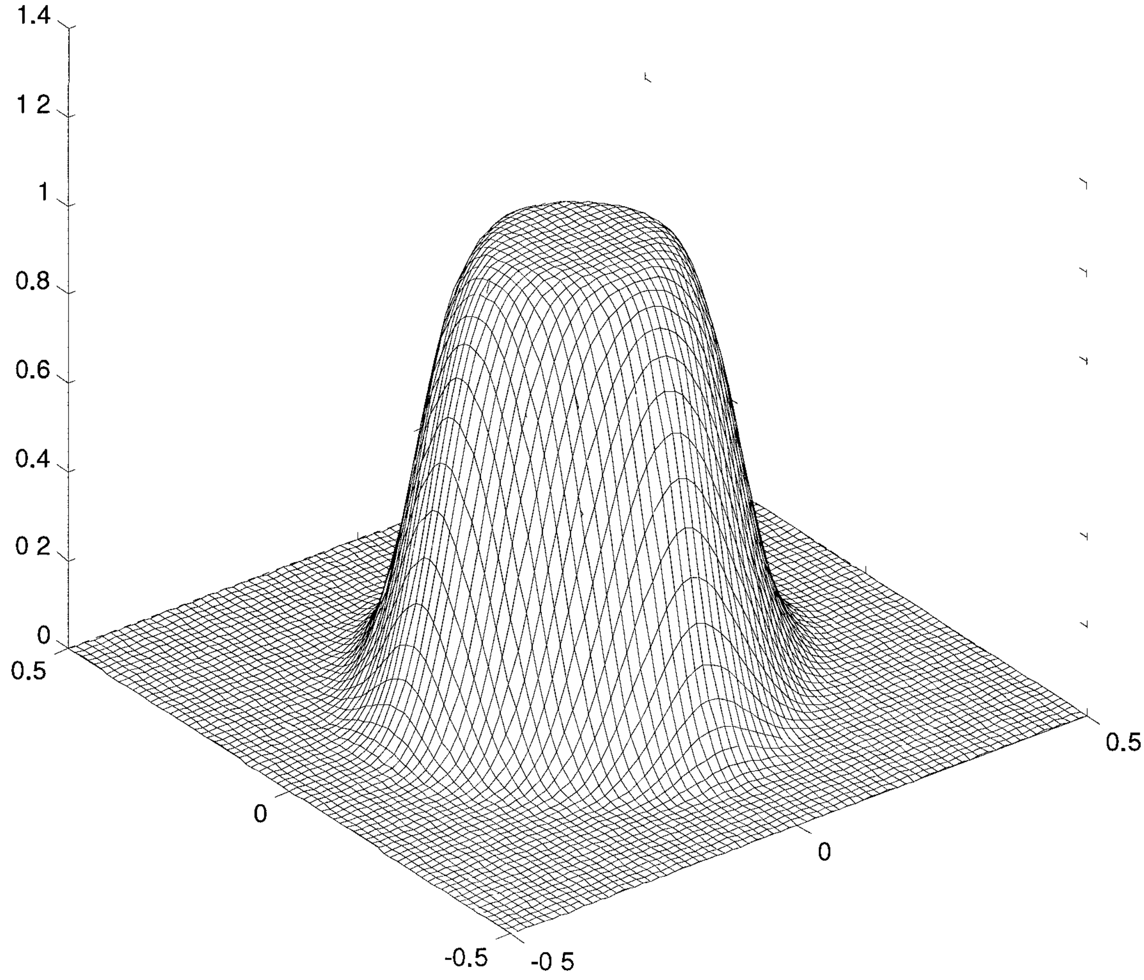


Figure (5 10) The 17th order low pass FIR by regular method

The coefficients of 1-D filter (f1)of the first section in decimal and CSD																
-0.009	**0**	**0**	**0**	**0**	**0**	**0**	**-1**	**0**	**0**	**-1**	**0**	**-1**	**0**	**1**	**0**	**-1**
0.009	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**1**	**0**	**1**	**0**	**-1**	**0**	**1**
-0.009	**0**	**0**	**0**	**0**	**0**	**0**	**-1**	**0**	**0**	**-1**	**0**	**-1**	**0**	**1**	**0**	**-1**
-0.00650	**0**	**0**	**0**	**0**	**0**	**0**	**-1**	**0**	**1**	**0**	**-1**	**0**	**-1**	**0**	**0**	**-1**
0.0065	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**1**	**0**	**1**	**0**	**0**	**1**
0.0287	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**-1**	**0**	**-1**	**0**	**0**	**-1**	**0**	**-1**
0.0065	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**1**	**0**	**1**	**0**	**0**	**1**
-0.1707	**0**	**-1**	**0**	**1**	**0**	**1**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**0**	**0**	**0**
-0.3880	**0**	**0**	**1**	**0**	**0**	**-1**	**0**	**1**	**0**	**-1**	**0**	**-1**	**0**	**0**	**0**	**0**
-0.3880	**0**	**0**	**1**	**0**	**0**	**-1**	**0**	**1**	**0**	**-1**	**0**	**-1**	**0**	**0**	**0**	**0**
-0.1707	**0**	**-1**	**0**	**1**	**0**	**1**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**0**	**0**	**0**
0.0065	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**1**	**0**	**1**	**0**	**0**	**1**
0.0287	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**-1**	**0**	**-1**	**0**	**0**	**-1**	**0**	**-1**
0.0065	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**1**	**0**	**1**	**0**	**0**	**1**
-0.0065	**0**	**0**	**0**	**0**	**0**	**0**	**-1**	**0**	**1**	**0**	**-1**	**0**	**-1**	**0**	**0**	**-1**
-0.009	**0**	**0**	**0**	**0**	**0**	**0**	**-1**	**0**	**0**	**-1**	**0**	**-1**	**0**	**1**	**0**	**-1**
0.009	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**1**	**0**	**1**	**0**	**-1**	**0**	**1**
-0.009	**0**	**0**	**0**	**0**	**0**	**0**	**-1**	**0**	**0**	**-1**	**0**	**-1**	**0**	**1**	**0**	**-1**

Table (5.13) The coefficients of 1-D filter (f1) of the first section of the 17^{th} order low pass FIR regular method

Design by Improved method

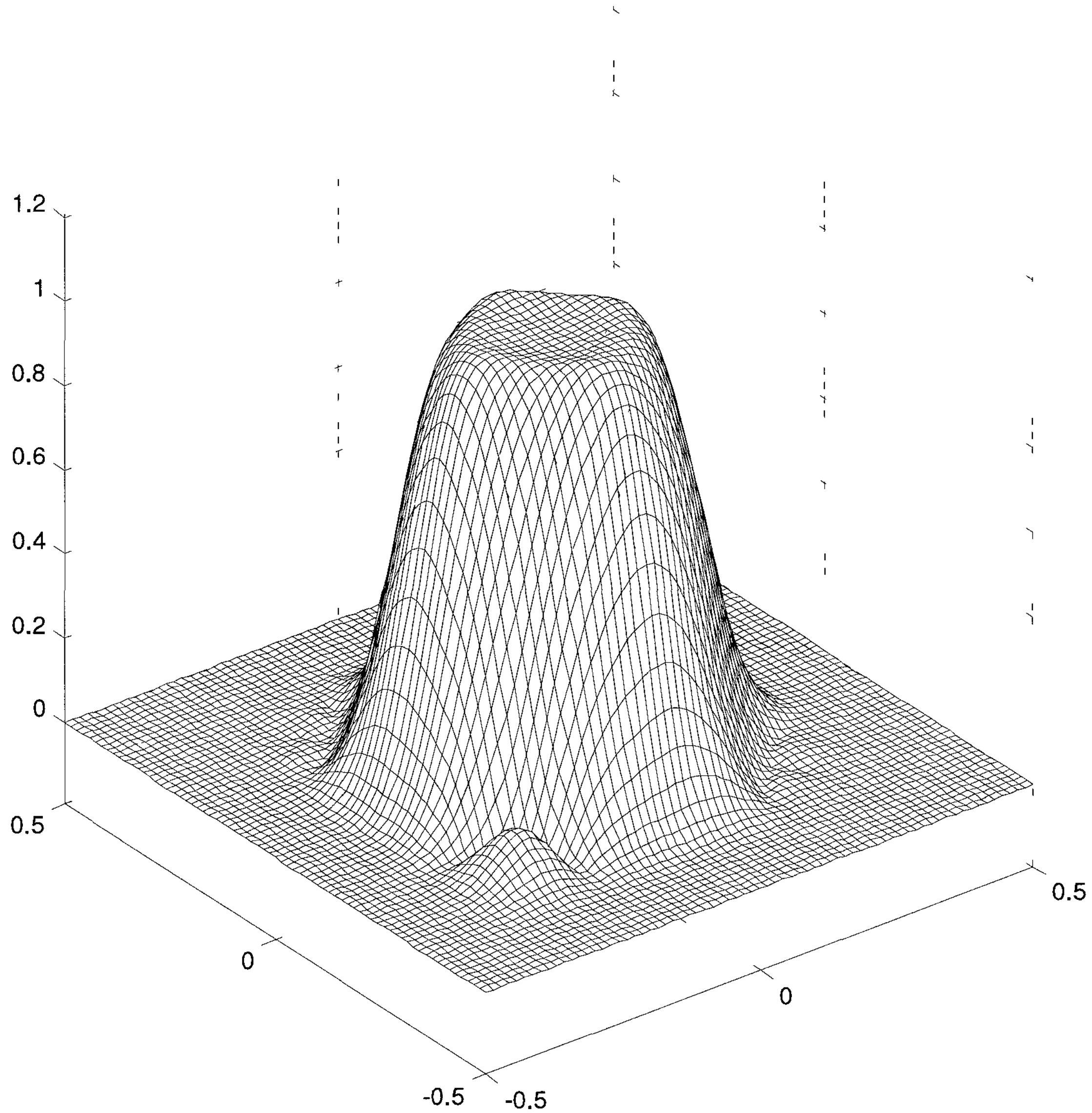


Figure (5.11) The 17^{th} order low pass FIR by improved method

The coefficients of 1-D filter (f1)of the first section in decimal and CSD																
- 0.009	0	0	0	0	0	0	-1	0	0	-1	0	-1	0	1	0	-1
-0.009	0	0	0	0	0	0	-1	0	0	-1	0	-1	0	1	0	-1
0.0057	0	0	0	0	0	0	1	0	-1	0	0	-1	0	1	0	1
-0.009	0	0	0	0	0	0	-1	0	0	-1	0	-1	0	1	0	-1
-0.0057	0	0	0	0	0	0	-1	0	1	0	0	1	0	-1	0	-1
0.0065	0	0	0	0	0	0	1	0	-1	0	1	0	1	0	0	1
0.0336	0	0	0	0	1	0	0	0	1	0	1	0	-1	0	1	0
0.0057	0	0	0	0	0	0	1	0	-1	0	0	-1	0	1	0	1
-0.1732	0	-1	0	1	0	1	0	0	0	-1	0	-1	0	0	0	0
-0.3886	0	0	1	0	0	-1	0	0	1	0	0	0	1	0	0	-1
-0.3886	0	0	1	0	0	-1	0	0	1	0	0	0	1	0	0	-1
-0.1732	0	-1	0	1	0	1	0	0	0	-1	0	-1	0	0	0	0
0.0057	0	0	0	0	0	0	1	0	-1	0	0	-1	0	1	0	1
0.0336	0	0	0	0	1	0	0	0	1	0	1	0	-1	0	1	0
0.0065	0	0	0	0	0	0	1	0	-1	0	1	0	1	0	0	1
-0.0057	0	0	0	0	0	0	-1	0	1	0	0	1	0	-1	0	-1
-0.009	0	0	0	0	0	0	-1	0	0	-1	0	-1	0	1	0	-1
0.0057	0	0	0	0	0	0	1	0	-1	0	0	-1	0	1	0	1
-0.009	0	0	0	0	0	0	-1	0	0	-1	0	-1	0	1	0	-1
-0.009	0	0	0	0	0	0	-1	0	0	-1	0	-1	0	1	0	-1

Table (5.14) The coefficients of 1-D filter (f1) of the first section of the 17^{th} order low pass FIR improved method

MSE FIR	Regular method	Improved method
Order of subfilters	17-17-17-17-17	19-15-13-11-9
PASSBAND	.0032	.00303
STOPBAND	.00054	.00049

Table (5.15) error comparison between improved and regular SVD of 17^{th} order FIR

5.4.2 Band pass FIR

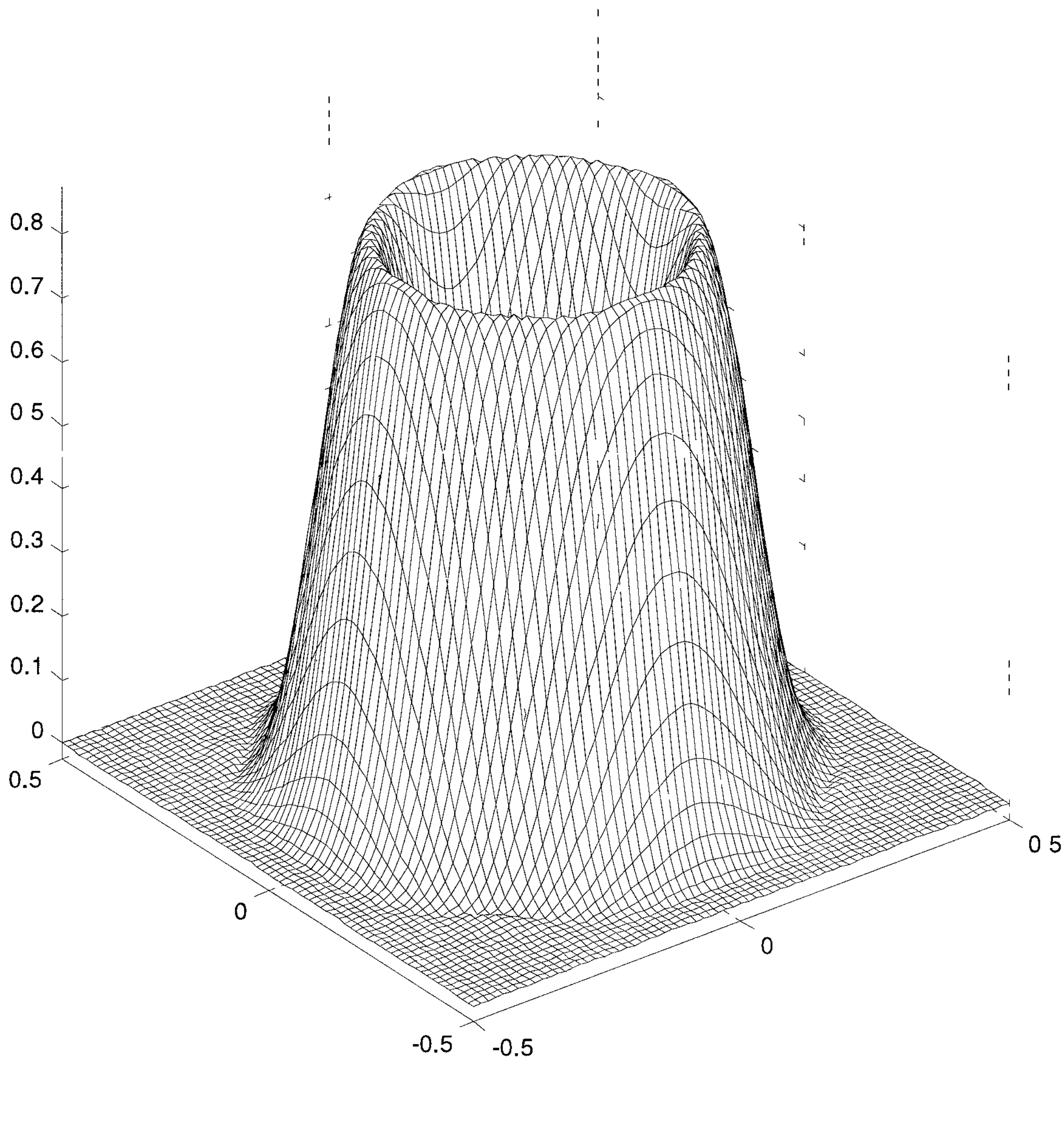


Figure (5.12) The 15th order band pass FIR by regular method

The coefficients of 1-D filter (f1)of the first section in decimal and CSD																
0.0000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
0.0057	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**0**	**-1**	**0**	**1**	**0**	**1**
-0.0065	**0**	**0**	**0**	**0**	**0**	**0**	**-1**	**0**	**1**	**0**	**-1**	**0**	**-1**	**0**	**0**	**-1**
-0.0200	**0**	**0**	**0**	**0**	**0**	**-1**	**0**	**-1**	**0**	**0**	**-1**	**0**	**0**	**1**	**0**	**1**
0.065	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**0**	**1**	**0**	**-1**
0.1021	**0**	**0**	**1**	**0**	**-1**	**0**	**1**	**0**	**0**	**0**	**1**	**0**	**0**	**1**	**0**	**0**
-0.2063	**0**	**-1**	**0**	**1**	**0**	**-1**	**0**	**-1**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**
-0.4681	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**-1**	**0**	**-1**
-0.2063	**0**	**-1**	**0**	**1**	**0**	**-1**	**0**	**-1**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**
0.1021	**0**	**0**	**1**	**0**	**-1**	**0**	**1**	**0**	**0**	**0**	**1**	**0**	**0**	**1**	**0**	**0**
0.065	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**0**	**1**	**0**	**-1**
-0.0200	**0**	**0**	**0**	**0**	**0**	**-1**	**0**	**-1**	**0**	**0**	**-1**	**0**	**0**	**1**	**0**	**1**
-0.0065	**0**	**0**	**0**	**0**	**0**	**0**	**-1**	**0**	**1**	**0**	**-1**	**0**	**-1**	**0**	**0**	**-1**
0.0057	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**0**	**-1**	**0**	**1**	**0**	**1**
0.0000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

Table (5.16) The coefficients of 1-D filter (f1) of the first section of the 15^{th} order band pass FIR regular method

Design by Improved method

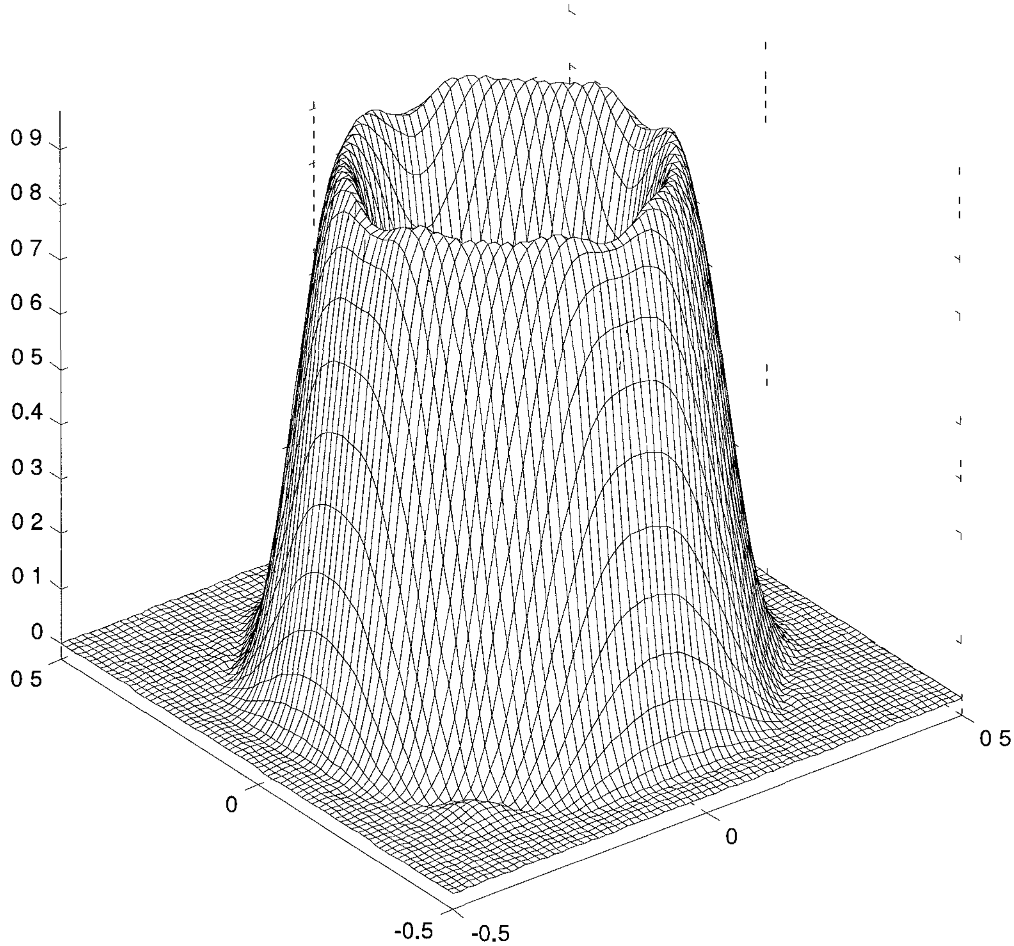


Figure (5.13) The 15^{th} order band pass FIR by improved method

The coefficients of 1-D filter (f1)of the first section in decimal and CSD																
0.0057	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**0**	**-1**	**0**	**1**	**0**	**1**
0.0000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
0.0057	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**0**	**-1**	**0**	**1**	**0**	**1**
0.0024	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**0**	**-1**	**0**	**0**	**1**	**0**	**-1**
-0.0250	**0**	**0**	**0**	**0**	**-1**	**0**	**1**	**0**	**-1**	**0**	**1**	**0**	**-1**	**0**	**0**	**0**
0.0128	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**1**	**0**	**0**	**0**	**1**	**0**	**-1**
0.1273	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**-1**	**0**	**-1**	**0**
-0.0195	**0**	**0**	**0**	**0**	**0**	**-1**	**0**	**-1**	**0**	**0**	**1**	**0**	**0**	**-1**	**0**	**1**
-0.3915	**0**	**0**	**1**	**0**	**0**	**-1**	**0**	**0**	**0**	**-1**	**0**	**0**	**1**	**0**	**0**	**-1**
-0.3915	**0**	**0**	**1**	**0**	**0**	**-1**	**0**	**0**	**0**	**-1**	**0**	**0**	**1**	**0**	**0**	**-1**
-0.0191	**0**	**0**	**0**	**0**	**0**	**-1**	**0**	**-1**	**0**	**0**	**1**	**0**	**0**	**-1**	**0**	**1**
0.1273	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**-1**	**0**	**-1**	**0**
0.0128	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**1**	**0**	**0**	**0**	**1**	**0**	**-1**
-0.0250	**0**	**0**	**0**	**0**	**-1**	**0**	**1**	**0**	**-1**	**0**	**1**	**0**	**-1**	**0**	**0**	**0**
0.0024	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**0**	**-1**	**0**	**0**	**1**	**0**	**-1**
0.0057	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**0**	**-1**	**0**	**1**	**0**	**1**
0.0000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
0.0057	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**0**	**-1**	**0**	**1**	**0**	**1**

Table (5.17) The coefficients of 1-D filter (f1) of the first section of the 15^{th} order band pass FIR improved regular method

MSE FIR	Regular method	Improved method
Order of subfilters	15-15-15-15-15-15-15	17-17-13-11-9-7-5
PASSBAND	1.0689e-003	1.69e-003
STOPBAND	1.65e-003	1.023e-003

Table (5.18) error comparison between improved and regular SVD of 15th order band pass FIR

5.4.3 High pass FIR

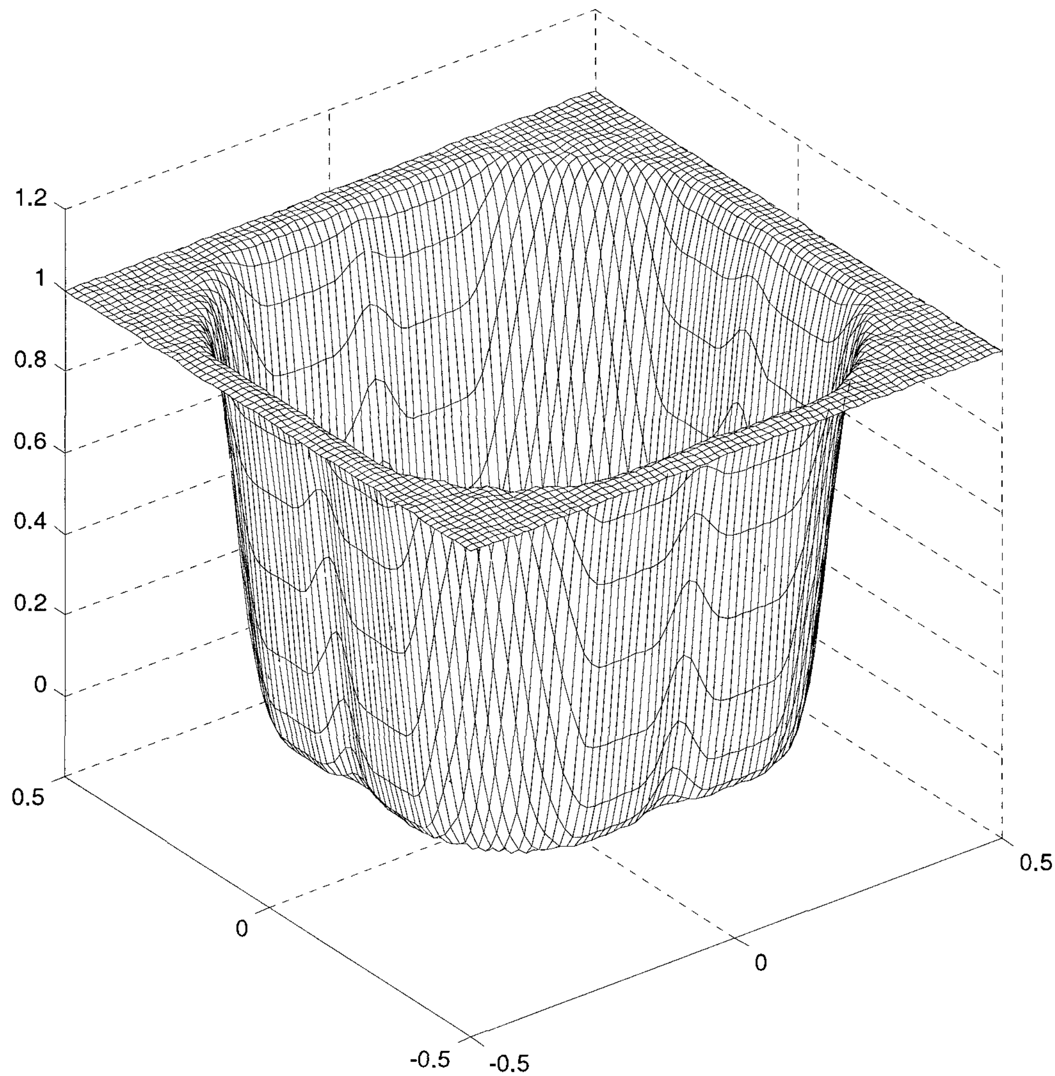


Figure (5.14) The 31st order high pass FIR by regular method

The coefficients of 1-D filter (f1)of the first section in decimal and CSD																
0.009	0	0	0	0	0	0	1	0	0	1	0	1	0	-1	0	1
-0.0057	0	0	0	0	0	0	-1	0	1	0	0	1	0	-1	0	-1
-0.009	0	0	0	0	0	0	-1	0	0	-1	0	-1	0	1	0	-1
0.009	0	0	0	0	0	0	1	0	0	1	0	1	0	-1	0	1
-0.0057	0	0	0	0	0	0	-1	0	1	0	0	1	0	-1	0	-1
-0.0065	0	0	0	0	0	0	-1	0	1	0	-1	0	-1	0	0	-1
0.0030	0	0	0	0	0	0	1	0	-1	0	0	-1	0	1	0	1
-0.0059	0	0	0	0	0	0	-1	0	1	0	0	1	0	-1	0	-1
0.009	0	0	0	0	0	0	1	0	0	1	0	1	0	-1	0	1
0.0065	0	0	0	0	0	0	1	0	-1	0	1	0	1	0	0	1
-0.0191	0	0	0	0	0	-1	0	-1	0	0	1	0	0	-1	0	1
0.0128	0	0	0	0	0	1	0	-1	0	1	0	0	0	1	0	-1
-0.009	0	0	0	0	0	0	-1	0	0	-1	0	-1	0	1	0	-1
-0.0191	0	0	0	0	0	-1	0	-1	0	0	1	0	0	-1	0	1
0.1724	0	1	0	-1	0	-1	0	0	0	0	1	0	0	0	0	-1
-0.7481	0	1	0	0	0	0	0	0	1	0	-1	0	1	0	0	1
0.1724	0	1	0	-1	0	-1	0	0	0	0	1	0	0	0	0	-1
-0.0191	0	0	0	0	0	-1	0	-1	0	0	1	0	0	-1	0	1
-0.009	0	0	0	0	0	0	-1	0	0	-1	0	-1	0	1	0	-1
0.0128	0	0	0	0	0	1	0	-1	0	1	0	0	0	1	0	-1
-0.0191	0	0	0	0	0	-1	0	-1	0	0	1	0	0	-1	0	1
0.0065	0	0	0	0	0	0	1	0	-1	0	1	0	1	0	0	1
0.009	0	0	0	0	0	0	1	0	0	1	0	1	0	-1	0	1
-0.0059	0	0	0	0	0	0	-1	0	1	0	0	1	0	-1	0	-1
0.0030	0	0	0	0	0	0	1	0	-1	0	0	-1	0	1	0	1
-0.0065	0	0	0	0	0	0	-1	0	1	0	-1	0	-1	0	0	-1
-0.0057	0	0	0	0	0	0	-1	0	1	0	0	1	0	-1	0	-1
0.009	0	0	0	0	0	0	1	0	0	1	0	1	0	-1	0	1
-0.009	0	0	0	0	0	0	-1	0	0	-1	0	-1	0	1	0	-1
-0.0057	0	0	0	0	0	0	-1	0	1	0	0	1	0	-1	0	-1
0.009	0	0	0	0	0	0	1	0	0	1	0	1	0	-1	0	1

Table (5.19) The coefficients of 1-D filter (f1) of the first section of the 31^{st} order high pass FIR regular method

Design by Improved method

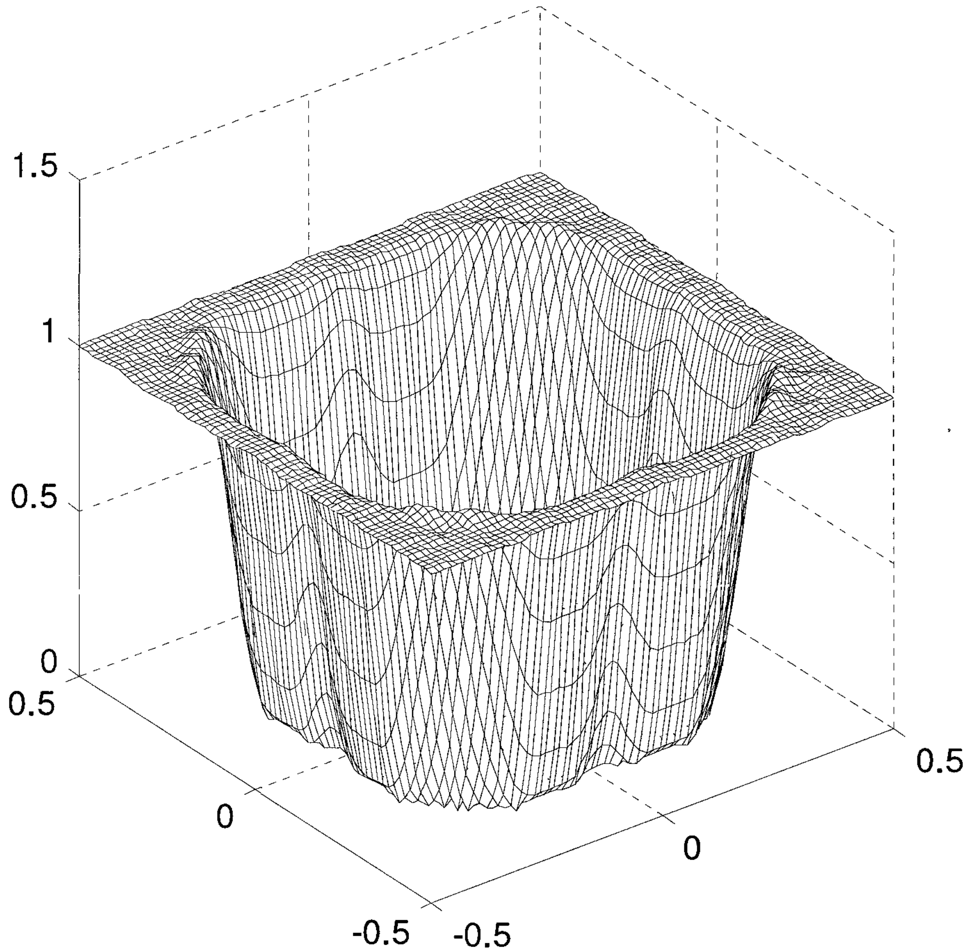


Figure (5.15) The 31^{st} order high pass FIR by improved method

The coefficients of 1-D filter (f1)of the first section in decimal and CSD																
-0.009	**0**	**0**	**0**	**0**	**0**	**0**	**-1**	**0**	**0**	**-1**	**0**	**-1**	**0**	**1**	**0**	**-1**
0.0057	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**0**	**-1**	**0**	**1**	**0**	**1**
-0.0057	**0**	**0**	**0**	**0**	**0**	**0**	**-1**	**0**	**1**	**0**	**0**	**1**	**0**	**-1**	**0**	**-1**
-0.0057	**0**	**0**	**0**	**0**	**0**	**0**	**-1**	**0**	**1**	**0**	**0**	**1**	**0**	**-1**	**0**	**-1**
0.0065	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**1**	**0**	**1**	**0**	**0**	**1**
-0.009	**0**	**0**	**0**	**0**	**0**	**0**	**-1**	**0**	**0**	**-1**	**0**	**-1**	**0**	**1**	**0**	**-1**
-0.0017	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**-1**	**0**	**0**	**1**	**0**	**0**	**0**	**-1**
0.0065	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**1**	**0**	**1**	**0**	**0**	**1**
-0.0065	**0**	**0**	**0**	**0**	**0**	**0**	**-1**	**0**	**1**	**0**	**-1**	**0**	**-1**	**0**	**0**	**1**
0.0057	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**0**	**-1**	**0**	**1**	**0**	**1**
0.0065	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**1**	**0**	**1**	**0**	**0**	**1**
-0.0173	**0**	**0**	**0**	**0**	**0**	**-1**	**0**	**0**	**-1**	**0**	**0**	**1**	**0**	**1**	**0**	**-1**
0.0173	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**1**	**0**	**0**	**-1**	**0**	**-1**	**0**	**1**
-0.009	**0**	**0**	**0**	**0**	**0**	**0**	**-1**	**0**	**0**	**-1**	**0**	**-1**	**0**	**1**	**0**	**-1**
-0.0707	**0**	**0**	**0**	**-1**	**0**	**0**	**-1**	**0**	**0**	**0**	**-1**	**0**	**1**	**0**	**0**	**-1**
0.1726	**0**	**1**	**0**	**-1**	**0**	**-1**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**0**	**0**	**0**
-0.7481	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**1**	**0**	**0**	**1**
0.1726	**0**	**1**	**0**	**-1**	**0**	**-1**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**0**	**0**	**0**
-0.0707	**0**	**0**	**0**	**-1**	**0**	**0**	**-1**	**0**	**0**	**0**	**-1**	**0**	**1**	**0**	**0**	**-1**
-0.009	**0**	**0**	**0**	**0**	**0**	**0**	**-1**	**0**	**0**	**-1**	**0**	**-1**	**0**	**1**	**0**	**-1**
0.0173	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**1**	**0**	**0**	**-1**	**0**	**-1**	**0**	**1**
-0.0173	**0**	**0**	**0**	**0**	**0**	**-1**	**0**	**0**	**-1**	**0**	**0**	**1**	**0**	**1**	**0**	**-1**
0.0065	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**1**	**0**	**1**	**0**	**0**	**1**
0.0057	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**0**	**-1**	**0**	**1**	**0**	**1**
-0.0065	**0**	**0**	**0**	**0**	**0**	**0**	**-1**	**0**	**1**	**0**	**-1**	**0**	**-1**	**0**	**0**	**1**
0.0065	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**1**	**0**	**1**	**0**	**0**	**1**
-0.0017	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**-1**	**0**	**0**	**1**	**0**	**0**	**0**	**-1**
-0.009	**0**	**0**	**0**	**0**	**0**	**0**	**-1**	**0**	**0**	**-1**	**0**	**-1**	**0**	**1**	**0**	**-1**
0.0065	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**1**	**0**	**1**	**0**	**0**	**1**
-0.0057	**0**	**0**	**0**	**0**	**0**	**0**	**-1**	**0**	**1**	**0**	**0**	**1**	**0**	**-1**	**0**	**-1**
-0.0057	**0**	**0**	**0**	**0**	**0**	**0**	**-1**	**0**	**1**	**0**	**0**	**1**	**0**	**-1**	**0**	**-1**
0.0057	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**-1**	**0**	**0**	**-1**	**0**	**1**	**0**	**1**
-0.009	**0**	**0**	**0**	**0**	**0**	**0**	**-1**	**0**	**0**	**-1**	**0**	**-1**	**0**	**1**	**0**	**-1**

Table (5.20) The coefficients of 1-D filter (f1) of the first section of the 31st order high pass FIR improved method

MSE FIR	Regular method	Improved method
Order of subfilters	31-31-31-31-31-31-31	33-33-29-27-25-23-21
PASSBAND	.0017	.0021
STOPBAND	.0044	.00505

Table (5.21) error comparison between improved and regular SVD of 31^{st} order high pass FIR

CHAPTER V

CONCLUSION

In this thesis, we design 2-D IIR and FIR digital filters by using SVD. In this approach, we have designed 2-D filters as a set of 1-D filters connected in cascade. Classical omptimization methods have been applied to design 1-D filters. Based on the singular values of 2-D filter, we choose the number of sections used for this design.

Genetic Algorithm combining with SVD is used to design 2-D filters. GA is used as an optimization technique to design 1-D filters. We encode filters' coefficients into CSD system. Each coefficient represnted with 16 bit wordlenght and different choice of non-zero digit.

An improvement to SVD was made by varying the order of 1-D filters in each section in according to their singular values. This improvement made the design more efficient by reducing the number of coefficients. We used this improved method with classical omtimization and GA to design 2-D filters.

We provide examples of designing 2-d filters by SVD, improved SVD, combination of SVD with GA and CSD, and combination of improved SVD with GA and CSD. The examples show that SVD provides efficient design with acceptable error in passband and stopband as well as the combination of SVD and GA and improved SVD and GA with less error in passband and stopband.

REFERENCES

1- S.Erfani,M.Ahamadi, and V.ramachandran" modified realization technique for digital filters derived from analog counterparts", J.Fraklin Inst.,Vol.322,No.4.,pp.221-228, Oct.1986

2- Mersereau, M.Russell, Mecklenbrauker,F.G. Wolfgang."McClellan Transformations for two dimensional digital filtering:I-design" IEEE Trans. On circuits and systems,Vol.CAS-23,No.7,pp.405-414,July 1976.

3- N.A. Pendergrass, Mitra, K. Sanjit, and E.I. Jury''Spectral transformations for two – dimensional digital filters'' IEEE Trans. On circuits and systems,Vol.CAS-23,No.1,pp.26-35,Jan 1976.

4- M. Ahmadi, V. Ramachandran,''New method for generating two-variable VSHPs and its application in the design of two-dimensiona recursive digital filters with prescribed magnitude and group delay responses,''IEEE proceedings, Vol. 131,pt. G, No. 4,pp. 151-155,Aug. 1984.

5- A. Lee, M.Ahmadi,''Digital filter design using genetic algorithm'', proc. Of 1988 IEEE Symp. On Advances in digital filtering and signal processing, pp.34-38, June 1998.

6- A.T.G. Fuller, B. Nowrouzian and F. Ashrafzadeh,"Optimization of FIR digital filters over the canonical signed-digit coefficient space using genetic algorithm", Proc. Of the 1998 Midwest Symposium on Circuits and Systems,pp.456-459, Aug. 1998.

7- T.Williams,M.Ahmadi," Design of high throughput 2-D FIR filters using singular value decomposition (SVD) and genetic algorithms", IEEE Pacific Rim Conference on Communications, Computers and signal processing, Vol. 2, 26-28, pp. 571-574, Aug. 2001

8- J.S. Lim, "Two-dimensional signal and image processing", Prentice Hall Inc 1990.

9- D.E Dudgeon and R, M, Mersereau, "Multidimensional digital signal processing", Englewood Cliffs, NJ: Prentice Hall Inc, 1984

10- T. W. Parks and C. S. Burrus, "Digital filter design", John Wiley & Sons Inc., 1987

11- A. Antoniou, "Design filters: Analysis, design and applications", McGraw-Hill Inc.,1993

12- Alan V. Oppenheim, Ronald W. Schafer and John R. Buck, "Discrete-time signal processing", Second Edition, Prentice-Hall Inc.,1998

13- Brian T. O'Connor and Thomas S. Huang, "Stability of general two-dimensional recursive digital filters." IEEE Transactions on acoustics, speech, and signal processing. ASSP-26, No. 6, pp-550-560, Dec. 1978.

14- John L. Shanks, Sven Treitel, and James H. Justice, "Stability and synthesis of two-dimensional recursive digital filters," IEEE Transactions on audio and electroacoustics, AU-20, No. 2, pp. 115-128, June. 1972.

15- Thomas S. Huang, "Stability of two-dimensional recursive filters," IEEE Transactions on audio and electroacoustics, AU-20, No. 2, pp. 284-286, June. 1972.

16- Dennis Goodman,"An Alternate proof of Huang's Theorem on stability,"IEEE Transactions on acoustics, speech, and signal processing. ASSP-24, No. 5, pp-426-427, Oct. 1976.

17- J. Leng, J. Brooke, T. Hewitt, and H. Davies, "Visualization of seismic data in geophysics and astrophysics," SGI Useres' Conference, Krakow, Poland, 11-14th Oct, 2000.

18- W. Sandham and M. Leggett, Geophysical Applications of artificial Neural networks and fuzzy Logic. Texas, USA: Springer, 2004.

19- D. E. Clark, I. Tena-Ruiz, Y. Petillot, and J. Bell, "Multiple-target tracking and data association in sonar image," Target Tracking: Algorithm and Applications, 20006. The IEE Seminar on (Ref. No. 2006/11359), pp. 147-154, 2006.

20- B. F. Burke and F. Graham-Smith, An introduction to radio astronomy. UK: Cambridge University Press; second edition, April 2002

21- T. S. Huang, Two- Dimensional digital signal processing, Vol.2. Applied Optics IP, vol. 20, Issue 16, p.2867, 1981

22- S. Chakrabarti, N.K. Bose and S.K. mitra, " Sum and Product separabilities of multivariable function and applications, " J. Franklin Inst., 299, pp.53-66, Jan. 1975.

23- A. Antoniou, M. Ahmadi and C. Charalambous, "Design of factorable low-pass 2-dimensional digital filters satisfying prescribed specifications", IEE proc., Vol. 128, Pt. G, No. 2, April 1981.

24- S. Treitel and J. L. Shanks, "The Design of multi-stage separable planar filters," IEEE Trans. Geoscience Electronics, pp. 10-27, Jan. 1971.

25- J. V. Hu and L. R. Rabiner, "Design Techniques for two-dimensional digital filters," IEEE Trans. Audio Electroacoustics, 20, pp. 249-257, Oct.1972.

26- P. K. Rajan, M. N. S. Swamy, " Design of separable denominator 2-dimensional digital filter possessing real circular symmetric frequency responses." IEE proc. Pt. G., Vol. pp. 235-240, 1982.

27- C. Charalambous, " Design of 2-dimensional circularly symmetric digital filters."IEE proc. Pt. G., Vol. 129, pp. 47,54,1982.

28- P. Karivaratharajan and M. N. S. Swamy, "Some results on the nature of a 2- dimensional filter function possessing certain symmetry in its magnitude response." Electronic circuits and systems. Vol. 2, No. 5, pp. 147-153, Sept. 1978.

29- P. Karivaratharajan and M. N. S. Swamy, "Maximally flat magnitude circularly symmetric two-dimensional low-pass IIR digital filters." IEEE Trans. on circuits and systems. Vol. CAS-27, No. 3, March, 1980.

30- P. Karivaratharajan and M. N. S. Swamy, " Quadrantal symmetry associated with two-dimensional digital transfer functions." Vol. CAS-25, pp. 340-343, 1978.

31- D.E. Dudgeon and R.M. Mersereau, Multidimensional digital signal processing, Englewood Cliffs, NJ: prentice Hall, 1984.

32- Chakrabarti, Satyabrata, and Mitra, K. Sanjit, " Design of two-dimensional digital filters via spectral transformations." proceedings of the IEEE, Vol. No. 6, pp. 905-914, June 1977.

33- A.N. Venetsanopoulos," Digital image processing and Analysis." (Les Houches, Session XLV, 12 Aout-6 Sep., 1985) Course 10 in signal processing. Vol. II, J.L. Lacoume, T.S. Durrani, and R. Stora, Eds., New York: North Holland, pp.575-668,1987.

34- D.E.Goldberg, Genetic algorithm in search, optimization, and machine learning, Addison; Wesley,1989.

35- K.S. Tang, K.F. Man, S. Kwong, Q.He "Genetic Algorithms and their applications."IEEE signal processing magazine, pp.22-37, Nov. 1996.

36- A. Antoniou and W. –S. Lu, "Application of the Singular Value Decomposition for the Design 2-DDigital Filters " communications, computers and signal processing, 1991, IEEE pacificRim conference on.

37- -A. Antoniou and W. –S. Lu, "Design of Two – Dimensional Digital filters with Arbitrary Amplitude and Phase Responses by using the Singular Value Decomposition " Circuits and Systems, 1991, IEEE International symposium on.

38- H.Safiri and M. Ahmadi, "Efficient Design of 2-D IIR digital filters using the Singular Value Decomposition" Circuits and Systems, 1991, IEEE International symposium on.

39- Rabiner, R. Lawrence, and Gold, Bernard,''Theory and application of digital signal processing, Prentice-Hall, New Jersey, 1975.''

40- F. Ashrafzadeh and ,B. Nowrouzian, Crossover and Mutotion In Genetic Algorithms Employing Canonical Signed-Digit Number System. Proceedings of the 1997 Midwest Symposium on Circuits and Systems, pp. 702-705, Aug.1997.

41- A. Lee, M. Ahmadi, G.A Julien, W.C. Miller: R.S. Lashkari. Digital Filter design Using Generic algorithm. Roceedings of IEEE DFSP'98, pp. 34-38, Victoria, B.C.: June 1998.

42- T. Williams, M. Ahmadi, W.C. Miller" Genetic Algorrithms for the design of digital filters using Canonical Signed Digit Coefficients". Communications, computers and signal processing, 2001, IEEE.PACRIM.2001 pacificRim conference on. pp.571-574 vol.2

43- M. Ahmadi and V. Ramachandran, "A method for the design of stable (N - D) analog and digital filters," in Proc. I981 IEEE Int. Conf. Acoustics, Speech, Signal Process., pp. 704-707.

44- R. E. Twogood and S. K. Mitra, "Computer-aided design of separable two-dimensional digital filters," IEEE Trans. Acoustics, Speech, Signal Process., vol. ASSP-25, no. 2, pp. 165-169, 1977.

45- Karaboga, N, and Cetinkaya, B, "A Study on the Effect of Error Functions on the Design of Optimum Digital IIR Filter Design Using Genetic Algorithm," 11. IEEE Sinyal Iseleme ve Uygulamalari Kurultayi (SIU'2003), 2003, pp. 200-203.

46- Karaboga, N, and Cetinkaya, B, "Optimal Design of Minimum Phase Digital FIR Filters by Using Genetic Algorithm," Third International Con. on Electrical and Electronics Engineering, December 2003, pp. 176-180.

47- N. Yazdi, M. Ahmadi and J.J. Soltis, "Efficient Design of 2-D Linear Phase FIR Filters using Singular Value Decomposition (SVD)", Electronics Letters. 19 Nov, 1992, Vol. 28, NO. 24, pp. 2256-2258.

48- A. Antoniou, W.S. Lu, "Design of Two-Dimensional Digital Filters by Using the Singular Value Decomposition", IEEE Trans. CAS-34, pp. 1 191- 1 198, 1987.

49- W.S. Lu, H.P. Wang, A. Antoniou, "Design of Two-Dimensional Digital Filters Using Singular-Value Decomposition and Balanced Approximation Method", IEEE Trans. CAS-39, pp.2253-2262, 1991.

50- G. A. Maria and M. M. Fahmy, "An lp design technique for two-dimensional digital recursive filters," IEEE Trans. Acoust., Speech, Signal Processing, vol. 22, pp. 15–21, Feb. 1974.

51- Li Liang, Majid Ahmadi, Maher A. Sid-Ahmed "Design of complementary filter pairs with Canonical Signed –Digit Coefficients using Genetic Algorithm". Electronics, Circuits and Systems, 2004. ICECS 2004. Proceedings of the 2004 11th IEEE International conference on. pp. 611-614.

52- M. Xiaomin, and Y. Yixian, "Optimal Design of FIR Digital Filter Using Genetic Algorithm", The Journal of China Universities of Posts and Telecommunications, vol. 5, No. 1, pp. 12-16, 1998.

53- Cetin Kaya Koc and Scott Johnson "Multiplication of Signed-Digit Numbers". Electronics Letters. Pp.840-841.vol.30. Issue:11.

54- S. Arno and F. S. Wheeler. Signed digit representations of minimal Hamming weight. IEEE Transactions on Computers, 42(8):1007{1010, August 1993.

55- I. Koren. Computer Arithmetic Algorithms. Englewood Cliffs, NJ: Prentice-Hall, 1993.

56- S. T. Tzeng, "Genetic algorithm approach for designing 2-D FIR digital filters with 2-D symmetric properties," Signal Processing, vol. 84, no. 10, pp. 1883-1893, October 2004.

57- K. S. Tang, K. F. Man, S. Kwong, and Q. He, "Genetic Algorithms and their applications," IEEE Signal Processing Magazine, pp.22-37, 1996.

58- Y.-H. Lee, M. Kawamata, and T. Higuchi, "GA-based design of multiplierless 2-D state-space digital filters with low roundoff noise," Proc. IEEE ., vol. 145, pp. 118–124, Apr. 1998

59- A. Antoniou, W.S. Lu, 'Practical Optimization Algorithms and Engineering Applications' 2007 Springer Science+Business Media, LLC

60- A. Antoniou, W.S. Lu, 'Two-Dimensional Digital Filters' Marcel Dekker Inc. 1992

61- R. King, M. Ahmadi,R.Naguib,A.Kwabwe, and M.Sadjadi "Digital filtering in one and two dimensions" 1989 Plenum press, New York

62- A. Antoniou, W.S. Lu, "Design of Two-Dimensional FIR Digital Filters Using Singular-Value Decomposition " IEEE Trans. Circuits Syst., vol. CAS-37, pp.35-46, 1990.

VITA AUCTORIS

Bashier Elkarami was born on June 7^{th} 1977. He received his Bachelor degree in Electrical Engineering in 2000 from Altahadi University. He is currentlly a master's candidate in Electrical and ComputerEngineering Department of the University of Windsor.