Design of quadrature mirror filter banks with canonical signed digit coefficients using genetic algorithms.

Haritha Uppalapati
University of Windsor

DESIGN OF QUADRATURE MIRROR FILTER BANKS WITH CANONICAL SIGNED DIGIT COEFFICIENTS USING GENETIC ALGORITHMS

BY

Haritha Uppalapati

A Thesis

Submitted to the Faculty of Graduate Studies and Research

Through Electrical Engineering

in Partial Fulfillment of the Requirements for the Degree of

Master of Applied Science at the

University of Windsor

Windsor, Ontario, Canada

Your file Votre référence
ISBN: 0-494-00135-6
Our file Notre référence
ISBN: 0-494-00135-6

1020584

Design of QMF Bank with CSD Coefficients using Genetic Algorithms

by

Haritha Uppalapati

APPROVED BY:

[illegible]

Dr. D. Ting
Mechanical, Automotive and Materials Engineering

Dr. B. Shahrrava
Electrical and Computer Engineering

[illegible]

Dr. M. A. Sid-Ahmed, Co-Advisor
Electrical and Computer Engineering

Dr. M. Ahmadi, Co-Advisor
Electrical and Computer Engineering

[illegible]

Dr. C. Chen, Chair of Defense
Electrical and Computer Engineering

October 1, 2004

ABSTRACT

This thesis is about the use of a genetic algorithm to design QMF bank with canonical signed digit coefficients. A filter bank has applications in areas like video and audio coding, data communication, etc.

Filter bank design is a multiobjective optimization problem. The performance depends on the reconstruction error of the overall filter bank and the individual performance of the composing lowpass filter. In this thesis we have used reconstruction error of the overall filter bank as our main objective and passband error, stopband error, stopband and passband ripples and transition width of the individual lowpass filter as constraints. Therefore filter bank design can be formulated as single objective multiple constraint optimization problem.

A unique genetic algorithm is developed to optimize filer bank coefficients such that the corresponding system's response matches that of an ideal system with an additional constraint that all coefficients are in canonical signed digit (CSD) format. A special restoration technique is used to restore the CSD format of the coefficients after crossover and mutation operators in Genetic algorithm. The proposed restoration technique maintains the specified word length and the maximum number of nonzero digits in filter banks coefficients.

Experimental results are presented at the end. It is demonstrated that the designed genetic algorithm is reliable, and efficient for designing QMF banks.

ACKNOWLEDGEMENTS

I would like to convey my sincere thanks to my Advisor Dr. M. Ahmadi, for his guidance through out my thesis. His advices helped me a lot to complete my research. I thank Dr. M. A. Sid Ahmed for his support.

I thank my husband, Jagadeesh, for his moral support and help. I convey love to my son, my parents and my brother.

TABLE OF CONTENTS

LIST OF TABLES AND FIGURES

FIGURES **PAGE**

CHAPTER 1
INTRODUCTION

Digital filters ,Multirate signal processing and Filter banks

DSP, or Digital Signal Processing, as the term suggests, is the processing of signals by digital means. Digital signal processing is one of the most powerful technologies. DSP technology, with its own algorithms, mathematics, and specialized techniques, revolutionized many areas in science and technology. Digital filters are a very important part of DSP. In fact, their extraordinary performance is one of the key reasons that DSP has become so popular.

DSP technology is nowadays commonplace in such devices as mobile phones, multimedia computers, video recorders, CD players, hard disc drive controllers and modems, and will soon replace analog circuitry in TV sets and telephones. An important application of DSP is in signal compression and decompression. In CD systems, for example, the music recorded on the CD is in a compressed form (to increase storage capacity) and must be decompressed for the recorded signal to be reproduced. Signal compression is used in digital cellular phones to allow a greater number of calls to be handled simultaneously within each local "cell". DSP signal compression technology allows people not only to talk to one another by telephone but also to see one another on the screens of their PCs, using small video cameras mounted on the computer monitors, with only a conventional telephone line linking them together.

1.1 DIGITAL FILTERS

Digital filters are used for two general purposes: Separation of signals that have been combined, and Restoration of signals that have been distorted in some way.Signal separation is needed when a signal is contaminated with interference, noise, or other signals. Signal restoration is needed when signal has been distorted in some way. For example, an audio recording made with poor equipment may be filtered to better represent the sound as it is actually occurred. Analog (electronic) filters can be used for these same tasks; however, digital filters can achieve far superior results.

1.1.1 FIR and IIR filters

The impulse response of a digital filter is the output sequence from the filter when a unit impulse is applied at its input. (A unit impulse is a very simple input sequence consisting of a single value of 1 at time n = 0, followed by zeros at all subsequent sampling instants). The impulse signal is defined as

$$\delta(n) \triangleq \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases} \tag{1.1}$$

Traditionally, digital filters have been classified into two large families: recursive filters and nonrecursive filters. A nonrecursive filter is one in which the current output ,y(n), is calculated solely from the current and previous input values (x(n), x(n-1), x(n-2),...). The output signal y(n) is therefore

$$y(n) = b_0 x(n) + b_1 x(n-1) + b_2 x(n-2) + \ldots + b_M x(n-M) \tag{1.2}$$

The transfer function is given by the z transform of the impulse response and it is a polynomial in the powers of z. For FIR filters, we have,

$$H(z) = \sum_{n=0}^{N-1} b_n z^{-n} \tag{1.3}$$

A recursive filter is one, which in addition to input values also uses previous output values. These, like the previous input values, are stored in the processor's memory. The word recursive literally means "running back", and refers to the fact that previously calculated output values go back into the calculation of the latest output. The expression for a recursive filter therefore contains not only terms involving the input values (x(n), x(n-1), x(n-2), ...) but also terms in y(n-1), y(n-2), ...The basic equation is shown below:

$$\begin{aligned} y(n) = {} & b_0 x(n) + b_1 x(n-1) + b_2 x(n-2) + \ldots\ldots.. + b_M x(n-M) \\ & - a_1 y(n-1) - \ldots\ldots.. - a_N y(n-N). \\ = {} & \sum_{m=0}^{M} b_m x(n-m) - \sum_{m=1}^{N} a_n y(n-m) \end{aligned} \tag{1.4}$$

Z transform is a mere substitution of each translation by m samples with a multiplication by z.

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b0 + b_1 z^{-1} + \ldots\ldots\ldots. + b_M z^{-M}}{1 + a_1 z^{-1} + \ldots\ldots\ldots.. + a_N z^{-N}} \triangleq \frac{B(z)}{A(z)} \tag{1.5}$$

An alternative terminology in which a non-recursive filter is known as an FIR (or Finite Impulse Response) filter, and a recursive filter as an IIR (or Infinite Impulse

Response) filter is often used. These terms refer to the differing "impulse responses" of the two types of filter.

An FIR filter is one whose impulse response is of finite duration. An IIR filter is one whose impulse response (theoretically) continues forever, because the recursive (previous output) terms feed back energy into the filter input and keep it going.

1.1.2 Frequency response:

Every linear filter has an impulse response, a step response and a frequency response. Each of these responses contains complete information about the filter, but in a different form. If one of the three is specified, the other two are fixed and can be directly calculated. The frequency response can be found by taking the DFT(discrete fourier transform) of the impuse response.

The Fourier transform of a continuous-time signal x(t) may be defined as

$$X(\omega) = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt, \qquad \omega \in (-\infty, \infty). \tag{1.6a}$$

The DFT, on the other hand, replaces the infinite integral with a finite sum:

$$X(\omega_k) \triangleq \sum_{n=0}^{N-1} x(t_n) e^{-j\omega_k t_n}, \qquad k = 0,1,2,......N-1 \tag{1.6b}$$

$x(t_n)$	$\triangleq$	input signal amplitude (real or complex)at time t_n (sec)
t_n	$\triangleq$	nT = sampling insatant (sec), n an integer ≥ 0
T	$\triangleq$	sampling interaval (sec)
$X(\omega_k)$	$\triangleq$	spectrum of x(complex valued), at frequency ω_k
ω_k	$\triangleq$	$k\Omega = kth$ frequency sample (radians per second)
Ω	$\triangleq$	$\frac{2\pi}{NT}$ = radian - frequency sampling interval($^{rad}/_{sec}$)
f_s	$\triangleq$	$1/T$ = sampling rate(samples per sec, or Hertz)
N	=	number of time samples = no. frequency samples

Figure 1.1 shows the four basic frequency responses. The purpose of these filters is to allow some frequencies to pass unaltered, while completely blocking other frequencies. The passband refers to those frequencies that are passed, while the stopband contains those frequencies that are blocked. The transition band is between. A fast roll-off means that the transition band is very narrow. The division between the passband and transition band is called the cutoff frequency.

1.2 MULTIRATE SIGNAL PROCESSING

All of the systems we have talked about so far are single-rate since the sampling rate does not change. Yet, there are many cases where multirate signal processing is either necessary or highly advantageous for example in audio, a CD is sampled at 44.1 kHz but DAT (Digital Audio Tape) is sampled at 48kHz; since studio recordings are mostly made on DAT, there needs to be a way to convert from one rate to the other. In video, PAL and NTSC run at different sampling rates, so to watch an American video in Europe, one needs a sample rate converter.

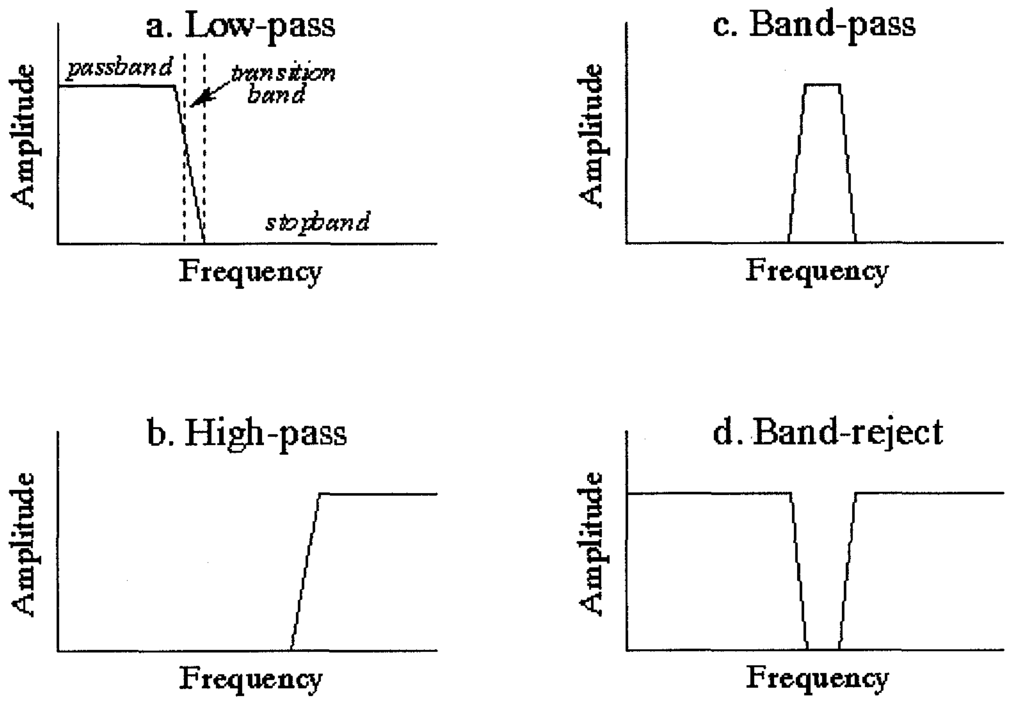


FIGURE 1.1: The four basic frequency responses

Multi-rate signal processing was designed to reduce the number of computations needed to convert a signal with an initial sampling rate to different sampling rates. Instead of going straight away from the initial sampling rate to the final one, which sometimes can be very computationally expensive, the rate conversion is accomplished by dividing the process into several steps (block sets), with each step producing a different sampling rate from the initial one. The process is resumed until the intended sampling rate is achieved.

Sampling rate conversion is often defined as the process of converting a signal from one rate to a different rate, either it is lower or higher than the original. The systems that use multiple sampling rates in processing digital signals are called multirate digital signal processing systems.

Changing a sampling rate of a signal has always been a part of DSP application. For example, in many telecommunication systems that transmit and receive various signals (facsimile, speech, video, etc), the sampling rate of the system has to be adjusted to suit the bandwidth of the signals in order to avoid signal degradation such as aliasing and imaging.

The most current example of multirate system is the “tri-band” cellular phone, which can be used over analog and two digital systems. When the phone “switches” its mode, the sampling rate is also changed in order to accommodate the different modulation scheme. Unfortunately, because of current DSP processing power (clock speed) limitation, most of this feature is probably done through hardware by employing different master clocks. Another example is the multimedia application, where data conversion is needed, e.g. from WAV (96.6 Kbps) to MP3 files (128 Kbps), and each format runs on different sampling rate.

1.3 FILTER BANKS

A digital filter bank is a set of band pass filters with either common input or summed output, each of which covers a band in the frequency spectrum. Other possible components of a filter bank include down samplers, up samplers and delay elements. In fig 1.2 maximally decimated filter bank is shown. At the analysis state, the input signal x[n] is passed through a bank of M analysis filters $H_i(z)$,each of which preserves a frequency band of uniform band width π/M. These M filtered signals are then decimated by M to preserve the system's overall sampling rate (thus this system is commonly labeled as maximally decimated or critically sampled filter bank). The resulting sub band signals can be encoded, processed, transmitted and decoded independently or jointly. All of these activities are grouped together in the processing block, which is typically not considered as a component of the filter bank. At the synthesis stage the sub bands are combined by a set of up samplers and M synthesis filters $G_i(z)$ to form the reconstructed signal x'(n).

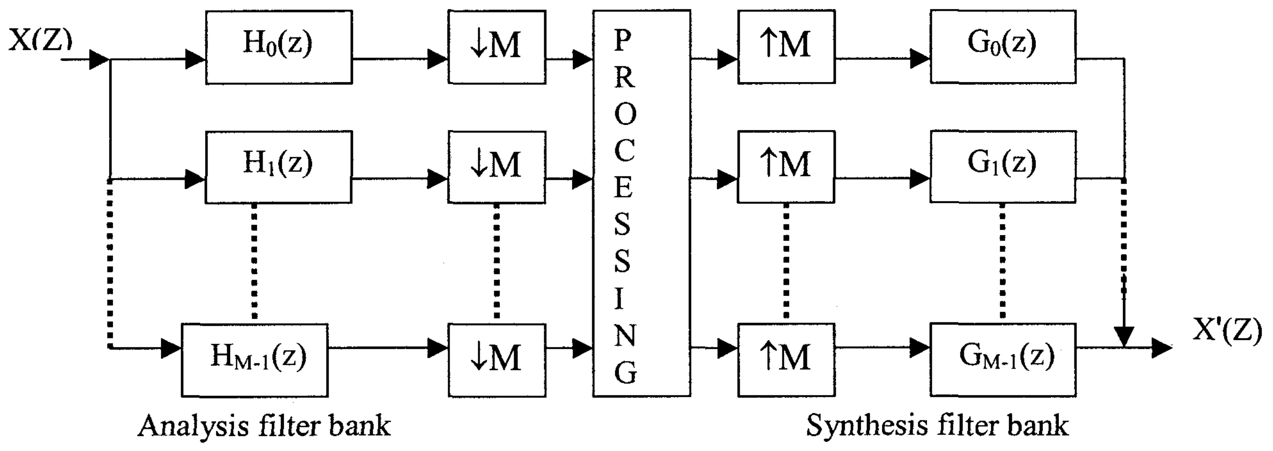


FIGURE 1.2: A typical M-Channel maximally decimated uniform filter bank

If the filters are ideal, no aliasing error occurs and perfect reconstruction is obtained. Perfect reconstruction is a very attractive property since it provides a loss-less signal representation.

1.3.1 Applications

There has been tremendous growth in the field of filter banks (FB) and multi rate systems in the last fifteen years. These systems provide new and effective tools to represent signals for processing, understanding, and compression purposes. It is quiet accurate to say that filter banks find applications virtually in every signal processing and closely related field: speech, audio and video compression, signal filtering, communication, time frequency representation and analysis; statistical signal processing; computer graphics, etc. Obviously, of extreme importance is the ability to design a filter bank (FB) that can fully exploit the properties and nature of a particular signal or application.

One particular application that filter banks have found tremendous success in is the compression of images. Image compression or image coding is the technology of image data reduction to save storage space and transmission bandwidth. With the recent explosion of internet. The search for better image compression techniques is becoming even more pressing. It is evidently to present images by the minimum number of binary digits given a fixed level of distortion, or for a given budget, to retain as much visual information as possible. Two dominant techniques in existing image compression standards and implementations are block transform coding [24] and sub band coding [34]. Both methods exhibit many similarities: operating in frequency domain, utilizing the same building blocks such as bit allocation, quantization, and entropy coding to achieve compression. In the coder both techniques rely heavily on filter banks to generate the frequency coefficients that can be quantized and the entropy coded. In the decoder, filter banks are again employed to combine and reconstruct signal [34]. Therefore, designing good filter banks plays an important role in the advancement of image coding technology.

One example is the quantization tables used in JPEG. JPEG is an image compression standard. Images encoded in JPEG format have much less storage than the original ones, so bandwidth, transmission time, and storage could be saved. In JPEG encoding, a still image is first partitioned into blocks, and in each block there are 8*8 Luminance components and 16*8 Chrominance components which are subsampled by two in horizontal frequency. Each block is then applied with an 8*8 DCT transform to remove the correlation among pixel values. After the DCT transform, the image is converted from spatial domain to frequency domain, and each DCT coefficient is quantized by the nearest quantization value and coded using Huffman coding or Arithmetic coding.

(a)Chrominance

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

(b)Luminance

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	74	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Table 1.1: JPEG Luminance and Chrominance quantization tables

Studies show that Human Visual System (HVS) response is highly dependent on spatial frequency. We know human eyes are more sensitive to low frequencies than to high frequencies. Better compression can be achieved by exploiting this feature. We need to decompose the image into different frequency space, and apply appropriate processing to the image structure that eyes can see and the structure that eyes are insensitive to. Hence, the quantization values used in JPEG, which are shown in Table 1.1, are visually weighted and frequency-dependant [30]. It is clear from Table 1.1 that quantization values of high frequencies are much larger than those of low frequencies, as human eyes are insensitive to high frequencies.

1.4QMF BANKS

In many applications a discrete time signal is first split into a number of subband signals by means of analysis filter bank,the subband signals are then processed and finally combined by a synthesis filter bank resulting in output signal. If the subband signals are band limited to frequency ranges much smaller than that of the origial input signal,they can be down sampled before processing.Beacause of lower sampling rate ,the processing of the downsampled signals can be carried out more efficiently.After processing these signals are upsampled before being combined by the synthesis bank into ahigher rate signal. The combined structure is called a QMF (quadrature mirror filter) bank [20].

A 2-channel filter bank is shown in Fig. 1.3. This filter bank consist of an analysis filter bank, formed by the analysis filters $H_0(z)$ and $H_1(z)$, and a synthesis bank formed by the synthesis filters $G_0(z)$, and $G_1(z)$, $H_0(z)$ and $G_0(z)$ in the upper channel are low-pass (LP) filters, and $H_1(z)$ and $G_1(z)$ in the lower channel are high-pass (HP) filters. Two decimators and interpolators are used for down sampling and up sampling.

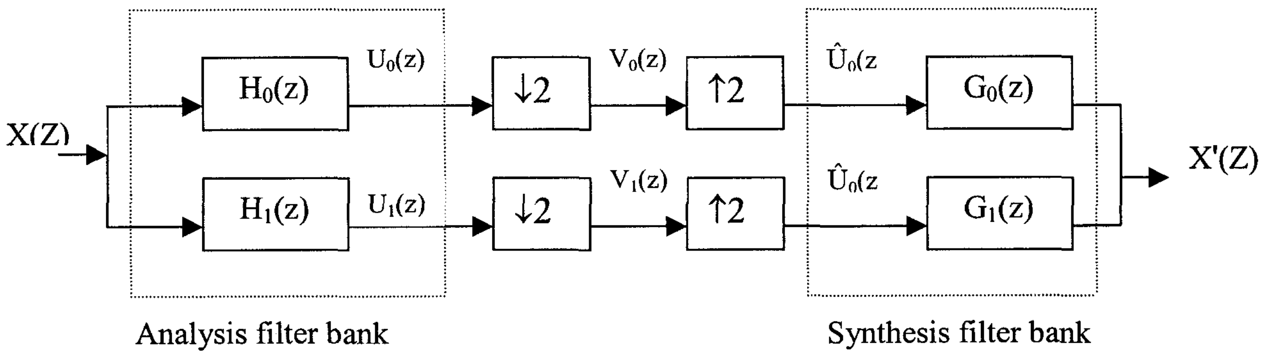


FIGURE 1.3: Two Channel QMF filter bank

When an incoming signal passes through the analysis filter bank, it is decomposed into low and high sub channels, and applied with the desired processing. The reconstructed signal x(z) is obtained after it passes through the synthesis filter bank.

1.4.1 Previous research:

QMF FIR filter banks By setting high-pass (HP) filter $h_l(n)$ in Figure 1.3 to be the quadrature mirror filter (QMF) of the low-pass (LP) filter $h_o(n)$, the filter bank design problem is simplified to be a single prototype ($h_o(n)$) filter design. Various researches were made for the design of QMF bank. Optimization based and non-optimization based algorithms can be used to design filter banks. Constraint based methods were applied for the design in both frequency domain [10,43,3,13,4, 7,46,36,24,28] and time domain [22,35,9]. Johnston [10] has designed a large class of low pass filters meeting a variety of specifications. This method of design uses a procedure based on Hooks and Jeevs algorithm, which requires a very high computational complexity. Nayebi[22] gave a time domain formulation with constraints in the frequency domain. Lagrange multiplier methods [46,8] were also used to design QMF banks. Novel [21,47,49] uses a continuous trace function to bring a search out of local minima rather than starting the search from a new starting point when the search finds a feasible design. While these methods result in good performance they use continuous coefficients in their design and becomes costly when realized in hardware. An elegant way of reducing the complexity of processing is to use powers of two coefficients instead of continuous coefficients. DLM-98[46]gave a new discrete Lagrange method for designing a multiplier less QMF bank .

Non-QMF filter banks Methods for designing non-QMF FIR filter banks include unconstrained optimization methods, similar to those for designing QMF FIR filter banks except that the former has more dimensions [28], and constrained optimization methods [1], including Lagrange multiplier [1, 46] and time domain methods [22].

IIR filter banks A very efficient way of representing the analysis filter bank canbe obtained by using polyphase components. In polyphase form, filter H(z) is expressed as $A_o(z^2) + z^{-1} A_1(z^2)$, where $A_o(z)$ and $A_1(z)$ are polyphase filters. If $H_1(z)$ is the quadrature mirror image of low-pass filter $H_o(z)$, then

$$H_1(z) = A_0(z) - z^{-1} A_1(z^2). \tag{1.7}$$

If the polyphase components consists of all pass transfer functions aliasing can be cancelled.

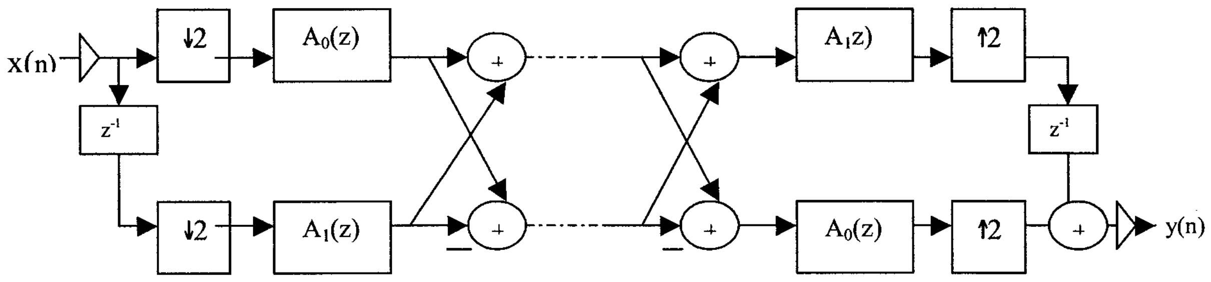


FIGURE 1.4: A magnitude preserving two channel QMF bank.

Usually IIR filter banks have shorter time delay and require fewer computations that FIR filter banks. However, IIR filters are generally more difficult to design than FIR filters because of their strong non-linearity and stability problem. There exit many methods for designing IIR filter banks [5, 33, 21, 42, 38, 27, 12, 50], including polyphase all-pass filter banks [33, 21, 42], unitary and non-unitary filter banks with all pass subfilters [26,27], cosine modulated filter banks [12,27], and direct optimization methods [32,41]. The main problem of using IIR filters in filter bank design in phase distortion.

Multiband and multirate filter banks Multiband and multirate filter banks use different sampling rates in different channels. They have been actively studied, due to their flexibility and tolerance to errors [40].

To restrict the search space, many properties, such as linear phase, paraunitary, and cosine modulation, are imposed. To further simplify the problem, most studies only consider aliasing errors in adjacent channels by assuming that stopbank attenuation in non-adjacent subbands is small. Existing design methods are based on either the frequency domain or the time domain. In the frequency domain, both unconstrained formulations [17, 12, 31, 29] and constrained formulations have been proposed. Other strategies include spectral factorization [45] and hybrid methods [25].

1.5 MOTIVATION AND THEISIS ORGANIZATION:

FIR QMF bank design can be specified as a problem that searches for prototype filter coefficients which improves

- The overall performance of the filter bank by minimizing the amplitude distortion.
- The individual performance of the composing filter by satisfying the magnitude response of $H_0(z)$ with $H_0(z) \cong 1$ in its pass band $H_0(z) \cong 0$ in its stop band.

The QMF bank design is an multiobjective optimization problem, with objectives of a design are to:

1. Minimize reconstruction error (E_r)
2. Minimize stopband energy (E_s),
3. Minimize passband energy (E_p),
4. Minimize stop band ripple (δ_s),
5. Minimize passband ripple (δ_p), and
6. Minimize transition bandwidth ($\Delta\omega$).

Where

$$Er = \int_{\omega=0}^{\pi/2} (|H_0(e^{j\omega})|^2 + |H_0(e^{j\omega-\pi})|^2 - 1)^2 d\omega \tag{1.8}$$

$$E_s = \int_{\omega=\omega_s}^{\pi} (|H_0(e^{j\omega})|^2) d\omega \tag{1.9}$$

$$E_p = \int_{\omega=0}^{\omega=\omega_s} (|H_0(e^{j(\omega-\pi)})|^2 - 1)^2 d\omega \tag{1.10}$$

This multiobjective optimization can be changed to single objective constrained optimization by keeping reconstruction error as our objective and converting other objectives into constraints with the help of best known solution as our reference. By doing so the QMF bank design is transformed as,

Minimize	E_r
Subject to	$E_s \leq c_{Es}$
	$E_p \leq c_{Ep}$
	$\delta_s \leq c_{\delta s}$
	$\delta_p \leq c_{\delta p}$

$$\Delta\omega \leq c_{\Delta\omega} \qquad (1.11)$$

This is single-objective, constrained nonlinear optimization problem. By using penalty method all constraints are absorbed into the objective function and weighed by penalty coefficients. This converts a constrained problem into an unconstrained problem.Minimize function 'f' which is formulated as

$$f = E_r + w_1(E_s - c_{Es}) + w_2(E_p - c_{Ep}) + w_3(\delta_s - c_{\delta s}) + w_4(\delta_p - c_{\delta p}) + w_5(\Delta\omega - c_{\Delta\omega}) \qquad (1.12)$$

Where w_1, w_2, w_3, w_4 and w_5 are penalty coefficients. Selected ahead of time these penalty coefficients are used to penalize a constraint when it is violated.

Algorithms for designing filter banks can be classified into two categories: optimization based and non-optimization based. Optimization based methods formulate the design problem as a multi-objective optimization problem [38], whose form may depend on the application and the composing filter type. The multi-objective optimization problem is then converted into single objective optimization using different methods [10,50,13,47,49]. The problem is solved by existing optimization methods, gradient descent, Lagrange multiplier [1,8,46], Novel [21,49]. Novel uses a continuous trace function to bring a search out of local minima rather than restarting the search from a new starting point when the search finds a feasible design. Filter bank design problems have also been designed by non-optimization algorithms [31]. These methods generally do not continue to find better designs once sub optimal design has been found [41]. Note that all these designs have continuous coefficients. Multiplication of such long floating-point numbers generally limits the speed of filtering. To overcome this limitation, filters with canonical signed digit coefficients have been proposed. A limited sequence of shifts and adds are usually much faster than full multiplication. The number of required arithmetic operations, ignoring shifts (which come free), can be reduced by expressing coefficients in canonical-signed-digit (CSD) form. In this thesis to reduce the computational complexity the filter bank is constrained to have canonical signed digit coefficients. A genetic algorithm is used as an optimization technique to minimize equation (1.12). Genetic algorithms are used to design FIR and IIR digital filters [11,15,16,18]. In general, genetic algorithms (GAs) rely upon the law of fittest member survival to optimize a set of possible outcomes or results. Loosely based on the laws of Darwinian genetics, GAs has many unique characteristics that make them ideal for many optimization problems. As Goldberg states (1989),

"They [GAs] combine survival of the fittest among string structures with a structured, yet randomized information exchange to form a search algorithm with some of the innovative flair of human search."

Our motivation is to design FIR Quadrature mirror filter bank with canonical signed digit coefficients using genetic algorithm as the optimization technique.

Design methods of analog filters, FIR and IIR digital filters and design of QMF Fir filter banks are explained in chapter 2.In chapter 3 representation of canonical signed digit numbers, the properties, advantages and application of CSD numbers [14] and multiplication arithmetic of QMF filter banks are discussed. Chapter 4 discusses the theory of genetic algorithms, the GA terminology, and the genetic operators. Chapter 5 discusses the proposed genetic algorithm for the design of QMF filter bank with CSD coefficients, its deviation from the traditional GA and its parameters. The new CSD restoration technique and its efficiency are presented. At the end the performance of the genetic algorithm with CSD coefficients is proved by the designing 12,24B, 24C and 32 order filter banks. The resulting CSD coefficients are presented.

CHAPTER 2
DESIGN METHODS

FIR and IIR digital filters, FIR QMF banks

In signal processing, the function of a filter is to remove unwanted parts of the signal, such as random noise, or to extract useful parts of the signal, such as the components lying within a certain frequency range. FIR filters are particularly useful for applications where exact linear phase response is required. The FIR filter is generally implemented in a non-recursive way, which guarantees a stable filter. Where as IIR filters can achieve a given filtering characteristic using less memory and calculations than a similar FIR filter. The IIR filter design procedure is carried out as follows:

1. Convert the tolerance specs to the analog domain
2. Design the analog filter
3. Convert the analog filter transfer function $Hc(s)$ back to the discrete-time domain to obtain $H(z)$.

The second step, the design of the analog filter, is done using a filter prototype. The analog prototypes ,Chebyshev, Butterworth and elliptic,are the most popular. The FIR filter is generally implemented in a non-recursive way, which guarantees a stable filter. There are essentially three well-known methods for FIR filter design namely:

1. The window method
2. The frequency sampling technique

3. Optimal Equiripple Design

2.1 ANALOG FILTERS

Butterworth filter:A Butterworth filter is described by the magnitude squared of its frequency response:

$$|Hc(j\Omega)|^2 = \frac{1}{1+\left(\frac{j\Omega}{j\Omega c}\right)^{2N}} \tag{2.1}$$

The Butterworth filter provides the best Taylor Series approximation to the ideal lowpass filter response at analog frequencies $\Omega=0$ and $\Omega=\infty$; for any order N, the magnitude squared response has $2N$-1 zero derivatives at these locations (*maximally flat* at $\Omega=0$ and $\Omega=\infty$). The best property of the Butterworth filter is that its magnitude response is monotonic, *i.e.* it has no ripples. The penalty we pay for that is that the roll off from the passband to the stopband is very gentle, *i.e.* we need to use a very high filter order to achieve either large stopband attenuation or a narrow transition band.

Chebyshev filter: There are two types of Chebyshev filters: type I and type II. Chebyshev I filters have ripple in the passband, but no ripple in the stopband. Chebyshev II filters have ripple in the stopband, but no ripple in the passband. The magnitude squared of the frequency response of the Chebyshev I filter is

$$|H_c(j\Omega)|^2 = \frac{1}{1+\left[\frac{1}{(1-\delta_1)^2}-1\right]V_N{}^2\left(\Omega/\Omega_p\right)} \tag{2.2}$$

Where $\Omega\ p$ is the frequency of the passband edge, $\delta\ 1$ is the maximum passband ripple, and N is the filter order. $VN\ (x)$ is the N th order Chebyshev polynomial defined as,

$$V_N(x) = \begin{cases} \cos(N\cos^{-1}x), & |x| \le 1 \\ \cosh(N\cosh^{-1}x), & |x| > 1 \end{cases} \tag{2.3}$$

The magnitude squared of the frequency response of the Chebyshev II filter is the same as that for Chebyshev I, except that we replace some terms with their reciprocals:

$$|H_c(j\Omega)|^2 = \frac{1}{1+\left[\left(\frac{1}{(1-\delta_1)^2}-1\right)V_N{}^2\left({}^{\Omega_p}\!/_{\Omega}\right)\right]^{-1}} \tag{2.4}$$

Elliptic Filter: Elliptic filters are optimal in the sense that for a given filter order N, and for fixed values of Ωp, $\delta 1$ and δ_2 they achieve the smallest possible transition band. The penalty we pay for this is that we now have ripples both in the passband and in the stopband the magnitude squared of the frequency response is now

$$|H_c(j\Omega)|^2 = \frac{1}{1+\left[\left(\frac{1}{(1-\delta_1)^2}-1\right)U_N{}^2\left({}^{\Omega}\!/_{\Omega_p}\right)\right]} \tag{2.5}$$

Here $U_n(X)$ is a Jacobian elliptic function of order N, given by

$$U_N(x) = \int_0^x \frac{dy}{\sqrt{(1-y^2)(1-N^2y^2)}} \tag{2.6}$$

2.2 FIR FILTER DESIGN

Windowing method: Consider the ideal, or "brick wall," digital lowpass filter with a cutoff frequency of ω_0 rad/s. This filter has magnitude 1 at all frequencies with magnitude less than ω_0, and magnitude 0 at frequencies with magnitude between ω_0 and π. Its impulse response sequence $h(n)$ is

$$h(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H(\omega) e^{j\omega n} d\omega = \frac{1}{2\pi} \int_{-\omega_0}^{\omega_0} e^{j\omega n} d\omega = \frac{\omega_0}{\pi} \sin c(\frac{\omega_0}{\pi} n) \tag{2.7}$$

This filter is not implemental since its impulse response is infinite and noncausal. We can make an FIR filter by taking only a finite number of impulse response samples. If a causal filter is desired, we can then shift the truncated impulse response to the right until the samples are all indexed by a non-negative integer. These two operations are known as *windowing* and *delaying.For* example: design an FIR length-N (where N is odd) causal lowpass filter with cutoff frequency wc. Desired frequency response is

$$H_d(e^{j\omega}) = \begin{cases} 1, & |\omega| < \omega_c \\ 0, & otherwise \end{cases} \tag{2.8}$$

Desired impulse response is,

$$h_d[n] = \frac{\sin(\omega_c n)}{\pi n} \tag{2.9}$$

Truncated impulse response is

$$\hat{h}[n] = \begin{cases} \dfrac{\sin(\omega_c n)}{\pi n}, & |n| \leq \dfrac{N-1}{2} \\ 0 & otherwise \end{cases} \tag{2.10}$$

Truncated and delayed impulse response is

$$h[n] = \hat{h}\left[n - \frac{N-1}{2}\right] = \begin{cases} \dfrac{\sin\left(\omega_c\left(n - \dfrac{N-1}{2}\right)\right)}{\pi\left(n - \dfrac{N-1}{2}\right)}, & 0 \leq |n| \leq N-1 \\ 0, & otherwise \end{cases} \tag{2.11}$$

The windowing operation can be expressed in the time domain via

$$\hat{h}[n] = h_d[n]\omega[n] \tag{2.12}$$

Where $w[n]$ is the rectangular window

$$\omega[n] = \begin{cases} 1, & |n| \leq \dfrac{N-1}{2} \\ 0, & otherwise \end{cases} \tag{2.13}$$

Some common windows are

$$\text{Rectangular} \quad \omega[\mathrm{n}] = \begin{cases} 1, & 0 \le n \le M \\ 0, & \textit{otherwise} \end{cases}$$

$$\text{Hanning} \quad \omega[\mathrm{n}] = \begin{cases} 0.5 - 0.5\cos(2\pi n / M), & 0 \le n \le M \\ 0, & \textit{otherwise} \end{cases}$$

$$\text{Hamming} \quad \omega[\mathrm{n}] = \begin{cases} 0.54 - 0.46\cos(2\pi n / M), & 0 \le n \le M \\ 0, & \textit{otherwise} \end{cases}$$

$$\text{Blackmann} \quad \omega[\mathrm{n}] = \begin{cases} 0.42 - 0.5\cos\frac{2\pi n}{N-1} + 0.08\cos\frac{4\pi n}{N-1}, & |n| \le \frac{N-1}{2} \\ 0, & \textit{otherwise} \end{cases}$$

(2.14)

The problem with the standard window design method is that it's difficult to tradeoff between attenuation and transition bandwidth and provide us with a *fixed* approximation error. our only control variable so far has been the window length, which only allows us to control the transition bandwidth. Kaiser window allows having lower approximation error. The window is given by,

$$K[n] \;=\; \begin{cases} Io\left(\beta\sqrt{1-\left(1-\frac{2n}{M}\right)^2}\right), & 0 \le n \le M \\ 0, & \textit{othterwise} \end{cases} \qquad (2.15)$$

$I_0(x)$ is the zeroth order modified Bessel function

M is the filter order and, as usual, it controls the transition bandwidth

β is the parameter that changes the shape of the window and it controls the window's side lobe attenuation

Kaiser provided some empirical design formulas:

$$M \approx \frac{A-8}{2.285\Delta\omega}$$

$$\beta \approx \begin{cases} 0.1102(A-8.7), & \text{if A} > 50 \text{ dB} \\ 0.5842(A-21)0.4+0.07886(A-21), & \text{if } 21\text{dB} \le \text{A} \le 50\,\text{dB} \\ 0, & \text{otherwise} \end{cases} \qquad (2.16)$$

Where Δω=|ωs-ωp |is the transition bandwidth and A = -20 log10 max {δ1, δ2} is the attenuation.

The Frequency Sampling Technique: In this method the desired frequency response is sampled at a set of equally spaced frequencies to obtain N samples. The basic approach is to specify the desired magnitude of the frequency response, $|H_d(w)|$, at a set of frequencies $w_k = k * d_w = k * 2*pi/N$. uniform sampling of the frequency response is represented by

$$H_d(k) \;= H_d(e^{j\Omega_k}) \;= \sum_{n=0}^{N-1} h_d(n)e^{-j2\pi kn/N} \qquad (2.17a)$$

Thus by using the IDFT formula, the filter coefficients can be calculated using the following formula

$$h(n) = \frac{1}{N}\sum_{n=0}^{N-1} H(k)e^{j\left(\frac{2\pi n}{N}\right)k} \qquad (2.17b)$$

Now using the above N-point filter response, the continuous frequency response is calculated as an interpolation of the sampled frequency response. The approximation error would then be exactly zero at the sampling frequencies and would be finite in frequencies between them. The smoother the frequency response being approximated, the smaller will be the error of interpolation between the sample points.

Optimal Equiripple Design: In the window design method the ripples are not distributed evenly over the passband and the stopband that is, the peak ripple occurs near the band edges and decreases away from the band edges. If the ripples were more evenly distributed, we would be able to do a better job of approximating the ideal desired response. In fact, the best thing would be to have an *equiripple* filter, *i.e.* one where the ripple alternates in sign between two equal amplitude levels. Consider an error function $E(\omega)$ that measures the (weighted) deviation between the desired response and the filter's actual response:

$$E(\omega) = W(\omega)[Hd(e^{j\omega}) - H(e^{j\omega})] \tag{2.18}$$

The weighting function $W(\omega)$ allows us to control the relative approximation error between different bands, *i.e.* if $W(\omega)$ is large for some ω_0 relative to other frequencies, it means that we would like the error at that frequency to be small (relative to the approximation errors at other frequencies).an equiripple filter can be achieved by minimizing the maximum weighted error, $|E(\omega)|$, in both the passband and stopband. Mathematically, this can be expressed as

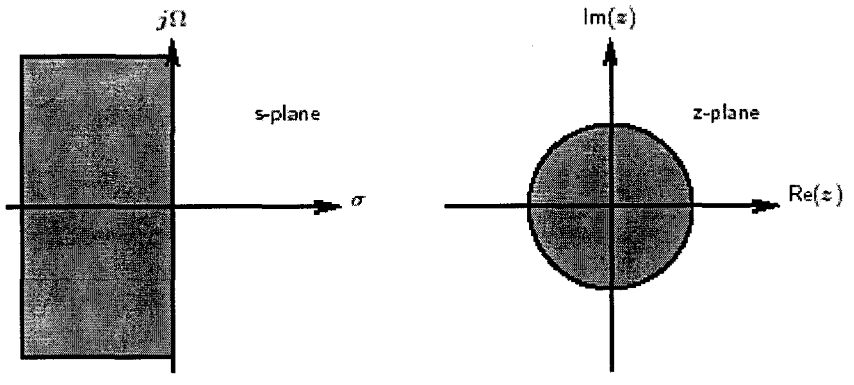
$$\min\,[\max |E(\omega)|] \tag{2.19}$$

over the passbands and stopbands.The objective then becomes to find an FIR linear phase filter that minimizes the above cost function. Parks & McClellan [19] solved it in the early 1970's based on an algorithm from the 1950's called *Remez Exchange.* In MATLAB the remez function implements the Parks-McClellan algorithm.

2.3 IIR FILTER DESIGN

After designing analog filters we transform them to discrete-time IIR filters. There are many different possible transformations that we can employ, but for a transformation to be legal, it must satisfy the following criteria:

Poles on the *j* axis in the *s*-plane must map to poles on the unit circle in the *z*-plane Poles in the left half of the *s*-plane must map to poles inside the unit circle in the *z*-plane.Now let's look at some of the most popular transformations: Impulse invariance and the bilinear transform.



$H_c(s) \leftrightarrow H(z)$

Impulse Invariant Transformation: The impulse invariant transformation converts the analog filter to a discrete-time one by sampling the impulse response of the analog filter.The impulse invariant transformation is:

$$\frac{1}{(s+d_i)} \leftrightarrow \frac{1}{1-z^{-1}e^{-d_iT}} \qquad (2.20)$$

The impulse invariant design requires H(s) in partial fraction form, and yields H(z) in the same form. Poles of H(s) in the left half plane map into poles of H(z) inside the unit circle, since |exp(-pT)|<1, thus, a stable H(s) transforms to a stable H(z).

Matched z-transform: The matched z-transform is

$$s + a \leftrightarrow 1 - z^{-1} e^{-at} \tag{2.21}$$

The zeros and poles of the filter in the s-domain were obtained then the transformation was performed. The matched z-transform converts an all pole analog system to an all pole digital system but may not preserve the frequency response of the analog system. it also suffers from aliasing errors.

Bilinear Transformation: Aliasing is a major problem with impulse invariance. Bilinear transformation avoids aliasing by performing a one-to-one mapping from the *s*-plane to the *z*-plane. To obtain the transfer function H(z) we apply the following transformation

$$s = \frac{2}{T}\left(\frac{z-1}{z+1}\right). \tag{2.22}$$

It is the mapping most often employed for digital filter design, it is simple to apply, provides a one to one mapping between the s-plane and z-plane, and avoids aliasing problems. It can thus be used for all types of systems (including high pass and band stop). It always maps a stable analog system into a stable digital system. It does suffer from warping effects that could be fixed by using a simple relation. It maps H(s) described in either cascaded or partial fraction forms.

2.4 DESIGN OF FILTER BANKS

A digital filter bank is a set of band pass filters with either common input or summed out put, each of which covers a band in the frequency spectrum. Other possible components of a filter bank include down samplers, up samplers and delay elements.

Decimation by a Factor D: Decimation is defined as the process of reducing the sampling rate by a factor of D, or in other words, downsampling by D. If a signal, x(n), were to be decimated by an integer factor D by selecting every Dth value of it, the result will be an aliased version of x(n), with a folding frequency of $F_x/2D$. Following Nyquist criterion, to avoid the aliasing, the bandwidth of x(n) must be reduced to $F_{max} = F_x / 2D$ or $\omega_{max} = \pi / D$. Then, we can downsample by D and avoid aliasing.

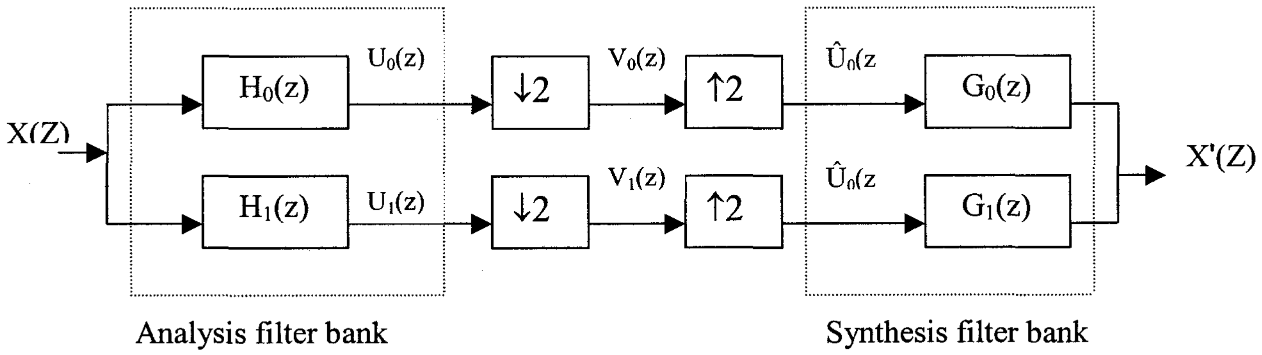


Figure 2.1: Two channel QMF filter bank

The decimation process can be seen in Figure 2.2. The input is filtered by a lowpass filter, with impulse response h(n) and a frequency response $H_D(\omega)$:

$$H_D(\omega) = \begin{cases} 1, & |\omega| \le \pi / D \\ 0, & otherwise \end{cases} \tag{2.23}$$

This will only pass the frequency components of x(n) in the range $|\omega| \leq \pi / D$ which is what we need in further process.

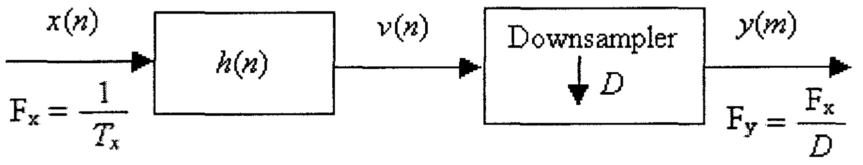


FIGURE 2.2: Decimation by a factor D

$$v(n) = \sum_{k=0}^{\infty} h(k)x(n-k) \tag{2.24}$$

v(n) is the output of the filter h(n) which is downsampled by D to get y(m) :

$$y(m) = v(mD)$$

$$\Rightarrow y(m) = \sum_{k=0}^{\infty} h(k)x(mD-k) \tag{2.25}$$

y(m) is the downsampled sequence with sampling rate of Fy = F_x / D, derived from x(n), which is what we expected to get from the decimation process.

Interpolation by a Factor I: Interpolation is the process of increasing the sampling rate by an integer factor of I. It is accomplished by adding I-1 zeros between successive values of the signal (in our case, x(n)), resulting in a new sequence v(m) with a rate $F_y = I\ F_x$:

(2.26)

$$v(m) = \begin{cases} x(m/I), & m = 0, \pm I, \pm 2I, \ldots\ldots \\ 0, & otherwise \end{cases}$$

The spectrum of $V(\omega_y)$ is an I-fold periodic repetition of $X(\omega_x)$ the input signal spectrum.

Next, we have to reject all the images of $X(\omega)$ above $\omega_y = \pi / I$, thus preserving the unique frequency components of $x(n)$ in the range $0 \leq \omega_y \leq \pi / I$. This is accomplished by passing $v(m)$ through a lowpass filter with frequency response $H_I(\omega_y)$:

$$H_I(\omega_y) = \begin{cases} C, & 0 \leq |\omega| \leq \pi / I \\ 0, & otherwise \end{cases} \tag{2.27}$$

where is a scaling factor for normalizing the output sequence $y(m)$.$y(m)$ is the result of convoluting $v(n)$ with $h(n)$, the impulse response of the lowpass filter, thus resulting in :

$$y(m) = \sum_{k=-\infty}^{\infty} h(m-k)v(k) \tag{2.28}$$

But, having the fact that $v(k) = 0$ except at multiples of I, where $v(kI) = x(k)$, gave us :

$$y(m) = \sum_{k=-\infty}^{\infty} h(m-kI)x(k) \tag{2.29}$$

2.5 FIR QMF BANK

There are different types of filter banks depending on the number of chanels, types of analysis and synthesis filter (FIR or IIR) etc. Our research focuses on 2-channel QMF FIR filter-bank design.. As shown in Fig. 1.2, a 2-channel analysis filter bank splits the frequency space into two parts, with the low frequency bank ranging from 0 to $\frac{\pi}{2}$ and the high frequency bank from $\frac{\pi}{2}$ to π. The amplitude response of H_1 is the mirror image of the amplitude response of H_0 with respect to the quadrature frequency $\frac{\pi}{2}$ hence the name quadrature mirror filter (QMF). The expressions for various intermediate signals in fig 2.1 are given as

$$U_k(z) = H_k(z)X(z) \tag{2.30}$$

$$V_k(z) = 1/2\{U_k(z^{1/2}) + U_k(-z^{1/2})\} \tag{2.31}$$

$$\hat{U}_k(z) = V_k(z^2) \tag{2.32}$$

Where,

$$H_k(z) = \sum_{n=0}^{N-1} h_0(n) z^{-n} \tag{2.33}$$

The ouput of the filter bank is given as

$$Y(z) = \frac{1}{2}\{G_0(z)H_0(z) + G_1(z)H_1(z)\}X(z) + \frac{1}{2}\{G_0(z)H_0(-z) + G_1(z)H_1(-z)\}X(-z) \tag{2.34}$$

The above can be written in a simplified form as

$$Y(z)=T(z)X(z)+A(z)X(-z) \tag{2.35}$$

Perfect Reconstruction (PR), which is a desirable property for filter-bank design, is obtained when the reconstructed signal x(n) is exactly the same as the input signal. In case of FIR QMF bank an aliasing caused due to the second term in the equation (2.34) and can be eliminated by choosing,

$$G_0(z) = H_1(-z) \tag{2.36}$$

$$-G_0(-z) = G_1(z) \tag{2.37}$$

$$H_1(z) = H_0(-z) \tag{2.38}$$

After above settings only one prototype filter $H_0(z)$ exists in the system.T(z) in equation (2.35) is called as distortion transfer function

$$T(z) = \tfrac{1}{2}\{ G_0(z)\, H_0(z) + G_1(z)\, H_1(z)\}. \tag{2.39}$$

As T(z) exhibits linear phase characteristic QMF bank of fig 2.1 has no phase distortion.QMF bank will exhibit amplitude distortion until $|T(e^{jw})|$ is constant for all values of w.One way to minimize amplitude distortion is to adjust the filter coefficients of $H_0(z)$ such as

$$|H_0(e^{jw})|^2 + |H_1(e^{jw})|^2 \cong 1 \text{ for all values of w.} \tag{2.40}$$

To this end with zero aliasing and phase distortions the FIR QMF bank design can be specified as a problem that searches for $h_0(n)$ which improves ,The overall performance of the filter bank by minimizing the amplitude distortion ,The individual performance of the composing filter by satisfying the magnitude response of $H_0(z)$ with $H_0(z) \cong 1$ in its pass band $H_0(z) \cong 0$ in its stop band.

CHAPTER 3
Canonical Signed Digit (CSD) Numbers
Properties,Conversion algorithm and Applications

Canonic Signed Digit (CSD) number representation is the representation of a given number as a sum and difference of powers of two. Multiplication to a CSD number is cheap since it requires fewer sums of shifted versions of the multiplicand when compared with multiplication to a binary number.

CSD implementation of FIR filter taps results in a realization of fast multiplication [14,18,48]. This dedicated arithmetic reduces the hardware complexity of the FIR QMF bank. The CSD arithmetic is the basis for efficient implementation of the modulator and demodulator filters and this chapter provides a general overview of CSD numbers and their properties.

3.1 DEFINITION

For a given M-bit of a number x Є $[-2^{M-1}, 2^{M-1}]$, the signed digit representation can be related to its 2's-complement version through :

$$x = -x_{M-1}2^{M-1} + \sum_{k=0}^{M-2} x_k 2^k \triangleq \sum_{k=0}^{M-1} s_k 2^k \tag{3.1}$$

Where x_k Є {0,1}, s_k Є {-1,0,1}.The Canonical Signed Digit (CSD) representation is the signed digit representation in which

$$s_k s_{k-1} = 0 \qquad (3.2)$$

for k = 1... M – 1.Signed digits are ternary, in contrast with 2's-complement digits which are binary. The value of a CSD number can be obtained by summing the $S_k \neq 0$ terms. Also, FIR filter coefficients are always normalized in the range x €[-1,1). Therefore, an alternative representation of CSD numbers that has been used in the literature is,

$$x = \sum_{k=0}^{M-1} s_k 2^{-pk} \qquad (3.3)$$

where L is the number of nonzero digits and p_k € {0,.......M-1}

3.2 PROPERTIES

Uniqueness: There could be more than one signed digit representations for a given binary number. For example, $00010\bar{1}$ and $001\bar{1}0\bar{1}$ both represent 3. Among these representations, the CSD representation is unique.

Minimality: CSD numbers are minimal in the sense that, among all signed digit representations; the CSD representation has the minimum number of nonzero digits (L). it is known that the probability of a canonical digit S_k to be nonzero is 1/3 for large M. This compares to a value of 2/3 for 2's – complement numbers. In general, there are $\frac{M+1}{2}$ (M odd) or $\frac{M}{2}$ (M even) M-bit CSD numbers which have the maximum number of nonzero digits, compared to the total of 2^M possible M-digit binary numbers. This ratio is 25%, 6.25%, and 0.39% for M=5, 8, 16, respectively.

***Nonuniform Distribution*:** The set of all representable CSD numbers with a fixed L is non-uniformly distributed. In other words, there are non – equal gaps between consecutive CSD numbers with the same number of nonzero digits. Furthermore, this distribution is denser for smaller numbers.

***Maximum L*:** A Positive number less than or equal to $\frac{2^{2L+1}-2}{3}$ requires at most L power-of-two terms to present. The number that can be represented by at most L power-of-two terms increases as 2^{2L} whereas the number that can be represented by an L-bit word increase as 2^L. This property is consistent with the sparse and nonuniform nature of the CSD numbers discussed above.

***Reduced Exponent Set*:** Equation 3.3 states that in general, there are M possible values for P_k. It has been shown that this number can be reduced. Let $S_{M,L}$ be the set of all CSD number which can be generated by Equation 3.3 for a given M and L. Let $Z_{M,L}(k) \subset \{0,\ldots,L\}$, $1 \le k \le L$ be a set of successive integers given by

$$Z_{M,L}(k) = \{2(k-1)+1,..,(M-1)-2(L-k)\}. \qquad (3.4)$$

Then $S_{M,L}$ can be generated from Equation 3.3 with $P_k \in Z_{M,L}(k)$. The number of elements in $Z_{M,L}(k)$ is $M - 2L + 2$ which is less than M.

3.3CONVERSION ALGORITHM

The conversion for recording an M-bit 2's – complement number x to its CSD representation y is as follows :

1. Let $x_M = x_{M-1}$ and $C_0 = 0$.
2. For $i = 0,\ldots, M-1$ Let $C_{i+1} = x_i x_{i+1} + x_i C_i + x_{i+1} C_i$.

3. For i = 0,..., M – 1 Let $y_i = x_i + C_i - 2C_{i+1}$.

Notice that the "+" in step 2 is a logical OR,while the "+" step is an arithmetic sum. As an example, consider – 5/8 = $(1.011)_{2\text{'s - complement}}$ = $(0.\bar{1}0\bar{1})_{CSD}$. The conversion algorithm for this example is shown in Table 3.1.

A unique CSD number can exactly represent every binary number. A fractional decimal number, however, may not have an exact CSD equivalent, since, it may not have a finite binary representation. Therefore, the number of bits of the CSD representation of a fractional decimal number depends on the tolerable truncation error.

i	x_i	x_{i+1}	c_i	c_{i+1}	y_i
0	1	1	0	1	$\bar{1}$
1	1	0	1	1	0
2	0	1	1	1	$\bar{1}$
3	1	1	1	1	0

TABLE 3.1: CSD conversion example

3.4 APPLICATIONS

The CSD number properties discussed in the previous make it appealing in high speed DSP applications. The key idea in such applications is the ease of multiplying a number in a power of two. Multiplication of a given number by a CSD number can be achieved by adding/subtracting some shifted versions of the input. This eliminates the need for multipliers which are both slow and impose some other limitations such as space requirement and high power consumption.

As discussed before, CSD representation introduces truncation errors when used for fractional numbers. This truncation error must be acceptable in a given application. In the case of FIR filters, it has been shown that proper CSD representation of filter coefficients could result to a negligible distortion in filter frequency response.

Fiter bank Multiplication arithmetics: Traditional FIR filters in QMF banks uses real or fixed-point numbers as filter coefficients. Multiplication of such long floating-point numbers generally limits the speed of filtering. To overcome this limitation, filters with canonical signed digit coefficients have been proposed. Before describing the multiplierless concept, we first note how a single arithmetic shift left (ASL) corresponds to a multiplication by two. Similarly, an n-bit shift left or right (ASR) corresponds to a multiplication or division by 2^n, respectively.
A binary multiplication represents a computationally intensive and relatively complex operation in any processor. Binary multiplications consist of a series of shifts and additions, which in turn require additional carry and overflow-handling logic. This additional complexity and delay can greatly hamper the performance of processors that are required to perform arithmetically intensive computations When multiplying a filter input (multiplicand) with one such coefficient (multiplier), the product can be founded by adding and shifting the number of times corresponding to the number of ONE bits in the multiplier.

As observed above multiplication involves two basic operations: generation of partial products and their accumulation. Smaller number of partial products reduces the complexity, reduces the time needed to accumulate and hence speed up the multiplication. A limited sequence of shifts and adds are usually much faster than full multiplication. The number of required arithmetic operations, ignoring shifts (which come free), can be reduced by expressing coefficients in canonical-signed-digit (CSD) form. CSD is a radix-two number system with ternary coefficient set {$\bar{1}$; 0; 1} and having the "canonical" property that $\bar{1}$ and 1 are always followed by 0 in CSD strings. CSD string 1 0 0 . $\bar{1}$0 1, for example, represents 3.625. So a CSD coefficient specifies which input-signal shifts to add to the output, which to subtract from it, and which to ignore.

Application of CSD number representation in the design of multiplierless realization[18] of FIR filters has been widely investigated in the literature The minimality property of CSD numbers implies that a minimum number of shifter are needed for the CSD number representation. The number of zeros in a CSD number only affects the number of shifts. Below, in Figure 3.1, we trace results through the required steps of a sample binary multiplication.

```
     Real multiplication                    CSD multiplication


 13   00001101                         13   00001101
×15   00001111                        ×15   00010001̄
      00001101 add                          00001101  subtract
    00001101  add                       00001101     (shift 3 times add)
   00001101   add                           11000011=(104)10
  00001101    add
  00011000011 =(104)10
```

FIGURE 3.1: Standard multiplication and CSD multiplication

Note how the standard multiplication by a binary number in the above example involves four additions and five carries, where as with multiplication with CSD numbers involve one addition and one subtraction. Note that the required numbers of shift operation are same for both methods.

CHAPTER 4
GENETIC ALGORITHMS

Introduction to GA terminology,Genetic operators and Design

Genetic algorithms are search algorithms based on the mechanics of natural selection, genetics, and evolution. It is widely accepted that the evolution of living beings is a process that operates on chromosomes – organic devices for encoding the structure of living beings. Natural selection is the link between chromosomes and the performance of the decoded structures. Processes of natural selection cause chromosomes that encode successful structures to reproduce more often than those that do not. In addition to reproductions, mutations may cause the chromosomes of children to be different from those of their biological parents, and recombination process may create quite different chromosomes in children by combining from the chromosomes of their two parents.

These features of natural evolution inspired the development of GAs. Roughly speaking, through a proper encoding mechanism GAs manipulate strings of binary digits (1s and 0s) called chromosomes, which represent multiple points in the search space. Each bit in string is called allele. They carry out simulated evolution on populations of chromosomes. Like nature, GAs solve the problem of finding good chromosomes by manipulating the material in the chromosomes blindly with out any knowledge about the type of problem they are solving. The only information they are given is an evaluation of each chromosome they produce. This evaluation is used to bias the selection of chromosomes so that those with the best evaluations tend to reproduce more often than those with bad evaluations. Genetic algorithms, using simple manipulations of chromosomes such as simple encoding and reproduction mechanisms, can display complicated behavior and solve some extremely difficult problems with out knowledge of the decoded world.

4.1 GA TERMINOLOGY

All genetic algorithms work on a population, or a collection of several alternative solutions to the given problem, each individual in the population is called a string or chromosome. Often these individuals are coded as binary strings, and the individual characters or symbols in the stings are referred to as genes. In each iteration of the GA, a new generation is evolved from the existing population in an attempt to obtain better solutions.

The *population* size determines the amount of information stored by the GA. the GA population is evolved over a number of generations.

An *evaluation function* (or the fitness function) is used to determine the fitness of each candidate solution. The fitness is the opposite of what is generally known as the cost in the optimization problems.

Individuals are selected from the population for reproduction, with the *selection* biased toward more highly fit individuals. Selection is one of the key operators on the GA that ensure the survival of the fittest. The selected individuals form pairs, called parents.

Crossover is the main operator used for reproduction. it combines portions of two parents to create two new individuals, called offspring, which inherit a combination of the features of the parents. For each pair of parents, crossover is performed with a high crossover probability P_c, which is called crossover probability. With the probability 1-P_c, crossover is not performed, and the offspring pair is same as the parent pair.

Mutation is an incremental change made to each member of the population, with a very small probability. Mutation enabled new features to be introduced into a population. It is performed probabilistically such that the probability of a change in each gene is defined as the mutation probability, P_m.

Inversion is a genetic operator, which does not change the solution represented by the chromosome, but rather changes the chromosome itself, or the representation of the solution. The inversion probability is denoted by P_i.

The generation gap is the fraction of individuals in the population that are replaced from one generation to the next and is equal to one for the simple GA.

4.2 GENETIC OPERATORS

The simple GA flowchart shown in figure 4.1. A GA in its simplest form uses three operators: reproduction, crossover and mutation.

4.2.1 Reproduction:

Reproduction is a process in which individual strings are copied according to their fitness values. This operator is an artificial version of natural selection. A fitness is assigned to each individual in the population, where high numbers denote good fit. Fitness is defined as being inversely proportional to the squared sum of difference between ideal and desired magnitude response. The reproduction (parent selection) process is conducted by spinning a simulated biased roulette wheel whose slots have different sizes proportional to the fitness values of the individuals. This technique is called roulette wheel parent selection. Each time an offspring is needed a simple spin of weighed roulette wheel yields

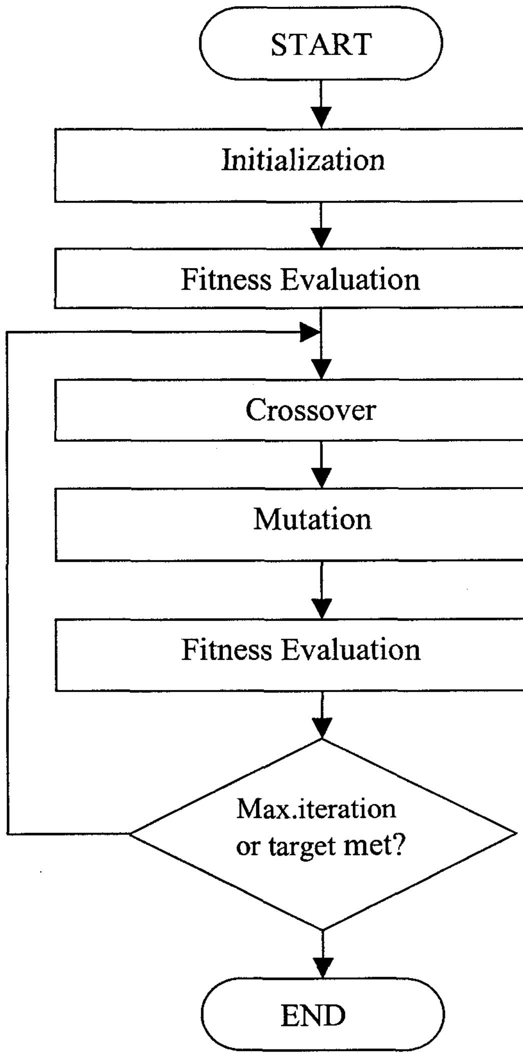


FIGURE 4.1: Basic flowchart

the reproduction candidate. This technique can be implemented algorithmically as in the following steps:

1. Sum the fitness of all the population members and call this result the total fitness.
2. Generate n, random number between zero and total fitness.
3. Return the first population member whose fitness, added to the fitness of the preceding population members (running total) is greater than or equal to n.

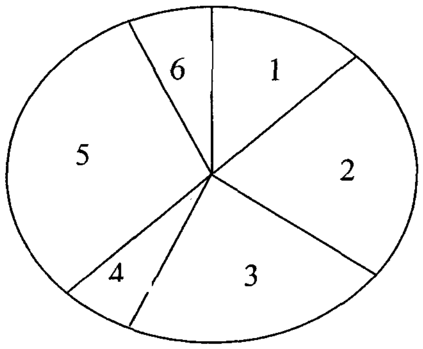


FIGURE 4.2: Roulette wheel parent selection

For example in Fig 4.2 the circumference of the roulette wheel is the sum of all six individuals fitness values. Individual five is the fittest individual and occupies largest interval where as individuals six and four are the least fit and have correspondingly smaller intervals with in the roulette wheel. To select an individual, a random number is generated in the interval [0,Sum] and individual whose segment spans the random number is selected. This process is repeated until the desired numbers of individuals have been selected.

4.2.2 Crossover:

Reproduction directs the search toward the best existing individuals but doesn't create any new individuals. In nature an offspring has two parents and inherits jeans from both. The main operator working on the parents is crossover, which happens for a selected pair with a crossover probability P_c. At first two strings from the reproduced population are mated at random, and crossover site is randomly selected. Then the strings are crossed and separated at the site. This process produces two new strings each of which takes after both parents. This type of crossover is known as single point crossover as there is one crossover site.

Multipoint crossover: For multipoint crossover, various crossover positions are chosen at random. Then, the bits between successive crossover points are exchanged between the two parents to produce two new offspring. This process is illustrated in fig 4.3. The idea behind multi point is that parts of the chromosome representation that contribute to the most of the performance of a particular individual may not necessarily be contained in adjacent sub strings. Multipoint crossover appears to encourage the exploration of the search space, rather than favoring the convergence to highly fit individuals, thus making the search more robust.

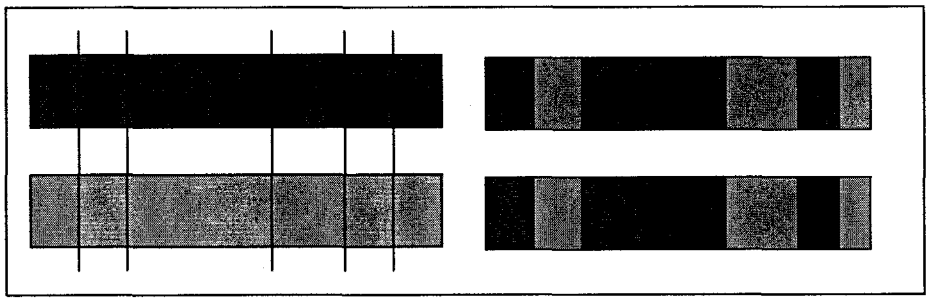


FIGURE 4.3: Multipoint Crossover

The idea behind multi point is that parts of the chromosome representation that contribute to the most of the performance of a particular individual may not necessarily be contained in adjacent sub strings. Multipoint crossover appears to encourage the exploration of the search space, rather than favoring the convergence to highly fit individuals, thus making the search more robust.

Uniform crossover:Single and multi point crossover define cross points as places between loci where a chromosome can be split. Uniform crossover generalizes this scheme to make every locus a potential crossover point. A crossover mask, the same length as the chromosome structure is created at random and the parity of the bits in the mask indicated which parent would supply the offspring with which bite. Consider the following example, crossover mask and the resulting off spring:

Parent1	=	1	0	1	1	0	0	0	1	1	1
Parent2	=	0	0	0	1	1	1	1	0	0	0
Mask	=	0	0	1	1	0	0	1	1	0	0
Offspring1	=	0	0	1	1	1	1	0	1	0	0
Offspring2	=	1	0	0	1	0	0	1	0	1	1

FIGURE 4.4: Uniform Crossover

Here, the offspring 1 is produced by taking the bit from parent1 if the corresponding mask bit is 1 or the bit from parent2 if the corresponding mask bit is 0. Offspring2 is created using the inverse of the mask.

4.2.3 Mutation:

Although reproduction and crossover produce many new strings they do not introduce any new information into the population at the bit level. As a source of new bits, mutation is introduced and is applied with low probability P_m. It inverts randomly chosen bit on a string. Mutation should be used sparingly because with high mutation rates the algorithm will become little more than a random search.

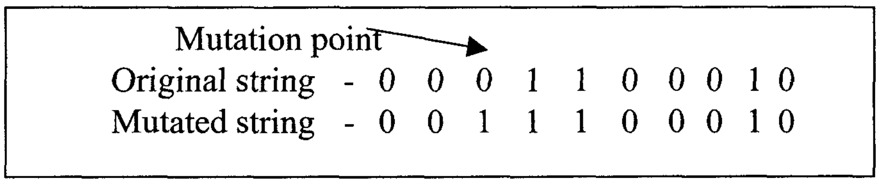

Mutation point
Original string - 0 0 0 1 1 0 0 0 1 0
Mutated string - 0 0 1 1 1 0 0 0 1 0

FIGURE 4.5: Mutation

In general, genetic algorithms (GAs) rely upon the law of fittest member survival to optimize a set of possible outcomes or results. Loosely based on the laws of Darwinian genetics, GAs has many unique characteristics that make them ideal for many optimization problems. As Goldberg states (1989),“They [GAs] combine survival of the fittest among string structures with a structured, yet randomized information exchange to form a search algorithm with some of the innovative flair of human search.” Since our goal is to optimize FIR filter bank to contain CSD coefficients, traditional design techniques are unsuitable for the construction of such filters. Restricting coefficients to CSD numbers forces us to explore other methods. The proposed technique, the deviations from the basic design and the experimental results are discussed in the following chapter.

CHAPTER 5
Proposed Design method and Experimental results

Design of FIR QMF bank with CSD coefficients using genetic algorithms is focused in this thesis. Since our goal is to optimize FIR filter bank to contain CSD coefficients, traditional design techniques are unsuitable for the construction of such filters. Restricting coefficients to CSD numbers forces us to explore other methods. The proposed technique employs a genetic algorithm to search for CSD filter coefficients.

In this chapter we present a genetic algorithm based design method followed by several experimental results of the proposed design. We can observe the performance of genetic algorithm in designing the QMF bank with respect to reconstruction error of the overall filter bank and individual error, sum of pass band error ,stopband error,passband and stopband ripplesand transition width, of the composing prototype filter.

5.1 DESCRIPTION OF DESIGNED GENTIC ALGORITHM

The genetic algorithm designed for this project is based upon conventional GA principles and framework: the algorithm starts with a random population, evaluates member fitnesses, breeds randomly selected fittest members, occasionally mutates and then repeats. It should be noted that since the optimization performed in this case is quite specific (CSD criterion), the designed GA must be accurately and explicitly defined. At the same time, the GA is designed in a modular fashion through the use of variable inputs in order to make important changes regarding the operation of the algorithm in a relatively straightforward manner.The designed algorithm makes several rapid departures from conventional GA design. These different approaches in design are due to the precise nature of multiplier less optimization and the notion of representing FIR QMF bank in terms of coefficient sets. The flowchart detailing the designed GA is shown in Figure 5.1

5.1.1 Parent Selection

In order to reproduce offspring, parents need to be selected. The most commonly used method, roulette wheel selection, is used in this paper. The idea behind the Roulette wheel selection technique is that each individual is given a chance to become a parent in proportion to its fitness. It is called roulette wheel selection as the chances of selecting a parent can be seen as spinning a roulette wheel with the size of the slot for each parent being proportional to its fitness. Obviously those with the largest fitness (slot sizes) have more chance of being chosen. Thus, it is possible for one member to dominate all the others and get selected a high proportion of the time.

5.1.2 Crossover and Mutation

Crossover and mutation are two basic operators of GA. Performance of GA depends on them very much. Single point crossover is used in this paper. The crossover operator performs the exchange of information between the individuals selected for breeding. One cutpoint is randomly selected from both parents, and the tails of the strings are swapped. This causes that the resulting individual to be different from either of its parents.This can be seen from results shown in figure 5.2.

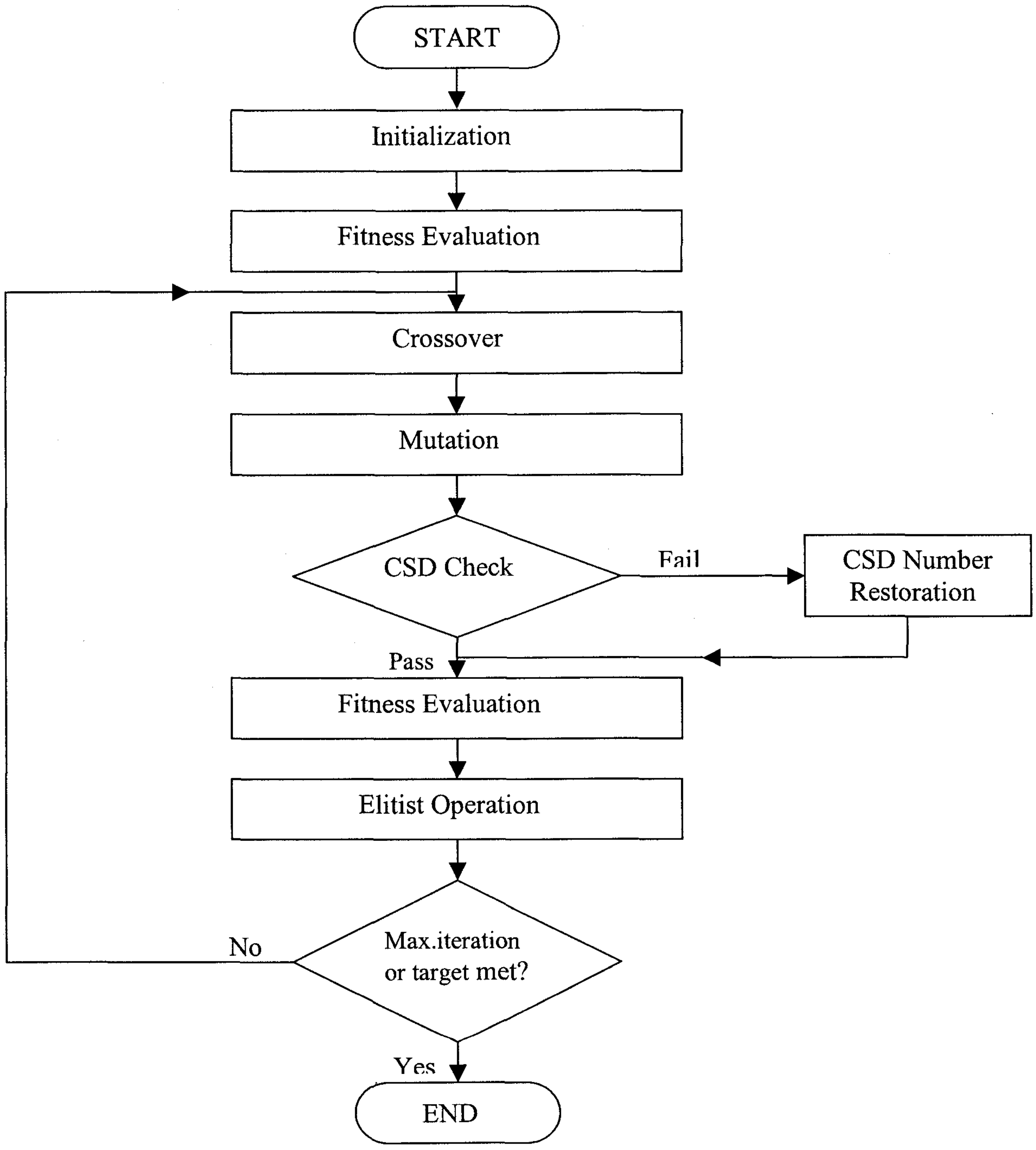


FIGURE 5.1: Flowchart of Designed Genetic Algorithm

Parent1	Parent2
001 10	1 1 0 1 1
Offspring1	Offspring2
0 0 1 1 1	1 1 0 1 0

FIGURE 5.2: Single point crossover technique

As source of new bits mutation is introduced,which inverts a randomly chosen bit on a string and is applied with lower probability ,P_m,otherwise the algorithm will be little more than random search. After crossover and mutattion each newly created bit stream(offspring) is examined to see if the individual coefficients satisfy the CSD constraints.Any violated coefficient will be restored to its nearest CSD number.

5.1.3 CSD Restoration

After crossover and mutattion each newly created bit stream(offspring) is examined to see if the individual coefficients satisfy the csd constraints.Any violated coefficient will be restored to its nearest CSD number.Several techniques have been used before such as using special encoding scheme in GA such as in [15] or use CSD number restoration technique such as in [2] and[6]. The former method can limit the maximum number of non-zero digits while the can solve the problem of consecutive nonzero digits,but cannot limit the maximum number of non-zero digits.CSD restoration can be obtained by decoding any coefficient which has violated the CSD format to its decimal number and converting this number to its nearest CSD reprentation.By using this new restoration technique,

Decimal number1	Decimal of the CSD conversion result of Decimal	Conversion error
0.1509	0.1484	1.6143%
0.6970	0.6953	0.3705%
0.3784	0.3760	0.6334%
0.8600	0.8594	0.0740%
0.8537	0.8516	0.2451%
0.5936	0.5928	0.1330%
0.4966	0.4980	0.3010%
0.8988	0.8984	0.1480%
0.8216	0.8203	0.1603%
0.6449	0.6484	0.5469%
0.8180	0.8203	0.2858%
0.6602	0.6641	0.5809%
0.3420	0.3418	0.0508%
0.2897	0.2891	0.2290%
0.3412	0.3428	0.4630%
0.5341	0.5313	0.5297%
0.7271	0.7266	0.0757%
0.3093	0.3105	0.4063%
0.8385	0.8359	0.3051%
0.5681	0.5693	0.2224%
0.3704	0.3682	0.6073%
0.7027	0.7031	0.0548%
0.5466	0.5459	0.1231%
0.4449	0.4443	0.1223%
0.6946	0.6953	0.1073%
0.6213	0.6230	0.2795%
0.7948	0.7969	0.2584%
0.9568	0.9541	0.2866%
0.5226	0.5225	0.0248%
0.8801	0.8818	0.1924%

TABLE 5.1: Results of conversion technique

which is used in this paper, the consecutive nonzero zero digits and the limit on themaximum number of nonzero digits can be achieved.The conversion algorithm for recording to CSD representation is presented in chapter 3. The conversion result and their corresponding errors are shown in table 5.1. The average error of the above test result is 0.329%, which shows the conversion is very efficient.

5.1.4 Fitness Evaluation

Fitness evaluation is based on the set of coefficients as a whole. This is a departure from conventional GA design in that each member of the population is traditionally represented as a single binary string. In design of QMF bank the performance of filter depends on minimizing the reconstruction error of the overall filter bank and pass band energy, stopband energy, passband ripple, stopband ripple and transition band width of the individual filter response. To calculate the fitness the total error is calculated first. The error in the designed genetic algorithm compares the candidate multiplier less coefficient system responses from the population to a specified ideal filter's response in this paper total error is calculated as,

$$Terr=Er+w1(Es-cEs)+w2(Ep-cEp)+w3(\delta s-c\delta s)+w4(\delta p-c\delta p)+w5(\Delta w-c\Delta w) \quad (5.1)$$

Where Er, the reconstruction error of the overall filter bank, is considered as the main objective in this optimization techniques and pass band energy (Ep), stop band energy (Es), pass band ripple (δp), stop band ripple (δs) and transition band width (Δw) of the individual filter response as considered as constraints. cEs, cEp,cδs,cδp,cΔw are constraint values given by reference area and we choose Johnston's design the best known as our reference. Fitness is defined as being inversely proportional to total error

$$fitness = \frac{1}{totalerror(Terr)} \quad (5.2)$$

Probability of selection to each member is based on the relative fitness amongst one another. The highly fit individuals have more probability of being chosen into next generation.

5.1.5 Elitist Operation

The designed GA also incorporates a form of elitism. Elitism is a technique implemented in our algorithm to prevent the fitness regression of the population. Since even a crossing of the two fittest members could result in offspring that are highly unfit. The performance of GA may get worse as it progress. Elitism is employed to prevent the loss of the fittest members due to crossover and mutation. This guarantees that even when parents do not successfully yield fitter offspring, they are held over to the next generation to try again. The best chromosome of the parent population is copied into the worst chromosome of the offspring population if its fitness is less than the best of parent population. Elitism ensures that creating a copy of them before crossover and using them to replace the least fit members in the subsequent generation do not lose the fittest members.

5.1.6 GA Parameters

The genetic algorithm is designed to be able to optimize several different types of filters as well as to adapt and modify its population in different ways. To do this, the GA incorporates a variety of different variables and parameters that can be altered depending on the application.

The first parameter is the number of genetic iterations. This is an important variable as it determines how long the population breeds in an attempt to improve

the fittest member. Generally it can be said that the higher the number of iterations chosen, the fitter the members of the population become.
A second and equally important input variable to the GA is the filter order. The filter order determines not only how many coefficients make up each member of the population, but also the filter's ability to approximate its ideal specified counterpart. Generally, it can be said that higher order filters are necessary in order to realize sharper responses. To accommodate for this factor, it is necessary to vary the filter order depending on the application.

Wordlength and weight are important factors in the designed GA,a wordlength determines the the number of genes can be used to represent CSD coefficient and weight determines the number of nonzero digits in CSD number representation. They can be set to any desired value.

5.2 PERFORMANCE OF GENETIC ALGORITHM

In our experiment we have used GA to design FIR QMF bank. The customized genetic algorithm starts with a randomly chosen initial population which is in CSD format, goes through roulette wheel parent selection, single point crossover and mutation. After crossover and mutation the members of population may not be in CSD format. Each coefficient is checked for the CSD format if failed they are restored back to its form by proposed CSD restoration technique. These operations are carried out until the desired results are met. The multi objective QMF bank design is converted to single objective constrained problem by means of penalty method the total error of each individual is evaluated as

$$Terr=Er+w1(Es-cEs)+w2(Ep-cEp)+w3(\delta s-c\delta s)+w4(\delta p-c\delta p)+w5(\Delta w-c\Delta w) \quad (5.3)$$

Where Er, the reconstruction error of the overall filter bank, is considered as the main objective in this optimization techniques and pass band energy (Ep), stop band energy (Es), pass band ripple (δp), stop band ripple (δs) and transition band width (Δw) of the individual filter response as considered as constraints. cEs, cEp, cδs,cδp,cΔw are constraint values given by reference area and we choose Johnston's design the best known as our reference and are given in table 5.2. Fitness is defined as being inversely proportional to total error.

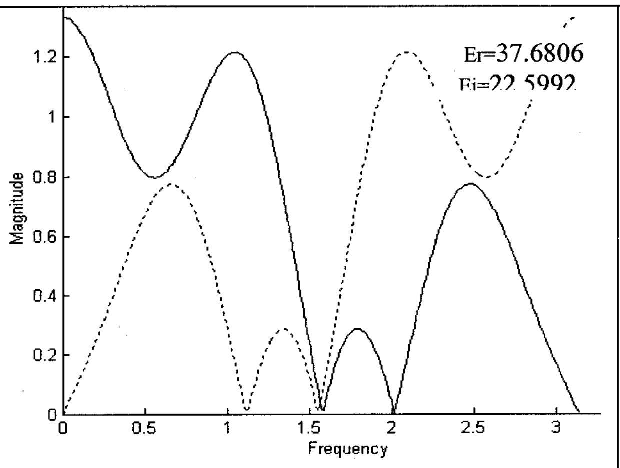
5.3 EXPERIMENTAL RESULTS

The design method described above has been used to design FIR QMF Banks of different order. The filter banks of order 12,24B, 24C, and 32 are designed. Note that 24 B and 24 C filter are of same order, 24, but have different transition bandwidths.

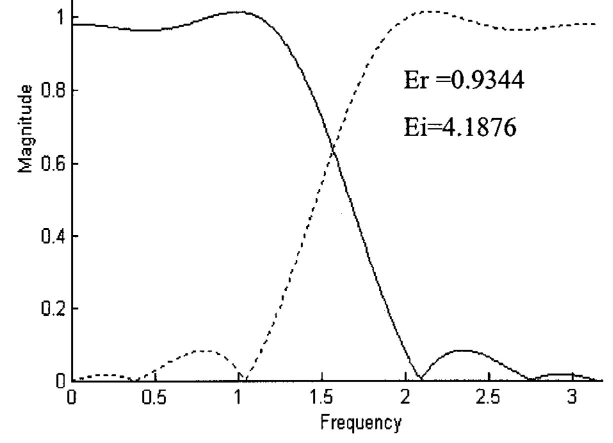
The individual filter error (Ei) of the prototype lowpass filter, sum of pass band energy (Ep) and stopband energy (Es), and the reconstruction error of the overall filter bank(Er)for all the designed filter banks are presented. The magnitude responses of the overall filter bank and the designed CSD coefficients are given.

In the first example a 12th order lowpass filter is designed. The results of using genetic algorithm for 12th order filter bank are presented in fig 5.3. In fig 5.3a the magnitude response of the initially chosen population is shown whose reconstruction error is 37.6806 and stopband error and passband error of the individual are15.8041 and 6.8255 respectively. The genetic algorithm operations, crossover, mutation, CSD restoration and elitist operations are carried out for some generations. Magnitude response of the filter bank after fifty generations is shown in fig 5.3b.In fig 5.3c the response of the filter bank after 100 generations is presented. The reconstruction error is 0.5058 and the passband and stopband errors are 0.5859 and 3.8119 respectively. We can see as the iterations increase the performance of the filter bank increases. The operations are carried out until the desired values of performance metrices are met or until the maximum number of iterations is done. The larger the number of the iterations the greater will be the performance. The resulting CSD coefficients of the designed 12th order filter bank are presented in table 5.3.

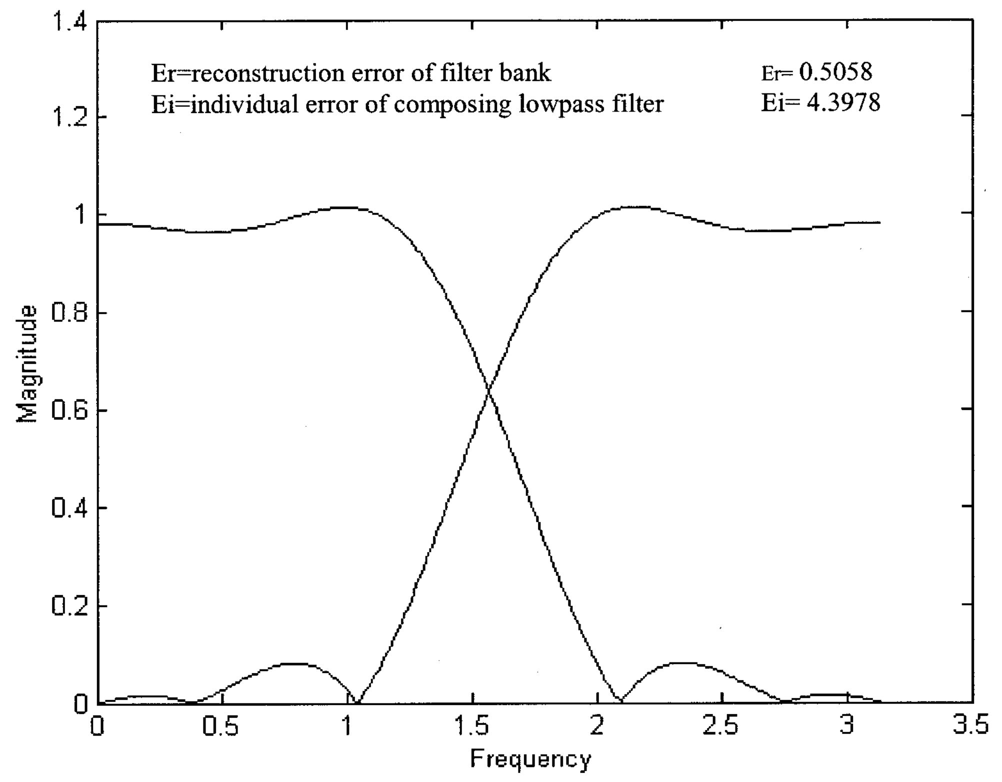
Fig 5.4,5.5,5.6 represents the magnitude responses of the 24 C, 24B and 32 order filter banks. The initial population properties and improvement in the filters performance after few iteration, its reconstruction error and the stopband and passband errors of the individual filter are presented in these figures. The resulting CSD coefficients of the designed order 24 C, 24B and 32-filter bank are presented in tables 5.3,5.4,5.5.



(a) Magnitude response of Initial population



(b) Magnitude response after 50 generations



(c) Magnitude response after 100 generations

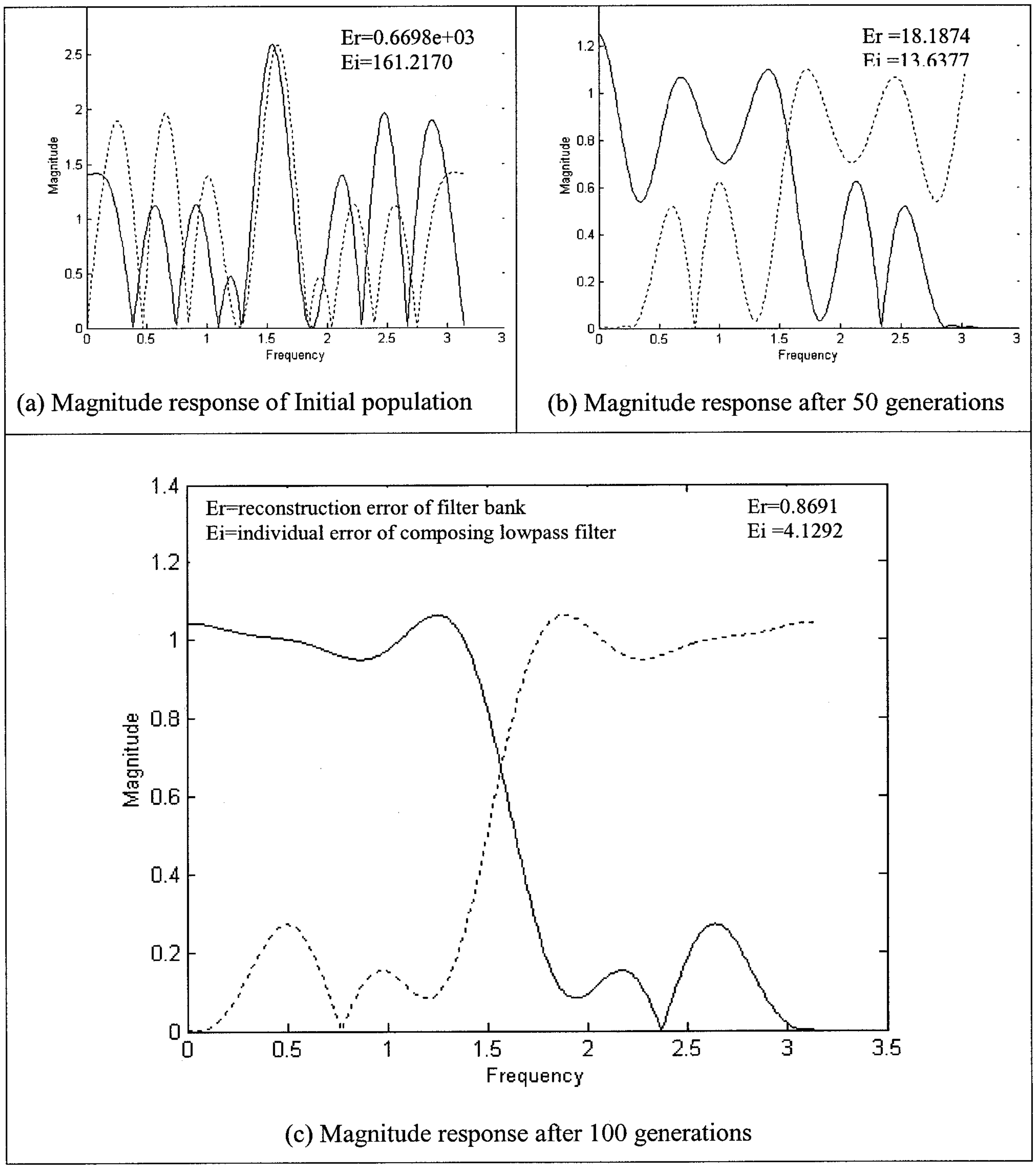
FIGURE 5.3: Performance of GA in designing order 12 QMF bank

	24B	24C	32C
Es	4.454029E-8	2.203661E-5	1.017746E-6
Ep	4.816752E-8	2.778178E-7	5.303133E-8
δs	6.313640E-4	9.922592E-3	2.486957E-3
δp	3.663081E-4	1.105189E-3	5.123051E-4
Δw	1.135977	0.7348368	0.6993391

TABLE 5.2: Performance parameters of Johnstons design

Coefficients	CSD representation
h[0]=h[11]	2^{-15}
h[1]=h[10]	$-2^{-5} -2^{-7} -2^{-15}$
h[2]=h[9]	$2^{-5}-2^{-7} +2^{-9} +2^{-15}$
h[3]=h[8]	$2^{-3}-2^{-5} +2^{-7} +2^{-10}$
h[4]=h[7]	$-2^{-3} +2^{-7} +2^{-10} +2^{-15}$
h[5]=h[6]	$-2^{-1} +2^{-5} +2^{-9} +2^{-15}$

TABLE 5.3:CSD coefficients of 12th order filter bank



(a) Magnitude response of Initial population

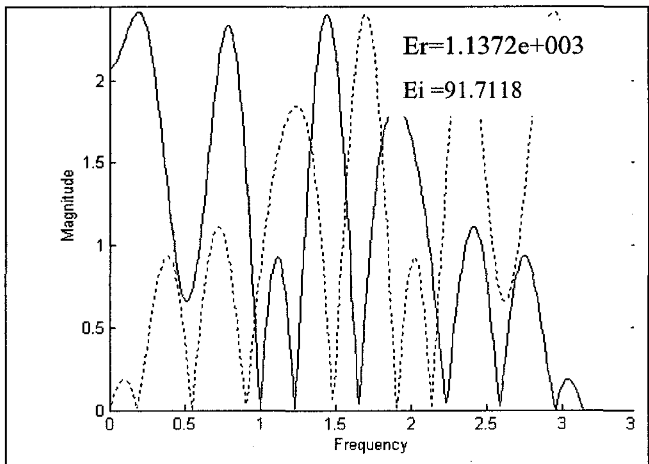
(b) Magnitude response after 50 generations

(c) Magnitude response after 100 generations

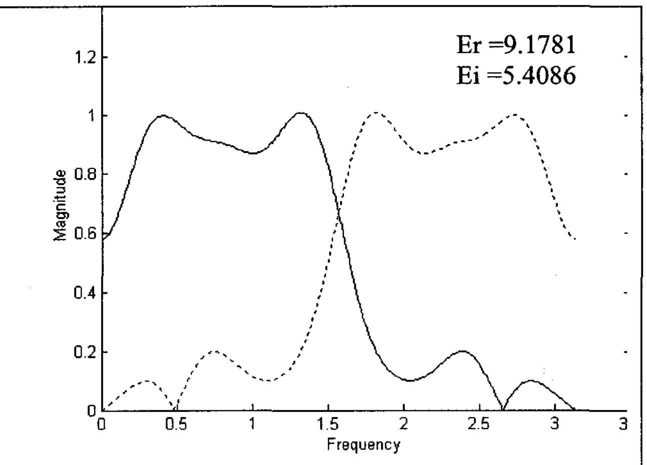
FIGURE 5.4: Performance of GA in designing 24B order QMF bank

Coefficients	N=24B
h(0)	2^{-10} =h(12)
h(1)	2^{-10} =h(13)
h(2)	$-2^{-7}+2^{-10}+2^{-14}$ =h(14)
h(3)	$2^{-5}+2^{-9}$ =h(15)
h(4)	$-2^{-5}+2^{-10}$ =h(16)
h(5)	$-2^{-5}+2^{-8}-2^{-10}+2^{-14}$=h(17)
h(6)	$2^{-5}+2^{-7}+2^{-9}$ =h(18)
h(7)	$2^{-5}-2^{-7}+2^{-10}-2^{-15}$=h(19)
h(8)	2^{-10} =h(20)
h(9)	$-2^{-3}+2^{-7}$ =h(21)
h(10)	$2^{-3}+2^{-7}+2^{-10}+2^{-15}$=h(22)
h(11)	$2^{-1}-2^{-5}+2^{-15}$ =h(23)

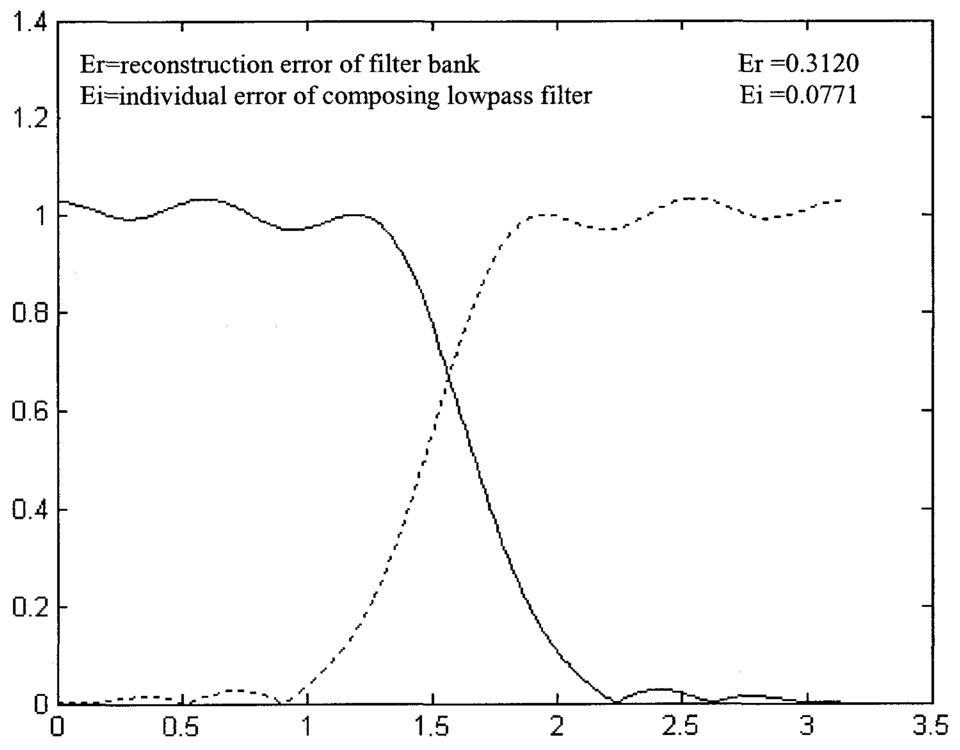
TABLE 5.4:CSD coefficients of 24C order filter bank



(a) Magnitude response of Initial population



(b) Magnitude response after 50 generations

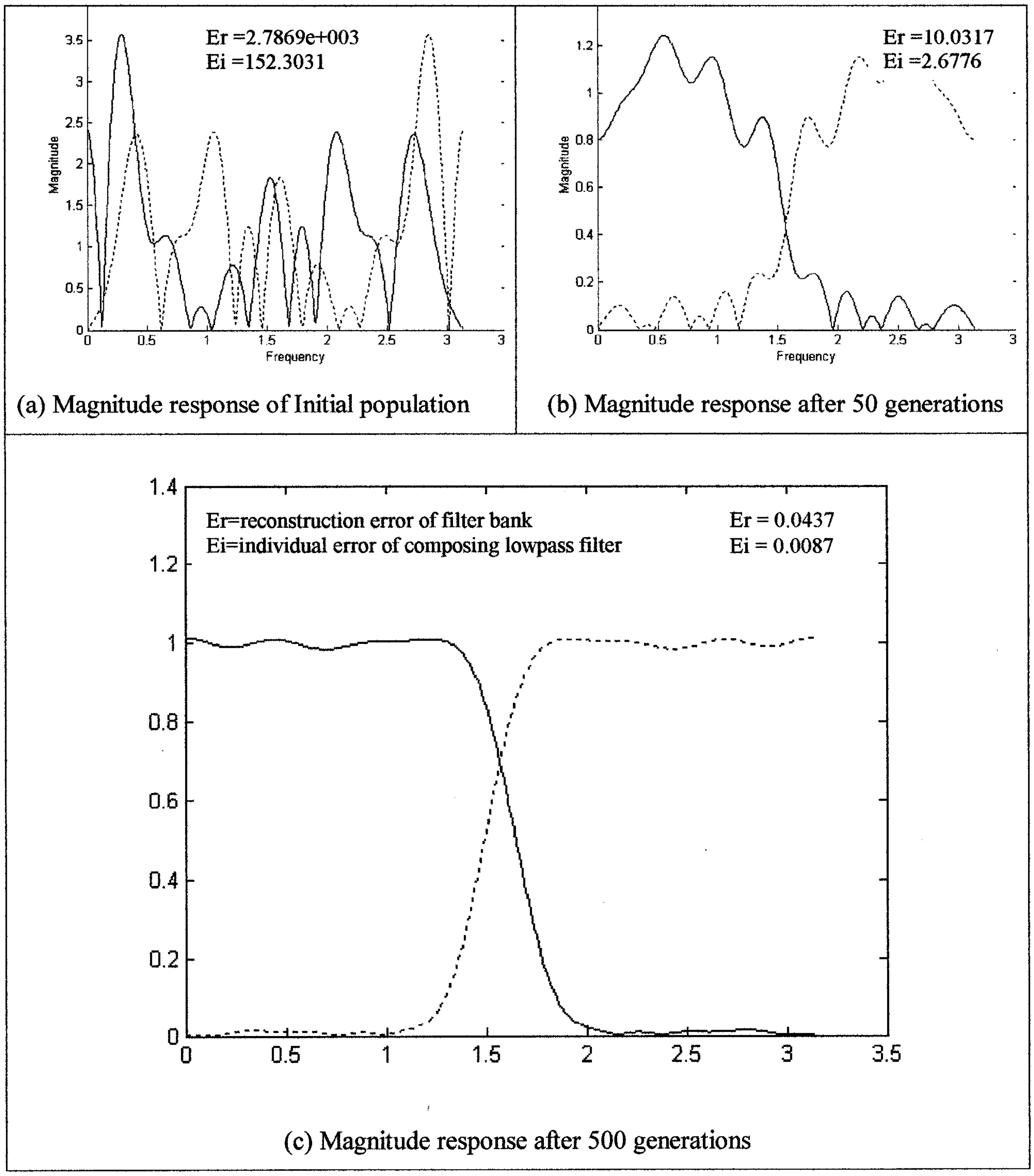


(c) Magnitude response after 100 generations

FIGURE 5.5: Performance of GA in designing 24C order QMF bank

Coefficients	N=24C	
h(0)	2^{-9}	=h(12)
h(1)	2^{-10}	=h(13)
h(2)	$2^{-7}-2^{-10}$	=h(14)
h(3)	$2^{-7}-2^{-9}$	=h(15)
h(4)	$-2^{-7}+2^{-9}$	=h(16)
h(5)	$-2^{-6}+2^{-8}-2^{-10}$	=h(17)
h(6)	2^{-10}	=h(18)
h(7)	$2^{-5}+2^{-7}$	=h(19)
h(8)	$-2^{-5}+2^{-7}+2^{-10}$	=h(20)
h(9)	$-2^{-3}+2^{-5}+2^{-10}$	=h(21)
h(10)	$2^{-3}-2^{-7}+2^{-9}$	=h(22)
h(11)	$2^{-1}-2^{-5}+2^{-7}-2^{-10}$	=h(23)

TABLE 5.5:CSD coefficients of 24C order filter bank



(a) Magnitude response of Initial population

(b) Magnitude response after 50 generations

(c) Magnitude response after 500 generations

FIGURE 5.6: Performance of GA in designing 32 order QMF bank

Coefficients	N=32
h(0)	2^{-9} =h(16)
h(1)	-2^{-10} =h(17)
h(2)	2^{-10} =h(18)
h(3)	$2^{-7}-2^{-9}$ =h(19)
h(4)	2^{-10} =h(20)
h(5)	$-2^{-7}-2^{-10}$ =h(21)
h(6)	2^{-9} =h(22)
h(7)	$2^{-6}-2^{-10}$ =h(23)
h(8)	$-2^{-7}+2^{-9}$ =h(24)
h(9)	$-2^{-5}+2^{-8}-2^{-10}$ =h(25)
h(10)	$2^{-6}-2^{-10}$ =h(26)
h(11)	$2^{-4}-2^{-7}-2^{-10}$ =h(27)
h(12)	$-2^{-5}-2^{-7}$ =h(28)
h(13)	$-2^{-3}+2^{-5}-2^{-8}-2^{-10}$=h(29)
h(14)	$2^{-3}+2^{-10}$ =h(30)
h(15)	$2^{-1}-2^{-5}-2^{-10}$ =h(31)

TABLE 5.6: CSD coefficients of designed 32-order filter bank.

CONCLUSION

In this thesis we have used genetic algorithms to design a FIR QMF bank. The performance of the design depends on the reconstruction error of the overall filter bank and the individual performance of the lowpass filter. Filter bank design is formulated as single objective multiple constraint optimization problem with reconstruction error of the overall filter bank as our main objective and passband error, stopband error, stopband and pass band ripples and transition width of the composing filter as constraints. The coefficients in the designed filter bank are in canonical signed digit format.

A genetic algorithm is used as an optimization technique. In general, genetic algorithms (GAs) rely upon the law of fittest member survival to optimize a set of possible outcomes or results. A new CSD restoration technique is presented to ensure that the algorithm generates CSD coefficients with the specified word length and nonzero digits.

The proposed design of using genetic algorithm for QMF bank design has been demonstrated by 12,24 and 32 order filters banks. The proposed genetic approach to design FIR QMF bank is general and can be extended to design other types of filter banks like IIR filter banks, multi channel filter banks etc with CSD coefficients.

REFERENCES

[1] E. Abdul Raheem, E. I. Guibaly, and A. Antoniou. "Design of low-delay FIR QMF banks using the Lagrange-multiplier approach." Electronics Letters, 30(12): 924-1430, June 1994.

[2] F. Asrafzadeh and B. Nowrouzian, "Crossover and mutation in genetic algorithms employing canonical signed digit number system: proceedings of the 1997 Midwest Symposium on Circuits and Systems, pp.702-705, aug.1997.

[3] C-K. Chen and J. -H. Lee "Design of quadrature mirror filters with linear phase in the frequency domain." IEEE Trans. on Circuits and Systems II, 39(9): 593-605, September 1992.

[4] Doganata, P. P. Vaityanathan, and T. Q. Nguyen. "General synthesis procedures for FIR lossless transfer matrices, for perfect-reconstruction multirate filter bank applications". IEEE Trans. on Acoustics, Speech and Signal Processing, 36(10): 1561-1574, October 1988.

[5] M. Ekanayake and K. Premaratne. "Two-channel IIR. QMF banks with approximately linear phase analysis and synthesis filters." IEEE Trans. on Signal Processing, 43(10): 2313-2322, October 1995.

[6] A. T. G. Fuller, B. Nowrouzian, " Optimization of FIR digital filters over the canonical signed digit coefficient space using genetic algorithms", Proc. of the 1998 Midwest Symposium on Circuits and Systems, pp.456: 459,aug.1998.

[7] R. Gandhi and S. K. Mitra, " A computationally efficient design of two-band QMF banks based on frequency-sampling approach," Proceedings of the 1998 IEEE International Symposium on Circuits and Systems, ISCAS '98 Monterey, CA, USA, 31 May-3 June 1998. Pp.421-424

[8] B. R. Horng and A. N. Willon, Jr. "Larrange multiplier approaches to the design of two-channel perfect-reconstruction linear-phase FIR filter banks." IEEE Trans. on Signal Processing, 40(2): 364-374, February 1992.

[9] V. K. Jain and R. E. Crochiere. "Quadrature mirror filter design in the time domain." IEEE Trans. on Speech and Signal Processing, 32(2): 353-361, April 1984.

[10] J. D. Johnston. "A filter family designed for use in Quadrature filter banks." In Proc.of Int'l Conf. On ASSP, pp. 291-294, 1980.

[11] M. Kawamata, J. Imakubo and T. Higuchi, "Optimal design of 2-D IIR digital filter based on a simple genetic algorithm", Proceedings of IEEE international Conference on image processing,Vol.1,13-16 pp.780-784,Nov.1994.

[12] R. D. Koilpillai and P. P. Vaidyanathan. "Cosine modulated FIR filter banks satisfying perfect reconstruction." IEEE Trans. On Signal Processing, 40(4): 770-783, April 1992.

[13] R. D. Koilpillai and P. P. Vaidyanathan, " A spectral factorization approach to pseudo-QMF design". IEEE Trans. on Signal Processing, 41(1): 82-92, January 1993.

[14] Israel Korean, “Computer arithmetic algorithms”, A K Peters Ltd 2002.

[15] A. Lee, M. Ahmadi, et al. “ Digital filter design using genetic algorithms,” Proc. of 1998 IEEE symp. On Advances in Digital Filtering and Signal Processing, pp. 34-38, June1998.

[16] Ng. S. C. Leung, W. H. Lau, S. H. Chung, C. V. Luk. The genetic approach: a new learning algorithm for adaptive IIR filtering. IEEE Signal Processing Magazine, pp.38-46,Nov 1996.

[17] Y. P. Lin and P. P. Vaidyanathan, “Linear phase cosine modulated maximally decimated filter banks with perfect reconstruction,” IEEE Trans. on Signal Processing, vol. 42, pp.2525-2539, Nov. 1995.

[18] Douglas J. Lockett, Christopher D. Roblee and Michael Rudko, “ Genetic algorithm based design and implementation of multiplier less Two Dimensional Image filters,” Union College CPe-198/199 Capstone Design Project, May 2003.

[19] J. Mc Clellan, “A unified approach to thede sign of optimum FIR linear phase FIR digital filters,” IEEE Trans,vol.CT-20,Nov,1973.

[20] Sanjit. K. Mitra, “ Digital signal processing: a computer based approach. ”Mc Graw Hill. 1998.

[21] S. K. Mitra, C. D. Creusere, and H. Babic. “A novel implementation of perfect reconstruction QMF banks using IIR filters for infinite length

signals." In Proceedings of 1992 Int'l Symposium on Circuit and Systems, pp.2312-2315, 1992.

[22] K. Nayebi, T. P. B. III, and M. J. T. Smith, "Time-domain filter bank analysis: A. new design theory," IEEE on Signal Processing, vol. 40, pp. 1412-1429, June 1992.

[23] K. Nayebi, T. P. Barnwell III, and M. J. T. Smith, "Nonuniform filter banks: A reconstruction and design theory," IEEE Trans. on Signal Processing, vol. 41, pp.1114-1127, Mar.1993.

[24] T. Q. Nguyen, "comparison of linear phase perfect reconstruction in wavelet transform based image compression, "CISS Washington D.C.Mar1995.

[25] T. Q. Nguyen, "Near-perfect reconstruction pseudo-QMF banks," IEEE Trans. on Signal Processing, vol. 42, pp. 65-76, Jan. 1994.

[26] T. Q. Nguyen, T. I. Laakso, and R. D. Koilpillai. "Eigen filter approach for the design of all pass filters approximating a given phase response." IEEE Trans. on Signal Processing, 42(9): 2257-2263, September 1994.

[27] T. Q. Nguyen, T. I. Laakso, and T.E. Tuncer. "On perfect reconstruction all pass based cosine modulated IIR filter banks." In Proc. of 1994 Int'l Symposium on Circuit and systems, pp. 32-36, 1994.

[28] T. Q. Nguyen, P. P. Vaidyanathan. "Two-channel perfect-reconstruction FIR QMF structures which yield linear-phase analysis and synthesis filters." IEEE Trans. on Acoustics, Speech and Signal Processing, 37(5): 676-690, May 1989.

[29] T. Q. Nguyen, P. P. Vaidyanathan, “Structures for m-channel perfect reconstruction FIR QMF structures which yield linear-phase analysis filters”. IEEE Trans. on Acoustics, Speech and Signal Processing, vol. 38 pp. 433-446, Mar 1990.

[30] W. Pennebaker and J. Mitchell, “JPEG still image data compression standard.” Van Nostrand Reinhold, 1993.

[31] J. Princen, “The design of Nonuniform modulated filter banks,” IEEE Trans. on Signal Processing, vol.43, pp.2250-2560.Nov.1995.

[32] J. G. Proakis, “ Digital Signal Processing: Principles, Algorithms, and Applications,” Maxwell Macmillan, 1992.

[33] T. A. Ramstad. “IIR filterbank for subband coding of images.” In Proc. of 1998 Int’l Symposium on Circuit and Systems, pages 827-830,1998.

[34] J. J. Shyu. “Design of two-channel perfect reconstruction linear –phase filter banks for subband image coding by the Lagrange multiplier approach.” IEEE Trans. on Circuits and Systems for Video Technology.1995.

[35] Iraj Sodagar, Kambiz Nayebi, and Thomas P. Barnwell. “Time-varying filter banks and wavelets.” IEEE Trans. on Signal Processing, 42(11): 2983-2996, Nov.1994.

[36] A. K. Soman, P. P. Vaidyanathan, and T. Q. Nguyen. ”Linear phase paraunitary filter banks theory, factorizations and design.” IEEE Trans. on Signal Processing, 41(12), 3480-3946-Dec 1993.

[37] S. Tang, K. F. Man, S .Kwong and Q. He, " Genetic algorithms and their applications ", IEEE signal Processing Magazine,pp.22-37,Nov. 1997.

[38] T. E. Tuncer and T. Q. Nguyen. "General analysis of two band QMF banks." IEEE Trans. on Signal Processing, 43(2), 544-548.Feb 1995

[39] T. E. Tuncer and T. Q. Nguyen. "Interpolated IIR mth-band filter design with allpass sub filters." IEEE Trans. on Signal Processing, 43(8), 1986-1990.Aug 1995

[40] P. P. Vaidyanathan. "Multirate digital filters, filter banks, polyphase networks, and applications." A tutorial. Proc. of the IEEE.78 (1): 56-93,January 1990.

[41] P. P. Vaidyanathan. "Multirate Systems and Filter Banks." Prentice-Hall Inc., 1993.

[42] P. P. Vaidyanathan and T. Chen. "Structure for time reversed inversion in filter banks," In Proc. of 1995 Int'l Symposium on Circuits and Systems, pages 585-588,1995.

[43] P. P. Vaidyanathan and P. Q. Hoang. "Lattice structures for optimal design and robust implementation of two channel perfect reconstruction QMF banks." IEEE Trans. on acoustics, Speech and Signal Processing, 36(1): 81-94, January 1988.

[44] M. Vetterli, " A theory of multirate filter banks." IEEE Trans. on Acoustics, Speech and Signal Processing, 35(3): 356:372 march 1987.

[45] M. Vetterli and D. L. Gall, " Perfect reconstruction FIR filter banks: some properties and factorizations," IEEE Trans. on Acoustics, Speech and Signal Processing, vol.37, pp.1057-1071, July 1989.

[46] Benjamin W.Wah,Yi Shang and Zhe Wu, "Discrete Lagrange Multiplier method for optimizing thed esign of Multiplier less QMF Banks", Proceedings of IEEE Transactions on Circuits and Systems-II:Analog and Digital Signal Processing,vol.46.NO.9,September 1999.

[47] Benjamin W. Wah,Yi Shang and T.Yu, "Global optimization of QMF bank design using Novel,"in proc.IEEE Int,conf.Acoutics ,Speech ,Signal processing. Munich, Germany, Apr 1997,vol.3,pp2081-2084

[48] X. Xu and B. Nowrouzian, "Local search algorithm for the design of multiplierless digital filters with CSD multiplier coefficients", Proc. of the 1999 IEEE Canadian Conference on Electrical and Computer Engineering,.pp.811-816, Alberta, Canada, May 1999.

[49] R. H. Yang and Y. C. Lim, "Novel efficient approach for the design for equiripple Quadrature mirror filters." IEEE Proc on Vision, Image Signal and Processing, 141(2): 95-100.April 1994.

[50] X. Zhang and H. Iwkura, "Design of QMF banks using all pass filters. Electronics Letters," 31(3): 172-174,February 1995.

APPENDIX

Source code files in Matlab

```
% main

%nb is length of CSD number
%pop_size  is the size of the population. Gen is the generation of the genetic
algorithm
%rec_err and ind_err are the reconstruction error of the overall filter bank and
ind_err is %the individual error of the filter.

order=input('order of filter =');
N=order/2;
nb=15;
pop_size=10,000;
Gen=1;

%to generate  random coefficients and convert them to CSD format.
[old_dec,old_csd]=hinitialize(N,nb, pop_size);

% to calculate fitness of the filter bank
[rec_err1,fitness1,ind_err1,t_csd]=hfit(old_csd,N,pop_size,nb);

%plot the results
figure(1);
[ini_rec_err, ini_ind_err,A]=hplot(fitness1,rec_err1,ind_err1,t_csd);

for Gen=1:10
   fprintf(1,'gen in %1.0f \n',Gen);
   datestr(now)
  if  min(rec_err1)>7.4586e-7

%roulletee wheel parent selection
[snew_dec,snew_csd]= hroulette_wheel(t_csd,fitness1,N,nb,pop_size);

%single point crossover
 [xnew_csd,sites] = hxover(snew_csd,nb);

%muatation
```

```
[mnew_csd]=hmutate(xnew_csd,N,nb,pop_size);

%csd restoration
[rm_csd]=hcsd_rest(mnew_csd,N,nb,pop_size);

[rec_err2,fitness2,ind_err2,t_csd2]=hfit(rm_csd,N,pop_size,nb);

%elitist strategy
if max(fitness2)<max(fitness1)
  for i=1:pop_size
     if fitness2(i)==min(fitness2)
          a=i;
    end
     if fitness1(i)==max(fitness1)
          b=i;
    end
 end

   t_csd2(a,:)=t_csd(b,:);
   fitness2(a)=fitness1(b);
   rec_err2(a)=rec_err1(b);
   ind_err2(a)=ind_err1(b);
 end
   t_csd=t_csd2;
   fitness1=fitness2;
  rec_err1=rec_err2;
  ind_err1=ind_err2;

  end
end
figure(2);
[fin_rec_err, fin_ind_err,A]=hplot(fitness1,rec_err1,ind_err1,t_csd2);
[csd]=gen_csd(A,N,nb,1);
ini_rec_err
ini_ind_err
fin_rec_err
fin_ind_err
```

```
% hinitialize.m
% To generate random coefficients and convert them to csd format.

% old_dec is random population,
 %old_csd is csd conversion

function [old_dec,old_csd]=hinitialize(N,nb, pop_size);
old_dec =[];
old_csd=[];
old_dec =randn(pop_size,N);
[old_csd]=gen_csd(old_dec,N,nb,pop_size);
```

```
% gen_csd.m
% CSD  generation
% dec is random population, df is binary conversion, csd is CSD conversion,
% w is no of non zero digits
% nb is length of CSD number
%df is binary conversion,


function [csd]=gen_csd(dec,N,nb,pop_size)

csd=[];
for i=1:pop_size
A=dec(i,1:N);
for i=1:N
   if A(i)>1|A(i)<-1
      A(i)=A(i)-fix(A(i));
   end
end
%convert to binary
df=[];
for i=1:N
 d=A(i);
 da=abs(d);
  t=da;
    for j=1:nb
        t=t*2;
       df(i,j+2)=floor(t);
        t=t-floor(t);
     end
end
for i=1:N
  df(i,nb+1)=df(i,nb);
end

%generate 2's complement to negative numbers
for i=1:N
  d=A(i);
   if d<0
      for j=1:size(df,2)
          df(i,j)=abs(df(i,j)-1);
      end
```

```
        c=1;
         for j=size(df,2):-1:1
            if df(i,j)==1 & c==1
                df(i,j)=0;
                 c=1;
           elseif df(i,j)==0 & c==1
                df(i,j)=1;
                 c=0;

            elseif df(i,j)==1 & c==0
                       df(i,j)=1;
                         c=0;

          else
              df(i,j)=0;
               c=0;
         end
      end
    end
end

%convert binary to csd

r_csd=[];
ci=0;
for i=1:N
for j=nb+2:-1:2

ci1=(ci & df(i,j))|(ci & df(i,j-1))|(df(i,j) & df(i,j-1));
r_csd(i,j)=df(i,j)+ci-2*ci1;
ci=ci1;
end
end
r_csd=r_csd(1:N,2:nb+2);
for i=1:N
w=0;
    for j=1:nb+1

            if  r_csd(i,j) ~= 0
                w=w+1;
             end
              if  w>4
                 r_csd(i,j) =0 ;
              end

         end
end
  r_csd=r_csd(1:N,1:nb+1);
csd =[csd;r_csd];
end
```

```
% fitness.m

% to calculate the fitness of the filter bank
% Reconstruction error (rec_err) , the individual error(ind_err),the sum of
%passband error and stop band error,pass band ripple(pr)stopband
%ripple(sr),transition width(wt) are calculated .
%jstop_err ,jpas_err ,jpr,jsr,jwt are the results of the johnstons design.
%total_err=rec_err+10*(( stop_err- jstop_err)+( pas_err-jpas_err)+( pr-jpr)+(sr-
jsr)+ (wt-jwt)).
%fitness is the reciprocal of total error.

function [rec_err,fitness,ind_err]=hfit(old_csd,N,pop_size,nb)


t=zeros(1,N*pop_size);
for i=1:N*pop_size
   for j=2:nb+1
     t(1,i)=t(1,i)+old_csd(i,j)*2^(-j+1);
   end
  t(1,i)=t(1,i)+old_csd(i,1);
end
a=0;
for k=1:pop_size
   for  i=1:N
     t_csd(k,i)=t(1,a+i);
  end
 a=i*k;
end


t_csd=[t_csd  fliplr(t_csd)];

pr=[];
sr=[];
ws=[];
wp=[];
wt=[];


for i=1:pop_size
   A=t_csd(i,1:2*N);
   wp1=0:0.01:pi/2;
   zp= exp(-sqrt(-1)*wp1);
    hp=abs(polyval(A,zp));
    pr=[pr max(hp)-1];
    ws1=pi/2:0.01:pi;
   zs= exp(-sqrt(-1)*ws1);
    hs=abs(polyval(A,zs));

    sr=[sr max(hs)];

    j=0;
    k=0;
   for i=1:length(wp1)
     if max(hp)==hp(i)&j==0
       wp=[wp wp1(i)];
      j=j+1;
     end
    if max(hs)==hs(i)&k==0
      ws=[ws ws1(i)];
       k=k+1;
    end
  end
end


wt=ws-wp;
rec_err=[];
pas_err=[];
stop_err=[];
 for i=1:pop_size
     A=t_csd(i,1:2*N);
     r_err=0;
     p_err=0;
       s_err=0;
for w=0:0.01:pi
    [h,g]=mag_res(w,A);
     r_err=r_err+(((h^2)+(g^2)-1)^2);
end
    rec_err=[rec_err  r_err];
for w=0:0.01:pi/2
    [h,g]=mag_res(w,A);
     p_err = p_err + (h-1)^2;
end
   pas_err=[pas_err  p_err];
```

```
for w=pi/2:0.01:pi
    [h,g]=mag_res(w,A);
    s_err = s_err + (h)^2;
end
  stop_err=[stop_err  s_err];
end
ind_err=pas_err+stop_err;

order=N*2;

 if order==24
  jstop_err= 2.20366616e-5;
  jpas_err= 2.77817874e-7;
  jpr=9.92259253e-3;
  jsr=1.10518942e-3;
 jwt=7.34836847e-1;
end

fitness=[];
total_err=rec_err+10*(( stop_err- jstop_err)+( pas_err-jpas_err)+ ( pr-jpr)+(sr-
jsr)+ (wt-jwt));

for i=1:length(total_err)
  fitness(i)=1/total_err(i);
end
```

```
% hroulette_wheel.m
% roulette wheel parent Selection

% list=fitness
% returns a randomly selected list of elements
% according to the "probability" values in the
% input list.
% e.g., if you input a list [2 1 3 5 4] then
% the 4th element of that list would have the
% highest chance of selection in the output list.


function [snew_dec,snew_csd]=hroulette_wheel(old_csd,list,N,nb,pop_size)

[m n] = size(list);

if (m~=1)
   error('ERROR! a list must have only one row.');
end

s = sum(list);
for i=1:N*pop_size
    t(1,i)=0;
end

for i=1:N*pop_size
   for j=2:nb+1
     t(1,i)=t(1,i)+old_csd(i,j)*2^(-j+1);
   end
 t(1,i)=t(1,i)+old_csd(i,1);
end
a=0;
for k=1:pop_size
  for  i=1:N
    t_csd(k,i)=t(1,a+i);
  end
 a=i*k;
end


for i=1:n
```

```
    roulette = 0;
    lucky = rand(1) * s;
    got_one = 0;
    j = 0;
    while (j<n) & (got_one==0)
        j = j+1;
        roulette = roulette + list(1,j);
        if (roulette > lucky)
            got_one = j;
        end
    end
    select(1,i)=got_one;
end
snew_dec=t_csd(select,:);
 [snew_csd]=gen_csd(snew_dec,N,nb,pop_size);
```

```
%crossover.m
%single point crossover

% XOVER  Creates a new generation from old generation using crossover.
% [xnew_csd,SITES] = hxover(snew_csd,nb) performs crossover
%  Creation on pairs of snew_csd with probability Pc.
%  Crossover SITES are chosen at random (re: there will be
%  half as many SITES as there are individuals.

function [xnew_csd,sites] = hxover(snew_csd,nb);

Pc=0.7;
lchrom = size(snew_csd,2);
sites = ceil(rand(size(snew_csd,1)/2,1)*(lchrom-1));
sites = sites.*(rand(size(sites))<Pc);
x=nb+1;

  for i = 1:length(sites);
  xnew_csd([2*i-1 2*i],:) = [snew_csd([2*i-1 2*i],1:sites(i)) ...
                      snew_csd([2*i 2*i-1],sites(i)+1:lchrom)];
end
```

```
% hmutate.m
% mutation

%changes the randomly selected bit from 0 to1 ,1 and -1 to 0
%pm is the mutation probability .

function [mnew_dec,mnew_csd]=hmutate(xnew_csd,N,nb,pop_size)

Pm=0.001;
mutated = find(rand(size(xnew_csd))<Pm);
mnew_csd = xnew_csd;
mnew_csd(mutated) = 1-abs(xnew_csd(mutated));

%togenerate the decimals

for i=1:N*pop_size
t(1,i)=0;
end

for i=1:N*pop_size
for j=2:nb+1
t(1,i)=t(1,i)+mnew_csd(i,j)*2^(-j+1);

end
t(1,i)=t(1,i)+mnew_csd(i,1);

end
a=0;
for k=1:pop_size
for  i=1:N
mnew_dec(k,i)=t(1,a+i);
end
a=i*k;
end
```

```
% hcsd_rest.m
% CSD restoration technique

% after crossover and mutation the CSD coefficients are checked for any
% Violation in its CSD format, if so they are restored back to its CSD format

function [rm_dec,rm_csd]=hcsd_rest(mnew_dec,N,nb,pop_size)

rm_dec =[];
rm_csd=[];

 [rm_csd]=gen_csd(mnew_dec,N,nb,pop_size);



t=zeros(1,N*pop_size);

for i=1:N*pop_size
for j=2:nb+1
t(1,i)=t(1,i)+rm_csd(i,j)*2^(-j+1);
end
t(1,i)=t(1,i)+rm_csd(i,1);
end
a=0;
for k=1:pop_size
for  i=1:N
rm_dec(k,i)=t(1,a+i);
end
a=i*k;
end
```

```
%hplot.m
%plot the results

function [fin_rec_err, fin_ind_err,A]=hplot(fitness,rec_err,ind_err,dec);

j=0;
for i=1:length(fitness)
if max(fitness)==fitness(i)&j==0
a=i;
j=j+1;
end
end
A=dec(a,:);
A=[A fliplr(A)];
w=0:0.02:pi;
[h,g]=mag_res(w,A);
plot(w,h,'-',w,g,':');
fin_rec_err=rec_err(a);
fin_ind_err=ind_err(a);
```

VITA ACTOURIS

Haritha Uppalapati was born in India on June 11th 1979.She completed her secondary studies in India on 1994.In 2000 she graduated bachelors in Electronics and Telecommunication Engineering from University of Gulbarga, India. She came to Canada as permanent resident in November, 2001.In 2002 she was accepted for masters program by department of Electrical Engineering of University of Windsor