University of Alberta

DESIGN AND DISCRETE OPTIMIZATION OF BIBO STABLE FRM DIGITAL FILTERS INCORPORATING IIR DIGITAL INTERPOLATION SUBFILTERS

by

Syed Baqir Bokhari

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Masters of Science
in
Communications

Electrical and Computer Engineering

Spring 2010
Edmonton, Alberta

Examining Committee

Dr. Behrouz Nowrouzian, Electrical and Computer Engineering

Dr. Venkata Dinavahi, Electrical and Computer Engineering

Dr. Ray Nilanjan, Computer Sciences

“Do or do not. There is no try.”
-Master Yoda

“I want to know God’s thoughts. The rest are details.”
-Einstien

Abstract

Digital filters having sharp transition band play a vital role in modern digital signal processing (DSP) applications. Emerging technologies require digital filters to be both computationally efficient in software/hardware realizations. This thesis is concerned with the design and structural-level optimization of sharp transition band digital filters employing the well known frequency response masking (FRM) approach. Unlike the conventional finite impulse response (FIR) based FRM approach, the FRM technique used in this thesis incorporates infinite impulse response (IIR) digital interpolation subfilters, thereby reducing the overall filter order that results in a reduction of hardware complexity. Two realization methods are discussed in this thesis, namely, the bilinear-lossless-discrete-integrators (bilinear-LDI) digital filter design technique, and the lattice wave digital filter (lattice WDF) digital filter design technique.

Diversity controlled (DC) genetic algorithm (GA) is employed to optimize both types of IIR based FRM digital filters over the efficient canonical signed digit (CSD) multiplier coefficient space. DCGAs represent FRM digital filters by a binary chromosome and proceed from a population pool of candidate chromosomes to future generations in order to arrive at the desired FRM digital filter satisfying the design specifications. A novel cost-function is used that allows the DCGA to simultaneously optimize both the amplitude-frequency and group-delay frequency response. A fast convergence speed has been observed.

Acknowledgements

The generous support and encouragement offered by Dr. Behrouz Nowrouzian has been most crucial in the successful completion of this thesis. I am deeply thankful to Dr. Nowrouzian for his patient advice and criticism during the period of my M.Sc studies.

I would like to thank all my candidacy committee and defence committee members for their valuable feedback. Through these years, they have continuously provided me with many useful guidelines and constructive comments for my research work as well as my doctoral life.

Last but not the least, I would like to thank my parents for their support and love over my life. Without them, I would never have completed my studies successfully.

Table of Contents

List of Figures

List of Tables

List of Acronyms

Acronyms	**Definition**
BIBO	Bounded-Input Bounded-Output
CPSS	Cross Generational Probability Survival Selection
CSD	Canonical Signed-Digit
DCGA	Diversity Controlled Genetic Algorithm
DSP	Digital Signal Processing
EMQF	Elliptic Minimal Q-Factor
FIR	Finite Impulse Response
FRM	Frequency Response Masking
GA	Genetic Algorithm
IF	Intermediate Frequency
IIR	Infinite Impulse Response
LDI	Lossless Discrete Integrator
LUT	Look-up Table
SA	Simulated Annealing
SPT	Signed Power of Two
WDF	Wave Digital Filter

List of Symbols

Symbols	**Definition**
α	DCGA exponent coefficient
α_i	Infinite-precision lattice WDF multiplier coefficient
$\hat{\alpha}_i$	Finite-precision lattice WDF multiplier coefficient
ϵ	Finite precision slack variable
ε_p	Maximum error in the passband
ε_a	Maximum error in the stopband
ς_p	Maximum group-delay distortion
B_ρ	Wordlength of a gene in a chromosome
β_i	Infinite-precision lattice WDF multiplier coefficient
$\hat{\beta}_i$	Finite-precision lattice WDF multiplier coefficient
c	DCGA shape coefficient
δ	Transition bandwidth of the interpolation digital subfilter
δ_p	Squared passband ripple tolerance for an EMQF filter
δ_a	Squared stopband ripple tolerance for an EMQF filter
$EvP(s)$	Even order part of polynomial $P(s)$
f_{3dB}	Frequency at which the digital filter has an attenuation of 3 dB
$fitness$	Fitness value of a chromosome
$fitness_{group-delay}$	Fitness based on group-delay frequency response
$fitness_{magnitude}$	Fitness based on magnitude-frequency response
$F(z)$	Overall FRM filter incorporating only FIR digital subfilters
$F_a(z)$	FIR digital interpolation subfilter
$F_b(z)$	FIR complementary digital interpolation subfilter

$F_{0\text{or}1}(z)$	FIR masking digital subfilter
$G_0(z)$	Odd-order allpass digital network
$G_1(z)$	Even-order allpass digital network
$H(z)$	Overall FRM filter incorporating IIR digital subfilters
$H_a(z)$	IIR digital interpolation subfilter
$H_a(s)$	IIR analog reference subfilter
$H_a(z^M)$	M interpolated IIR digital subfilter
$H_b(z)$	IIR complementary digital interpolation subfilter
$H_b(z^M)$	M interpolated IIR complementary digital subfilter
h	Hamming distance between two chromosomes
K	Number of image lobes allowed through by the masking digital subfilters
k	Ratio of analog passband and stopband edge frequencies
L	Chromosome length
M	Interpolation factor
m_{Cp}/m_{Lp}	Infinite-precision bilinear-LDI multiplier coefficients
$\hat{m}_{Cp}/\hat{m}_{Lp}$	Finite-precision bilinear-LDI multiplier coefficients
N	Population pool size
N_{FIR}	Order of FIR interpolation digital subfilter $F_a(z)$
N_{IIR}	Order of IIR interpolation digital subfilter $H_a(z)$
N_{mating}	Number of chromosomes in mating pool
$\tilde{n}$	Order of a digital allpass network $G_{0\text{or}1}(z)$
$OdP(s)$	Odd order part of polynomial $P(s)$
Ω_A	Analog frequency
ω_d	Digital frequency
ω_p	Passband edge frequency of the FRM digital filter
ω_a	Stopband edge frequency of the FRM digital filter
ϕ	Stopband edge frequency of the interpolation digital subfilter
$P(s)$	Hurwitz polynomial
$P(t)$	Current population pool
$P(t+1)$	Next generation population pool

$\hat{P}(t)$	Enlarged population pool
p_F	Fixed probability of complementing bits in the chromosome
p_M	Fixed probability of mutation
p_s	Probability of selecting a chromosome
R	Radix position
R_p	Passband ripple in dB
R_a	Stopband ripple in dB
s	Continuous time complex frequency variable
θ	Passband edge frequency of the interpolation digital subfilter
T	Sampling period
t	Generation number
W	CSD multiplier coefficient wordlength
w	Maximum number of non-zero digits in CSD multiplier coefficients
W_p	Passband weighting factor
W_a	Stopband weighting factor
W_{gd}	Group-delay weighting factor
$Y(s)$	Realizable admittance
$Z(s)$	Realizable reactance
z	Discrete time complex frequency variable

Chapter 1

Introduction

Digital filters play a central role in modern digital signal processing (DSP) systems. With the emerging DSP technologies and applications, there is an ever growing demand for digital filter structures that lend themselves to efficient hardware implementations. In practical applications, e.g. in audio signal processing, image processing or data compression, digital filters with sharp transition bands are required. Anti-aliasing filters used in digital photography are a commonplace example of where such sharp transition band digital filters are required. Anti-aliasing filter remove signal components that have a higher frequency than is able to be properly resolved by a digital recording device before the sampling phase. If this is not done it causes undesirable artifacts to appear in the digitally reproduced image.

An ideal anti-aliasing filter could be considered as a “brick-wall” filter in that it perfectly leaves all desired low frequencies intact and completely cuts off all undesired high frequencies. Such ideal filters cannot be realized in practise, and so they are approximated using filters having a small finite transition bandwidth. However, since the order of digital filters is inversely proportional to their transition bandwidth, narrow transition bandwidth digital filters become less economical if designed by using the conventional techniques.

Frequency-response masking (FRM) approach has been developed [1] for the design of practical sharp transition band digital filters in terms of low-order finite impulse response (FIR) interpolation subfilters and FIR masking subfilters. The interpolation subfilters have gradual transition bands, but are interpolated and arranged in such a manner as to realize very narrow transition bandwidths in the overall FRM FIR digital filter. The resulting FIR interpolated digital subfilters have an inherently large number of zero-valued multiplier coefficients, leading to a substantial reduction in the computational complexity of the resulting digital filter. However, even the conventional FRM digital filter design approach may require FIR interpolation subfilters with prohibitively high orders.

It is well known that infinite impulse response (IIR) digital filters generally possess orders which are substantially lower than those of the corresponding FIR digital filters satisfying similar design specifications. Therefore, IIR digital filters prove to be attractive alternative candidates for interpolation subfilters in the FRM digital filter design approach [2]. However, the resulting IIR interpolation digital subfilters may suffer from bounded-input bounded-output (BIBO) stability problems, in addition to inevitably giving rise to group-delay distortions.

This thesis is concerned with the development of novel techniques for the design and rapid optimization of FRM digital filters employing hardware efficient IIR interpolation subfilters. In these optimization techniques, the corresponding design specifications concern both the magnitude-frequency as well as the group-delay frequency response characteristics of the resulting FRM digital filters.

From a practical point of view, particularly in the hardware implementation of the FRM digital filters, there is every incentive to represent the constituent multiplier coefficients in computationally efficient number systems, e.g. in SPT or canonical signed-digit (CSD) number systems, while still satisfying all the other design specifications. This is mainly due to the fact that such number systems permit the representation of the constituent multiplier coefficient values in terms of only a few non-zero digits within the coefficient wordlength, thereby reducing the corresponding multiplication operations to few shift-and-add operations.

Of course, the above optimization problems tend to have multimodal cost functions, calling for built-in internal or external mechanisms for escaping from local optimal solutions in the course of optimization. It is well known that genetic algorithms (GAs) provide a promising approach to solve discrete and multimodal optimization problems due to the fact that they are capable of automatically finding near-optimal solutions while keeping the computational complexity of the optimization at moderate levels. Consequently, they have emerged as an efficient alternative for the optimization of FIR as well as IIR digital filters. These algorithms encode the digital filter realization into a chromosome, and proceed toward an optimal solution through the evolution of a population of potential candidate chromosomes in an iterative manner from one generation to the next.

However, the conventional GAs do not search the solution space robustly due to lack of mechanisms through which entrapment at local optimal solutions can be avoided. It was demonstrated in [3] that diversity control can be exploited to help to increase the convergence speed of conventional GAs. The main idea behind diversity controlled (DC) GA is to increase the diversity of the population pool through the incorporation of additional non-elite chromosomes based on a pair of external control parameters.

Unfortunately, a direct application of DCGA to the optimization of the above FRM digital filters over the CSD multiplier coefficient space gives rise to two different problems:

- The first problem arises because of the fact that the operations of crossover and mutation in the course of DCGA optimization may lead to chromosomes that may no longer conform to the CSD number format.
- The second problem, on the other hand, stems from the fact that DCGA optimization may result in a solution chromosome that satisfies the given magnitude and group-delay frequency-response design specifications, but that the corresponding IIR interpolation digital subfilter is not BIBO stable.

In this way, DCGA lacks inherent mechanisms to ensure that the optimized digital FRM filters conform to CSD number format, and that their IIR interpolation subfilters are BIBO stable.

In [4], the former problem was resolved in the context of the DCGA optimization of FRM digital filters having FIR interpolation digital subfilters. This was facilitated by generating an indexed look-up table (LUT) of permissible CSD multiplier coefficient values (with pre-specified wordlengths and pre-specified maximum number of non-zero digits), and by employing the indices of the resulting multiplier coefficient values (as opposed to the values themselves) to represent FRM digital filter chromosomes. The key point in generating the CSD LUT is to ensure that the constituent indices form a closed set under the operations of crossover and mutation (or other similar operations) in the course of the underlying DCGA optimization so as to preserve adherence to the CSD number format.

In this thesis, the latter problem mentioned above is resolved by resorting to the use of a set of novel CSD LUTs. These LUTs are constructed to consist of such CSD multiplier coefficient values which automatically (i.e., without the need to check the stability) lead to FRM digital filter chromosomes whose IIR interpolation subfilters remain BIBO stable under the operations of crossover and mutation throughout the course of DCGA optimization. In this way, the resulting FRM digital filter chromosomes not only conform to the CSD number system, but is also BIBO stable (without ever making recourse to the process of gene repair).

The proposed DCGA optimization is first applied to a seed FRM digital filter consisting of infinite-wordlength multiplier coefficient values. The corresponding infinite-wordlength IIR interpolation subfilters are realized in terms of bilinear-lossless-discrete-integrator (bilinear-LDI) allpass digital networks [5]. The main advantage of the bilinear-LDI design approach is that it leads to digital filters which exhibits exceptionally low passband sensitivity to their multiplier coefficient values (permitting reduced multiplier coefficient wordlengths), while being minimal in the number of digital multipliers and digital unit-delays (and practically minimal in the number of digital adders as well).

The investigations are restricted only to elliptic function digital filters. It is well known that an elliptic IIR filter can achieve a sharper transition band between the passband and stopband edges than any other filter with the same number of multiplier coefficients.

For the implementation of an IIR elliptic filter, different realizations can be used including a direct, cascade, parallel, or ladder connection of the two allpass networks. The given transfer function is realized with a minimal number of multipliers (equal to the filter order) if the parallel connection of two allpass networks is chosen [6]. Another advantage of this realization is that the allpass characteristic is insensitive to the coefficient wordlength (thus allowing fewer bits necessary to represent the multiplier coefficients).

The infinite-precision multiplier coefficient values are subsequently quantized to their closest CSD counterparts (to within the pre-specified wordlength and the pre-specified maximum number of non-zero digits), in order to obtain a corresponding FRM digital filter consisting of finite-precision multiplier coefficient values. The resulting CSD seed FRM digital filter is subsequently used to form an initial population pool of CSD FRM digital filter chromosomes for DCGA optimization. A novel cost-function is employed in the DCGA optimization process so as to simultaneously optimize the magnitude-frequency as well of the group-delay frequency response of the desired FRM digital filters. The group-delay frequency response is calculated using the adjoint network method [7].

A second application of DCGA consists of optimizing a seed FRM digital filter consisting of infinite-wordlength IIR interpolation subfilters realized utilizing lattice wave digital filter (WDF) allpass networks. The overall FRM filter still consists of IIR interpolation subfilters and FIR masking subfilters. The lattice WDFs have low sensitivity to multiplier coefficient accuracy. Also, it is minimal in the number of digital multipliers (equal to the filter order).

Once again, only elliptic function digital filters are considered. In addition to the sharp transition bands, elliptic filters also provide the possibility of realizing a transfer function with all its z-plane poles located on a circle in the digital frequency domain [8]. Subsequently, the obtained locations of the poles is used to adjust the values of half of the multiplication coefficients in the lattice WDF design. These coefficients can be adjusted such that they no longer need to be quantized in order to be represented in finite-wordlength. This eliminates half of the most sensitive multiplier coefficients as optimization variables.

Similar to the case of bilinear-LDI, a parallel addition and subtraction of allpass networks is used to realize the IIR interpolation digital subfilters. The allpass networks are further decomposed into first and second order allpass sections [9]. These can be implemented using known techniques that have low noise and low limit cycle oscillations [10].

The infinite-precision multiplier coefficient values are subsequently quantized to their closest CSD counterparts, in order to obtain a corresponding FRM digital filter consisting of finite-precision multiplier coefficient values. The resulting CSD seed FRM digital filter is subsequently used to form an initial population pool of CSD

FRM digital filter chromosomes for DCGA optimization. The cost-function allows for simultaneous optimization of the magnitude-frequency as well of the group-delay frequency response of the desired FRM digital filters.

This thesis is organized as follows. Chapter 2 provides background information on different types of analog and digital filters. It briefly compares the salient features of the bilinear-LDI and lattice WDF digital filter design technique. Additionally, it provides an overview of conventional FRM technique using only FIR digital filters for both the interpolation and masking digital subfilters. Finally, it also defines the CSD number system and works out its main characteristics in terms of the worst case quantization as a function of wordlength and non-zero coefficients.

Chapter 3 is concerned with the design and DCGA optimization of FRM digital filters incorporating elliptic IIR digital interpolation subfilters. The digital interpolation subfilters have a topology consisting of parallel branches that add and subtract two allpass networks in order to produce the required interpolation and complementary interpolation filters. The allpass networks are realized through the use of the bilinear-LDI design technique. The required constraints that guarantee BIBO stability of the bilinear-LDI allpass digital networks (and thus the overall FRM digital filter) are derived. A novel LUT-based scheme is developed to ensure that the interpolation digital subfilters remain BIBO stable throughout the course of DCGA optimization. A detailed design methodology is also presented.

Chapter 4 is concerned with the design and DCGA optimization of FRM digital filters incorporating elliptic IIR digital interpolation subfilters, and the overall digital filter topology is similar to that described in Chapter 3 with the exception that the allpass networks are realized through the use of the lattice WDF design technique. Digital filter realization using cascaded first and second order digital allpass sections is presented. The theory used to fix half the multiplier coefficients of the interpolation digital subfilters to an constant, easily implementable value (thereby avoiding the need to quantize them) is detailed.

Chapter 5 illustrates the use of DCGA through its application on lowpass FRM digital filters. Both bilinear-LDI and lattice WDF techniques are illustrated and compared.

Finally, Chapter 6 summarizes the entire thesis and draws a conclusion. The authors contributions in this thesis are highlighted and suggestions for future work are made.

The remainder of this chapter proceeds as follows. Section 1.1 gives an overview of the different types of optimization techniques available. Section 1.2 provides a brief overview of the conventional GA and contains a step-by-step summary of the procedure. Section 1.3 describes the DCGA optimization technique and highlights its salient features and the differences between DCGA and GAs. Finally, Section 1.4 provides a brief summary of this chapter.

1.1 Overview of Optimization Techniques

Broadly speaking, there are two main approaches available for the optimization of digital filters. One approach is based on the conventional continuous optimization techniques, while the other is based on heuristic discrete evolutionary optimization techniques [11].

Continuous optimization techniques are usually greedy (i.e. they make use of gradient information). These techniques are used to optimize applications with continuous (infinite-precision) design variables, and they utilize the gradient information of the objective function for guiding the direction of search. Examples of continuous optimization techniques include the Newton's method and the Parks-McClellon algorithm [12]. While continuous optimization techniques have the advantage of being able to directly find the optimal solution (search direction is deterministic in nature), they also have the following drawbacks:

- The final implementable digital filter is finite-precision. Continuous optimization techniques are not directly applicable unless one uses such techniques as branch-and-bound (computationally extensive).

- In some cases, it may be difficult to obtain gradient information accurately. This may be the case when the gradient is not analytically defined or the optimization variables are discrete.

- Continuous optimization techniques do not have reliable techniques to escape local optima of multimodal objective functions.

Discrete optimization techniques, on the other hand, may or may not require gradient information. These techniques can be successfully employed over discrete domain (i.e. when the optimization parameters are discrete variables). The most commonly used discrete optimization techniques are the GA and simulated annealing (SA) [13]. While the GA and SA are probabilistic in nature, they do not require gradient information to obtain near global optimums, and are well known to be effective in solving complex multivariate, multimodal optimization problems.

In [14] a non-linear programming approach was developed for the optimization of FIR based FRM digital filters over the signed-power-of-two (SPT) multiplier coefficients space. In [15], GAs were applied to the same FIR based FRM digital filter optimization problem, leading to FRM digital filters with superior magnitude frequency-response characteristics. In addition, it was shown that the use of GAs overcomes the inherent drawback of the commonly used linear programming techniques which require the separate optimization of the constituent FRM digital subfilters. In [16], GAs were applied to the optimization of the magnitude frequency-response of an intermediate-frequency (IF) digital filters over the CSD multiplier coefficient space, where the IF digital filter chromosome was formed by the concatenation of the ternary representation of the constituent multiplier coefficient values.

GAs are evolutionary optimization algorithms that take inspiration from natural selection and survival of the fittest in the biological world. GAs perform a parallel search by forming a pool of candidate solution called chromosomes. As the GA progresses from one generation to the next, new chromosomes are formed based on well performing current chromosomes, and only the best performing chromosomes survive. In this way, several regions of the search space are explored simultaneously, thus allowing the GA to potentially attain a global optimum solution.

SA is another evolutionary optimization technique in which the search is made by random moves from a starting point. If a move leads to a better result, it is accepted (upward move). However, if it leads to a worse result (downward move), it is only accepted with a given probability. This probability is high at first, resulting in regular downward moves, but is gradually reduced so as to reach an optimum solution. The reason for accepting downward moves is that they provide a mechanism for escaping local optimal points. However, too many negative moves lead the optimization away from the optimal solution [17].

GAs and SAs have a parallel and linear search nature, respectively. GAs are therefore effective in exploring the search space more comprehensively, whereas SAs are normally used in situations where the application is so complex that a parallel search is beyond reasonable limits of computational complexity.

1.2 Conventional GA

GA was introduced by Goldberg [18] in the 1970s and became a popular heuristic technique for the optimization of complex non-linear systems. GA is an optimization technique that simulates natural selection and reproduction to move towards an optimal solution, and closely resembles the concept of biological evolution. In general, GAs consist of the following steps:

- Initialization: A seed chromosome is formed by concatenating the design variables represented in their binary form. The initial value of these design variables is specific to the application being optimized. Subsequently, an initial population pool consisting of N chromosomes is generated by randomly complementing bits of the seed chromosome [19].

- Evaluation: Each chromosome in the current population pool $P(t)$, where t represents the generation number (initially $t = 0$), is evaluated based on an application specific cost-function (also called the fitness function) that ascertains the degree to which the design specifications are satisfied. Each chromosome is given a single *fitness* value, even though the fitness function itself may be based on one or more performance criteria. Subsequently, the chromosomes in the pool are ranked based on the value of their fitness [19].

- Generation of mating pool: In order to produce the next generation population pool $P(t+1)$, a mating pool is constructed by selecting N_{mating} ($N_{mating} < N \& N_{mating}$ is even) chromosomes from the current population pool $P(t)$. If the best fit chromosome from $P(t)$ is *guaranteed* inclusion in N_{mating}, then the GA is described as elitist. If there is a chance that the best fit chromosome from $P(t)$ may not be selected then the GA is non-elitist. The selection of N_{mating} is based on some stochastic function (e.g. Bernoulli distribution [4]) that usually have a greater tendency to select chromosomes with a high fitness value. However, it is necessary to include a few chromosomes with low fitness values to increase diversity in the mating pool. In this way, the offspring of a low fitness chromosome and a high fitness one may outperform its parents.

- Parent selection: $N_{mating}/2$ pairs of parents are selected from the mating pool to generate offspring by using the conventional roulette wheel selection or the correlative roulette wheel selection method [20]. Correlative roulette wheel selection ensures parent chromosomes pairs have sufficient hamming distance between them. This results in a more diverse offspring.

- Formation of next generation population pool: The next generation population pool $P(t+1)$ is selected from the current population pool $P(t)$ after it has been augmented with chromosomes produced through genetic operations simulating reproduction. Many variations of reproduction operations exist but a typical approach is as follows:

  - Crossover operation: An example of a typical two-point crossover operation is as shown in Fig. 1.1. The $N_{mating}/2$ parent chromosome pairs in the mating pool undergo a two-point crossover operation, producing two offspring per parent pair. The crossover points are chosen randomly for every parent pair. The resulting N_{mating} chromosomes are added to the original population pool $P(t)$ to form an enlarged population pool $\hat{P}(t)$ consisting of $N + N_{mating}$ chromosomes. The enlarged population pool $\hat{P}(t)$ then undergoes the mutation operation.

  - Mutation operation: An example of a typical mutation operation is shown in Fig. 1.2. The enlarged population pool $\hat{P}(t)$ undergo the mutation operation. Mutation involves a very small probability $p_{_M}$ that one or more arbitrarily chosen bit in a random chromosome will be flipped (either from '0' to '1' , or from '1' to '0'). In some cases, the best fit chromosome in $P(t)$ is excluded from the possibility of mutation.

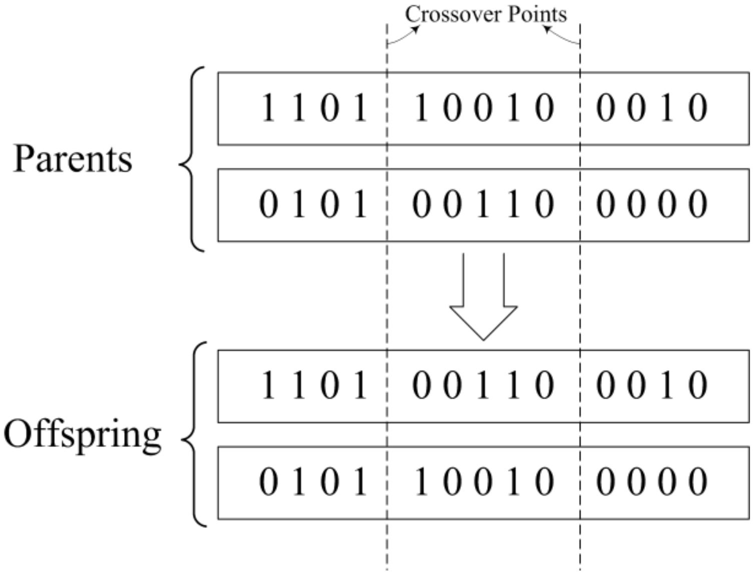

Figure 1.1: Example of a Two-Point Crossover

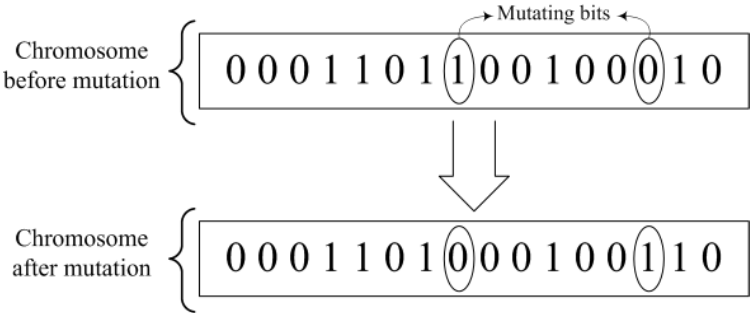

Figure 1.2: Example of Mutation

The crossover operation generates new chromosomes which often share many of the desirable characteristics of their parents, a combination of which could result in higher fitness results. While crossover produces new chromosomes, it does so within the solution space of the current population $P(t)$[1]. Mutations allow the GA to explore entirely new search areas and increase the diversity of the population pool.

Once the genetic operations have been applied to create the enlarged population pool $\hat{P}(t)$, the newly made chromosomes are evaluated (similar to the above step of Evaluation) and all the chromosomes are once again ranked based

[1] Note however that there is a possibility of mutation at the crossover points.

on the value of their fitness. The next generation population pool $P(t+1)$ is selected from the best performing chromosomes of $\hat{P}(t)$. Consequently, the next generation population pool is expected to have an average fitness value greater than the current generation since only the best solutions from the current generation are selected for reproduction (along with a small proportion of less fit members).

- Termination: The above steps are repeated from one generation of the population pool to the next generation, with incremental improvements as the algorithm progresses, until a pre-specified termination condition is met. This termination condition may be based on one or more performance criteria, and it usually ensures that all design specifications are met. The algorithm is also terminated if the pre-set maximum number of allowed iterations is reached without the GA converging to a satisfactory solution [21].

The above steps can be arranged in the flowchart shown in Fig. 1.3.

The salient features of the GA described above are as follows:

- GAs can perform a parallel search using a population pool of potential candidate chromosomes, making them effective at finding optimum solutions to complex multimodal optimization problems. The parallel nature of GA allows it to rapidly narrow down the search space to the region having the global optimum, and this is the primary reason why it is often preferred over SA in applications involving large search spaces.

- GAs do not require any gradient information to perform the optimization. This is useful in highly non-linear cases and also if the number of design variables is very large (as is the case with digital filter optimization).

Unfortunately, GAs also suffer from well known issues concerning the speed of convergence. Even though GAs are parallel in nature, it is possible that all candidate solutions belong to a small region of the search space that does not contain the global optimum. This is because a few chromosomes with a comparatively high fitness value (but not globally optimal) may rapidly dominate the population pool. As a result, there will be a rapid decline in diversity of the population pool

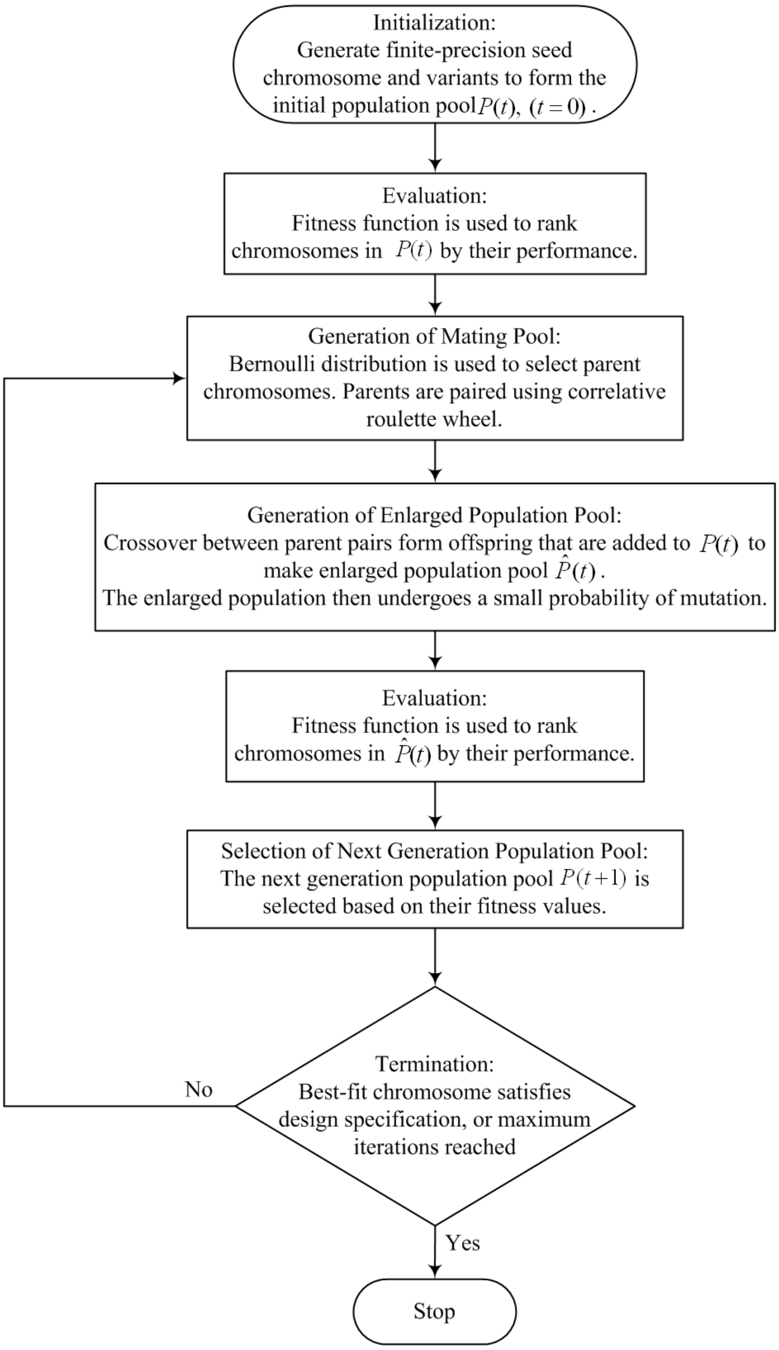
Initialization:
Generate finite-precision seed chromosome and variants to form the initial population pool $P(t)$, $(t=0)$.
Evaluation:
Fitness function is used to rank chromosomes in $P(t)$ by their performance.
Generation of Mating Pool:
Bernoulli distribution is used to select parent chromosomes. Parents are paired using correlative roulette wheel.
Generation of Enlarged Population Pool:
Crossover between parent pairs form offspring that are added to $P(t)$ to make enlarged population pool $\hat{P}(t)$.
The enlarged population then undergoes a small probability of mutation.
Evaluation:
Fitness function is used to rank chromosomes in $\hat{P}(t)$ by their performance.
Selection of Next Generation Population Pool:
The next generation population pool $P(t+1)$ is selected based on their fitness values.
Termination:
Best-fit chromosome satisfies design specification, or maximum iterations reached
No
Yes
Stop

Figure 1.3: Flowchart for a Conventional GA

from one generation to the next, since crossover occurs between very similar parent chromosomes. This makes the future generations very similar and curtails the ability of GAs to continue to search for better solutions. When this happens, the GA cannot converge to the global optimal, rendering the algorithm ineffective. Such cases of premature convergence happen if the probability of mutation is too low or if the seed chromosome is chosen inappropriately.

1.3 DCGA Optimization Technique

In GAs, the method of selection is such that if there is a dominant chromosome (one that shows a comparatively high fitness value) in the population pool, then it is likely that it will produce several offspring. On the other hand, other chromosomes with lower fitness values contribute less or even no offspring at all. Consequently, in such cases, as the GA progresses from one generation to the next, the population pool experiences a rapid decline in diversity and all its chromosomes start looking very similar to or are exact replicas of the dominant chromosome. Obviously, this loss of diversity happens at an exponential rate and results in the GA converging prematurely [3].

The solution to avoiding this entrapment at local optima is to ensure that the population pool remains diverse. However, care must be taken that the GA is not made to be so diverse that the convergence rate starts to look like that of a completely random search. Consequently, an appropriate level of diversity must be constantly maintained in the population pool at all times. There have been many methods developed to achieve this dual goal, based on increasing the rate of mutation, or using selection techniques that maintain a diverse population pool.

Srinivas [22] proposed the use of adaptive probabilities for crossover and mutation to maintain the diversity of the GA, where the probabilities of crossover and mutation are varied depending on chromosome fitness values. Ling Cen *et. al.* introduced SA to prevent GA from premature convergence [23]. SA is based on the idea that initial iterations are likely to produce results that are worse off than their predecessors in order to incorporate as much diversity as possible in the earlier stages of the algorithm. As the SA continues, the likelihood of upward changes is reduced, allowing the algorithm to converge. Shimodaira [3] proposed DCGA where a cross generational probability survival selection (CPSS) scheme is used to select candidate chromosomes for the next generation that successfully explore the solution space and escape from local optimum points.

DCGA is used from here on as it strikes a good balance between maintaining a diverse population pool among GA without making the GA so random that the convergence time renders the algorithm ineffective. DCGA has some essential differences compared to the conventional GA and consists of the following steps:

- Initialization: A seed chromosome is formed similar to that in the GA (see Section 1.2). Once again, the seed chromosome is based on the application. The initial population pool consists of N chromosomes, where N is even.

- Generation of mating pool: In DCGA, the members of a current population pool $P(t)$ and their offspring chromosomes are combined to form an enlarged population pool $\hat{P}(t)$ of $2N$ chromosomes. The offspring are produced using two-point crossover (using random pairing) and mutation at a constant rate. Mutation is not allowed to affect the best fit chromosome (seed chromosome when $t = 0$). Any duplicates formed by the genetic processes are removed from the population pool.

- Evaluation: The chromosomes in the enlarged population pool $\hat{P}(t)$ are ranked by evaluating their fitness values. The fitness function depends on the application under optimization, and usually ensures that all design specifications are satisfied.

- Selection of the next-generation population pool: The best chromosome in $\hat{P}(t)$ is directly allowed to be a member of the next-generation population pool $P(t+1)$. The remaining $2N1$ chromosomes in the population pool $P(t+1)$ are selected from the enlarged population pool $\hat{P}(t)$ based on the following CPSS probability relation:
$$p_s = [(1-c) \times h/L + c]^{\alpha} \tag{1.1}$$
where h represents the hamming distance (i.e. the number of bit locations at which a given chromosome is different from another chromosome) between a candidate chromosome and the best fit chromosome in $\hat{P}(t)$, where L represents the total bit-length of the individual (binary) chromosomes, and where c and α denote, respectively, the shape coefficient and the exponent parameter. In this way, the selected chromosome with hamming distance h is chosen as

a candidate for the next generation population pool if p_s is greater than a locally generated uniform random number between '0' and '1'. This selection process is referred to as the CPSS scheme.

The CPSS scheme is repeated until $N - 1$ chromosomes are selected for the next generation population pool $P(t + 1)$ (the best fit chromosome having already been selected). However, in situations when CPSS leads to less than $N - 1$ chromosomes, the remaining chromosomes are formed by randomly complementing bits in the best fit chromosome. It should be pointed out that according to Eqn. 1.1, chromosomes with a large hamming distance from the best fit chromosome will have a greater chance to be selected as members of the next generation. In other words, the more similar the structure of one chromosome is compared to the best fit one, the less the chance it will have of being selected.

- Termination: The above steps are repeated until the design specifications are satisfied or the maximum number of allowed iterations is reached. In principle, DCGA is capable of finding the global optimal solution provided that no bound is imposed on the constituent number of generations. However, in practical situations, DCGA is set to terminate once all of the design specifications have been satisfied. In such situations, the resulting solution may or may not represent the global optimal solution, but simply a solution that satisfies all of the given design specifications.

There is no known relationship between the values of c and α and convergence rate. Therefore, the values of these parameters must be selected empirically, and is highly dependent on the characteristics of the fitness function. The speed of convergence to an optimal solution is, in general, affected by both c and α values. Therefore, empirical investigation is best conducted over a range of values of c and α to identify the values that correspond to a rapid convergence. A DCGA having appropriate c and α values has the following advantages over other discrete optimization techniques [3]:

- The best fit chromosome is always selected as a member of the next-generation population pool. This acts as an anchor point, so even if the rest of the population pool is made diverse through, for example, high rates of mutation, the search region always remains within the vicinity of the best fit chromosome. This is particularly useful if the best fit chromosome lies in a region near the global optimum. In the conventional GAs, however, crossover and mutation may destroy the best fit chromosome and this setback usually has a slow recovery.

- If the best fit chromosome is close to a region having a local optimum, there is a chance that it might become dominant and cause a premature convergence in a conventional GA. However, with DCGA the next-generation selection is based on the hamming distances of the best fit chromosome with the remaining population pool. This means that even if the best fit chromosome is near a local optimum value, it is not allowed to become a dominant chromosome in the population pool, thus preventing premature convergence.

- Through the use of the shape coefficient c and the exponent parameter α, DCGAs permit the desired external control of the diversity in the population pool for rapid convergence to an optimal solution. Different applications require different values of c and α [11]. If the application being optimized presents a fitness function having only a few local optima than a high value for c and/or low value for α results in a rapid optimization. This is because, with a few local optima, the chance of entrapment is low and therefore low levels of diversity suffice. Conversely, if the application presets a fitness function having several local optima then a low value of c and/or high value of α ensures that the population pool maintains a high level of diversity, thus allowing it to escape the local optima and avoid premature convergence.

- The amount of computations per generation in DCGA and conventional GAs are almost the same (only addition is the CPSS). The computational time of DCGA is much less than the that of the joint optimization technique employing SA in [23] and the variable crossover and mutation rate method in [22].

DCGA tends to have a convergence time around an order of magnitude less than conventional GAs. For example, in [19] it took GA a 1000 generations to arrive at a bandpass FRM FIR digital filter, whereas the proposed DCGA took only

144 generations for satisfying the same design specifications. For the applications presented in this work, the value of c and α are determined empirically, but a comparison between GA and DCGA is not made. The results of DCGA optimization are, however, compared with an adaptive version (see Appendix 1).

1.4 Summary

This section has presented a brief description of two main discrete optimization algorithms, namely the GA and the DCGA. These algorithms are known to be effective in optimizing functions that are highly non-linear, multimodal and multivariate, and can work in discrete variable domains. DCGA is applied to the optimization of FRM digital filters having finite-wordlength CSD multiplier coefficients. Two main categories of FRM digital filters are introduced: FRM digital filters incorporating IIR interpolation digital subfilters realized as a parallel combination of two allpass networks that are implemented using bilinear-LDI design technique, and FRM digital filters incorporating IIR interpolation digital subfilters realized as a parallel combination of two allpass networks that are implemented using lattice WDF design technique.

GAs perform a parallel search over the domain by forming a population pool of potential candidate solutions called chromosomes. The GA creates new chromosomes based on the current best performing chromosomes, and the population pool increases in average performance as the GA progresses from one generation to the next. The parallel nature of GAs gives it the advantage of being able to quickly find the global optimum, and this is done without any recourse to gradient information. Unfortunately, GAs are susceptible to premature convergence to a local optimum as a result of a rapid decline in diversity within the population pool.

DCGA is an effective mechanism by which the diversity of a population pool can be externally controlled using the shape coefficient c and exponent parameter α. DCGA is just as effective as GAs in finding an optimal solution, but at the same time it is capable of avoiding entrapment at local optimal points. DCGA allows the best fit chromosome of a current generation to be automatically selected for the next generation, thus making sure that if the best fit chromosome is in a region close to the global optimum the algorithm can quickly converge. It also employs an innovative selection mechanism called cross-generational probabilistic survival scheme (CPSS) that maintains diversity in the population pool. The CPSS ensures non-elite chromosomes are chosen based on their hamming distances from the best fit chromosome, thus giving the offspring the potential to break free of entrapment at local optima. It is observed that DCGA increases the convergence speed of GA by an order of magnitude.

The direct application of DCGA to optimize the IIR based FRM digital filters introduces the problems of maintaining the required CSD number format of multiplier coefficients and maintaining BIBO stability of the overall filter. In the case of realization using the bilinear-LDI design technique these problems are resolved using a novel set of LUTs such that DCGA optimization leads to FRM digital filter chromosomes whose IIR interpolation subfilters remain automatically BIBO stable under the operations of crossover and mutation throughout the course of optimization. In the case of realization using the lattice WDF technique, a LUT-based approach ensures the DCGA only searches the multiplier coefficient space that adheres to a CSD number format.

Chapter 2

Background

The previous chapter provided an introduction to GA and DCGA optimization techniques and discussed their applications. This chapter provides general background to the proposed topics in this thesis.

This chapter proceeds as follows. Section 2.1 includes a discussion about the types of filters available, their frequency characteristics, and the difference equations of typical FIR and IIR digital filters. Section 2.2 provides an overview of the conventional FRM technique. Section 2.3 gives an explanation of the CSD number system used to represent digital multiplier coefficients and highlights some of the properties of the CSD LUTs used in DCGA optimization. Finally, Section 2.4 provides a summary of this chapter.

2.1 Analog and Digital Filters

Filters are electrical devices that are designed to manipulate the frequency spectrum of a signal. They are used in particular to attenuate the unwanted frequency components (in the stopband region of the filter) while leaving the wanted frequency components intact (in the passband region of the filter). The classical filter characteristics include the lowpass filters, the highpass filters, the bandpass filters, the bandstop filters and finally, the allpass filters [19]. Multiband filters provide a combination of such characteristics.

Filters can be broadly categorized as analog filters and digital filters. Analog filters deal with input signals that are continuous-time infinite-precision quantities such as a voltage or a current. In general, analog filters can be classified by the type of components that realize them [11]:

- Passive filters: Filters that are implemented by using resistors, capacitors and inductors. Passive filters have the advantage of guaranteed BIBO stability. However,they cannot be manufactured on integrated circuits (ICs) due to difficulties in integrating the constituent inductors.

- Active filters: Filters that are implemented by using the active components instead of inductors, lending themselves to IC fabrications. Active filters include certain active-RC filters and switched-capacitor filters.

Digital filters deal with signals which are discrete-time finite precision quantities, e.g. a voltage that is sampled and quantized. They use delays, multipliers and adders to perform numerical manipulations on sampled values of the signal. In order to convert the signal from analog to digital or, conversely, from digital to analog, A/D converters and D/A converters are required, respectively. Digital filters offer many advantages over the analog counterparts, including programmability, adaptability, high stability with respect to aging, temperatures changes, and manufacturing errors. In addition, with the impact of silicon technology scaling, digital filters can be successfully implemented for high-frequency applications with increased speeds, smaller chip areas and lower power consumptions [19].

Digital filters can be broadly classified by the length of their impulse response as FIR or IIR filters. FIR filters have the following difference equation:

$$y(n) = b_0 x(n) + b_1 x(n-1) + b_2 x(n-2) + \cdots + b_{\breve{M}} x(n-\breve{M}) \tag{2.1}$$

where $x(n)$ represents the input signal, $y(n)$ represents the output signal of the filter and $\breve{M}$ is a positive integer. The transfer function of FIR filter $H(z)$ can be derived by taking the z-transform of Eqn. (2.1):

$$H(z) = \frac{Y(z)}{X(z)} = b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_{\breve{M}} z^{-\breve{M}} \tag{2.2}$$

(subject to zero initial conditions)

In contrast to FIR filter, IIR filters have the following difference equation:

$$y(n) = b_0 x(n) + b_1 x(n-1) + \cdots + b_{\breve{M}} x(n-\breve{M}) + a_1 y(n-1) + \cdots + a_{\breve{N}} y(n-\breve{N}) \tag{2.3}$$

where both $\breve{M}$ and $\breve{N}$ are positive integers. The transfer function of an IIR filter is of the form:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_{\breve{M}} z^{-\breve{M}}}{1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_{\breve{N}} z^{-\breve{N}}} \tag{2.4}$$

In this thesis, the FRM design technique uses both IIR and FIR digital filters. Realization of FIR digital filters is straightforward. In order to realize IIR filters, one need to start from a corresponding analog prototype filter and transform the filter transfer function from analog domain to digital domain using the bilinear transformation technique. Alternatively, one can apply the wave digital filter technique to transform the filter structure from analog to digital.

Compared to the lattice WDF technique, the bilinear transformation technique is well known for preserving the BIBO stability and sensitivity properties of the analog prototype reference filter. However, in both these two techniques, one must design carefully to avoid any delay free loops. Moreover, IIR filters usually suffer from BIBO stability problems because of the existence of feedback loops (IIR filters are stable if all the poles are inside the unit circle of the complex z plane). They may also require a secondary allpass filter in order to compensate for phase distortions.

FIR filters do not suffer from the above problems. They are always BIBO stable and can be designed with having an exact linear-phase. Conventionally, FIR filters are designed by truncating an infinite duration impulse response sequence using windowing techniques (e.g. Rectangular window, Bartlett window, Kaiser window) to obtain the desired finite impulse response. However, at a cost for BIBO stability and linear-phase, FIR filters are generally more complex and involve larger number of multipliers than IIR filters. As a result, IIR filters can achieve a given filtering characteristic using less memory and shorter filter order than a similar FIR filter design.

2.2 Overview of Conventional FRM FIR Design Technique

The block diagram in Fig. 2.1 shows a conventional FRM digital filter. Let $F_a(z)$ represent the transfer function of a linear-phase FIR interpolation lowpass digital subfilter, and let $F_a(e^{j\omega}$ represent the corresponding frequency response, and where

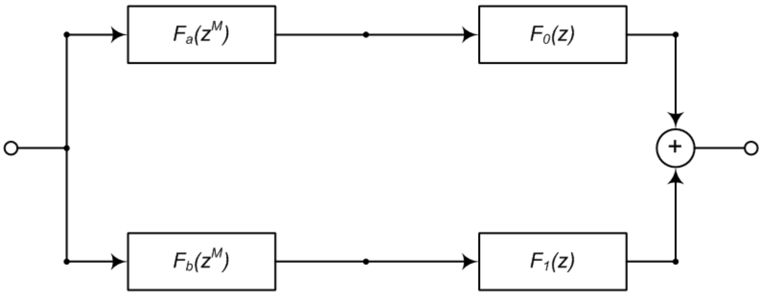

Figure 2.1: Block Diagram Representation of Frequency-Response Masking

$F_b(z)$ represents a complementary counterpart of $F_a(z)$ with $F_b(e^{j\omega}$ its corresponding frequency response. Here, z represent the discrete-time complex frequency, and ω represents the corresponding (normalized) real frequency-variable. In addition, let θ and ϕ represent the passband and stopband edge frequencies of $F_a(e^{j\omega})$, and let $\delta = \theta - \phi$ represent the transition bandwidth associated with $F_a(e^{j\omega})$. $F_b(e^{j\omega})$ represents a magnitude complementary counterpart of $F_a(e^{j\omega})$ in accordance with

$$|F_a(e^{j\omega}) + F_b(e^{j\omega})| = 1 \tag{2.5}$$

Moreover, $F_0(z)$ and $F_1(z)$ represent FIR masking digital subfilters while $F_a(z^M)$ and $F_b(z^M)$ represent M-fold interpolated versions of $F_a(z)$ and $F_b(z)$, respectively.

For a linear-phase filter $F_a(z)$ of order N_{FIR}, the relationship between $F_b(z)$ and $F_a(z)$ is as follows:

$$F_b(z) = z^{(N_{FIR}+1)/2} - F_a(z) \tag{2.6}$$

and hence $F_b(z)$ can be implemented by subtracting the output of $F_a(z)$ from the delayed version of the input, as shown in Fig. 2.2.

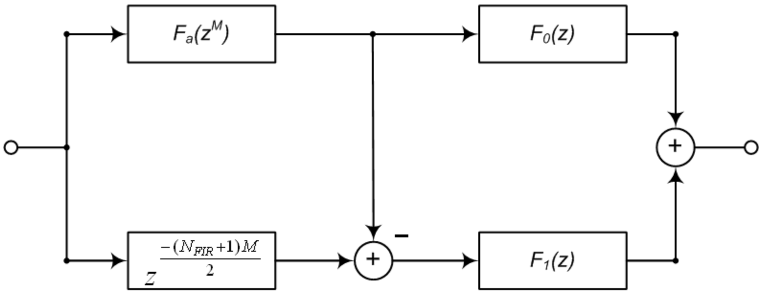

Figure 2.2: Block Diagram Representation of Frequency-Response Masking

The FRM FIR digital filter in Fig. 2.1 has an overall transfer function

$$F(z) = F_a(z^M)F_0(z) + F_b(z^M)F_1(z) \tag{2.7}$$

The masking digital subfilters $F_0(z)$ and $F_1(z)$ are employed to suppress the unwanted image bands produced by the interpolated digital subfilters $F_a(z^M)$ and $F_b(z^M)$. The masking filters are made to have equal order (by zero padding) in order to ensure that their phase characteristics are similar. The interpolated digital subfilters $F_a(z^M)$ and $F_b(z^M)$ can realize transition bands which are a factor of M sharper (i.e. δ/M) than those of $F_a(z)$ and $F_b(z)$, without increasing the number of required non-zero digital multipliers. Consequently, this technique produces overall filters with very sparse coefficients and so the resulting filter has very low computational complexity.

The magnitude frequency-responses of the various subfilters incorporated by the FRM FIR digital filter design approach are shown in Fig. 2.3. Here, Case I design is when the transition band of $F(z)$ is extracted from that of $F_a(z^M)$. Similarly, Case II design refers to the case in which the transition band of $F(z)$ is extracted from that of $F_b(z^M)$.

The edge frequencies of the the overall digital FIR filter and its constituent subfilters is given in Table 2.1, where K represents an integer value that determines the number of image lobes to be masked.

In this thesis, the conventional FRM FIR digital filter is replaced by an FRM digital filter incorporating IIR digital interpolation subfilters. The details of the design of an FRM IIR digital filter are given in Section 3.3.

2.3 CSD Number System and Quantization Errors

As mentioned in the previous chapter, from a hardware implementation point of view, a suitable design employs finite-wordlength multiplier coefficients with sparse non-zero coefficients. In this thesis, we choose the very commonly used CSD number system. Subsequently, DCGA optimization is carried out using a LUT-based scheme, where the LUTs consists of permissable CSD multiplier coefficients.

Care must be taken in making the LUTs, since making it too few entries would result in large quantization errors in the multiplier coefficients, thereby not permit-

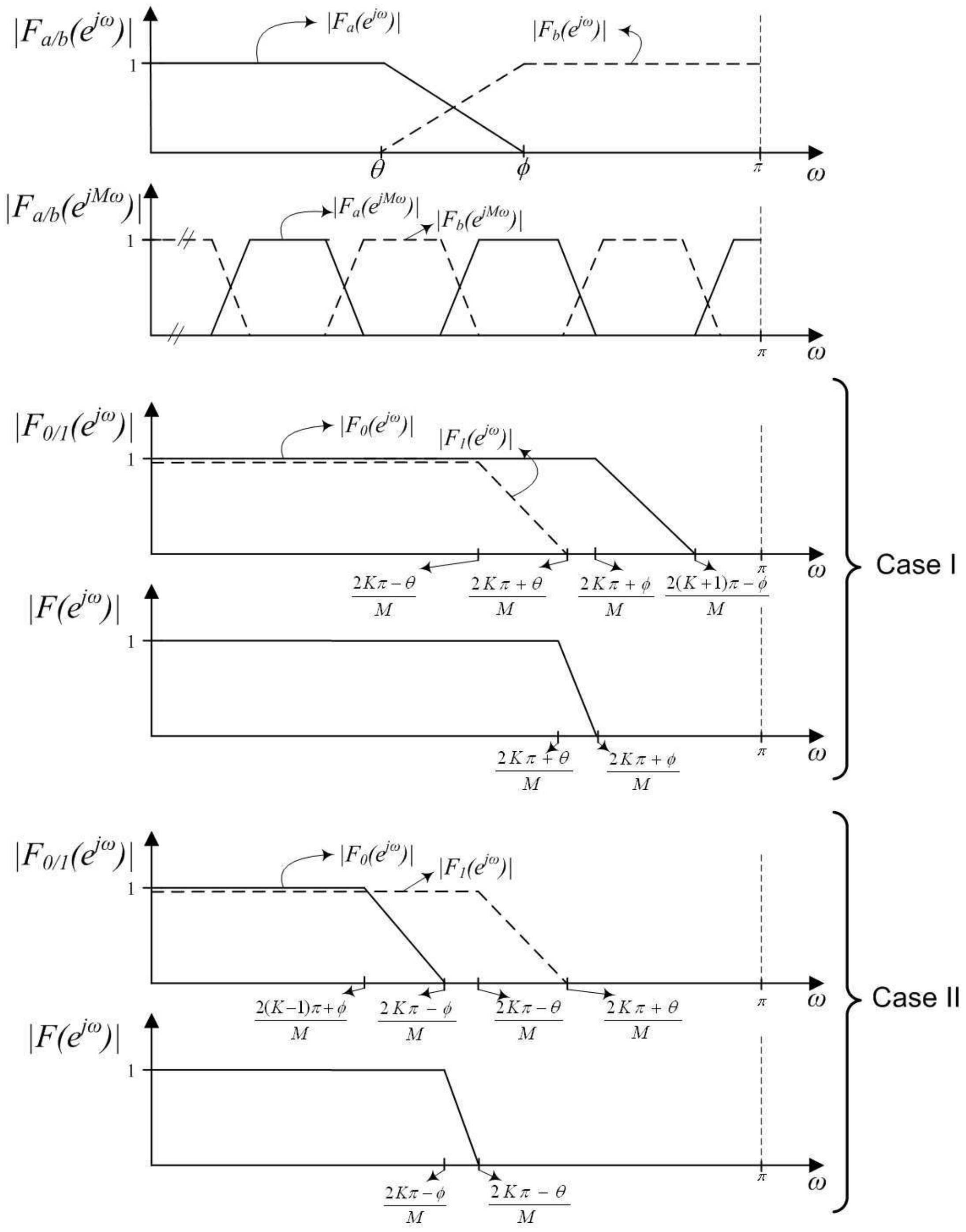

Figure 2.3: Magnitude Frequency-Response of Frequency-Response Masking Digital Filters

ting the DCGA optimization to converge to a filter satisfying design specifications. Conversely, making the LUTs too many entries greatly increase the solution space,

Table 2.1: Edge Frequencies of the Overall FRM FIR filter and Masking Subfilters

Filter	Passband Edge	Stopband Edge
Case I		
$H(z)$	$(2K\pi + \theta)/M$	$(2K\pi + \phi)/M$
$F_0(z)$	$(2K\pi + \phi)/M$	$(2(K+1)\pi - \phi)/M$
$F_1(z)$	$(2K\pi - \theta)/M$	$(2K\pi + \theta)/M$
Case II		
$H(z)$	$(2K\pi - \phi)/M$	$(2K\pi - \theta)/M$
$F_0(z)$	$(2(K-1)\pi + \phi)/M$	$(2K\pi - \phi)/M$
$F_1(z)$	$(2K\pi - \theta)/M$	$(2K\pi + \theta)/M$

and this slows down the rate of convergence of the DCGA optimization.

Let us consider a FRM digital filter consisting of CSD multiplier coefficients $\hat{m}_{FRM} \in CSD(W, w)$, where $CSD(W, w)$ represents the set of all possible CSD numbers having a wordlength of W digits and a maximum number of w non-zero digits. In this way, the CSD multiplier coefficients $\hat{m}_{FRM}$ can be expressed in the general form

$$\hat{m}_{FRM} = \sum_{n=1}^{W} D_n \times 2^{(R-n)} \tag{2.8}$$

and satisfying the constraints

$$D_n \in \{-1, 0, 1\} \tag{2.9}$$

$$D_n \times D_{n+1} = 0 \tag{2.10}$$

$$\sum_{n=1}^{W} |D_n| \leq w \tag{2.11}$$

with R representing a fixed value radix-point in the range $0 < R < W$. Constraint (2.10) implies

$$\begin{aligned} \max[w] &= W/2 && \text{for even } W \\ \max[w] &= (W+1)/2 && \text{for odd } W \end{aligned}$$

The choice of radix-point R for the LUTs depends upon the largest multiplier coefficient m_{FRM}, and can be easily determined. Setting values for W and w is more complicated, and depend on the passband and stopband ripple specification and stopband frequency of the overall FRM digital filter $F(z)$, as well as the order and stopband frequency of the interpolation digital subfilter $F_a(z)$. The greater the restriction on passband and stopband ripples, the higher is the required resolution of the LUTs (i.e. LUTs having a lower average quantization from infinite-precision to finite-precision domain). Higher resolution LUTs can be generated by increasing the wordlength W, as shown in Fig. 2.4, or the maximum number of non-zero digits w, as shown in Fig. 2.5.

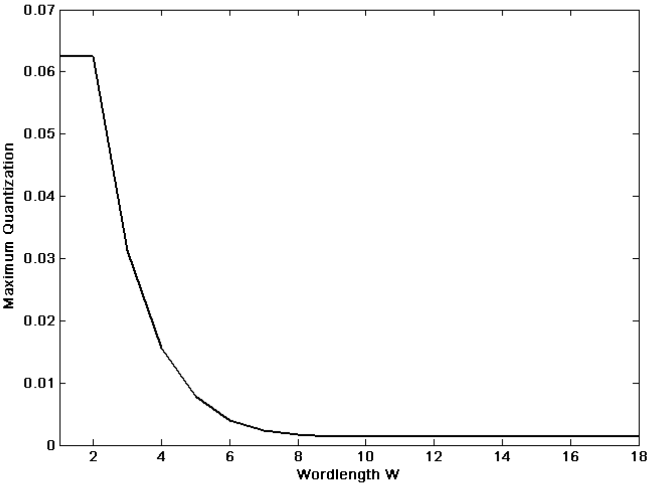

Figure 2.4: Worst Case Normalized Quantization with $w = 3$

Looking at Fig. 2.4, it is seen that the worst case normalized CSD quantization is not very sensitive to changes in W, especially after 8 bits. Therefore, it may be necessary to increase w in addition to increasing W in order to have a LUT resolution great enough to achieve desired filter specifications after optimization. But while the worst case quantization is highly sensitive to w, as seen in Fig. 2.5, increasing w is much more detrimental to hardware efficiency than simply increasing W, and w is therefore kept minimal. Note should be made that in the case of IIR interpolation digital subfilter the passband sensitivity to quantization is very low. Therefore, if the passband ripple specification is tight, it usually translates into

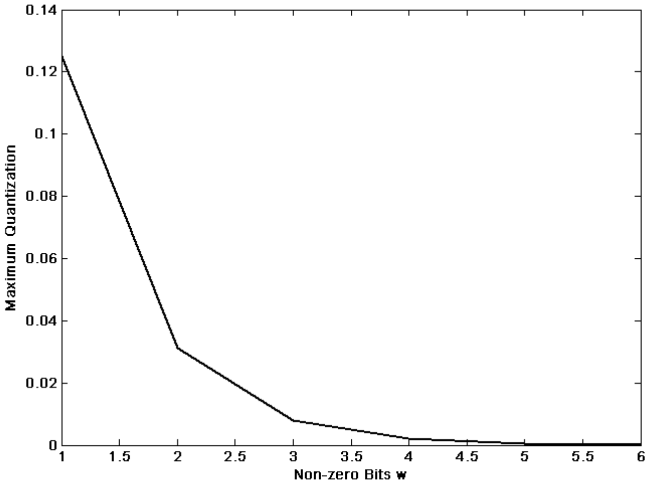

Figure 2.5: Worst Case Normalized Quantization Error with $W = 11$

requiring a higher resolution LUT for the FIR masking digital subfilters. If the stopband ripple specification is strict, the LUTs for both the interpolation digital subfilters and the FIR masking digital subfilter need to have a high resolution.

CSD LUTs with a limited value of w have a non-uniform distribution, which means that the quantization error isn't constant over the CSD range. Fig. 2.6 shows how the quantization error is distributed over a representative normalized CSD range. As can be seen, the worst case quantization increases as it advances from the least significant to the most significant end of the CSD number range. This pattern remains more or less the same regardless of the chosen values of W or w. This quantization pattern plays an important role in deciding what values of W and w while building the required LUTs. If, for instance, $F(z)$ is to have a wide band, then the corresponding FIR masking filters $F_0(z)$ and $F_1(z)$ are also wideband. This in turn results in a large central multiplier coefficient compared to the rest of the coefficient values. Since the most significant values of the CSD range are more sparsely spread, this large multiplier coefficient usually has a high quantization error going from the infinite-precision to a finite precision value. A wideband $F(z)$ therefore normally requires large values of W and/or w to reach an acceptably low ripple size as compared to a narrowband $F(z)$.

Similarly, the required resolution for the LUTs also depends upon the stopband

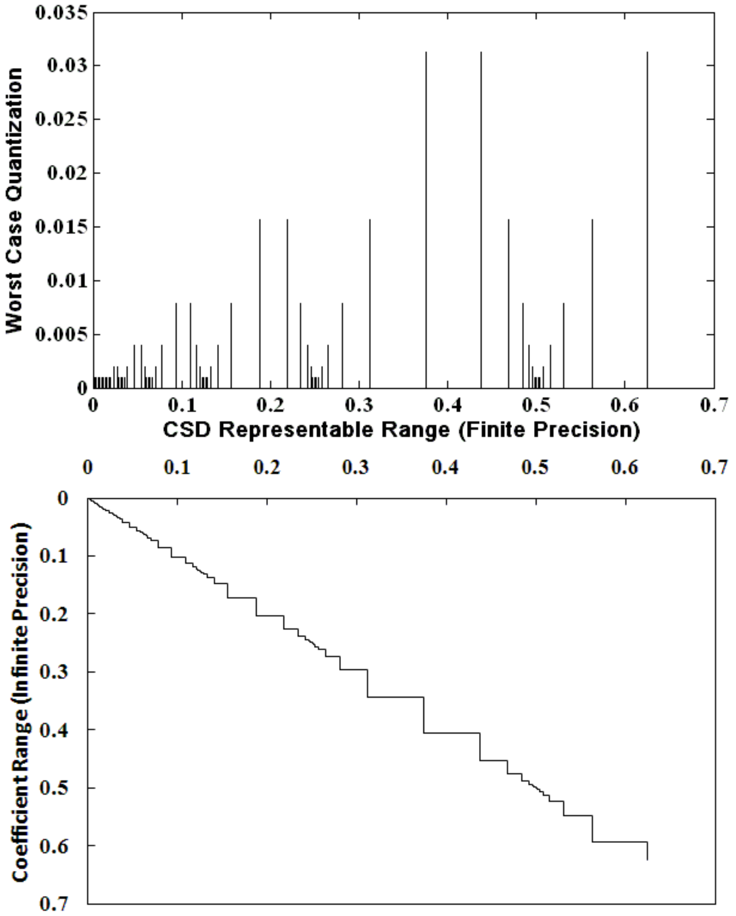

Figure 2.6: Worst Case Quantization with $W = 9$ and $w = 2$

edge specification of the interpolation filter $F_a(z)$, which in turn depends on frequency edge specifications of $F(z)$ and the interpolation factor M. The larger the stopband edge of $F_a(z)$, ϕ, the greater is the value of its central multiplier coefficient. Therefore, ϕ is kept low by choosing an appropriate value of M. (It should be noted, however, that if ϕ is made excessively low, it results in very large order FIR masking digital subfilters.)

2.4 Summary

A brief background of the basic concepts pertaining to this thesis were presented in this section. Distinctions were made between analog and digital filters, as well as between the FIR and IIR digital filters. An overview of the conventional FRM digital filter design technique, using only FIR digital subfilters, was presented. Finally, the CSD number system was discussed, with particular reference to size of CSD LUTs required in order to satisfy design specifications.

Chapter 3

Design of FRM Filters Incorporating Bilinear-LDI Digital Subfilters

This chapter discusses in detail the design, realization and discrete optimization of FRM digital filters employing IIR interpolation digital subfilters. The IIR digital subfilter design topology consists of a parallel combination of a pair of allpass networks such that its frequency-magnitude response matches that of an odd order elliptic minimum Q-factor (EMQF) transfer function. This design is realized using the bilinear-LDI approach, with multiplier coefficients represented as finite-precision CSD numbers.

The above resulting digital filter is optimized over the discrete multiplier coefficient space, resulting in an FRM digital filter which is capable of direct implementation in digital hardware without any need for further optimization. A general set of constraints is derived in terms of multiplier coefficients to guarantees that the IIR bilinear-LDI interpolation digital subfilters remain BIBO stable throughout the course of DCGA optimization. Particular attention is given to ensuring that the overall FRM digital filter is *automatically* (i.e. without the need for constantly checking digital filter chromosomes) BIBO stable throughout the course of optimization procedure. To this end, a novel worst-case LUT-based system is proposed.

This chapter proceeds as follows. Section 3.1 describes the design procedure for FRM filters incorporating IIR interpolation digital subfilters and discusses formation of power complementary filter pairs through the use of a parallel allpass digital network realization. Section 3.2 outlines the special type elliptic filters used for the design of FRM digital filters. Section 3.3 presents the design methodology for implementing an allpass network using the bilinear-LDI approach. Section 3.4 introduces a novel set of stability constraints that guarantee the BIBO stability of digital filters described in Section 3.3. Section 3.5 presents a novel LUT-based scheme that allows the DCGA to maintain BIBO stability of every chromosome throughout the optimization process. Section 3.6 outlines in detail the design and DCGA optimization of FRM digital filters incorporating bilinear-LDI based IIR digital subfilters. Finally, Section 3.7 provides a summary of this chapter.

3.1 Design of FRM Digital Filters Incorporating IIR Interpolation Digital Subfilters

In the case of FRM digital filters with IIR interpolation digital subfilters, the FIR interpolation subfilters $F_a(z)$ and $F_b(z)$ (see Section 2.2) are replaced by IIR interpolation subfilters $H_a(z)$ and $H_b(z)$. The masking filters $F_0(z)$ and $F_1(z)$ are not changed (i.e. they are still equal order FIR digital filters). Then, the FRM IIR digital filter has an overall transfer function given by:

$$H(z) = H_a(z^M)F_0(z) + H_b(z^M)F_1(z) \tag{3.1}$$

The IIR interpolation filter $H_a(z)$ is chosen to have an odd-order N_{IIR}. It is shown in [24] that odd-order elliptic transfer function can be represented as a sum of or difference between two allpass transfer functions. Therefore, $H_a(z)$ is realized as an addition of two allpass digital networks $G_0(z)$ and $G_1(z)$:

$$H_a(z) = \frac{G_0(z) + G_1(z)}{2} \tag{3.2}$$

where $G_0(z)$ is odd-ordered and $G_1(z)$ is even ordered. It so happens that the difference between $G_0(z)$ and $G_1(z)$ results in a filter that is *power complementary* to $H_a(z)$, and can therefore be used as the complementary interpolation subfilter $H_b(z)$.

$$H_b(z) = \frac{G_0(z) - G_1(z)}{2} \tag{3.3}$$

It can be easily verified that $H_a(z)$ and $H_b(z)$ are power complementary and satisfy the following relationship [9]:

$$|H_a(e^{j\omega})|^2 + |H_b(e^{j\omega})|^2 = 1 \tag{3.4}$$

In addition, it is well known that this is the most economical implementation since it requires a total of only $N_{_{IIR}}$ multiplier coefficients to realize *both* $H_a(z)$ and $H_b(z)$ (effectively halving the number of multiplier coefficients required). The block diagram in Fig. 3.1 shows the IIR interpolation digital subfilters realized as a parallel combination of two allpass networks. Fig. 3.2 shows an overall FRM IIR digital filter realization.

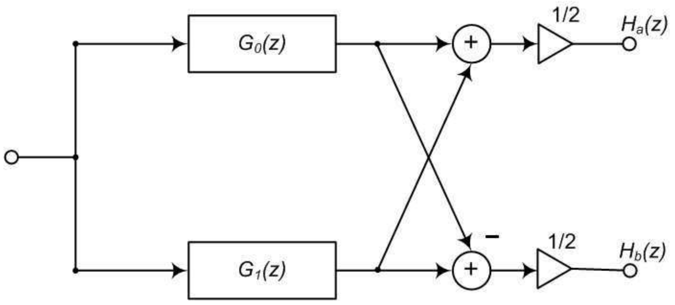

Figure 3.1: Block Diagram of Interpolation and Complementary Filters as a Parallel Combination of Two Allpass Networks

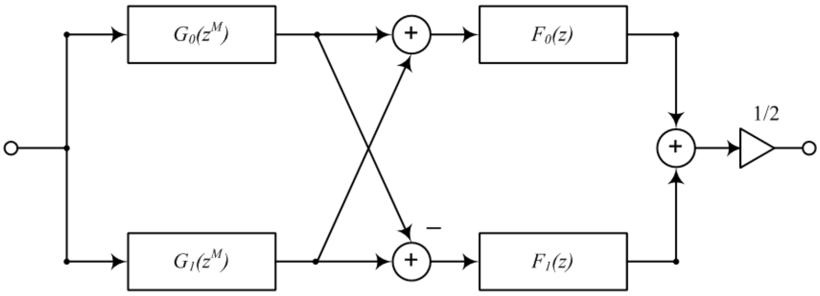

Figure 3.2: Block Diagram of the Overall FRM IIR Digital Filter

The structure in Fig. 3.2 can be rearranged in the following way. Using Eqs. (3.2-3.3), the overall transfer function of $H(z)$ given by Eqn. (3.1) can be expressed as:

$$H(z) = \frac{G_0(z^M) + G_1(z^M)}{2} F_0(z) + \frac{G_0(z^M) - G_1(z^M)}{2} F_1(z) \tag{3.5}$$

Let

$$A(z) = \frac{F_0(z) + F_1(z)}{2} \tag{3.6}$$

$$B(z) = \frac{F_0(z) - F_1(z)}{2} \tag{3.7}$$

Then $H(z)$ simplifies to

$$H(z) = G_0(z^M)A(z) + G_1(z^M)B(z) \tag{3.8}$$

The block diagram representing Eqn. (3.8) is shown in Fig. 3.3

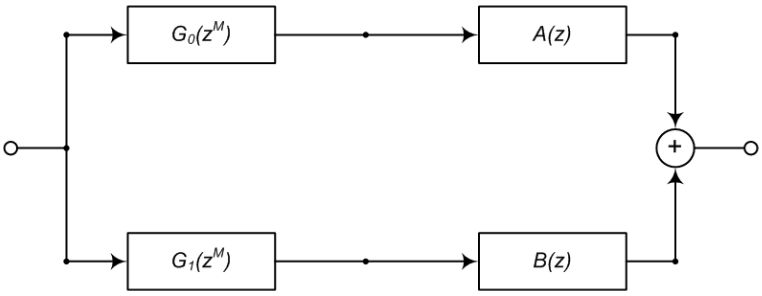

Figure 3.3: Alternative Block Diagram of the Overall FRM IIR Digital Filter

The advantage of realizing the IIR based FRM digital filter as shown in Fig. 3.3 is that two adders shown in Fig. 3.2 are no longer required. This amounts to a simplification in hardware implementation. However, it should be pointed out that the FIR masking filters $F_0(z)$ and $F_1(z)$ are made to be equal order using zero padding, and this results in the masking filters being moderately sparse. This is not the case when $A(z)$ and $B(z)$ are used instead. Therefore, the gain in hardware that could be achieved by using the realization in Fig. 3.3 is offset by a greater number of non-zero multiplier coefficients required.

3.2 Using Elliptic Filters with Minimum Q-factor (EMQF) to Realize the IIR Interpolation Digital Subfilters

As mentioned in the Chapter 2, bilinear-LDI falls into the category of digital filter realization techniques that require a corresponding analog reference filter. To this end, a suitable analog reference filter $H_a(s)$ and its complement $H_b(s)$ have to be determined in order to derive the multiplier coefficients of the IIR interpolation digital subfilters $H_a(z)$ and $H_b(z)$. This section discusses how to generate the transfer functions of the suitable analog reference filters $H_a(s)$ and $H_b(s)$, where s is the analog frequency domain variable. Once $H_a(s)$ and $H_b(s)$ have been determined, the interpolation digital subfilters $H_a(z)$ and $H_b(z)$ are derived by using bilinear-LDI technique (see Section 3.3).

As indicated by Eqn. (3.4), the squared ripple in the passband region of $H_a(s)$ is equal to the squared ripple in the stopband region of $H_b(s)$. Similarly, the squared ripple in the stopband region of $H_a(s)$ is equal to the squared ripple in the passband region of $H_b(s)$. Also, depending on whether the design specifications require a Case I or Case II FRM technique, either $H_a(s)$ or $H_b(s)$ could determine the maximum passband and stopband ripple of the overall FRM IIR digital filter $H(z)$. Consequently, the interpolation filter $H_a(s)$ is chosen to have equal passband and stopband squared tolerances. In this way, the resulting $H_b(s)$ also displays equal passband and stopband squared tolerances. Such filters fall under a special class of elliptic filters called EMQF.

An EMQF filter has all its s domain poles around a circle and so that its poles have equal magnitudes. While filters having EMQF transfer functions are maximally insensitive to component variations, they do at the expense of not being able to independently specify passband and stopband ripples [25],[26]. Additionally, EMQF filters have exceedingly low passband attenuation. Given a squared passband and stopband tolerance of δ_p and δ_a, respectively, for an EMQF filter, the passband ripple R_p and minimum stopband attenuation R_a are related by [8]:

$$R_p = -10\log(1-\delta_p) \tag{3.9}$$

$$R_a = -10\log(\delta_a) \tag{3.10}$$

The required passband and stopband edge frequencies for the analog reference filter $H_a(s)$ can be determined using design specifications along with Table 2.1. Frequency wrapping from digital to analog domain, and vise versa, has to be accounted for in accordance with

$$\Omega_A = \frac{2}{T}\tan(\frac{\omega_d T}{2}) \tag{3.11}$$

where Ω_A is the analog frequency variable, ω_d is the digital frequency variable,and T represents the sampling period.

Once the transfer function of the analog reference filter $H_a(s)$ (and its complement $H_b(s)$) is determined, it is represented as a sum (and difference, respectively) of two allpass analog filters $G_0(s)$ and $G_1(s)$. The poles of $G_0(s)$ and $G_1(s)$ are determined by cyclically distributing the poles of the prototype filter $H_a(s)$ [8], in accordance with Fig. 3.4.

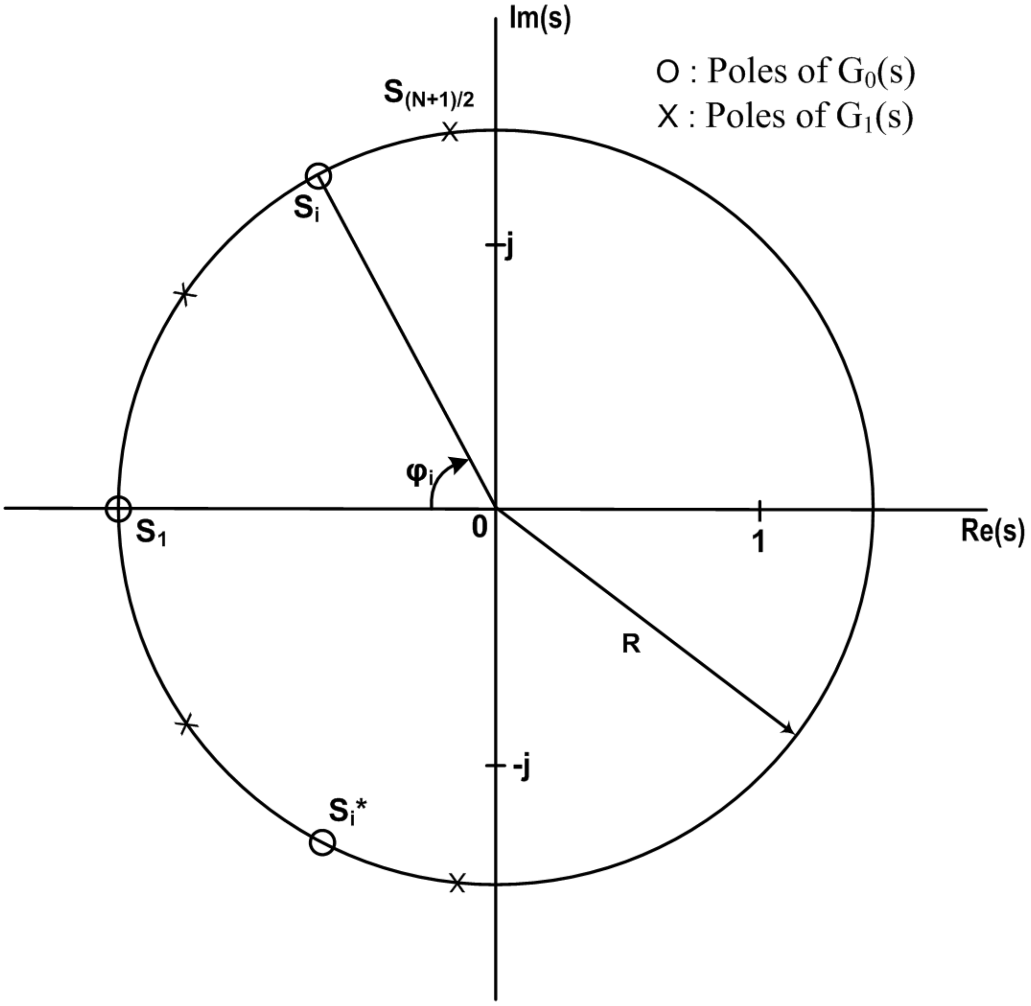

Figure 3.4: Distribution of the s-Plane Poles of $H_a(s)$

The real pole belongs to $G_0(s)$, making it an odd-ordered allpass function, while $G_1(s)$ ends up an even-ordered allpass function. The zeros are then chosen to adhere to the general form of an allpass transfer function [5]

$$G(s) = \frac{P(-s)}{P(s)} \tag{3.12}$$

where $P(s)$ is a Hurwitz polynomial of order, say, $\tilde{n}$. Moreover, $P(s)$ is expressed as

$$P(s) = EvP(s) + OdP(s) \tag{3.13}$$

where $EvP(s)$ denotes the even and $OdP(s)$ denotes the odd part of $P(s)$.

3.3 Implementation of EMQF Interpolation Subfilters Using Bilinear-LDI Design Approach

In this section, the bilinear-LDI approach is used to design physically realizable digital filters $G_0(z)$ and $G_1(z)$. The design procedure is given in [5], [27] and is briefly included here for completeness.

The bilinear frequency transformation is given by

$$s = \frac{2}{T}\frac{z-1}{z+1} \tag{3.14}$$

where T represents the sampling period, for mapping the transfer function of a prototype reference filter from the analog domain (represented by the analog frequency-variable s) to the digital domain (represented by digital frequency-variable z). The bilinear frequency transform is well known for its characteristic ability of preserving the BIBO stability (in infinite wordlength) and sensitivity properties of the analog prototype reference filter. Unfortunately, care has to taken to ensure that there are no delay-free loops using this technique.

The LDI frequency transformation is given by

$$s = \frac{1}{T}\left(z^{\frac{1}{2}} - z^{-\frac{1}{2}}\right) \tag{3.15}$$

and it maps the hardware realization of the prototype reference filter from the analog to digital domain. While the LDI frequency transformation ensures the absence of delay-free loops in the digital implementation, it results in a digital filter having poor magnitude-frequency responses.

The bilinear-LDI technique is a combination of the two above mentioned realization schemes. The method essentially consists of the application of the conventional LDI design technique to a network resulting from a pre-compensated analogue prototype filter. The pre-compensation is such that the application of the LDI design technique results in a filter that exactly matches the bilinear frequency transform of the original (un-compensated) analogue prototype filter.

The resulting bilinear-LDI digital filters themselves have the following desirable features from a hardware realization perspective:

- They are minimal in the number of digital multiplication operations and practically minimal in the number of digital adders and unit-delays.

- They lend themselves to fast two-cycle parallel digital signal processing speeds.

- They exhibit exceptionally low passband sensitivity to their multiplier coefficient values, permitting small coefficient wordlengths.

As discussed in Section 3.2, the analog reference filter $H_a(s)$ is decomposed into two allpass analog networks $G_0(s)$ and $G_1(s)$. The allpass digital networks $G_0(z)$ and $G_1(z)$ are obtained from $G_0(s)$ and $G_1(s)$ using the bilinear-LDI design technique as follows.

By simple manipulation of Eqns. (3.12) and (3.13) we get

$$G(s) = \tilde{K}\frac{1 - Z(s)}{1 + Z(s)} \tag{3.16}$$

Here, $\tilde{K} = 1$ or -1, and $Z(s)$ is a realizable reactive impedance given by

$$Z(s) = \begin{cases} \frac{OdP(s)}{EvP(s)} & \text{for even} \quad \tilde{n} \\ \\ \frac{EvP(s)}{OdP(s)} & \text{for odd} \quad \tilde{n} \end{cases} \tag{3.17}$$

where $\tilde{n}$ is the order of $G(s)$ (odd when realizing $G_0(s)$ and even when realizing $G_1(s)$). Thus the impedance $Z(s)$ has a zero at $s = 0$ for even $\tilde{n}$ and a pole at $s = 0$ for odd $\tilde{n}$, while having a zero at $s = \infty$ both for even $\tilde{n}$ and odd $\tilde{n}$.

The bilinear-LDI digital realization of $G(s)$ is achieved through the following four steps:

- *Step 1*:The transfer function $G(s)$ is decomposed in the form

$$G(s) = 1 - 2g(s) \tag{3.18}$$

where

$$g(s) = \frac{Z(s)}{1 + Z(s)} \tag{3.19}$$

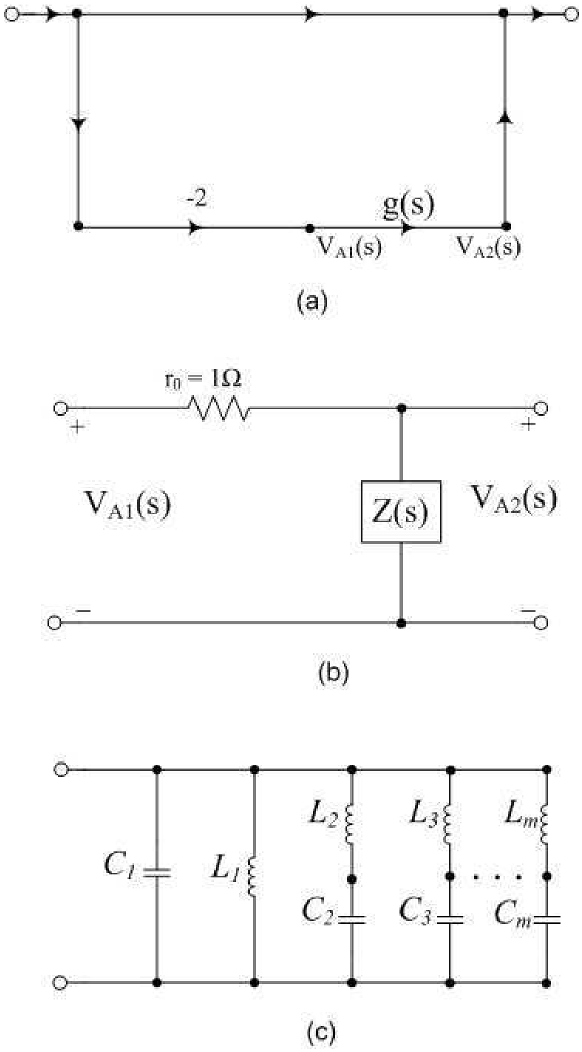

Figure 3.5: Realization of $G(s)$: (a) Signal-flow graph of $G(s)$, (b) Realization of $g(s)$ as a voltage divider, (c) Realization of $Z(s)$ as a Foster II canonical impedance

Here, $G(s)$ can be realized as the transfer function of the signal-flow graph in Fig. 3.5a. Furthermore, $g(s)$ represents a lowpass or highpass analog filter that can be realized as the transfer function of the voltage divider network in Fig. 3.5b. Finally, $Z(s)$ represents realizable reactances (consisting of capacitors and inductors only) and can be decomposed into its Foster II canonical form, as in Fig. 3.5c, in accordance with

$$Z(s) = \frac{1}{Y(s)} \tag{3.20}$$

$$Y(s) = sC_1 + \frac{1}{sL_1} + \sum_{i=2}^{m} \frac{sC_i}{s^2 C_i L_i + 1} \tag{3.21}$$

where $m = \tilde{n}/2$ for even $\tilde{n}$ and $m = (\tilde{n}+1)/2$ for odd $\tilde{n}$, and where C_p represent capacitances and L_p represent inductances (for $p = 1, 2, \cdots, m$), and inductor L_1 is only present for even $\tilde{n}$.

- *Step 2*: Pre-compensation is applied to the resulting network. This amounts to a modification of circuit elements in accordance with

$$V_{A1}^{'}(s) = \frac{V_{A1}(s)}{1 - sT/2} \tag{3.22}$$

$$r_0^{'} = r_0 * z^{1/2} \tag{3.23}$$

and

$$L_1^{'} = L_1 \tag{3.24}$$

$$C_1^{'} = C_1 + \frac{T}{2} + \frac{T^2}{4L_1} + \sum_{i=2}^{m} \frac{C_i \frac{T^2}{4L_i}}{C_i + \frac{T^2}{4L_i}} \tag{3.25}$$

$$L_i^{'} = L_i \left[\frac{C_i + \frac{T^2}{4L_i}}{C_i} \right]^2 \tag{3.26}$$

$$C_i^{'} = \frac{C_i^2}{C_i + \frac{T^2}{4L_i}} \tag{3.27}$$

with $r_0 = 1\Omega$ and for $i = 2, 3, ..., m$

- *Step4*: The analog integrators in the signal-flow graph of the pre-compensated network are replaced by LDI digital integrators, and the resulting network is scaled by $z^{-1/2}$ to eliminate any half-delays. The resulting digital network is displayed in Fig. 3.6. The multiplier coefficients here are given by

$$m_{Lp} = \frac{T}{L_p^{'}} \tag{3.28}$$

$$m_{Cp} = \frac{T}{C_p^{'}} \tag{3.29}$$

for $p = 1, 2, ..., m$

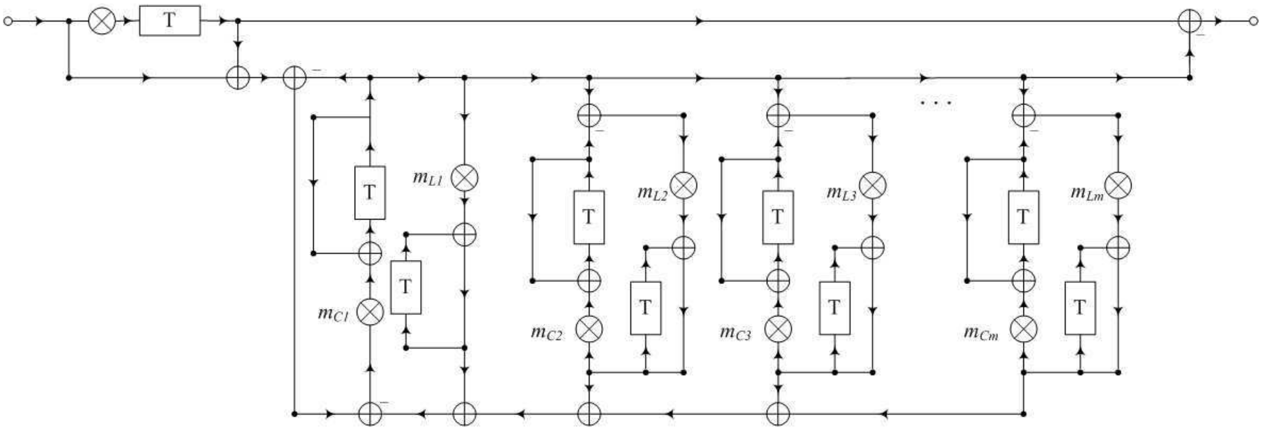

Figure 3.6: Realization of $G(z)$

3.4 Constraints for Guaranteed BIBO Stability

In order for the FRM digital filter consisting of CSD multiplier coefficients $\hat{m}_{FRM}$ to be BIBO stable, it is both necessary and sufficient for the bilinear-LDI IIR interpolation subfilters $H_a(z)$ and $H_b(z)$ to be BIBO stable. Similarly, in order for the interpolation subfilters $H_a(z)$ and $H_b(z)$ to be BIBO stable, it is both necessary and sufficient for the bilinear-LDI allpass digital networks $G_0(z)$ and $G_1(z)$ be BIBO stable. In this way, it is required that the bilinear-LDI digital allpass networks $G_0(z)$ and $G_1(z)$ remain BIBO stable throughout the course of the DCGA optimization over the CSD multiplier coefficient space $CSD(W, w)$.

In the course of DCGA optimization, the infinite-precision multiplier coefficients m_{Lp} and m_{Cp} can only take quantized values $\hat{m}_{Lp}$ and $\hat{m}_{Cp}$. By taking into account the mapping properties of the bilinear analog-to-digital frequency transformation, in order for the bilinear-LDI digital allpass networks $G_0(z)$ and $G_1(z)$ to remain BIBO stable, it required that the values of the corresponding reactive elements $\hat{L}_p$ and $\hat{C}_p$ remain positive [28]. To this end, in accordance with Eqns. (3.28) and (3.29), one has

$$\hat{L}_p' = \frac{T}{\hat{m}_{Lp}} \tag{3.30}$$

$$\hat{C}_p' = \frac{T}{\hat{m}_{Cp}} \tag{3.31}$$

Moreover, in accordance with Eqns. (3.24)-(3.27), one has

$$\hat{L}_1' = \hat{L}_1 \tag{3.32}$$

$$\hat{C}_1' = \hat{C}_1 + \frac{T}{2} + \frac{T^2}{4\hat{L}_1} + \sum_{i=2}^{m} \frac{\hat{C}_i \frac{T^2}{4\hat{L}_i}}{\hat{C}_i + \frac{T^2}{4\hat{L}_i}} \tag{3.33}$$

$$\hat{L}_i' = \hat{L}_i \left[\frac{\hat{C}_i + \frac{T^2}{4\hat{L}_i}}{\hat{C}_i} \right]^2 \tag{3.34}$$

$$\hat{C}_i' = \frac{\hat{C}_i^2}{\hat{C}_i + \frac{T^2}{4\hat{L}_i}} \tag{3.35}$$

where $\hat{L}_1 = \infty$ for odd-ordered allpass network $G_0(z)$.

By substituting Eqns. (3.30) and (3.31) into Eqns. (3.32)-(3.35), and by solving the resulting equations for the reactive elements $\hat{L}_p$ and $\hat{C}_p$, one can obtain

$$\hat{L}_1 = \frac{T}{\hat{m}_{L1}} \tag{3.36}$$

$$\hat{C}_1 = \frac{T\left\{ \frac{4}{\hat{m}_{C1}} - \hat{m}_{L1} - 4\left(\sum_{i=2}^{m} \frac{1}{\frac{4}{\hat{m}_{Li}} - \hat{m}_{Ci}} \right) - 2 \right\}}{4} \tag{3.37}$$

$$\hat{L}_i = \frac{T(\hat{m}_{Li}\hat{m}_{Ci} - 4)^2}{16\hat{m}_{Li}} \tag{3.38}$$

$$\hat{C}_i = \frac{-4T}{\hat{m}_{Ci}(\hat{m}_{Li}\hat{m}_{Ci} - 4)} \tag{3.39}$$

From Eqns. (3.36)-(3.39), $\hat{L}_p > 0$ and $\hat{C}_p > 0$ provide that

$$\hat{m}_{L1} > 0 \tag{3.40}$$

$$\hat{m}_{Li} > 0 \tag{3.41}$$

$$\hat{m}_{Ci} < \frac{4}{\hat{m}_{Li}} \tag{3.42}$$

$$\hat{m}_{C1} < \frac{4}{\left\{ \hat{m}_{L1} + 4\left(\sum_{i=2}^{m} \frac{1}{\frac{4}{\hat{m}_{Li}} - \hat{m}_{Ci}} \right) + 2 \right\}} \tag{3.43}$$

Then, in order to make the CSD FRM digital filter BIBO stable, it is necessary and sufficient to choose the values of the multiplier coefficients $\hat{m}_{FRM} \in CSD(W, w)$ such that the inequality constraints (3.40)-(3.43) are satisfied. The equations and corresponding condition required for BIBO stability are summarized in Table 3.1.

Table 3.1: Relations for Elements of Back-Transformed Reactance

Element	Equation	Inequality Constraints
$\hat{L}_1$	$\frac{T}{\hat{m}_{L1}}$	$\hat{m}_{L1} > 0$
$\hat{C}_1$	$\frac{T\left\{\frac{4}{\hat{m}_{C1}}-\hat{m}_{L1}-4\left(\sum_{i=2}^{m}\frac{1}{\frac{4}{\hat{m}_{Li}}-\hat{m}_{Ci}}\right)-2\right\}}{4}$	$\hat{m}_{C1} < 4\left\{\hat{m}_{L1}+4\left(\sum_{i=2}^{m}\frac{1}{\frac{4}{\hat{m}_{Li}}-\hat{m}_{Ci}}\right)+2\right\}^{-1}$
$\hat{L}_i$	$\frac{T(\hat{m}_{Li}\hat{m}_{Ci}-4)^2}{16\hat{m}_{Li}}$	$\hat{m}_{Li} > 0$
$\hat{C}_i$	$\frac{-4T}{\hat{m}_{Ci}(\hat{m}_{Li}\hat{m}_{Ci}-4)}$	$\hat{m}_{C_i} < 4\left(\hat{m}_{Li}\right)^{-1}$

In order to make the CSD lowpass digital IIR FRM filter BIBO stable, it is necessary and sufficient to choose the values of the multiplier coefficients $\hat{m}_{Li}$, $\hat{m}_{Ci} \in CSD(W, w)$ such that the inequality constraints of Table 3.1 are satisfied.

It should be pointed out that constraint (3.42) is most stringent when $\hat{m}_{Li}$ is at its largest possible value. Similarly, constraint (3.43) is most stringent when $\hat{m}_{L1}$, $\hat{m}_{Li}$ and $\hat{m}_{Ci}$ are all at their largest possible values (while $\hat{m}_{Li}$ and $\hat{m}_{Ci}$ still adhere to constraint $\hat{m}_{Ci} < 4\left(\hat{m}_{Li}\right)^{-1}$).

3.5 Generation of CSD LUTs

The proposed DCGA optimization of BIBO stable FRM digital filters is carried out over the CSD multiplier coefficient space $CSD(W_{0\,\mathrm{or}\,1}, w_{0\,\mathrm{or}\,1})$, where $W_{0\,\mathrm{or}\,1}$ represents the multiplier coefficient wordlength, and where $w_{0\,\mathrm{or}\,1}$ represents the maximum number of non-zero digits (for FIR or IIR digital subfilters, respectively). The multiplier coefficients values are taken from a set of CSD LUTs which are constructed as follows [11]:

- One LUT is constructed for all multiplier coefficient values $\hat{m}_{_{FIR}} \in CSD(W_0, w_0)$ for the masking digital subfilters $F_0(z)$ and $F_1(z)$. The values of W_0 and w_0 are determined empirically based on the amplitude frequency-response of the masking digital subfilters $F_0(z)$ and $F_1(z)$.
- $N_{_{IIR}} - 2$ LUTs are constructed for all multiplier coefficient values $\hat{m}_{_{IIR}} \in$

$CSD(W_1, w_1)$ except for $\hat{m}_{C1}$ in the allpass digital networks $G_0(z)$ and $G_1(z)$. That is, one LUT is constructed to include all permissible CSD values for the multiplier coefficient $\hat{m}_{L1} \in CSD(W_1, w_1)$ of $G_1(z)$ in inequality constraint (3.40), and $N_{_{IIR}}-3$ LUTs are constructed to include all permissible CSD values for the multiplier coefficients $\hat{m}_{Li} \in CSD(W_1, w_1)$ and $\hat{m}_{Ci} \in CSD(W_1, w_1)$ of $G_0(z)$ and $G_1(z)$ in the inequality constraints (3.41) and (3.42). Once again, the values of W_1 and w_1 are determined empirically. Also, it is expedient to assume that $\hat{m}_{_{IIR}}$ have only positive values.

- Unlike the inequality constraints (3.40)-(3.42), the inequality constraint (3.43) for the multiplier coefficient $\hat{m}_{C1}$ involves more than one other multiplier coefficient. To circumvent this problem, let us replace the inequality constraint (3.43) by a corresponding equality constraint in accordance with

$$\hat{m}_{C1} = \frac{4}{\left\{\hat{m}_{L1} + 4\left(\sum_{i=2}^{m} \frac{1}{\frac{4}{\hat{m}_{Li}} - \hat{m}_{Ci}}\right) + 2\right\}} - \epsilon \tag{3.44}$$

  where ϵ is a positive infinite-precision slack variable. Then the problem under consideration amounts to a judicious selection of $\epsilon > 0$. In this way, one can bypass any direct reference to the value of the multiplier coefficient $\hat{m}_{C1}$. Instead, one can incorporate the value of the slack variable ϵ as a gene in the construction of the FRM digital filter chromosome. This is best facilitated by replacing the slack variable ϵ by a corresponding finite-precision counterpart $\hat{\epsilon}$, and by representing $\hat{\epsilon}$ by a binary number having a suitable wordlength. Then, the equality constraint (3.44) is replaced by the best approximation

$$\begin{aligned} \hat{m}_{C1} &\in CSD(W_1, w_1) \\ &\approx \frac{4}{\left\{\hat{m}_{L1} + 4\left(\sum_{i=2}^{m} \frac{1}{\frac{4}{\hat{m}_{Li}} - \hat{m}_{Ci}}\right) + 2\right\}} - \hat{\epsilon} \end{aligned} \tag{3.45}$$

  One $\hat{\epsilon}$ is required for each of the allpass digital networks $G_0(z)$ and $G_1(z)$.

Unfortunately, introducing the slack variables causes certain problems. For instance, the multiplier coefficient being represented by the slack variable could possibly undergo unwanted changes as a result of changing other multiplier coefficients, even if the slack variable remains fixed. In addition, there often tends to be a many-to-one mapping between the slack variable and the multiplier coefficient it represents, since the slack variable is binary and the rest of the multiplier coefficients are CSD. This causes redundancy during the optimization process. Moreover, a lot of unnecessary operations (e.g. conversion of the binary slack variable to its decimal value) are involved.

In order to circumvent the problems mentioned above, a novel LUT-based approach is described here. The salient feature of the proposed LUT scheme is that it makes no recourse to slack variables for referencing the values of the CSD multiplier coefficients, while still satisfying all constraints necessary to guarantee BIBO stability of the overall FRM digital filter. The multiplier coefficients values are taken from a set of CSD LUTs which are constructed as follows:

- One LUT is constructed for all multiplier coefficient values $\hat{m}_{_{FIR}} \in CSD(W_0, w_0)$ for the masking digital subfilters $F_0(z)$ and $F_1(z)$.

- A template LUT is constructed for all multiplier coefficient values $\hat{m}_{_{IIR}} \in CSD(W_1, w_1)$ for the digital allpass networks $G_0(z)$ and $G_1(z)$. Once again, it is expedient to assume that $\hat{m}_{_{IIR}}$ have only positive values.

- The template CSD LUT is used to form one size-reduced LUT per multiplier coefficient for digital allpass networks $G_0(z)$ and $G_1(z)$, where each size-reduced LUT initially includes CSD values bounded from below by the smallest representable value belonging to $CSD(W_1, w_1)$, and from above by the corresponding value of the finite-wordlength seed coefficient. The size-reduced LUTs are augmented before DCGA optimization process commences. The purpose of this augmentation is to ensure that the exploration space include as many of those CSD multiplier coefficients $\hat{m}_{L1}$, $\hat{m}_{C1}$, $\hat{m}_{Li}$ and $\hat{m}_{Ci}$ which still satisfy the BIBO stability constraints (3.40)-(3.43). A total of $N_{_{IIR}}$ size-reduced LUTs are formed.

If necessary, the above LUTs are increased in size in such a manner that the number of CSD values in each LUT is a power-of-two. Then, the indices into the CSD LUT in each LUT can be represented by a finite wordlength binary number, and the resulting index sets become automatically closed under genetic operations.

Having constructed the above LUTs at the outset, they remain fixed throughout the course of DCGA optimization. In this way, FRM digital filter chromosomes are formed by concatenating the binary indices into the corresponding CSD LUTs for the constituent multiplier coefficients.

3.6 Design Methodology

The design methodology for the proposed DCGA optimization of BIBO stable bilinear-LDI based FRM digital filters over the CSD multiplier coefficient space can be summarized as follows:

1. *Designing the IIR interpolation digital subfilter*: the first step in determining the interpolation subfilter specifications is to fix the interpolation factor M from a pre-specified range. This is done in a way that the order of the FIR masking filters is kept minimal. Using the passband edge frequency ω_p and stopband edge frequency ω_a and the expressions for boundary frequencies given in Table 2.1, one can determine the filter case and calculate the approximate passband edge $\tilde{\theta}$ and stopband edge $\tilde{\phi}$ of the digital interpolation lowpass subfilter $H(e^{j\omega})$, for every value of the user specified range of interpolation factors M. The order of the FIR masking filters depends on the minimum distance between consecutive image replicas of either the interpolated subfilter $H_a(e^{jM\omega})$ or its complement $H_b(e^{jM\omega})$. Then, displacement λM and distance $\tilde{D}_M$ for each interpolation factor M are given as:

$$\lambda_M = \max[|(\frac{\pi}{2} - \tilde{\theta})|, |(\frac{\pi}{2} - \tilde{\phi})|] \tag{3.46}$$

$$\tilde{D}_M = \frac{\pi}{M} - \frac{2\lambda}{M} \tag{3.47}$$

   To minimize the length of FIR-masking filters, the value of M that results in the largest value of $\tilde{D}_M$ is chosen. This determines the optimal interpolation factor M as well as the approximate passband edge $\tilde{\theta}$ and stopband edge $\tilde{\phi}$ of the digital interpolation subfilter $H(e^{j\omega})$.

   EMQF filters have the property of equal square magnitude ripple size in the passband and stopband. Therefore, of the two ripple specifications, whichever gives the smallest tolerance in the squared magnitude response determines both the passband ripple R_p and stopband ripple R_a of the interpolation digital subfilter $H_a(e^{j\omega})$. The interpolation digital subfilter order $N_{_{IIR}}$ is then determined using R_p, R_a, $\tilde{\theta}$ and $\tilde{\phi}$. $N_{_{IIR}}$ must be rounded to the nearest larger odd integer so that it can be implement by a parallel combination of two allpass networks. With the order $N_{_{IIR}}$, and passband and stop band ripples R_p and R_a fixed, the ratio of the analog passband edge θ_A and stopband edge ϕ_A is a constant k given by [29]

$$D = \frac{10^{0.1R_a} - 1}{10^{0.1R_p} - 1} \tag{3.48}$$

$$q = 10^{\frac{-\log(16D)}{N_{IIR}}} \tag{3.49}$$

$$q = q_0 + 2q_0^5 + 15q_0^9 + 150q_0^{13} \tag{3.50}$$

$$k_p = \left[\frac{1-2q_0}{1+2q_0}\right]^2 \tag{3.51}$$

$$k = \sqrt{1-k_p^2} \tag{3.52}$$

In order to satisfy to the passband edge specification, the digital passband edge $\theta = \tilde{\theta}$ for Case I filters. The digital stopband edge ϕ is then determined using the analog ratio k. (Here, frequency warping from digital to analog domain, and vise versa, given by Eqn. (3.11) needs to be taken into account .) Similarly, $\phi = \tilde{\phi}$ for Case II filters, and θ can be determined by using ratio k. Also, using given ripple specifications along with the boundary frequencies described in Table 2.1, one can determine the transfer function of the FIR masking filters $F_0(e^{j\omega})$ and $F_1(e^{j\omega})$.

2. *Generation of seed FRM digital filter chromosome*: The seed FRM digital filter chromosome is formed in an ordered manner by concatenating:

   - A block of $B_{_{IIR}}$ bits serving as the binary index into the corresponding CSD LUT for each multiplier coefficient in the bilinear-LDI allpass digital networks $G_0(z)$ and $G_1(z)$ (except for the multiplier coefficient $\hat{m}_{C1}$ in case when a slack variable is used) .
   - A block of $B_{\hat{\epsilon}}$ bits serving as the binary value of the slack variable $\hat{\epsilon}$, in the case when a slack variable is used. If the worst-case LUT technique is used this step is skipped.

- A block of $B_{_{FIR}}$ bits serving as the binary index into the corresponding CSD LUT for each multiplier coefficient in the FIR masking digital subfilters $F_0(z)$ and $F_1(z)$.

In this way, each of the above blocks of bits forms a gene in FRM digital filter chromosome.

3. *Generation of Initial Population Pool*: An initial population pool of N chromosomes is formed by scanning the FRM digital filter seed chromosome successively one gene at a time, and by randomly flipping the bits in the ρ-th bit in a given gene in accordance with the probabilistic relationship $p_{_F} \times 0.5^{B_\rho+1-b_\rho}$ (with b_ρ representing the current bit position within the gene, and with B_ρ representing the wordlength of the gene), where $p_{_F}$ is a fixed probability factor.

4. *Formation of the Next Generation Population Pool*: The current population pool $P(t)$ of size N is replaced by an enlarged population pool $\hat{P}(t)$ of size $2N$ by using the following genetic operations:

   - *Crossover Operations*: Chromosomes in the population pool $P(t)$ are randomly paired as parents in such a manner that each chromosome is chosen only once as a parent. The resulting parent chromosome pairs undergo two-point crossover operations, reproducing two offspring for each parent pair. The resulting offspring are then combined with the initial population pool $P(t)$ to form the enlarged population pool $\hat{P}(t)$.
   - *Mutation Operations*:A few chromosomes (determined by a small mutation probability $p_{_M}$) in the enlarged population pool $\hat{P}(t)$ undergo mutation operation by randomly flipping their bits to enhance diversity.

   Any resulting duplicate chromosomes are eliminated so as to maintain diversity, and the chromosomes in the enlarged population pool $\hat{P}(t)$ are ranked by evaluating their fitness values. The next generation population pool of $P(t+1)$ is then formed by using the CPSS scheme.

5. *Fitness Evaluation*: The fitness of each of the resulting FRM digital filter chromosomes in the enlarged population pool $\hat{P}(t)$ is evaluated in accordance

with

$$fitness_{magnitude} = -20log[max(\varepsilon_p, \varepsilon_a)] \tag{3.53}$$

$$fitness_{group-delay} = \varsigma_p \tag{3.54}$$

$$fitness = fintess_{magnitude} - fitness_{group-delay} \tag{3.55}$$

where

$$\varepsilon_p = \underbrace{max}_{\omega \in \Delta\omega_p} [W_p|H(e^{j\omega}) - 1|] \tag{3.56}$$

$$\varepsilon_a = \underbrace{max}_{\omega \in \Delta\omega_a} [W_a|H(e^{j\omega})|] \tag{3.57}$$

$$\varsigma_p = \underbrace{max}_{\omega \in \Delta\omega_p} [W_{gd}|\tau(\omega) - \mu_\tau|] \tag{3.58}$$

with $\Delta\omega_p$ representing the passband frequency regions, $\Delta\omega_p$ representing the stopband frequency region, and $\tau(\omega)$ representing the group-delay frequency response of the FRM digital filter. Here, W_p, W_a, and W_{gd} represent the passband, stopband, and group delay weighting factors, and μ_τ being the average group delay over the passband region.

The group-delay of $H(e^{j\omega})$ is given by

$$T_G(\omega) = -\text{Im}\left\{ \frac{1}{H(e^{j\omega})} \frac{dH(e^{j\omega})}{d\omega} \right\} \tag{3.59}$$

The passband weighting factors W_p and stopband weighting factors W_a, are easily determined from user specifications. The group-delay weighting factor is formulate as

$$W_{gd} = \frac{\zeta \times fitness_{magnitude}}{fitness_{group-delay}} \tag{3.60}$$

where ζ is a fixed constant such that $0 < \zeta < 1$, and where $fitness_{magnitude}$ and $fitness_{group-delay}$ are determined from the seed FRM digital filter chromosome. The importance of group-delay in the optimization increases as $\zeta \rightarrow 1$.

In Section 5.1, Section 5.2 and Section 5.3 examples illustrating the DCGA optimization of the design procedure described above are presented. Comparisons are made between the two methods to generate CSD LUTs, namely, the LUT-based approach that employs slack variables, and the worst-case LUT approach.

3.7 Summary

In this section, the design and DCGA optimization of a FRM digital filter having IIR interpolation digital subfilters is discussed. The IIR interpolation subfilters are chosen from a special class of elliptic filters having EMQF transfer function, and are designed in an efficient manner using the addition and subtraction of allpass networks. The allpass networks are realized using the bilinear-LDI design technique. The BIBO stability criterion for a general order bilinear-LDI allpass network is derived in terms of the constituent multiplier coefficients. Automatic BIBO stability of the FRM digital filter is guaranteed by means of a novel LUT scheme that uses the mentioned BIBO stability criterion. Finally the DCGA optimization procedure is discussed in detail, with particular reference to the novel cost-function that is capable of simultaneously optimizing both magnitude-frequency and group-delay frequency response.

Chapter 4

Design of FRM Filters Incorporating Lattice Wave Digital Subfilters

In the previous chapter, the design and DCGA optimization of FRM digital filters incorporating bilinear-LDI allpass networks was described. This chapter discusses in detail the design, realization and discrete optimization of FRM digital filters employing lattice WDF interpolation digital subfilters. The IIR digital subfilter design topology consists of a parallel combination of a pair of allpass networks such that its frequency-magnitude response matches that of an odd order elliptic minimum Q-factor (EMQF) transfer function. The allpass networks are realized using lattice WDF, with multiplier coefficients represented as finite-precision CSD numbers.

The lattice WDF design is decomposed into first order and second order allpas sections. These sections are then cascaded to produce the required allpass network characteristics. The salient feature of this technique is that half of all the second order allpass section multiplier coefficients can be fixed at a single, easily implementable value. This common multiplier coefficient is independent of the filter order and its transition bandwidth, and its value only depends on the frequency for which the filter attenuation is 3dB. If the 3dB attenuation frequency is adjusted, the common multiplier coefficient can be implemented using a predetermined number of shift-and-add operations.

Subsequently, the realizable digital filter is optimized over the discrete multiplier coefficient space, resulting in an FRM digital filter which is capable of direct implementation in digital hardware without any need for further optimization. The common multiplier coefficient does not need to be optimized, since it implemented as an exact value (it does not undergo quantization). This reduces the number of design variables to be optimized to half, thereby increasing the speed of convergence. Unlike the bilinear-LDI filter implementation technique, there is no BIBO stability issue with lattice WDFs implementation. This stems from the fact that a the lattice WDF is composed of first and second allpass sections cascaded together, and this simplifies the BIBO stability criteria on the multiplier coefficients.

This chapter proceeds as follows. Section 4.1 provides an overview of lattice WDFs and contains its general transfer functions. Section 4.2 describes how lattice WDFs are cascaded into two parallel branches of allpass networks in order to realize the interpolation digital subfilter. Section 4.3 works out the way to halve the number of quantized multiplier coefficients required to implement the interpolation digital subfilter and its complement. Section 4.4 outlines in detail the design and DCGA optimization of FRM digital filters incorporating lattice WDF based IIR digital subfilters. Finally, Section 4.5 provides a summary of this chapter.

4.1 Overview of Lattice WDFs

The principle of WDFs are described in [6]. Only basic definitions, as given in [30], are repeated here.

The lattice WDFs are derived from real lossless symmetric two-port equally resistively terminated reference filters. The reference filter is an analog prototype designed in the s-domain. The transfer function of the lattice WDF is obtained by the bilinear transformation given by Eqn. (3.14). The relationship between the frequencies of reference analog filter and the frequencies of the WDF is related by Eqn. (3.11):

The wave-flow diagram of a lattice WDF is shown in Fig. 4.1. In the branches of the lattice WDF, $S_1(s)$ and $S_2(s)$ are allpass functions. Therefore, they may be expressed in the following general form:

$$S_1(s) = \frac{g_1(-s)}{g_1(s)} \tag{4.1}$$

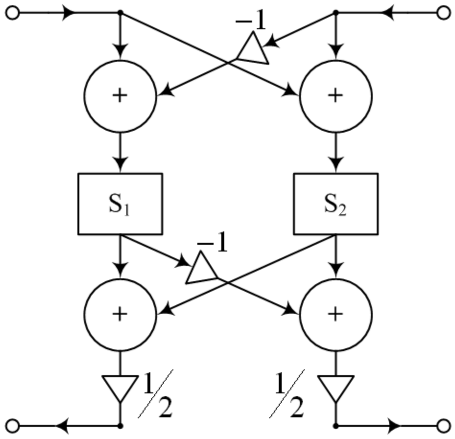

Figure 4.1: Wave-Flow Diagram of Lattice WDFs

and

$$S_2(s) = \frac{g_2(-s)}{g_2(s)} \tag{4.2}$$

where $g_1(s)$ and $g_2(s)$ are Hurwitz polynomials of, say, degrees $\tilde{n_1}$ and $\tilde{n_2}$, respectively.

It is well known that the transfer functions that are realized by these WDFs are given by:

$$S_{11}(s) = S_{22}(s) = \frac{S_1(s) + S_2(s)}{2} = \frac{\tilde{\tilde{h}}(s)}{\tilde{\tilde{g}}(s)} \tag{4.3}$$

$$S_{12}(s) = S_{21}(s) = \frac{S_2(s) - S_1(s)}{2} = \frac{\tilde{\tilde{f}}(s)}{\tilde{\tilde{g}}(s)} \tag{4.4}$$

where $\tilde{\tilde{h}}(s)$, $\tilde{\tilde{f}}(s)$ and $\tilde{\tilde{g}}(s)$ are polynomials in s.

In the following it is expedient to assume that $\tilde{n}_1$ is odd and $\tilde{n}_2$ is even. The opposite choice would simply amount to changing the sign of Eqn. (4.4). It is seen from Eqns. (4.1)-(4.4) that:

$$\tilde{\tilde{g}}(s) = g_1(s) \times g_2(s) \tag{4.5}$$

Here, $\tilde{\tilde{g}}(s)$ is a Hurwitz polynomial of degree $\tilde{n}$ where $\tilde{n} = \tilde{n}_1 + \tilde{n}_2$ and must therefore always be odd.

Also, from Eqns. (4.1)-(4.4) it is clear that:

$$\tilde{\tilde{h}}(s) = \frac{g_1(-s)g_2(s) + g_1(s)g_2(-s)}{2} \tag{4.6}$$

and

$$\tilde{\tilde{f}}(s) = \frac{g_1(s)g_2(-s) + g_1(-s)g_2(s)}{2} \tag{4.7}$$

i.e. $\tilde{\tilde{h}}(s)$ and $\tilde{\tilde{f}}(s)$ are even and odd degree polynomials, respectively.

It is known that the transfer functions are related at the real frequencies $s = j\Omega$ by the Feldkeller equation:

$$|S_{11}(j\Omega)|^2 + |S_{21}(j\Omega)|^2 = 1 \tag{4.8}$$

4.2 Synthesis of FRM Digital Filters Using Cascaded Allpass Networks

As in Section 3.1, the FRM IIR digital filter has an overall transfer function given by:

$$H(z) = H_a(z^M)F_0(z) + H_b(z^M)F_1(z) \tag{4.9}$$

where H_a and H_b are a pair of power complementary filters given by:

$$H_a(z) = \frac{G_0(z) + G_1(z)}{2} \tag{4.10}$$

$$H_b(z) = \frac{G_0(z) - G_1(z)}{2} \tag{4.11}$$

Here, $H_a(z)$ is chosen to have an odd-order N_{IIR}, so that $H_a(z)$ and $H_b(z)$ can be realized as an addition and subtraction, respectively, of two allpass digital networks $G_0(z)$ and $G_1(z)$. This is done by cyclically distributing the poles of $H_a(z)$ between $G_0(z)$ and $G_1(z)$. Further, the allpass transfer functions $G_0(z)$ and $G_1(z)$ can be expressed in the form of products of first-order and second-order rational functions [9, 8]

$$G_0(z) = z \prod_{1}^{[(n+3)/4]} \frac{\beta_i + \alpha_i(1+\beta_i)z^{-1} + z^{-2}}{1 + \alpha_i(1+\beta_i)z^{-1} + \beta_i z^{-2}} \tag{4.12}$$

$$G_1(z) = \prod_{[(n+7)/4]}^{[(n+1)/2]} \frac{\beta_i + \alpha_i(1+\beta_i)z^{-1} + z^{-2}}{1 + \alpha_i(1+\beta_i)z^{-1} + \beta_i z^{-2}} \tag{4.13}$$

Eqns. (4.12) and (4.13) represent G_0 and G_1 as a cascade realization of allpass networks in terms of coefficients α_i and β_i. If the poles of the prototype filter are given by

$$z_i = r_i e^{\pm j\theta_i} \tag{4.14}$$

then the coefficients α_i and β_i are determined as [8]

$$\beta_1 = 0 \tag{4.15}$$

$$\alpha_1 = -r_1 \tag{4.16}$$

$$\beta_i = r_i^2 \tag{4.17}$$

$$\alpha_i = -2\frac{r_i \cos(\theta_i)}{1 + r_i^2} \qquad i > 1 \tag{4.18}$$

The block diagram in Fig. 4.2 shows the FRM digital filter with the IIR interpolation digital subfilters realized as a parallel combination of two allpass networks.

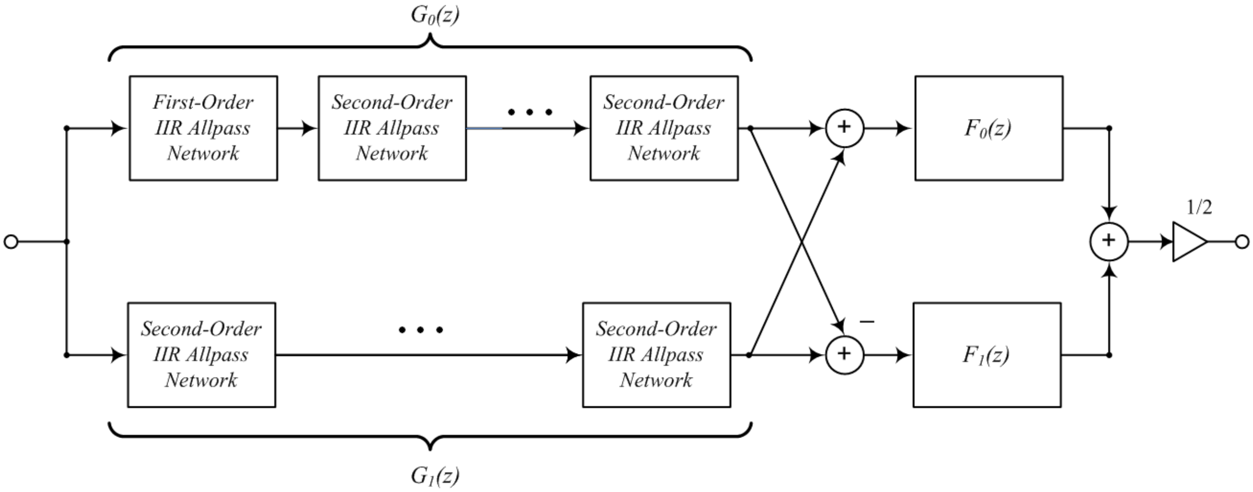
G0(z)
First-Order IIR Allpass Network
Second-Order IIR Allpass Network
Second-Order IIR Allpass Network
+
F0(z)
1/2
+
Second-Order IIR Allpass Network
Second-Order IIR Allpass Network
−
+
F1(z)
G1(z)

Figure 4.2: Diagram of the FRM Digital Filter Using Cascaded Allpass Networks

The most commonly used first and second-order allpass networks using lattice WDF design technique are shown in Fig. 4.3 and Fig. 4.4, respectively.

4.3 Using EMQF to Realize IIR Interpolation Filters with Reduced Number of Multipliers

Since the interpolation digital subfilters $H_a(z)$ and $H_b(z)$ are power complementary, in order to accommodate both Case I and Case II FRM digital filter a special class of elliptic filters called EMQF is used (similar to the case of realization using bilinear-LDI design technique). This keeps the interpolation digital subfilters equal-ripple in both passband and stopband regions. Given a squared passband and stopband

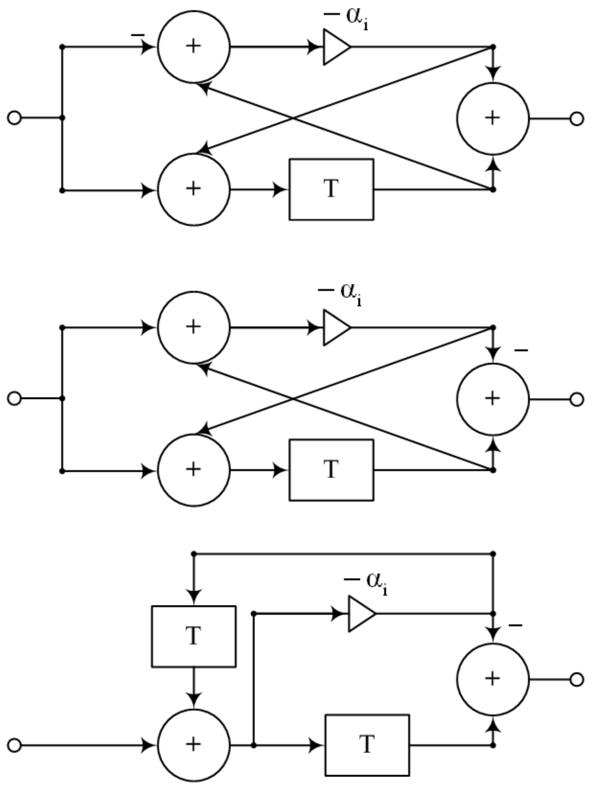

Figure 4.3: Various First-Order Allpass Lattice WDFs

tolerance of δ_p and δ_a, respectively, for an EMFQ filter, the passband ripple R_p and minimum stopband attenuation R_a are related by Eqns. (3.9-3.10). The required passband and stopband edge frequencies for $H_a(z)$ can be determined using design specifications along with Table 2.1.

EMQF analog filters have the important property that their s-plane poles are distributed around the circle $|s| = \sqrt{\Omega_a}$ [25], where Ω_a is the normalized analog stopband edge (with respect to the analog passband edge) given by:

$$\sqrt{\Omega_a} = \sqrt{\frac{\tan(\pi f_a)}{\tan(\pi f_p)}} \tag{4.19}$$

where f_p and f_a are passband and stopband edge frequencies, respectively, of the EMQF digital filter. Also, the frequency at which the digital filter has an attenuation of 3 dB, $f_{3\mathrm{dB}}$, corresponds in the analog filter domain to the frequency $\sqrt{\Omega_a}$ and can be determined directly from the relationship:

$$\tan^2(\pi f_{3\mathrm{dB}}) = \tan(\pi f_p)\tan(\pi f_a) \tag{4.20}$$

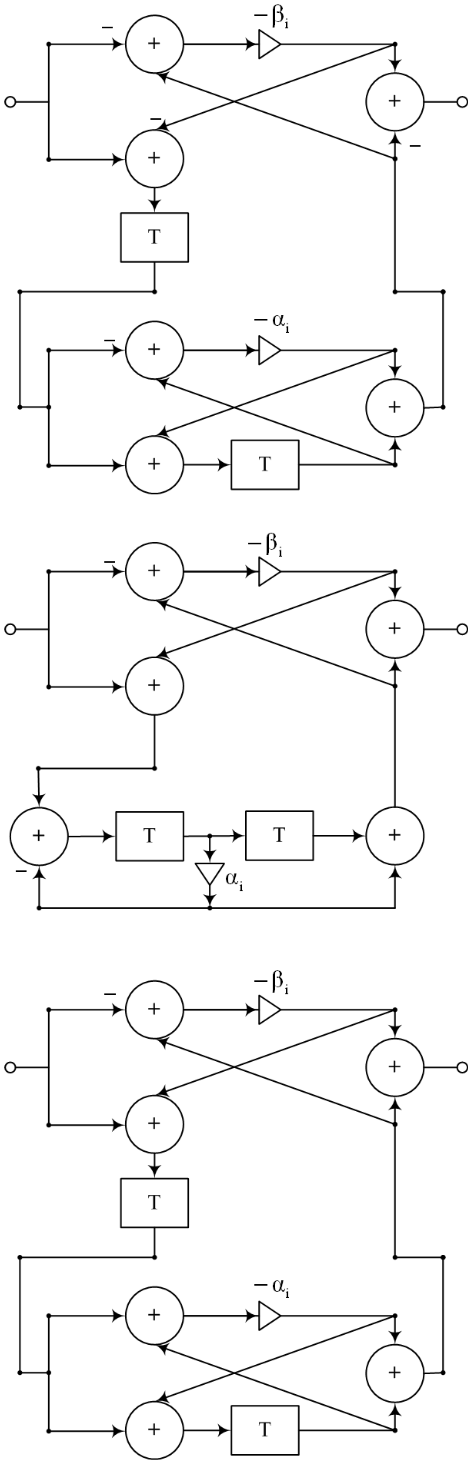

Figure 4.4: Various Second-Order Allpass Lattice WDFs

By using the bilinear transform given by Eqn. (3.14), the analog poles around the circle $|s| = \sqrt{\Omega_a}$ are mapped into a circle orthogonal to the unit circle and centered on the real axis of the z-plane, as shown in Fig. 4.5.

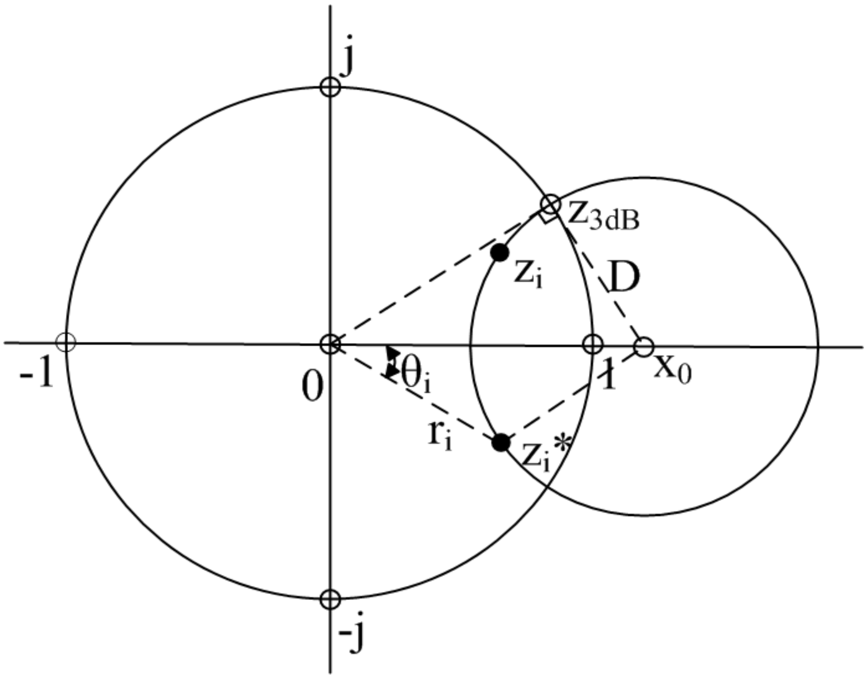

Figure 4.5: Pole Distribution in z-plane

The digital piles z_i are inside the unit circle and are distributed around the circle (x_0, D), and intersects the unit circle at the frequency where the IIR filter has a 3 dB attenuation (z_{3dB}). Using this fact, it is evident that x_0 is related to the 3 dB attenuation frequency, $f_{3\text{dB}}$, by [9]

$$x_0 = \frac{1}{\cos(2\pi f_{3\text{dB}})} = \frac{1 + \tan^2(\pi f_{3\text{dB}})}{1 - \tan^2(\pi f_{3\text{dB}})} \tag{4.21}$$

Also, the poles r_i can be related to x_0 in accordance with (see Fig. 4.5)

$$2r_i \cos(\theta_i) = \frac{1 + r_i^2}{x_0} \tag{4.22}$$

Comparing (4.22) and (4.18) we get

$$\begin{aligned}\alpha_1 &= -x_0\left(1-\sqrt{1-\frac{1}{x_0^2}}\right)\\ &= -\frac{1-\tan(\pi f_{\text{3dB}})}{1+\tan(\pi f_{\text{3dB}})} \qquad (4.23)\\ \alpha_i &= -\frac{1}{x_0}, \qquad i>1\\ &= -\cos(2\pi f_{\text{3dB}}) \qquad (4.24)\end{aligned}$$

Thus, all α_i can be determined solely by f_{3dB}, and all α_i of second-order sections ($i > 1$) have the same value.

In summary, the poles of allpass functions $G_0(z)$ and $G_1(z)$ are determined by cyclically distributing the poles of the interpolation digital subfilter $H_a(z)$ [8], which is an odd-order EMQF filter. With the poles of $G_0(z)$ and $G_1(z)$ determined, α_i and β_i are determined. Further, α_i and β_i are related to f_{3dB} as shown in Table 4.1 [26]. By a judicious selection of the value of f_{3dB}, all α_i of second-order allpass sections can be represented by a single, easily implementable multiplier coefficient. Lattice WDF first and second-order sections can then be made, as shown in respectively Fig. 4.3 and Fig. 4.4, and implemented in cascade fashion to form the parallel allpass branches $G_0(z)$ and $G_1(z)$.

Table 4.1: Allpass Network Coefficients for Lattice WDF Design

Pole (i)	Coefficients (α, β)
$i = 1$ real pole	$\alpha_1 = -\frac{1-\tan(\pi f_{\text{3dB}})}{1+\tan(\pi f_{\text{3dB}})}$ $\beta_1 = 0$
$2 \leq i \leq (N_{IIR} + 1)/2$ complex conjugate pair	$\alpha_i = -\cos(2\pi f_{\text{3dB}})$ $\beta_i = r_i^2$

It is important to note that the lattice WDF digital filter design technique does not suffer from the BIBO stability constraints required for the bilinear-LDI design technique described in Section 3.4. Therefore, only two CSD LUTs are required: one for the interpolation digital subfilters that ranges from (−1,1), and another one for the FIR masking digital subfilters that also has ranges from (−1,1).

Some important features of the resulting interpolation digital subfilters are as follows:

- The absolute value of the constant α_i for $i > 1$ ($|\alpha_i| = |1/x_0|$) is always smaller than unity because the centre of the poles loci x_0 is always placed outside the unit circle.

- The distance of the poles from the unit circle is nearly maximal compared to elliptical filters having different passband ripple. This provides lower magnitude sensitivity to coefficients β_i.

- Relations given in Eqns. (4.23) and (4.24) are independent of the filter order N_{IIR}. Also, half the multipliers α_i for $i > 1$ obtained have the same value that, by the proper choice of f_{3dB}, can be adjusted to be implemented using a minimal number of shift-and-add operations, without any unwanted influence on the other filter characteristics.

4.4 Design Methodology

The design methodology for the proposed DCGA optimization of BIBO stable lattice WDF based FRM digital filters over the CSD multiplier coefficient space can be summarized as follows:

1. *Designing the IIR interpolation digital subfilter*: Using the passband edge frequency ω_p and stopband edge frequency ω_a, and the expressions for boundary frequencies given in Table 2.1, we can determine the filter case and calculate the approximate passband edge $\tilde{\theta}$ and stopband edge $\tilde{\phi}$ of the digital interpolation lowpass subfilter $H(e^{j\omega})$ in exactly the same manner as described in Section 3.6. The passband ripple R_p and stopband ripple R_a of the interpolation digital subfilter $H_a(e^{j\omega})$ is also calculated in the same fashion as described in Section 3.6, in order to obtain a filter having an EMQF transfer function. The interpolation digital subfilter order N_{IIR} is then determined using R_p, R_a, $\tilde{\theta}$ and $\tilde{\phi}$. N_{IIR} must be rounded to the nearest larger odd integer so that it can be implement by a parallel combination of two allpass networks.

Using $\tilde{\theta}$ and $\tilde{\phi}$, the approximate 3 dB frequency of the interpolation digital subfilter $H(e^{j\omega})$, $\tilde{f}_{3\text{dB}}$, by (see Eqn. (4.20)):

$$\tilde{f}_{3\text{dB}} = \frac{1}{\pi} \arctan\left(\sqrt{\tan\left(\tilde{\theta}/2\right) \tan\left(\tilde{\phi}/2\right)}\right) \tag{4.25}$$

Subsequently, $\tilde{f}_{3\text{dB}}$ is used to get the corresponding infinite-precision value of α_i $(i > 1)$ by (see Eqn. (4.24)):

$$\alpha_i = -\cos(2\pi \tilde{f}_{3\text{dB}}) \tag{4.26}$$

The infinite-precision value of α_i is approximated to its the nearest *larger* finite-wordlength CSD value $\hat{\alpha}_i$, using the CSD LUTs. A larger value is chosen so that that $f_{3\text{dB}}$ is increased, thus ensuring that the passband edge frequency specification ω_p is not violated. The finite-wordlength value of $\hat{\alpha}_i$ is then used to calculate an adjusted value of $\tilde{f}_{3\text{dB}}$:

$$\hat{\tilde{f}}_{3\text{dB}} = \frac{1}{2\pi} \arccos(-\hat{\alpha}_i) \tag{4.27}$$

Next, the adjusted 3 dB frequency $\hat{\tilde{f}}_{3\text{dB}}$ is used to calculate the final passband and stopband edge frequencies, θ and ϕ, respectively, of the interpolation digital subfilter $H_a(z)$. As mentioned in Section 3.6, the ratio of the analog passband edge θ_A and stopband edge ϕ_A is a constant k [29] given by Eqns. (3.48-3.52). This relationship is used to find θ and ϕ as follows:

$$\theta = 2\arctan(\tan(\pi \hat{\tilde{f}}_{3\text{dB}} \sqrt{k}) \tag{4.28}$$

$$\phi = 2\arctan(\tan(\pi \hat{\tilde{f}}_{3\text{dB}} / \sqrt{k}) \tag{4.29}$$

With the order N_{IIR}, passband ripple R_p and stopband ripple R_a, and passband edge θ of the interpolation digital subfilter $H_a(e^{j\omega})$ all determined, the transfer function coefficients of $H_a(z)$ can be obtained, and subsequently the values of coefficients α_i and β_i. Also, using given ripple specifications along with the boundary frequencies described in Table 2.1, one can determine the transfer function of the FIR masking filters $F_0(e^{j\omega})$ and $F_1(e^{j\omega})$.

A salient difference between the FRM digital filters designed using bilinear-LDI and lattice WDF design techniques is that the FRM digital filter using bilinear-LDI design technique matches the design specification for passband edge ω_p exactly, while the FRM digital filter using lattice WDF design technique mostly has a passband edge frequency that is marginally greater than ω_p.

2. *Generation of seed FRM digital filter chromosome*: The seed FRM digital filter chromosome is formed in an ordered manner by concatenating:

    - A block of $B_{_{IIR}}$ bits serving as the binary index into the corresponding CSD LUT for each multiplier coefficient $\hat{\alpha}_1$ and $\hat{\beta}_i$ in the lattice WDF allpass digital networks $G_0(z)$ and $G_1(z)$. Note that $\hat{\alpha}_i$ is not a optimization variable for DCGA algoritm.
    - A block of $B_{_{FIR}}$ bits serving as the binary index into the corresponding CSD LUT for each multiplier coefficient in the FIR masking digital subfilters $F_0(z)$ and $F_1(z)$.

The remainder of the DCGA optimization proceeds in exactly the same fashion as described in Section 3.6 and is therefore mentioned only briefly as follows. Once the seed chromosome is constructed, the initial population pool generated by random perturbations of the seed chromosome. The initial population pool is enlarged by the inclusion of offspring chromosomes produced by the crossover operation. The enlarged population then undergoes the mutation operation with a small fixed probability. This is followed by the formation of the next generation population pool through the fitness evaluation and selection of the enlarged population pool. The cost-function used to evaluate fitness is exactly as described in the case of the FRM digital filter using the bilinear-LDI design technique.

In Section 5.4and Section 5.5 examples illustrating the DCGA optimization of the design procedure described above are presented. Also, comparisons are made between the lattice WDF and bilinear-LDI digital filter design approach.

In Section 5.4, and Section 5.5 examples illustrating the DCGA optimization of the design procedure described above are presented. Comparisons are made between the FRM digital filters based on bilinear-LDI and lattice WDF design approach.

4.5 Summary

This section discusses another design technique for FRM digital filters with IIR interpolation digital subfilters. The interpolation subfilters are efficiently realized using the addition and subtraction of two allpass networks that are implemented using the lattice WDF design technique. The salient feature of this technique is that half the most sensitive coefficients of the interpolation digital subfilter are of the same value. By judiciously selecting the edge frequencies of the interpolation digital subfilter, these exact value of these coefficients can be implemented using a minimal number of shift-and-add operations. The design procedure for the seed chromosome of a FRM digital filter utilizing the lattice WDF design technique is described in detailed.

Chapter 5

Application Examples

In the previous two chapters, the design and DCGA optimization of FRM digital filters using bilinear-LDI and lattice WDF design approach were discussed. This chapter presents examples of DCGA optimization of these FRM digital filters, together with the details on how all the constituent subfilters are designed.

Two different sets of lowpass FRM digital filter specifications are used, with the first set of specifications resulting in a fifth-order interpolation digital subfilter $H_a(z)$, and the second set resulting in a seventh-order subfilter $H_a(z)$. Each of these design specifications are met using the bilinear-LDI and lattice WDF design approach, and relevant comparisons are made. In the case of the bilinear-LDI design approach, the LUT-based techniques that uses both the slack variables as well as the worst-case LUT approach are considered.

This chapter proceeds as follows. Section 5.1 contains an example of a FRM digital filter incorporating a fifth-order interpolation digital subfilter $H_a(z)$ realized using the bilinear-LDI design approach. In this example, the DCGA optimization technique employs slack variables. Section 5.2 contains an example of a FRM digital filter incorporating a seventh-order $H_a(z)$. It also makes use of the bilinear-LDI design approach and slack variables. Section 5.3 illustrates the design of a FRM digital filter with the same design specifications as in Section 5.2, but this time the worst-case LUT approach is employed. Section 5.4 contains an example of a FRM digital filter incorporating a fifth-order $H_a(z)$ realized using the lattice WDF design approach. The design specifications in this example match those in Section 5.1. Section 5.5 also contains an example that employs the lattice WDF design approach, but with a seventh-order $H_a(z)$ (design specifications in this example match those in Section 5.2 and Section 5.3). Finally, a summary of the conclusions drawn is presented in Section 5.6.

Note that all frequency values in this chapter are normalized so as to maintain a Nyquist sampling frequency of 2π radians. Also, FIR masking digital subfilters $F_0(z)$ and $F_1(z)$ are always made to have equal orders (by zero-padding) to ensure that they maintain similar phase characteristics.

5.1 Application Example 1

This section is concerned with the design of a lowpass FRM digital filter satisfying the specifications given in Table 5.1, by using the bilinear-LDI design technique. The LUT scheme utilizing slack variables is employed for the DCGA optimization in this example.

Table 5.1: Example 1: Design Specifications

Maximum Passband Ripple A_p	0.05[dB]
Minimum Stopband Loss A_a	38[dB]
Passband-Edge Normalized Frequency ω_p	0.20π[Rad]
Stopband-Edge Normalized Frequency ω_a	$< 0.24\pi$[Rad]
Normalized Sampling Period T	1[s]
Interpolation Factor Range M	$6 - 15$

The first step in the design of the FRM digital filter is to determine the optimal value of the interpolation factor M. Here, optimal M refers to the value of interpolation factor in the range $6 < M < 15$ that results in the lowest order of the FIR masking digital subfilters $F_0(z)$ and $F_1(z)$. Using the above specifications and Table 2.1, we can calculate the approximate passband edge frequency $\tilde{\theta}$ and stopband edge $\tilde{\phi}$ frequency of the digital interpolation subfilter $H_a(e^{j\omega})$ for every value of interpolation factor $6 < M < 15$. Further, by using Eqns. (3.46-3.47), the optimal interpolation factor M can be determined as $M = 7$. Also, the value of K in Table 2.1 turns out to be 1, and the overall FRM digital filter turns out to be of the type Case II.

Once M is fixed at its optimal value, and the corresponding passband edge $\tilde{\theta}$

and stopband edge $\tilde{\phi}$ are calculated from Table 2.1, the requirements for the digital subfilters have to be obtained. The order of interpolation digital subfilter $H_a(z)$, $N_{_{IIR}}$, is obtained using standard MATLAB routine "ellipord". If $N_{_{IIR}}$ is even it is rounded to the first larger odd integer value. In this way $N_{_{IIR}} = 5$. Then, Eqns. (3.48-3.52) are used (as described in Section 3.6) to get the exact edge frequencies θ and ϕ of filter $H_a(z)$. This is followed by using Table 2.1 to get the corresponding edge frequencies of subfilters $F_0(z)$ and $F_1(z)$. The resulting requirements for this example are given in Table 5.2.

Table 5.2: Example 1: Required Subfilter Requirements

Subfilter	Order	R_p [dB]	R_a [dB]	Passband Edge	Stopband Edge
$H_a(e^{j\omega})$	5	0.0007	38	1.1912	1.8850
$F_0(e^{j\omega})$	37	0.05	38	0.2693	0.6283
$F_1(e^{j\omega})$	39	0.05	38	0.7274	1.0678

$F_0(z)$ has to be zero-padded by one zero on each end. The overall FRM digital filter $H(z)$ has a normalized passband edge $\omega_p = 0.2000\pi$, and a normalized stopband edge $\omega_a = 0.2315\pi$.

The values of the inductances and capacitances of allpass networks $G_{0\,\mathrm{or}\,1}(z)$, $L_{0\,\mathrm{or}\,1,p}$ and $C_{0\,\mathrm{or}\,1,p}$, the values of the precomensated inductances and capacitances $L^{'}_{0\,\mathrm{or}\,1,p}$ and $C^{'}_{0\,\mathrm{or}\,1,p}$, and the values of the multiplier coefficients $m_{L_{0\,\mathrm{or}\,1,p}}$ and $m_{C_{0\,\mathrm{or}\,1,p}}$ can be obtained as summarized in Table 5.3.

Table 5.3: Example 1: Analog Component Values and Corresponding Digital Multiplier Values

Variables	Values
$C_{0,1}$; $C^{'}_{0,1}$; $m_{C_{0,1}}$	0.3828[F] ; 1.0123[F] ; 0.9878
$C_{0,2}$; $C^{'}_{0,2}$; $m_{C_{0,2}}$	0.3173[F] ; 0.1878[F] ; 5.3253
$L_{0,2}$; $L^{'}_{0,2}$; $m_{L_{0,2}}$	1.1423[H] ; 3.2615[H] ; 0.3066
$C_{1,1}$; $C^{'}_{1,1}$; $m_{C_{1,1}}$	0.4081[F] ; 1.2888[F] ; 0.7759
$L_{1,1}$; $L^{'}_{1,1}$; $m_{L_{1,1}}$	0.6567[H] ; 0.6567[H] ; 1.5227

The next step is to construct the LUTs required for DCGA optimization. A set of five CSD LUTs are required, four LUTs with entries belonging to $CSD(11,3)$

(with 7-bit fractional part) for the multiplier coefficients $m_{C_{0,2}}$, $m_{L_{0,2}}$, $m_{C_{1,1}}$, and $m_{L_{1,1}}$ constituent in the interpolation digital subfilters $H_a(z)$ and $H_b(z)$, and one LUT with entries belonging to $CSD(11,3)$ (with 10-bit fractional part) for all the multiplier coefficients constituent in the masking digital subfilters $F_0(z)$ and $F_1(z)$. An 11-bit finite-wordlength slack-variable $\hat{\epsilon}$ is used.

An initial population size of $N = 500$ was chosen for the ensuing DCGA optimization. The remaining DCGA optimization parameters are given by

- $c = 0.6$, $\alpha = 0.4$
- $p_{_M} = 0.3$
- $p_{_F} = 0.5$
- $\zeta = 0.4$

The DCGA parameter used in [11] showed slow convergence in empirical computational investigations. Therefore, smaller/larger values of c/α were used, resulting in a faster convergence speed. The value of ζ was also determined through empirical investigations. The ζ value that provided a good balance between magnitude-frequency and group-delay frequency response optimization was selected. [1]

The DCGA optimization converged in around 150 generations to a desired FRM digital filter with the CSD multiplier coefficients $\hat{m}_{L_{0\,\mathrm{or}\,1},p}$ and $\hat{m}_{C_{0\,\mathrm{or}\,1},p}$ as given in Table 5.4 (where the over-bared digit $\bar{1}$ is used to represent -1).

Table 5.4: Example 1: Finite-Precision Multiplier Coefficient Values After DCGA Optimization

Multiplier	CSD Representation	Decimal Value
$\hat{m}_{C_{0,1}}$	$0001.00000\bar{1}0$	0.9844
$\hat{m}_{C_{0,2}}$	0101.0100000	5.2500
$\hat{m}_{L_{0,2}}$	0000.0101001	0.3203
$\hat{m}_{C_{1,1}}$	$0001.0\bar{1}01000$	0.8125
$\hat{m}_{L_{1,1}}$	$0010.\bar{1}000000$	1.5000

Note should be made that the FIR masking digital subfilters CSD multiplier coefficients $\hat{m}_{_{FIR}}$ are also optimized during DCGA optimization.

[1] These parameters are kept constant in all the remaining examples.

The analysis of the seed FRM digital filter before DCGA optimization and optimized FRM digital filter after DCGA optimization revealed frequency response characteristics as given in Table 5.5.

Table 5.5: Example 1: Frequency-Response Analysis Before and After DCGA Optimization

Frequency-response characteristic	Before DCGA	After DCGA
Maximum Passband Ripple A_p	0.1286[dB]	0.0440[dB]
Minimum Stopband Loss A_a	31.7879[dB]	38.9543[dB]
Maximum Normalized Group-Delay τ	48.7	46.0

The magnitude and group-delay frequency-responses of the interpolation digital subfilters before and after the DCGA optimization are obtained as shown in Fig. 5.1 and Fig. 5.2.

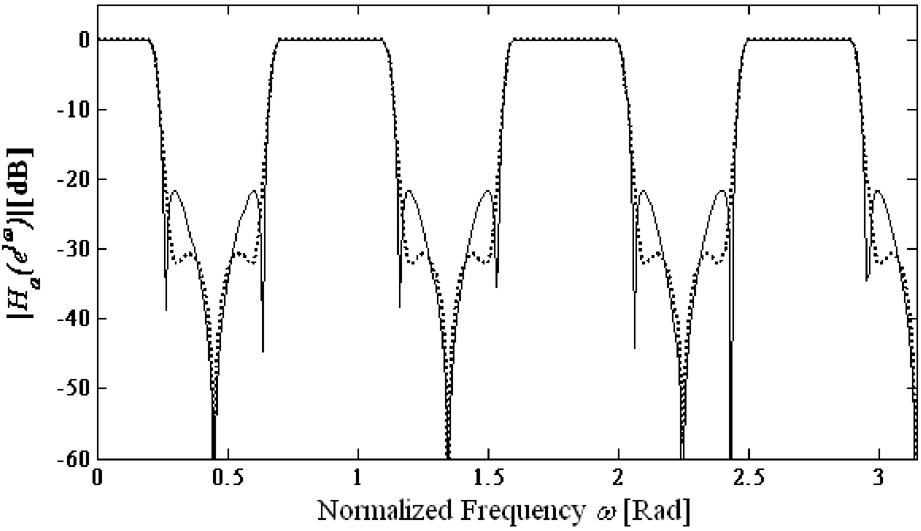

Figure 5.1: Example 1: $H_a(e^{j\omega})$ Magnitude Response: Non-optimized (dash) / Optimized (solid)

The magnitude and group-delay frequency-responses of the FIR digital subfilters before and after the DCGA optimization are obtained as shown in Fig. 5.3 and Fig. 5.4

The magnitude and group-delay frequency-responses of the FRM digital filter before and after the DCGA optimization are obtained as shown in Fig. 5.5 and Fig.

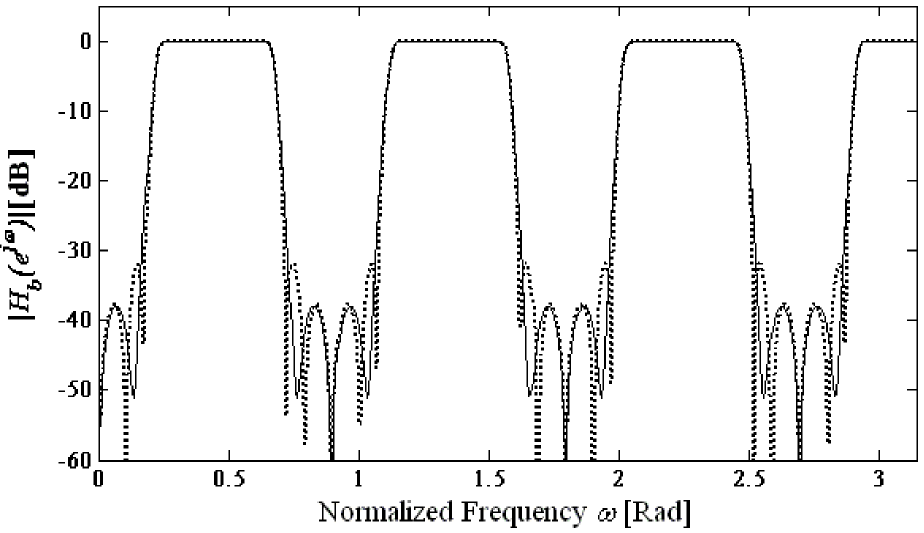

Figure 5.2: Example 1: $H_b(e^{j\omega})$ Magnitude Response: Non-optimized (dash) / Optimized (solid)

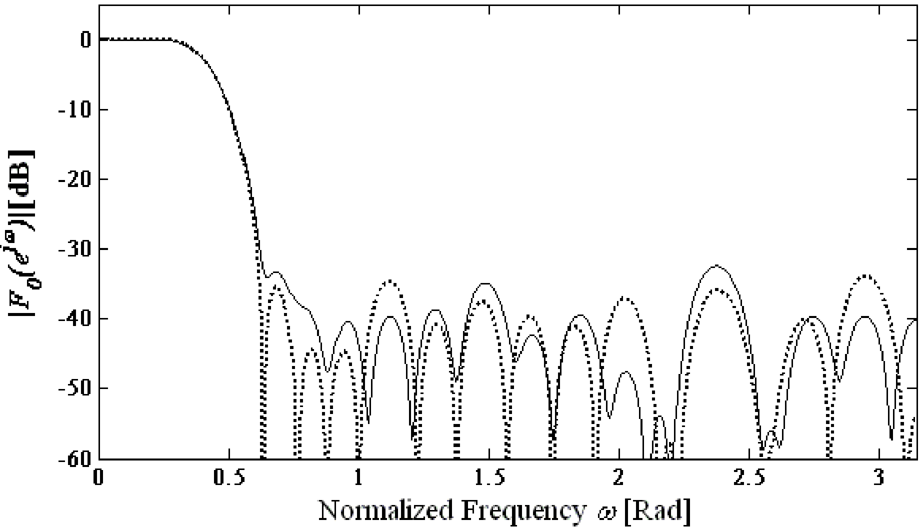

Figure 5.3: Example 1: $F_0(e^{j\omega})$ Magnitude Response: Non-optimized (dash) / Optimized (solid)

5.6

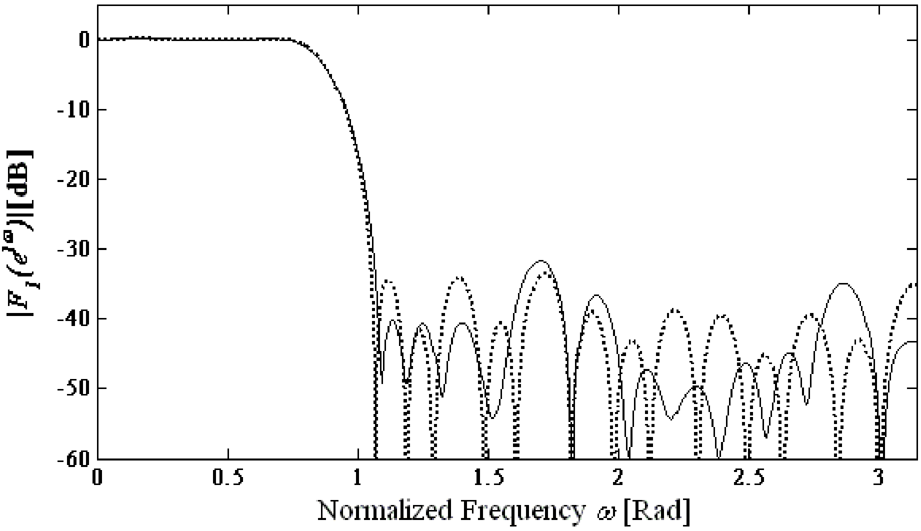

Figure 5.4: Example 1: $F_1(e^{j\omega})$ Magnitude Response: Non-optimized (dash) / Optimized (solid)

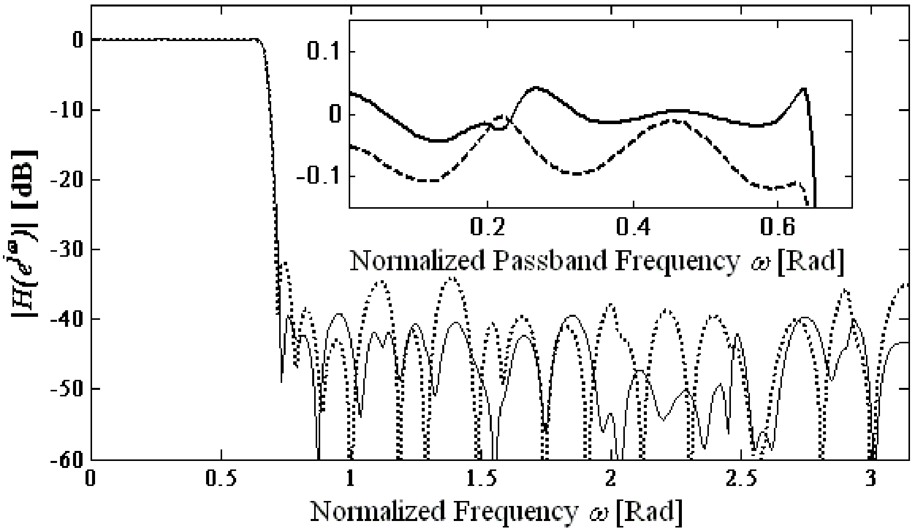

Figure 5.5: Example 1: $H(e^{j\omega})$ Magnitude Response: Non-optimized (dash) / Optimized (solid)

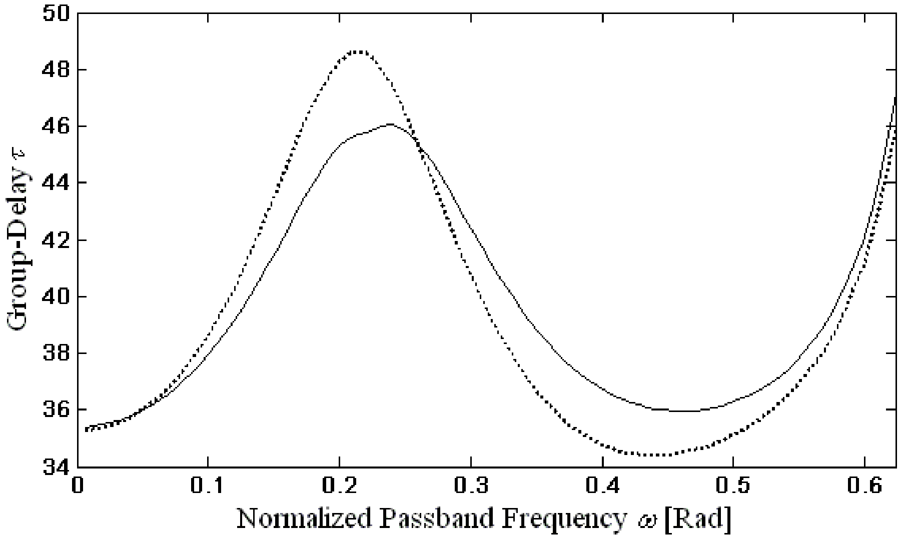

Figure 5.6: Example 1: $H(e^{j\omega})$ Group Delay: Non-optimized (dash) / Optimized (solid)

5.2 Application Example 2

This section is concerned with the design of a lowpass FRM digital filter satisfying the specifications given in Table 5.6. These specifications require a seventh-order $H_a(z)$. The rest of the design approach is similar to Section 5.1.

Table 5.6: Example 2: Design Specifications

Maximum Passband Ripple A_p	0.1[dB]
Minimum Stopband Loss A_a	40[dB]
Passband-Edge Normalized Frequency ω_p	0.60π[Rad]
Maximum Stopband-Edge Normalized Frequency ω_a	$< 0.62\pi$[Rad]
Normalized Sampling Period T	1[s]
Interpolation Factor Range M	$6 - 15$

Once again, the first step is to determine the optimal value of M so as to get minimal order FIR masking filters $F_0(z)$ and $F_1(z)$. In this example $M = 6$ (see Table 2.1 and Eqns. (3.46-3.47)). Also, the value of K from Table 2.1 turns out to be 2 and the overall FRM digital filter is of the type Case II.

The next step is to obtain the value of N_{IIR}. Similar to the previous section, this is achieved by using MATLAB once the approximate edge frequencies $\tilde{\theta}$ and $\tilde{\phi}$ have been calculated. $N_{_{IIR}}$ turns out to be seven. Next, Eqns. (3.48-3.52) are used to calculate the exact edge frequencies θ and ϕ . The resulting requirements are given in Table 5.7.

Table 5.7: Example 2: Required Subfilter Requirements

Subfilter	Order	R_p [dB]	R_a [dB]	Passband Edge	Stopband Edge
$H_a(e^{j\omega})$	7	0.0004	40	0.9811	1.2566
$F_0(e^{j\omega})$	20	0.1	40	1.2566	1.8850
$F_1(e^{j\omega})$	38	0.1	40	1.9309	2.2579

$F_0(z)$ has to be zero-padded by nine zeros on each end. The overall FRM digital filter $H(z)$ has a normalized passband edge $\omega_p = 0.6000\pi$ and a normalized stopband edge $\omega_a = 0.6146\pi$.

The values of the inductances and capacitances of allpass networks $G_{0\,\mathrm{or}\,1}(z)$, $L_{0\,\mathrm{or}\,1,p}$ and $C_{0\,\mathrm{or}\,1,p}$, the values of the precomensated inductances and capacitances $L^{'}_{0\,\mathrm{or}\,1,p}$ and $C^{'}_{0\,\mathrm{or}\,1,p}$, and the values of the multiplier coefficients $m_{L_{0\,\mathrm{or}\,1},p}$ and $m_{C_{0\,\mathrm{or}\,1},p}$ can be obtained as summarized in Table 5.8.

Table 5.8: Example 2: Analog Component Values and Corresponding Digital Multiplier Values

Variables	Values
$C_{0,1}$; $C^{'}_{0,1}$; $m_{C_{0,1}}$	0.5110[F] ; 1.1595[F] ; 0.8624
$C_{0,2}$; $C^{'}_{0,2}$; $m_{C_{0,2}}$	0.7499[F] ; 0.6014[F] ; 1.6629
$L_{0,2}$; $L^{'}_{0,2}$; $m_{L_{0,2}}$	1.3496[H] ; 2.0987[H] ; 0.4765
$C_{1,1}$; $C^{'}_{1,1}$; $m_{C_{1,1}}$	0.5340[F] ; 1.2712[F] ; 0.7866
$L_{1,1}$; $L^{'}_{1,1}$; $m_{L_{1,1}}$	1.2064[H] ; 1.2064[H] ; 0.8289
$C_{1,2}$; $C^{'}_{1,2}$; $m_{C_{1,2}}$	0.1071[F] ; 0.0772[F] ; 12.9560
$L_{1,2}$; $L^{'}_{1,2}$; $m_{L_{1,2}}$	6.0137[H] ; 6.0137[H] ; 0.0863

The next step is to construct the LUTs required for DCGA optimization. A set of 7 CSD LUTs are required, six LUTs with entries belonging to $CSD(12,3)$ (with 7-bit fractional part) for the multiplier coefficients $m_{C_{0,2}}$, $m_{L_{0,2}}$, $m_{C_{1,1}}$, $m_{L_{1,1}}$, $m_{C_{1,2}}$, and $m_{L_{1,2}}$ constituent in the interpolation digital subfilters $H_a(z)$ and $H_b(z)$, and one LUT with entries belonging to $CSD(11, 3)$ (with 10-bit fractional part) for all the multiplier coefficients constituent in the masking digital subfilters $F_0(z)$ and $F_1(z)$. A 12-bit finite-wordlength slack-variable $\hat{\epsilon}$ is used. A larger wordlength is used in this example than that in to Example 1 since using $CSD(11, 3)$ (with 7-bit fractional part) for the interpolation digital subfilter multiplier coefficients resulted in the DCGA optimization experiencing difficulty in converging to the desired overall FRM digital filter.

The DCGA optimization converged in around 200 generations to a desired FRM digital filter with the CSD multiplier coefficients $\hat{m}_{L_{0\,\mathrm{or}\,1},p}$ and $\hat{m}_{C_{0\,\mathrm{or}\,1},p}$ as given in Table 5.9.

Table 5.9: Example 2: Finite-Precision Multiplier Coefficient Values After DCGA Optimization

Multiplier	CSD Representation	Decimal Value
$\hat{m}_{C_{0,1}}$	$00001.00\bar{1}0010$	0.8906
$\hat{m}_{C_{0,2}}$	$00010.0\bar{1}0\bar{1}000$	1.6875
$\hat{m}_{L_{0,2}}$	$00000.100000\bar{1}$	0.4922
$\hat{m}_{C_{1,1}}$	$00001.0\bar{1}00010$	0.7656
$\hat{m}_{L_{1,1}}$	$00001.00\bar{1}0010$	0.8906
$\hat{m}_{C_{1,2}}$	$10\bar{1}00.0001000$	12.0625
$\hat{m}_{L_{1,2}}$	$00000.0010\bar{1}01$	0.1016

The analysis of the seed FRM digital filter before DCGA optimization and optimized FRM digital filter after DCGA optimization revealed frequency response characteristics as given in Table 5.10.

Table 5.10: Example 2: Frequency-Response Analysis Before and After DCGA Optimization

Frequency-response characteristic	Before DCGA	After DCGA
Maximum Passband Ripple A_p	0.4207[dB]	0.0875[dB]
Minimum Stopband Loss A_a	19.4949[dB]	41.1543[dB]
Maximum Normalized Group-Delay τ	202.7	188.8

The magnitude and group-delay frequency-responses of the FRM digital filter before and after the DCGA optimization are obtained as shown in Fig. 5.7 and Fig.

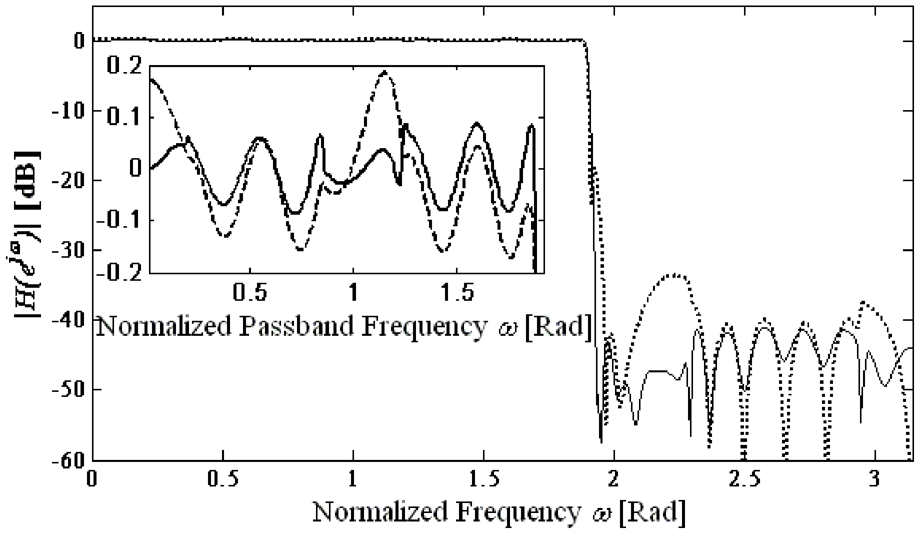

Figure 5.7: Example 2: $H(e^{j\omega})$ Magnitude Response: Non-optimized (dash) / Optimized (solid)

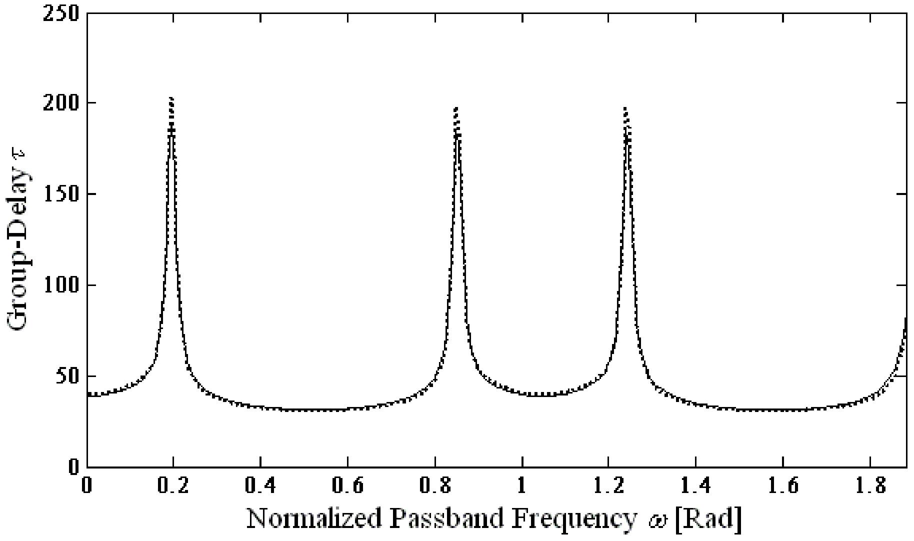

Figure 5.8: Example 2: $H(e^{j\omega})$ Group Delay: Non-optimized (dash) / Optimized (solid)

5.3 Application Example 3

This section is concerned with the design of a lowpass FRM digital filter satisfying the specifications given in Table 5.11. The design approach used in this example employs worst-case LUTs.

Table 5.11: Example 3: Design Specifications

Maximum Passband Ripple A_p	0.1[dB]
Minimum Stopband Loss A_a	40[dB]
Passband-Edge Normalized Frequency ω_p	0.30π[Rad]
Maximum Stopband-Edge Normalized Frequency ω_a	0.31π[Rad]
Normalized Sampling Period T	1[s]
Interpolation Factor M	$6 - 15$

The design specifications here are the same as that in Example 2, and therefore the design procedure matches the previous section save for the construction of LUTs for DCGA optimization.

In this example, a set of eight CSD LUTs are required, seven LUTs with entries belonging to $CSD(12, 3)$ (with 7-bit fractional part) for the multiplier coefficients $m_{C_{0,1}}$, $m_{C_{0,2}}$, $m_{L_{0,2}}$, $m_{C_{1,1}}$, $m_{L_{1,1}}$, $m_{C_{1,2}}$ and $m_{L_{1,2}}$ constituent in the digital allpass networks $G_0(z)$ and $G_1(z)$, and one LUT with entries belonging to $CSD(11, 3)$ (with 10-bit fractional part) for all the multiplier coefficients constituent in the masking digital subfilters $F_0(z)$ and $F_1(z)$.

The DCGA optimization converged in around 200 generations to a desired FRM digital filter with the CSD multiplier coefficients $\hat{m}_{L_{0\,\mathrm{or}\,1},p}$ and $\hat{m}_{C_{0\,\mathrm{or}\,1},p}$ as given in Table 5.12.

The analysis of the seed FRM digital filter before DCGA optimization and optimized FRM digital filter after DCGA optimization revealed frequency response characteristics as given in Table 5.13.

The magnitude and group-delay frequency-responses of the FRM digital filter before and after the DCGA optimization are obtained as shown in Fig. 5.9 and Fig. 5.10

A comparison of fitness function values over 200 generations (averaged over ten iterations) between the LUT scheme using slack variables in Example 2 and the

Table 5.12: Example 3: Finite-Precision Multiplier Coefficient Values After DCGA Optimization

Multiplier	CSD Representation	Decimal Value
$\hat{m}_{C_{0,1}}$	$00001.00\bar{1}00\bar{1}0$	0.8594
$\hat{m}_{C_{0,2}}$	$00010.0\bar{1}0\bar{1}000$	1.6875
$\hat{m}_{L_{0,2}}$	$00000.1000\bar{1}00$	0.4688
$\hat{m}_{C_{1,1}}$	$00001.0\bar{1}01000$	0.8125
$\hat{m}_{L_{1,1}}$	$00001.0\bar{1}01000$	0.8125
$\hat{m}_{C_{1,2}}$	$10\bar{1}01.0000000$	13.0000
$\hat{m}_{L_{1,2}}$	$00000.0010\bar{1}00$	0.0938

Table 5.13: Example 3: Frequency-Response Analysis Before and After DCGA Optimization

Frequency-response characteristic	Before DCGA	After DCGA
Maximum Passband Ripple A_p	0.3089[dB]	0.0723[dB]
Minimum Stopband Loss A_a	19.3509[dB]	42.7777[dB]
Maximum Normalized Group-Delay τ	203.7653	188.4375

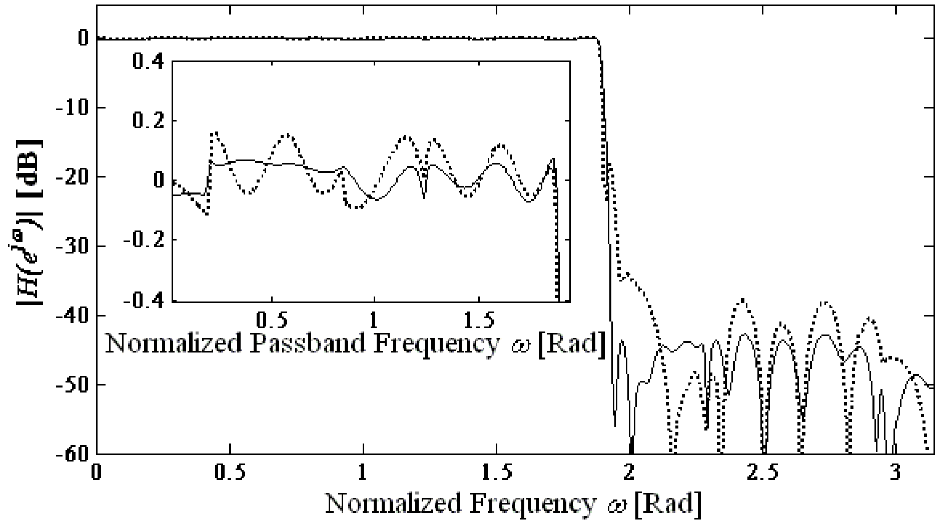

Figure 5.9: Example 3: $H(e^{j\omega})$ Magnitude Response: Non-optimized (dash) / Optimized (solid)

worst-case LUT scheme of this example is shown in Fig. 5.11. As seen, worst-case

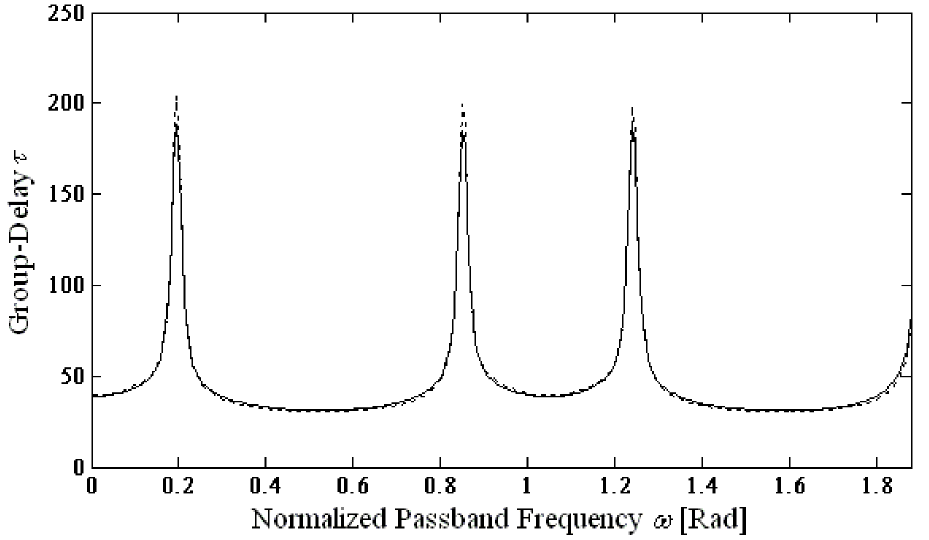

Figure 5.10: Example 3: $H(e^{j\omega})$ Group Delay: Non-optimized (dash) / Optimized (solid)

LUT formation technique results in a significant improvement in convergence speed.

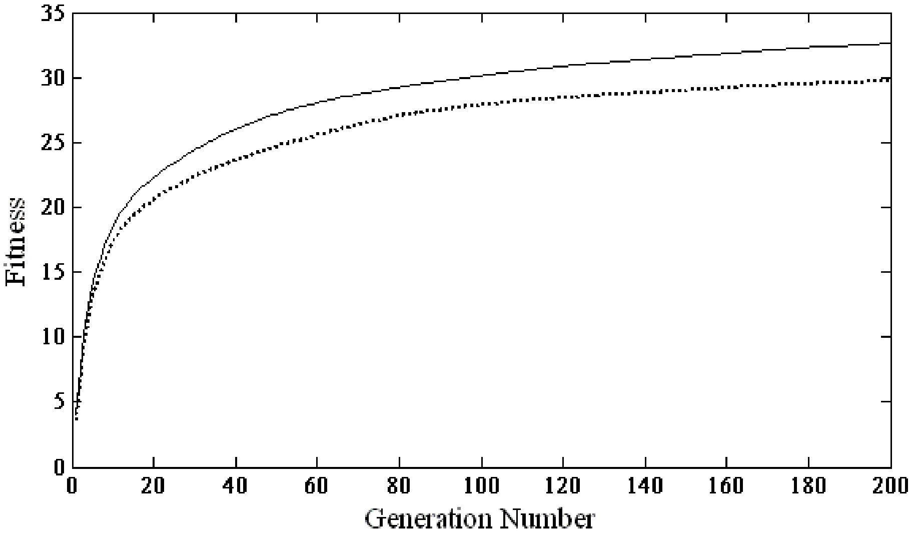

Figure 5.11: Averaged Fitness Function Values, with Slack-Variables(dash), and for Proposed Technique (without Slack-Variables)(solid)

5.4 Application Example 4

This section is concerned with the design of a lowpass FRM digital filter satisfying the specifications given in Table 5.14. Lattice WDF design technique is to be used here. The design specifications in this example match those of Section 5.1.

Table 5.14: Example 4: Design Specifications

Maximum Passband Ripple A_p	0.05[dB]
Minimum Stopband Loss A_a	38[dB]
Passband-Edge Normalized Frequency ω_p	$> 0.20\pi$[Rad]
Maximum Stopband-Edge Normalized Frequency ω_a	$< 0.24\pi$[Rad]
Normalized Sampling Period T	1[s]
Interpolation Factor Range M	$6 - 15$

The first step is to determine the optimal interpolation factor M, and this is done in a similar fashion as in the previous sections. M turns out to be seven, the value of K from Table 2.1 turns out to be 1 and the overall FRM digital filter is of the type Case II.

Using MATLAB routine "ellipord", the order of interpolation digital subfilter $H_a(z)$ turns out to be $N_{_{IIR}} = 5$. The next step is to calculate the required edge frequencies of $H_a(z)$, and masking digital subfilters $F_0(z)$ and $F_1(z)$. Eqns. (4.25-4.27) are used to get the final $\hat{\hat{f}}_{3\text{dB}}$ frequency. This $\hat{\hat{f}}_{3\text{dB}}$ frequency allows an exact realization of $\hat{\alpha}_i$, without any need to quantize it any further. The final $f_{3\text{dB}}$ frequency is used to calculate the exact values of the passband edge frequency θ and stopband edge ϕ frequency of the digital interpolation subfilter $H_a(e^{j\omega})$ in accordance with Eqns. (4.28) and (4.29). Table 2.1 can now be used to get the final edge frequencies of all the digital subfilters and the overall FRM digital filter, and the resulting requirements are given in Table 5.15.

$F_0(z)$ is zero-padded by four zeros on each end. The overall FRM digital filter $H(z)$ has a normalized passband edge $\omega_p = 0.2045\pi$ and a normalized stopband edge $\omega_a = 0.2358\pi$.

The values of the multiplier coefficients of allpass network $G_{0\,\text{or}\,1}(z)$, $\alpha_{0,1}$ and $\beta_{0\,\text{or}\,1,i}$ can be obtained as summarized in Table 5.16.

Note here that $\hat{\alpha}_{0\,\text{or}\,1,i}$ is fixed at -0.1367, and is not included as a variable in

Table 5.15: Example 4: Required Subfilter Requirements

Subfilter	Order	R_p [dB]	R_a [dB]	Passband Edge	Stopband Edge
$H_a(e^{j\omega})$	5	0.0007	38	1.1912	1.8850
$F_0(e^{j\omega})$	34	0.05	38	0.2551	0.6425
$F_1(e^{j\omega})$	42	0.05	38	0.7408	1.0544

Table 5.16: Example 4: Digital Multiplier Coefficient Values

Variables	Values
$\alpha_{0,1}$	-0.0687
$\beta_{0,1}$	0.7028
$\beta_{1,1}$	0.2283

the DCGA optimization.

Next, the LUTs for DCGA optimization are generated. Two CSD LUTs are required, one LUT with entries belonging to $CSD(11,3)$ (with 10-bit fractional part) for the multiplier coefficients $\alpha_{0,1}$, $\beta_{0,1}$, and $\beta_{1,1}$ constituent in the interpolation digital subfilters $H_a(z)$ and $H_b(z)$, and one LUT with entries belonging to $CSD(11,3)$ (with 10-bit fractional part) for all the multiplier coefficients constituent in the masking digital subfilters $F_0(z)$ and $F_1(z)$.

The DCGA optimization converged in around 300 generations to a desired FRM digital filter with the CSD multiplier coefficients $\hat{\alpha}_1$, and $\hat{\beta}_{0\,\mathrm{or}\,1,i}$ as given in Table 5.17.

Table 5.17: Example 4: Finite-Precision Multiplier Coefficient Values After DCGA Optimization

Multiplier	CSD Representation	Decimal Value
$\hat{\alpha}_1$	$0.000\bar{1}00\bar{1}001$	-0.0693
$\hat{\beta}_{0,1}$	$1.0\bar{1}0\bar{1}000000$	0.6875
$\hat{\beta}_{1,1}$	$0.0100\bar{1}01000$	0.2266

Note that $\hat{\alpha}_{0\,\mathrm{or}\,1,i} = -0.1367(0.00\bar{1}00\bar{1}0100)$ throughout the optimization procedure.

The analysis of the seed FRM digital filter before DCGA optimization and optimized FRM digital filter after DCGA optimization revealed frequency response characteristics as given in Table 5.18.

Table 5.18: Example 4: Frequency-Response Analysis Before and After DCGA Optimization

Frequency-response characteristic	Before DCGA	After DCGA
Maximum Passband Ripple A_p	0.1310[dB]	0.0411[dB]
Minimum Stopband Loss A_a	28.8881[dB]	38.1820[dB]
Maximum Normalized Group-Delay τ	102.2921	102.5812

The magnitude and group-delay frequency-responses of the interpolation digital subfilters before and after the DCGA optimization are obtained as shown in Fig. 5.12 and Fig. 5.13.

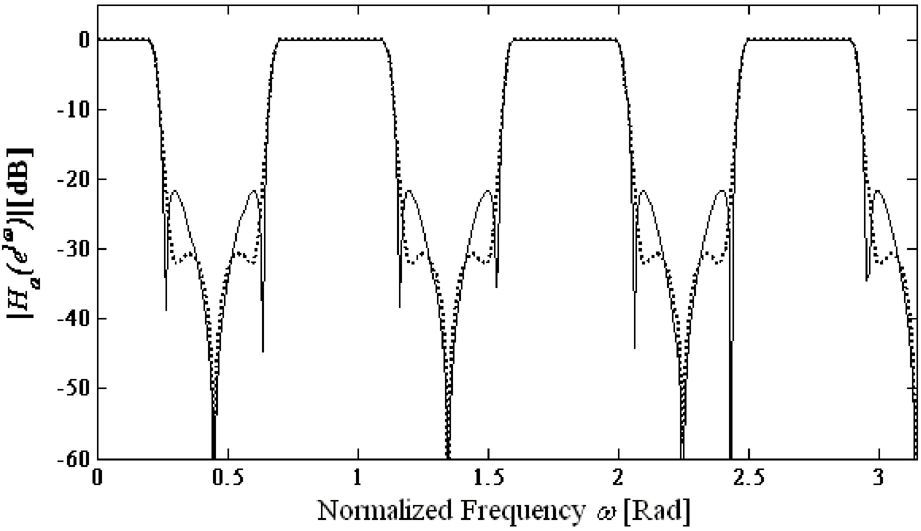

Figure 5.12: Example 4: $H_a(e^{j\omega})$ Magnitude Response: Non-optimized (dash) / Optimized (solid)

The magnitude and group-delay frequency-responses of the FIR digital subfilters before and after the DCGA optimization are obtained as shown in Fig. 5.14 and Fig. 5.15

The magnitude and group-delay frequency-responses of the FRM digital filter before and after the DCGA optimization are obtained as shown in Fig. 5.16 and Fig. 5.17

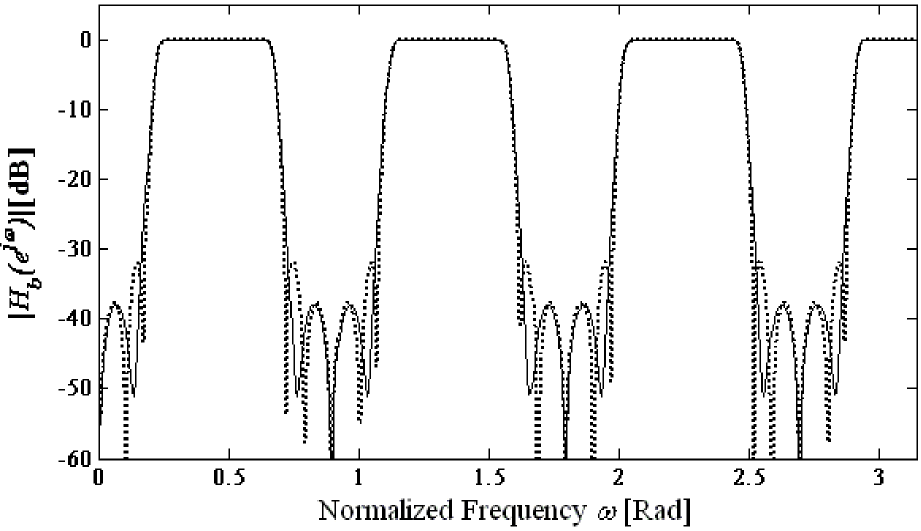

Figure 5.13: Example 4: $H_b(e^{j\omega})$ Magnitude Response: Non-optimized (dash) / Optimized (solid)

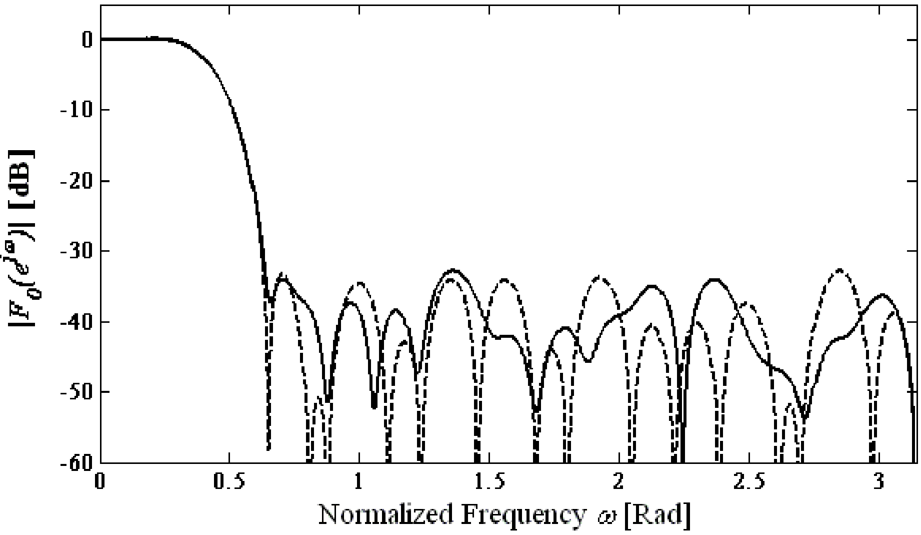

Figure 5.14: Example 4: $F_0(e^{j\omega})$ Magnitude Response: Non-optimized (dash) / Optimized (solid)

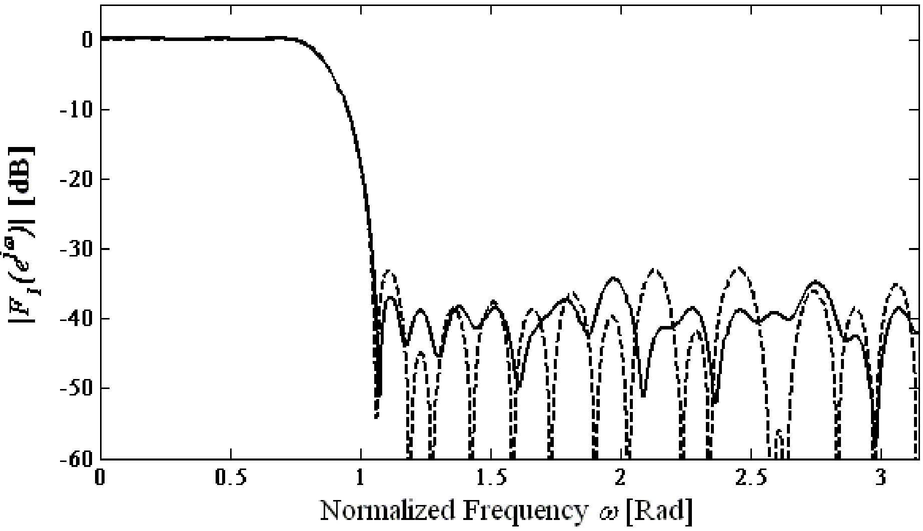

Figure 5.15: Example 4: $F_1(e^{j\omega})$ Magnitude Response: Non-optimized (dash) / Optimized (solid)

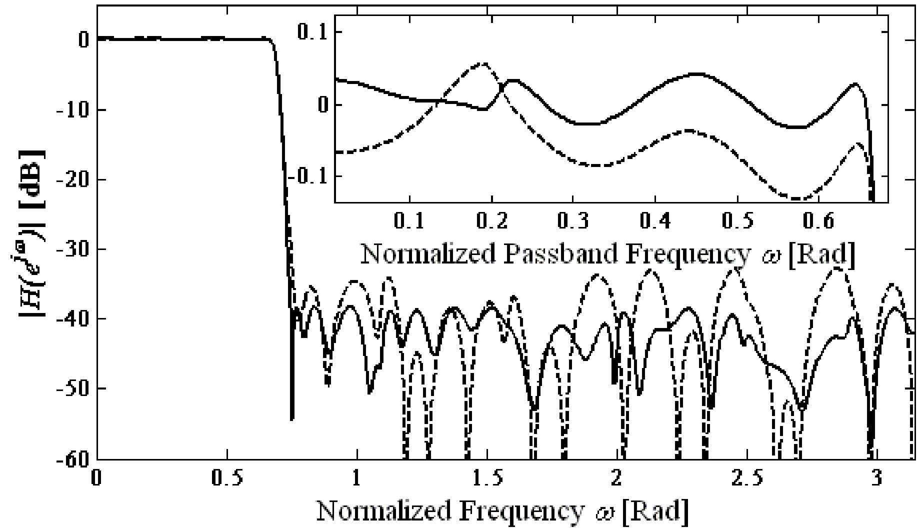

Figure 5.16: Example 4: $H(e^{j\omega})$ Magnitude Response: Non-optimized (dash) / Optimized (solid)

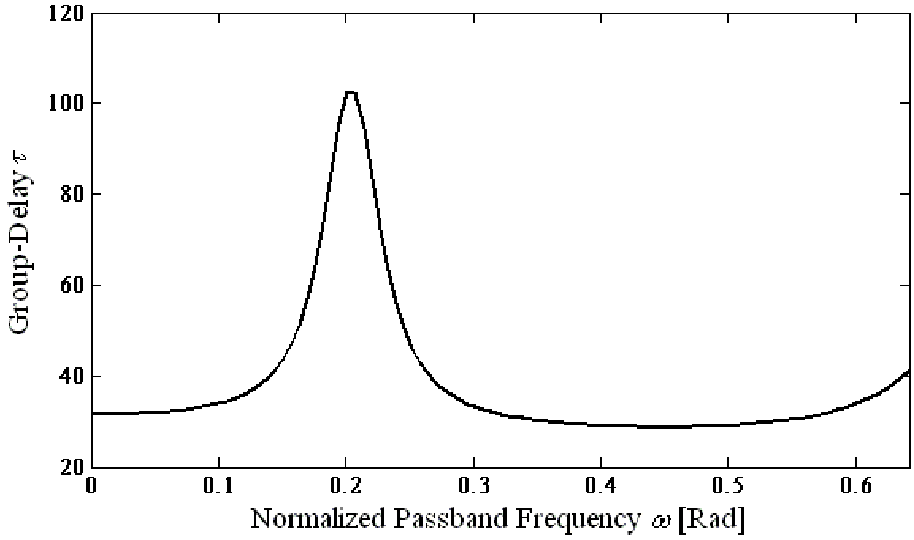

Figure 5.17: Example 4: $H(e^{j\omega})$ Group Delay: Non-optimized (dash) / Optimized (solid)

5.5 Application Example 5

This section is concerned with the design of a lowpass FRM digital filter satisfying the specifications given in Table 5.19. Lattice WDF design technique is to be used here. The design specifications in this example match those of Section 5.2 and Section 5.3.

Table 5.19: Example 5: Design Specifications

Maximum Passband Ripple A_p	0.1[dB]
Minimum Stopband Loss A_a	40[dB]
Passband-Edge Normalized Frequency ω_p	0.30π[Rad]
Maximum Stopband-Edge Normalized Frequency ω_a	0.31π[Rad]
Normalized Sampling Period T	1[s]
Interpolation Factor M	$6-15$

Similar to that in Section 5.4, the above specifications and Table 2.1 are used to calculate the interpolation factor M as well as the approximate passband edge frequency $\tilde{\theta}$ and stopband edge $\tilde{\phi}$ frequency of the digital interpolation subfilter $H_a(z)$. In this example, $M = 7$. Also, the value of K from Table 2.1 turns out to be 2 and the overall FRM digital filter is of the type Case II.

Using the design procedure in the previous section, the order of $H_a(z)$ turns out to be $N_{IIR} = 7$. Next, the exact values of the passband edge frequency θ and stopband edge ϕ frequency of $H_a(z)$, in accordance with Eqns. (4.28) and (4.29). The final edge frequencies of all the digital subfilters and the overall FRM digital filter are given in Table 5.20.

Table 5.20: Example 5: Required Subfilter Requirements

Subfilter	Order	R_p [dB]	R_a [dB]	Passband Edge	Stopband Edge
$H_a(e^{j\omega})$	7	0.0004	40	1.1912	1.8850
$F_0(e^{j\omega})$	20	0.1	40	1.2467	1.8948
$F_1(e^{j\omega})$	40	0.1	40	1.9395	2.2493

$F_0(z)$ is zero-padded by ten zeros on each end. The overall FRM digital filter $H(z)$ has a normalized passband edge $\omega_p = 0.6031\pi$ and a normalized stopband edge $\omega_a = 0.6173\pi$.

The values of the multiplier coefficients of allpass network $G_{0\,\mathrm{or}\,1}(z)$, $\alpha_{0,1}$ and $\beta_{0\,\mathrm{or}\,1,i}$ can be obtained as summarized in Table 5.21.

Table 5.21: Example 5: Digital Multiplier Coefficient Values

Variables	Values
$\alpha_{0,1}$	-0.2619
$\beta_{0,1}$	0.6016
$\beta_{1,1}$	0.2575
$\beta_{1,2}$	0.8788

Note that $\hat{\alpha}_{0\,\mathrm{or}\,1,i}$ is fixed at -0.4902. Next, the LUTs for DCGA optimization are generated. Two CSD LUTs are required, one LUT with entries belonging to $CSD(12, 3)$ (with 10-bit fractional part) for the multiplier coefficients $\alpha_{0,1}$, $\beta_{0,1}$, $\beta_{1,1}$, and $\beta_{1,2}$ constituent in the interpolation digital subfilters $H_a(z)$ and $H_b(z)$, and one LUT with entries belonging to $CSD(11, 3)$ (with 10-bit fractional part) for all the multiplier coefficients constituent in the masking digital subfilters $F_0(z)$ and $F_1(z)$.

The DCGA optimization converged in 300 generations to a desired FRM digital filter with the CSD multiplier coefficients $\hat{\alpha}_1$, and $\hat{\beta}_{0\,\mathrm{or}\,1,i}$ as given in Table 5.22.

Table 5.22: Example 5: Finite-Precision Multiplier Coefficient Values After DCGA Optimization

Multiplier	CSD Representation	Decimal Value
$\hat{\alpha}_1$	$0.0\bar{1}00000\bar{1}000$	-0.2539
$\hat{\beta}_{0,1}$	0.10010000001	0.5630
$\hat{\beta}_{1,1}$	0.01000010001	0.2583
$\hat{\beta}_{1,2}$	$1.00\bar{1}0000\bar{1}000$	0.8711

Note that $\hat{\alpha}_{0\,\mathrm{or}\,1,i} = -0.4902(0.\bar{1}0000010100)$. The analysis of the seed FRM digital filter before DCGA optimization and optimized FRM digital filter after DCGA optimization revealed frequency response characteristics as given in Table 5.23.

Table 5.23: Example 5: Frequency-Response Analysis Before and After DCGA Optimization

Frequency-response characteristic	Before DCGA	After DCGA
Maximum Passband Ripple A_p	0.2166[dB]	0.0959[dB]
Minimum Stopband Loss A_a	34.6919[dB]	40.2657[dB]
Maximum Normalized Group-Delay τ	74.9322	69.5239

The magnitude and group-delay frequency-responses of the FRM digital filter before and after the DCGA optimization are shown in Fig. 5.18 and Fig. 5.19

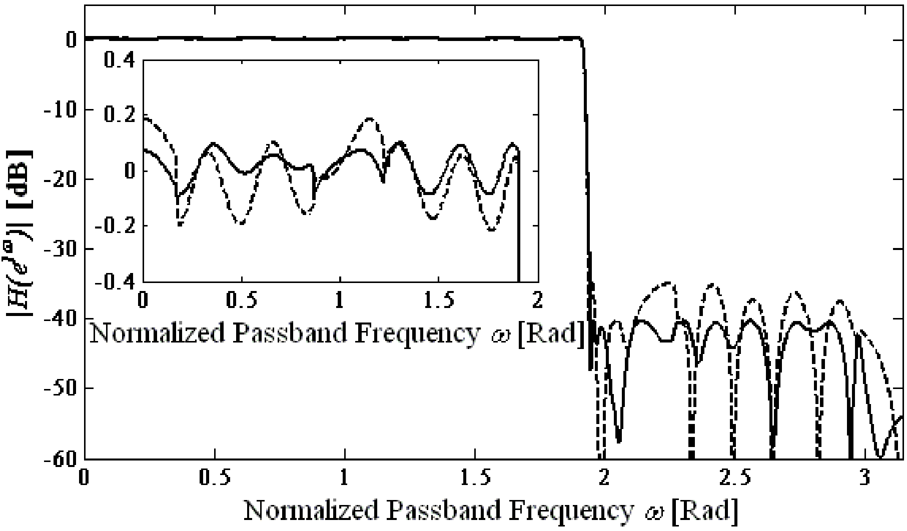

Figure 5.18: Example 5: $H(e^{j\omega})$ Magnitude Response: Non-optimized (dash) / Optimized (solid)

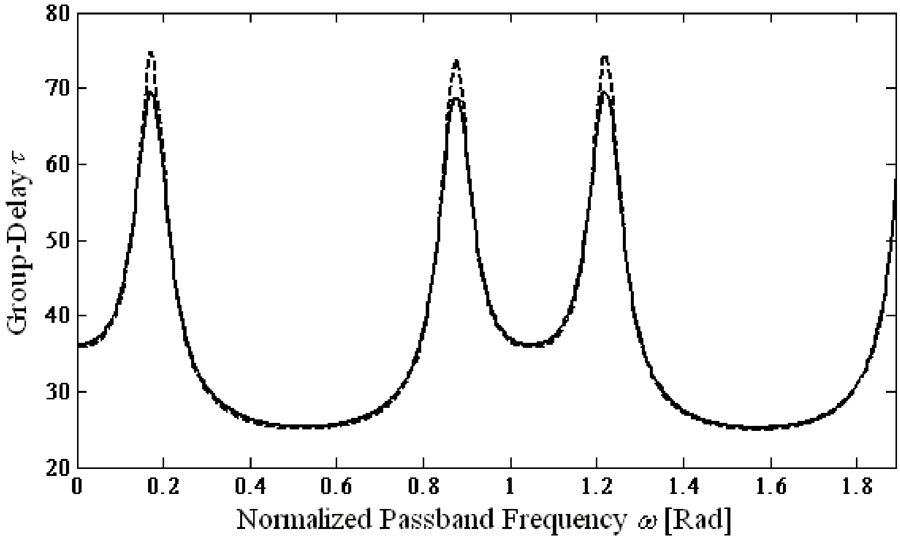

Figure 5.19: Example 5: $H(e^{j\omega})$ Group Delay: Non-optimized (dash) / Optimized (solid)

5.6 Summary

This section presented the application of DCGA optimization to bilinear-LDI and lattice WDF based FRM digital lowpass filters.

In the first two examples, the LUT scheme utilizing slack variables was used. From Example 1 to Example 2, the specifications increase in the passband frequency region, but at the same time the required transition bandwidth is reduced. This caused certain issues. As the passband frequency region gets larger, it usually results in less options for optimizing the masking digital subfilter coefficients (see Section 2.3). At the same time, the narrower the required transition band, the higher is the order of interpolation digital subfilter. It is empirically observed that higher order interpolation filters also tend to have larger coefficient values (and more aspect ratio), and this too causes problems in the DCGA optimization. As a result of the above mentioned problems, a larger coefficient wordlength is used in Example 2. Even then, the convergence speed is slower for Example 2 than Example 1.

The simplifications made to the LUT scheme in Example 3 result in significant improvement to the DCGA convergence speed compared to the convergence speed in Example 2.

The last two examples are based on lattice WDF FRM digital filters. Since half the interpolation digital subfilter coefficients are fixed and not part of the optimization procedure, lattice WDF FRM digital filters exhibited slower convergence speeds compared to bilinear-LDI FRM digital filters. Note should be made, however, that even though the specifications of Example 1 and Example 4, and Example 2 and Example 5 are exactly the same, the two realizations are not directly comparable. This is because the final edge frequencies do not match exactly, and the FIR masking digital filter orders are not the same.

The group-delay frequency response varies for bilinear-LDI FRM digital filters and lattice WDF FRM digital filters based on the given specifications.

Chapter 6

Conclusions

6.1 Conclusions

This thesis has been concerned with the design and discrete optimization of two different types of FRM digital filters employing IIR interpolation digital subfilters. The first type of FRM digital filter employed interpolation subfilters that were realized using an addition and subtraction of allpass digital networks that were implemented using bilinear-LDI realization technique.

The second type of FRM digital filter employed interpolation subfilters that were also realized using addition and subtraction of allpass digital networks, but that were implemented using lattice WDF realization technique. In the latter case, half of the most sensitive coefficients were fixed to an easily implementable value, so as to avoid distortion due to quantization error.

The optimization of IIR based FRM digital filters can be achieved by employing either the existing gradient-based optimization techniques or conventional GAs. The latter approach is well known for its effectiveness and efficiency in solving complex multimodal optimization problems over a discrete solution space. However, the conventional GAs suffer from low convergence speed problems that usually have a high tendency to converge towards a local optimal point. Therefore, this thesis has incorporated diversity control in the conventional GAs for the rapid optimization of FRM digital filters. DCGA has the advantage of increasing the diversity of the population pool through the incorporation of additional non-elite chromosomes (chosen based on their hamming distance from the chromosome with the best fitness value). DCGA optimization was used in this thesis to optimize the FRM digital filters over the CSD multiplier coefficient space, so that the resulting optimized filter would be directly implementable in hardware.

A direct application of DCGA to optimize a bilinear-LDI based FRM digital filter would result in two separate problems: first, the chromosomes may not conform to the CSD number format, and second, the resulting filter may not be BIBO stable. This thesis resolved the latter problem by developing novel stability constraints directly on the CSD multiplier coefficients. The constraints are developed for an arbitrary-order bilinear-LDI allpass network. Further, a novel worst-case LUT scheme is developed in order to simplify previous techniques involving slack-variables. The resulting DCGA optimization is shown to be automatically BIBO stable. Finally, a novel cost-function is utilized so as to simultaneously optimize the FRM digital filter in both the magnitude-frequency and group-delay frequency response.

DCGA optimization was also applied to lattice WDF based FRM digital filters. Half the IIR interpolation subfilter multiplier coefficients were removed through the use of filters having EMQF transfer functions. The cost-function here too carried out a simultaneous optimization of the FRM digital filter in both the magnitude-frequency and group-delay frequency response.

In both cases, the usefulness of the proposed DCGA optimization has been demonstrated through its application to the design of a lowpass FRM digital filter satisfying stringent specifications. The results of the DCGA optimization have been discussed.

6.2 Summary of Contributions

- This thesis has presented a novel IIR based FRM digital filter incorporating interpolation digital subfilters realized using bilinear-LDI design technique.

- A step-by-step procedure for the design of FRM digital filter incorporating interpolation digital subfilters realized using bilinear-LDI design technique is presented.

- A novel set of stability criteria are developed in this thesis that guarantee the BIBO stability of a general order bilinear-LDI allpass digital network. The stability criteria are based directly on CSD multiplier coefficients, allowing automatic BIBO stability throughout the course of DCGA optimization in a computationally efficient manner.

- A novel worst-case LUT scheme is introduced in this thesis, that simplifies the previous techniques required to ensure automatic BIBO stability throughout the course of DCGA optimization. The worst-case LUT scheme is demonstrated to yield faster optimization convergence rates.
- A novel cost-function is developed that simultaneously optimizes both the magnitude-frequency and group-delay frequency response. The group-delay frequency response is calculated efficiently using adjoint networks technique.
- The usefulness of DCGA optimization of bilinear-LDI based FRM digital filters has been demonstrated.
- The usefulness of DCGA optimization of lattice WDF based FRM digital filters has been demonstrated.
- A novel adaptive DCGA is introduced in Appendix 1 that is capable of externally *generating* diversity.
- The design and DCGA optimization of a IIR Nyquist filter is detailed in Appendix 2. This design is novel in that the finite-wordlength representation does not require any time domain optimization (i.e., it ensures perfect zero-crossings), and only frequency domain optimization is required.

6.3 Suggestions for Future Work

Further work involves the improvement of DCGA convergence speed through dynamic adjustments of the shape coefficient c and the exponent α. It may be reasonable to encode those two parameters as chromosomes in the process of DCGA optimization. This will increase the length of the chromosome, but may lead to noticeable reduction in the time spent on empirical investigations.

This thesis has investigated the bilinear-LDI ladder and lattice WDF digital filter realization techniques. Future work involves different realization approaches of the FRM digital filter (e.g. the bilinear-LDI digital Jaumann filter realization approach).

Appendix A

Adaptive DCGA

While DCGA offers large improvements in convergence speed compared to conventional GAs, it still lacks in some aspects. One of the main issues with DCGA is that while it is capable of maintaining a threshold level of diversity in the population pool, it does not include any mechanism that actively generates diversity in the population pool. Mutation does allow some exploration, but in the case where the population pool starts to converge to a local optimum, the mutation rate is often too low to effectively steer the algorithm out of a premature convergence. If the mutation rate is increased too much, the algorithm becomes too random and this adversely affects the rate of convergence of the DCGA.

A second issue arises from the way in which DCGA determines the diversity of the population pool. The CPSS uses hamming distance as a measure of diversity between the chromosome that has the highest fitness value and the remaining chromosomes in the population pool. Hamming distance counts the number of bit locations at which a given chromosome is different from another chromosome.

Unfortunately, hamming distance is not directly proportional to the diversity between chromosomes. That is to say, a large hamming distance might correspond to a small actual difference between two chromosome, whereas a small hamming distance might actually correspond to a large actual difference between two chromosome. An example of this is shown in Fig. A.1.

Another issue with the CPSS is that it only uses the hamming distance between the chromosome that has the highest fitness value and the chromosome being evaluated as a criteria for selection. This means that the CPSS does not have a

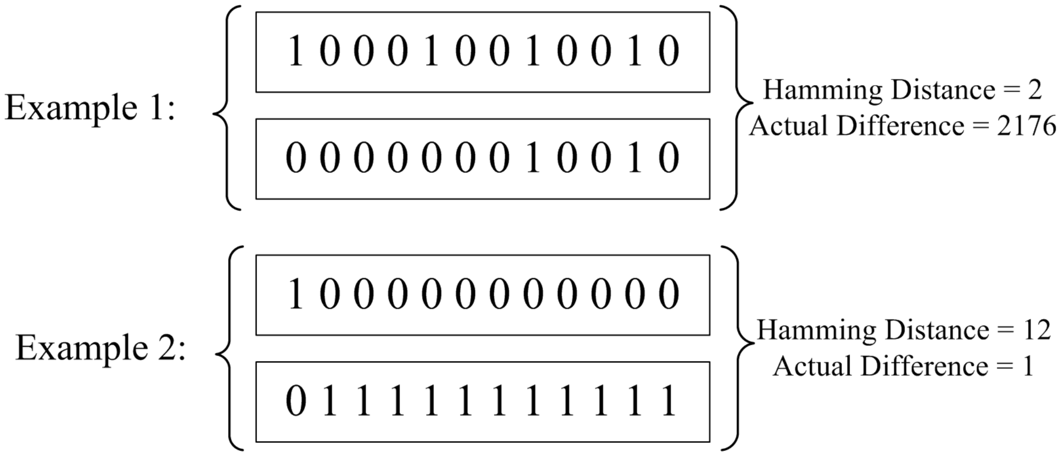

Figure A.1: Examples of Hamming Distance

mechanism to ensure that the chromosomes of a population pool are evenly spread out, but it simply ensures that the chromosomes selected for the next generation population pool differ from the best-fit chromosome. Consequently, even with the use of the CPSS it is possible to select a population pool with a low overall diversity.

DCGA also experiences difficulty in converging to the global optimum if the initial population pool is generated with a low overall diversity. The same is true if the seed chromosome is generated randomly or exhibits a low fitness (for example, if the multiplier coefficients of the seed chromosomes are not first optimized in infinite-precision using gradient-based techniques).

In such cases, in addition to a judicious selection of the shape coefficient c and the exponent parameter α (low value of c and/or high value of α, see Section 1.3), a high mutation rate is required to allow the DCGA to properly explore the solution space. This, however, is only required until the DCGA a region containing the global optimum. Once the DCGA starts to converge, a high mutation rate and low value of c and/or high value of α makes the algorithm unnecessarily explorative and actually reduces the convergence speed of the algorithm.

A.1 Adaptive DCGA Optimization Technique

In considering the above problems, a novel variation of the DCGA with an adaptive mutation operation is presented here. The main goal of this adaptive algorithm is to maintain a threshold level of diversity through the CPSS and also to *generate* diversity by a increasing the rate of mutation whenever the DCGA starts to stagnate and shows no improvement in the best-fit chromosome. At the same time, the mutation rate is reduced if the DCGA locates the region of the solution space containing the global optimum and begins to converge. The main steps of the adaptive DCGA are as follows:

- Initialization: This step is the same as in DCGA. A seed chromosome is formed by concatenating the design variables represented in their binary form. Subsequently, an initial population pool consisting of an even number of N chromosomes is generated by randomly complementing bits of the seed chromosome [19].

- Generation of the enlarged population pool: As in DCGA, the members of a current population pool $P(t)$ are randomly selected into $N/2$ parent pairs. Each parent pair then undergoes a two-point crossover operation to produce two offspring that are added to $P(t)$ to produce an enlarged population $\hat{P}(t)$. The crossover operation here is similar to the one discussed in Section 1.2, with one exception. The two bit-locations that correspond to the crossover points are not selected randomly from the entire chromosome length. Instead, the crossover points are selected randomly from only those bit-locations of the chromosome that do not sever a bit-string representing a design variable (such as a multiplier coefficient). That is to say, the crossover points can be placed at the start of end of a bit-string representing a design variable, but cannot be placed in between the length of the design variable, as shown in Fig. A.2.

  The reasoning behind this change is that if the crossover points are randomly selected from the entire chromosome length, then its possible to produce an offspring that contains a design variable value that is not present in any chromosome of the original population pool $P(t)$ (effectively resulting in gene mutation). Since an adaptive mutation operation is already present in the algorithm to generate any necessary diversity, the diversity resulting from the crossover operation is no longer required, and may even reduce the convergence rate when low mutation rates are required.

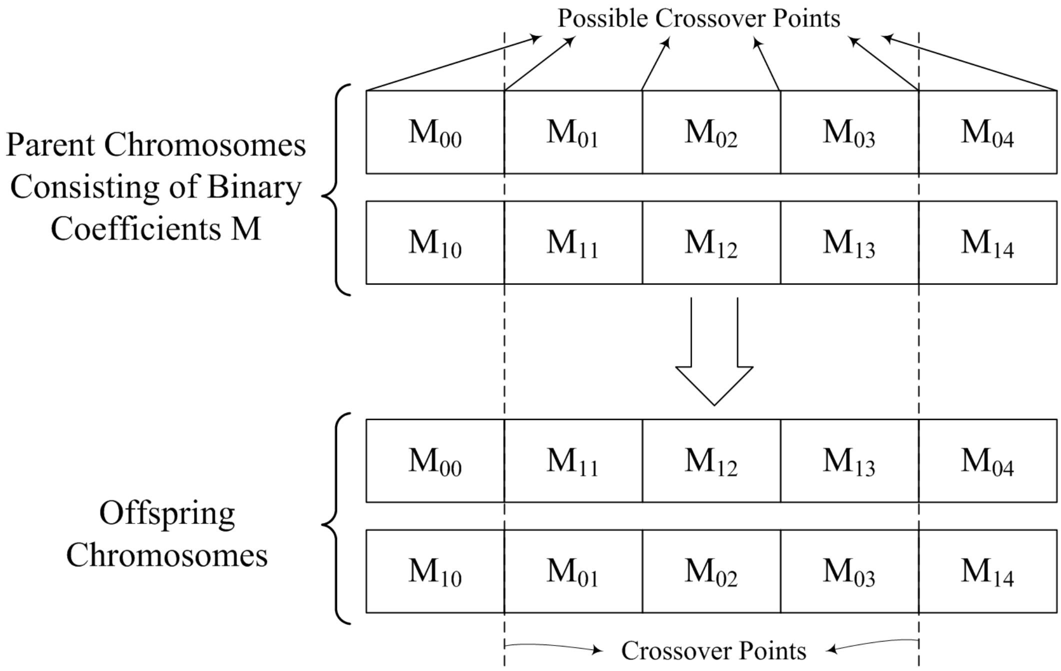

Figure A.2: Example of a Two-Point Crossover Based on Coefficient Edges

- Adaptive mutation: The mutation probability is now determined based on the variance of the fitness of the current population pool $P(t)$. Note here that in the first iteration of the adaptive DCGA the fitness of $P(t)$ is not yet determined, and so the mutation probability is kept at zero. The reason for using variance of the population fitness is that its a single simple calculation based on fitness values and it provides a fairly good indication as to the level of diversity in the population pool. If the fitness variance is low, it most likely indicates that the chromosomes are very similar and thus all have the same fitness. Conversely, a large fitness variance indicates that a lot of non-elite chromosomes are included in the population pool. Consequently, if the variance of the fitness of a population pool falls below a particular threshold, the mutation probability is increased so as to generate a population pool with greater diversity, and vise versa. Additionally, the best-fit chromosome is kept exempt from the mutation operation, thus making the adaptive DCGA an elitist algorithm. The overall effect of such an algorithm on the fitness variance of the population through the course of DCGA is shown in Fig. A.3.

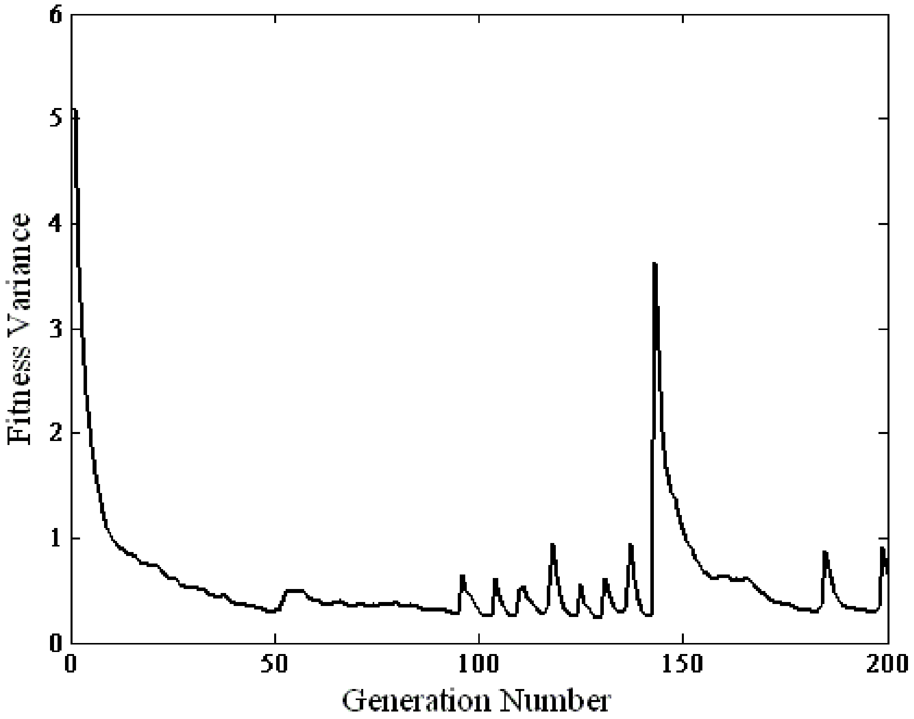

Figure A.3: Characteristic Profile of the Fitness Variance using Adaptive DCGA

Along with mutation probability, the strength of mutation is also made adaptive. Conventionally, the strength of a mutation refers to the frequency and location of the bit-flipping operation that defines a mutation: the strength of mutation increases as a bit is flipped from the least significant bit position of a design variable in a chromosome to the most significant bit position. However, in this case the strength of mutation refers to the change in the decimal value of the design variables in the chromosome. For example, if the value of a design variable shifts from the binary number 10000000 to 01111111, then it is considered a low strength mutation since the decimal value of the design variable shifted only by one.

The strength of the mutation is adapted based on two factors. The first factor is the average population of the population pool $P(t)$. If the average population fitness is low, it normally means that the solution space hasn't been properly explored yet. In that case, the strength of a mutation is a randomly selected from a large range of values (i.e. the average strength of mutation is increased). This allows the DCGA to start searching completely new and relatively distant regions of the solution space. However, as the average population fitness increases, it is often due to the fact that the region containing the global optimum is being searched by the DCGA. Consequently, the strength of a mutation is now selected randomly from a smaller range of values (i.e. the average strength of mutation is decreased), thus preventing the DCGA from exploring lower fitness regions of the solution space.

The second factor that is used to adaptively change the strength of mutation is the number of consecutive iterations that do not result in an improvement in the best-fit chromosome. If the DCGA begins to stagnate, and shows no improvement over successive generations, then the strength of mutation is increased in hopes of encouraging the DCGA into producing a new best-fit chromosome. If a new best-fit chromosome is obtained, the mutation strength is reduced to its previous level.

- Evaluation: Once the mutation operation is completed, the chromosomes in the enlarged population pool $\hat{P}(t)$ are evaluated to determine their fitness and ranked by their fitness values. This step is similar to the conventional DCGA technique.

- Selection of the next-generation population pool: This is done using the conventional CPSS technique described in Section 1.3. It should be noted that hamming distance is still used to ensure that a threshold of diversity is maintained in the next generation population pool $P(t+1)$ (i.e. fitness variance is not used here, it is only used for adapting the mutation rates).

- Possible Reset: Finally, even with a variable mutation probability and mutation strength, there are cases where the DCGA stagnates at a local optimum and that greatly reduces the speed of convergence. While the adaptive mutation and CPSS ensure the population pool is diverse, if the best-fit chromosome becomes dominant in the population pool there is no competition in the population pool. Therefore, the DCGA quickly converges back to the local optimum points. In the event that this happens, it is characteristic to see prolonged periods where the best-fit chromosome remains unchanged from one generation to the next.

If the adaptive DCGA begins to show signs of stagnation, the entire population pool is reset. A reset is carried out in exactly a similar manner as the generation of the initial population pool described in Initialization. However, instead of using the original seed chromosome, the reset population pool is formed by randomly flipping bits of the current best-fit chromosome. This is done so that the good information obtained from the best-fit chromosome is not completely lost. A critical part of the reset operation is that the best-fit chromosome used to form the reset population is not itself included in the population pool. This prevents the DCGA from being pulled back into the region containing the local optimum as a result of a dominant best-fit chromosome.

- Termination: The above steps are repeated until the design specifications are satisfied or the maximum number of allowed iterations is reached.

The adaptive DCGA is a simply an extension of the DCGA since it still maintains the CPSS scheme. However, it does include the following additional salient features:

- The crossover operation no longer has the potential to inherently produce mutations at the crossover points. Consequently, the crossover operation results in a rapid loss of diversity in the population pool. This means that the adaptive DCGA is much more rapid at converging at a global or a local optimum.

- The mutation rate is now increased when the population diversity goes below a pre-defined minimum, so as to increase the diversity of the population pool, and the mutation rate is decreased when the population diversity goes above a pre-defined value, so as to allow for a more rapid convergence to the global optimum value.

- The strength of mutation is now varied to ensure that the adaptive DCGA is less likely to stagnate at a local optimum value.

- The parameters on which the mutation operation is adapted (population fitness mean and variance) are easily calculated, and thus the adaptive DCGA is not significantly more computationally demanding than the conventional DCGA.

- The selection of the next-generation population pool uses the CPSS, and so the diversity of the population pool is maintained in terms of both the hamming distance between the best-fit chromosome and the rest of the chromosomes in the population pool (used in CPSS) as well as the fitness variance (used for adaptive mutation operation).
- The reset operation usually results in a lower average fitness of the population pool as well as a lower fitness value of the best-fitness chromosome. However, it is empirically observed to significantly increases the likelihood of pulling the adaptive DCGA away from a region containing a local optimum and thus preventing premature convergence.

In empirical investigations, the adaptive DCGA showed significant improvements from the conventional DCGA in two respects. Firstly, it converged to desired specifications in less iterations, and secondly, it showed itself more immune to cases of premature convergence. A FRM digital filter with seventh-ordered bilinear-LDI based IIR interpolation digital subfilters was optimized using both the conventional DCGA technique and the adaptive DCGA technique and the results are compared in Fig. A.4.

Fig. A.4 shows that the adaptive DCGA converges to a higher fitness value in 200 generations compared to the conventional DCGA technique, even though the conventional DCGA technique started off with a seed chromosome with a higher fitness value. Also, the dip seen in the maximum fitness value of the adaptive DCGA technique corresponds to a population reset when the algorithm was beginning to prematurely converge at a local optima.

The adaptive DCGA is particularly useful in cases where the seed chromosome is not optimized in the infinite-precision domain and thus has a low fitness value, or if the initial population pool having a low diversity is generated. Also, its convergence speed increases significantly when the solution space contains several local optima, or if the values of c and α are not optimized for a particular application through empirical investigations.

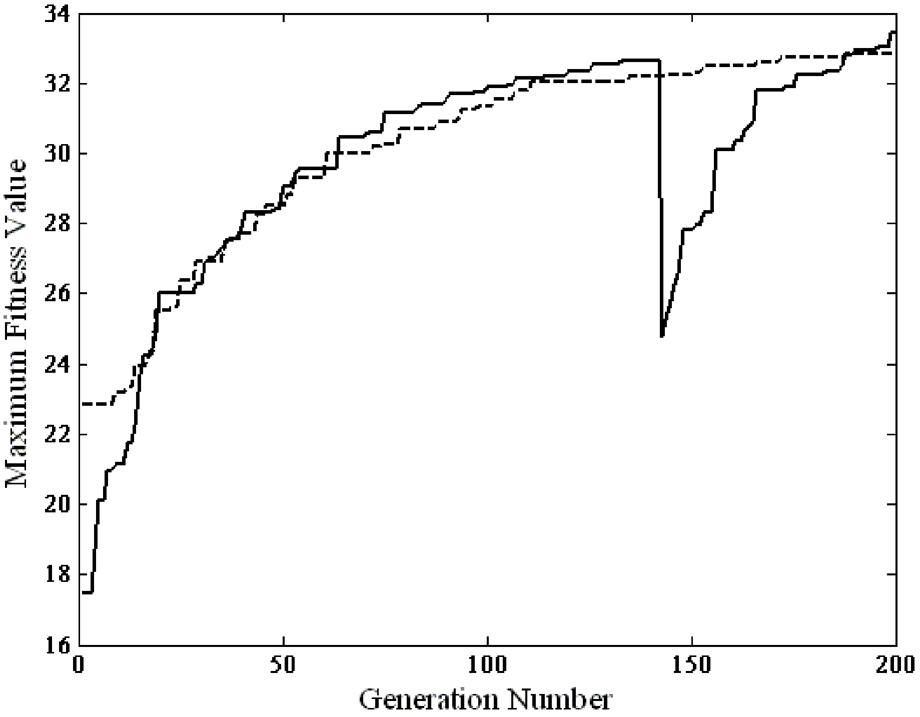

Figure A.4: Comparison of Convergence Speed Between Adaptive DCGA (solid) and Conventional DCGA (dashed)

Unfortunately, the adaptive DCGA is also associated with certain issues. An important issue comes from the fact that fitness variance is not directly proportional to the diversity of a population (as is the case with hamming distance). Also, since the absolute values of fitness mean and fitness variance depend on the application being optimized, the parameters for adaptive mutation (for example, the fitness variance threshold to increase mutation probability) have to be optimized for a particular application (as is the case with c and α). However, parameters such fitness mean and fitness variance could be normalized and thus used for any general application, but this is subject to future investigations. Finally, by making the DCGA an adaptive algorithm, more parameters are required. The issue with having more optimization parameters is that they are normally arbitrarily set and thus need to be properly set to ensure best results.

References

[1] Y. Lim, “Frequency-Response Masking Approach for the Synthesis of Sharp Linear Phase Digital Filters,” *IEEE Transactions on Circuits and Systems*, vol. 33, no. 4, pp. 357–364, 1986.

[2] H. Johansson and L. Wanhammar, “High-speed Recursive Filtering Using the Frequency-Response Masking Approach,” in *Proceedings of the IEEE Int. Symposium on Circuits and Systems*, 1997, pp. 2208–2211.

[3] H. Shimodaira, “A Diversity-Control-Oriented Genetic Algorithm (DCGA): Performance in Function Optimization,” *Proceedings of 2001 Congress on Evolutionary Computation*, vol. 1, pp. 44–51, May 2001.

[4] P. Mercier *et al.*, “Optimization of FRM FIR Digital Filters Over CSD and CDBNS Multiplier Coefficient Spaces Employing a Novel Genetic Algorithm,” *Journal of Computers*, vol. 2, no. 7, pp. 20–31, Sept. 2007.

[5] B. Nowrouzian, “Minimal Multiplier Realisation of Bilinear-LDI Digital Allpass Networks,” in *IEE Proceedings on Devices and Systems, G Circuits*, vol. 136, Jun. 1989, pp. 114–117.

[6] Fettweiss *et al.*, “Wave Digital Lattice Filters,” *Int. Journal of Circuit Theory Appl*, pp. 203–211, June 1974.

[7] G. Temes, “Exact computation of group delay and its sensitivities using adjoint-network concept,” *Electronics Letters*, vol. 6, pp. 483–485, July 1970.

[8] M. Lutovac and L. Milid, “Design of Computationally Efficient Elliptic IIR Filters with a Reduced Number of Shift-and-Add Operations in Multipliers,” *IEEE Trans. on Signal Processing*, vol. 45, no. 7, pp. 2422–2430, Oct. 1997.

[9] ——, "IIR Filters Based on Frequency-Response Masking Approach," in *Telecommunications in Modern Satellite, Cable and Broadcasting Service, TELSIKS 2001*, vol. 1, Sept. 2001, pp. 163–170.

[10] B. L. R. Ansari, "A Class of Low-Noise Computationally Efficient Recursive digital Filters with Applications to Sampling Rate Alterations," *IEEE Trans. on Acoust., Speech, Signal Processing*, vol. ASSP-33, pp. 90–97, Feb 1985.

[11] Y. Wu, "Design and Optimization of Analog and Digital IF Filters Employing a Diversity Controlled Genetic Algorithm," Master's thesis, University of Alberta, Edmonton, Canada, 2008.

[12] T. Parks and J. McClellan, "Chebyshev Approximation for Nonrecursive Digital Filters with Linear Phase," *IEEE Trans. Circuit Theory*, vol. vol. CT-19, no. 2.

[13] S. Kirkpatrick *et al.*, "Optimization by Simulated Annealing," *Science*, vol. 220, no. 4598.

[14] J. H. Yu and Y. Lian, "Design Equations for Jointly Optimized Frequency-ResposneMasking Filters," in *Ciucuits Systems and Signal Processing*, vol. 26, no. 1, Jan 2007, pp. 27–42.

[15] Y. J. Yu and Y. C. Lim, "Genetic Algorithm Approach for the Optimization of Multiplierless Sub-filters Generated by the Frequency Response Masking Technique," in *IEEE International Conference on Electronics, Circuits and Systems*, vol. 3, no. 1, 2002, pp. 1163–1166.

[16] Y. Wu and B. Nowrouzian, "Application of Diversity Controlled Genetic Algorithms to the Design and Optimization of Bilinear Digital LDI IF Filters," in *Proceedings of the IEEE Int. Symposium on Circuits and Systems*, May.

[17] F. Busetti, "Genetic algorithms overview," available: citeseer.ist.psu.edu/busetti01genetic.html.

[18] D. E. Goldberg, *Genetic Algorithms in Search, Optimizaito, and Maching Learning.* Reading, MA: Addison-Wesley, 1989.

[19] S. Kilambi, "Design and Optimization of FRM FIR Digital Filters using Genectic Algorithms over the CSD and DBNS Multiplier Coefficient Spaces," Master's thesis, University of Alberta, Edmonton, Canada, 2007.

[20] S. Kilambi and B. Nowrouzian, "A Novel Genetic Algorithm for Optimization of FRM Digital Filters over DBNS Multiplier Coefficient Space Based on Correlative Roulette Selection," in *IEEE International Symposium on Signal Processing and Information Technology*, Aug. 2006, pp. 228–231.

[21] S. Kilambi *et al.*, "A Novel Diversity Controlled Genetic Algorithm for Rapid Optimization of Bandpass FRM FIR Digital Filters Over CSD Multiplier Coefficient Space," in *Journal of Circuits, Systems and Signal Processing (CSSP)*, accepted for publication.

[22] Srinivas.M *et al.*, "Adaptive probabilities of crossover and mutation in genetic algorithms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 24, pp. 656–667, April 1994.

[23] L. Cen and Y. Lian, "Hybrid Genetic Algorithm for the Design of Modified Frequency-Response Masking Filters in a Discrete Space," *Circuits, Systems, and Signal Processing*, vol. 25, pp. 153–174, April 2006.

[24] A. Willson and H. Orchard, "Insights into Digital Filters Made as the Sum of Two Allpass Functions," *IEEE Trans. On Circuits And Syst.*, vol. 42, pp. 129–137, Mar. 1995.

[25] D. Rabrenovic and M. Lutovac, "Elliptic filters with minimal Q-factors," in *IEE Electronics Letters Online*, vol. 30, no. 3, Feb. 1994, pp. 206–207.

[26] M. Lutovac and L. Milid, "Design of Multiplierless Elliptic IIR Filters with a Small Quantization Error," *IEEE Trans. on Signal Processing*, vol. 47, no. 2, pp. 469–479, Feb. 1999.

[27] B. Nowrouzian, "A Novel Approach to the Exact Design of LDI Symmetrical Digital and Switched-Capacitor Filters," in *Proceedings of 33rd Midwest Symposium on Circuits and Systems*, vol. 2, Aug. 1990, pp. 967–972.

[28] V. Valkenburg, *Introduction to Modern Network Synthesis.* John Wiley and Sons, Inc., 1965.

[29] A. Antoniou, *Digital Filters: Analysis, Design, and Applications.* McGraw Hill, Inc., 1993.

[30] L. Gazsi, "Explicit Formulas for Lattice Wave Digital Filters," *IEEE Trans. Circuits Syst*, vol. CAS-32, pp. 66–88, 1985.